

---

# **IroniC Documentation**

***Release 16.2.1.dev2***

**OpenStack Foundation**

**Feb 19, 2021**



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation Guide</b>	<b>3</b>
2.1	Bare Metal Service Installation Guide . . . . .	3
<b>3</b>	<b>Upgrade Guide</b>	<b>109</b>
3.1	Bare Metal Service Upgrade Guide . . . . .	109
<b>4</b>	<b>User Guide</b>	<b>121</b>
4.1	Bare Metal Service User Guide . . . . .	121
<b>5</b>	<b>Administrator Guide</b>	<b>129</b>
5.1	Drivers, Hardware Types and Hardware Interfaces . . . . .	129
5.2	Administrators Guide . . . . .	681
<b>6</b>	<b>Configuration Guide</b>	<b>1545</b>
6.1	Configuration Reference . . . . .	1545
<b>7</b>	<b>Bare Metal API References</b>	<b>1685</b>
7.1	REST API Conceptual Guide . . . . .	1685
<b>8</b>	<b>Command References</b>	<b>1699</b>
8.1	Command References . . . . .	1699
<b>9</b>	<b>Contributor Guide</b>	<b>1705</b>
9.1	Developers Guide . . . . .	1705
<b>10</b>	<b>Release Notes</b>	<b>4463</b>
	<b>Python Module Index</b>	<b>4465</b>
	<b>Index</b>	<b>4469</b>



## 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:

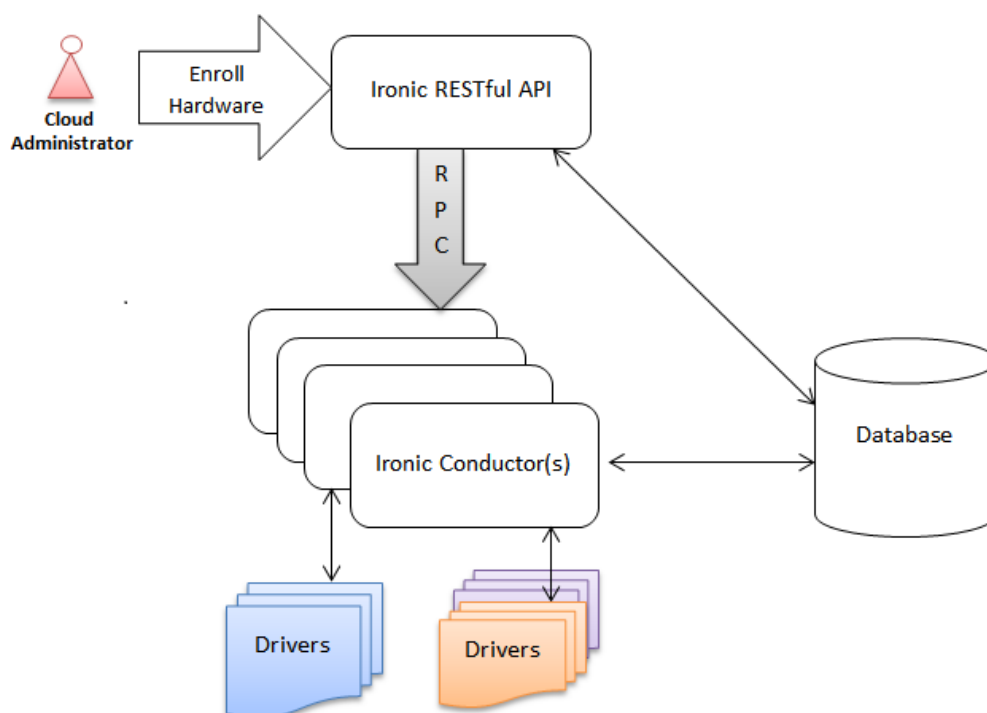
- A database to store hardware information and state. You can set the database back-end type and location. A simple approach is to use the same database back end as the Compute service. Another approach is to use a separate database back-end to further isolate bare metal resources (and associated metadata)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

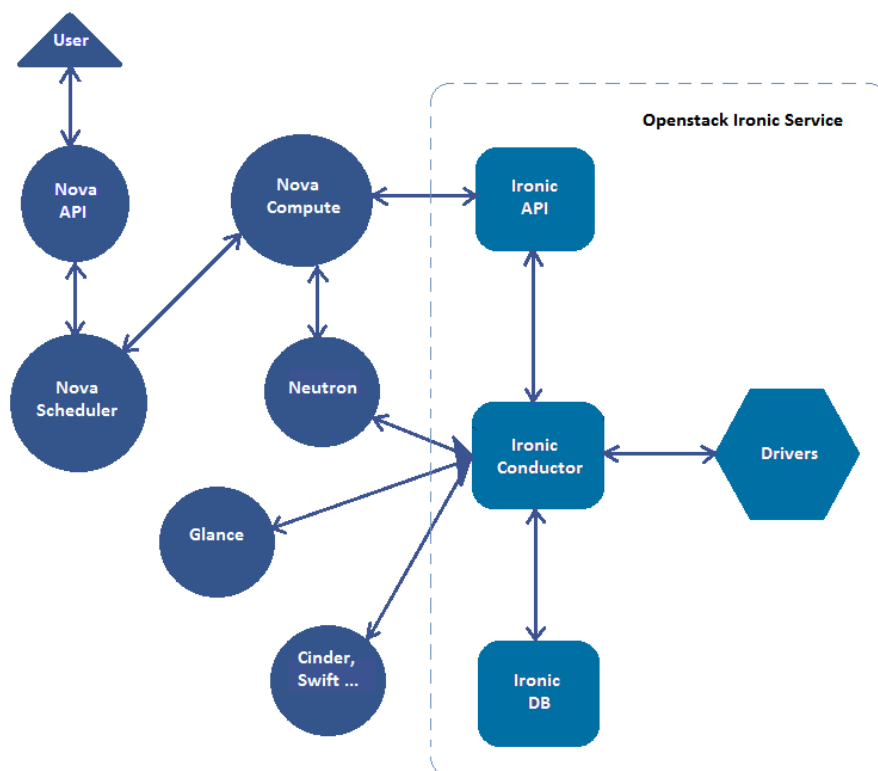


- the OpenStack Compute service (`nova`) works with the Bare Metal service and acts as a user-facing API for instance management, while the Bare Metal service provides the admin/operator API for hardware management. The OpenStack Compute service also provides scheduling facilities (matching flavors <-> 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 users 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<sup>1,2</sup> that reliability degrades when handling approximately 300 bare metal nodes per conductor.

---

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

<sup>2</sup> <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.
- The `iscsi` deploy method requires caching of the whole instance image locally during the deployment. The image has to be converted to the raw format, which may increase the required amount of disk space, as well as the CPU load.

---

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

---

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

---

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

---

## Other services

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

## Scenarios

### Small cloud with trusted tenants

#### Story

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

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



## Components

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

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

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

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

## Node roles

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

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

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

## Networking

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

### Control plane network

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

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

## Public network

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

---

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

---

## Bare metal network

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

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

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

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

## Management network

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

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

---

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

---

## Controllers

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

## Bare Metal services

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

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

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

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

---

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

---

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

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

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

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

## Shared services

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

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

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

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

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

---

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

---

### Bare metal nodes

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

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

### Storage

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

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

### 2.1.3 Install and configure the Bare Metal service

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

Note that installation and configuration vary by distribution.

#### Install and configure for Red Hat Enterprise Linux and CentOS

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

## Install and configure prerequisites

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

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

## Set up the database for Bare Metal

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

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

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

## Install and configure components

1. Install from packages

- Using `dnf`

```
# dnf install openstack-ironic-api openstack-ironic-conductor_
↪python-ironicclient
```

- Using `yum`

```
# yum install openstack-ironic-api openstack-ironic-conductor_
↪python-ironicclient
```

2. Enable services

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

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

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

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

## Configuring ironic-api service

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

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

```
[database]

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

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

```
[DEFAULT]

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

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

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

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

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

# Username (string value)
username=ironic

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

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

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

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

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

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

3. Configure the ironic-api service to use these credentials with the Identity service. Replace PUBLIC\_IDENTITY\_IP with the public IP of the Identity server, PRIVATE\_IDENTITY\_IP with the private IP of the Identity server and replace IRONIC\_PASSWORD with the password you chose for the ironic user in the Identity service:

```
[DEFAULT]

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

[keystone_authtoken]

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

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

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

# Service username. (string value)
username=ironic

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

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

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

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

4. Create the Bare Metal service database tables:

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

5. Restart the ironic-api service:

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

## Configuring ironic-api behind mod\_wsgi

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

1. Install the apache service:

RHEL7/CentOS7:

```
sudo yum install httpd
```

Fedora:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

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

Fedora/RHEL7/CentOS7:

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

Debian/Ubuntu:

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

SUSE:

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

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

Fedora/RHEL7/CentOS7:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:



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

SUSE:

```
sudo systemctl reload apache2
```

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

## Configure another WSGI container

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

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

If you want to pass a configuration file, use:

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

## Configuring ironic-conductor service

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

```
[DEFAULT]

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

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

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

```
[database]

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

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

**[DEFAULT]**

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

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

**[DEFAULT]**

```
rpc_transport = json-rpc
```

**[keystone\_authtoken]**

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

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

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

# Service username. (string value)
username=ironic

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

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

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

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

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

**[json\_rpc]**

```
host_ip = 192.168.0.10
port = 9999
```

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

4. Configure credentials for accessing other OpenStack services.

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

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

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

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

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

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

```
[neutron]

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

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

# Username (string value)
username=ironic

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

(continues on next page)

(continued from previous page)

```
password=IRONIC_PASSWORD

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

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

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

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

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

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

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

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

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

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

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

## Install and configure for Ubuntu

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

### Install and configure prerequisites

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

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

### Set up the database for Bare Metal

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

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

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

### Install and configure components

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

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

2. Enable services

Services are enabled by default on Ubuntu.

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

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

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

## Configuring ironic-api service

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

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

```
[database]

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

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

```
[DEFAULT]

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

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

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

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

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

# Username (string value)
username=ironic

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

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

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

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

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

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

3. Configure the ironic-api service to use these credentials with the Identity service. Replace PUBLIC\_IDENTITY\_IP with the public IP of the Identity server, PRIVATE\_IDENTITY\_IP with the private IP of the Identity server and replace IRONIC\_PASSWORD with the password you chose for the ironic user in the Identity service:

```
[DEFAULT]

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

[keystone_authtoken]

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

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

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

# Service username. (string value)
username=ironic

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

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

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

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

4. Create the Bare Metal service database tables:

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

5. Restart the ironic-api service:

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

## Configuring ironic-api behind mod\_wsgi

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

1. Install the apache service:

RHEL7/CentOS7:

```
sudo yum install httpd
```

Fedora:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

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

Fedora/RHEL7/CentOS7:

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

Debian/Ubuntu:

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

SUSE:

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

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

Fedora/RHEL7/CentOS7:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:



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

SUSE:

```
sudo systemctl reload apache2
```

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

## Configure another WSGI container

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

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

If you want to pass a configuration file, use:

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

## Configuring ironic-conductor service

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

```
[DEFAULT]

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

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

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

```
[database]

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

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

**[DEFAULT]**

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

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

**[DEFAULT]**

```
rpc_transport = json-rpc
```

**[keystone\_authtoken]**

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

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

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

# Service username. (string value)
username=ironic

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

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

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

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

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

**[json\_rpc]**

```
host_ip = 192.168.0.10
port = 9999
```

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

4. Configure credentials for accessing other OpenStack services.

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

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

For simplicity, you can use the same service user for all services. For backward compatibility, this should be the same user configured in the `[keystone_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)
```

(continues on next page)

(continued from previous page)

```
password=IRONIC_PASSWORD

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

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

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

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

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

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

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

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

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

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

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

## Install and configure for openSUSE and SUSE Linux Enterprise

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

---

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

---

## Install and configure prerequisites

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

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

## Set up the database for Bare Metal

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

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

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

## Install and configure components

1. Install from packages

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

2. Enable services

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

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

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

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

### Configuring ironic-api service

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

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

```
[database]

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

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

```
[DEFAULT]

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

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

```
[DEFAULT]

rpc_transport = json-rpc

[json_rpc]

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

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

# Username (string value)
username=ironic

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

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

(continues on next page)

(continued from previous page)

```
project_name=service

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

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

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

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

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

```
[DEFAULT]

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

[keystone_authtoken]

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

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

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

# Service username. (string value)
username=ironic

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

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

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

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

4. Create the Bare Metal service database tables:

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

5. Restart the ironic-api service:

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

## Configuring ironic-api behind mod\_wsgi

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

1. Install the apache service:

RHEL7/CentOS7:

```
sudo yum install httpd
```

Fedora:

```
sudo dnf install httpd
```

Debian/Ubuntu:

```
apt-get install apache2
```

SUSE:

```
zypper install apache2
```

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

Fedora/RHEL7/CentOS7:

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

Debian/Ubuntu:

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

SUSE:

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

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

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



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

Fedora/RHEL7/CentOS7:

```
sudo systemctl reload httpd
```

Debian/Ubuntu:

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

SUSE:

```
sudo systemctl reload apache2
```

---

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

---

## Configure another WSGI container

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

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

If you want to pass a configuration file, use:

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

## Configuring ironic-conductor service

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

### [DEFAULT]

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

---

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

---

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

```
[database]

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

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

```
[DEFAULT]

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

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

```
[DEFAULT]

rpc_transport = json-rpc

[keystone_authtoken]

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

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

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

# Service username. (string value)
username=ironic

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

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

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

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

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

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

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

#### 4. Configure credentials for accessing other OpenStack services.

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

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

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

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

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

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

```
[neutron]

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

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

# Username (string value)
username=ironic

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

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

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

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

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

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

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

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

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

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

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

6. Configure the network for ironic-conductor service to perform node cleaning, see [Node cleaning](#) from the admin guide.
7. Restart the ironic-conductor service:

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

### 2.1.4 Create user images for the Bare Metal service

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

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

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

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

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

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.
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_weighted_hosts = false
```

3. Carefully consider the following option:

```
[compute]

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

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

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

```
[ironic]

# Ironic authentication type
auth_type=password

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

# Ironic keystone project name
project_name=service

# Ironic keystone admin name
username=ironic

# Ironic keystone admin password
password=IRONIC_PASSWORD

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

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

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

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

(continues on next page)

(continued from previous page)

```
Ubuntu:
sudo service nova-scheduler restart
```

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

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

Ubuntu:
sudo service nova-compute restart
```

## Configure the Networking service for bare metal provisioning

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

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

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

---

**Note:** When the baremetal ML2 components are *not* used, ports in the Networking service will have status: DOWN, and binding\_vif\_type: binding\_failed. This was always the status for Bare Metal service flat network interface ports prior to the introduction of the baremetal ML2 integration. For a non-routed network, bare metal servers can still be deployed and are functional, despite this port binding state in the Networking service.

---

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

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

```
[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
```

(continues on next page)

(continued from previous page)

```
# Replace eth2 with the interface on the neutron node which you
# are using to connect to the bare metal server
```

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

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

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

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

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

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

9. Restart the Open vSwitch agent:

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

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

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

```
$ ovs-vsctl show

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

(continues on next page)

(continued from previous page)

```

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 Networking Service Ussuri release. It was also backported to the stable Train release.

---

## Creating networks and subnets in the Networking Service

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

---

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

---

## Configure the Image service for temporary URLs

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

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

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

---

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

---

2. Enable TempURLs for the Object Storage account used by the Image service for storing images in the Object Storage service.

1. Check if TempURLs are enabled:

```
# executed under credentials of the user used by Image service
# to access Object Storage service
```

(continues on next page)

(continued from previous page)

```
$ openstack object store account show
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| Account    | AUTH_bc39f1d9dcf9486899088007789ae643 |
| Bytes      | 536661727                               |
| Containers | 1                                        |
| Objects    | 19                                       |
| properties | Temp-Url-Key='secret'                   |
+-----+-----+
```

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

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

3. Optionally, configure the ironic-conductor service. The default configuration assumes that:
  1. the Object Storage service is implemented by `swift`,
  2. the Object Storage service URL is available from the service catalog,
  3. the project, used by the Image service to access the Object Storage, is the same as the project, used by the Bare Metal service to access it,
  4. the container, used by the Image service, is called `glance`.

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

```
[glance]

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

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

## Enabling HTTPS

### Enabling HTTPS in Swift

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

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



## Enabling HTTPS in Image service

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

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

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

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

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

## Enabling HTTPS communication between Image service and Object storage

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

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

1. [Enabling HTTPS in Swift](#)
2. [Configure Swift Storage Backend](#)
3. [Enabling HTTPS in Image service](#)

## Enabling HTTPS communication between Image service and Bare Metal service

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

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

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

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

---

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

---

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

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

3. Restart ironic-conductor service:

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

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

## Configure the Bare Metal service for cleaning

---

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

---

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

```
$ openstack network list
```

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

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

3. Restart the Bare Metal services ironic-conductor:

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

Ubuntu:
    sudo service ironic-conductor restart
```

## Configure tenant networks

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

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

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

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

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

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

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

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

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

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

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

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

---

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

---

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

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

---

**Note:** When using the `flat` network interface, bare metal instances are normally spawned onto the provisioning network. This is not supported with the `neutron` interface and the deployment

will fail. Please ensure a different network is chosen in the Networking service when a bare metal instance is booted from the Compute service.

---

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

- DHCP
  - TFTP
  - egress port used for the Bare Metal service (6385 by default)
  - ingress port used for ironic-python-agent (9999 by default)
  - if using *iSCSI deploy*, the ingress port used for iSCSI (3260 by default)
  - if using *Direct deploy*, the egress port used for the Object Storage service (typically 80 or 443)
  - if using iPXE, the egress port used for the HTTP server running on the ironic-conductor nodes (typically 80).
- 

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

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

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

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

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

5. Install and configure a compatible ML2 mechanism driver which supports bare metal provisioning for your switch. See [ML2 plugin configuration manual](#) for details.

6. Restart the ironic-conductor and ironic-api services after the modifications:

- Fedora/RHEL7/CentOS7:

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

- Ubuntu:

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

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

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

where FILENAME is the file located at the TFTP server.

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

## Add images to the Image service

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

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

For *partition images*:

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

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

Store the image uuid obtained from the above step as MY\_VMLINUZ\_UUID.

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

Store the image UUID obtained from the above step as MY\_INITRD\_UUID.

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

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

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

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

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

### 3. Build or download the deploy images

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

There are several methods to build or download deploy images, please read the [Building or downloading a deploy ramdisk image](#) section.

### 4. Add the deploy images to the Image service

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

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

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

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

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

## Create flavors for use with the Bare Metal service

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

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

#### 1. Change these to match your hardware:

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

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

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

---

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

---

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

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

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

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

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

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

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

## Example

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

```
$ 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
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
```

(continues on next page)

(continued from previous page)

```
enabled_power_interfaces = ipmitool,redfish
enabled_management_interfaces = ipmitool,redfish
```

**[DEFAULT]**

```
enabled_hardware_types = redfish
enabled_deploy_interfaces = iscsi
enabled_power_interfaces = redfish
enabled_management_interfaces = redfish
```

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

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

---

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

---

## Configuring interface defaults

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

**[DEFAULT]**

```
default_deploy_interface = direct
default_network_interface = neutron
```

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

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

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

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

## Configuring PXE and iPXE

### DHCP server setup

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

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

---

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

---

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

---

**Note:** Refer [Configure the Networking service for bare metal provisioning](#) for details about the two precedent steps.

---

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

---

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

---

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

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

---

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

---

### TFTP server setup

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

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

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

2. Install tftp server:

Ubuntu:

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

**RHEL7/CentOS7:**

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

**Fedora:**

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

**SUSE:**

```
sudo zypper install tftp xinetd
```

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

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

and restart the xinetd service:

**Ubuntu:**

```
sudo service xinetd restart
```

**Fedora/RHEL7/CentOS7/SUSE:**

```
sudo systemctl restart xinetd
```

---

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

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

---

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

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

## UEFI PXE - Grub setup

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

1. Install Grub2 and shim packages:

Ubuntu (16.04LTS and later):

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

RHEL7/CentOS7:

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

Fedora:

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

SUSE:

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

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

Ubuntu (16.04LTS and later):

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

Fedora:

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

RHEL7/CentOS7:

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

SUSE:

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

3. Create master grub.cfg:

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

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

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

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

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

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

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

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

Create directory GRUB\_DIR:

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

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

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

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

Change the permission of grub.cfg:

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

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

---

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

---

## Legacy BIOS - Syslinux setup

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

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

Ubuntu (16.04LTS and later):



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

RHEL7/CentOS7:

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

Fedora:

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

SUSE:

```
sudo zypper install syslinux
```

2. Copy the PXE image to /tftpboot. The PXE image might be found at<sup>1</sup>:

Ubuntu (16.04LTS and later):

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

RHEL7/CentOS7/SUSE:

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

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

Ubuntu (16.04LTS and later):

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

Fedora:

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

RHEL7/CentOS7/SUSE:

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

4. If the version of syslinux is **greater than 4** we also need to make sure that we copy the library modules into the /tftpboot directory<sup>2</sup>. 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.

<sup>1</sup> 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

<sup>2</sup> [http://www.syslinux.org/wiki/index.php/Library\\_modules](http://www.syslinux.org/wiki/index.php/Library_modules)

## 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 ^([^\s]) /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:

```
[pxe]

# IroniC compute node's tftp root path. (string value)
tftp_root=/tftpboot

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

[deploy]

# IroniC compute node's http root path. (string value)
http_root=/httpboot

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

4. Install the iPXE package with the boot images:

Ubuntu:

```
apt-get install ipxe
```

RHEL7/CentOS7:

```
yum install ipxe-bootimgs
```

Fedora:

```
dnf install ipxe-bootimgs
```

---

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

---

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

Ubuntu:

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

Fedora/RHEL7/CentOS7:

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

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

```
[pxe]

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

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

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

---

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

---



---

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

---

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

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

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

**[DEFAULT]**

```
default_boot_interface=ipxe
enabled_boot_interfaces=ipxe,pxe
```

---

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

---

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

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

**[pxe]**

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

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

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

9. Restart the `ironic-conductor` process:

Fedora/RHEL7/CentOS7/SUSE:

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

Ubuntu:

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

## PXE multi-architecture setup

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

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

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

```
[pxe]

# Bootfile DHCP parameter. (string value)
pxe_bootfile_name=pxelinux.0

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

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

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

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

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

---

**Note:** The grub implementation may vary on different architecture, you may need to tweak the pxe config template for a specific arch. For example, `grubaa64.efi` shipped with CentoOS7 does not support `linuxefi` and `initrdefi` commands, 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 doesn't 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.
| 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 |
+-----+-----+
```

(continues on next page)

(continued from previous page)

```
$ 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
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=$USER \
  --driver-info ipmi_password=$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 youre 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 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

(continues on next page)

(continued from previous page)

When using the Compute Service with the Bare Metal service, it is safe to ignore the deployment 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:

---

(continues on next page)

(continued from previous page)

inspect	True		
management	True		
network	True		
power	True		
raid	None	not supported	
storage	True		
+-----+-----+-----+-----+			
+-----+-----+-----+-----+			
+-----+-----+-----+-----+			

## Making node available for deployment

In order for nodes to be available for deploying workloads on them, nodes must be in the `available` provision state. To do this, nodes created with API version 1.11 and above must be moved from the `enroll` state to the `manageable` state and then to the `available` state. This section can be safely skipped, if API version 1.10 or earlier is used (which is the case by default).

After creating a node and before moving it from its initial provision state of `enroll`, basic power and port information needs to be configured on the node. The Bare Metal service needs this information because it verifies that it is capable of controlling the node when transitioning the node from `enroll` to `manageable` state.

To move a node from `enroll` to `manageable` provision state:

```
$ 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                                     |
+-----+-----+
| ...                | ...                                     |
+-----+-----+
| 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                                     |
+-----+-----+
| 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                                |
+-----+-----+
```

(continues on next page)

(continued from previous page)

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	
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
+-----+
```

(continues on next page)

(continued from previous page)

Property	Value
uuid	dfc6189f-ad83-4261-9bda-b27258eb1987
driver_info	{}
extra	{}
driver	ipmi
chassis_uuid	
properties	{}
name	None

Then the defaults for the interfaces that will be used by the node in this example are calculated as follows:

**deploy** An explicit value of `direct` is provided for `default_deploy_interface`, so it is used.

**power** No default is configured. The `ipmi` hardware type supports only `ipmitool` power. The intersection between supported power interfaces and values provided in the `enabled_power_interfaces` option has only one item: `ipmitool`. It is used.

**console** No default is configured. The `ipmi` hardware type supports the following console interfaces: `ipmitool-socat`, `ipmitool-shellinbox` and `no-console` (in this order). Of these three, only two are enabled: `no-console` and `ipmitool-shellinbox` (order does not matter). The intersection contains `ipmitool-shellinbox` and `no-console`. The first item is used, and it is `ipmitool-shellinbox`.

**management** Following the same calculation as *power*, the `ipmitool` management interface is used.

## Hardware Inspection

The Bare Metal service supports hardware inspection that simplifies enrolling nodes - please see [Hardware Inspection](#) for details.

## Tenant Networks and Port Groups

See [Multi-tenancy in the Bare Metal service](#) and [Port groups support](#).

## 2.1.9 Using Bare Metal service as a standalone service

### Service settings

It is possible to use the Bare Metal service without other OpenStack services. You should make the following changes to `/etc/ironic/ironic.conf`:

1. Choose an authentication strategy which supports standalone, one option is `noauth`:

```
[DEFAULT]
...
auth_strategy=noauth
```

Another option is `http_basic` where the credentials are stored in an [Apache htpasswd](#) format file:

```
[DEFAULT]
...
auth_strategy=http_basic
http_basic_auth_user_file=/etc/ironic/htpasswd
```

Only the `bcrypt` format is supported, and the Apache `htpasswd` utility can be used to populate the file with entries, for example:

```
htpasswd -nbB myName myPassword >> /etc/ironic/htpasswd
```

2. If you want to disable the Networking service, you should have your network pre-configured to serve DHCP and TFTP for machines that you're deploying. To disable it, change the following lines:

```
[dhcp]
...
dhcp_provider=none
```

---

**Note:** If you disabled the Networking service and the driver that you use is supported by at most one conductor, PXE boot will still work for your nodes without any manual config editing. This is because you know all the DHCP options that will be used for deployment and can set up your DHCP server appropriately.

If you have multiple conductors per driver, it would be better to use Networking since it will do all the dynamically changing configurations for you.

---

3. If you want to disable using a messaging broker between conductor and API processes, switch to JSON RPC instead:

```
[DEFAULT]
rpc_transport = json-rpc
```

JSON RPC also has its own authentication strategy. If it is not specified then the strategy defaults to [DEFAULT] `auth_strategy`. The following will set JSON RPC to `noauth`:

```
[json_rpc]
auth_strategy = noauth
```

For `http_basic` the conductor server needs a credentials file to validate requests:

```
[json_rpc]
auth_strategy = http_basic
http_basic_auth_user_file = /etc/ironic/htpasswd-json-rpc
```

The API server also needs client-side credentials to be specified:

```
[json_rpc]
auth_type = http_basic
username = myName
password = myPassword
```

## Preparing images

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

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.

## 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
```

## 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 don't need to add properties like `memory_mb`, `cpus` etc. as Bare Metal service will require UUID of a node you're going to deploy.
3. Then create a port to inform Bare Metal service of the network interface cards which are part of

the node by creating a port with each NICs MAC address. In this case, they're used for naming of PXE configs for a node:

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

## Populating instance\_info

1. You also 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=$MD5HASH \
  --instance-info kernel=$KERNEL \
  --instance-info ramdisk=$RAMDISK \
  --instance-info root_gb=10
```

With a SHA256 hash:

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

(continues on next page)

(continued from previous page)

```
--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
```

2. 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
```

3. *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.

---

4. 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" }'
```

5. 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.

6. 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`.

7. 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>
```

## 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	
network	True	
power	True	
raid	True	
storage	True	

2. Now you can start the deployment, run:

```
baremetal node deploy $NODE_UUID
```

## Ramdisk booting

Advanced operators, specifically ones working with ephemeral workloads, may find it more useful to explicitly treat a node as one that would always boot from a Ramdisk.

This functionality is largely intended for network booting, however some other boot interface, such as the `redfish-virtual-media` support enabling the same basic functionality through the existing interfaces.

To use, a few different settings must be modified.

1. Change the `deploy_interface` on the node to `ramdisk`:

```
baremetal node set $NODE_UUID \  
    --deploy-interface ramdisk
```

2. Set a kernel and ramdisk to be utilized:

```
baremetal node set $NODE_UUID \  
    --instance-info kernel=$KERNEL_URL \  
    --instance-info ramdisk=$RAMDISK_URL
```

3. Deploy the node:

```
baremetal node deploy $NODE_UUID
```

**Warning:** Configuration drives, also known as a configdrive, is not supported with the ramdisk deploy interface. Please ensure your ramdisk CPIO archive contains all necessary configuration and credentials. This is as no disk image is written to the disk of the node being provisioned with a ramdisk.

The node ramdisk components will then be assembled by the conductor, appropriate configuration put in place, and the node will then be powered on. From there, normal node booting will occur. Upon undeployment of the node, normal cleaning procedures will occur as configured with-in the conductor.

## Ramdisk booting with ISO media

Currently supported for the use of ramdisks with the `redfish-virtual-media` and `ipxe` boot interfaces, an operator may request an explicit ISO file to be booted.

1. Store the URL to the ISO image to `instance_info/boot_iso`, instead of a kernel or ramdisk setting:

```
baremetal node set $NODE_UUID \  
    --instance-info boot_iso=$BOOT_ISO_URL
```

2. Deploy the node:

```
baremetal node deploy $NODE_UUID
```

**Warning:** This feature, when utilized with the `ipxe boot_interface`, will only allow a kernel and ramdisk to be booted from the supplied ISO file. Any additional contents, such as additional ramdisk contents or installer package files will be unavailable after the boot of the Operating System. Operators wishing to leverage this functionality for actions such as OS installation should explore use of the standard ramdisk `deploy_interface` along with the `instance_info/kernel_append_params` setting to pass arbitrary settings such as a mirror URL for the initial ramdisk to load data from. This is a limitation of iPXE and the overall boot process of the operating system where memory allocated by iPXE is released.

## Other references

- [Enabling local boot without Compute](#)

### 2.1.10 Enabling the configuration drive (configdrive)

The Bare Metal service supports exposing a configuration drive image to the instances.

The configuration drive is used to store instance-specific metadata and is present to the instance as a disk partition labeled `config-2`. The configuration drive has a maximum size of 64MB. One use case for using the configuration drive is to expose a networking configuration when you do not use DHCP to assign IP addresses to instances.

The configuration drive is usually used in conjunction with the Compute service, but the Bare Metal service also offers a standalone way of using it. The following sections will describe both methods.

#### When used with Compute service

To enable the configuration drive for a specific request, pass `--config-drive true` parameter to the **nova boot** command, for example:

```
nova boot --config-drive true --flavor baremetal --image test-image_
↪instance-1
```

It's 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 9660<sup>1</sup> 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
```

<sup>1</sup> A configuration drive could also be a data block with a VFAT filesystem on it instead of ISO 9660. But it's unlikely that it would be needed since ISO 9660 is widely supported across operating systems.

Starting with the Stein release and *ironicclient* 2.7.0, you can request building a configdrive on the server side by providing a JSON with keys `meta_data`, `user_data` and `network_data` (all optional), e.g.:

```
baremetal node deploy $node_identifier \  
    --config-drive '{"meta_data": {"hostname": "server1.cluster"}}'
```

### **Configuration drive storage in an object store**

Under normal circumstances, the configuration drive can be stored in the Bare Metal service when the size is less than 64KB. Optionally, if the size is larger than 64KB there is support to store it in a swift endpoint. Both swift and radosgw use swift-style APIs.

The following option in `/etc/ironic/ironic.conf` enables swift as an object store backend to store config drive. This uses the Identity service to establish a session between the Bare Metal service and the Object Storage service.

```
[deploy]  
...  
  
configdrive_use_object_store = True
```

Use the following options in `/etc/ironic/ironic.conf` to enable radosgw. Credentials in the swift section are needed because radosgw will not use the Identity service and relies on radosgw's 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:
            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        |
+-----+-----+-----+-----+

$ 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
```

(continues on next page)

(continued from previous page)

```
$ nova hypervisor-list
```

ID	State	Status	Hypervisor hostname
584cfdc8-9afd-4fbb-82ef-9ff25e1ad3f3	up	enabled	86a2b1bb-8b29-4964-a817-f90031debddb

```
$ 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
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-shellinbox`. See [Console](#) for details.

---

For out-of-tree drivers you may need to reach out to their maintainers or figure out the appropriate interfaces by researching the source code.

## Configuration

You will need to enable hardware types and interfaces that correspond to your currently enabled classic drivers. For example, if you have the following configuration in your `ironic.conf`:

```
[DEFAULT]
enabled_drivers = pxe_ipmitool,agent_ipmitool
```

You will have to add this configuration as well:

```
[DEFAULT]
enabled_hardware_types = ipmi
enabled_boot_interfaces = pxe
enabled_deploy_interfaces = iscsi,direct
enabled_management_interfaces = ipmitool
enabled_power_interfaces = ipmitool
```

**Note:** For every interface type there is an option `default_<INTERFACE>_interface`, where `<INTERFACE>` is the interface type name. For example, one can make all nodes use the direct deploy method by default by setting:

```
[DEFAULT]
default_deploy_interface = direct
```

## Migrating nodes

After the required items are enabled in the configuration, each nodes `driver` field has to be updated to a new value. You may need to also set new values for some or all interfaces:

```
export OS_BAREMETAL_API_VERSION=1.31

for uuid in $(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.

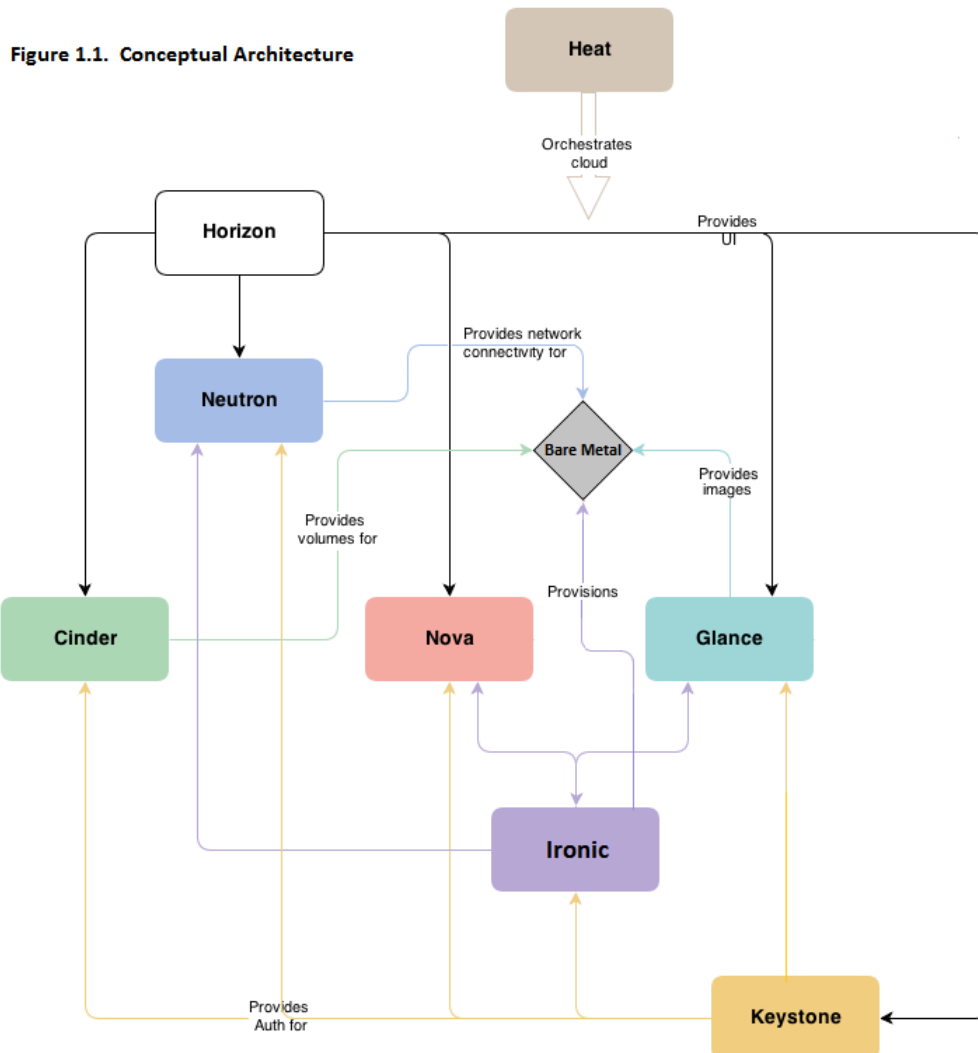
### **4.1.1 Why Provision Bare Metal**

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 (LIinux 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. Nova's 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.

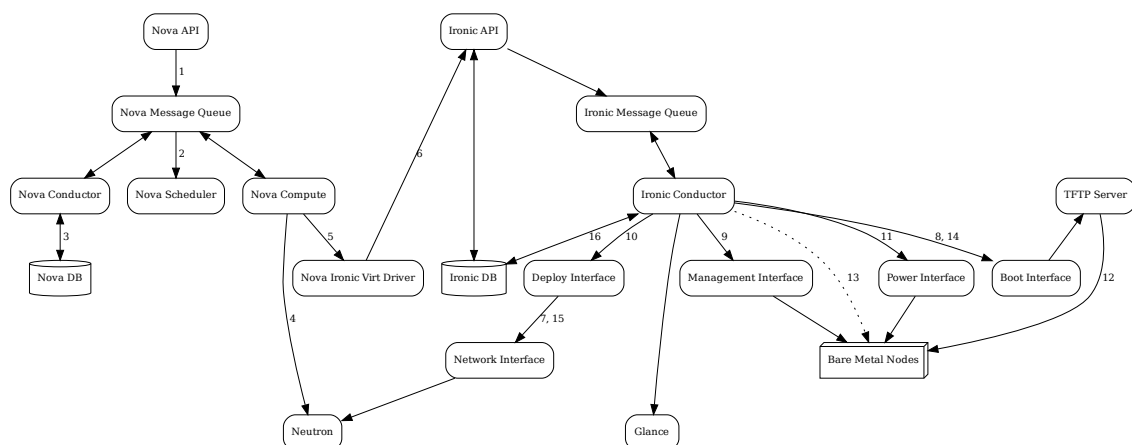
---

**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`.

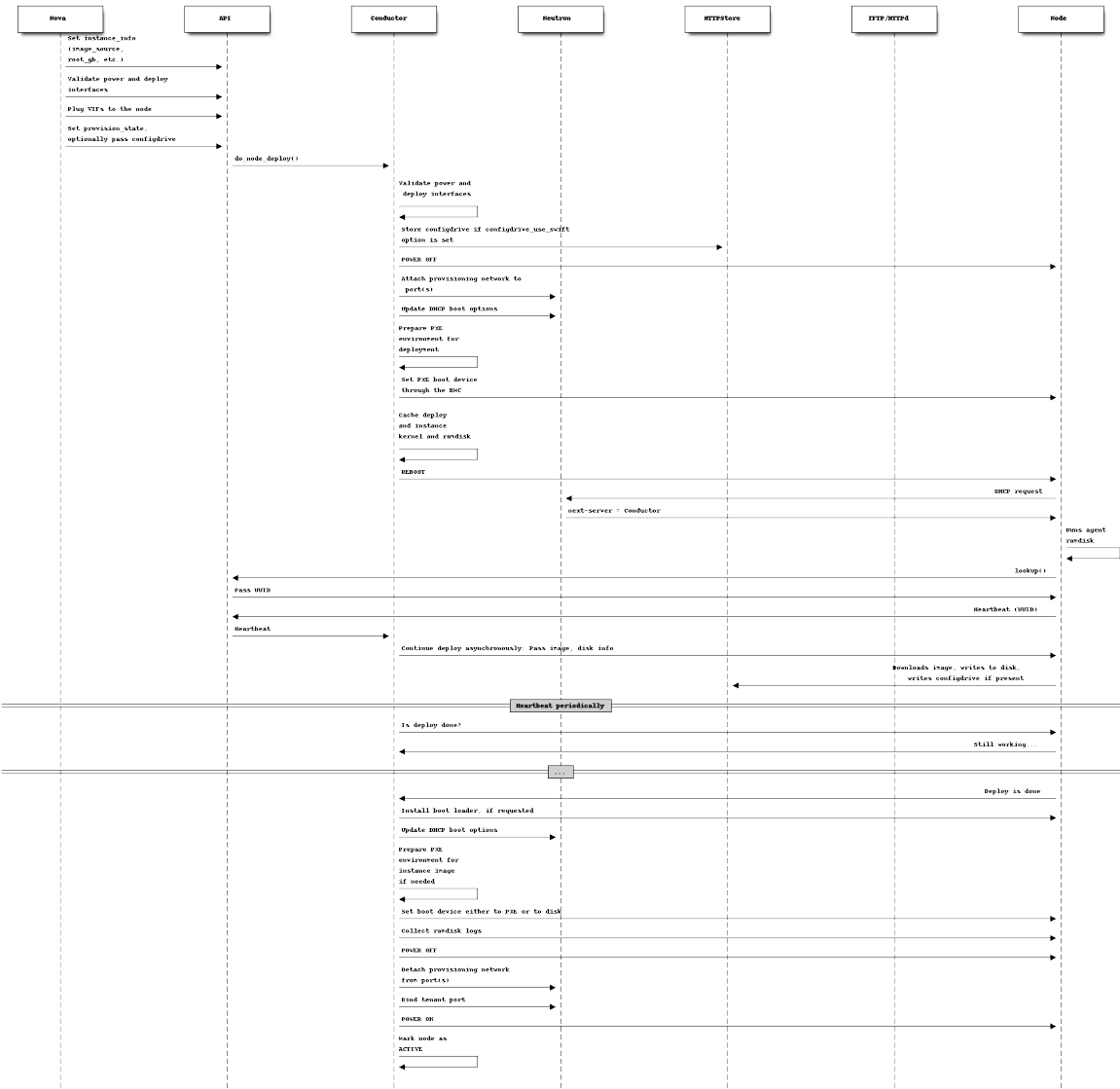
Below is the diagram that describes the above process.



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

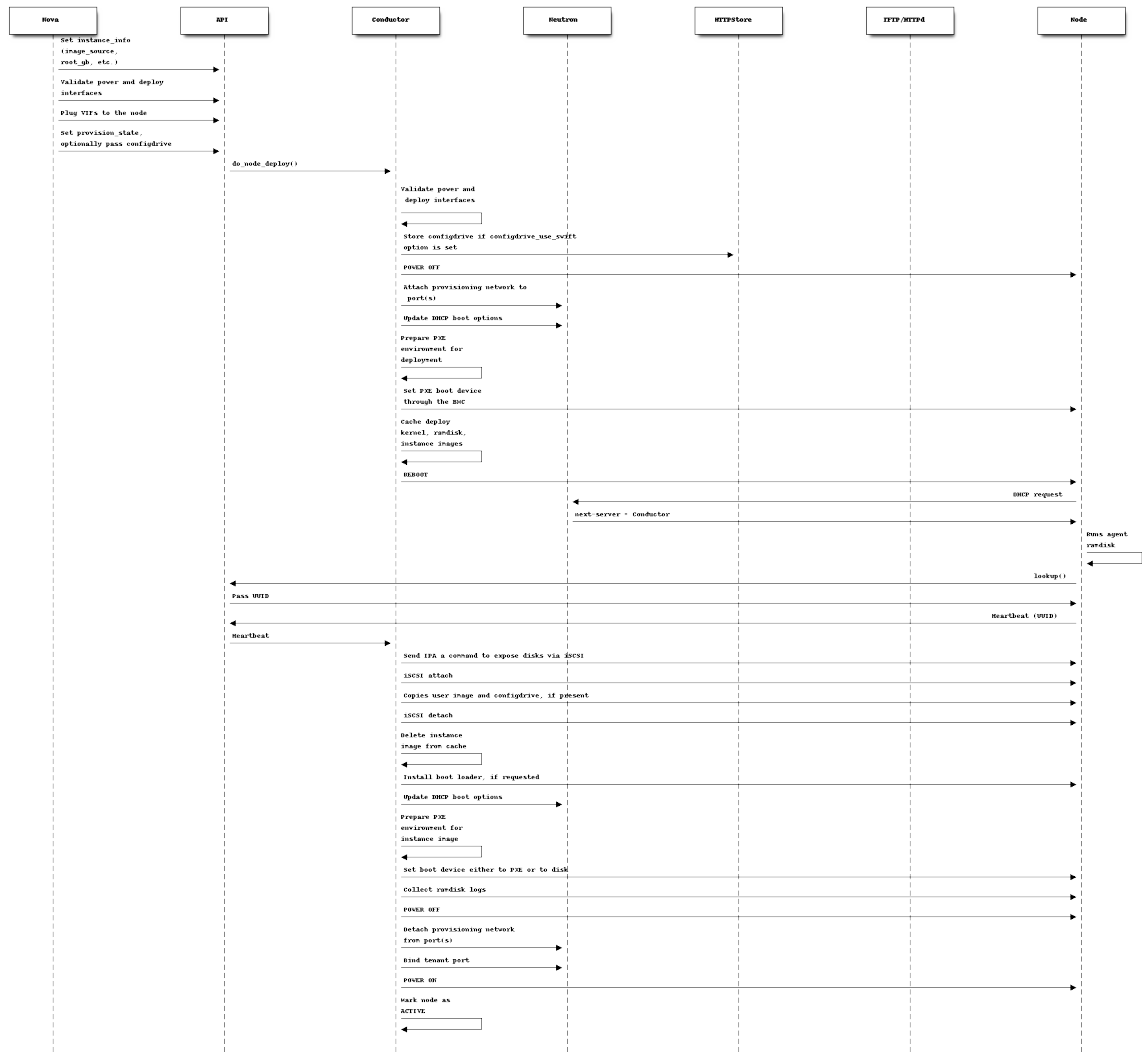
This process is how *Direct deploy* works.



(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
```

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 [Ironiic 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**



(continues on next page)

(continued from previous page)

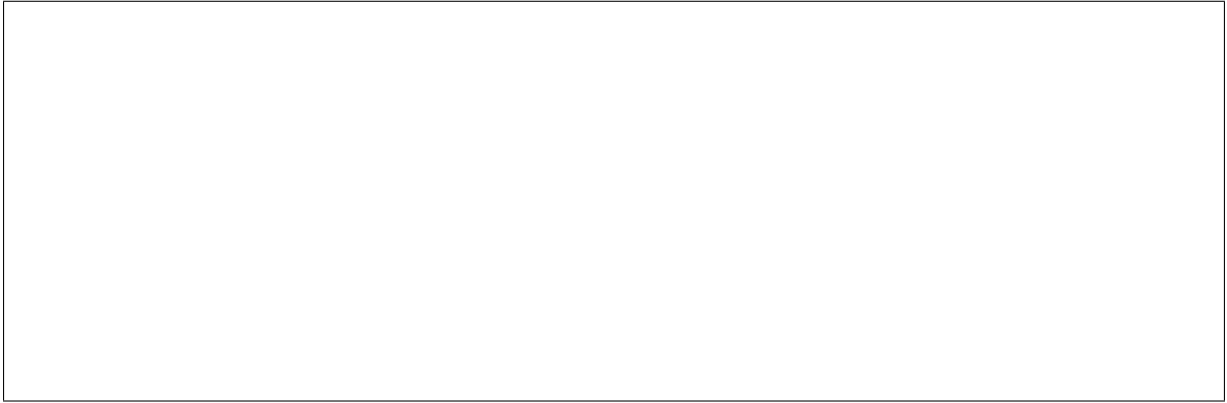
--

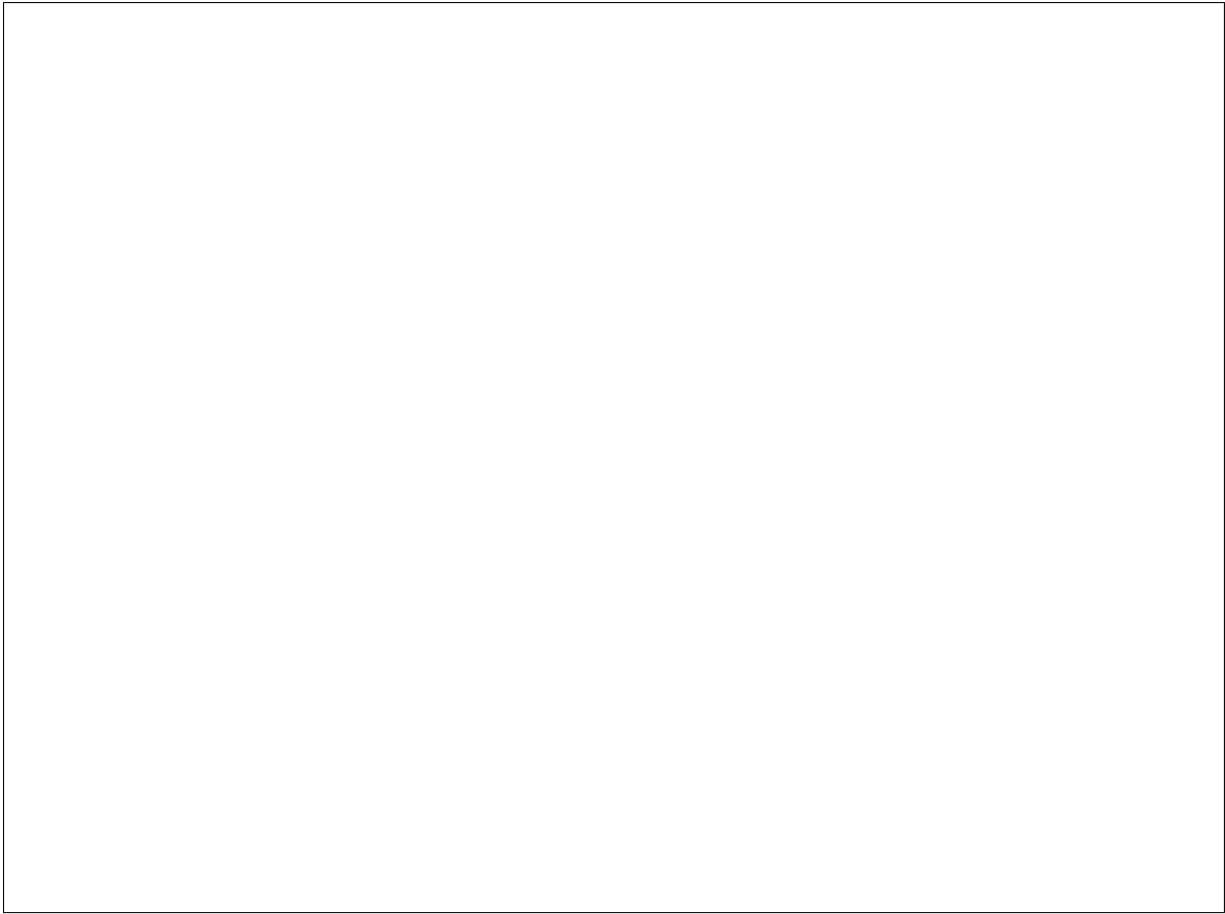


## **Enabling**

match WSMAN and Redfish interfaces.

**Note:** Redfish is supported for only the bios, inspect, management, and power interfaces at the present time.







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, 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





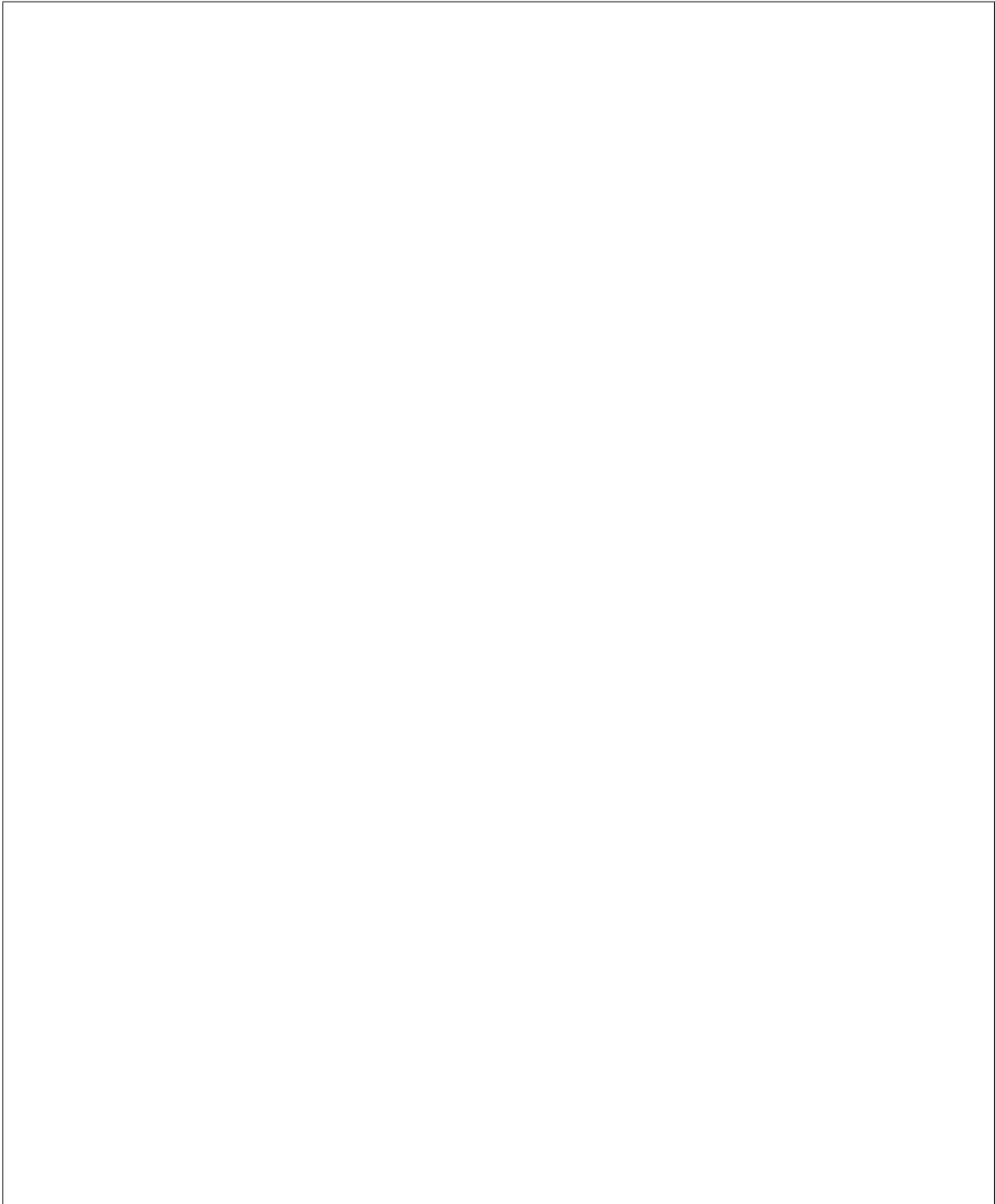






interfaces, where some use WSMAN and others use Redfish, both the WSMAN and Redfish properties must be supplied.

## **Enrolling**







(continues on next page)

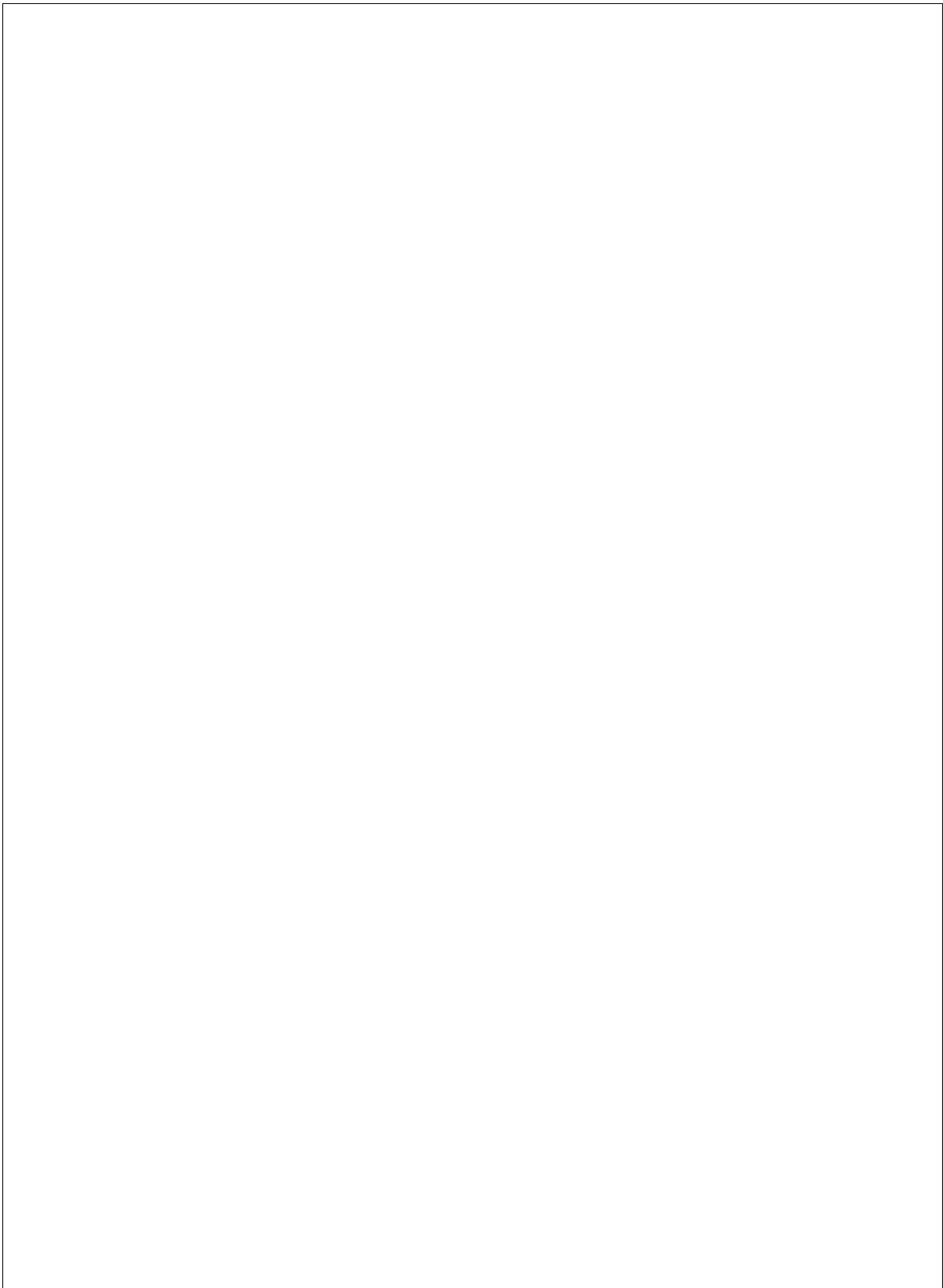
(continued from previous page)



(continues on next page)

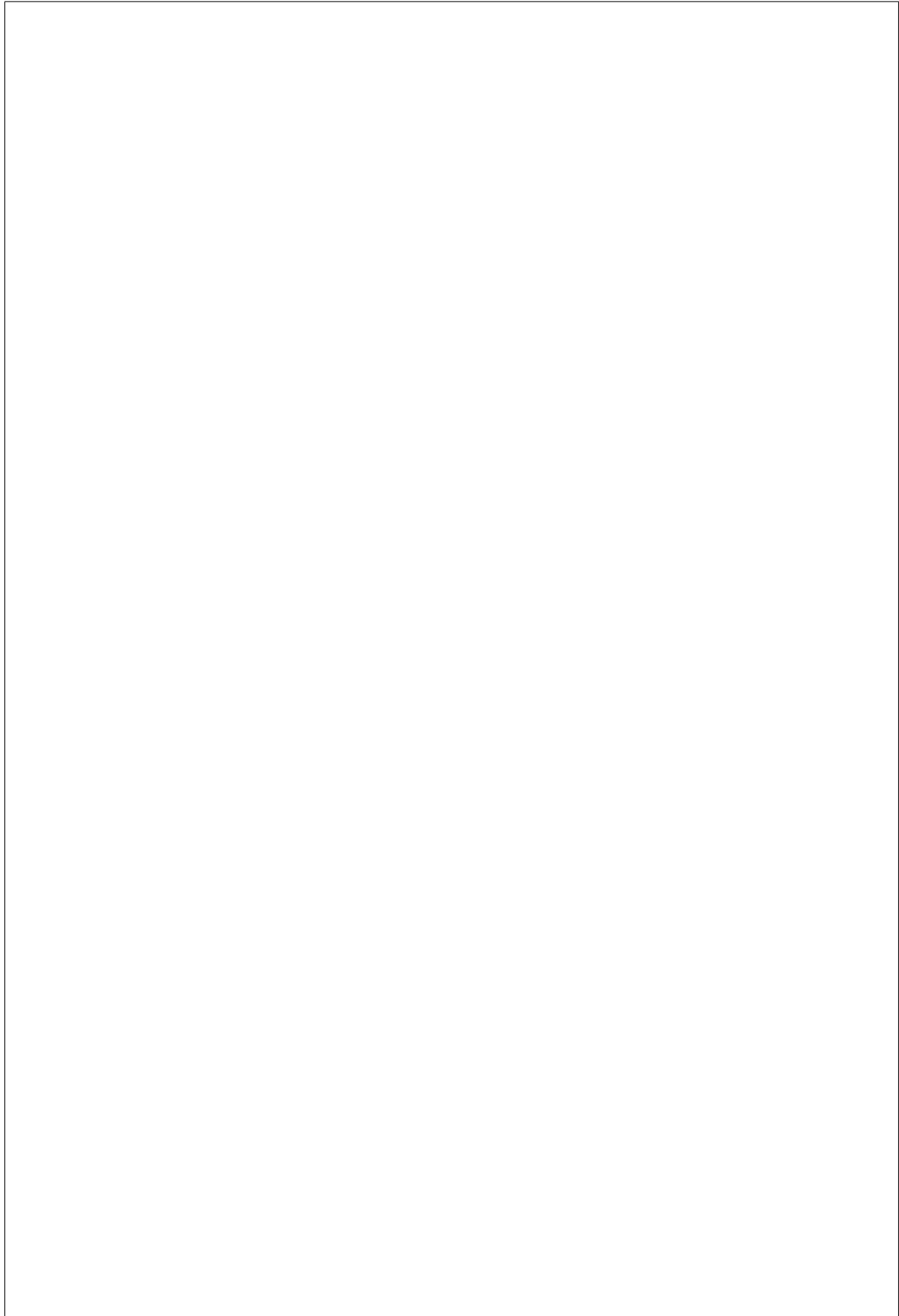
(continued from previous page)





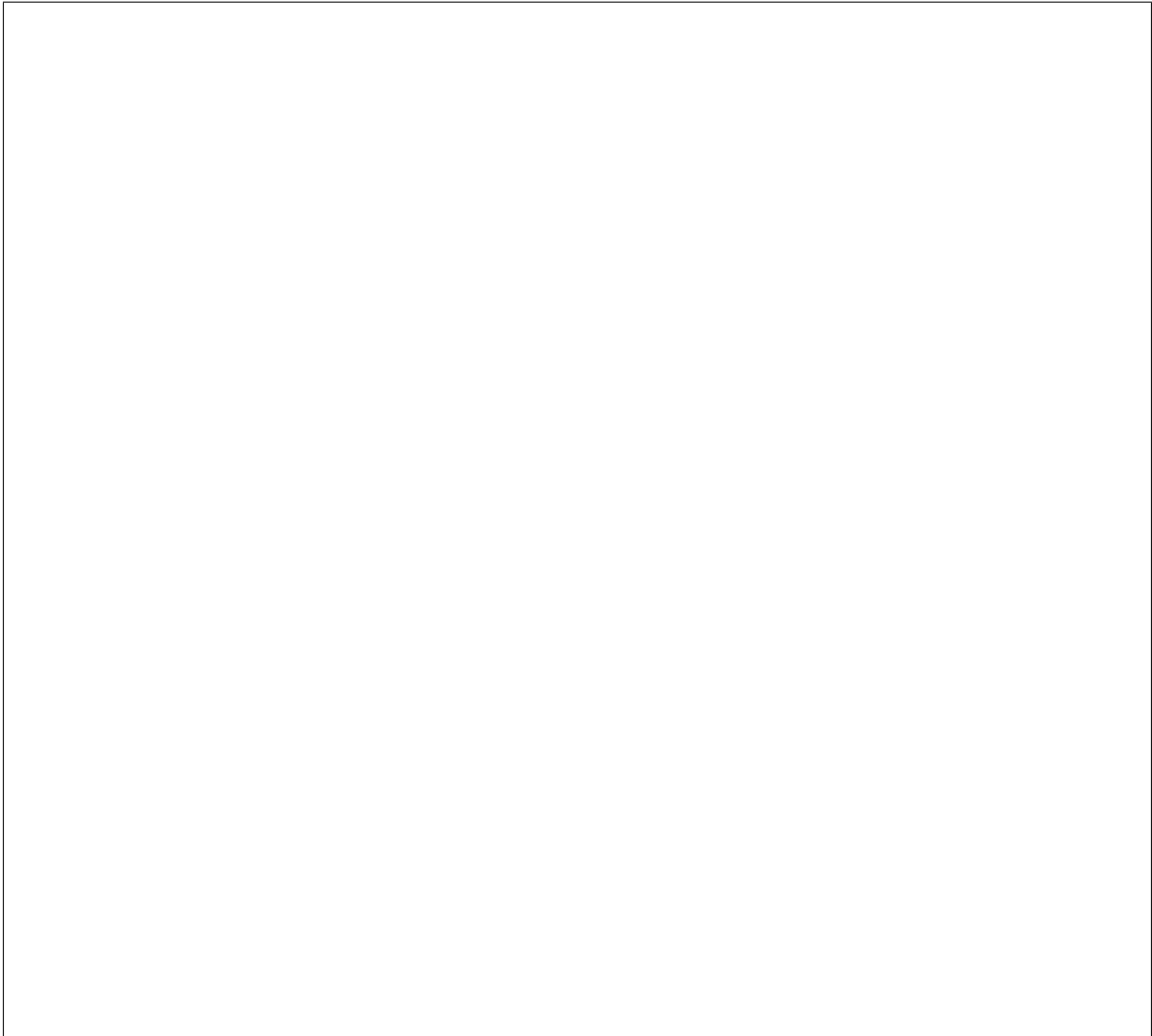
(continues on next page)

(continued from previous page)



(continues on next page)

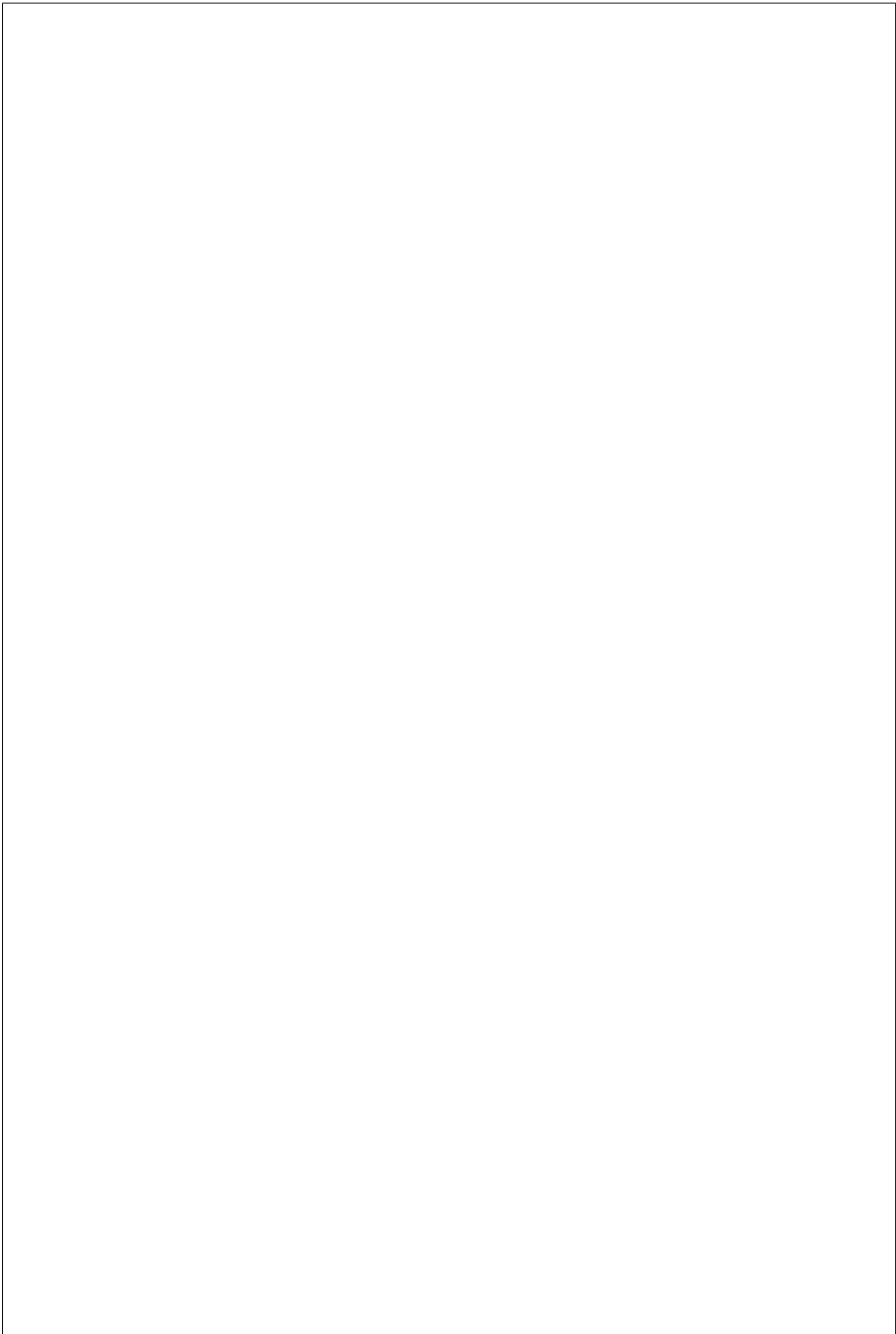
(continued from previous page)



**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**

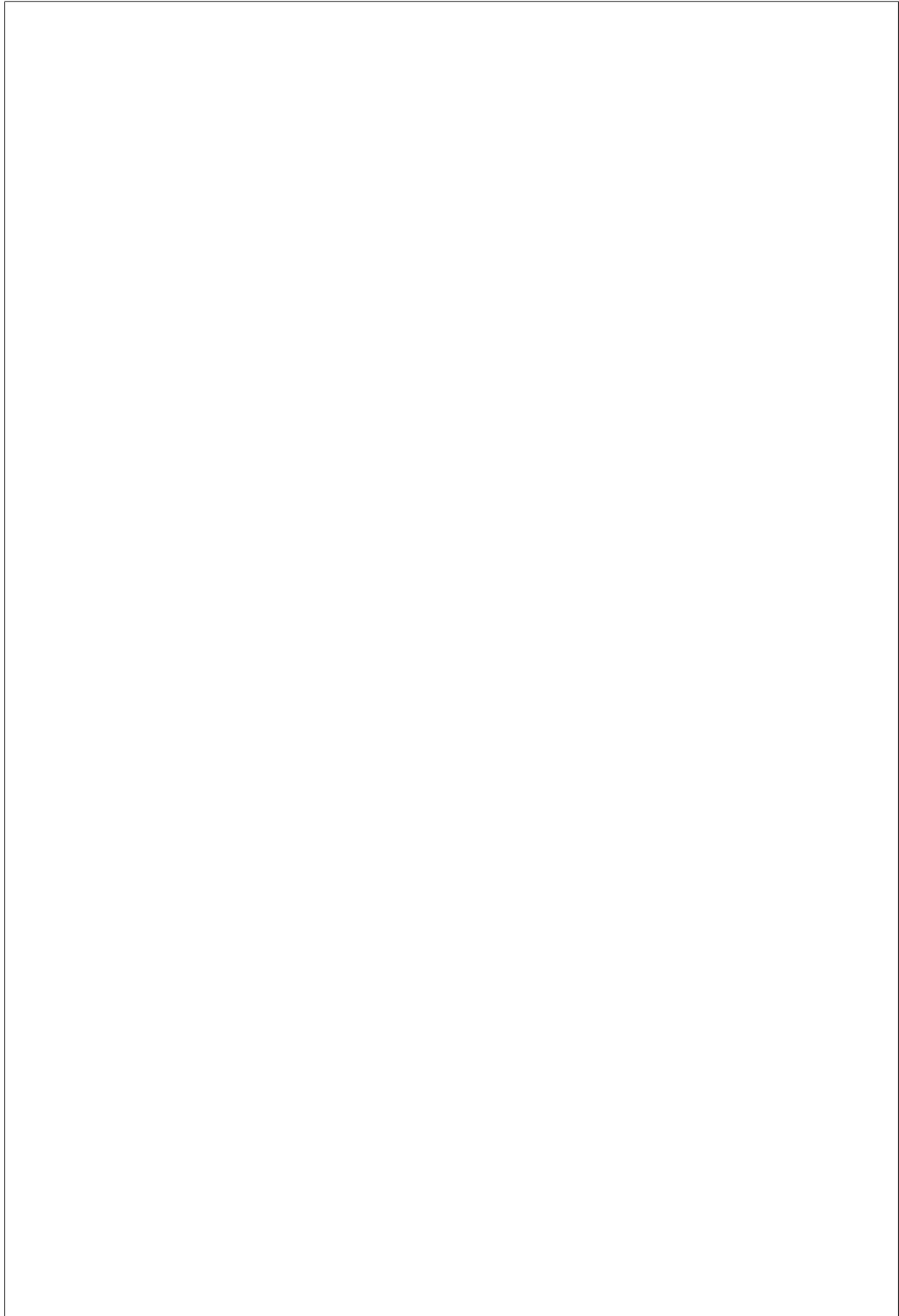
**Example**



(continues on next page)



(continued from previous page)



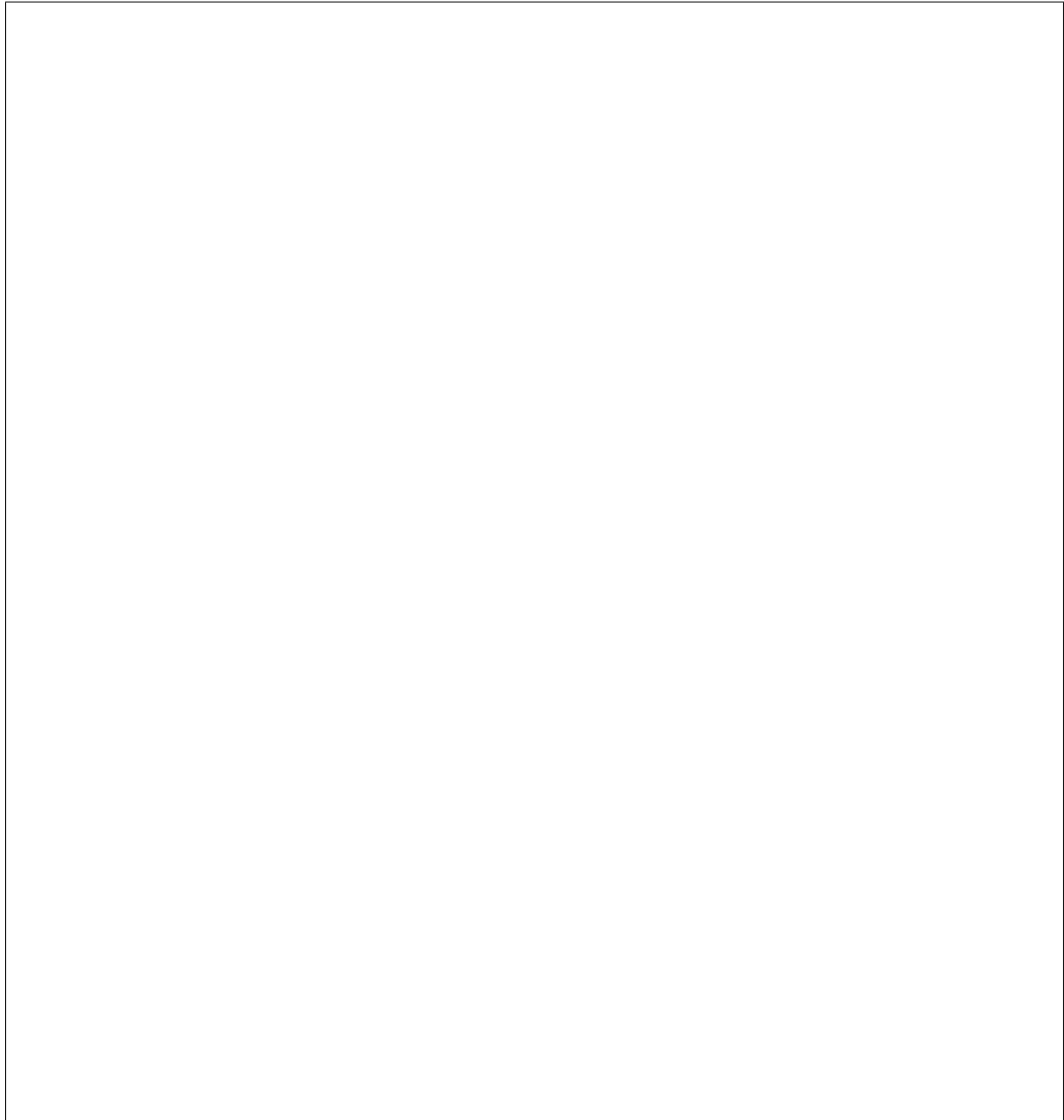
(continues on next page)

(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**



## **RAID Interface**

## **Mandatory properties**

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

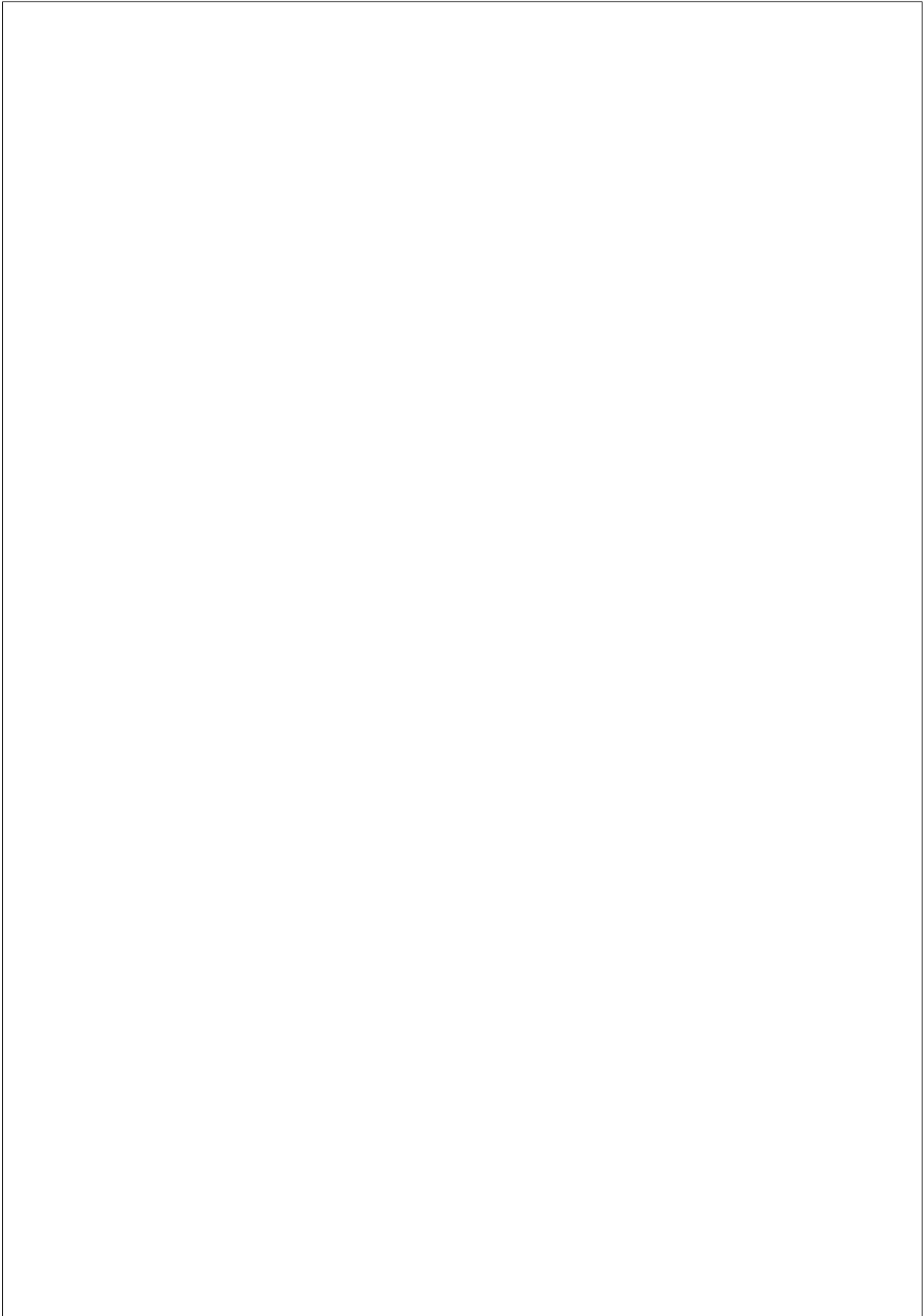
### **Optional properties**

## **Backing physical disk hints**

## **Backing physical disks**

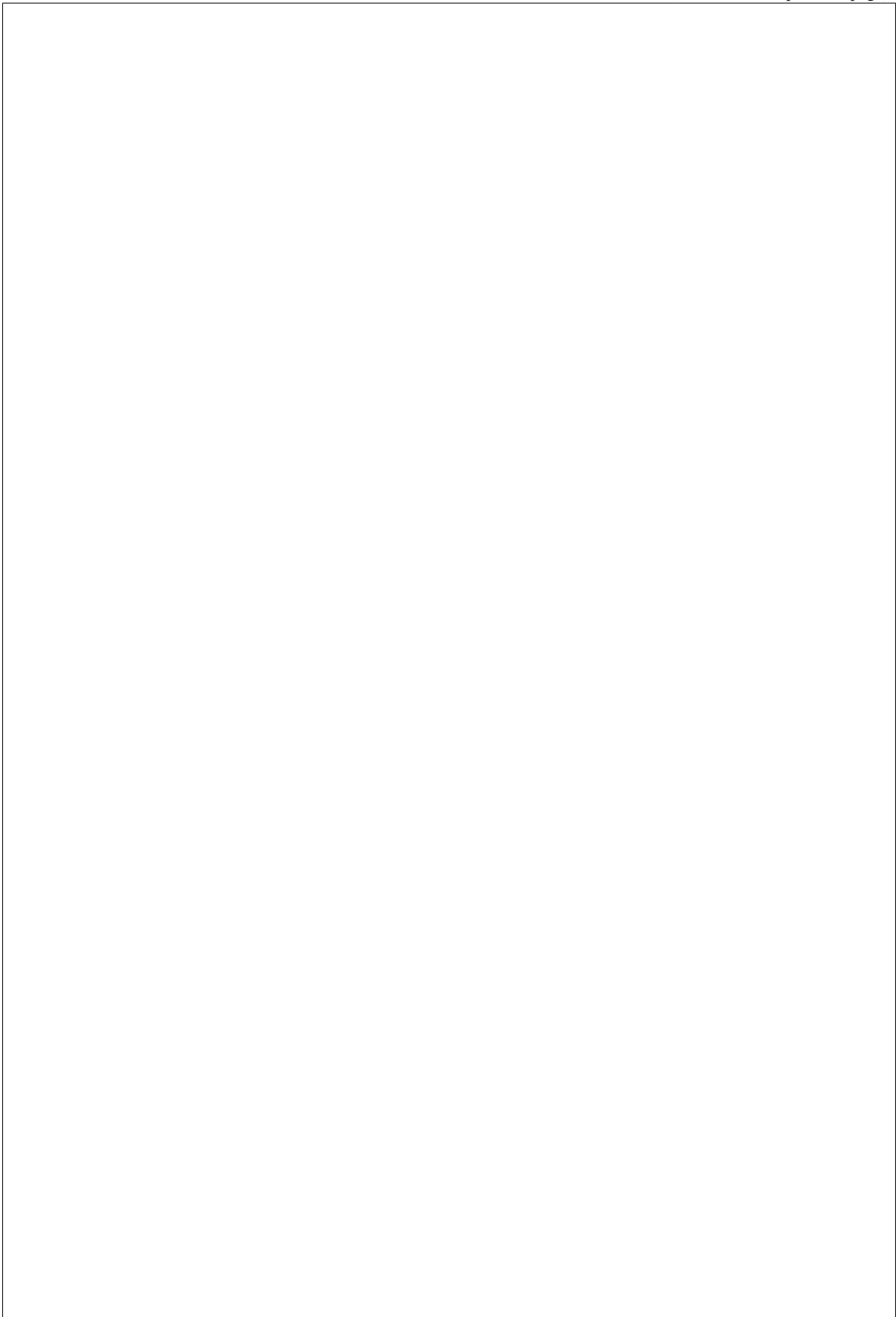
**Note:** `physical_disks` is a mandatory parameter if the property `size_gb` is set to `MAX`.

## **Examples**



(continues on next page)

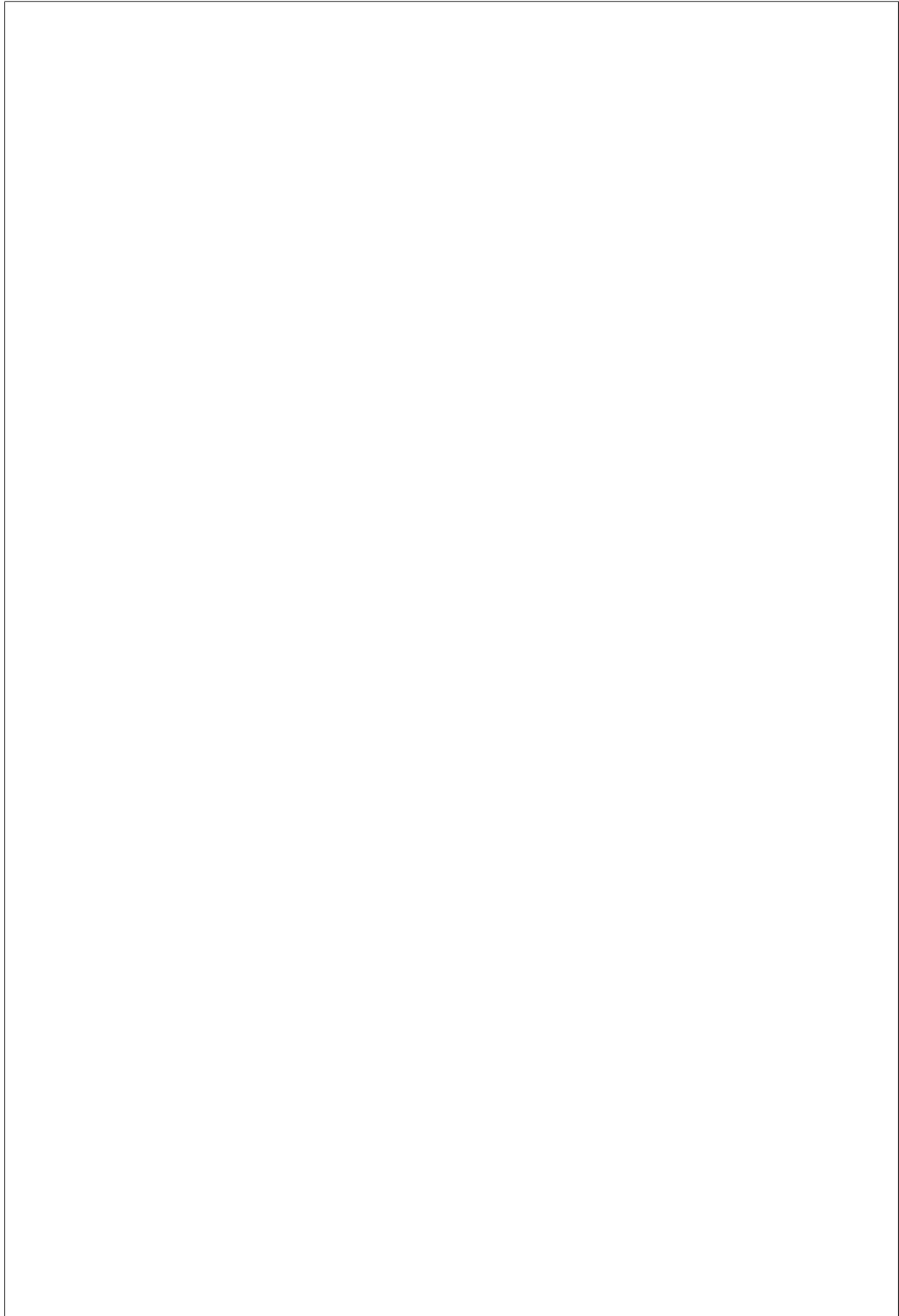
(continued from previous page)



(continues on next page)



(continued from previous page)



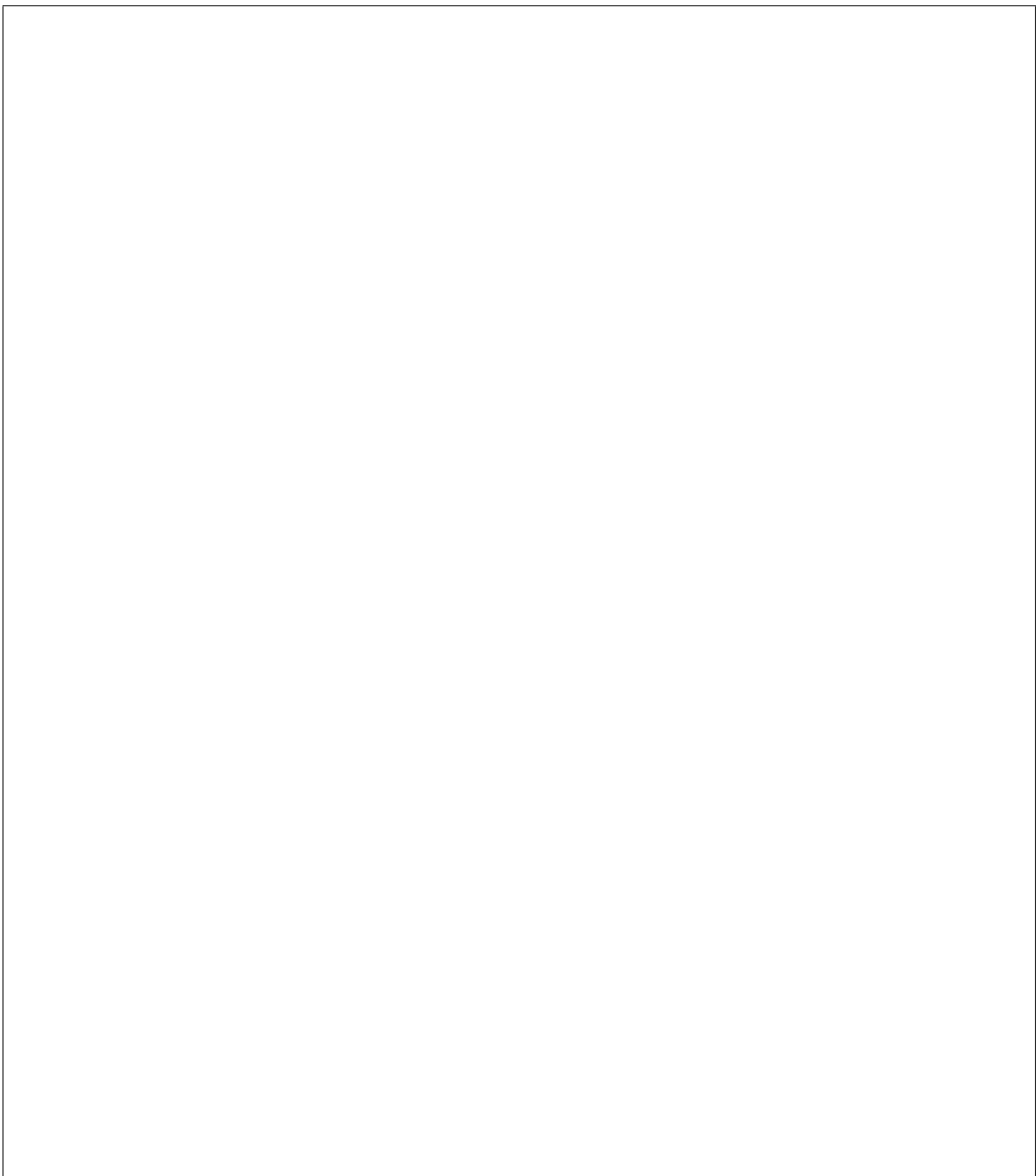
(continues on next page)

(continued from previous page)

---

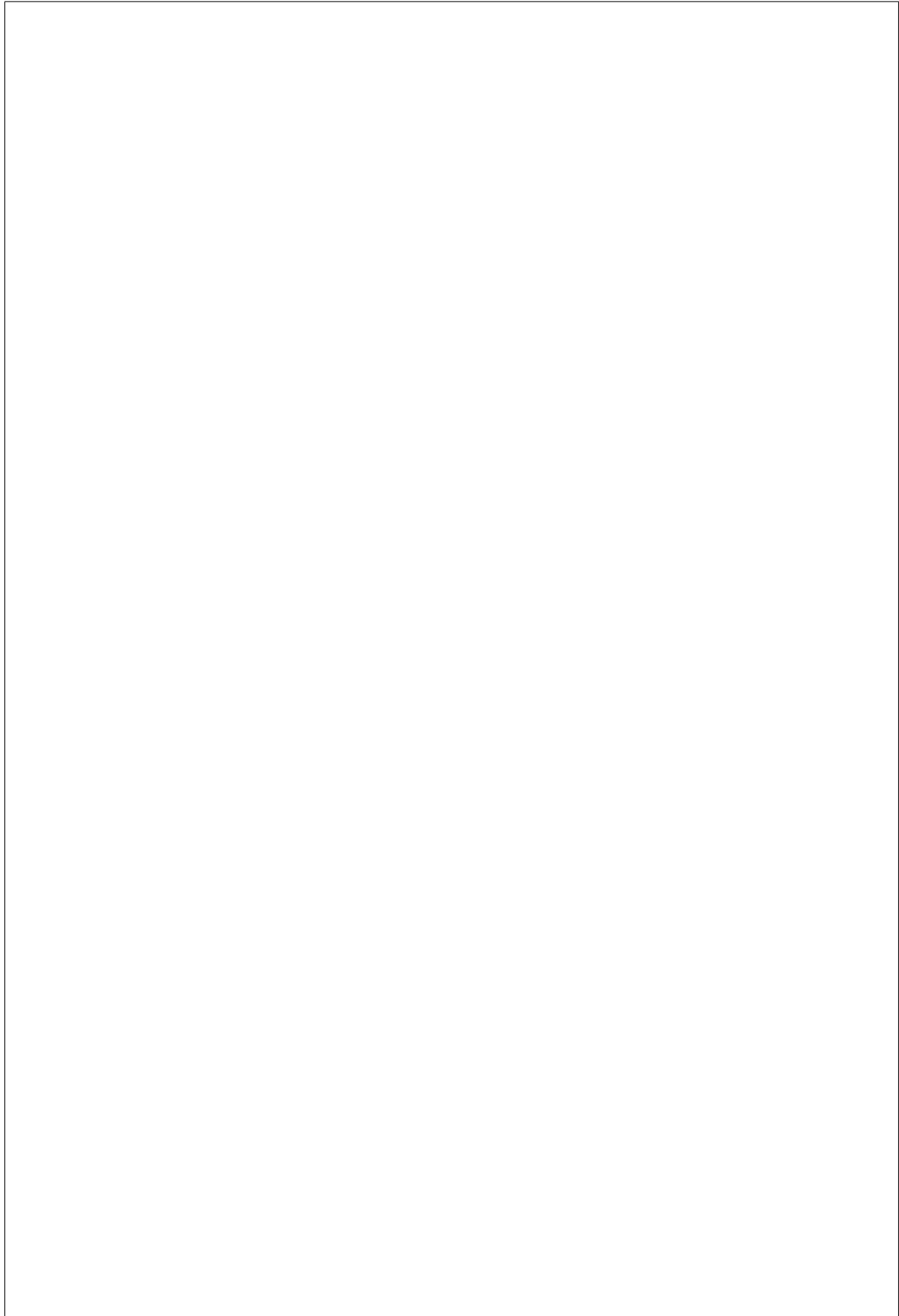
## **Manual RAID Invocation**





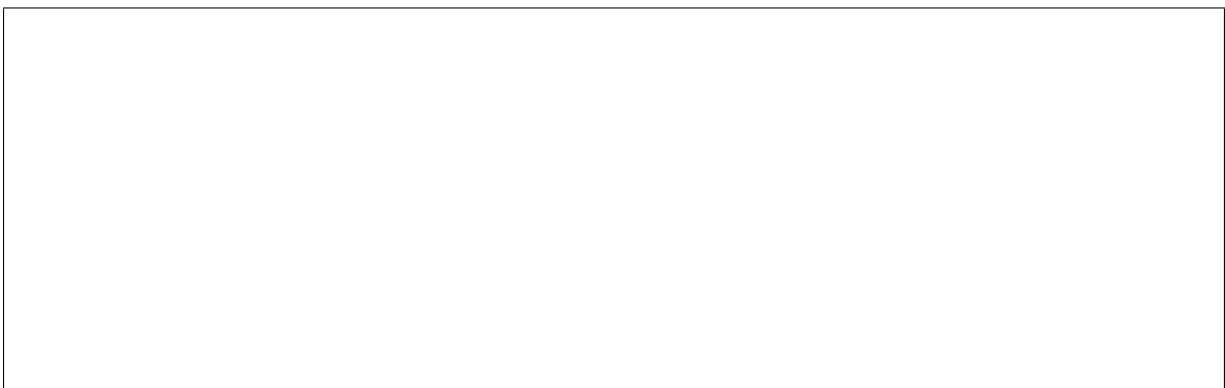
(continues on next page)

(continued from previous page)



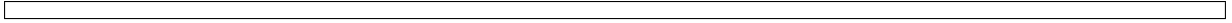
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



## Vendor Interface

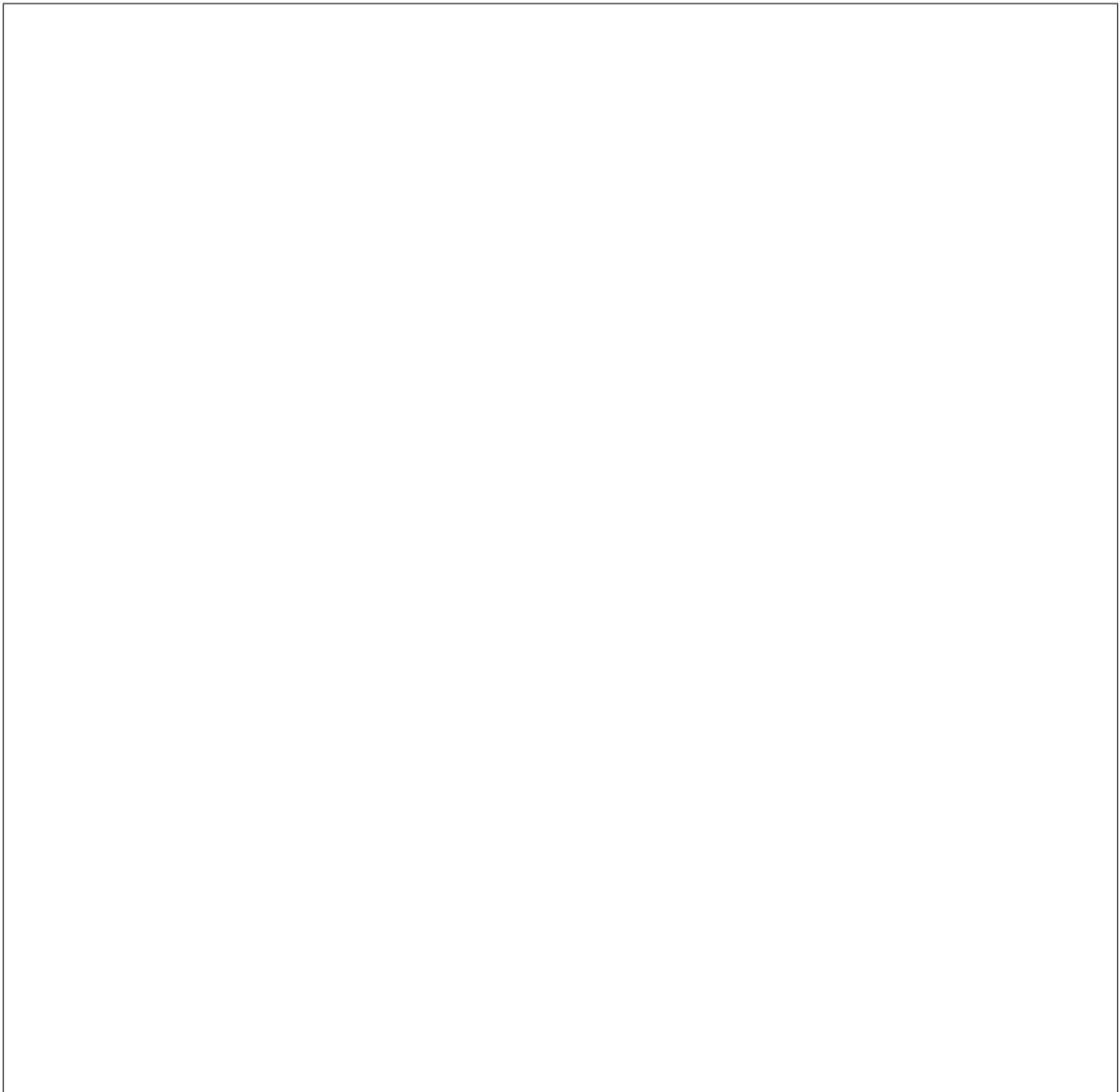
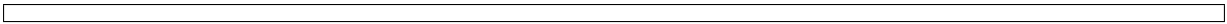
Method Name	HTTP Method	Description
abandon_bios	DELETE	Abandon a BIOS configuration job.
commit_bios	POST	Commit a BIOS configuration job submitted through set_bios_config. Required argument: <code>reboot</code> - indicates whether a reboot job should be automatically created with the config job. Returns a dictionary containing the <code>job_id</code> key with the ID of the newly created config job, and the <code>reboot_required</code> key indicating whether the node needs to be rebooted to execute the config job.
get_bios	GET	Returns a dictionary containing the nodes BIOS settings.
list_unfinished_bios	GET	Returns a dictionary containing the key <code>unfinished_jobs</code> ; its value is a list of dictionaries. Each dictionary represents an unfinished config job object.
set_bios	POST	Change the BIOS configuration on a node. Required argument: a dictionary of <code>{AttributeName: NewValue}</code> . Returns a dictionary containing the <code>is_commit_required</code> key indicating whether <code>commit_bios_config</code> needs to be called to apply the changes and the <code>is_reboot_required</code> value indicating whether the server must also be rebooted. Possible values are <code>true</code> and <code>false</code> .

## Examples

### Get BIOS Config

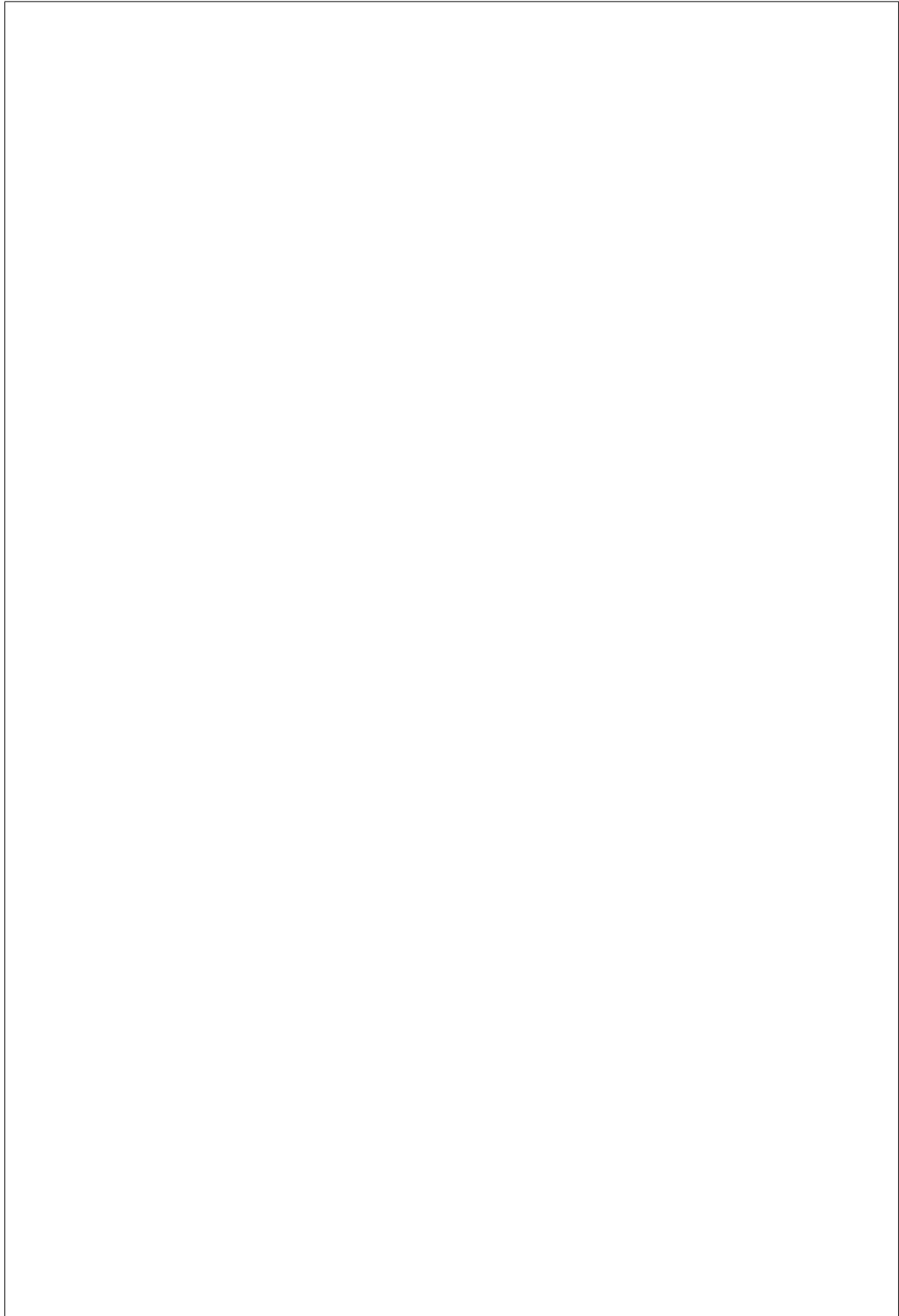
(continues on next page)

(continued from previous page)



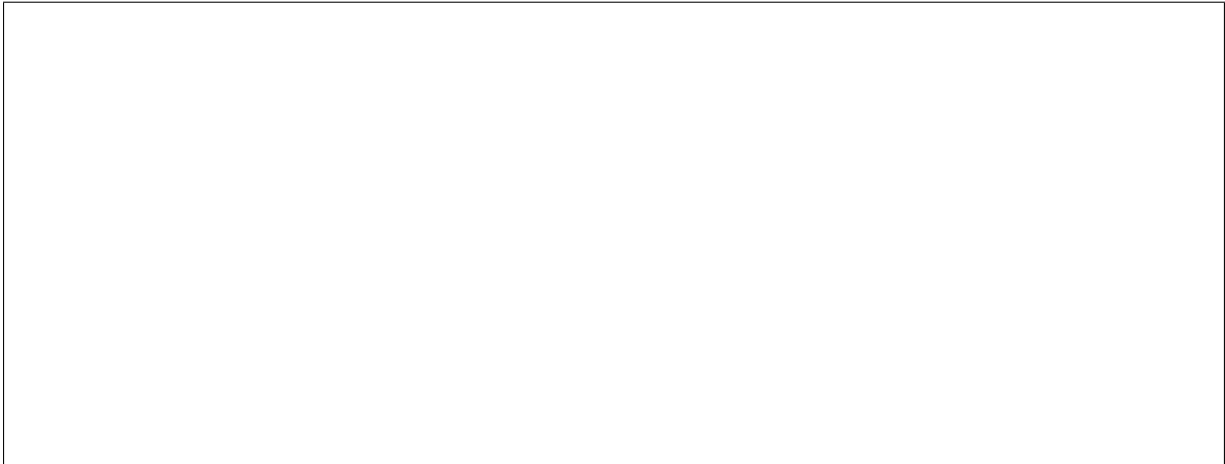
(continues on next page)

(continued from previous page)



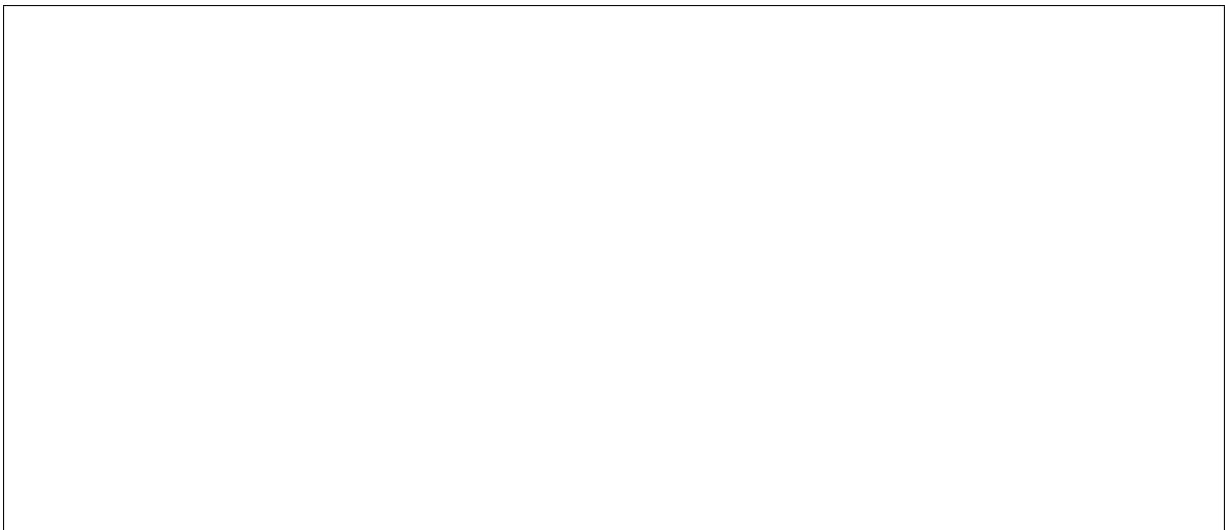
(continues on next page)

(continued from previous page)

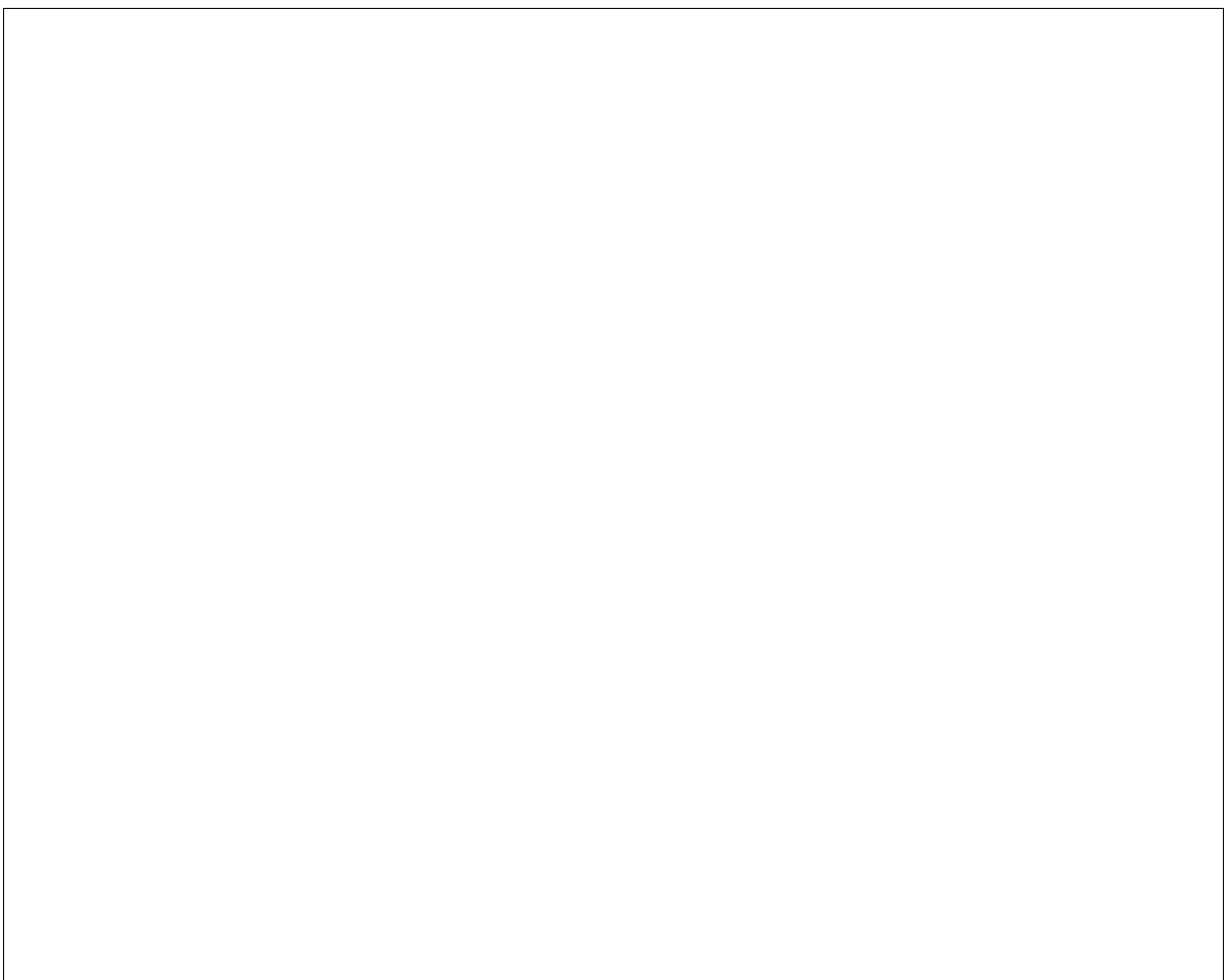




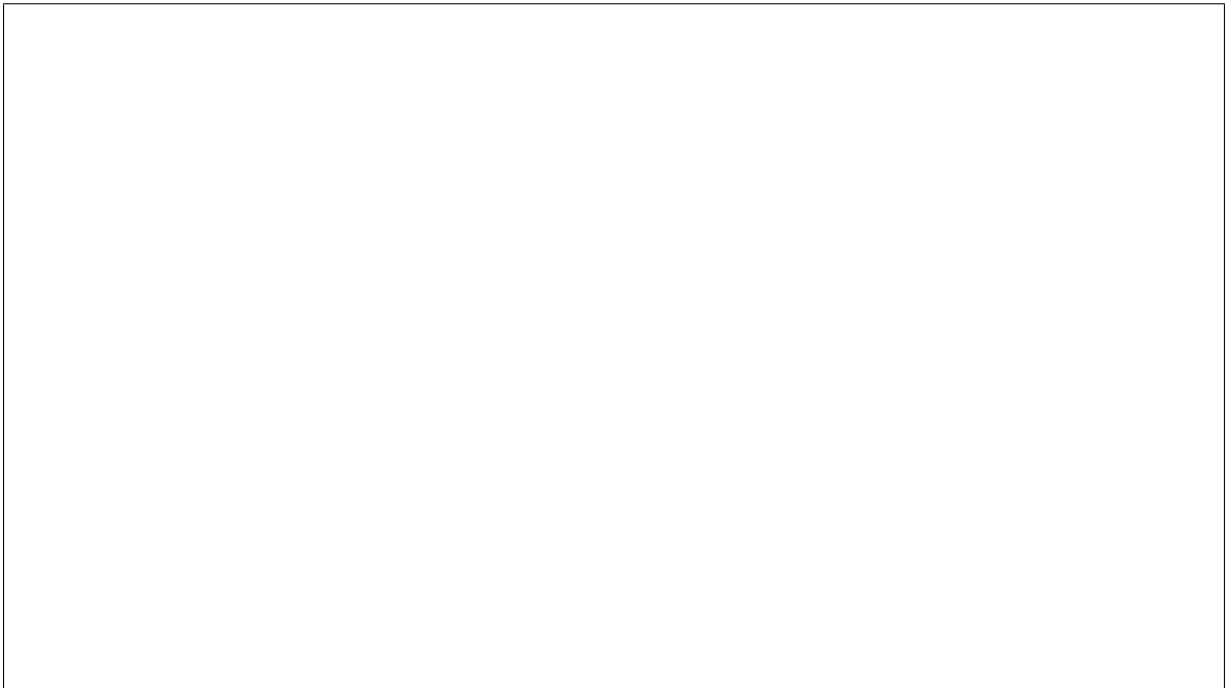
## **Set BIOS Config**





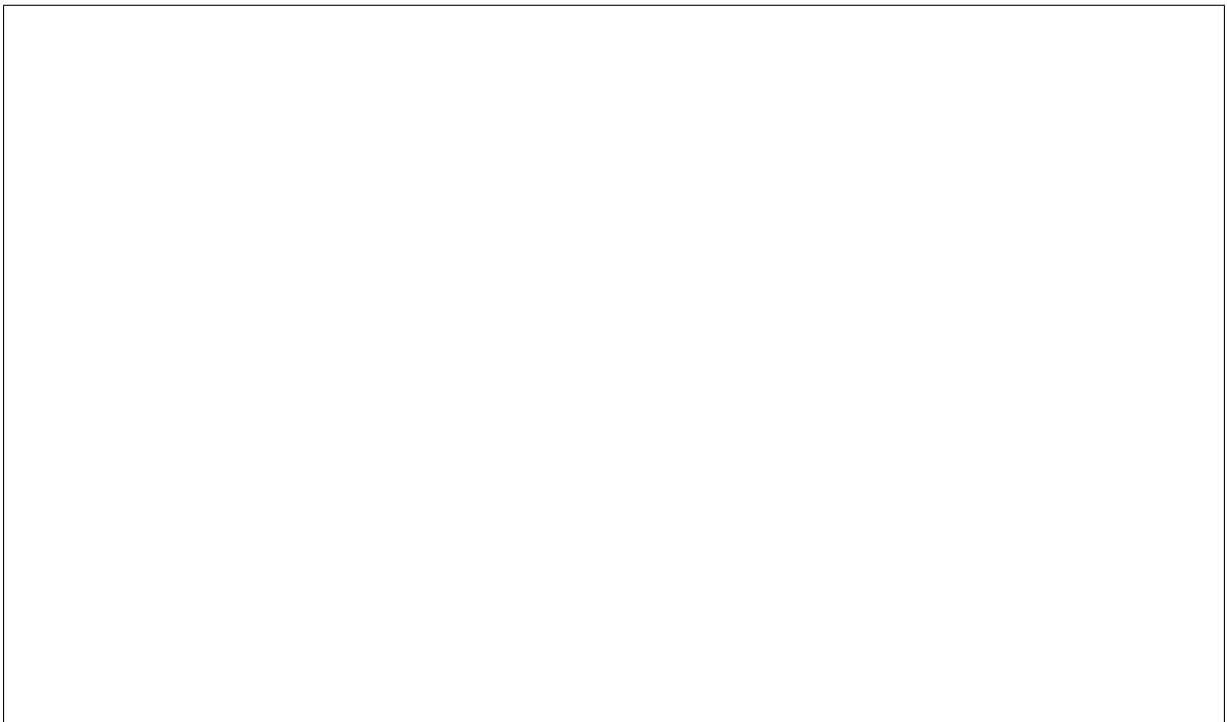


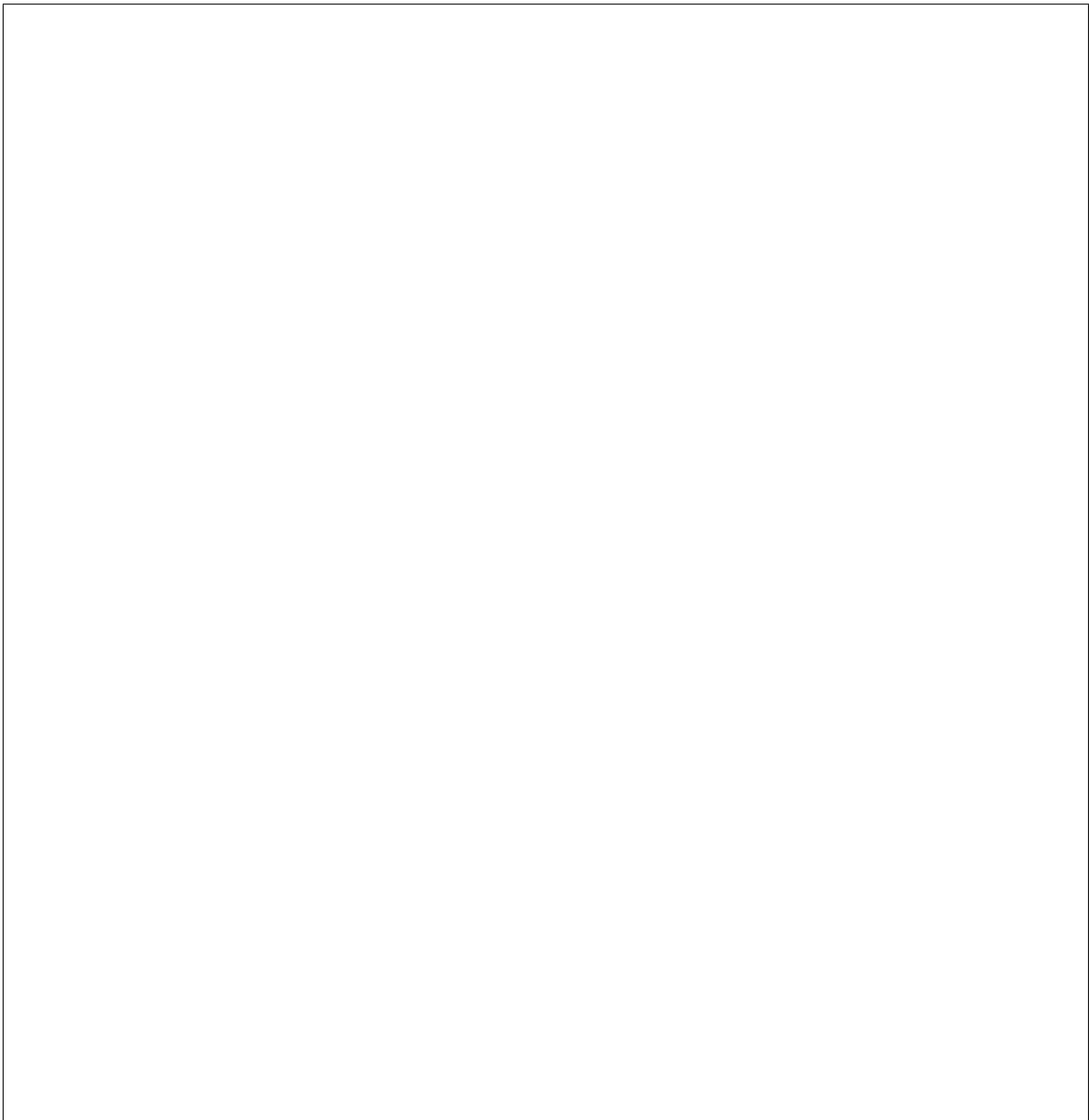
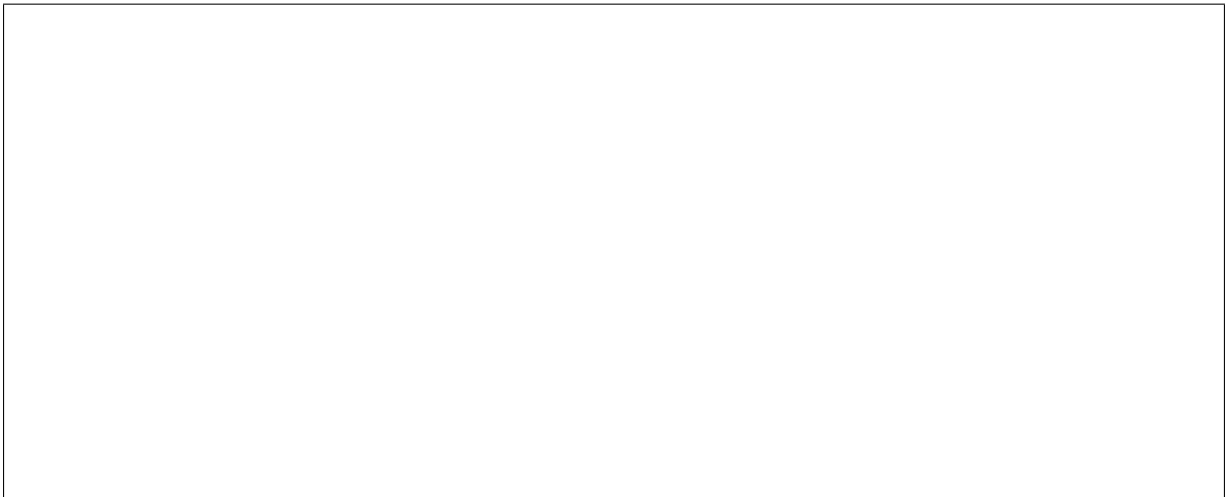




### **Commit BIOS Changes**

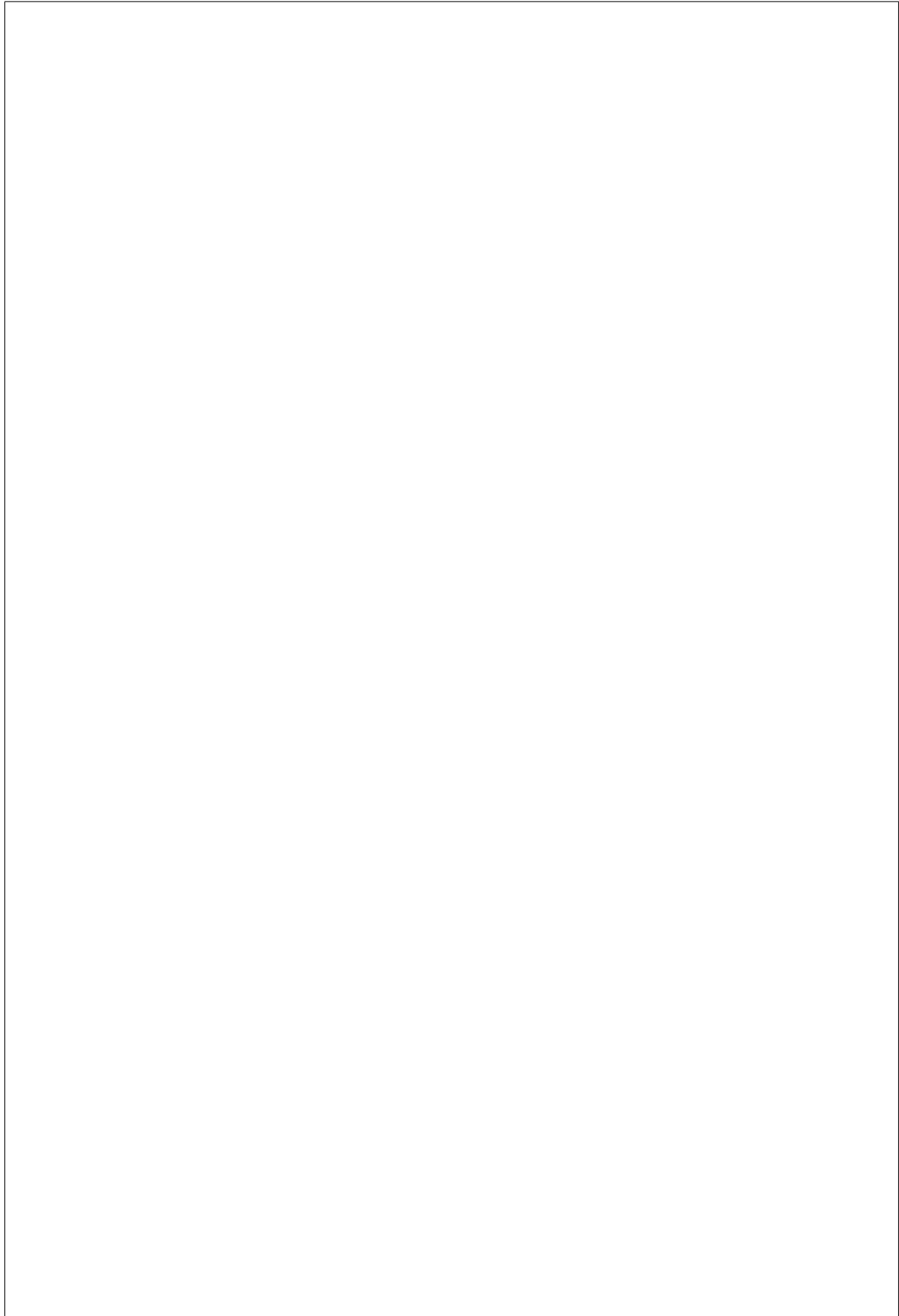
ically by the `commit_bios_config` call. If the `reboot` argument is not supplied, the job is still created, however it remains in the `scheduled` state until a reboot is performed. The reboot can be initiated through the IroniC power API.





(continues on next page)

(continued from previous page)

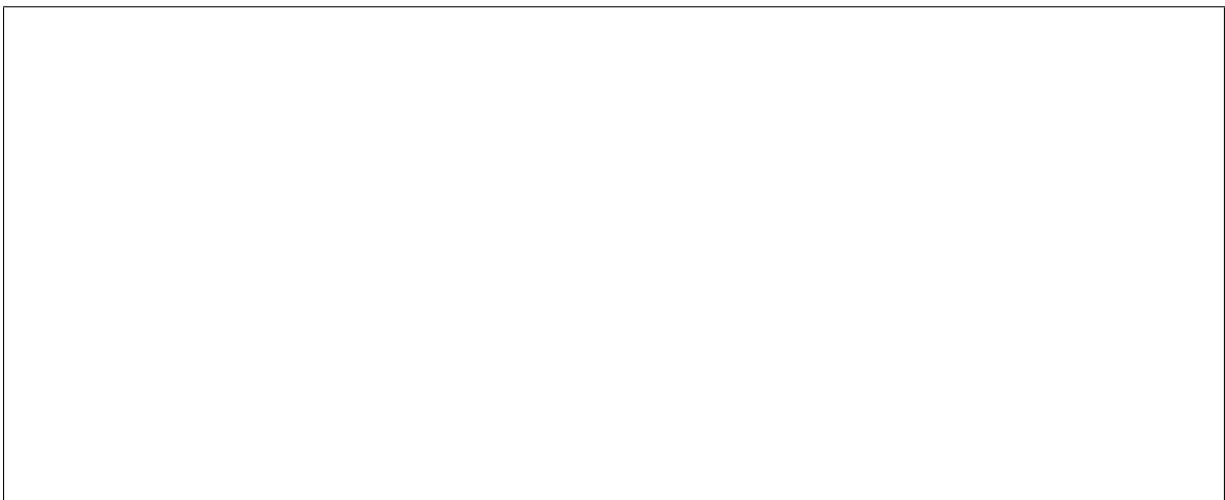


(continues on next page)

(continued from previous page)

---

## **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.



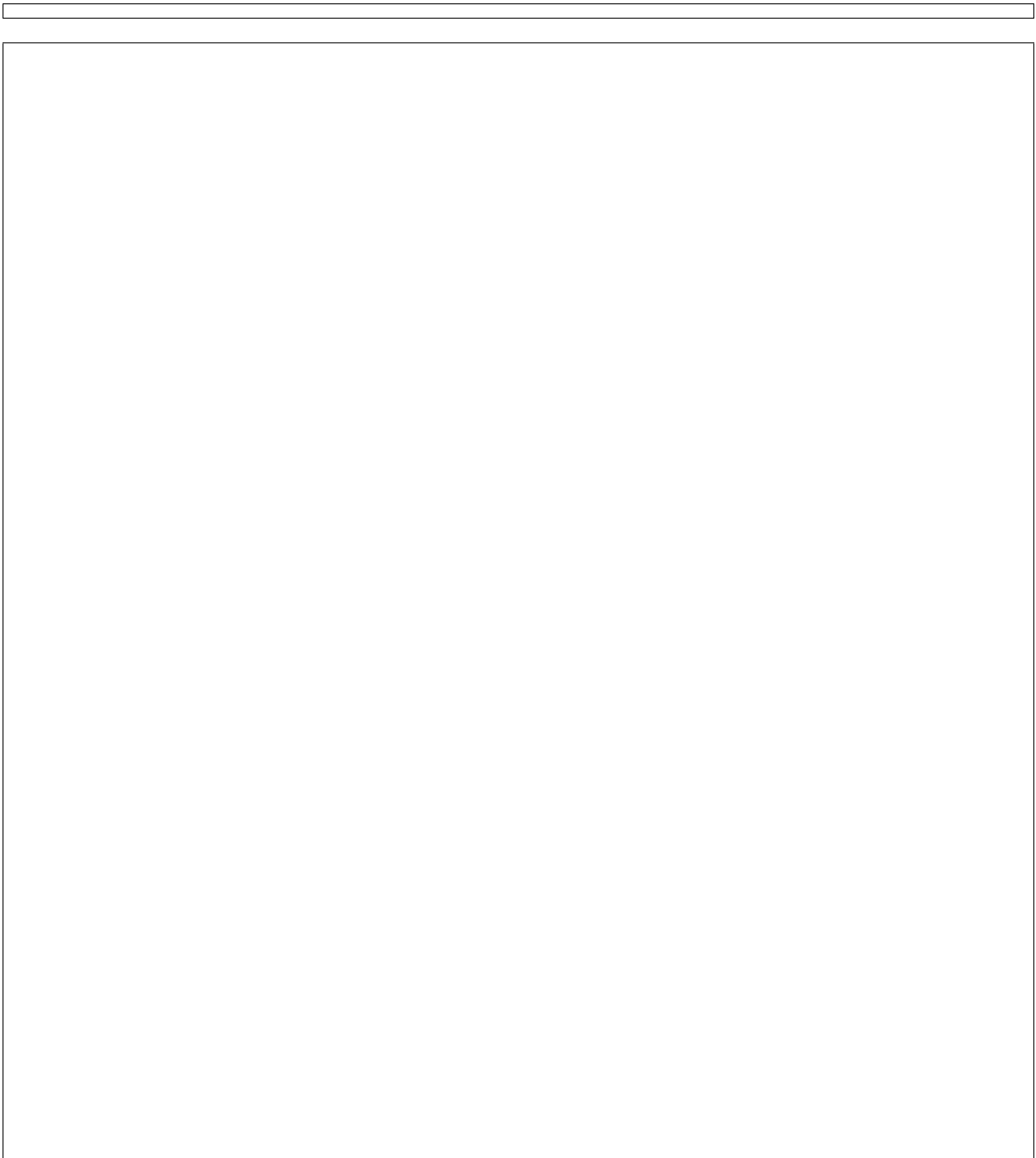
agement tasks as a traditional BIOS. However, UEFI does change the interfaces and data structures the BIOS uses to interact with I/O device firmware and operating system software. The primary intent of UEFI is to eliminate shortcomings in the traditional BIOS environment, enabling system firmware to continue scaling with industry trends.





(continues on next page)

(continued from previous page)



## **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.

### Vendor passthru timeout

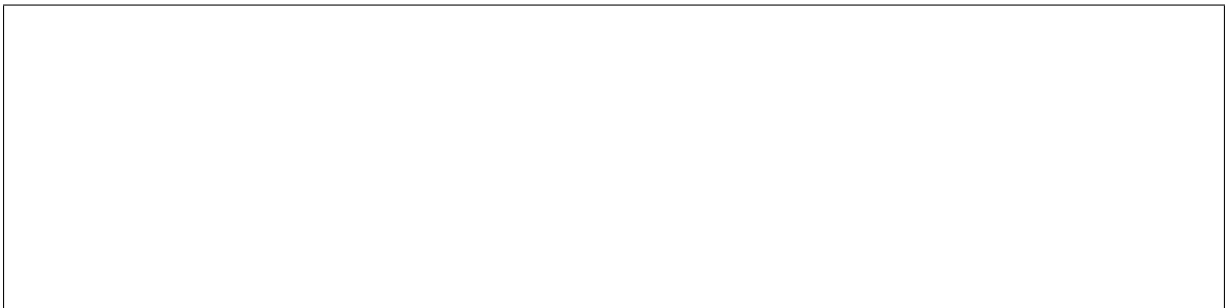






**Timeout when powering off**

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



## **iLO driver**

### **Overview**

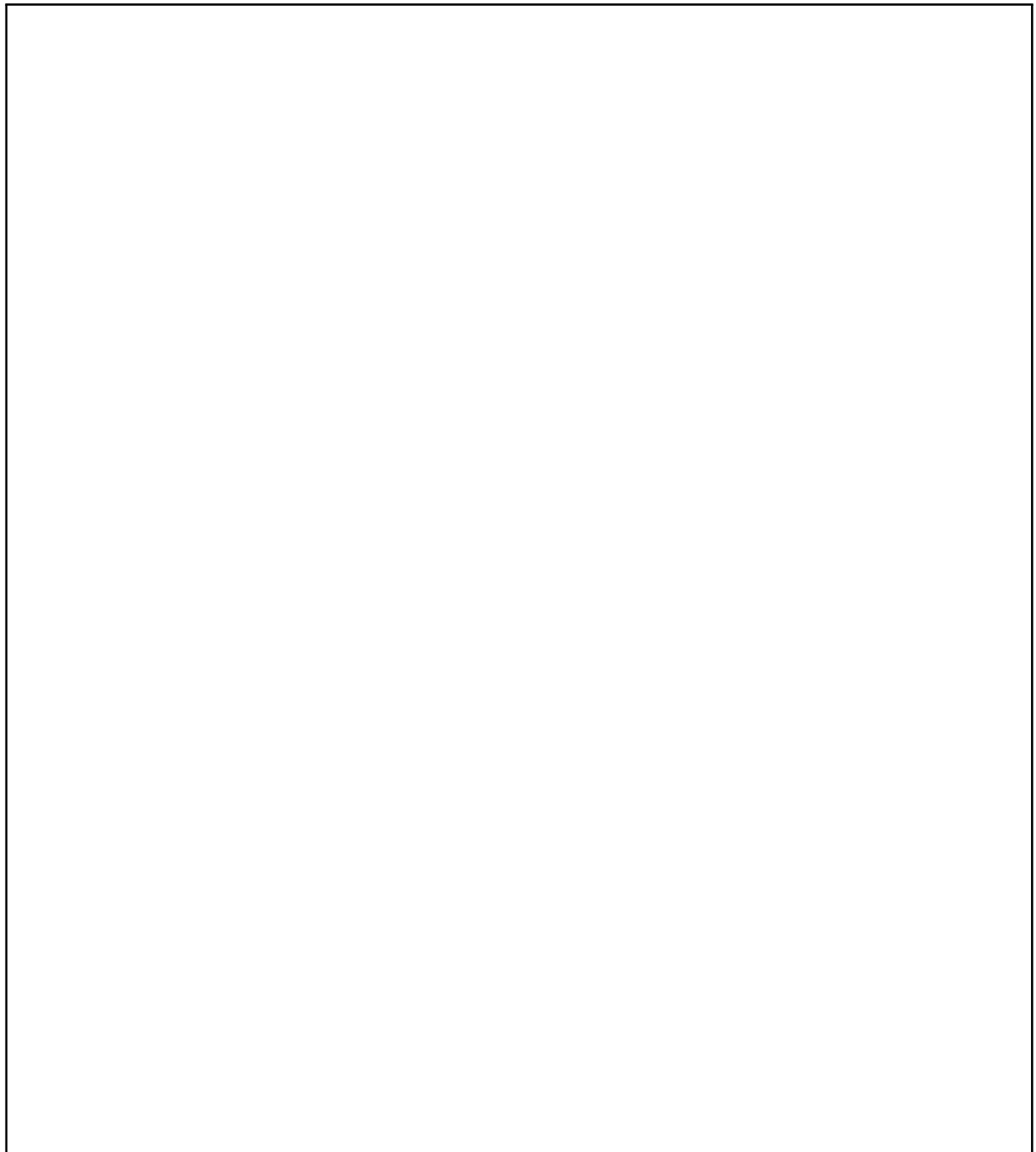
`ilo` hardware type supports ProLiant Gen10 systems which have [iLO 5 management engine](#). iLO5 conforms to [Redfish](#) API and hence hardware type `redfish` (see [Redfish driver](#)) is also an option for this kind of hardware but it lacks the iLO specific features.

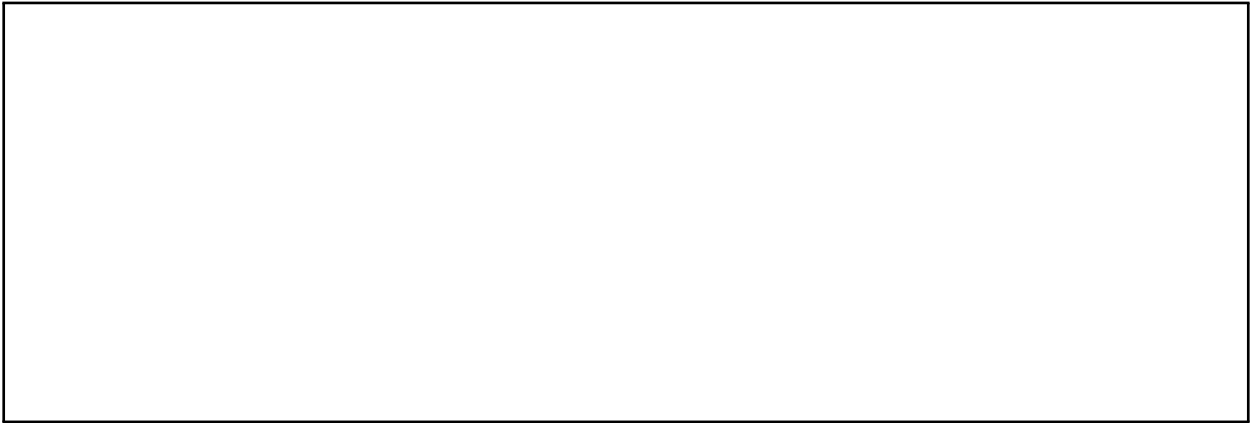


## **Hardware type**

*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`.







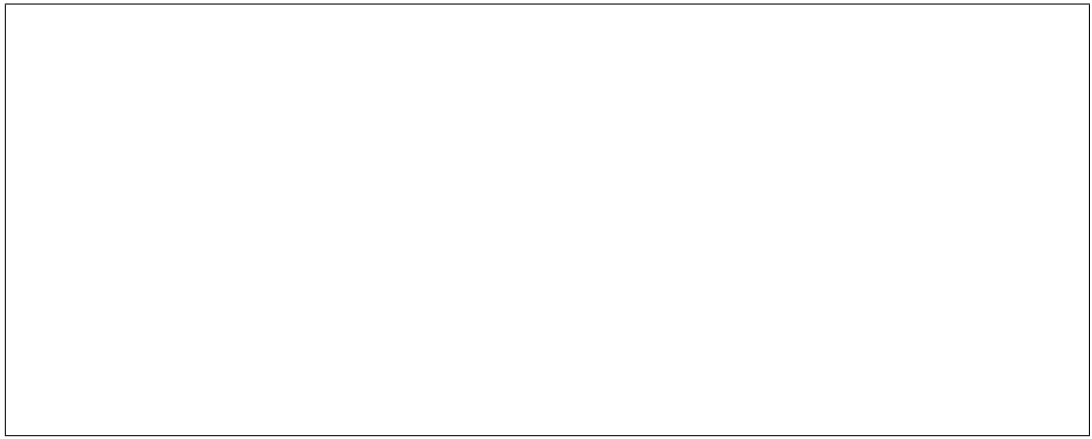




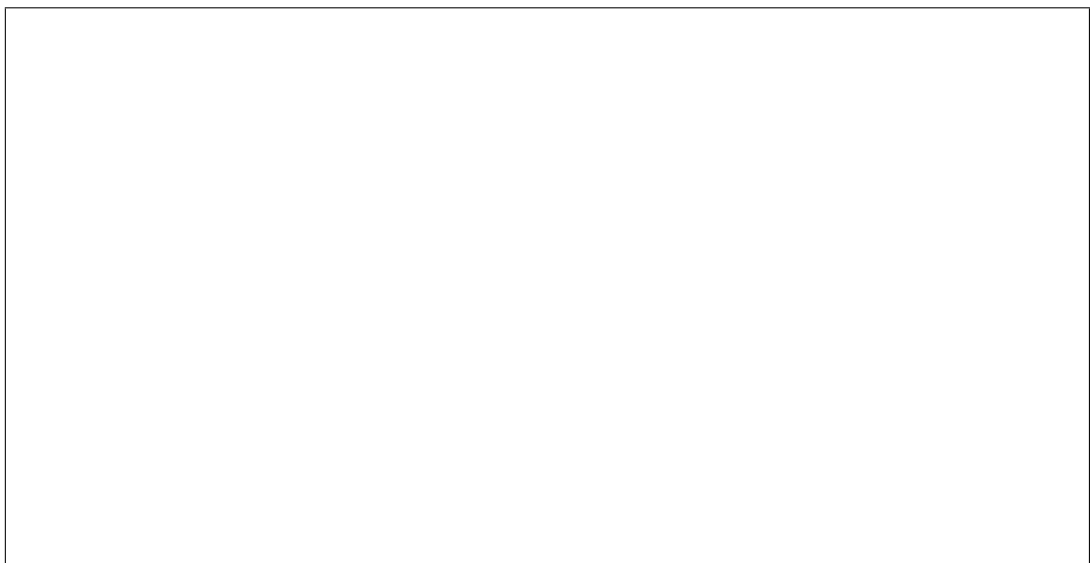


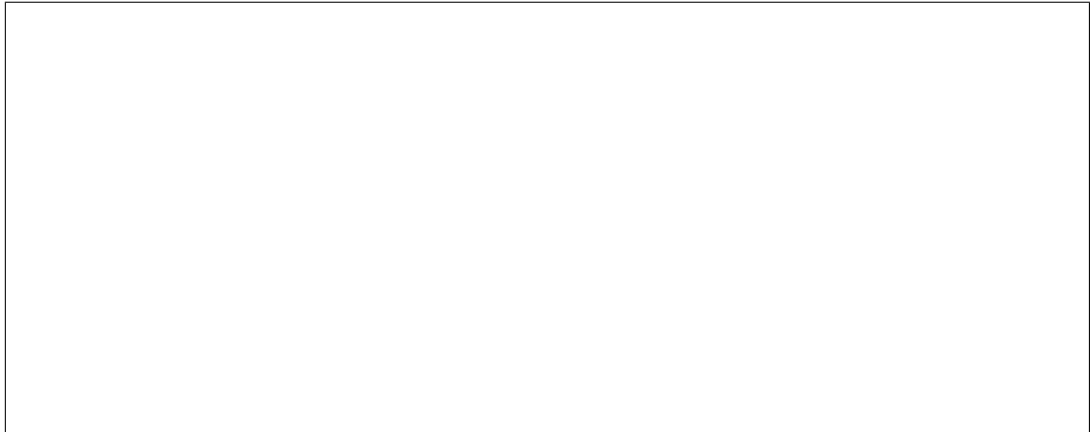


## **Hardware interfaces**



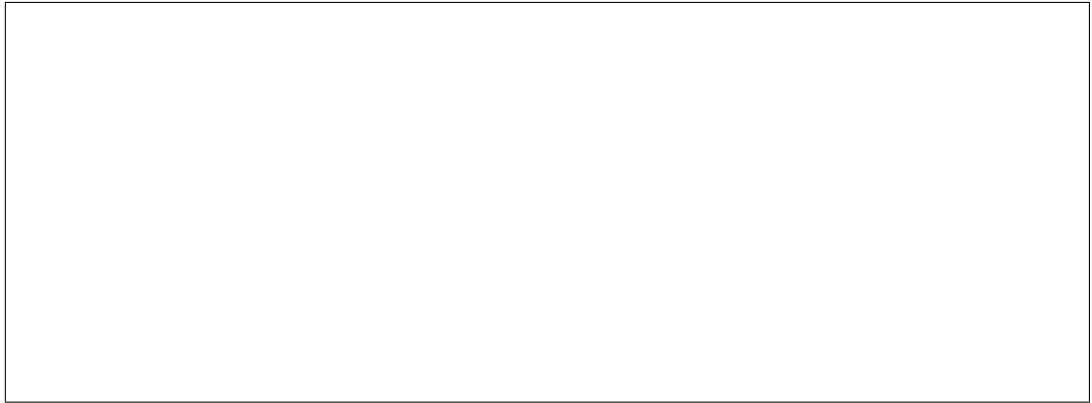
iPXE respectively for deployment(just like *PXE boot*). These interfaces do not require iLO Advanced license. They can be enabled by using the `[DEFAULT]enabled_boot_interfaces` option in `ironic.conf` as given below:

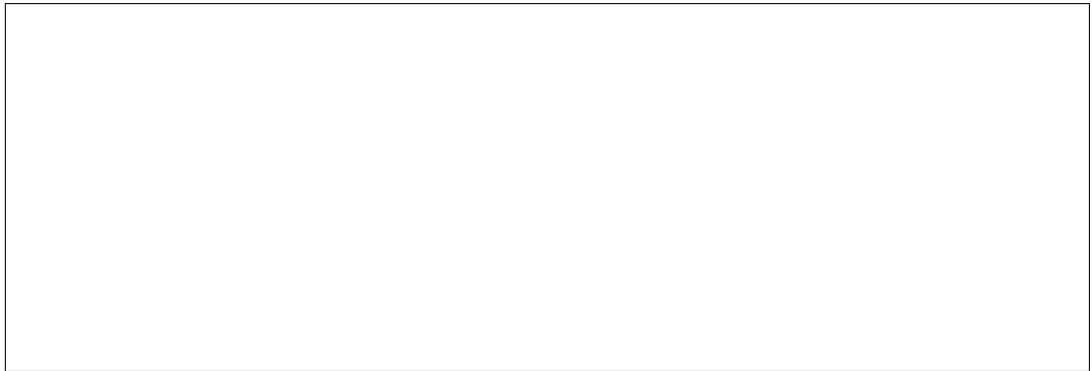






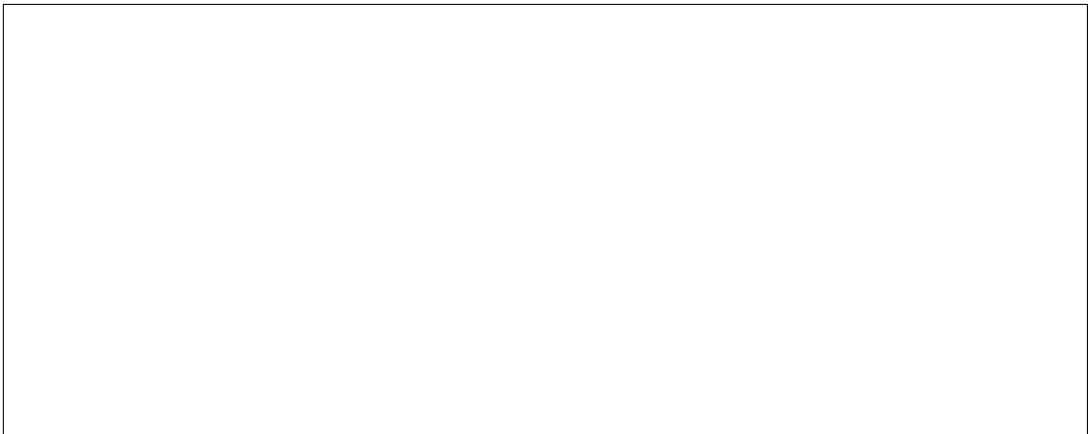
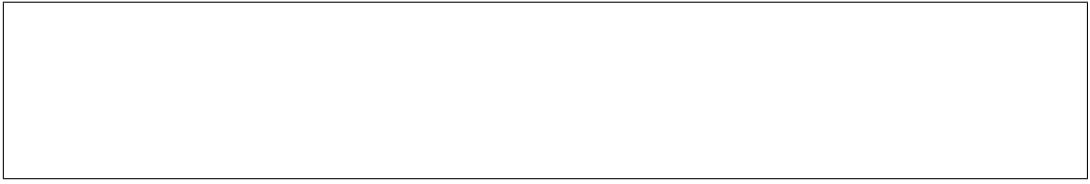


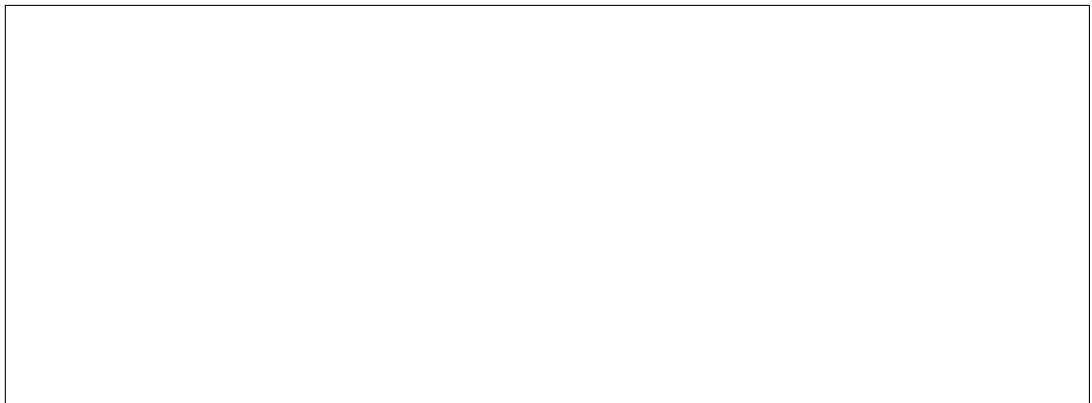




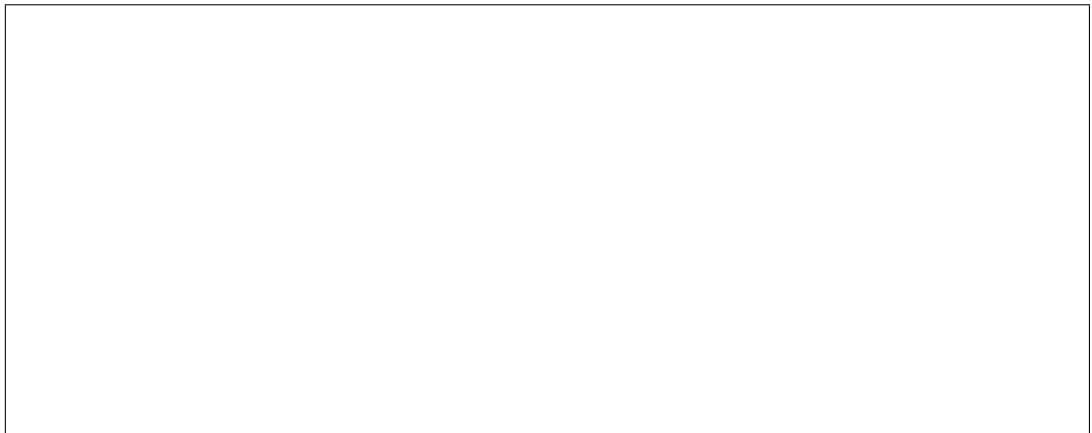
(continues on next page)

(continued from previous page)

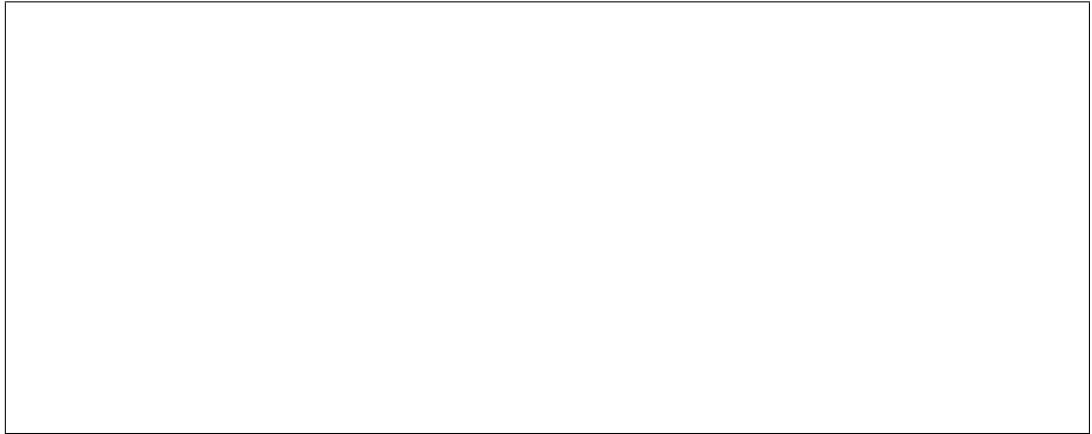






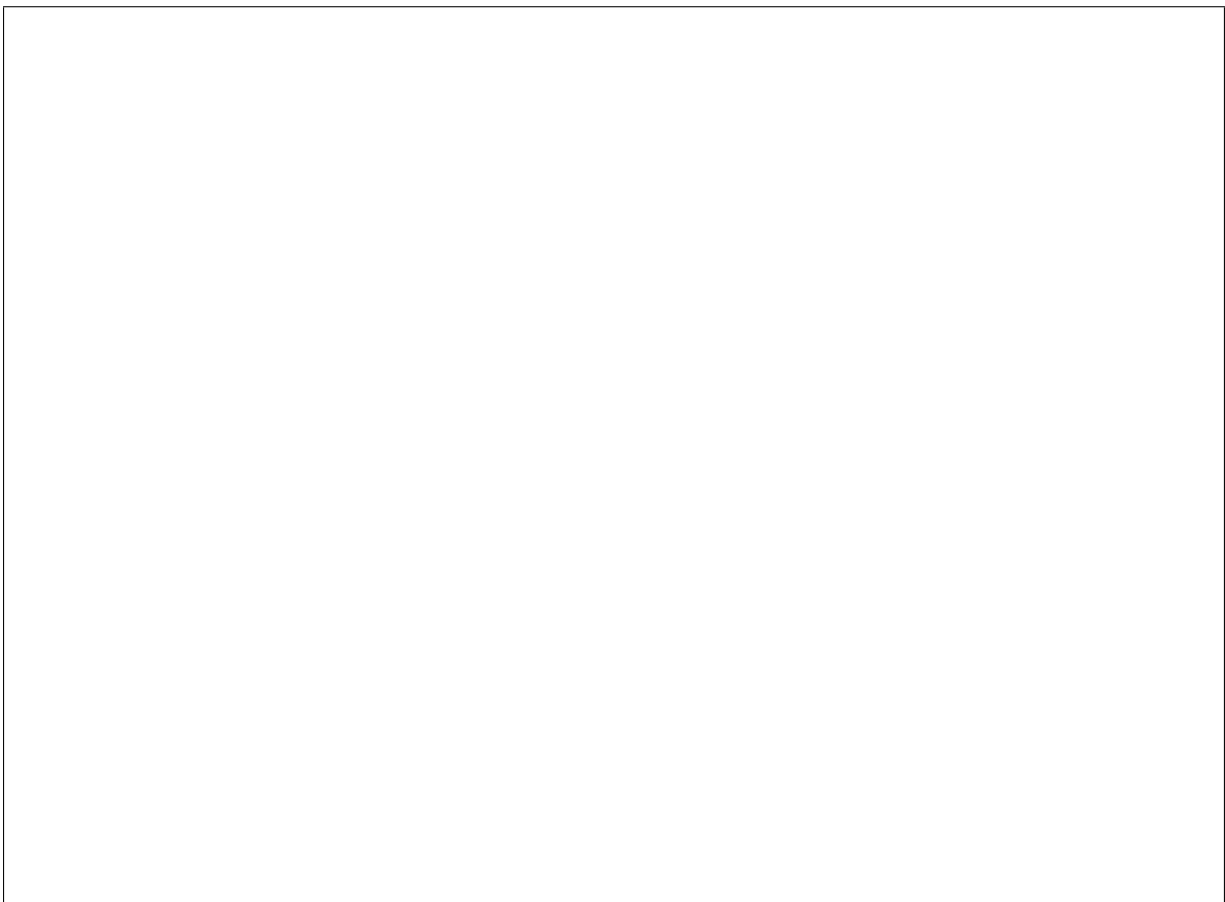






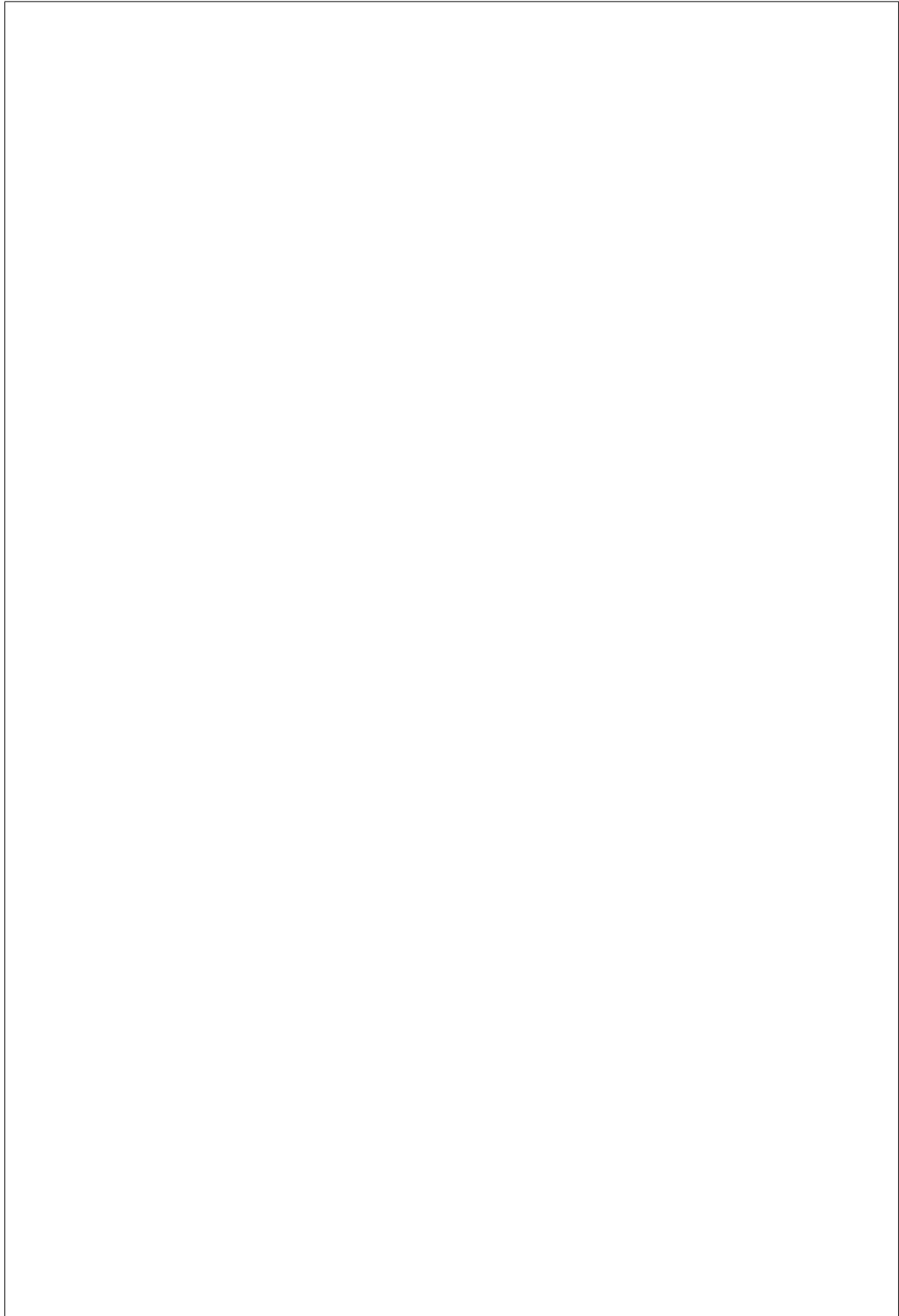






(continues on next page)

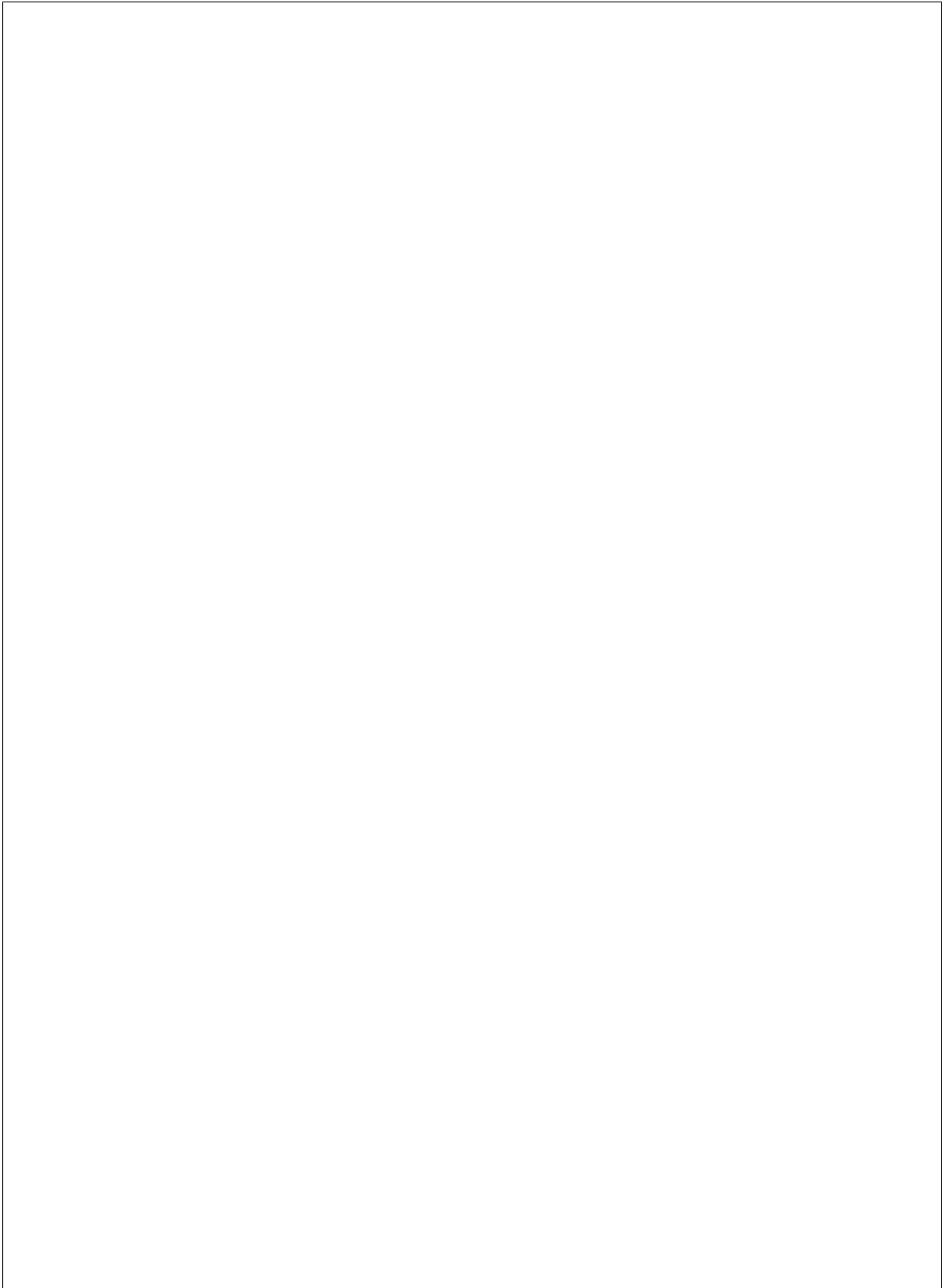
(continued from previous page)



(continues on next page)

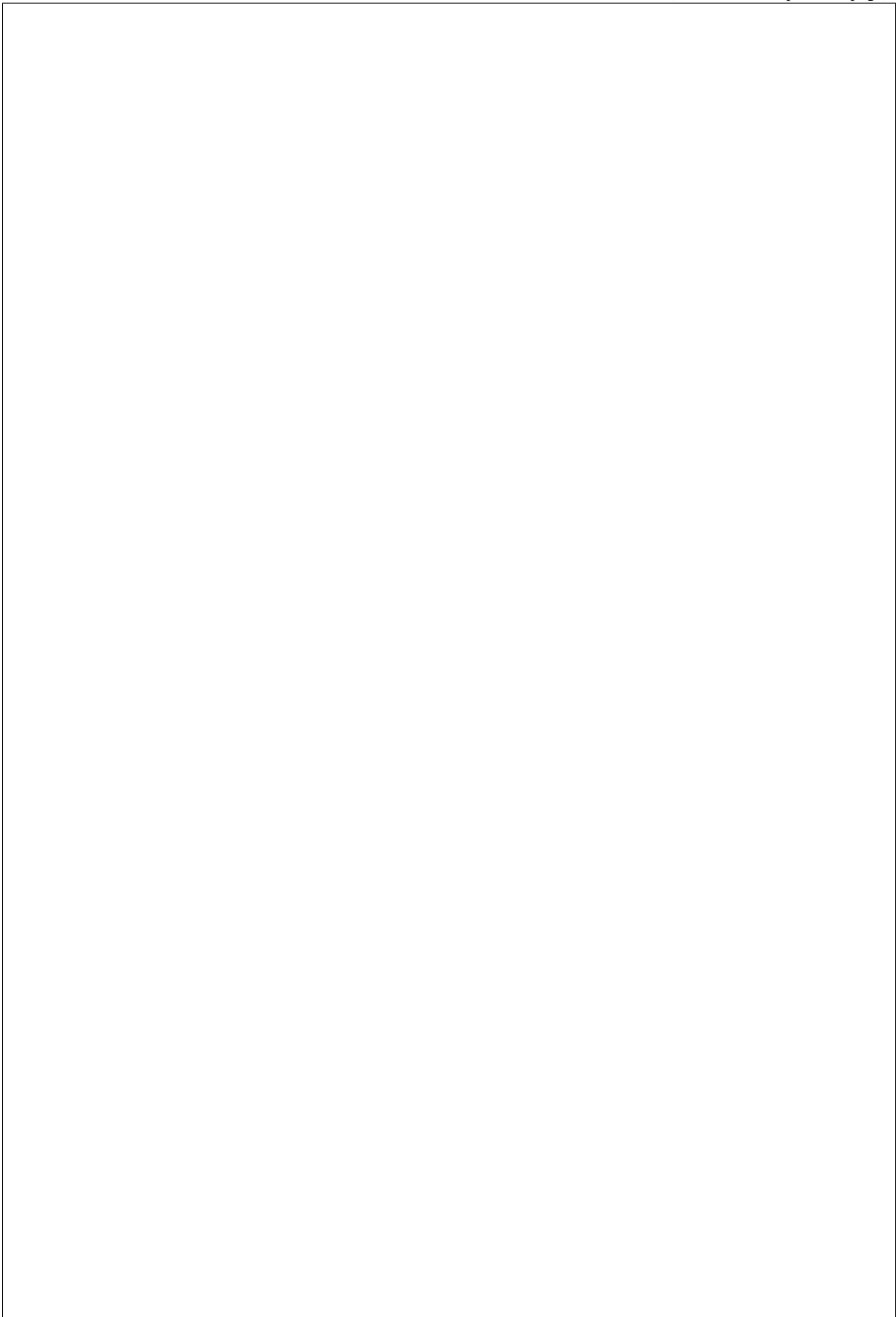
(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

--

## **Node configuration**



























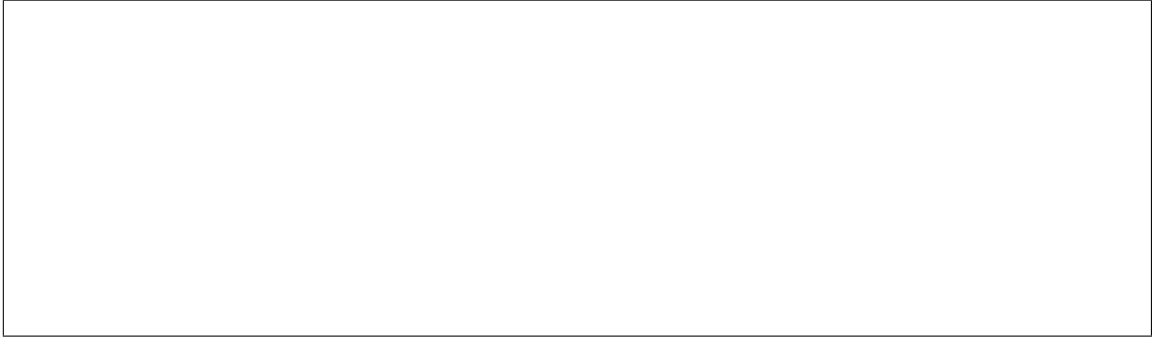
UEFI.





**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**

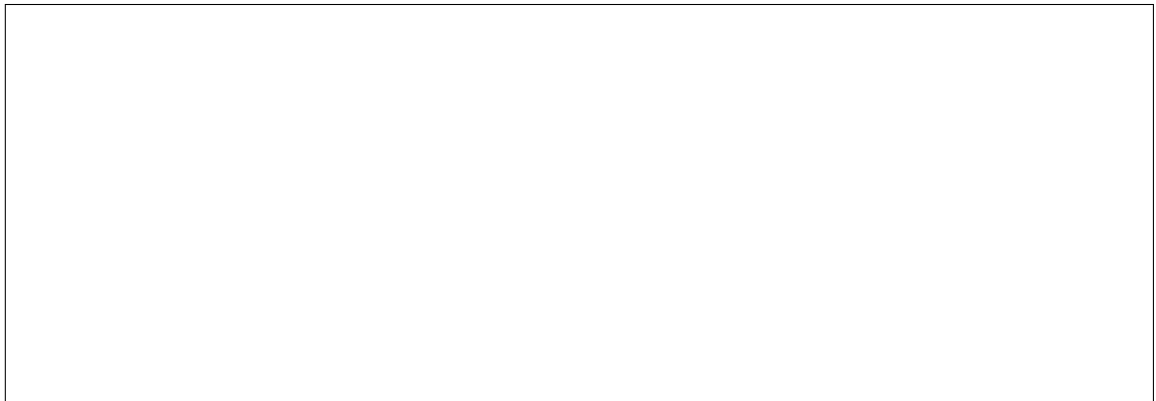


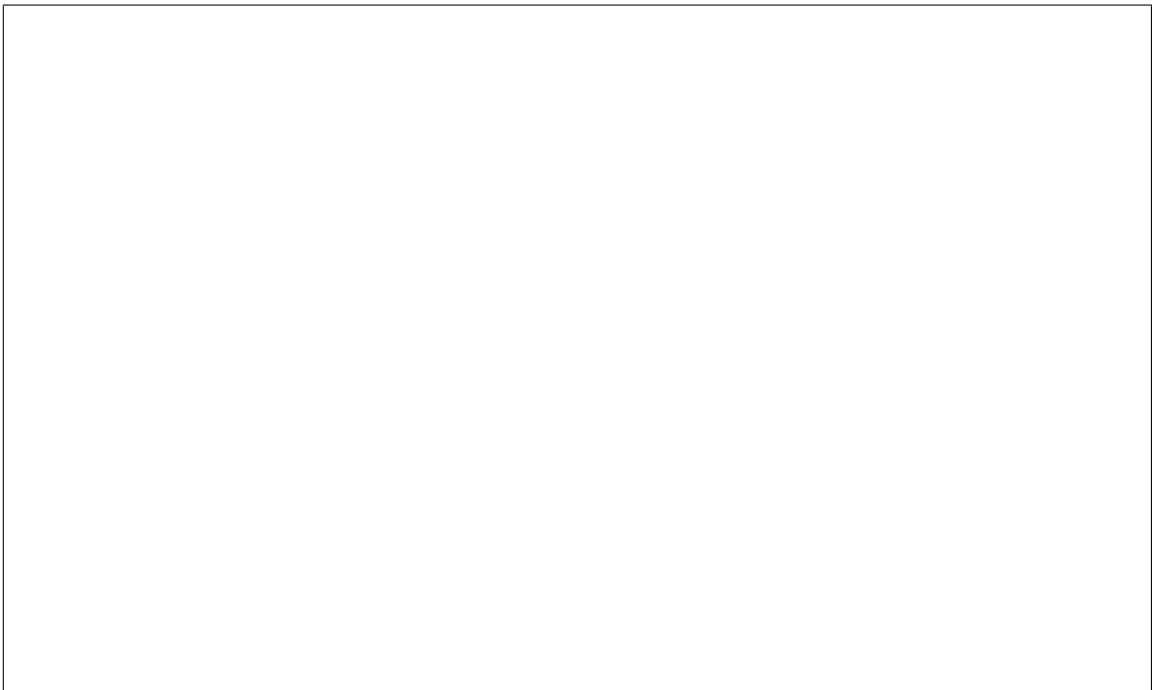


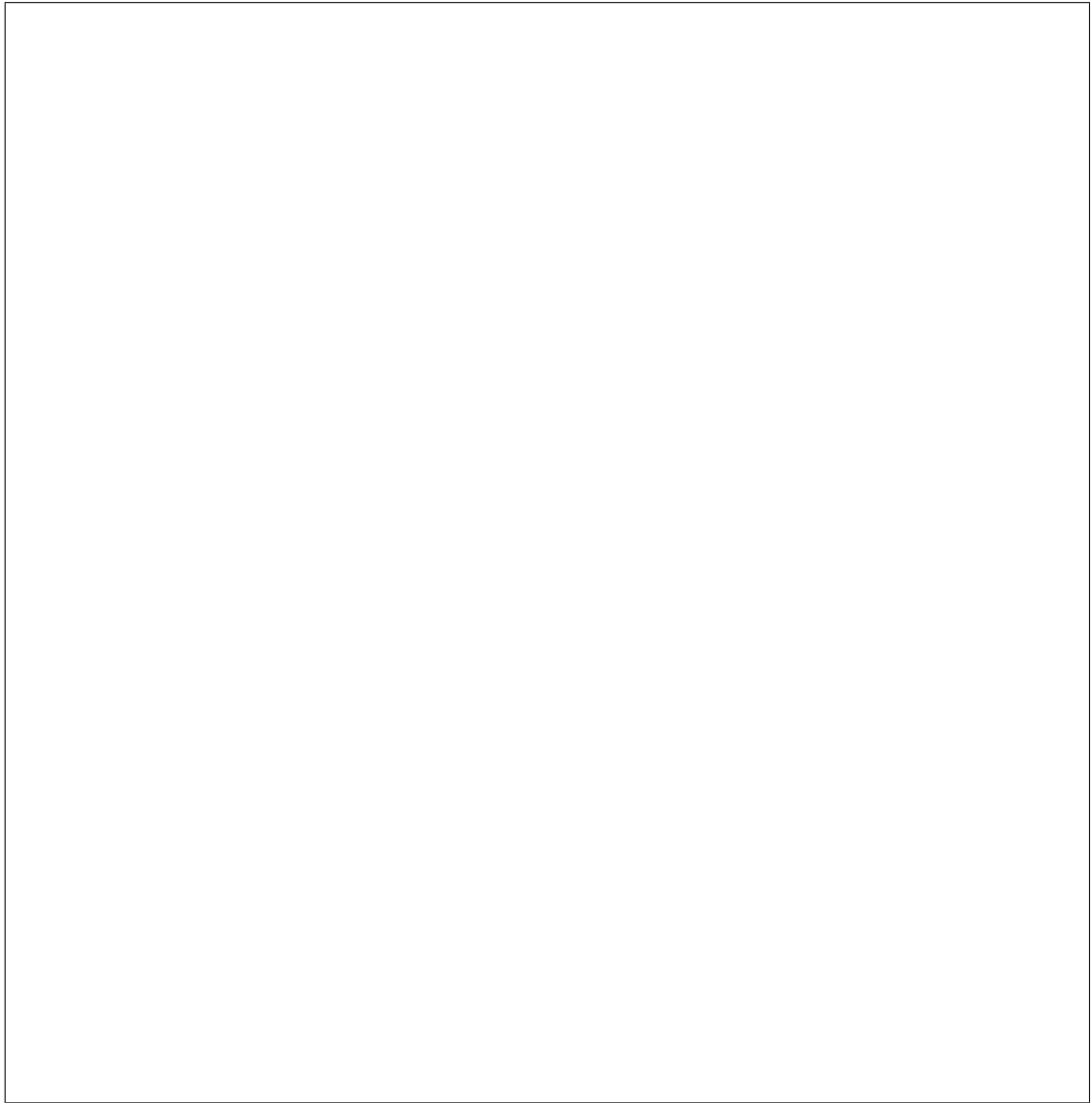
sion.

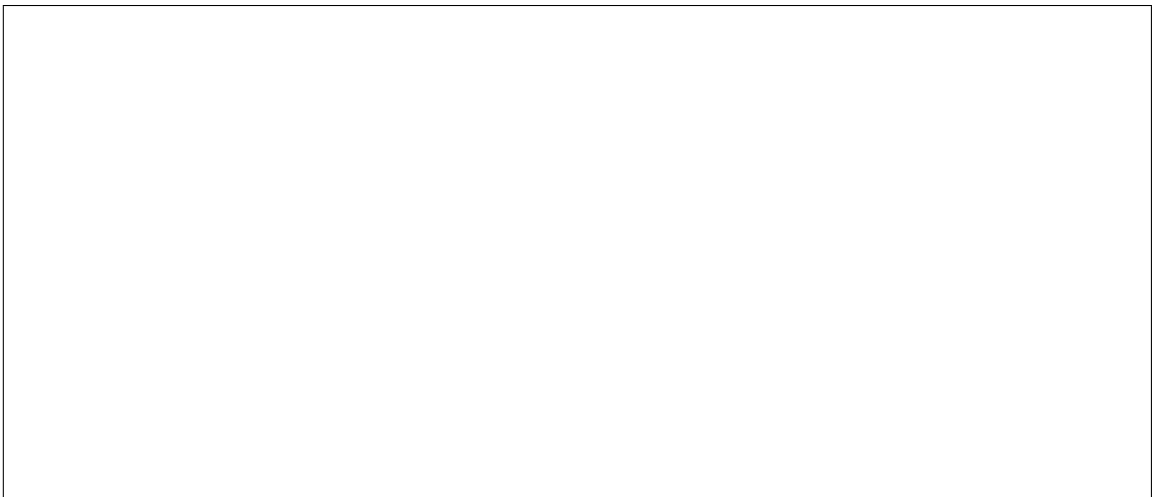
**Different configuration for ilo hardware type**

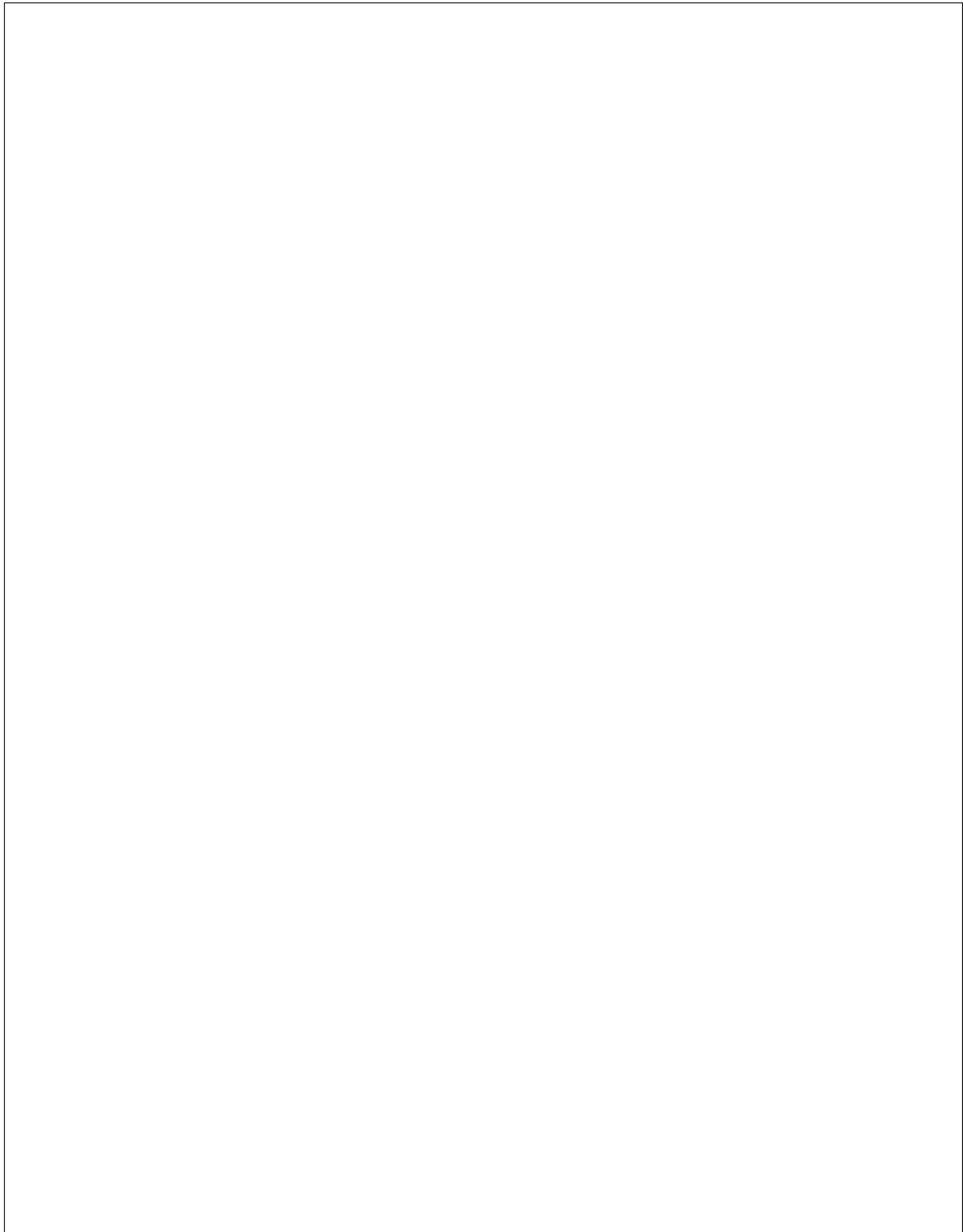
**Glance Configuration**





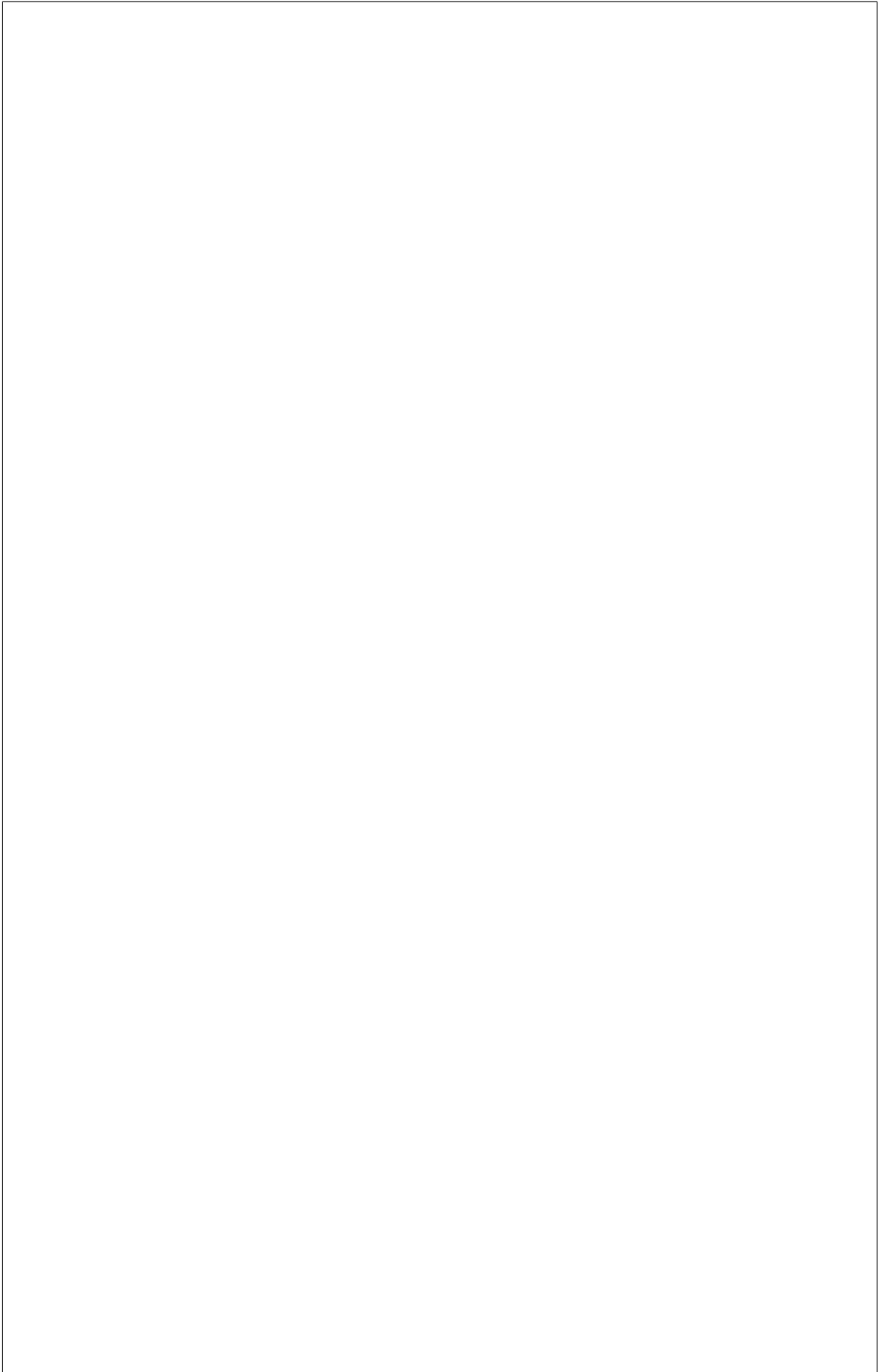






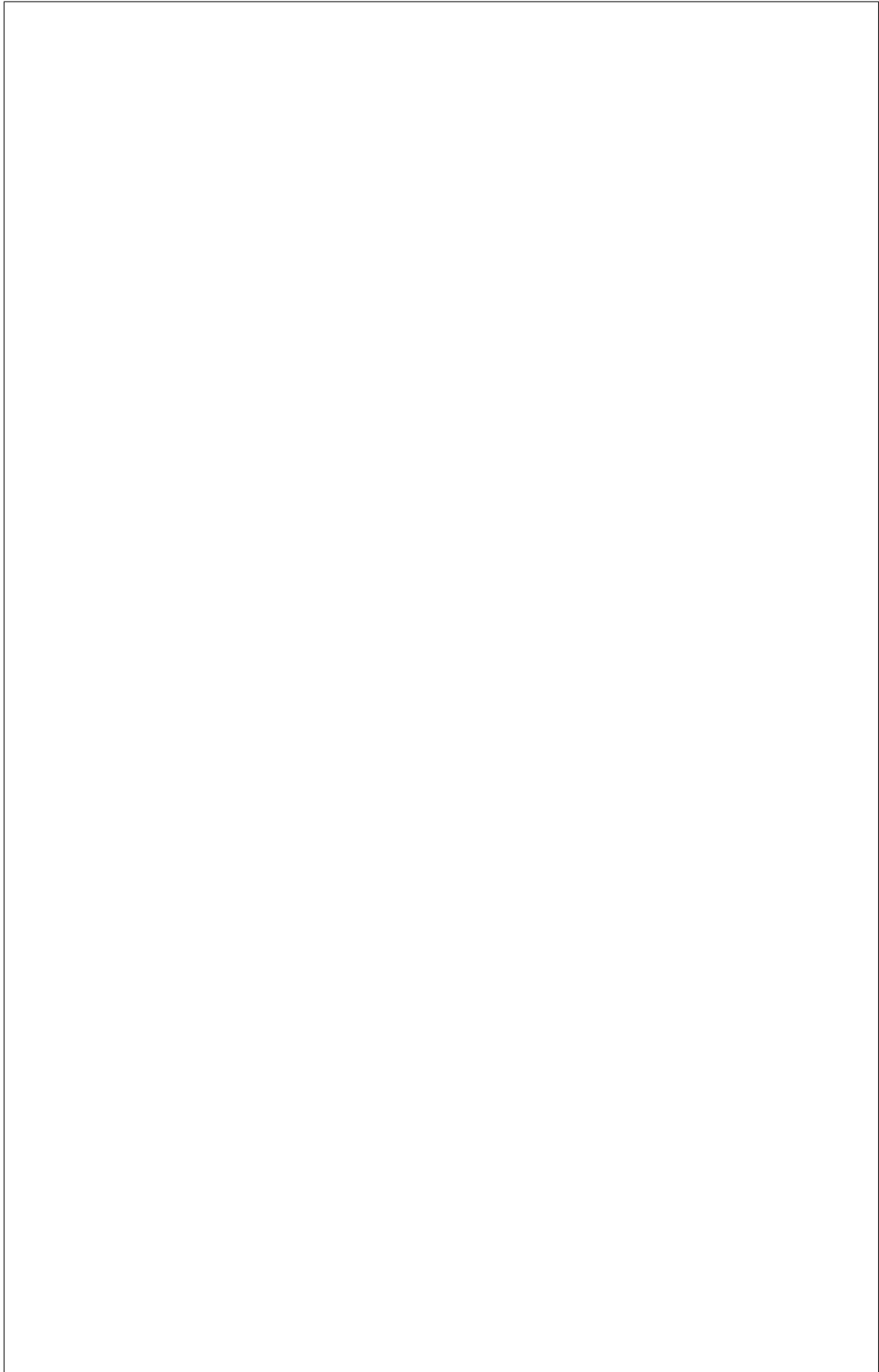
(continues on next page)

(continued from previous page)



(continues on next page)

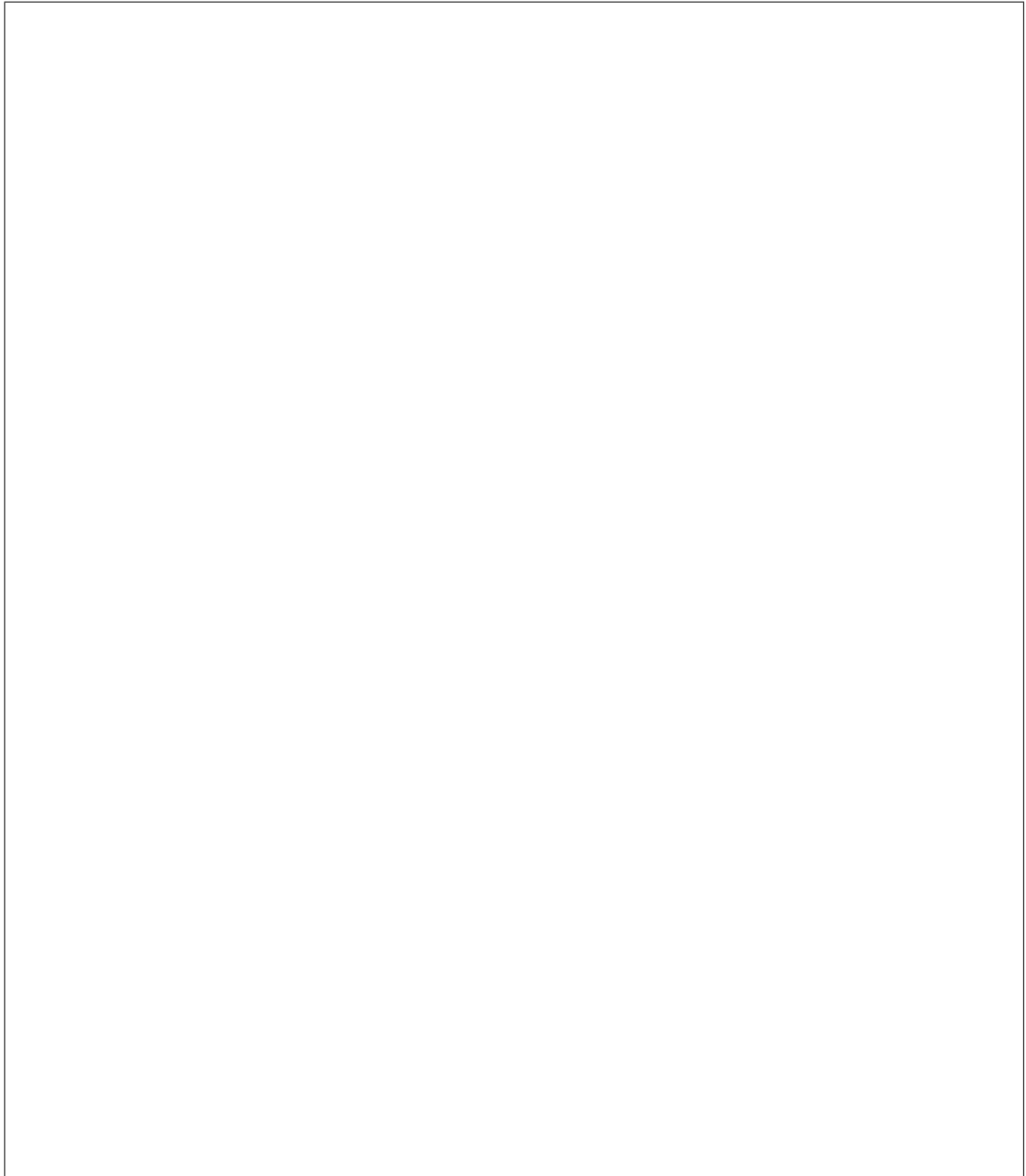
(continued from previous page)



(continues on next page)



(continued from previous page)

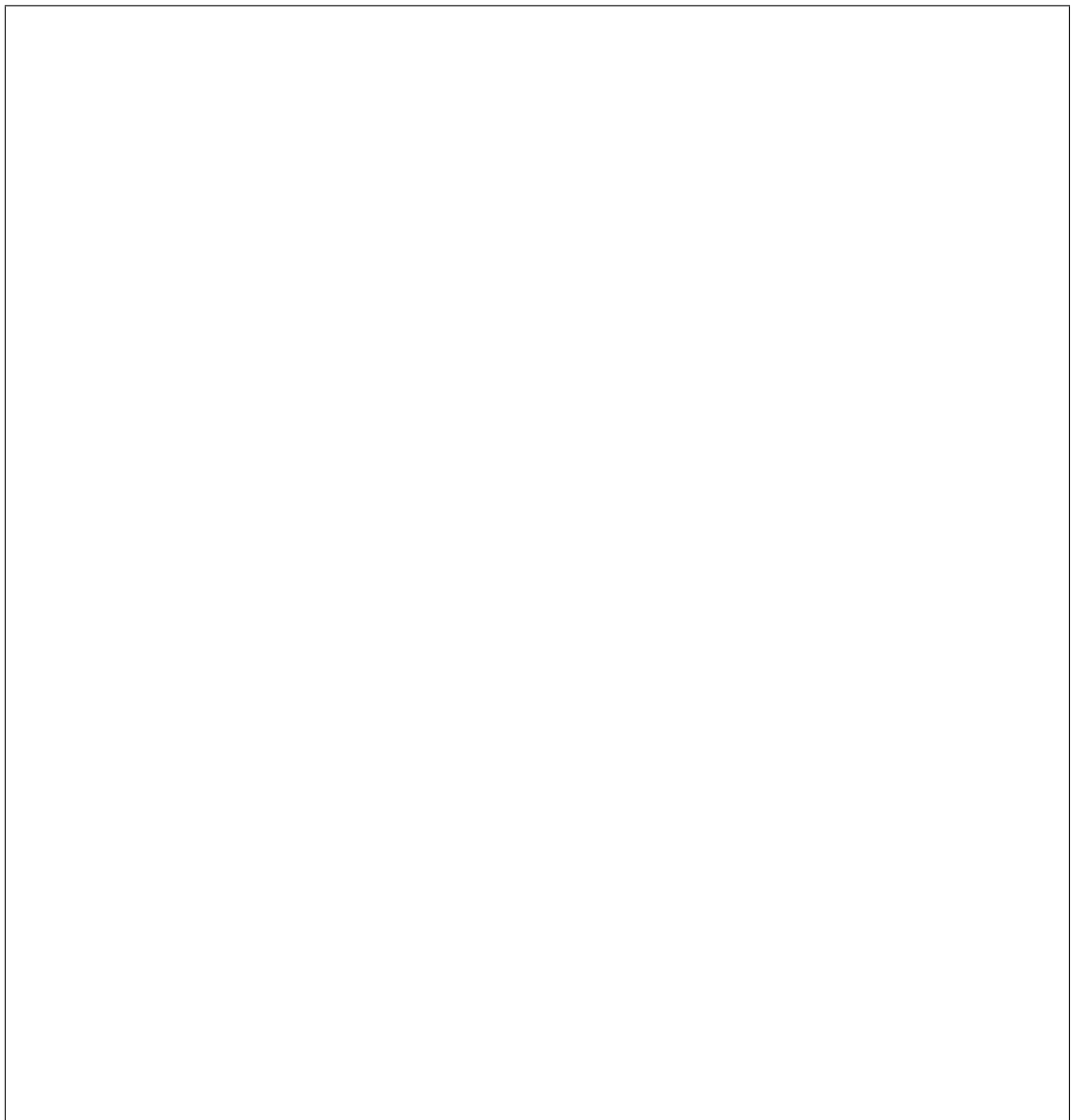


(continues on next page)

(continued from previous page)

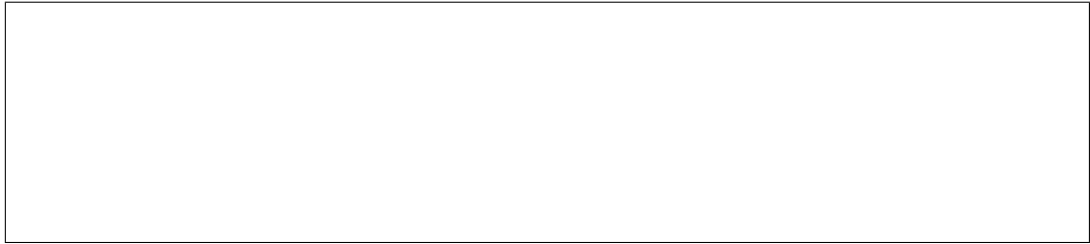
---

## **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**

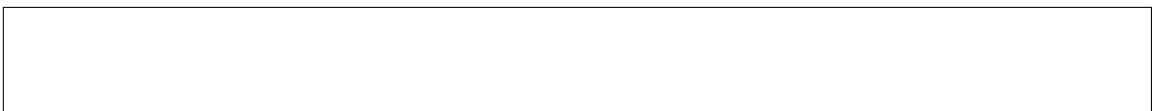
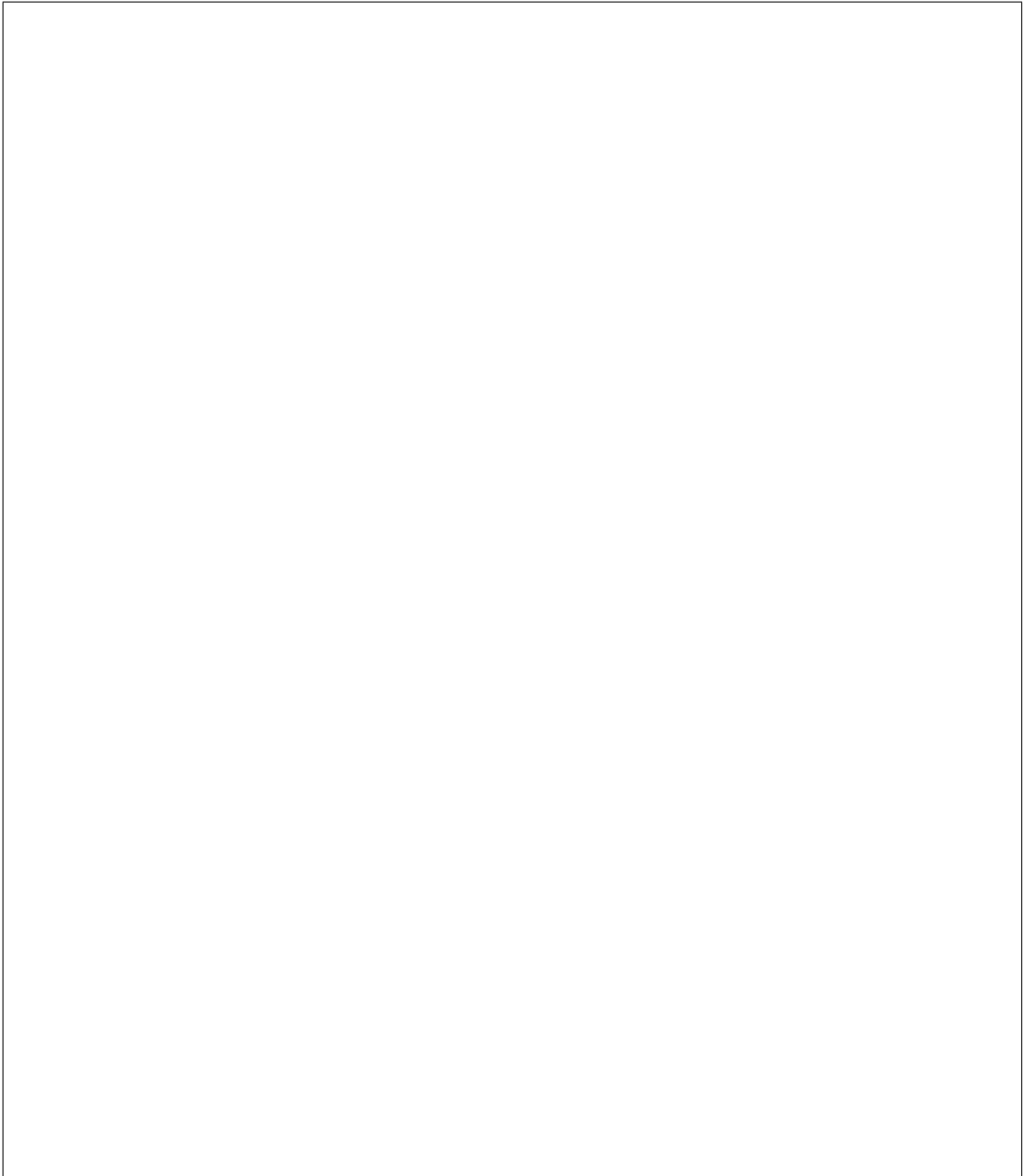




(continues on next page)



(continued from previous page)



(continues on next page)

(continued from previous page)

---

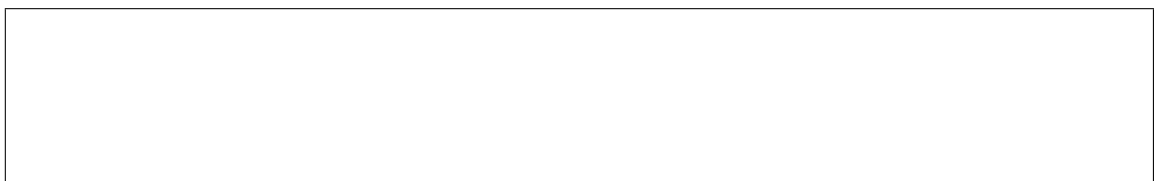
## **Optional functionalities for the `ilo` hardware type**

### **Boot mode support**



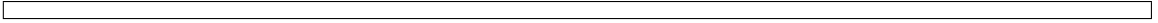






(continues on next page)

(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**





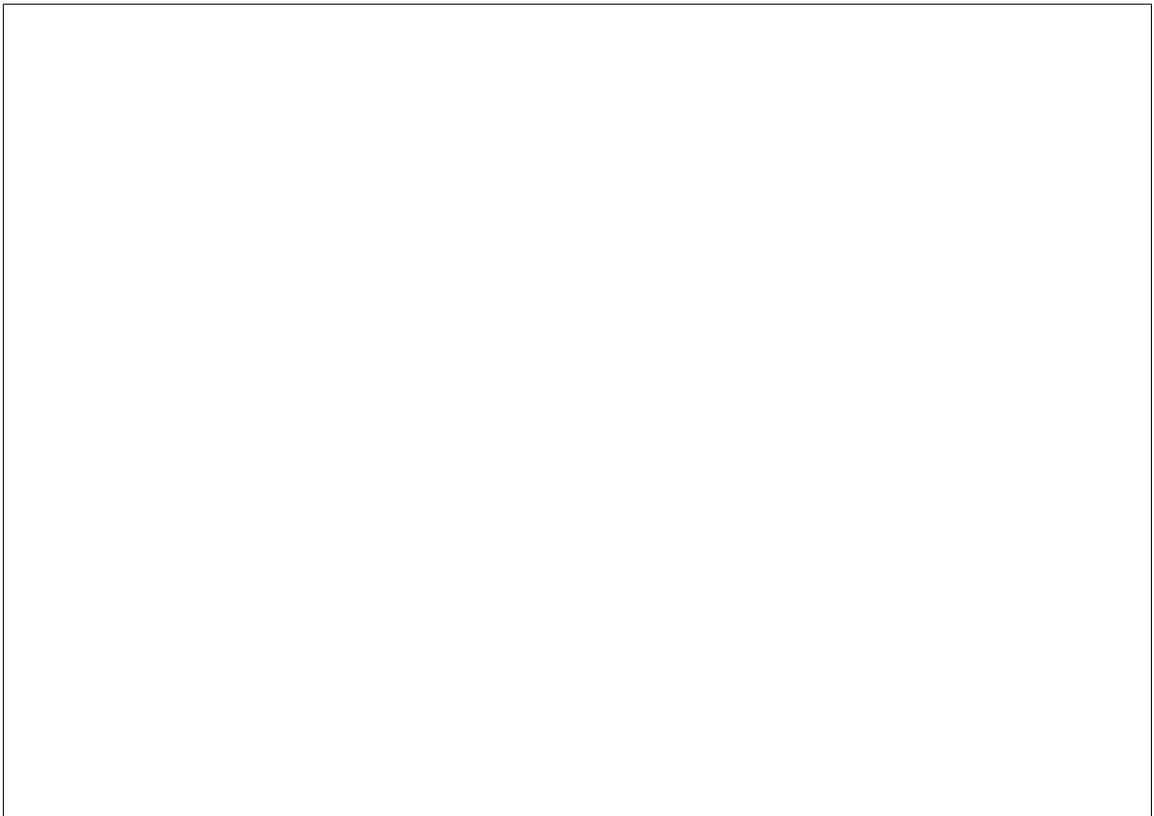




disks visible to SSA in Proliant servers only with the ramdisk created using diskimage-builder from Ocata release. By default, this step is disabled. See [Disk Erase Support](#) for more details.

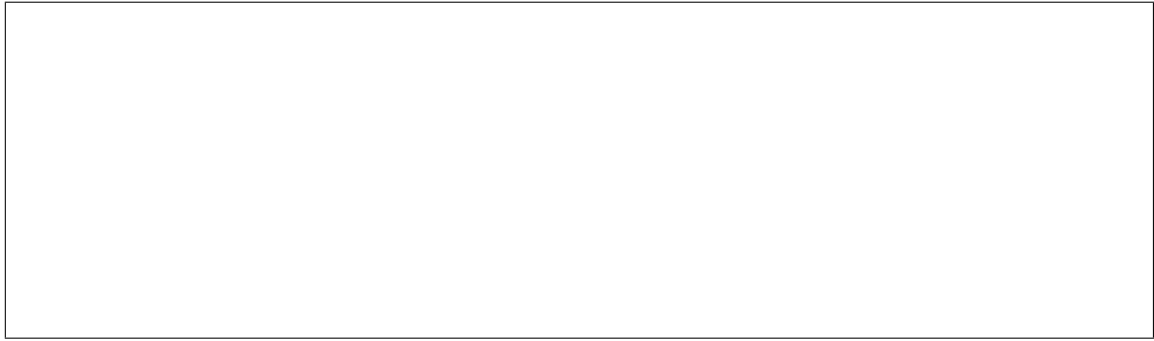


priority should be updated in `ironic.conf`.



(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**











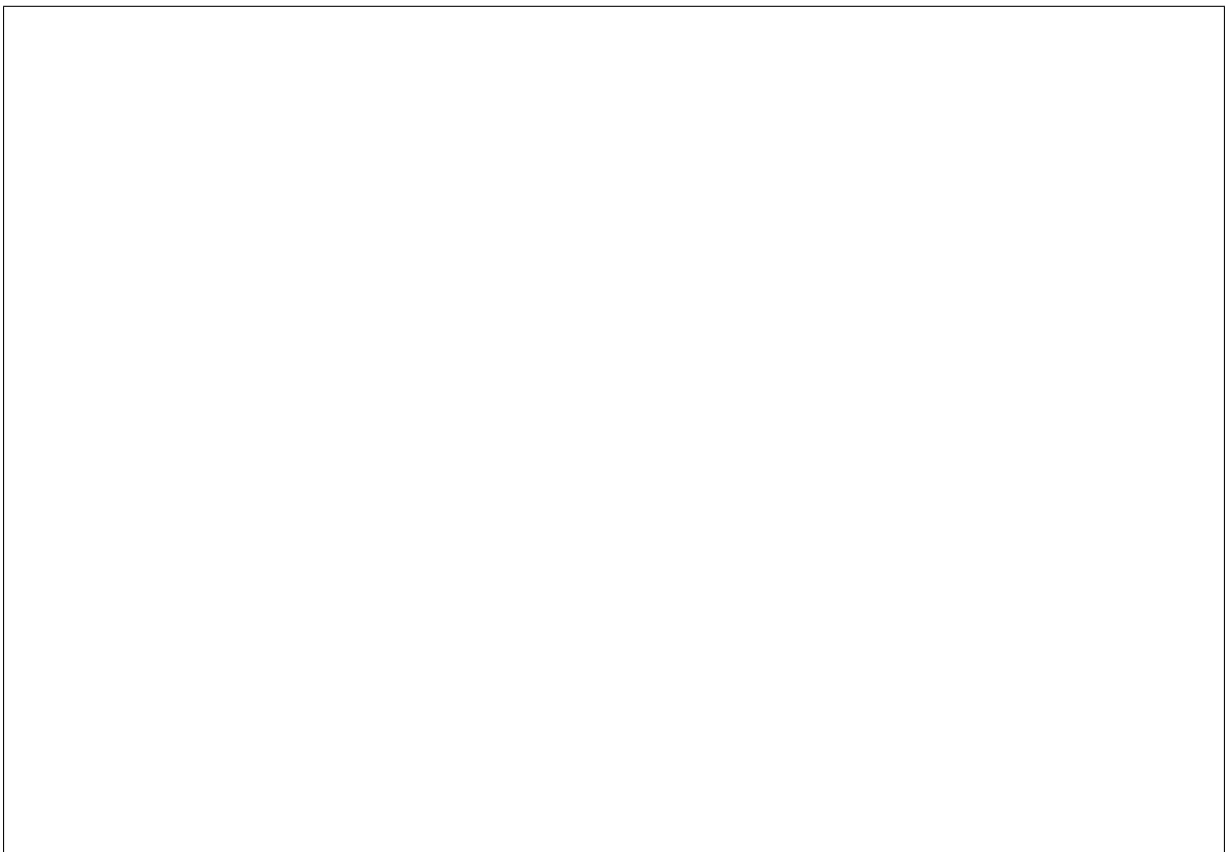


`chassis`. This step is part of management interface. Please refer to below table for their commonly used descriptions.



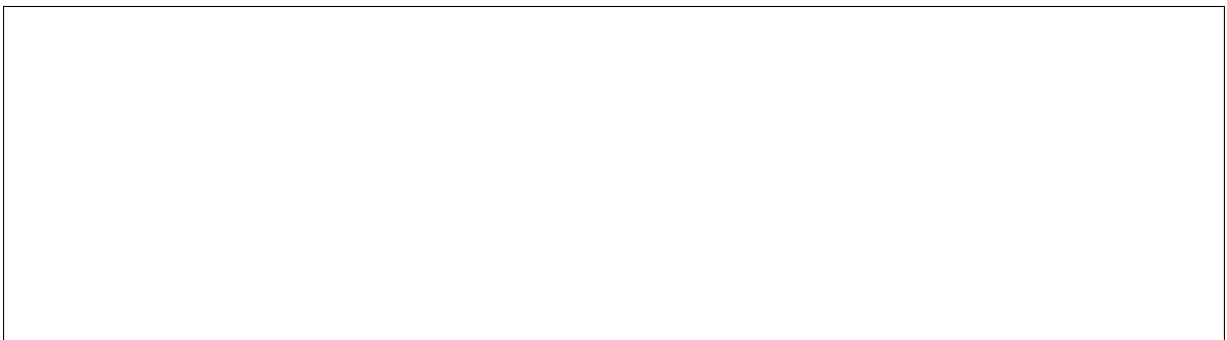
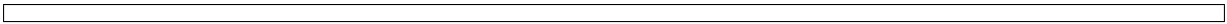
more information on usage.

### **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.

rameters 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.



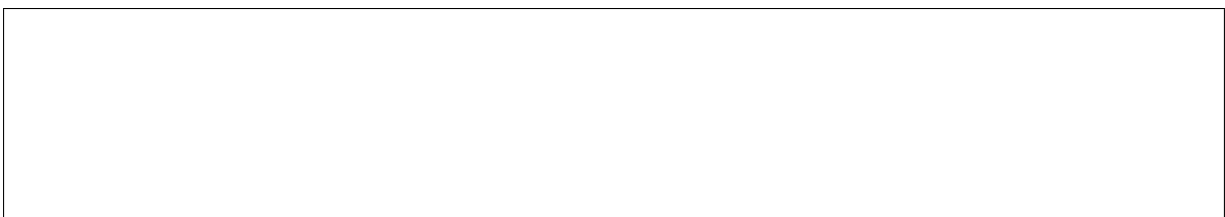






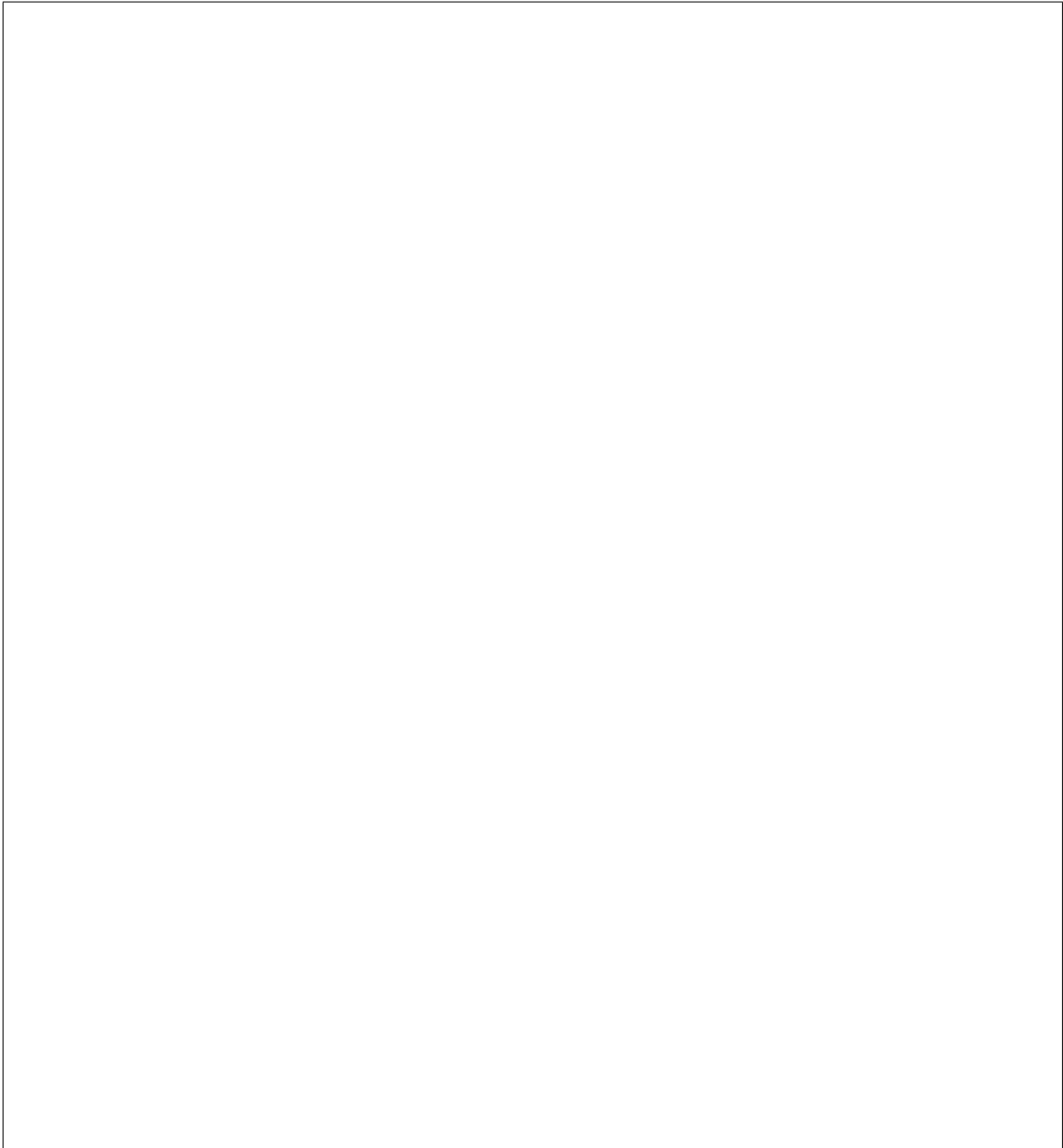
otherwise) NICs for Gen8 and Gen9 servers and ironic ports are created for all of them. Inspection logs a warning if the node under inspection is Gen8 or Gen9.

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**

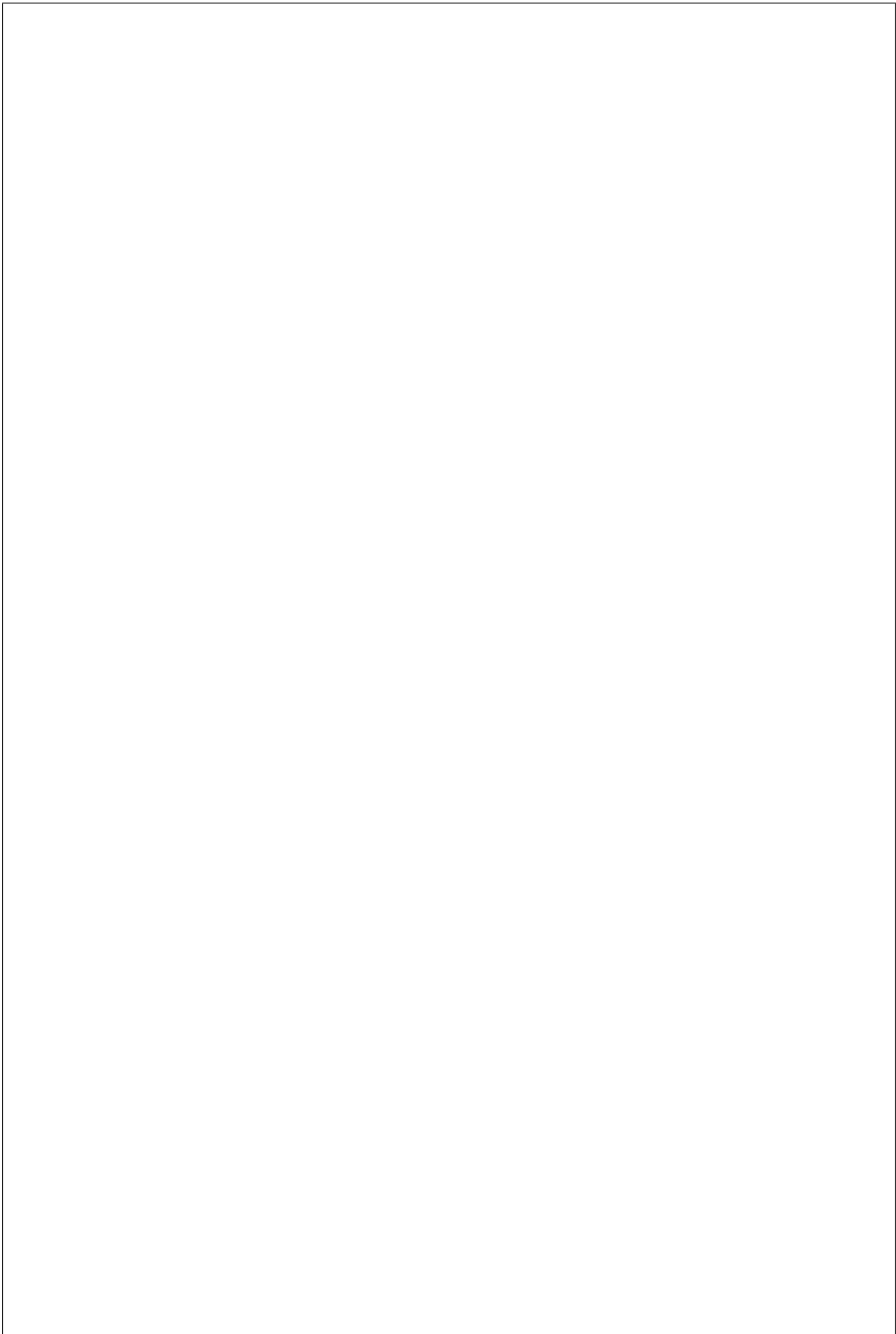
**Localboot with HTTP(S) based deploy**

**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)

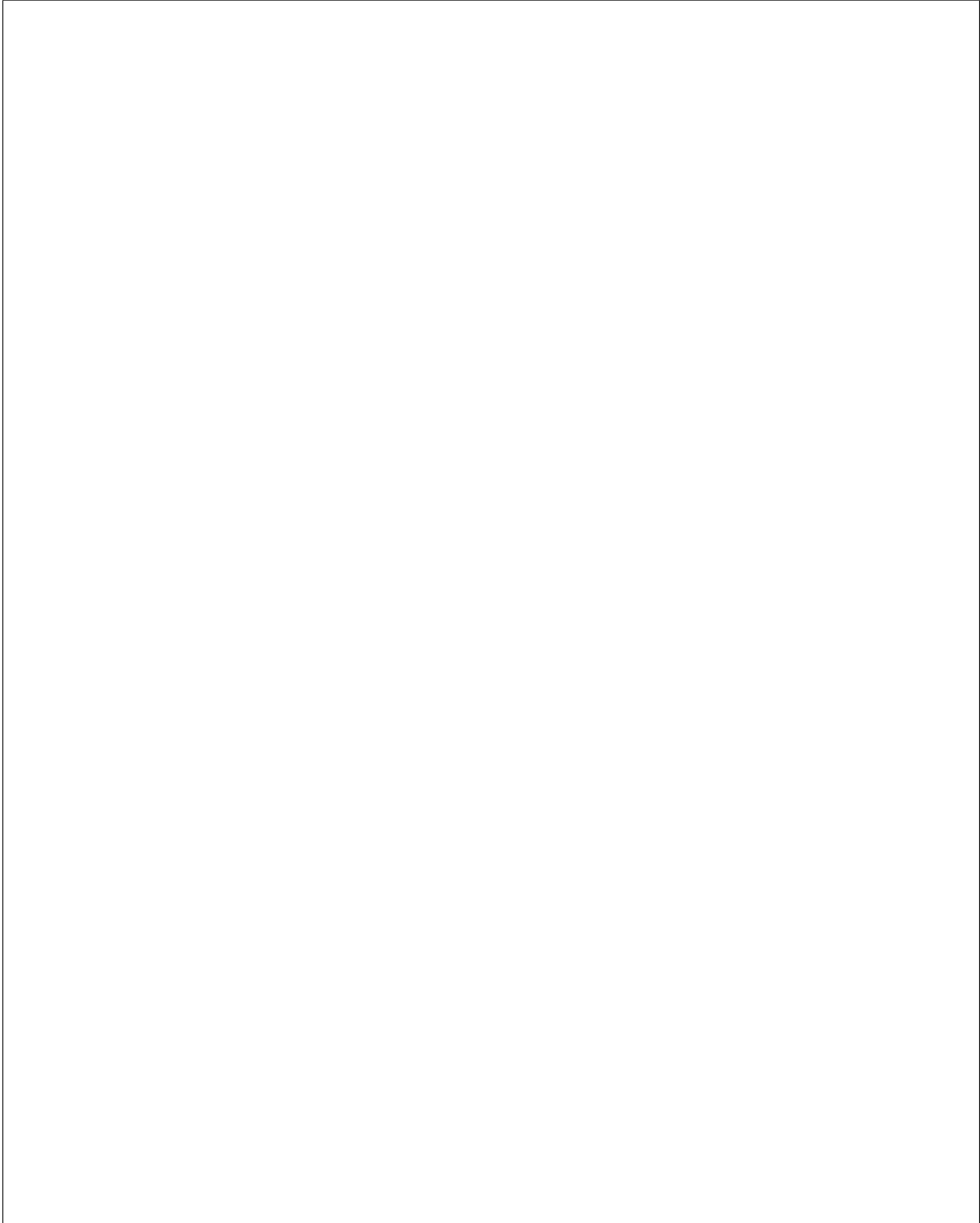
--	--

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.





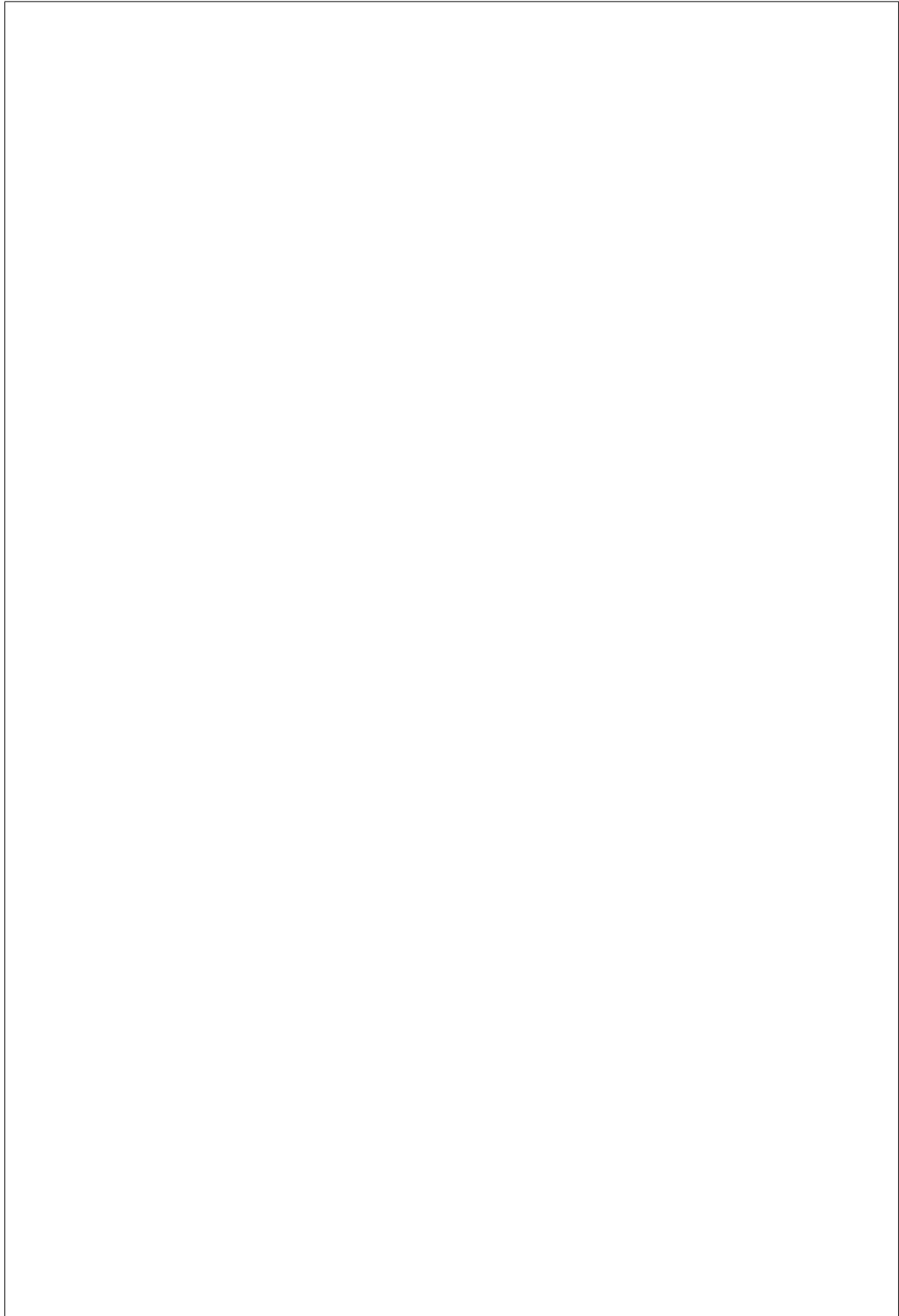
(continues on next page)

(continued from previous page)



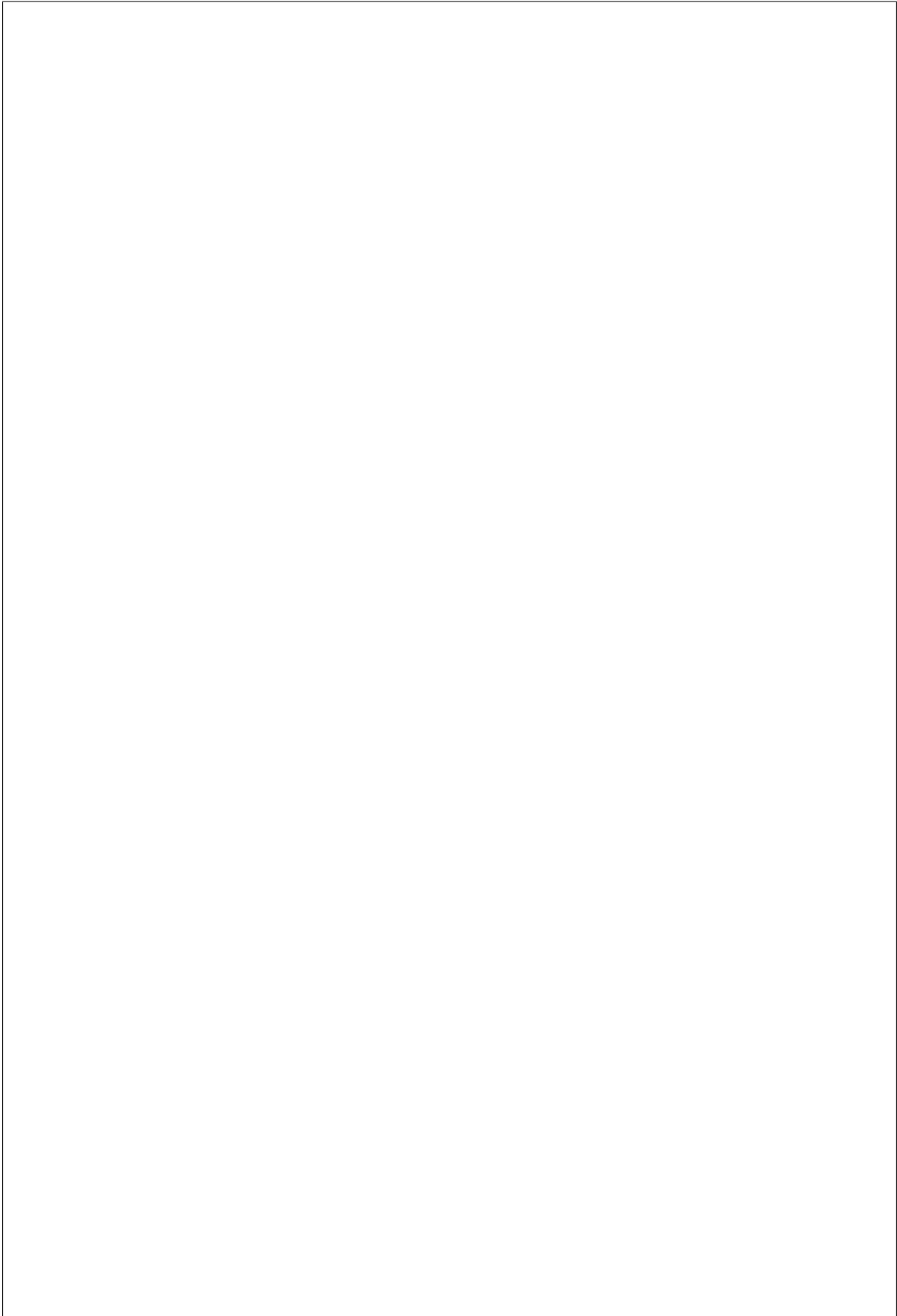
(continues on next page)

(continued from previous page)



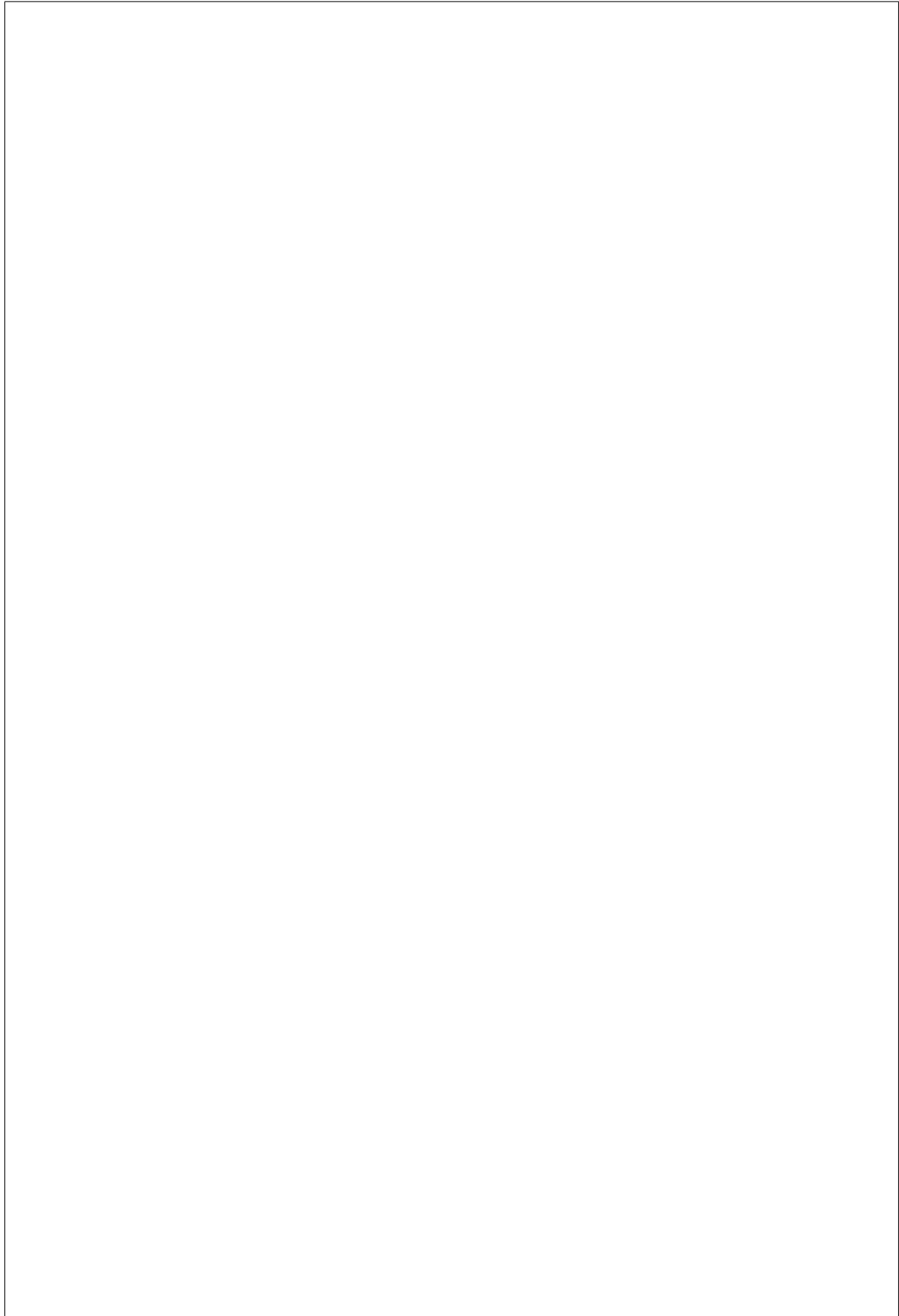
(continues on next page)

(continued from previous page)



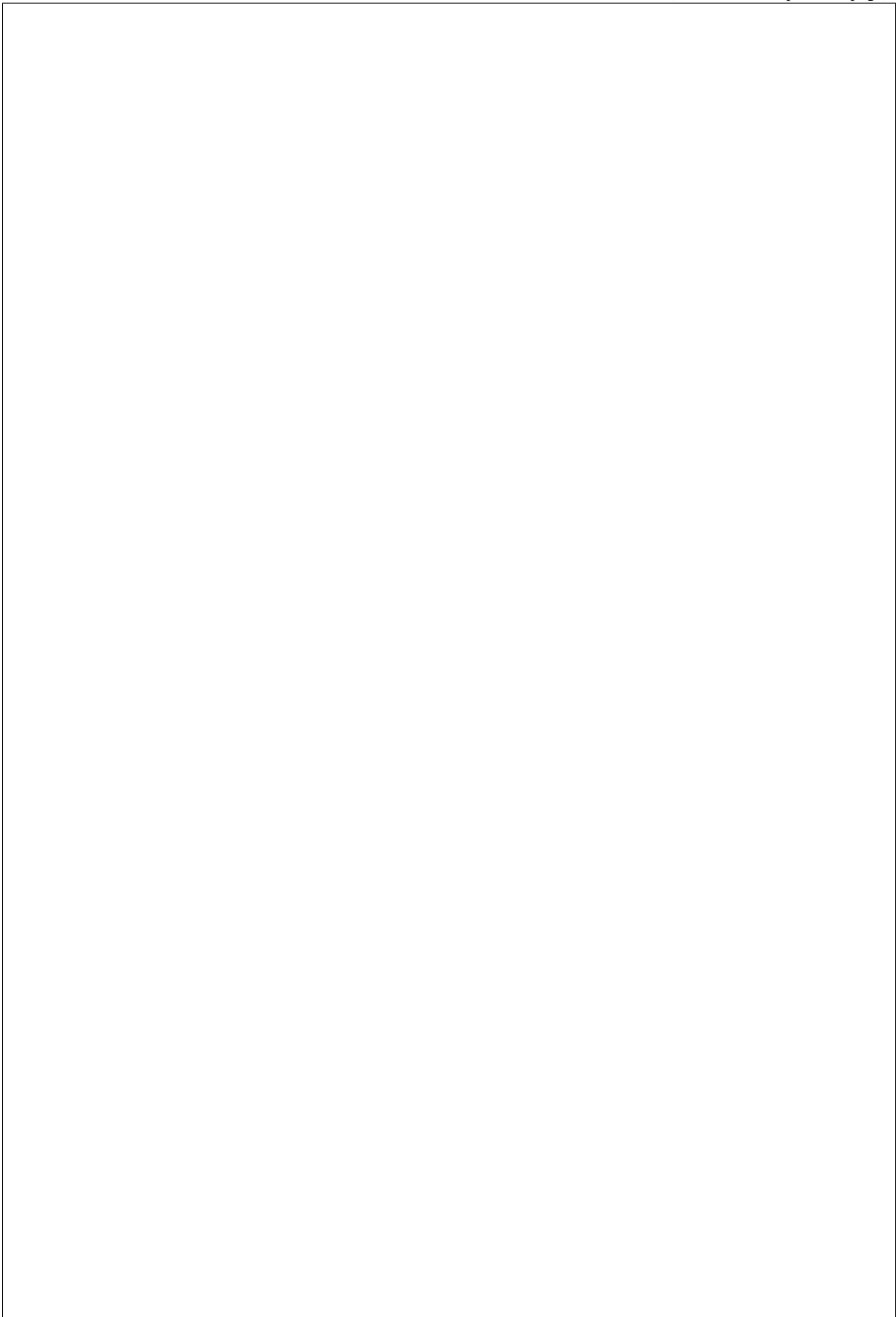
(continues on next page)

(continued from previous page)



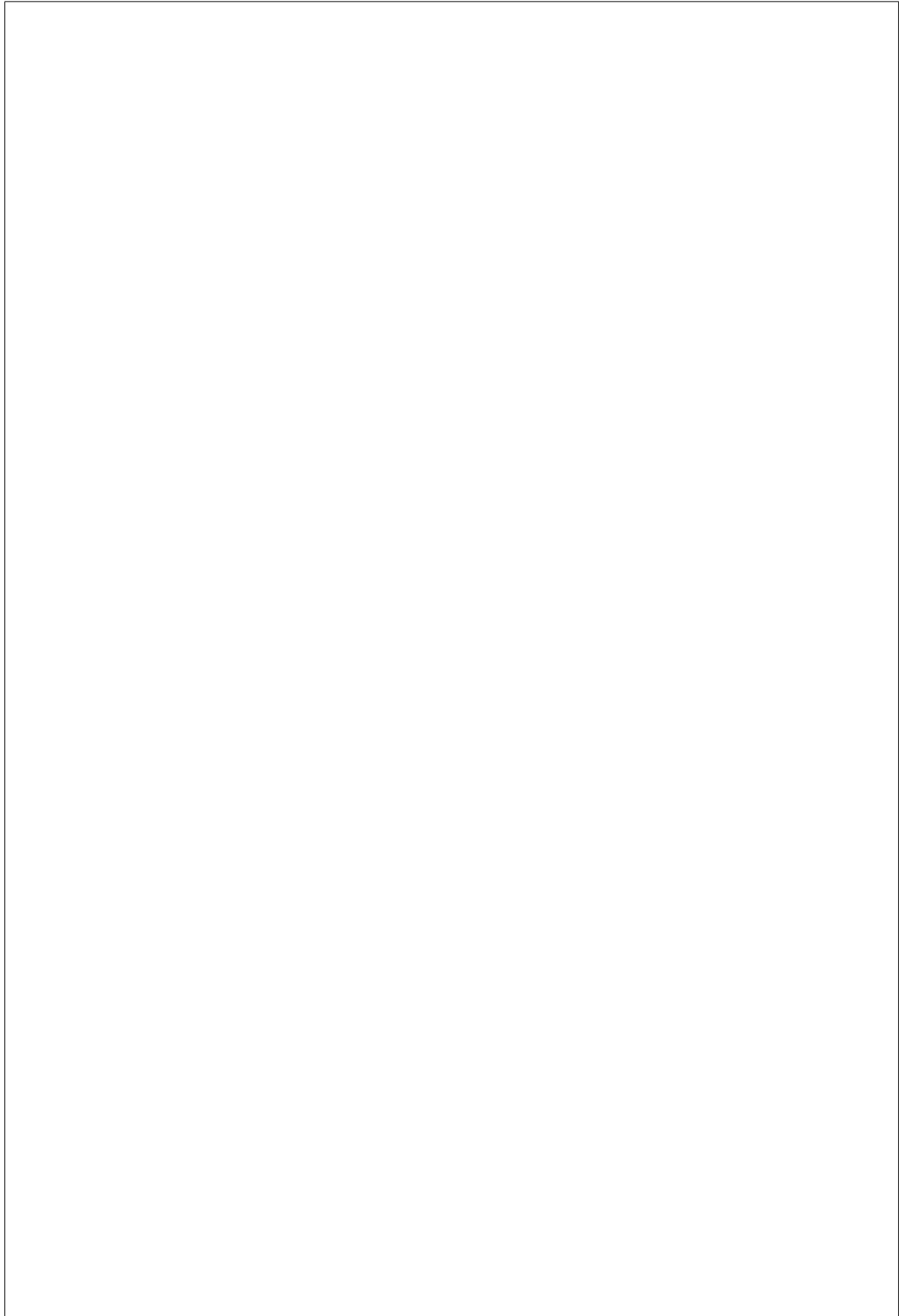
(continues on next page)

(continued from previous page)



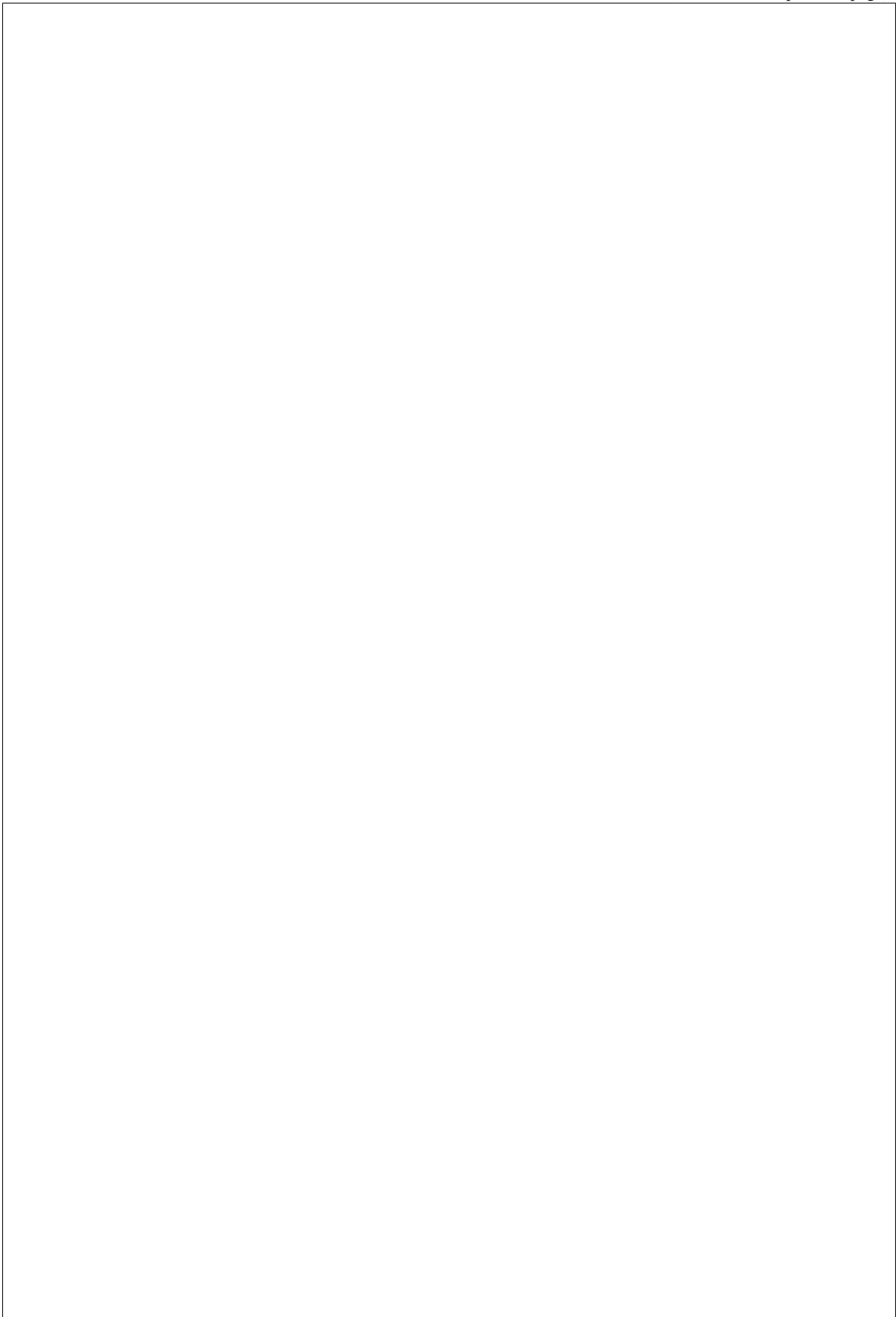
(continues on next page)

(continued from previous page)



(continues on next page)

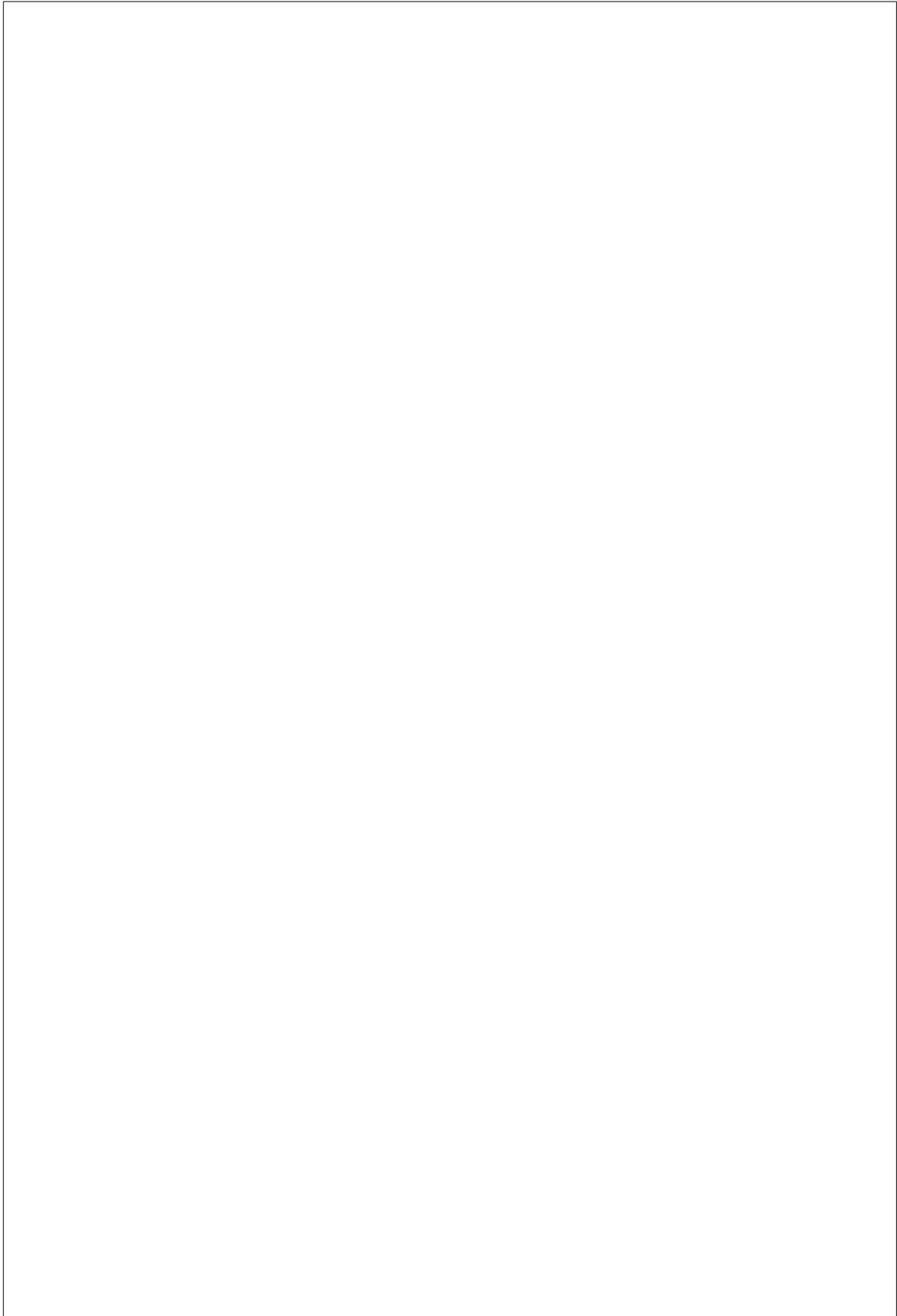
(continued from previous page)



(continues on next page)



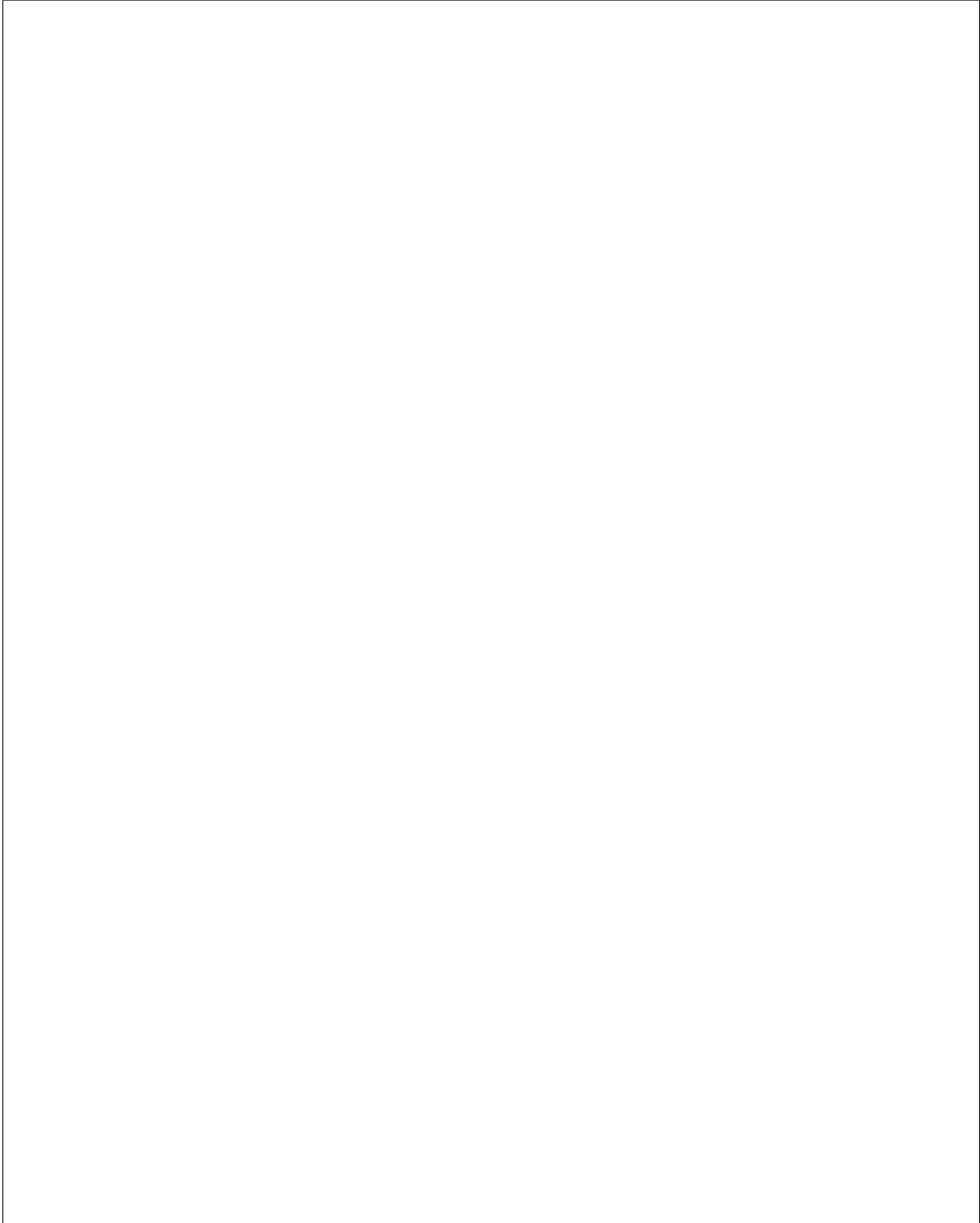
(continued from previous page)



(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.

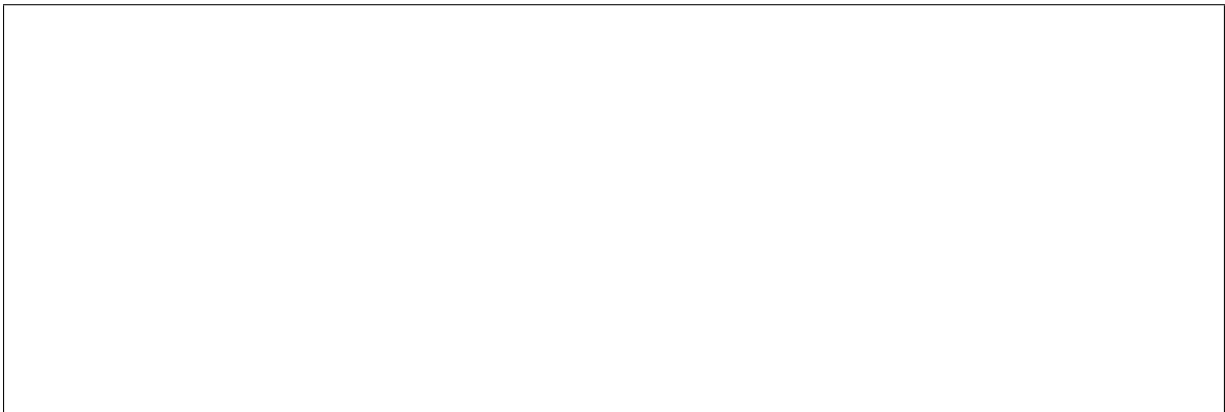
**Note:** `flash_firmware_sum` deploy step requires the agent ramdisk with Proliant Hardware Manager from the `proliantutils` version 2.9.5 or higher. See *DIB support for Proliant Hardware Manager* to create the agent ramdisk with Proliant Hardware Manager.

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:

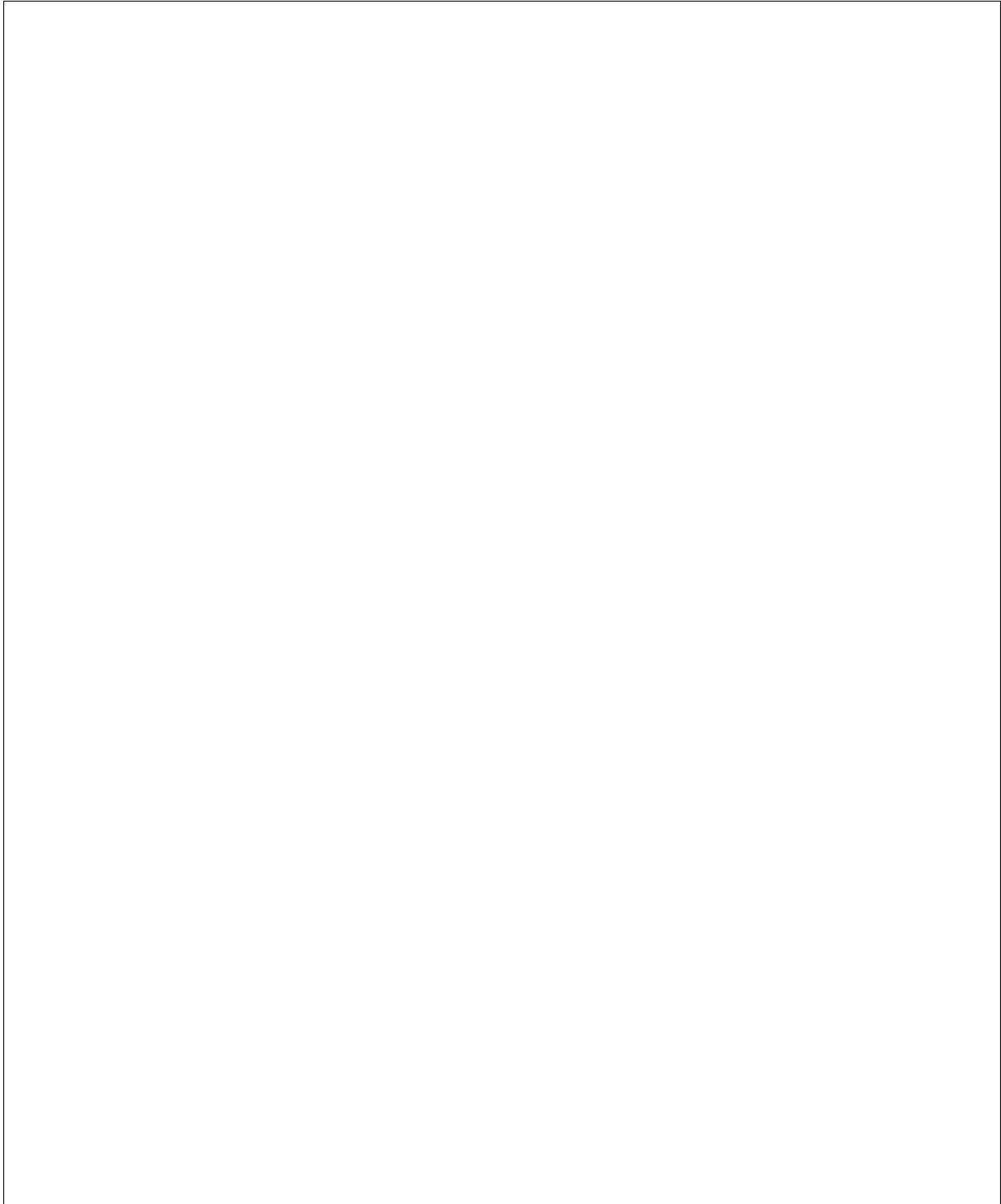


(continues on next page)

(continued from previous page)





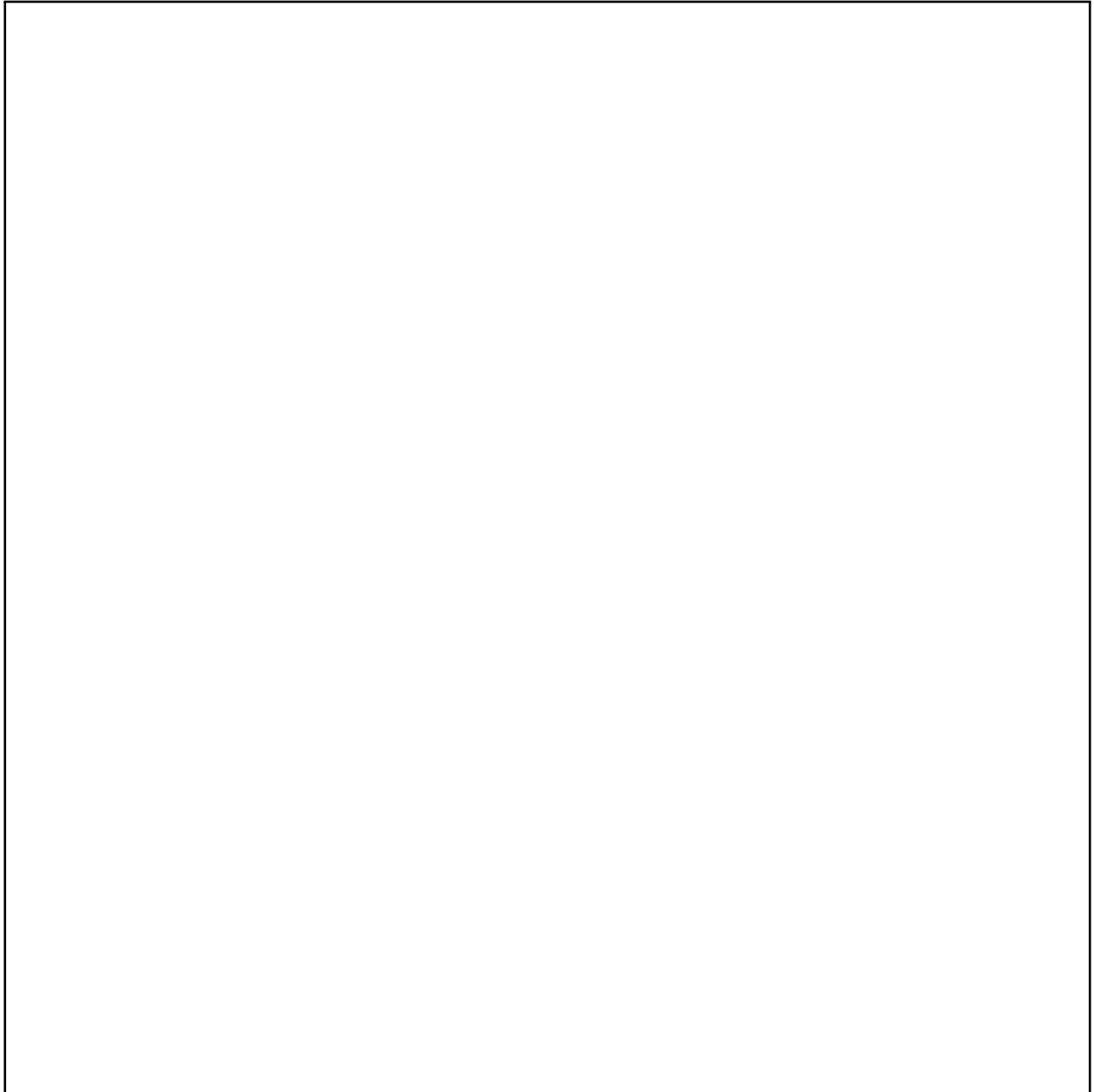


(continues on next page)

(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:

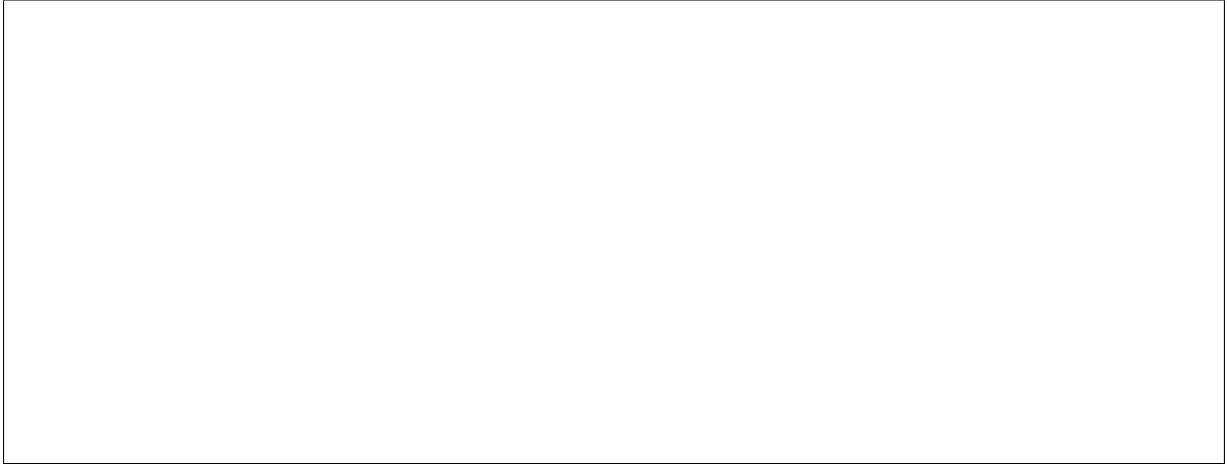


## **DIB support for Proliant Hardware Manager**

---

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





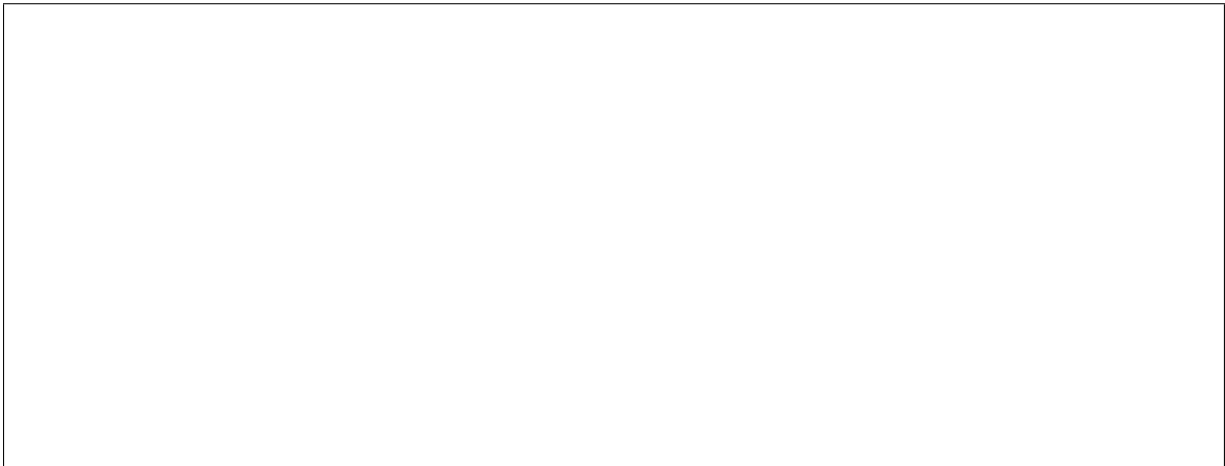
## **Disk Erase Support**



ported by SSA.

ssaccli 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**



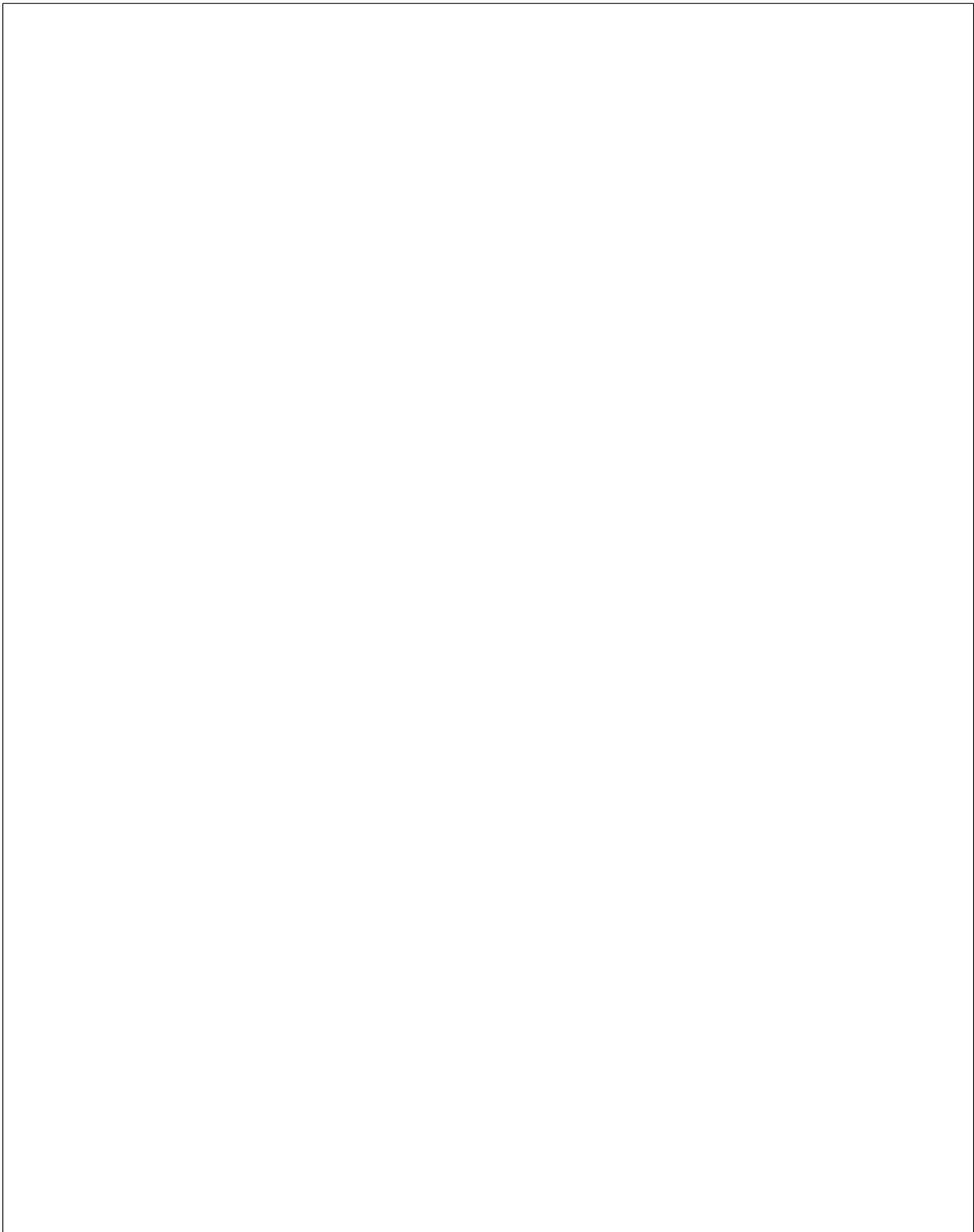
(continues on next page)

(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.

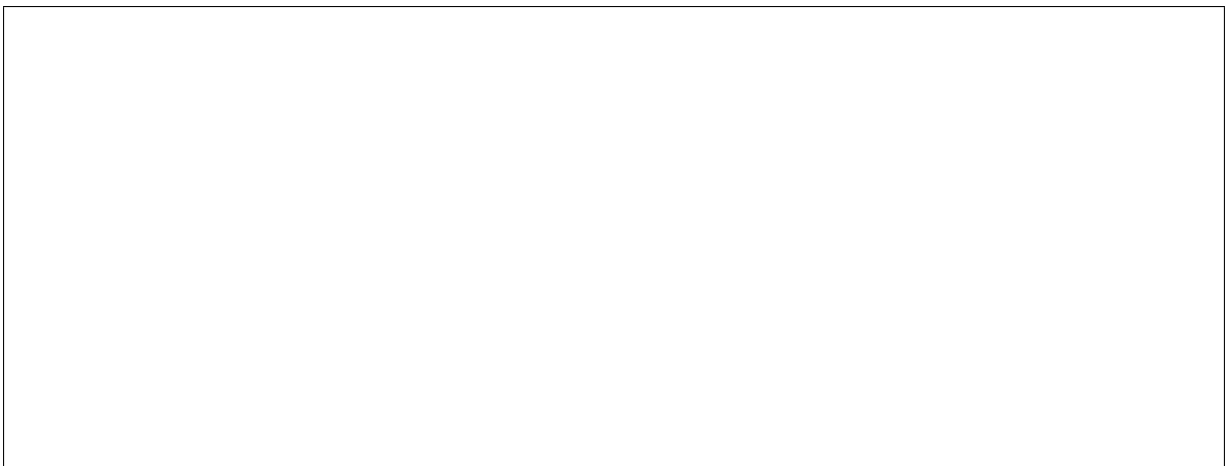
**Note:** Server POST state is used to track the power status of HPE ProLiant Gen9 servers and beyond.

## **Out of Band RAID Support**

*RAID Configuration* for more information.







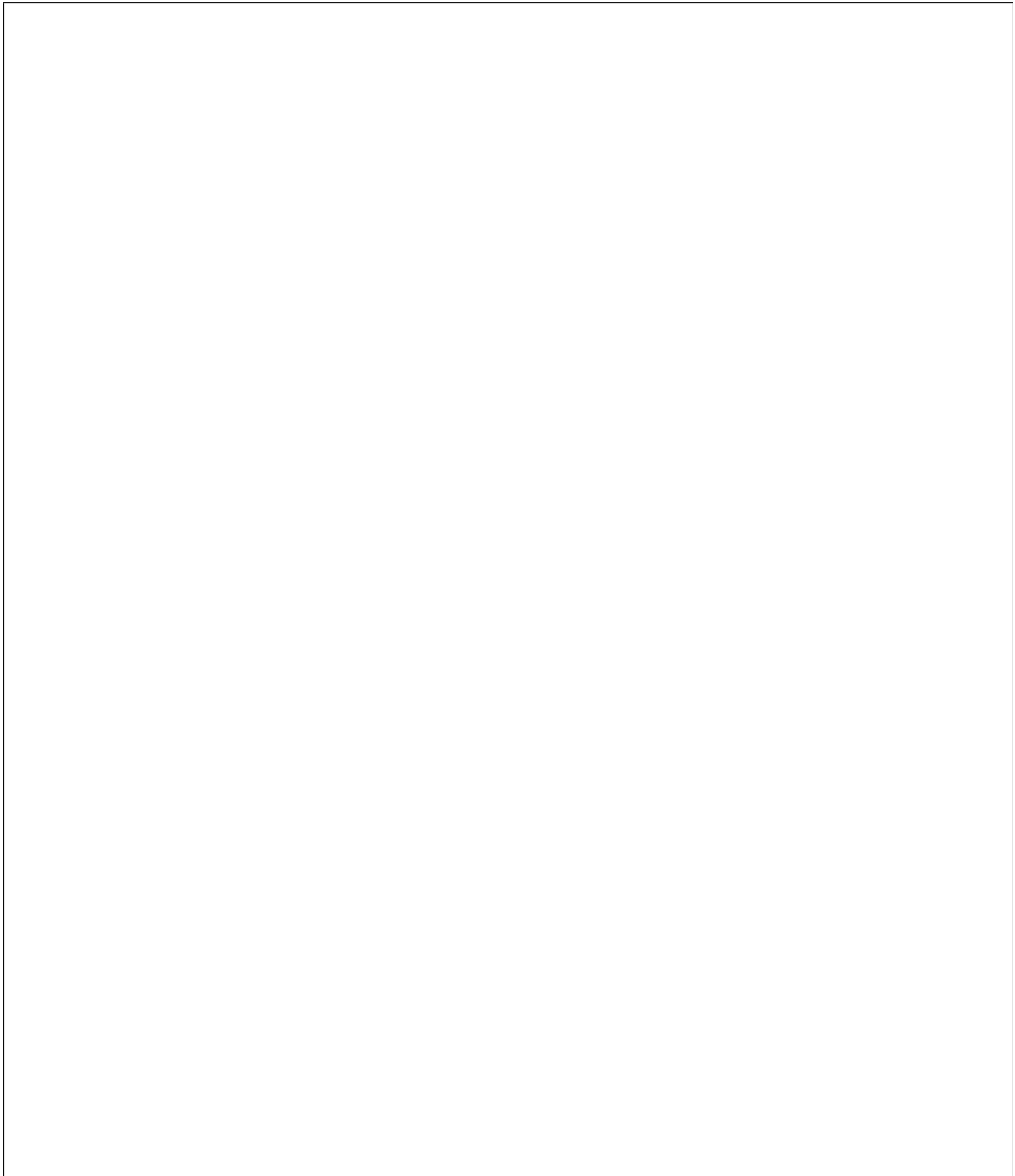
(continues on next page)

(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.





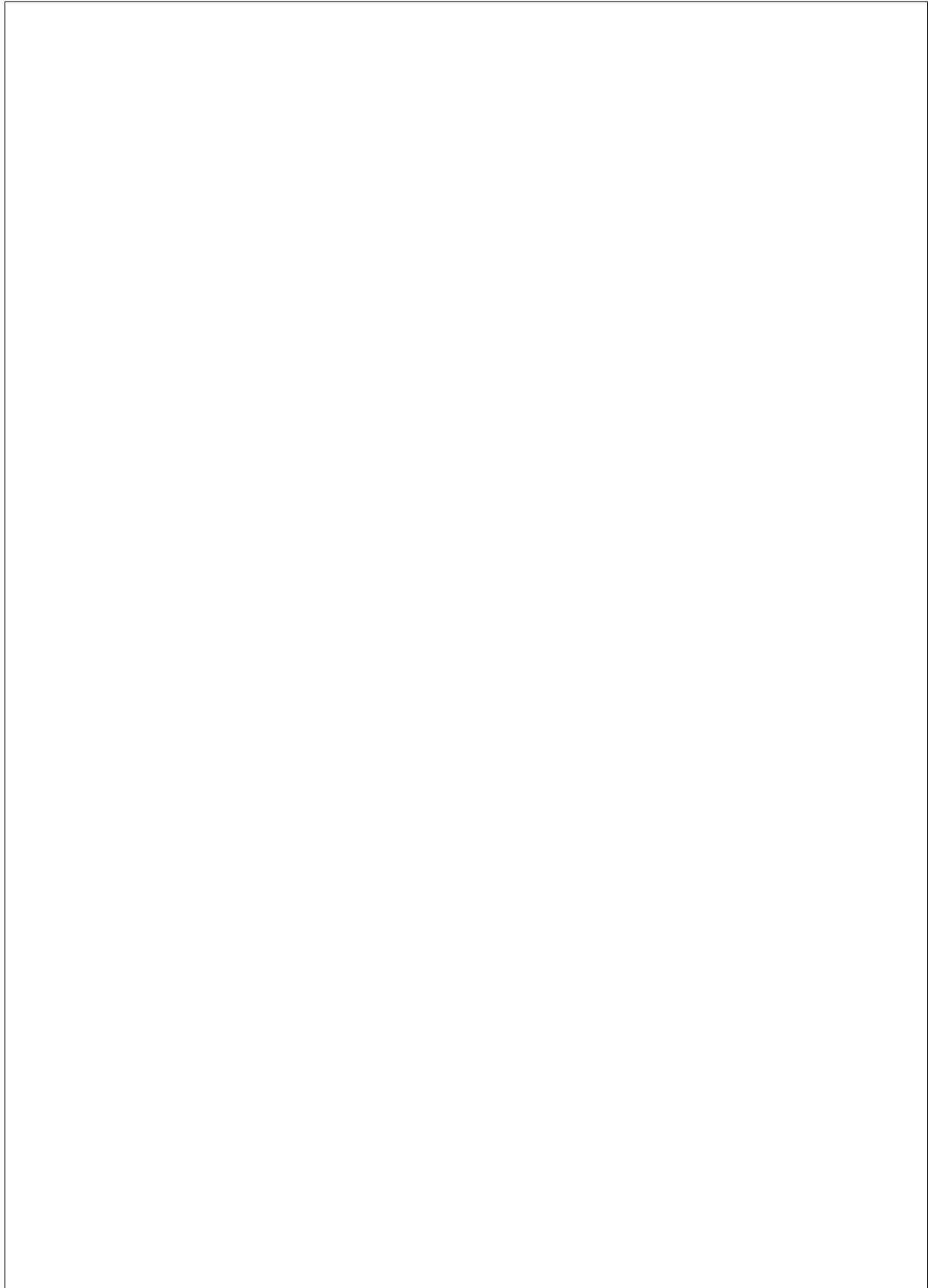
(continues on next page)

(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.

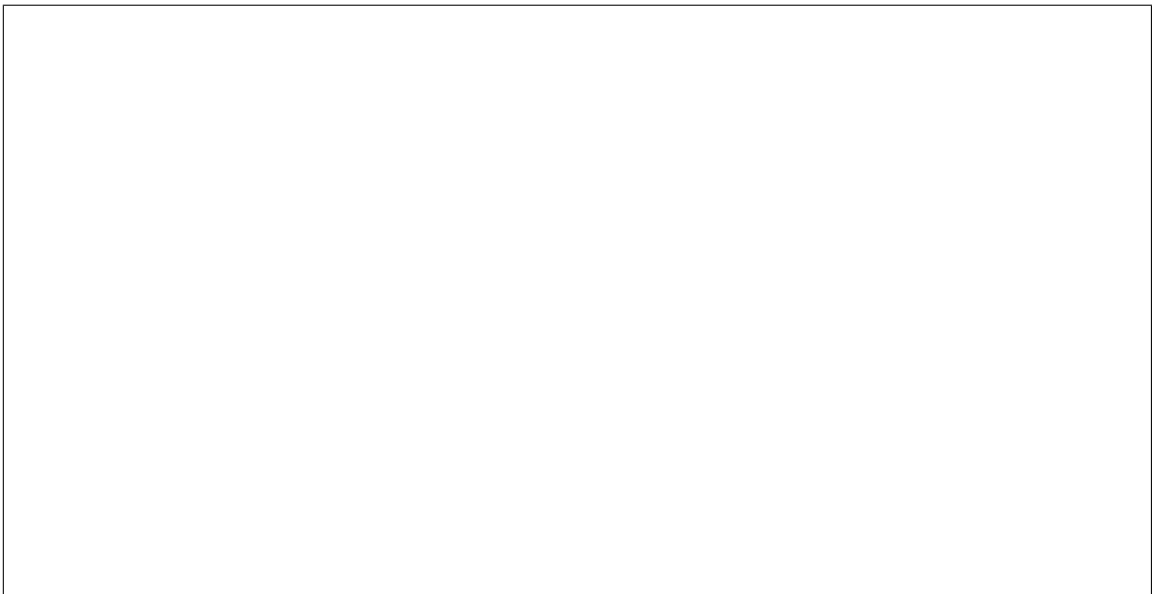
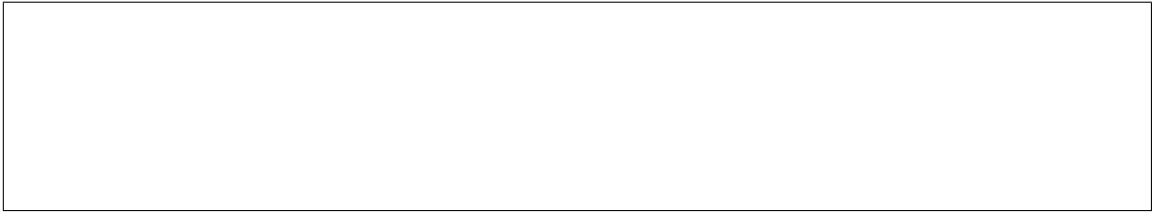
## **Glossary**

## **Enabling the IntelIPMI hardware type**



(continues on next page)

(continued from previous page)



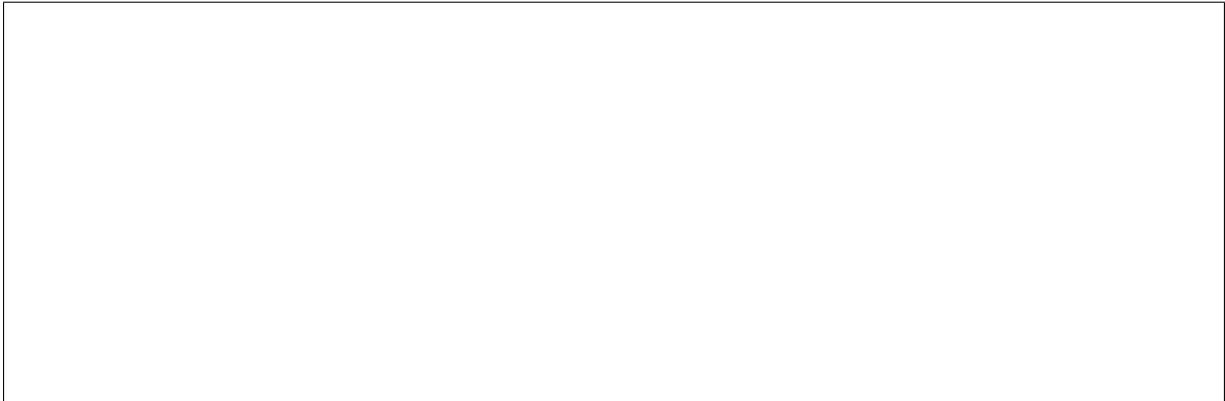
### **Registering a node with the IntelIPMI driver**





(continues on next page)

(continued from previous page)



## **Features of the `intel-ipmi` hardware type**

### **Intel SST-PP**

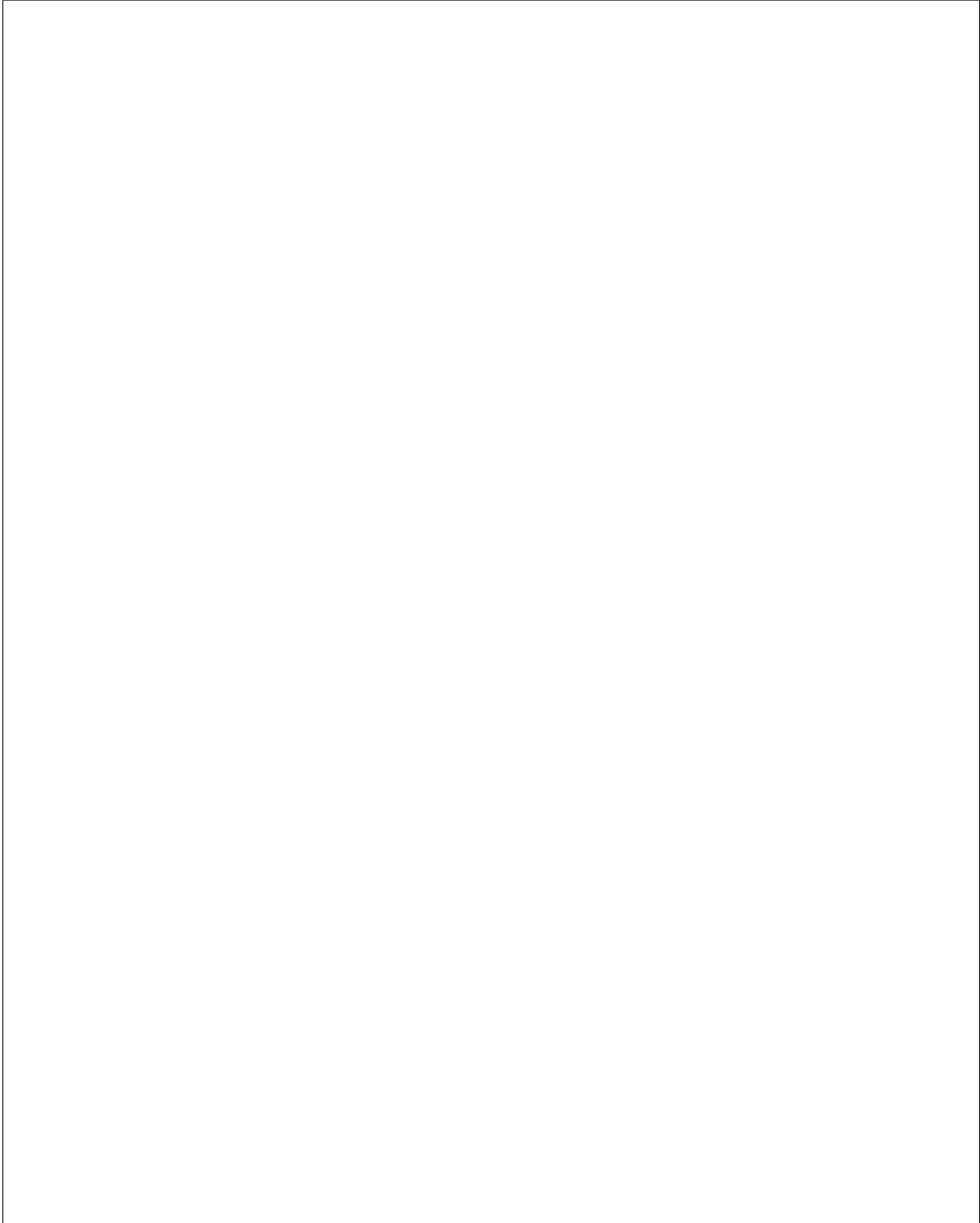
spectively. The input value must be a string.





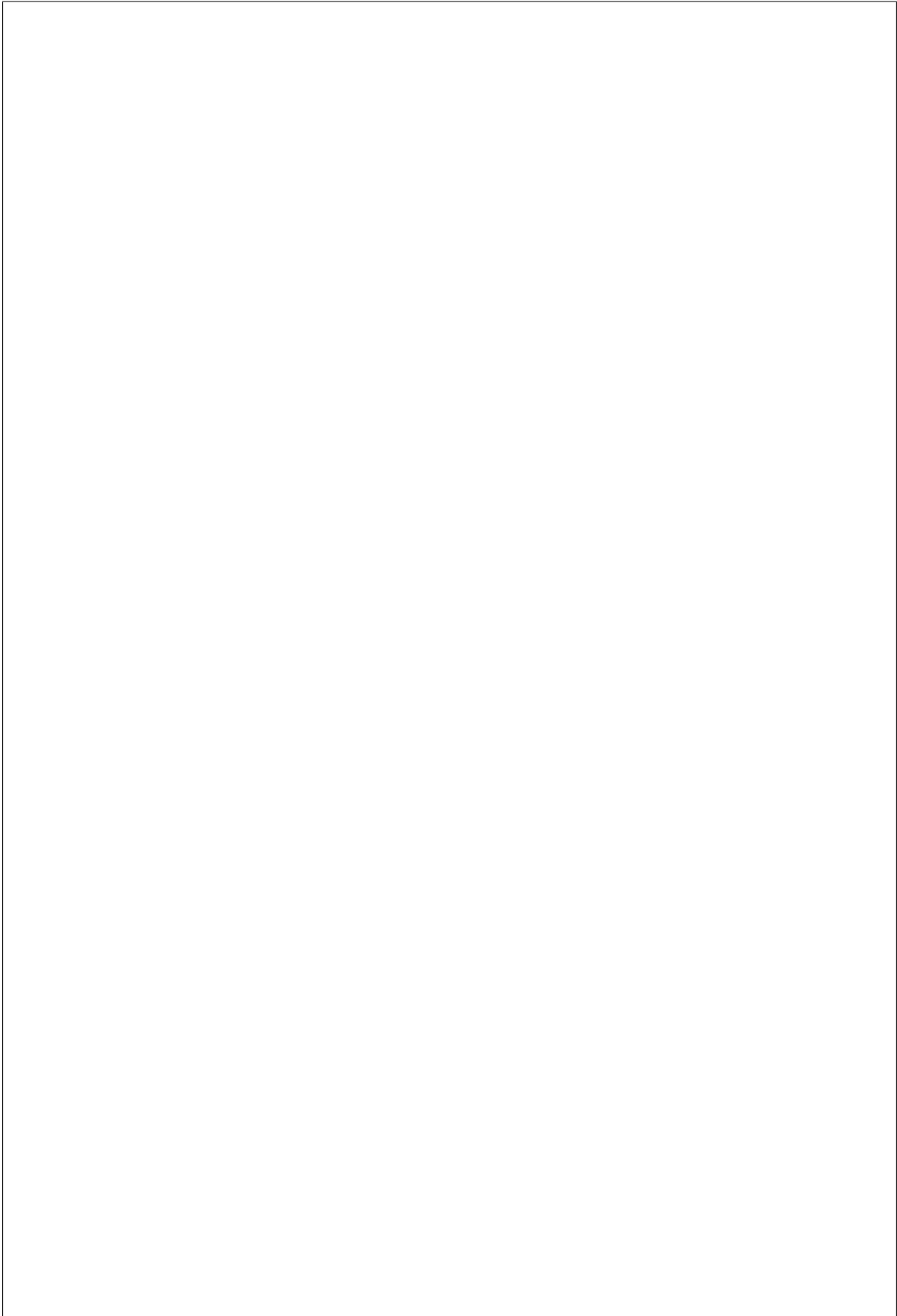






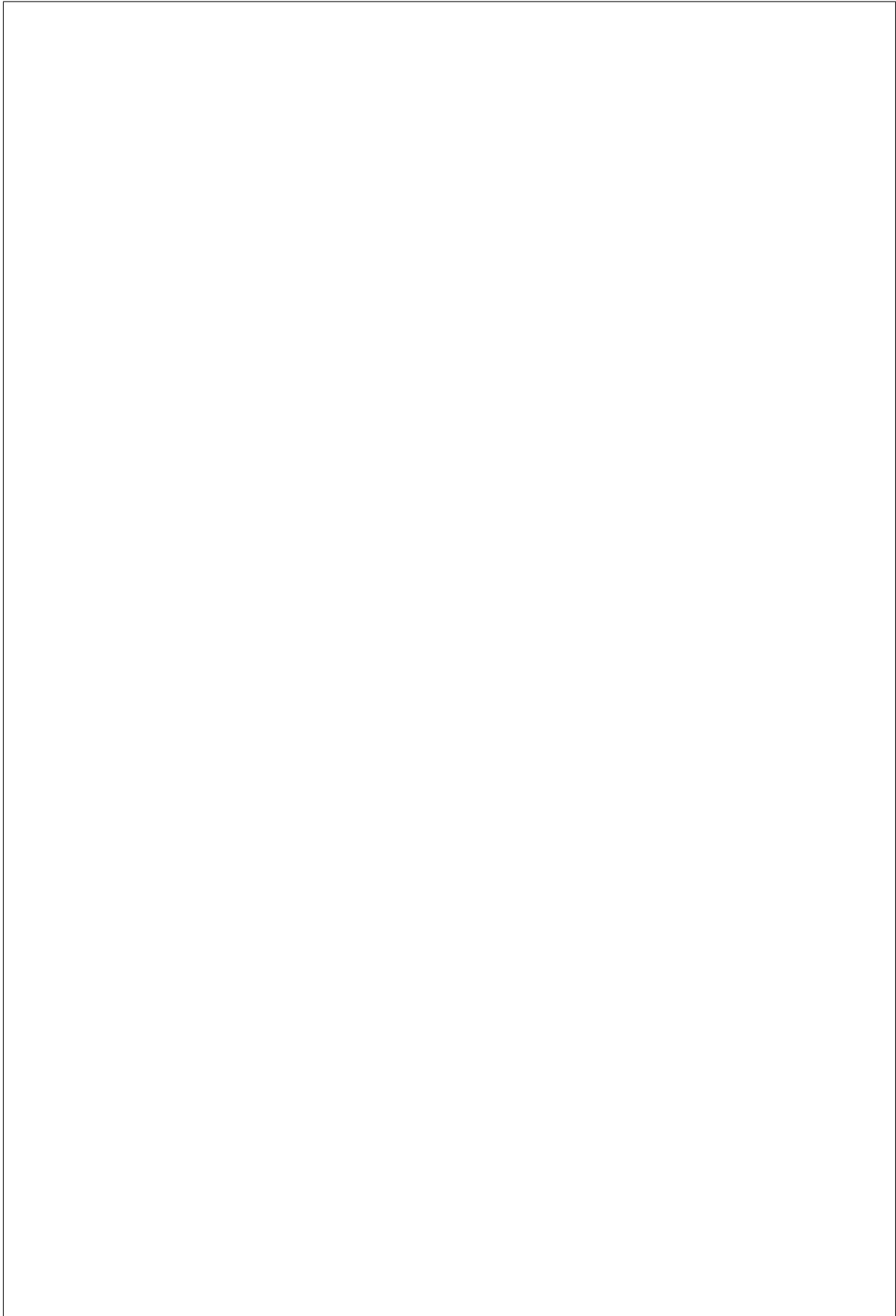
(continues on next page)

(continued from previous page)



(continues on next page)

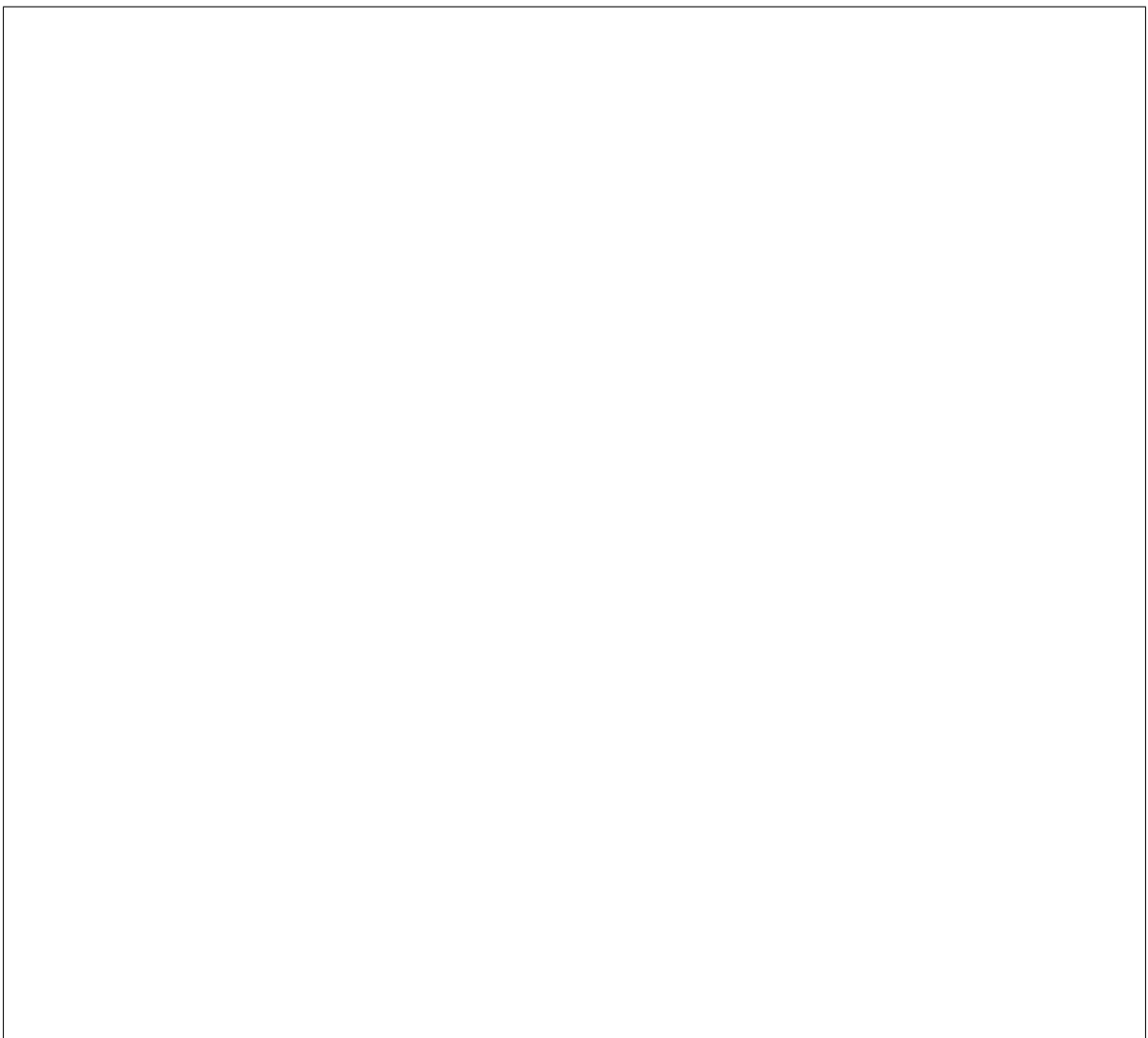
(continued from previous page)



(continues on next page)

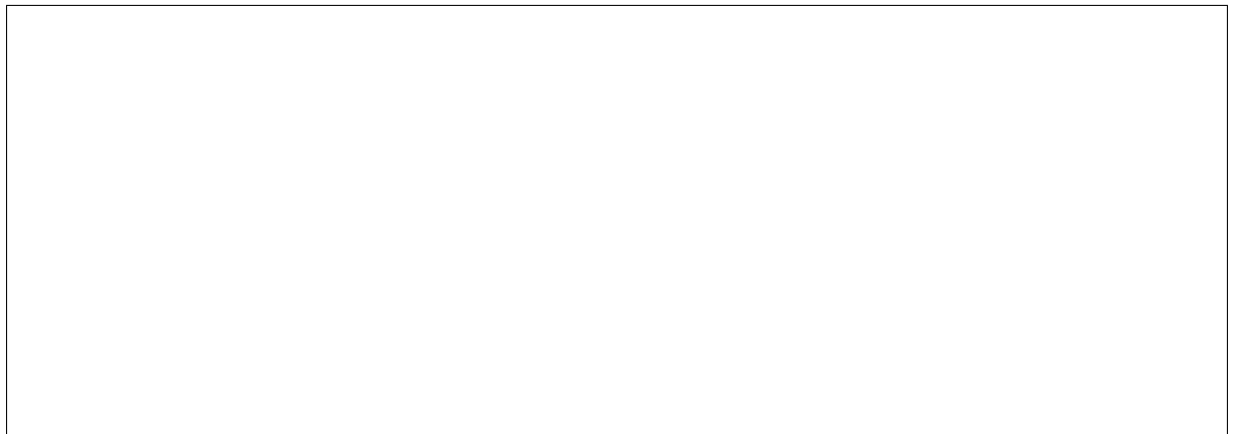
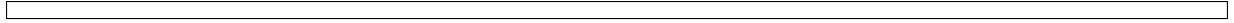
(continued from previous page)

---



(continues on next page)

(continued from previous page)



**IPMI driver**

**Overview**

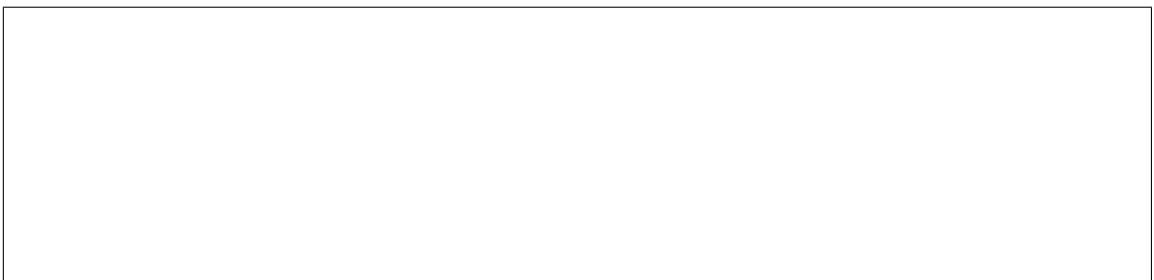
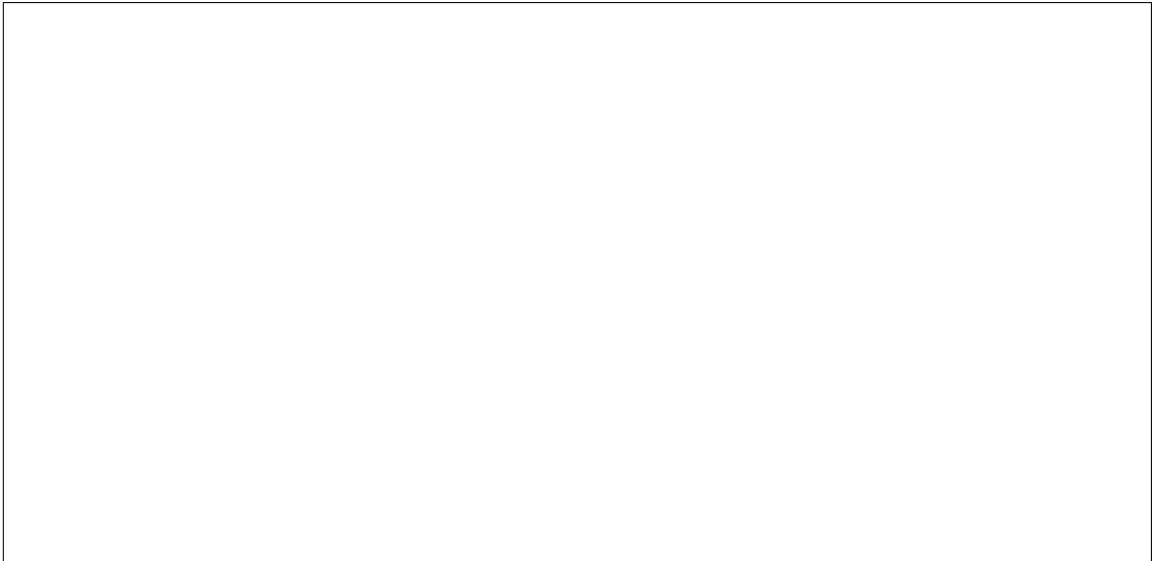


ling IPMI-enabled devices.

## **Glossary**



## **Enabling the IPMI hardware type**



(continues on next page)

(continued from previous page)

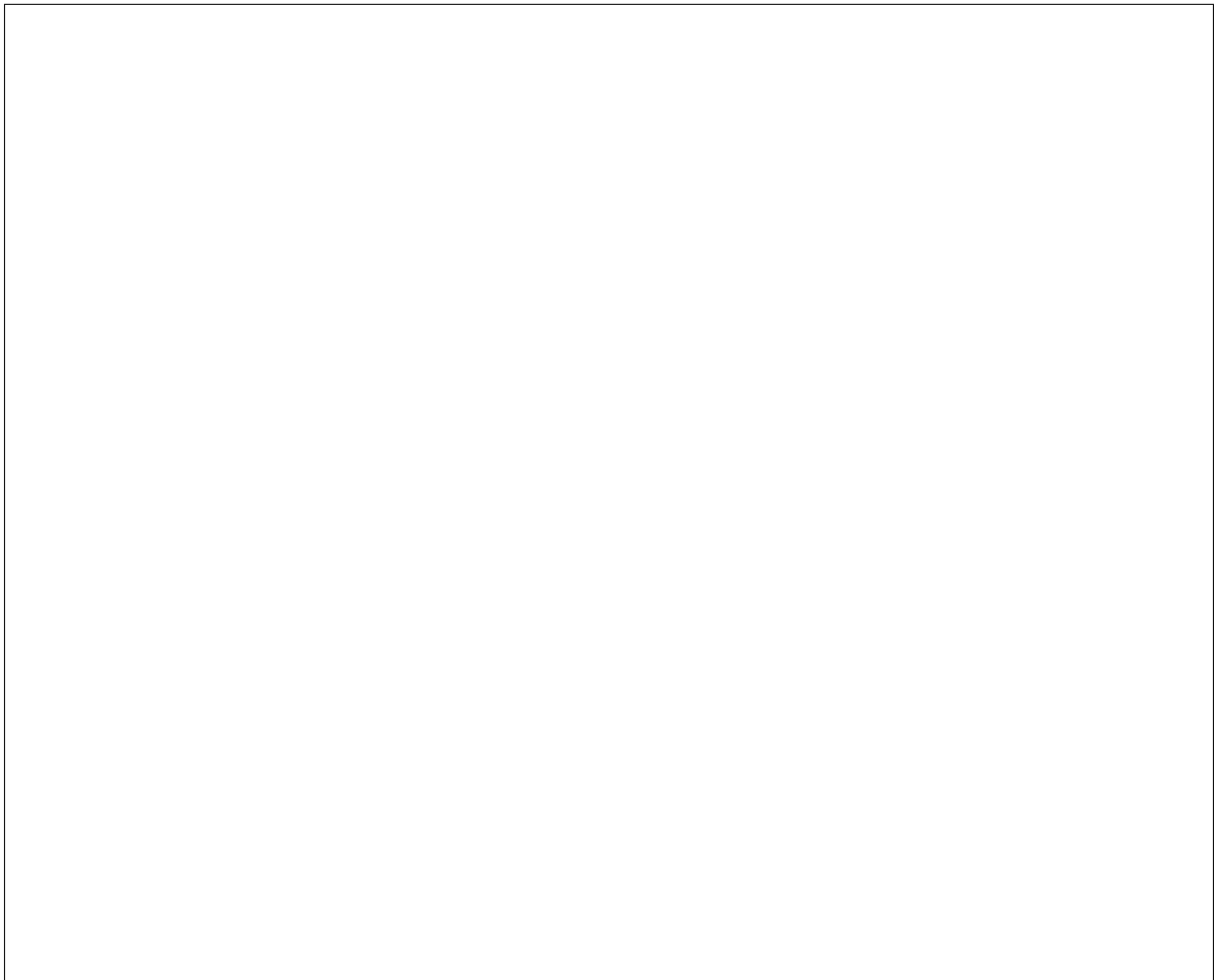


## **Registering a node with the IPMI driver**



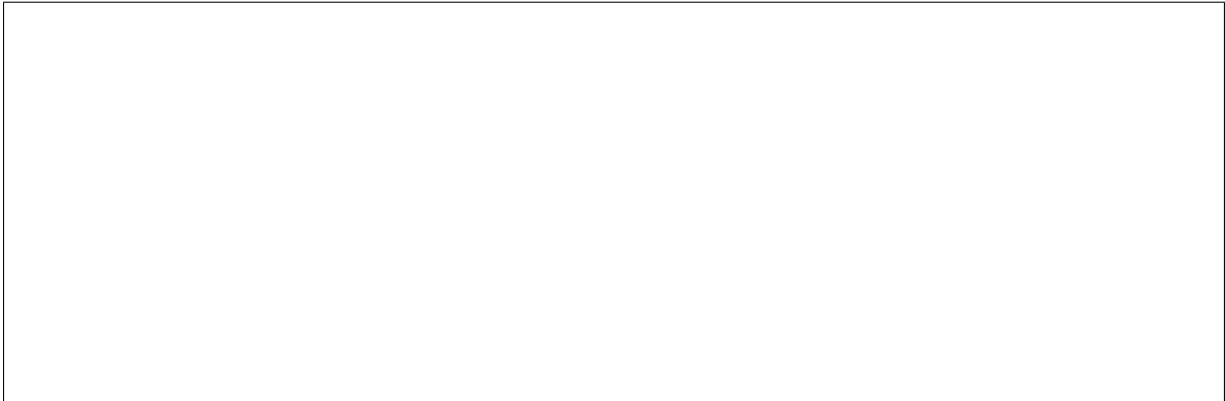
**Note:** It is highly recommend that you setup a username and password for your BMC.





(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.



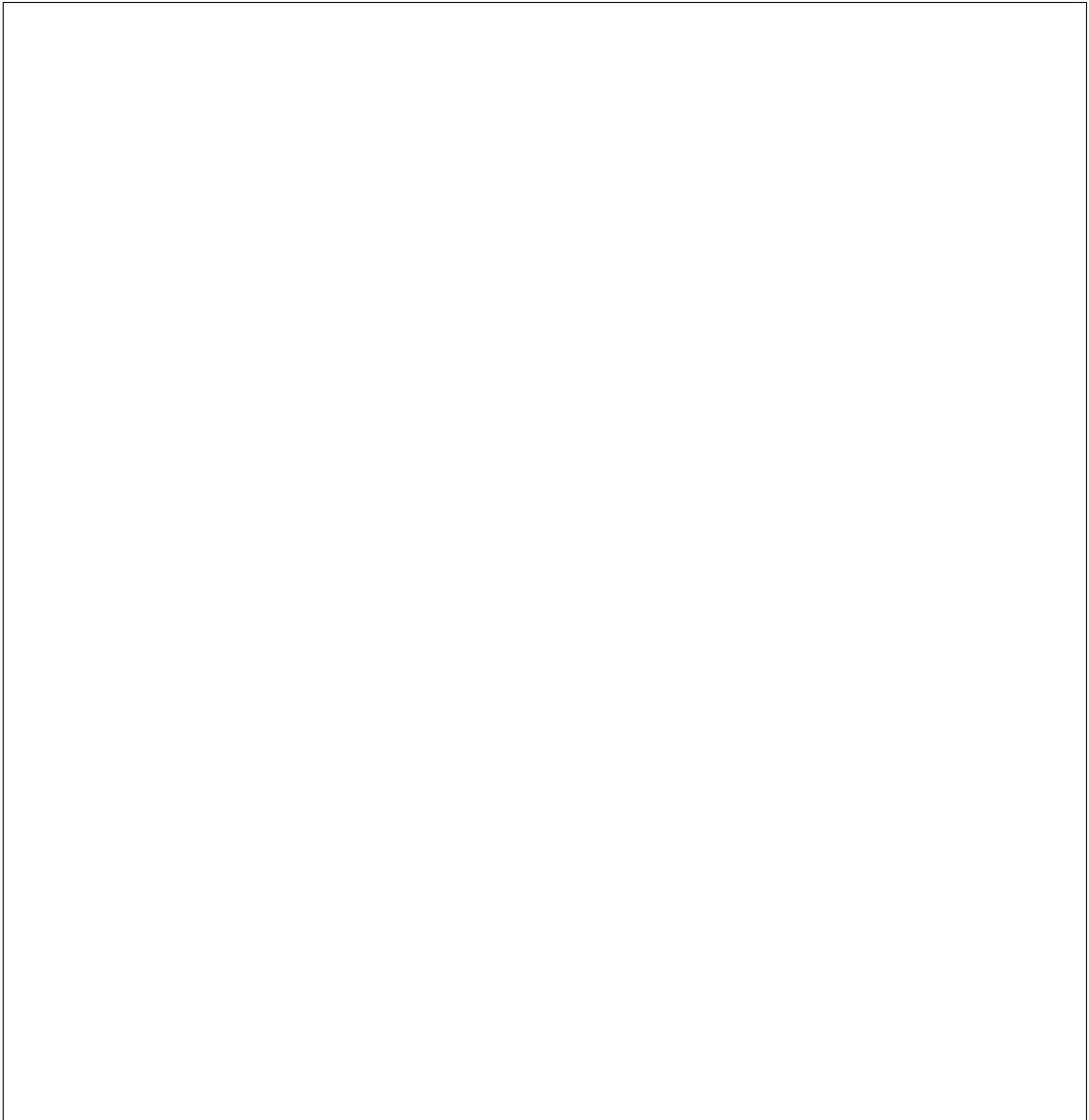






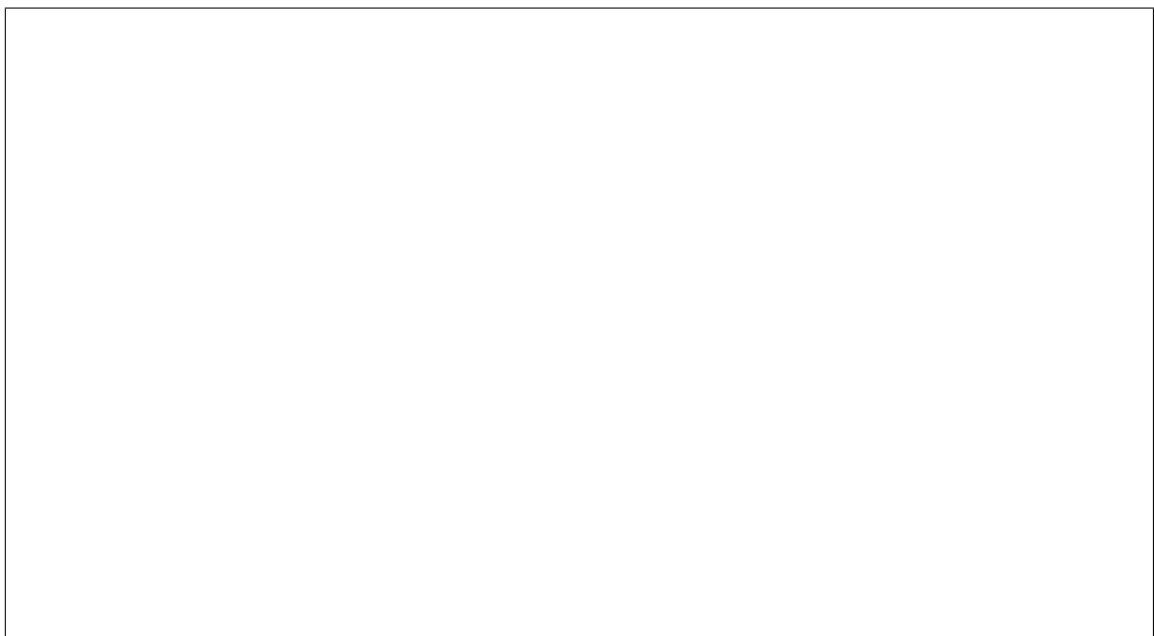






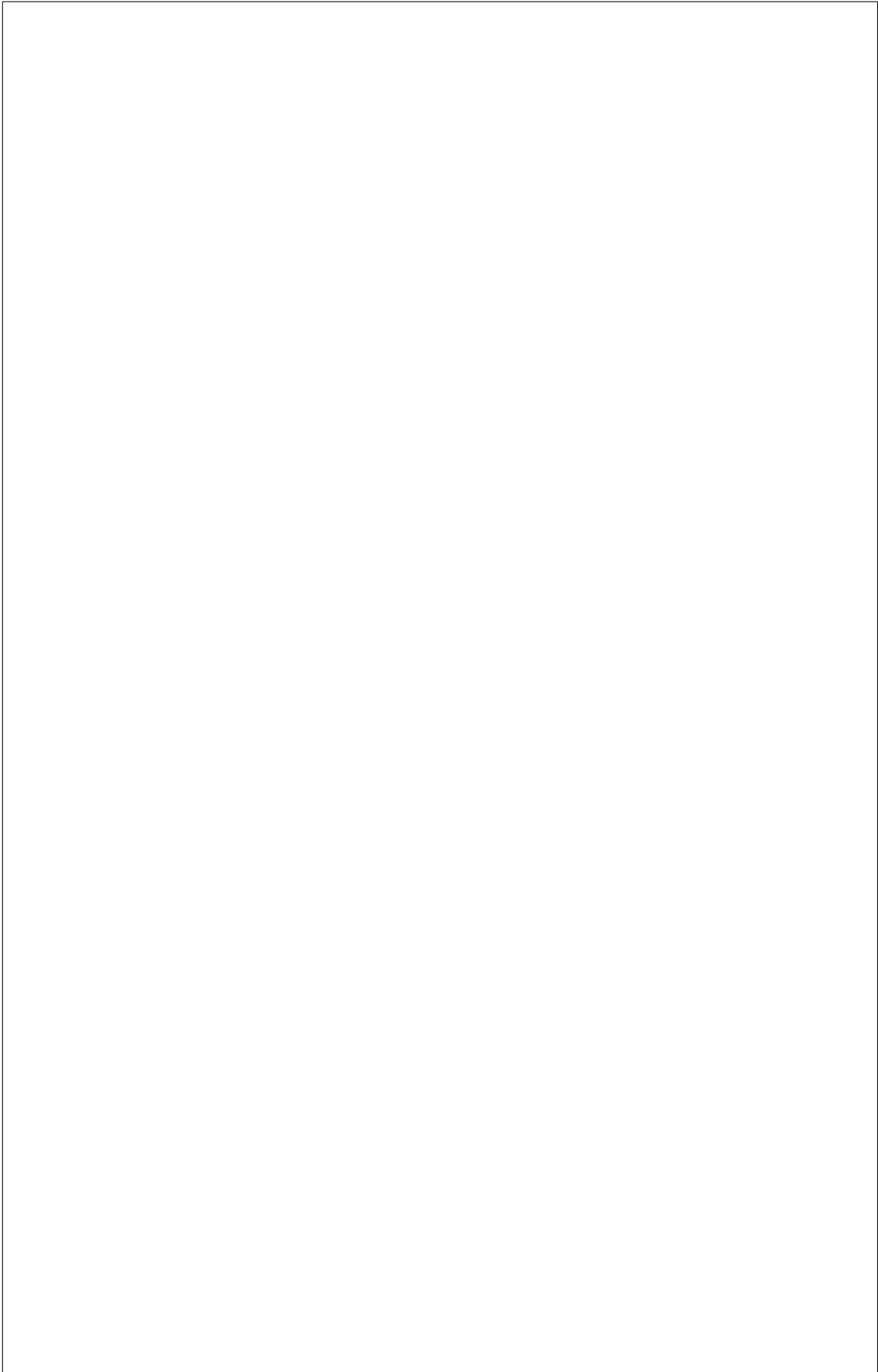
(continues on next page)

(continued from previous page)



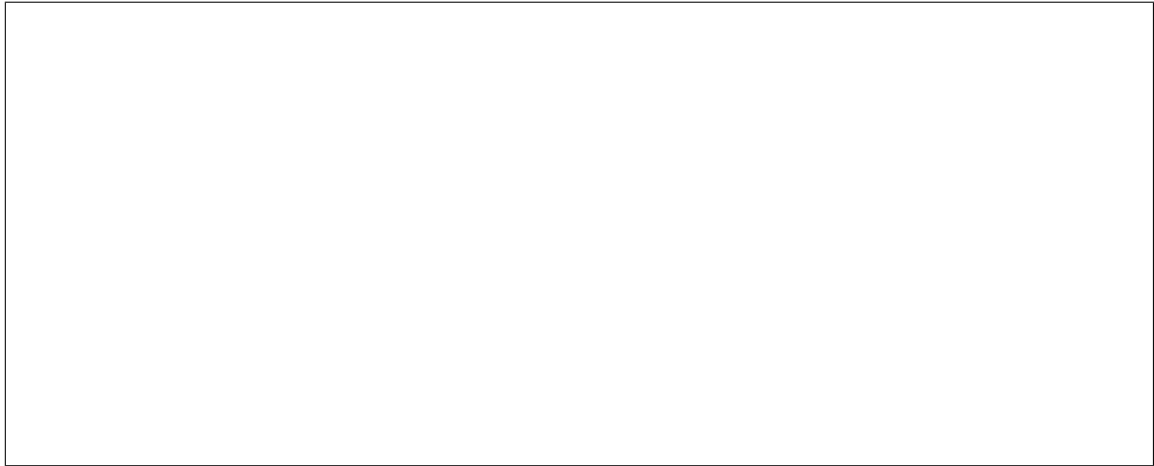
(continues on next page)

(continued from previous page)



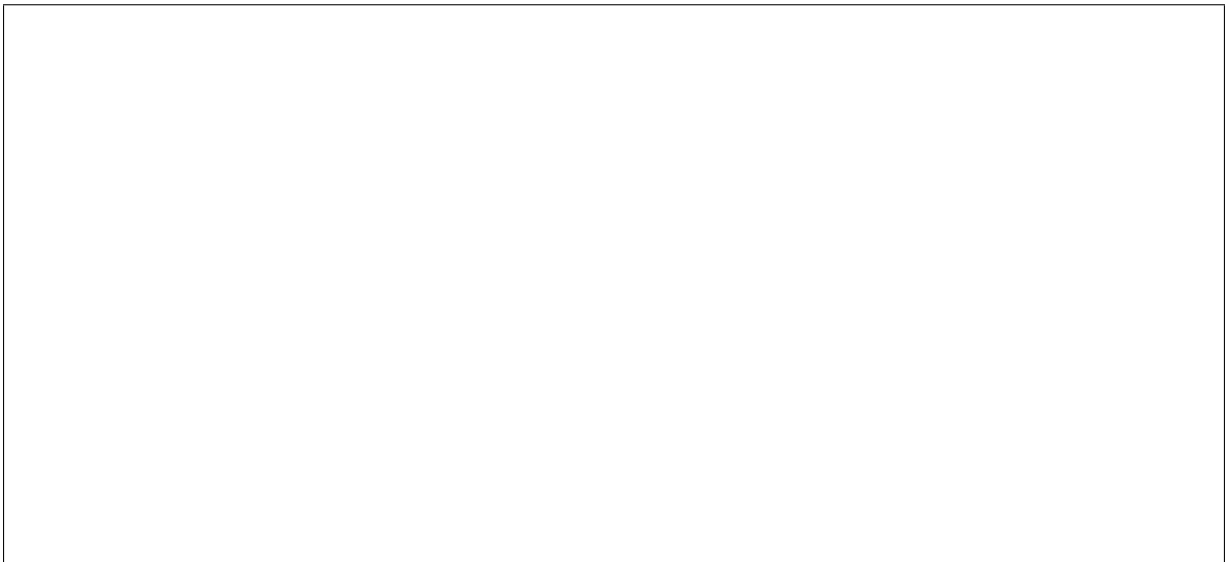
(continues on next page)

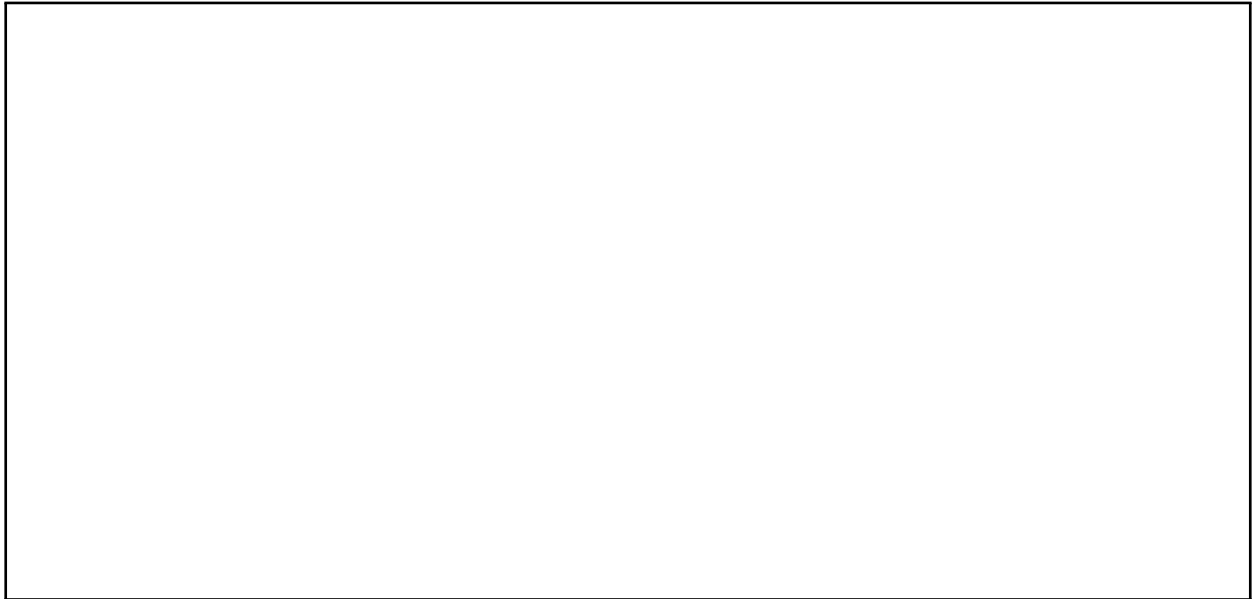
(continued from previous page)



## **Changing the version of the IPMI protocol**



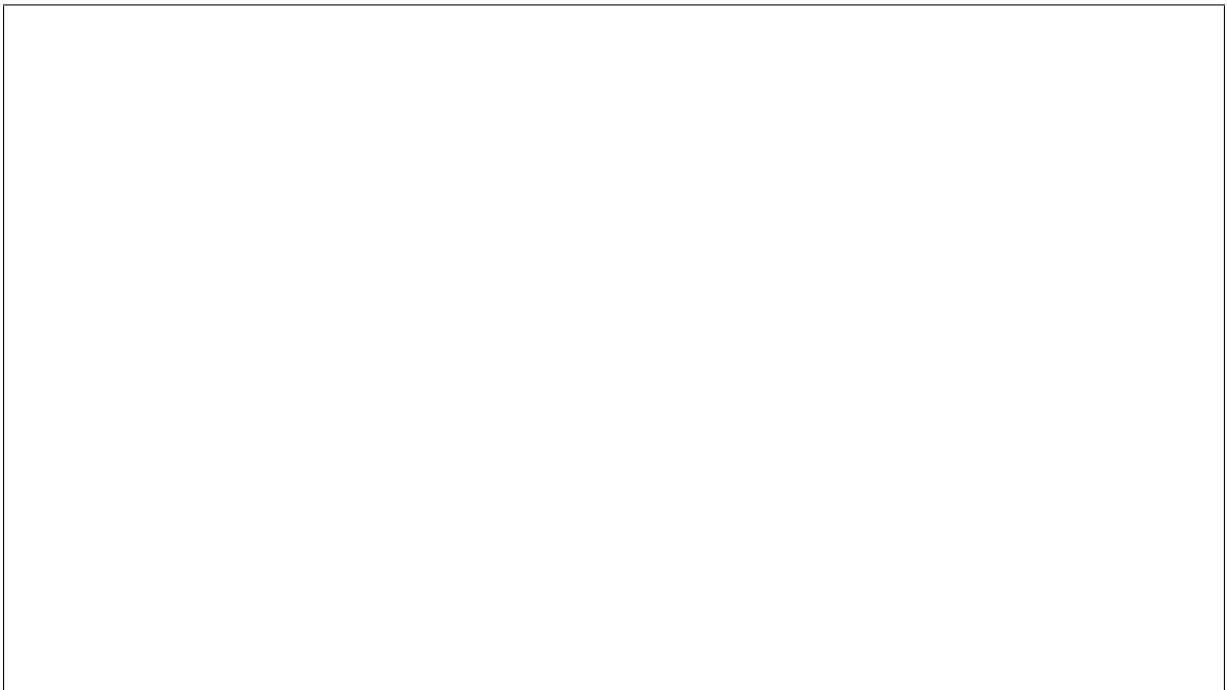




## **Cipher suites**

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







(continues on next page)

(continued from previous page)

---

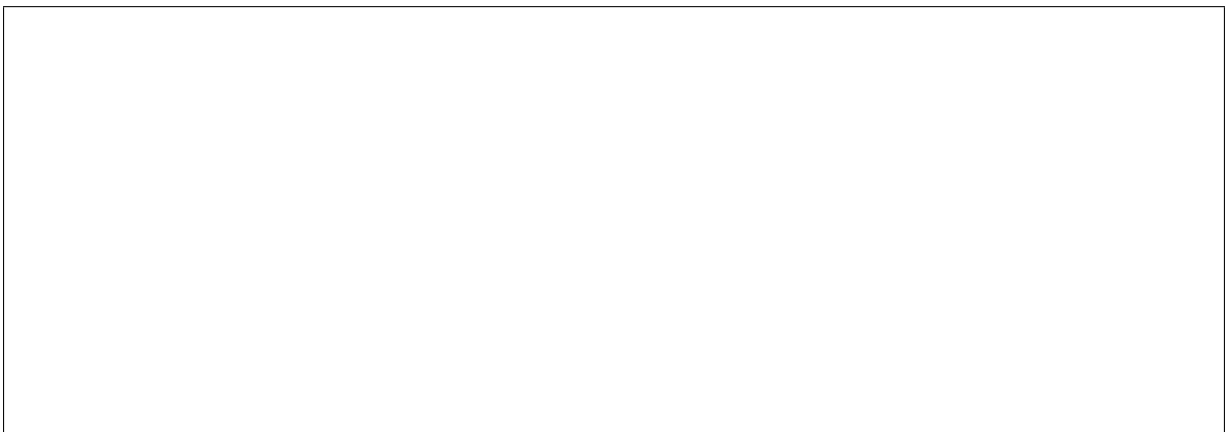
## **Static boot order configuration**

## **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)



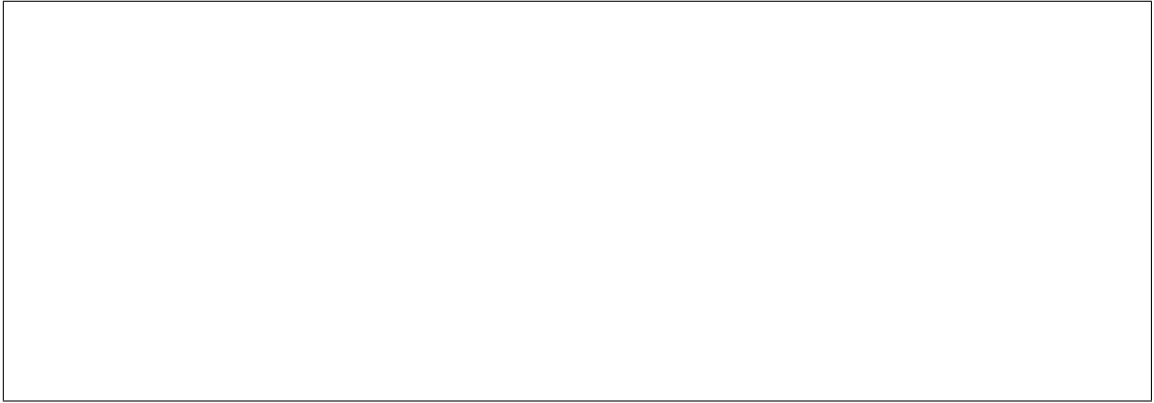
(continued from previous page)



## **iRMC driver**

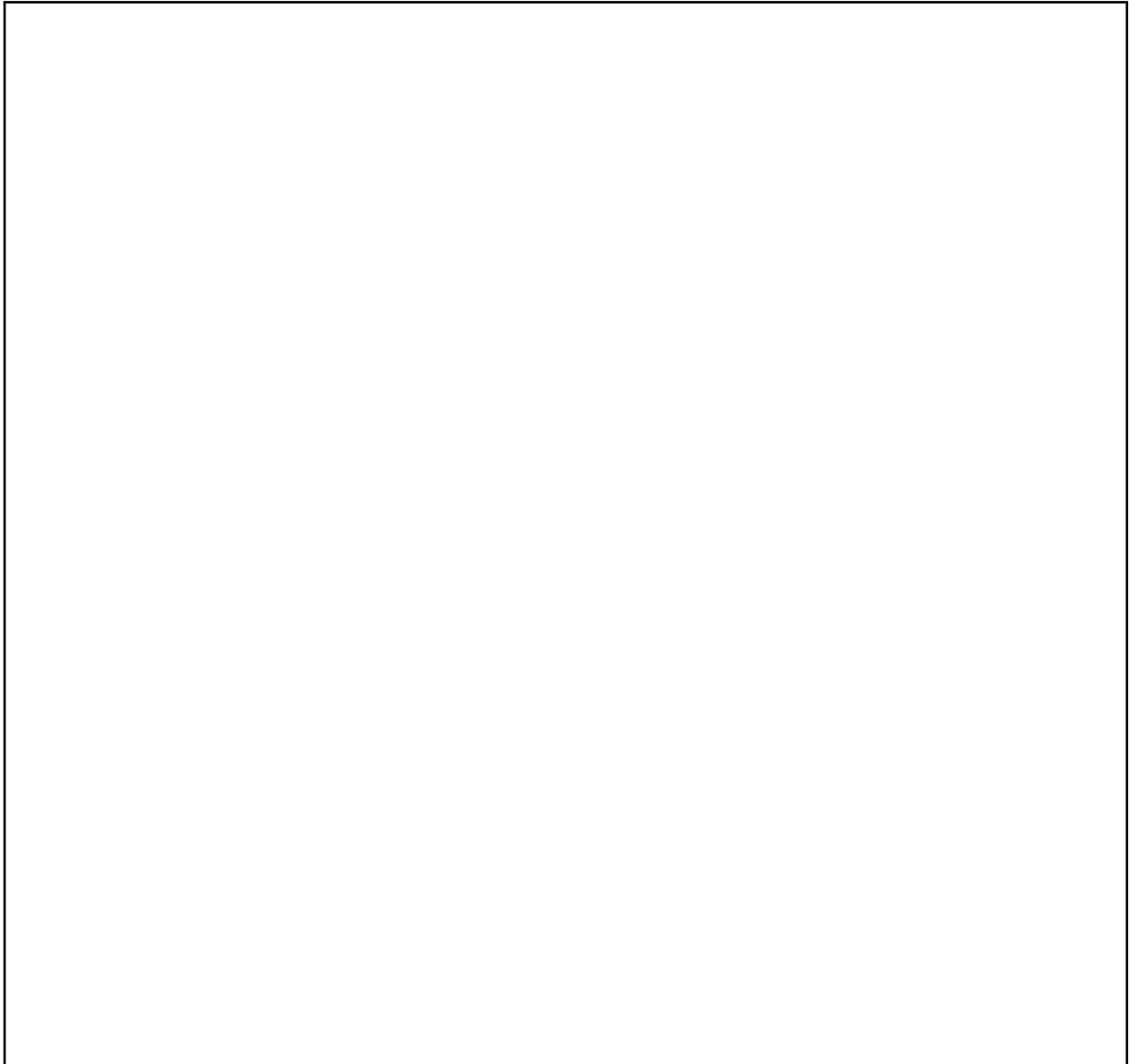
### **Overview**

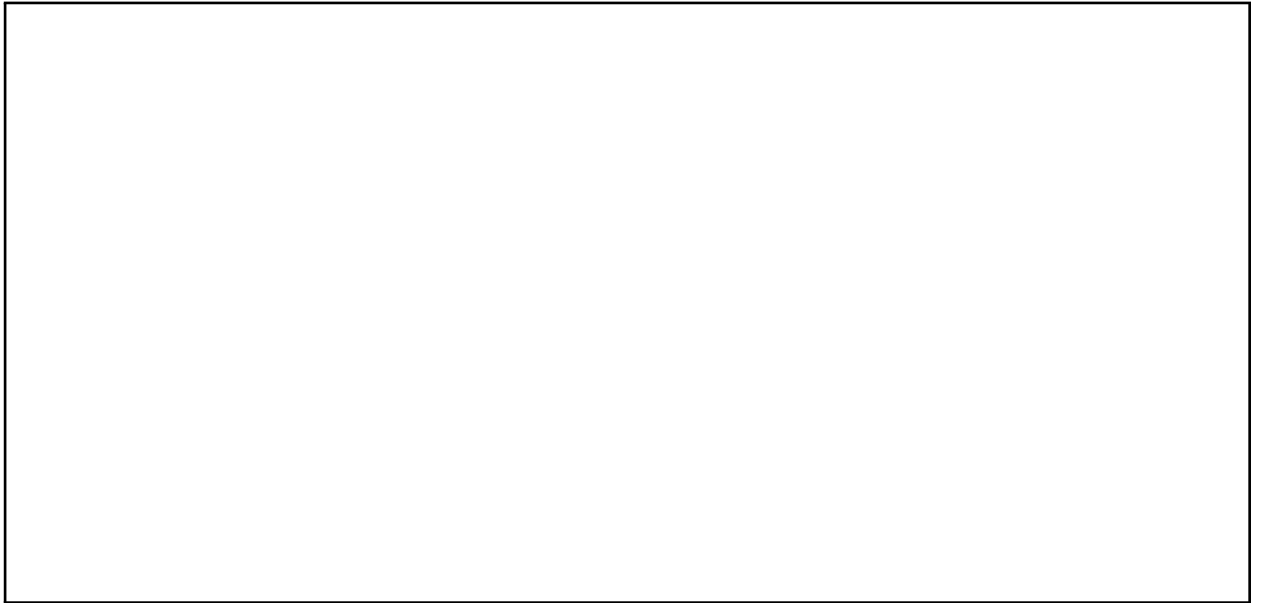
## **Prerequisites**



## **Hardware Type**

## **Hardware interfaces**







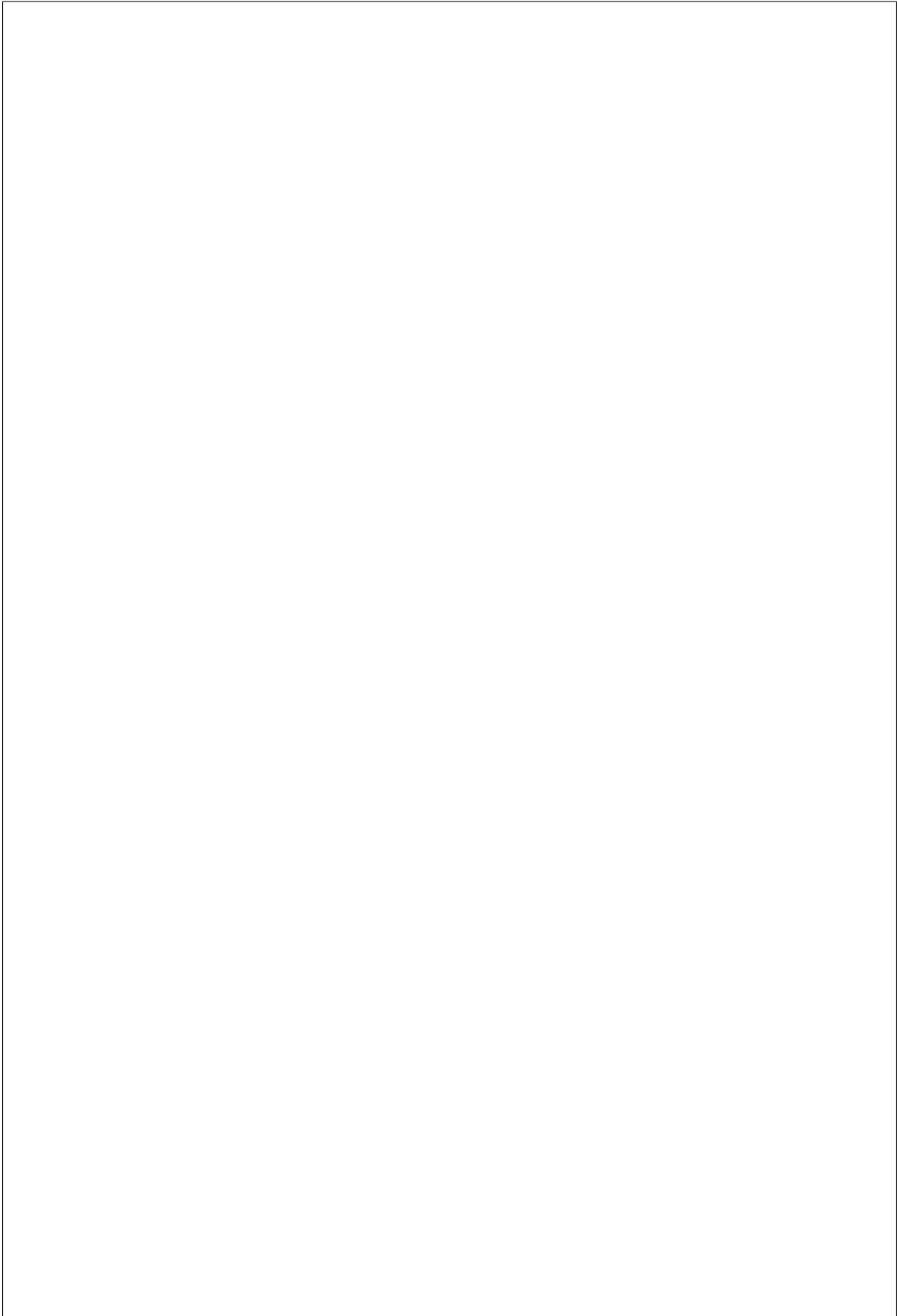






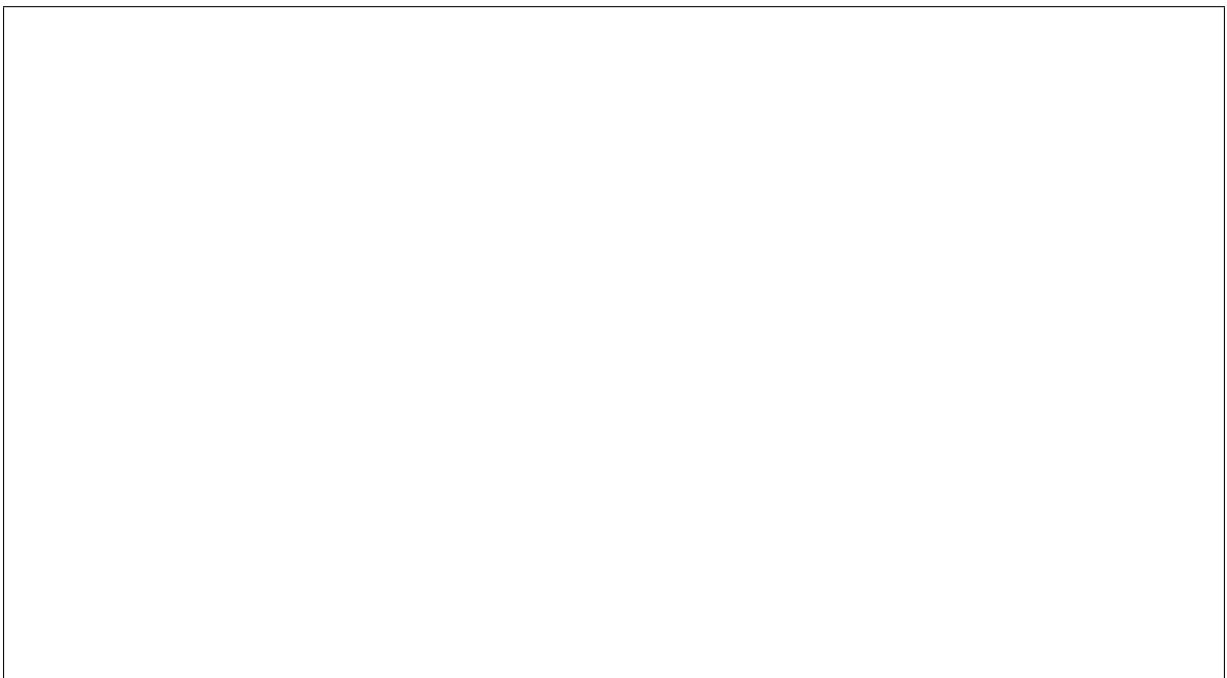
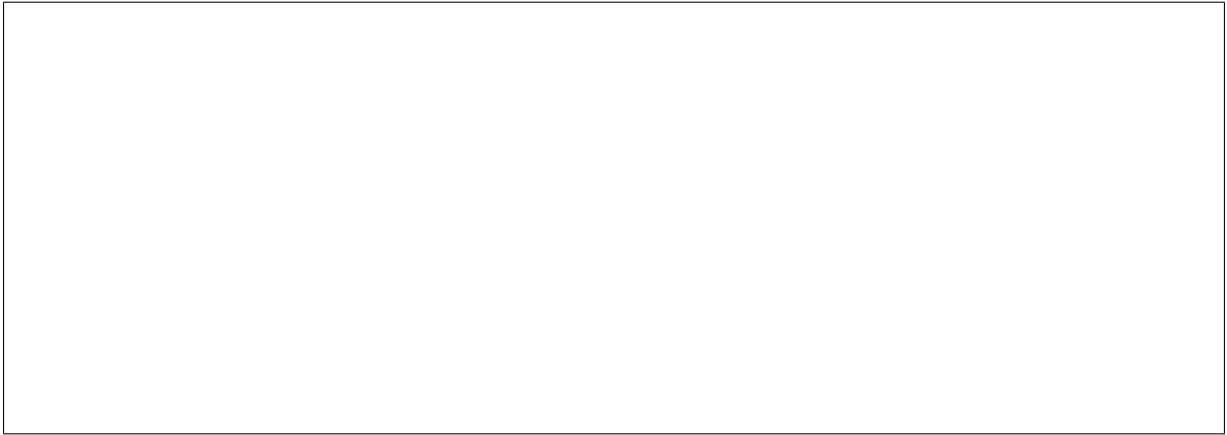
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



## **Node configuration**





























## **Optional functionalities for the `irmc` hardware type**

### **UEFI Secure Boot Support**

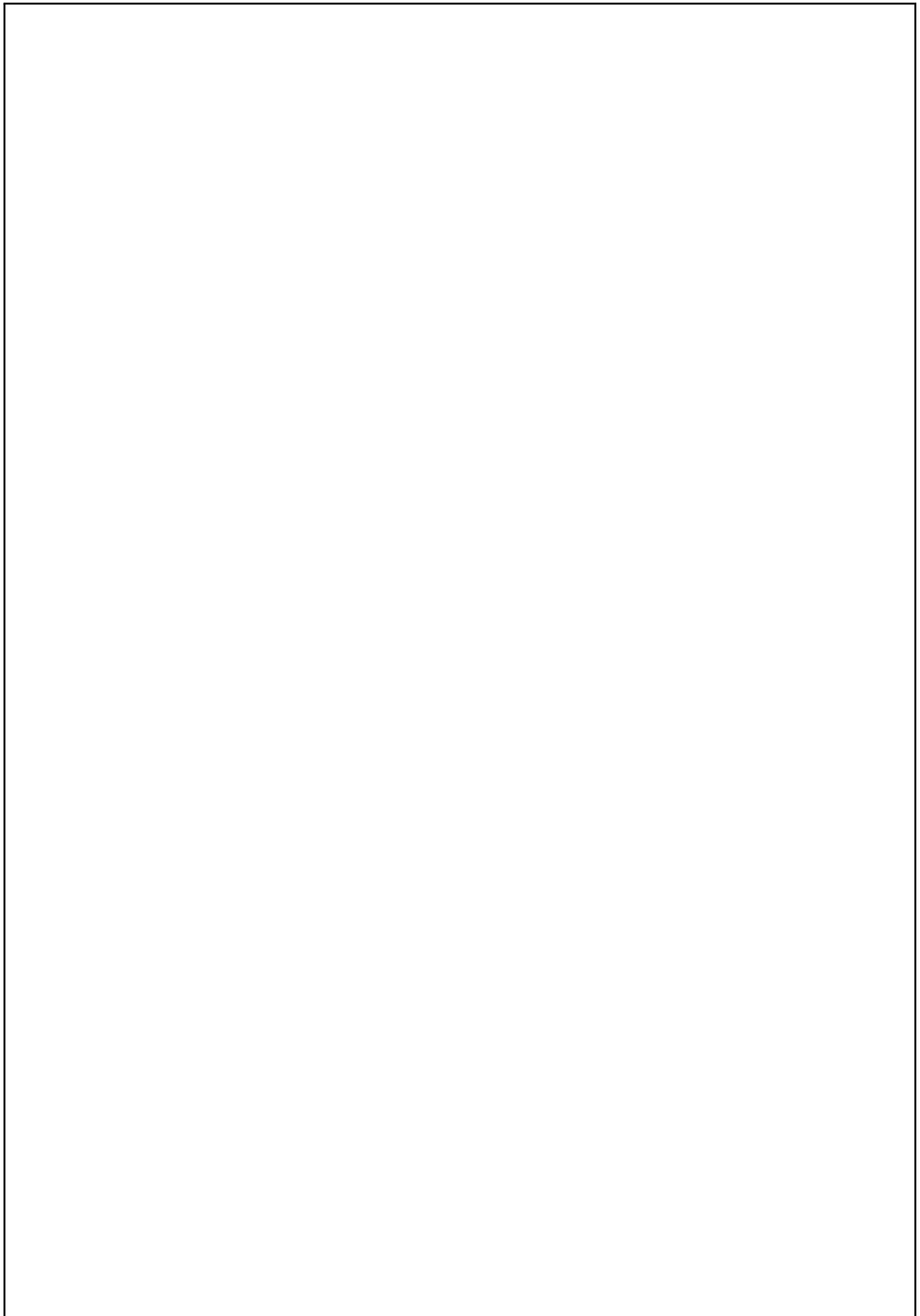


## **Node Cleaning Support**

### **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**

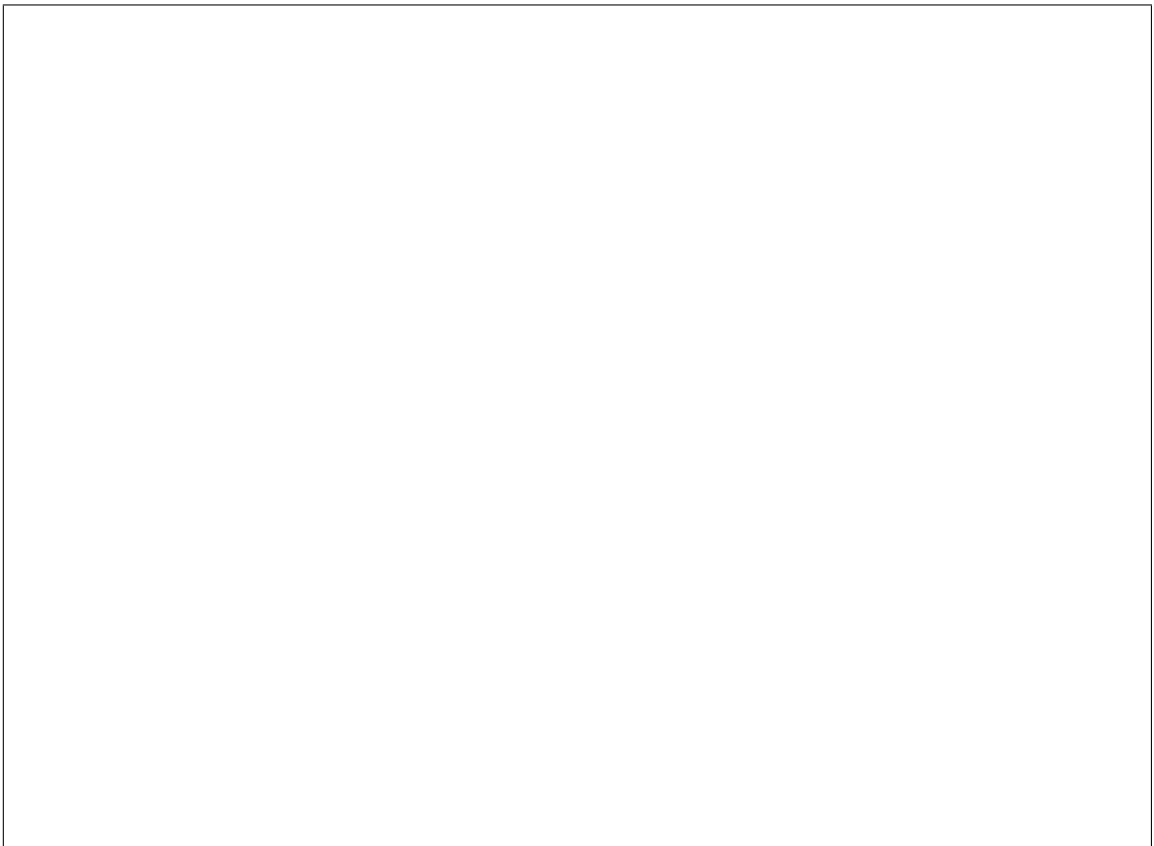
of iRMC. It supports iSCSI and FibreChannel.

## **Configuration**



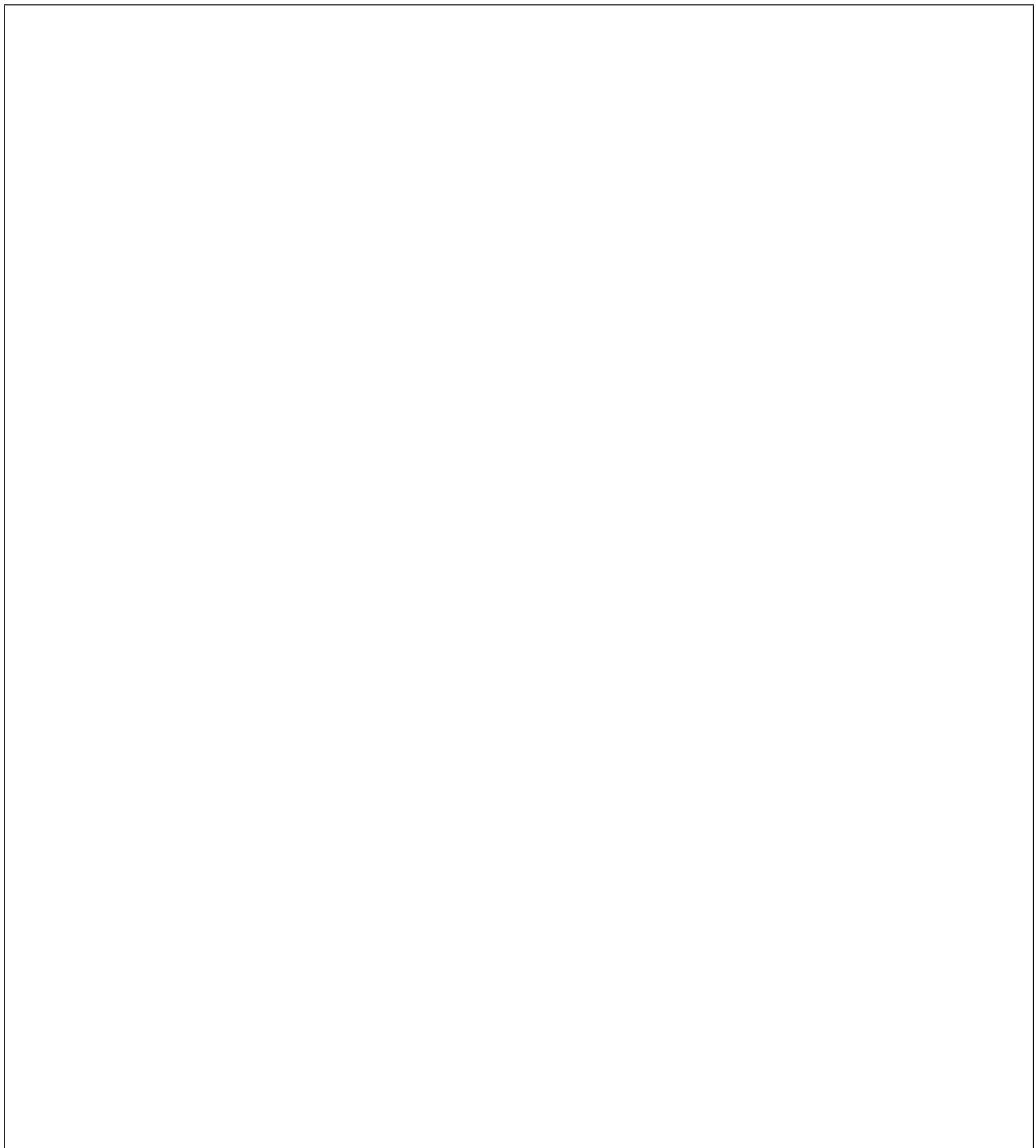






(continues on next page)

(continued from previous page)

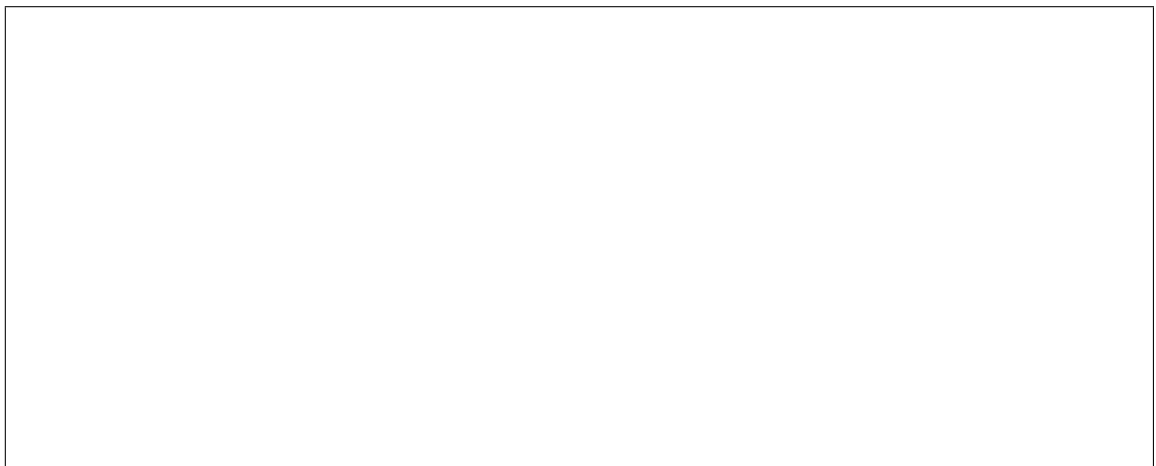


(continues on next page)

(continued from previous page)

--

mand:



## **Supported hardware**

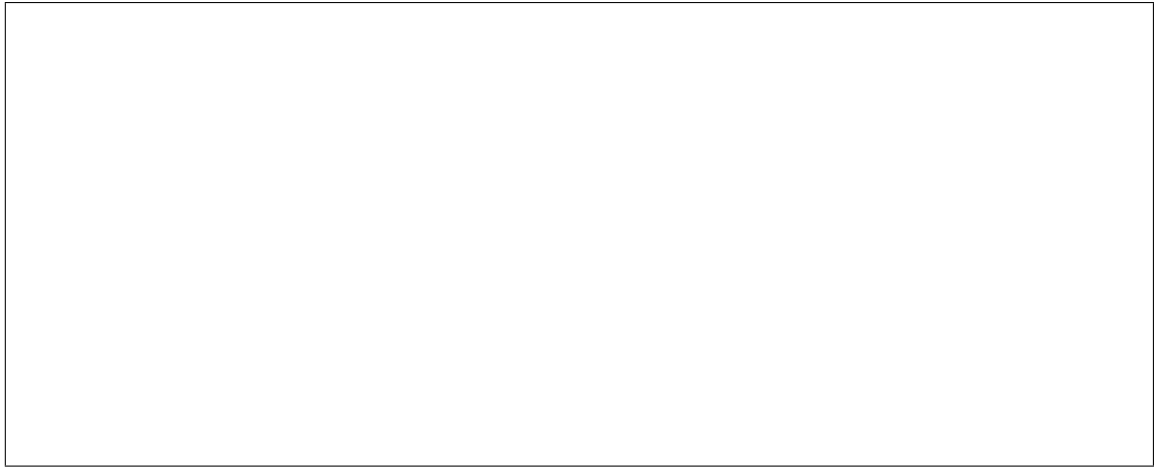
## **Hardware Inspection Support**

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

## **Configuration**







loaded from [here](#).

### **Supported properties**



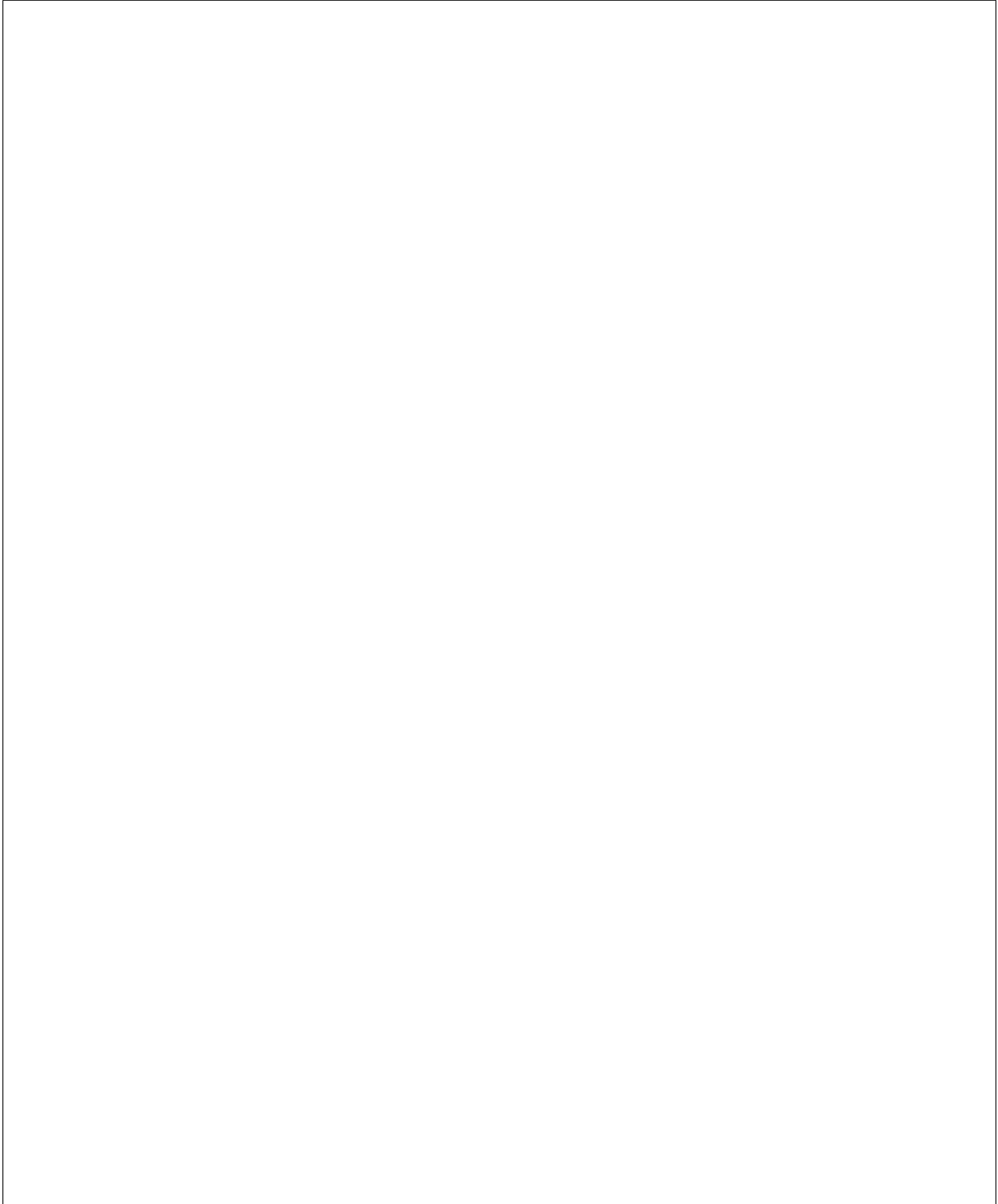


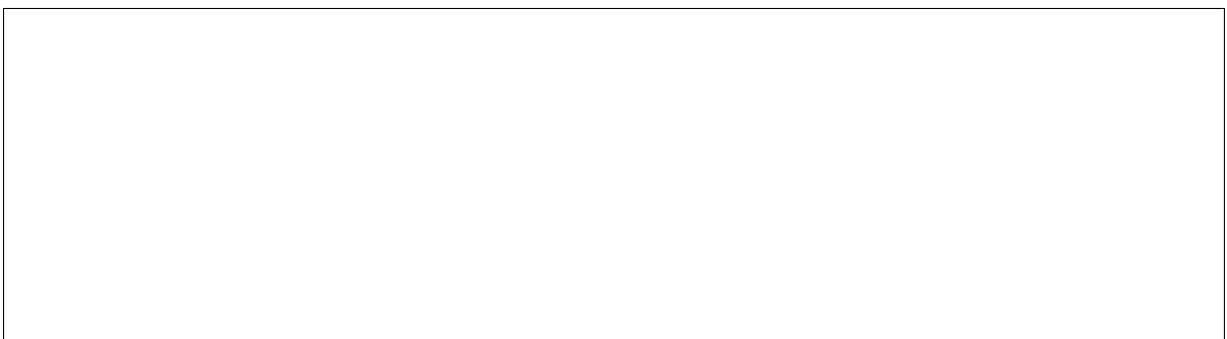
**Note:**



state.





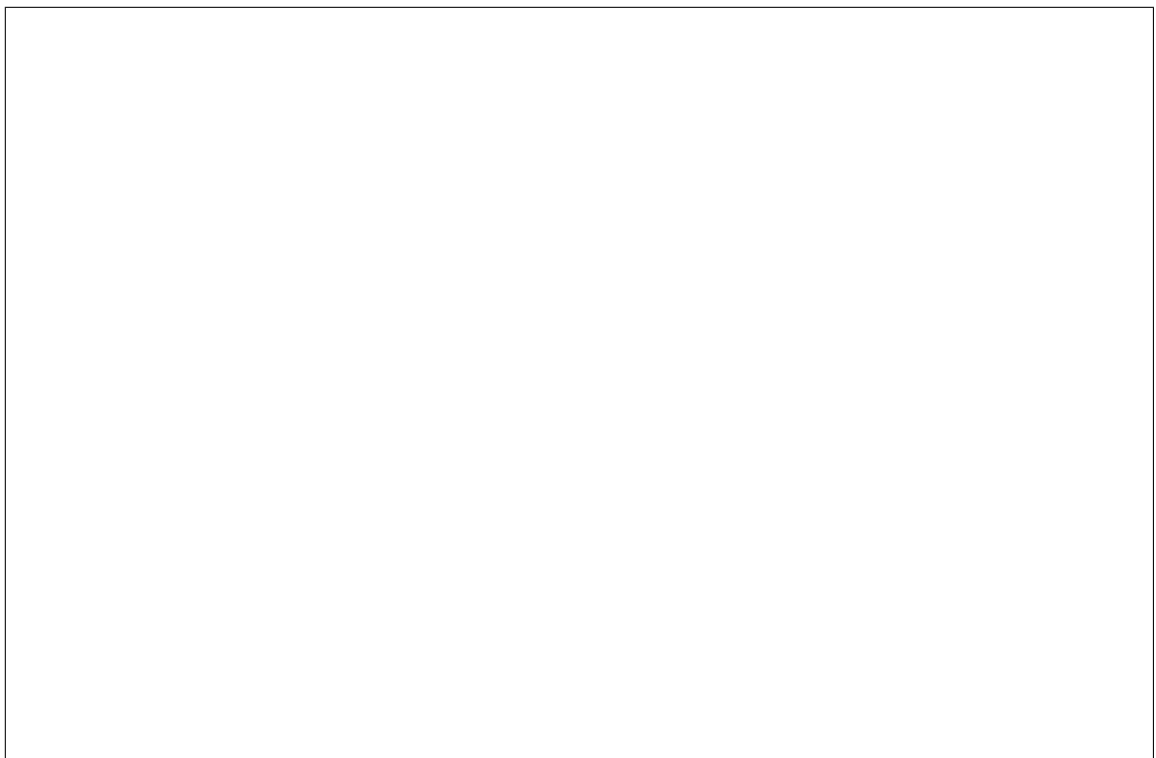




## **RAID configuration Support**

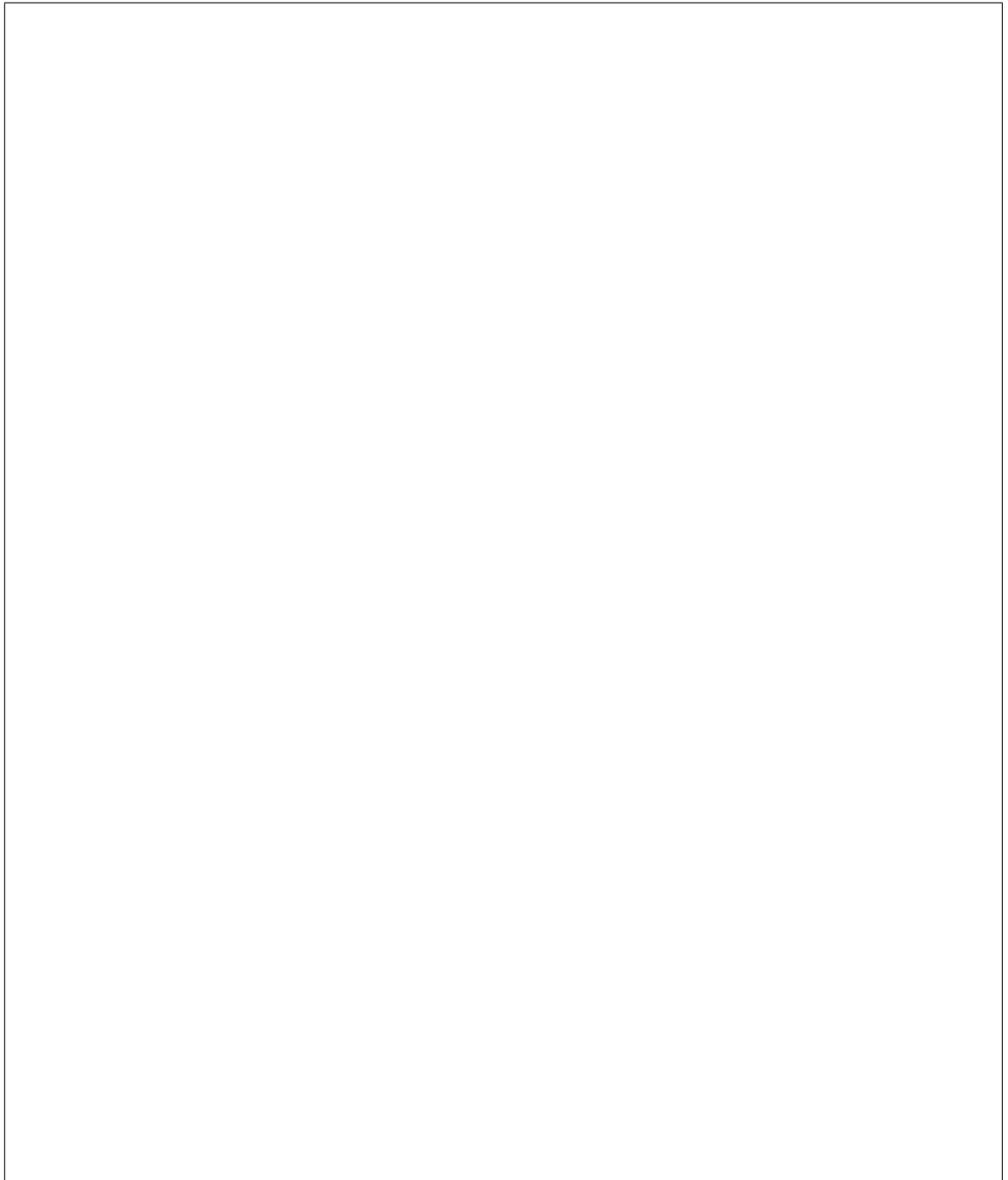
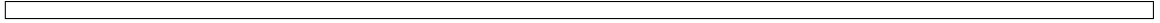
**Note:**

## **Configuration**



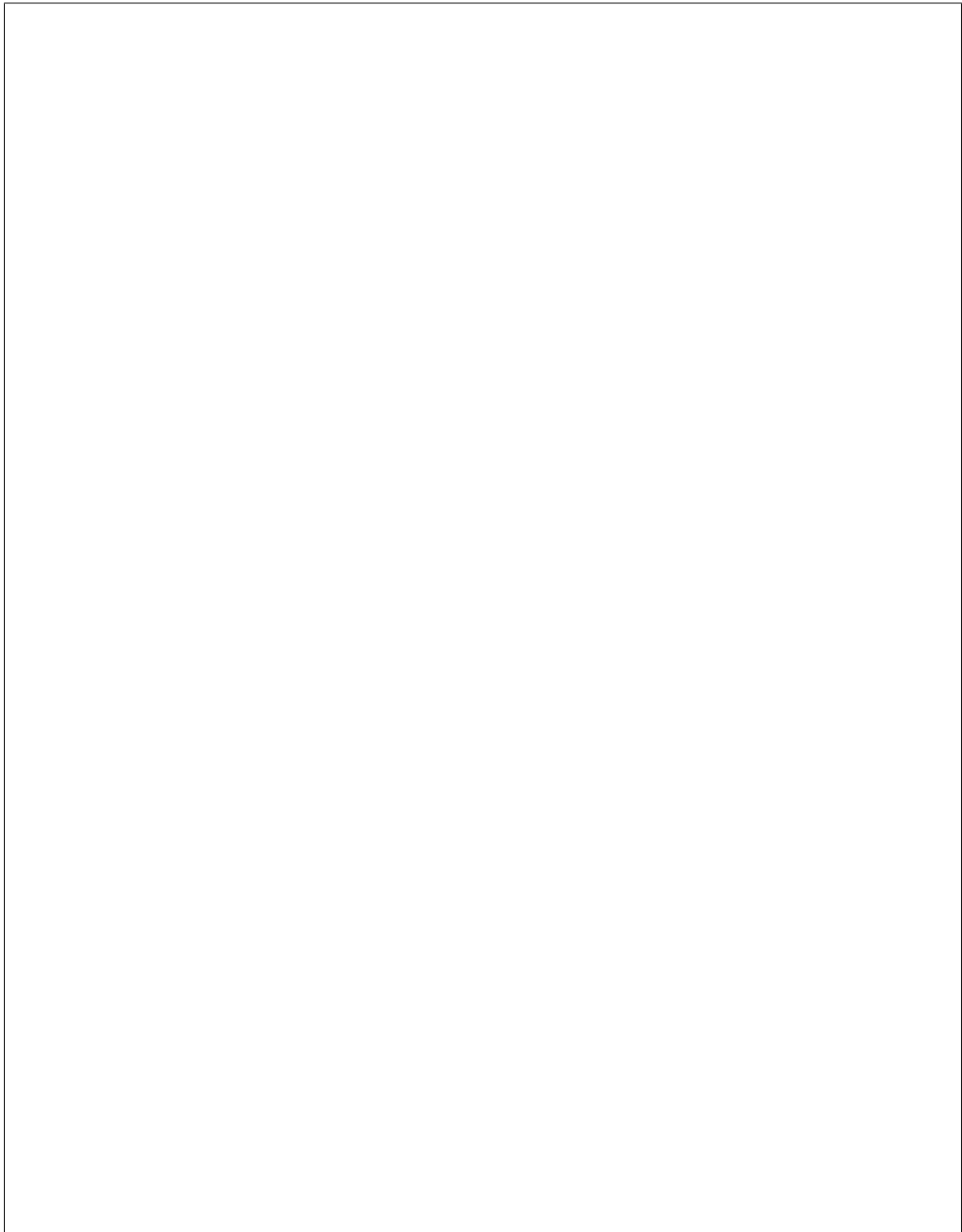
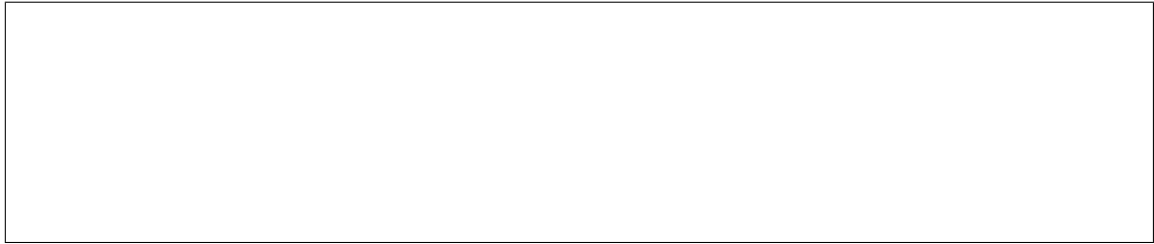
(continues on next page)

(continued from previous page)



(continues on next page)

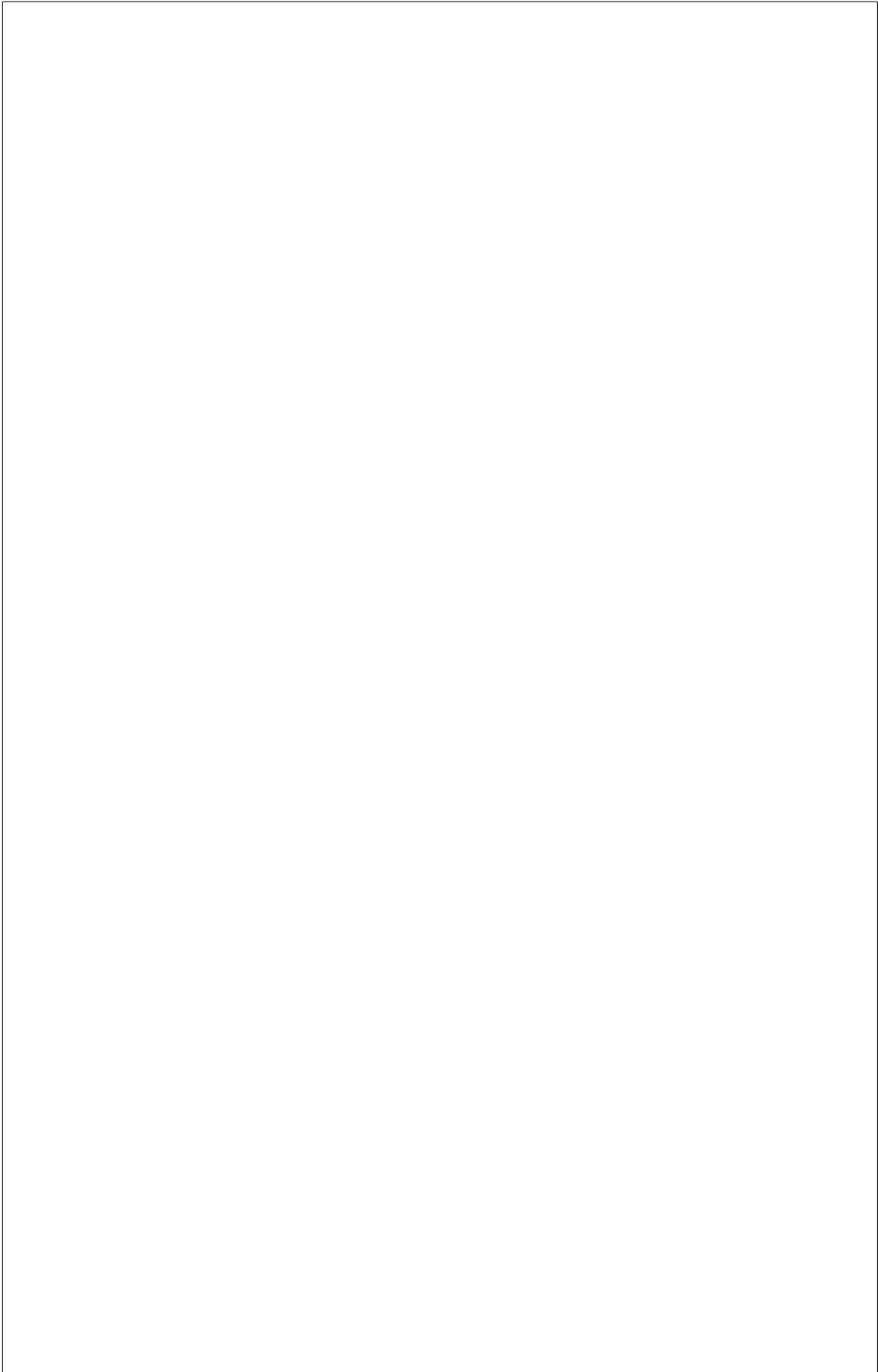
(continued from previous page)



(continues on next page)

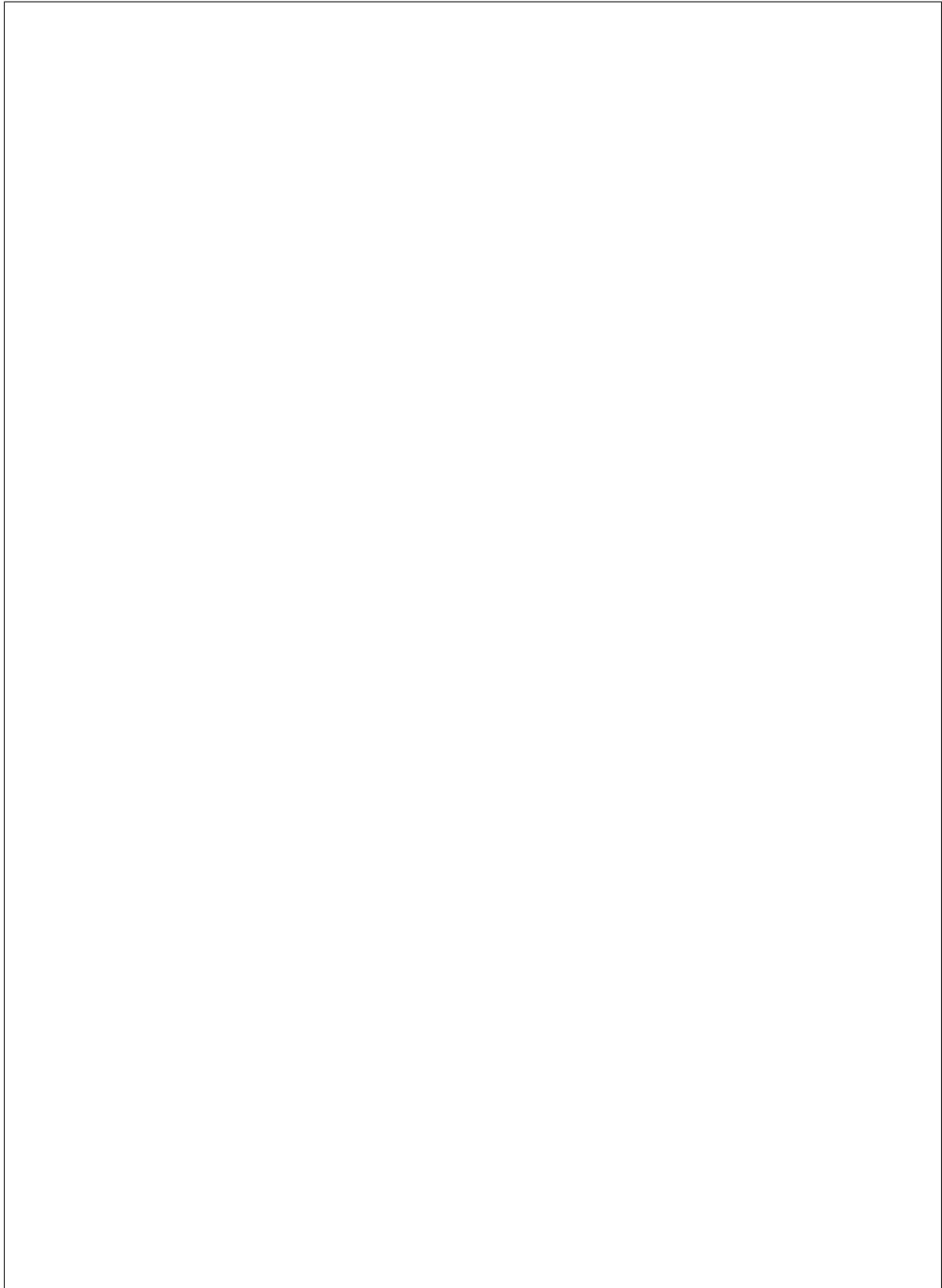


(continued from previous page)



(continues on next page)

(continued from previous page)

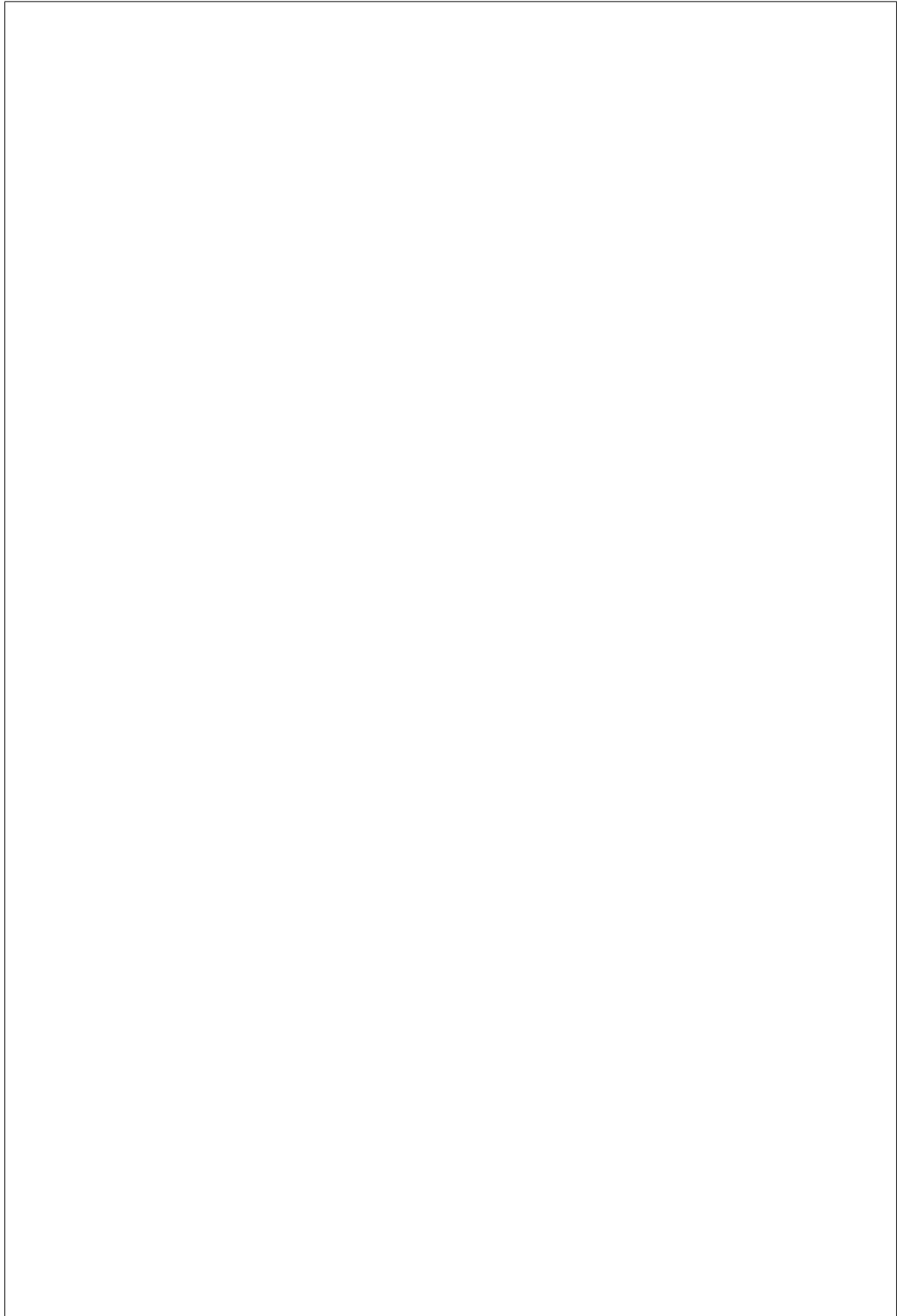


**Note:** RAID 1+0 and 5+0 in iRMC driver does not support property `physical_disks` in `target_raid_config` during create raid configuration yet. See following example:



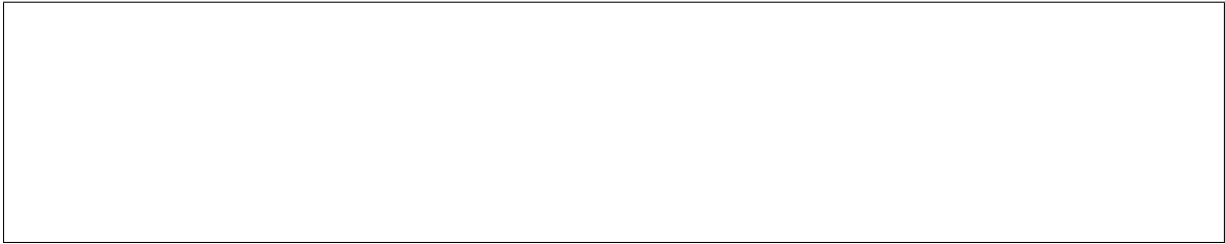
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



## **Supported properties**

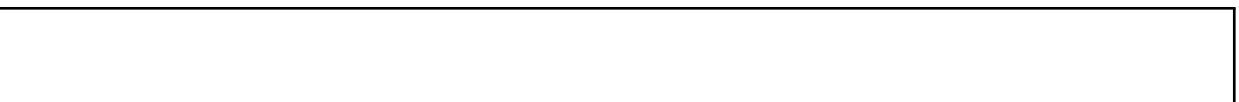




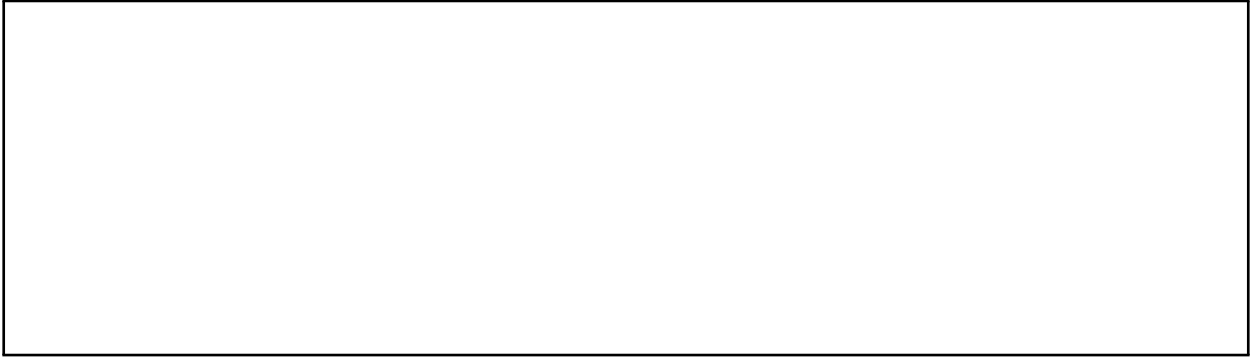
**Note:**

ated RAID on iRMC server.

## **BIOS configuration Support**







## **Configuration**





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















## **Supported platforms**





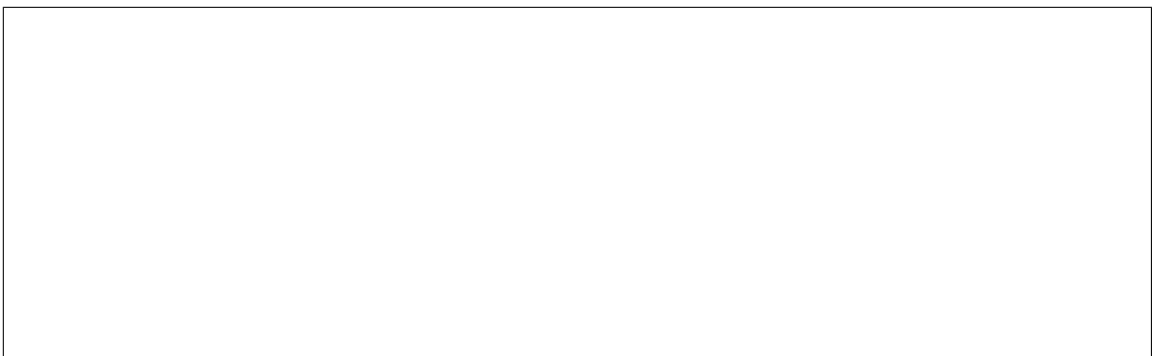
## **Redfish driver**

### **Overview**

### **Prerequisites**

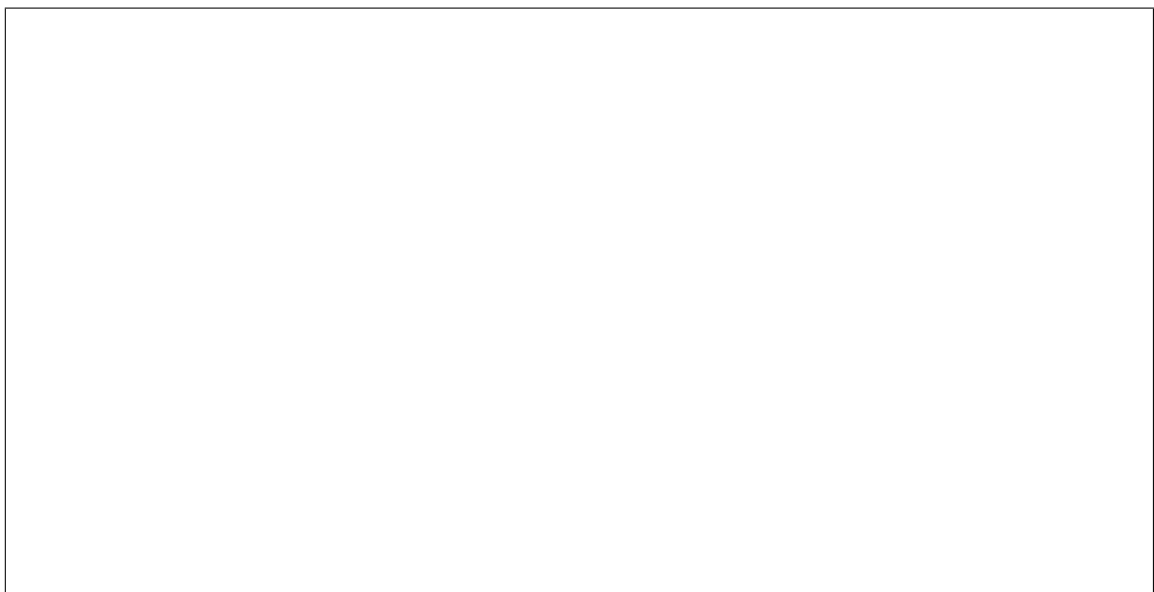


### **Enabling the Redfish driver**



(continues on next page)

(continued from previous page)





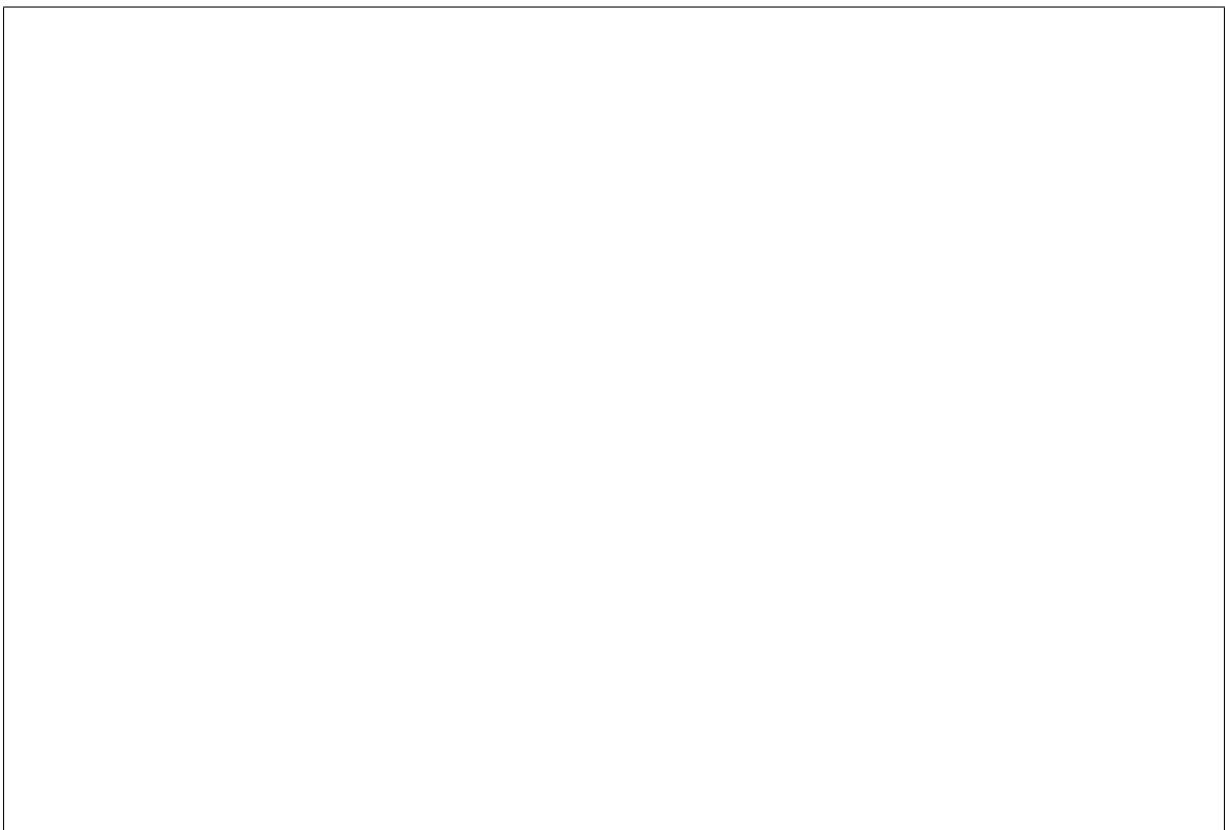
## **Registering a node with the Redfish driver**



erwise ironiC will pick the only available ComputerSystem automatically. For example: /redfish/v1/Systems/1.



tificate file or directory with trusted certificates that the driver will use for verification. To disable verifying [TLS](#), set this to False. This is optional.



(continues on next page)

(continued from previous page)



## **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.

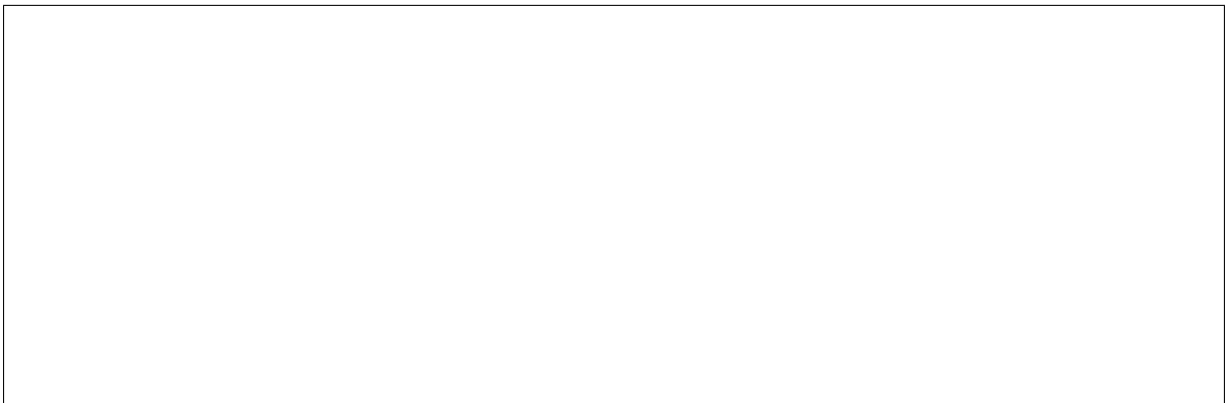
**Note:** The `local_gb` property cannot always be discovered, for example, when a node does not have local storage or the Redfish implementation does not support the required schema. In this case the property will be set to 0.

## 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.







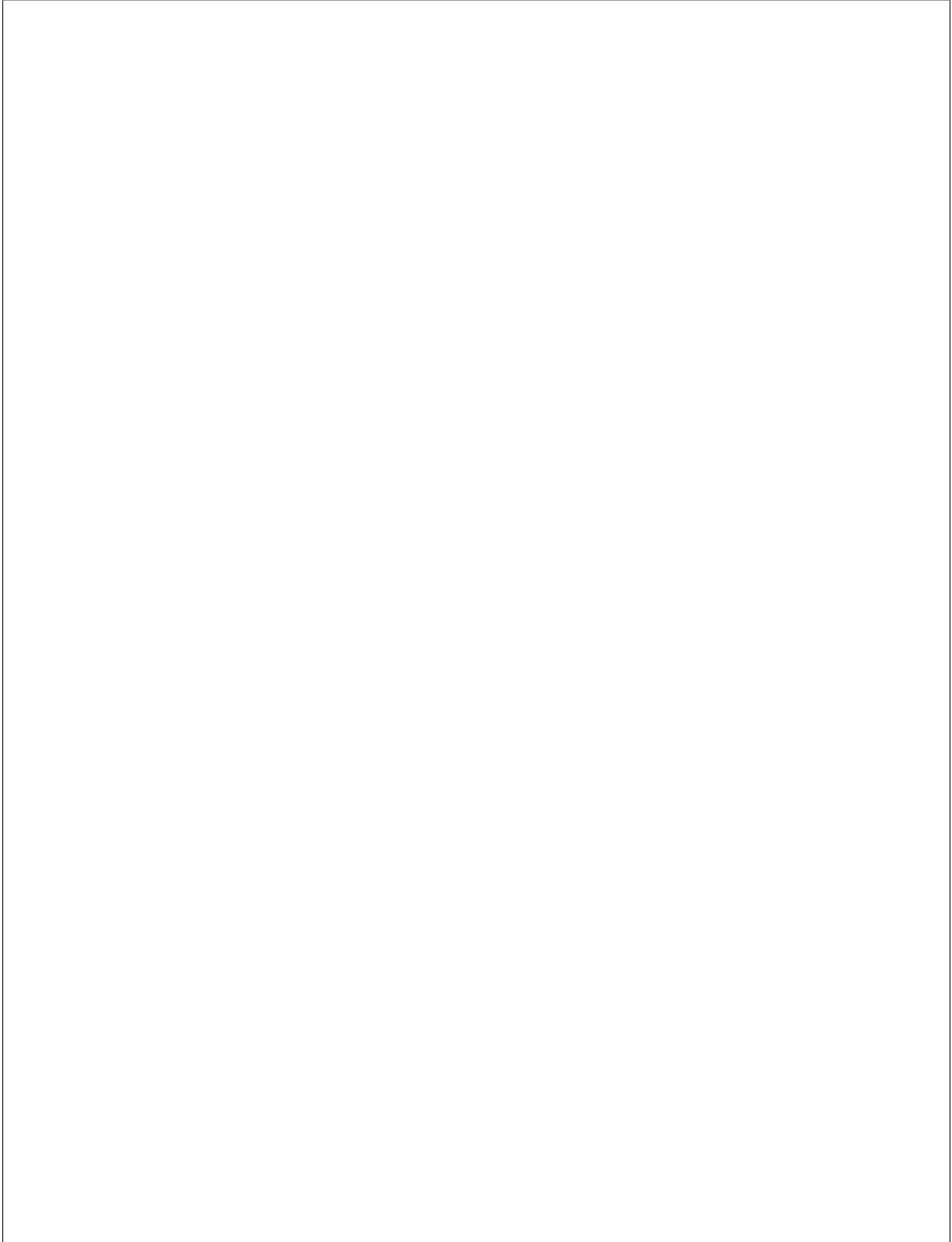




serves the same purpose.

### **Configuring an ESP image**

to get them.



(continues on next page)

(continued from previous page)

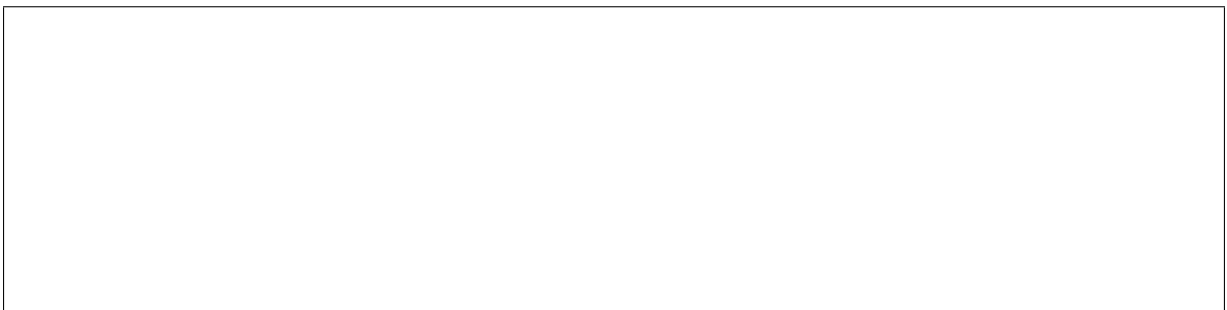
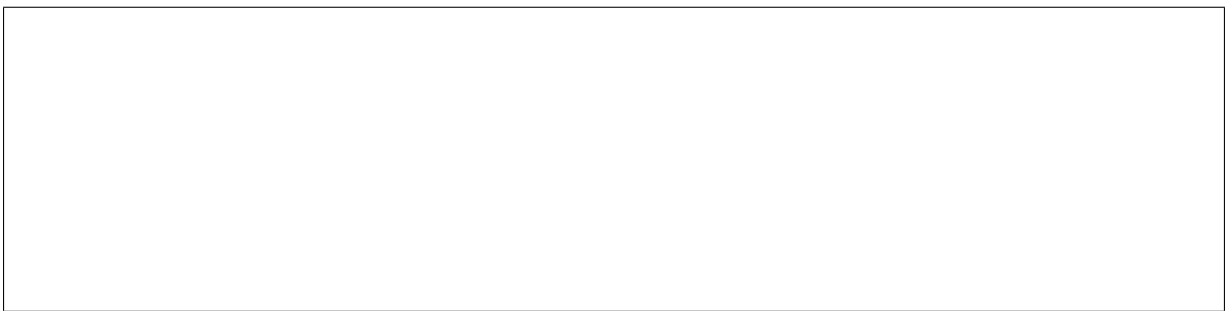


**Note:** If you use an architecture other than x86-64, you'll need to adjust the destination paths.







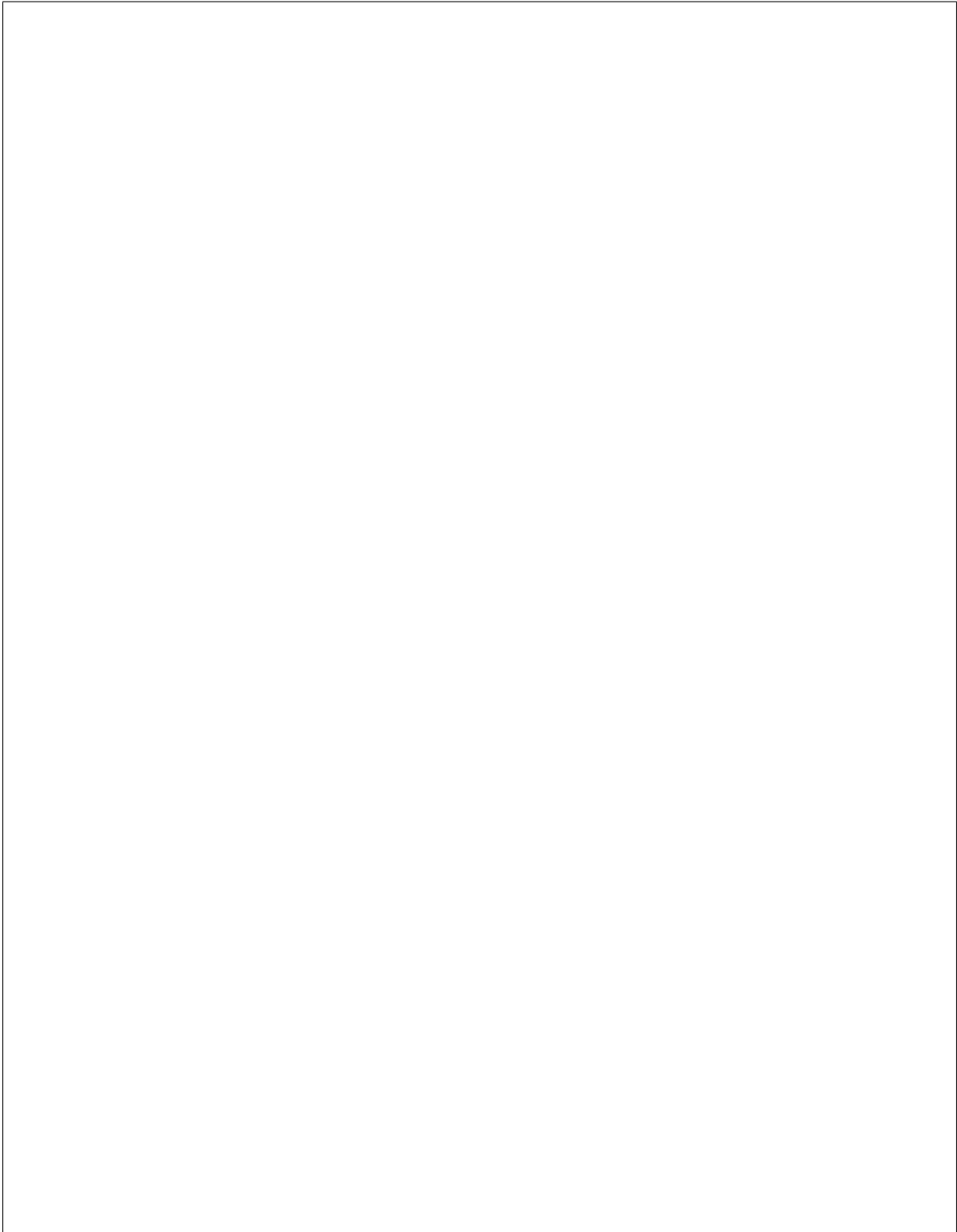


**Note:** Unlike in the script above, these paths are case-sensitive!

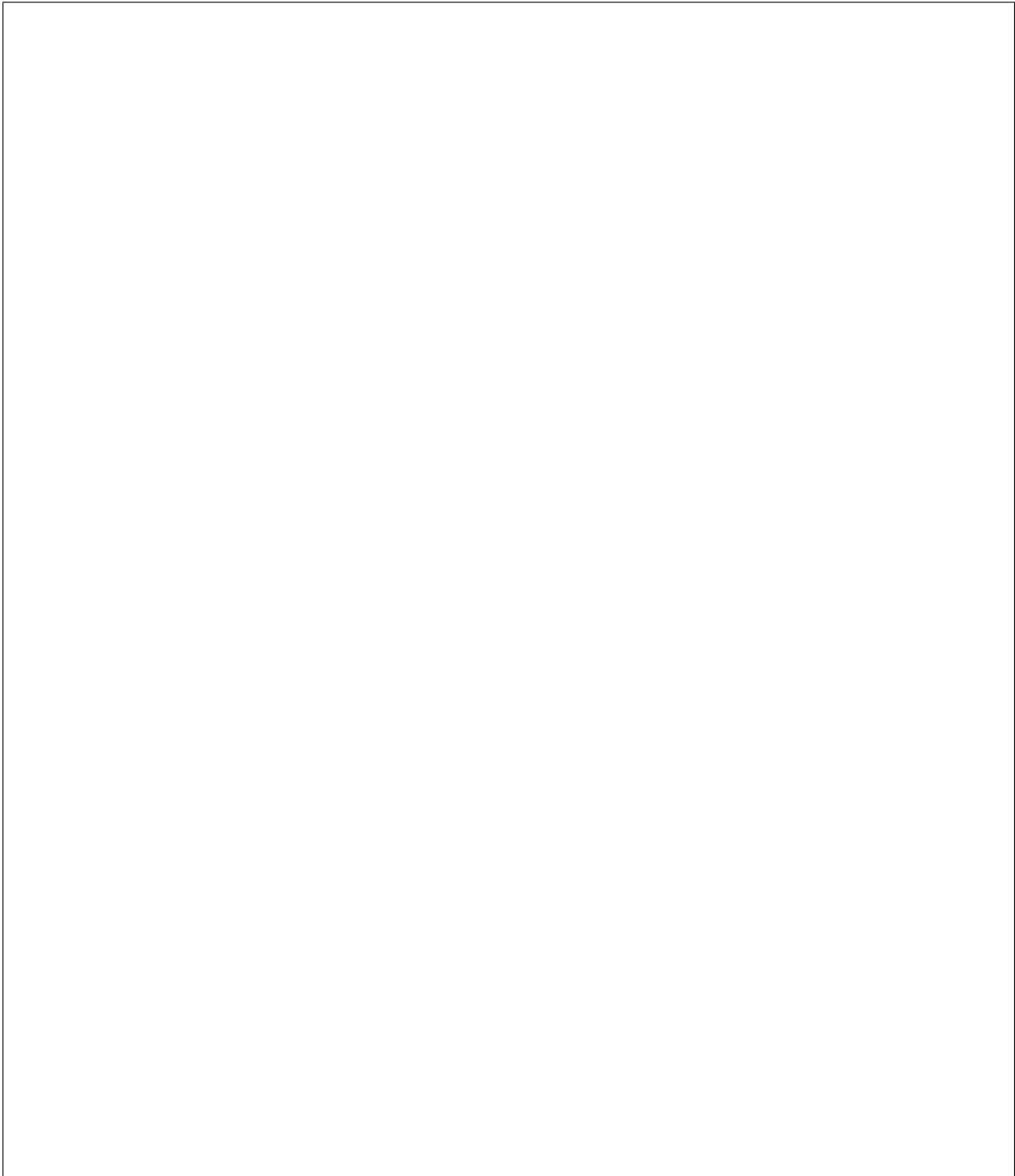
### **Virtual Media Ramdisk**

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`.









**Layer 3 or DHCP-less ramdisk booting**

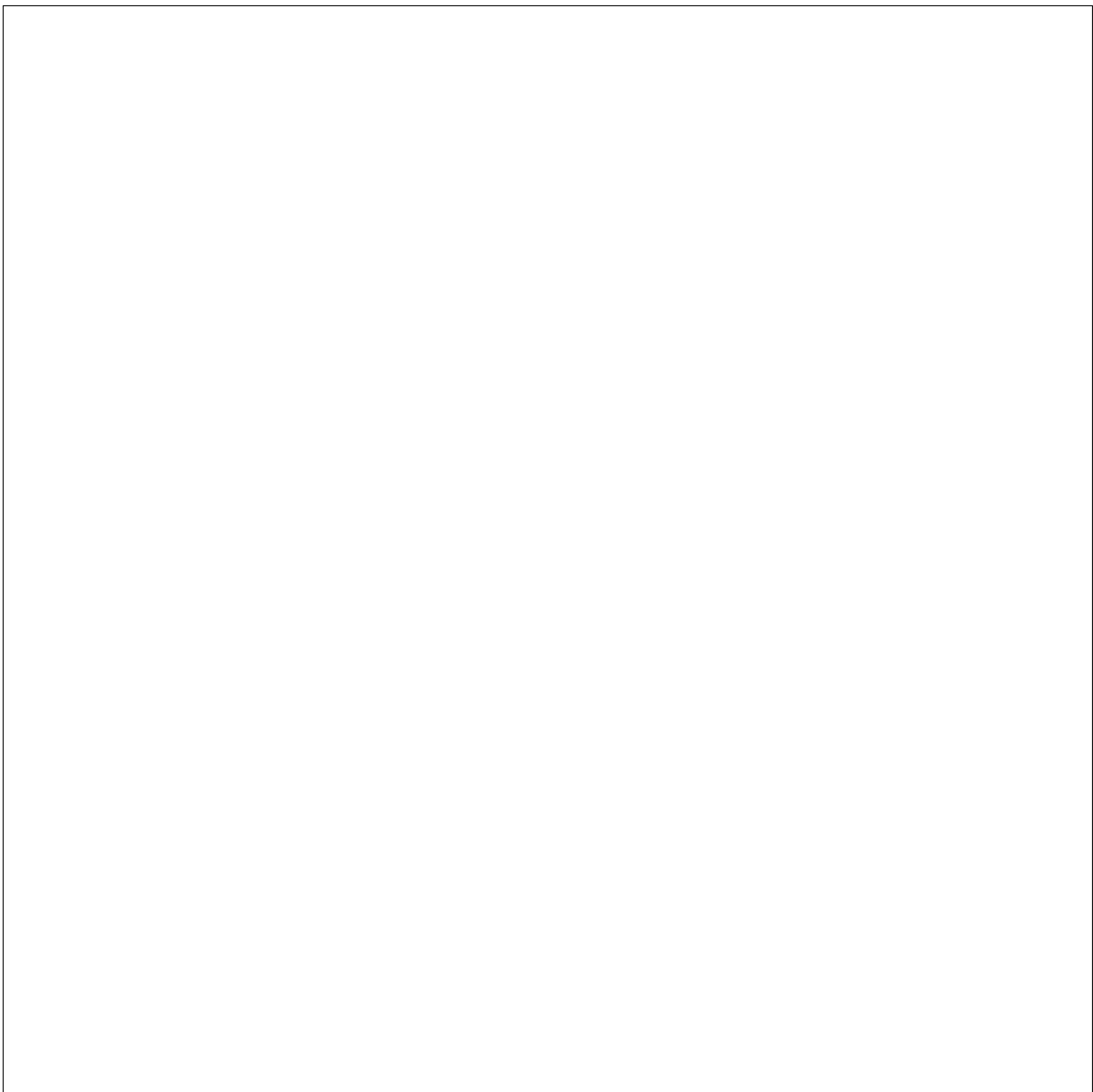
## **Firmware update using manual cleaning**



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.



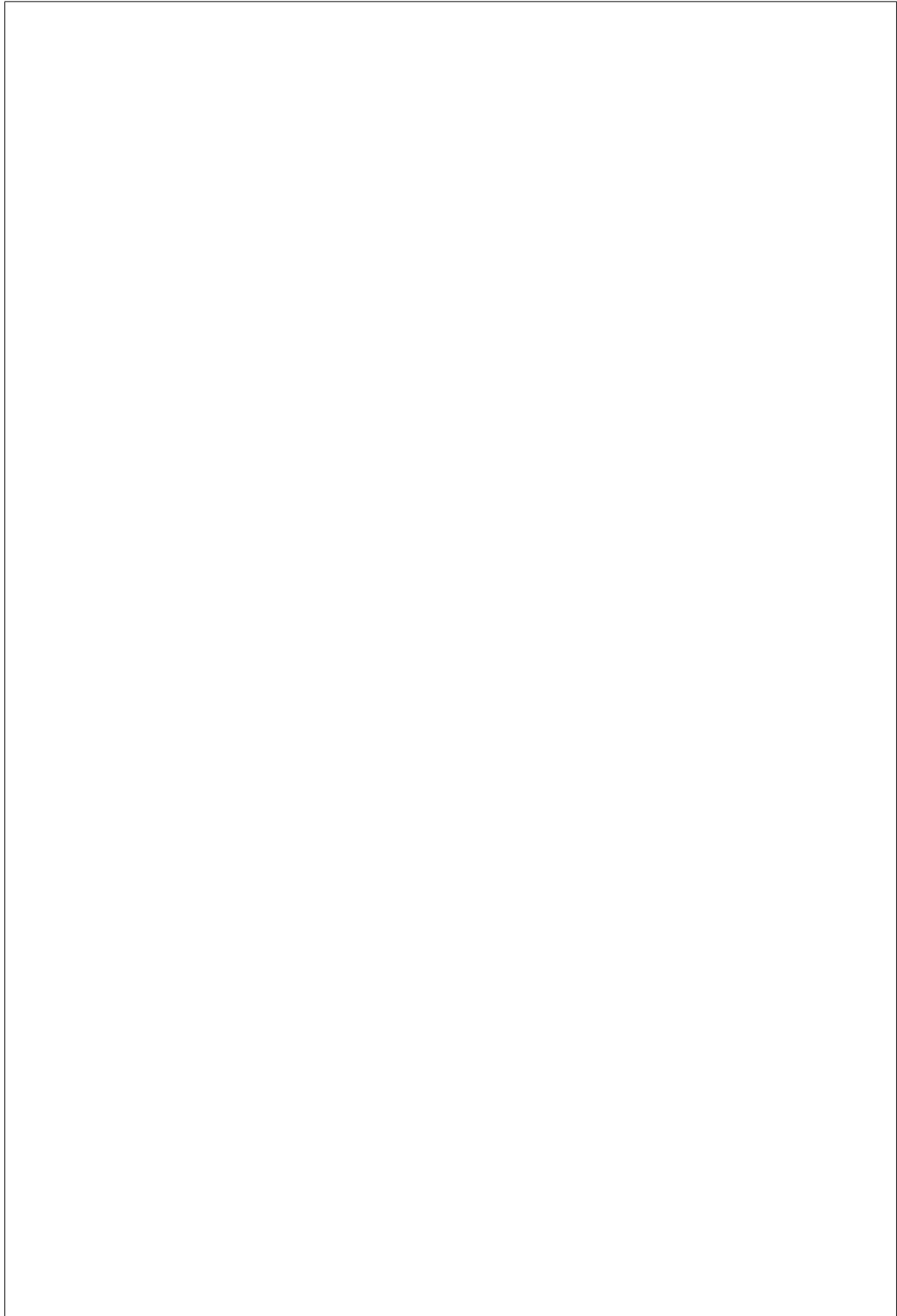
(continues on next page)

(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>&lt;name&gt;: &lt;value&gt;</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.



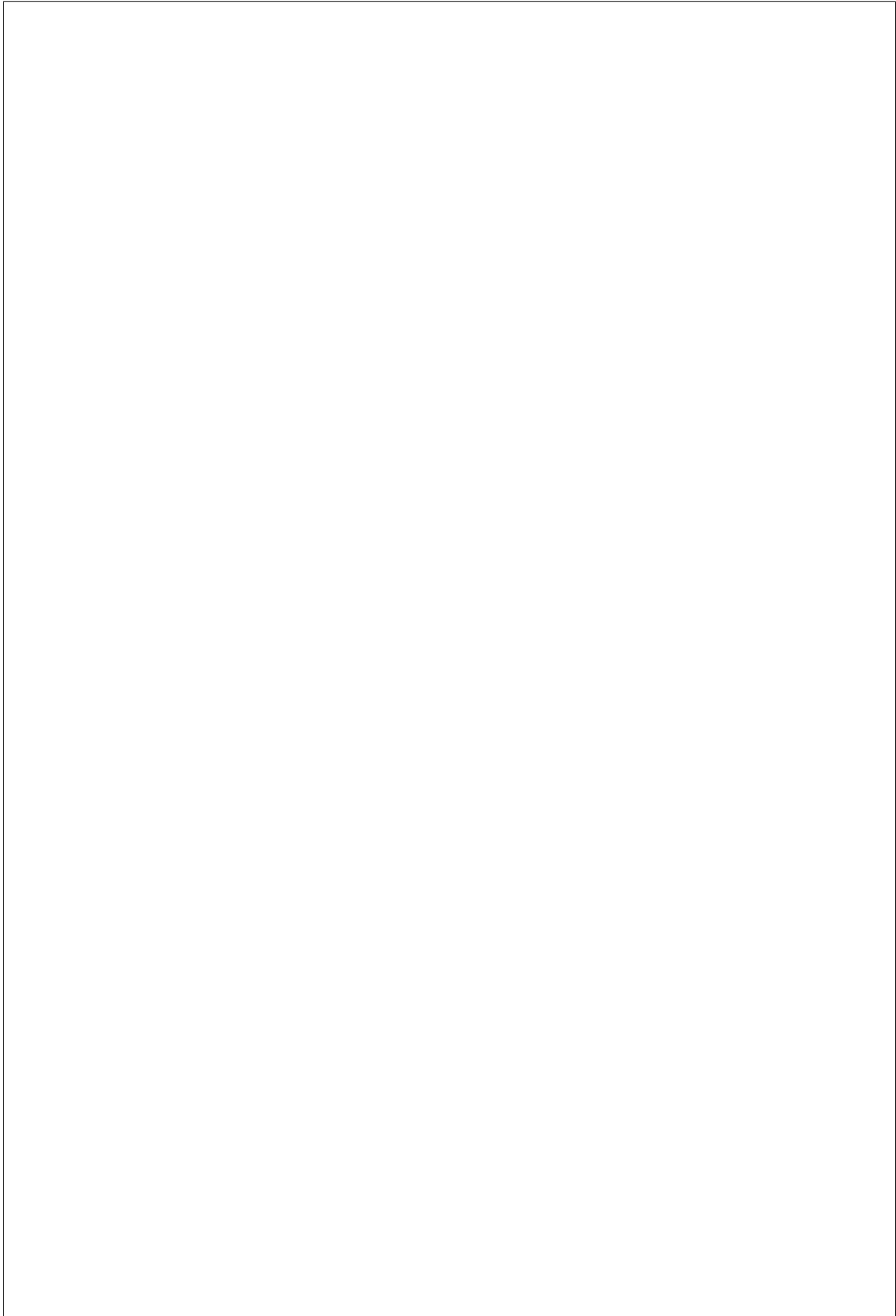
lar tool on a server that has network access to the BMC, try downloading the firmware to verify that the URLs are correct and that the web server is configured properly.

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)



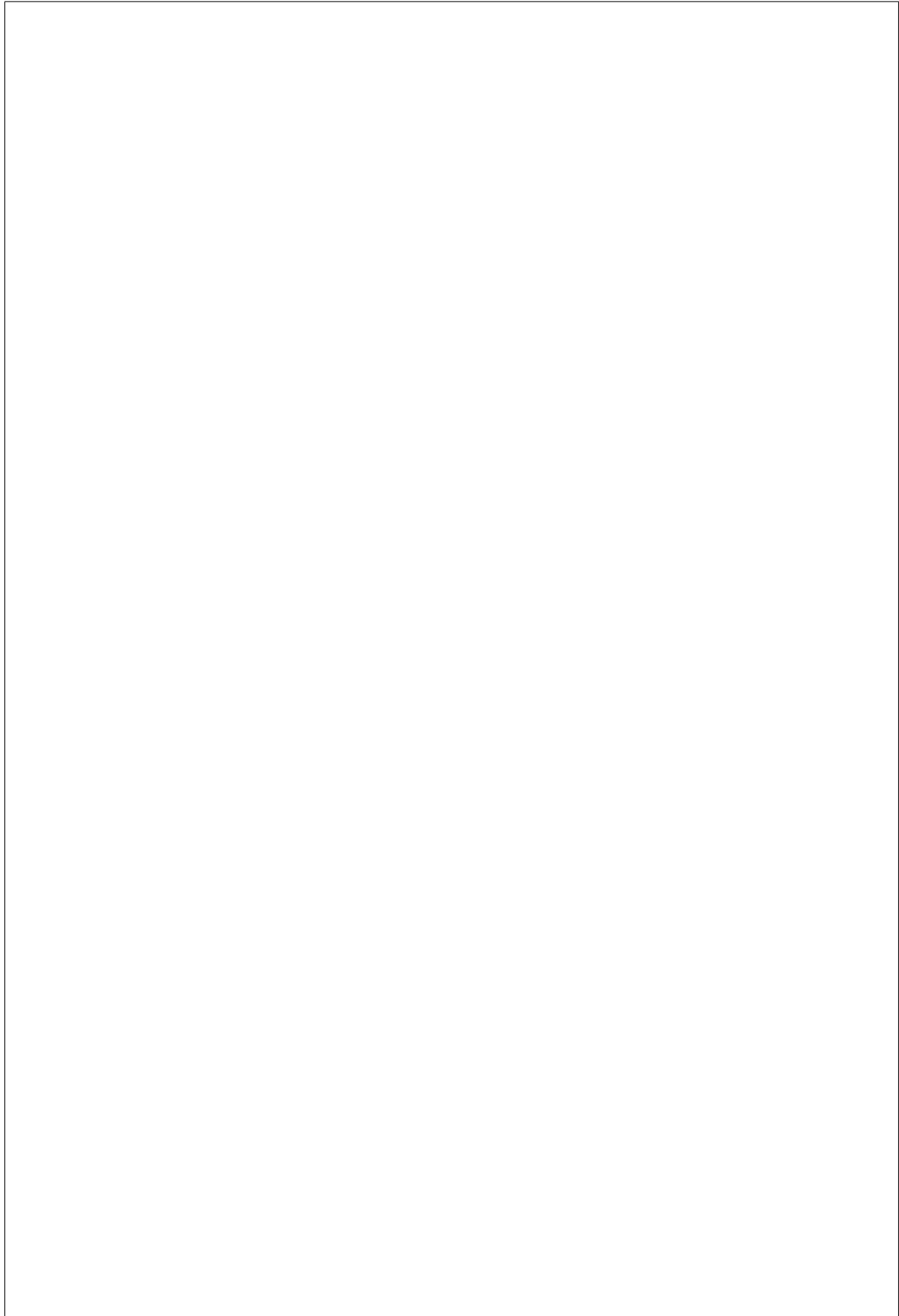
(continues on next page)

(continued from previous page)



(continues on next page)

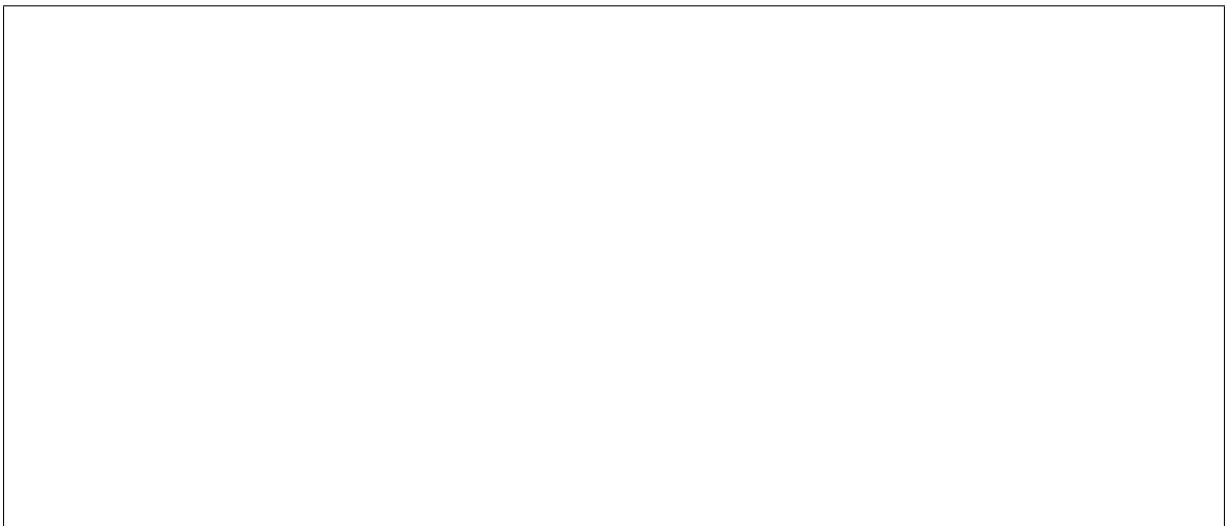
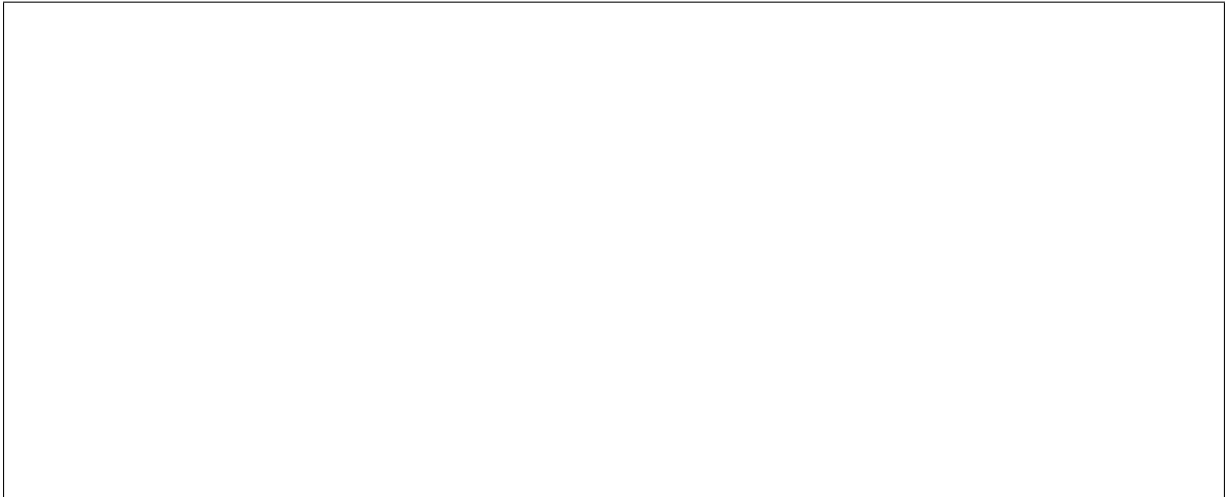
(continued from previous page)



(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**

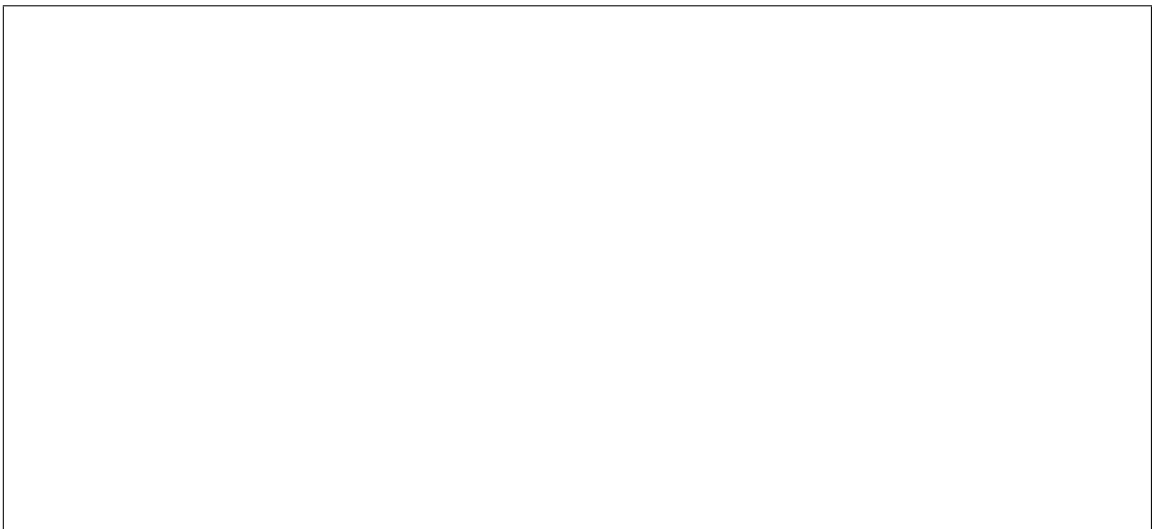
**Note:** Unlike most of the other power interfaces, the SNMP power interface does not have a corresponding management interface. The SNMP hardware type uses the `noop` management interface instead.

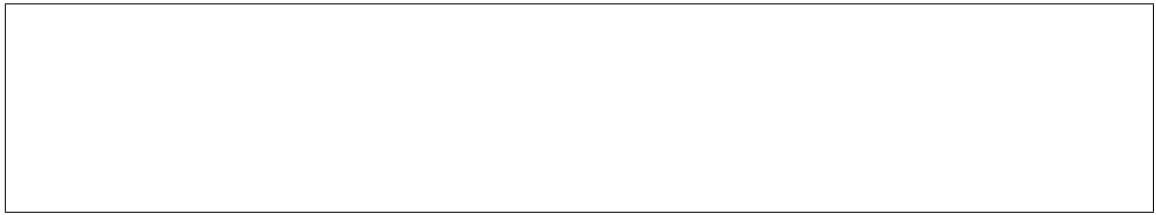
## **List of supported devices**

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

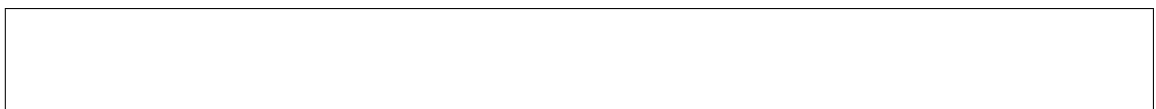
## **Enabling the SNMP Hardware Type**







below:



(continues on next page)



(continued from previous page)

--

## **IroniC Node Configuration**











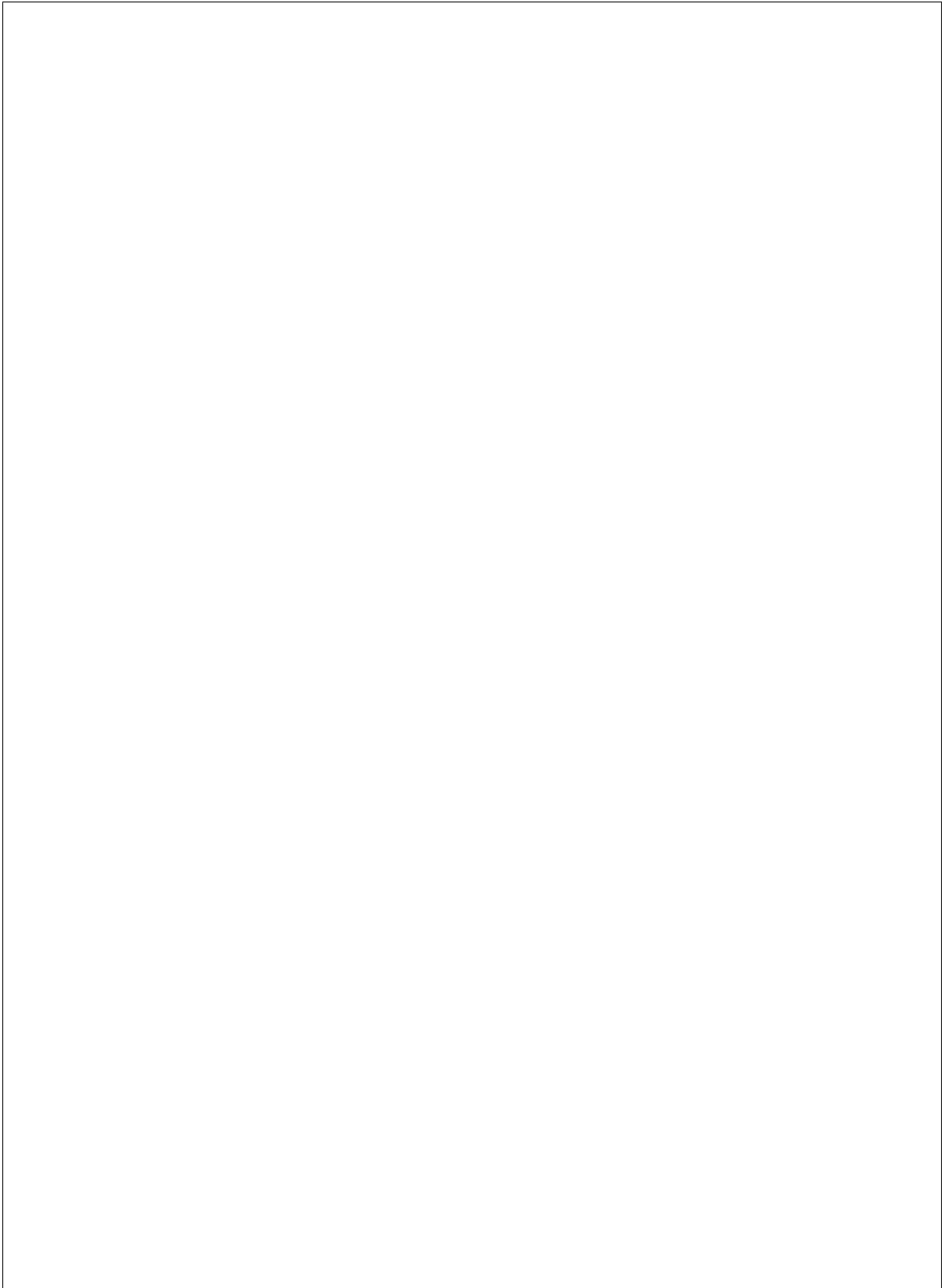
provided. In the latter case `md5` is the default.





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)

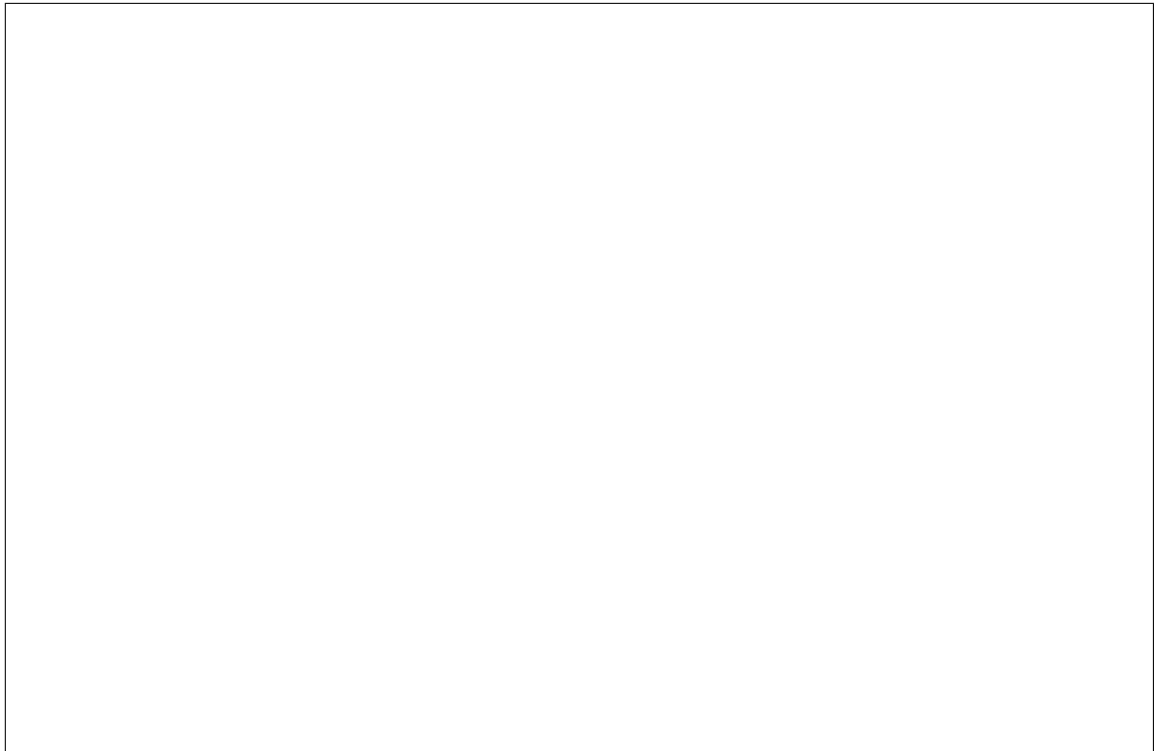
## **XClarity driver**

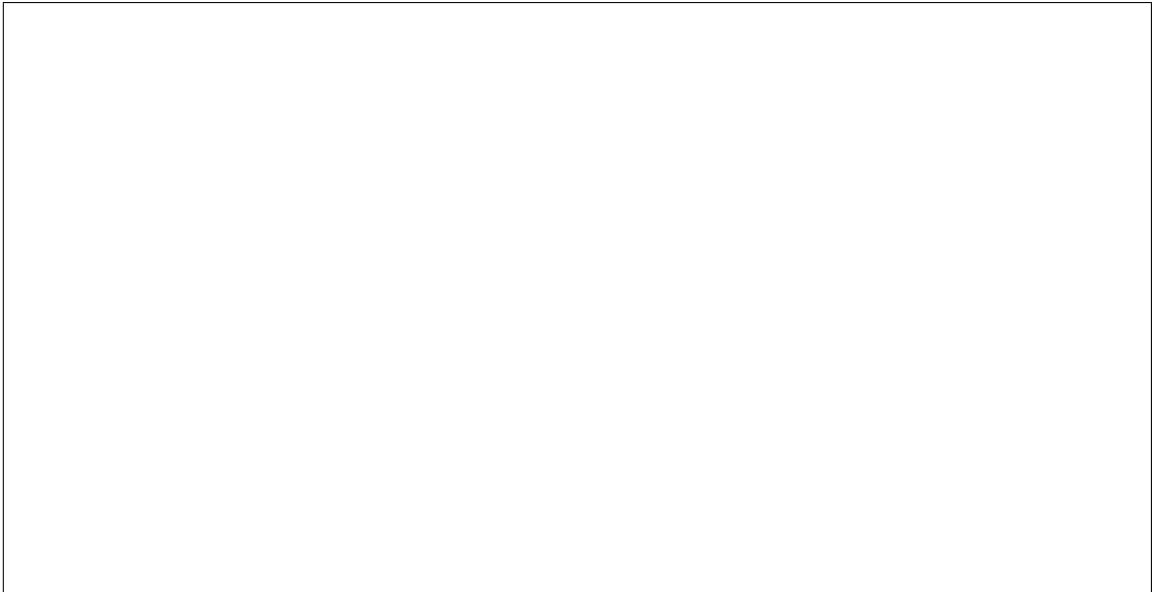
### **Overview**

## **Prerequisites**



## **Enabling the XClarity driver**





**Registering a node with the XClarity driver**







(continues on next page)

(continued from previous page)



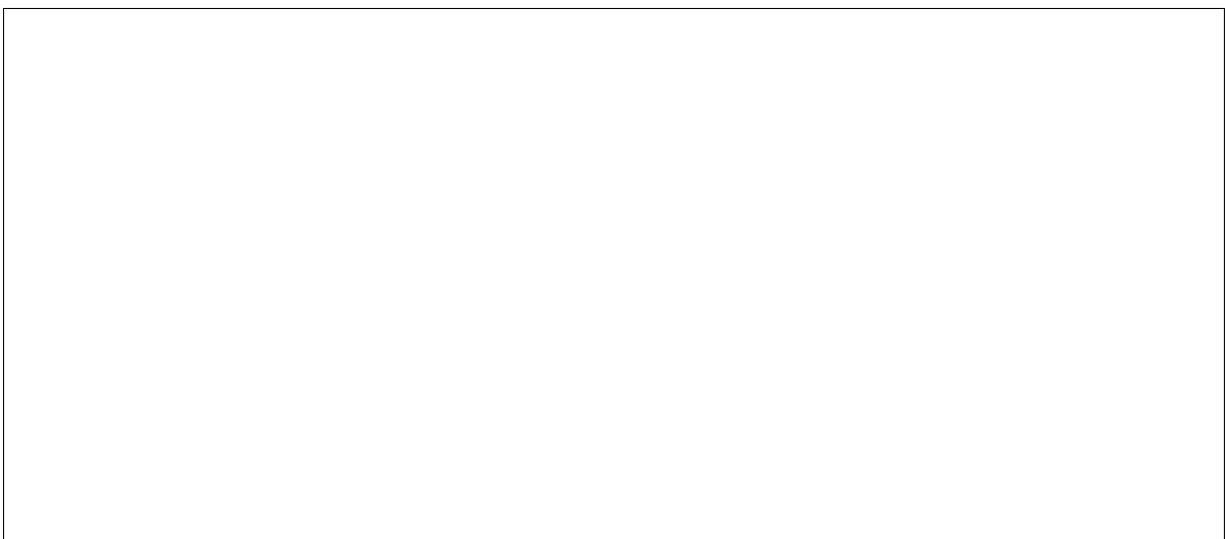


(continues on next page)

(continued from previous page)

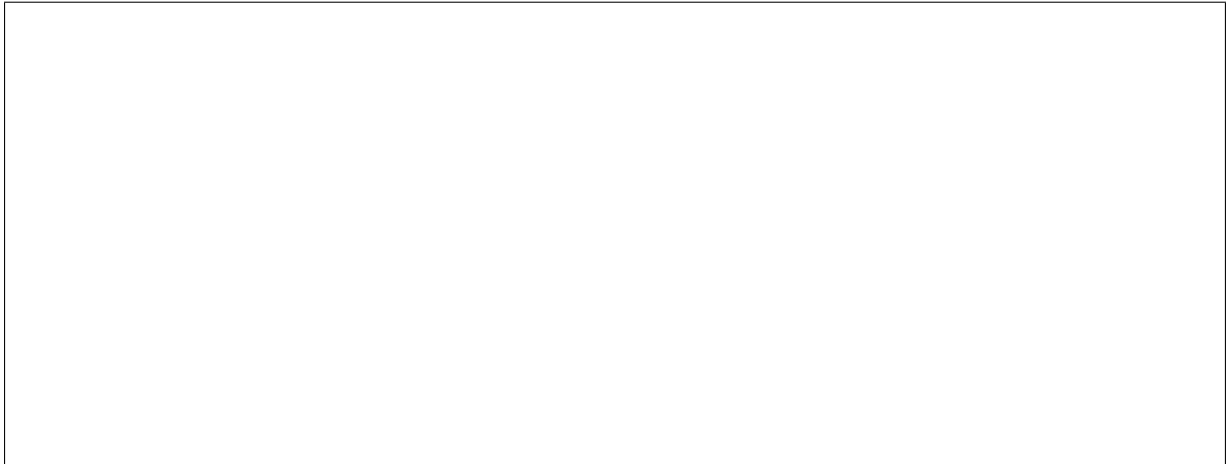


## **Changing Hardware Interfaces**



(continues on next page)

(continued from previous page)

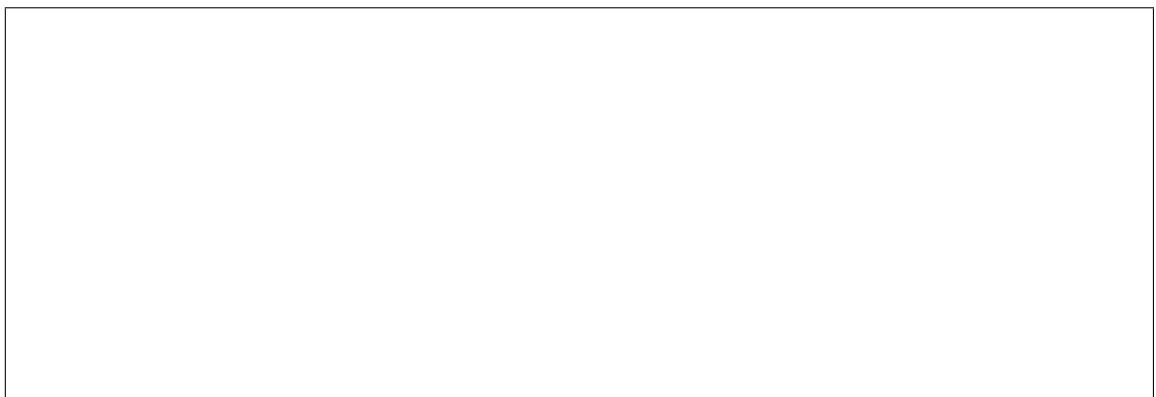


## **Changing Hardware Type**

not work:



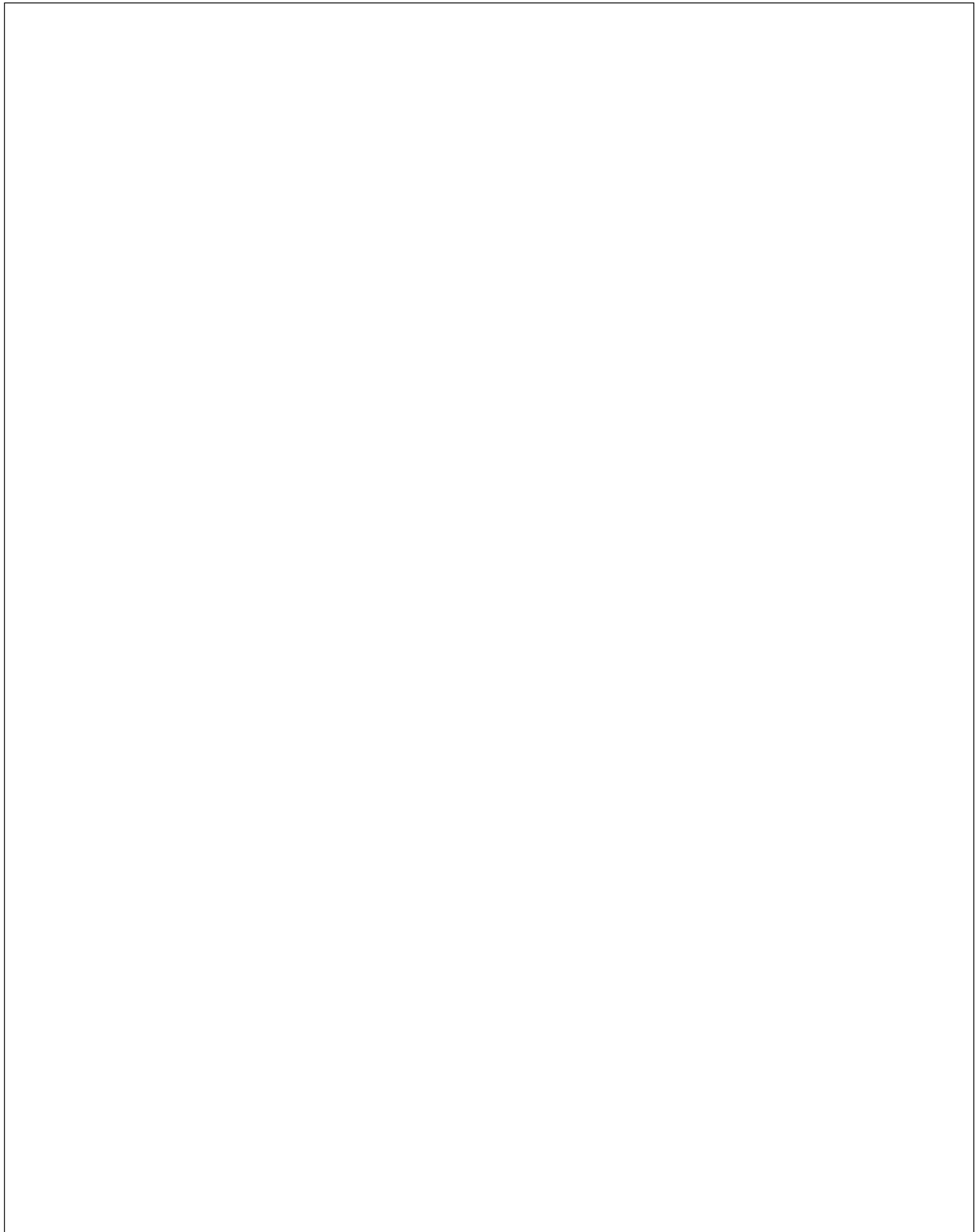




(continues on next page)

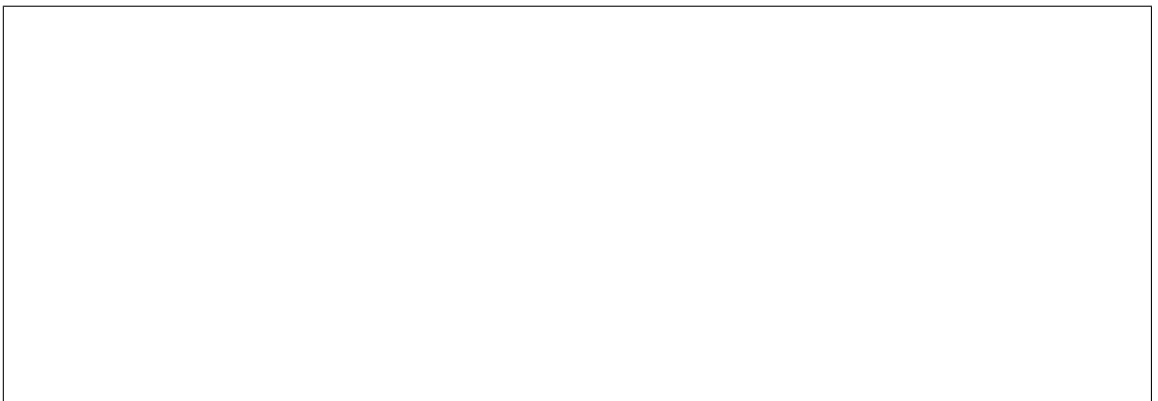
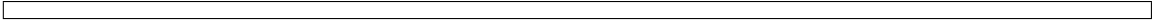
(continued from previous page)

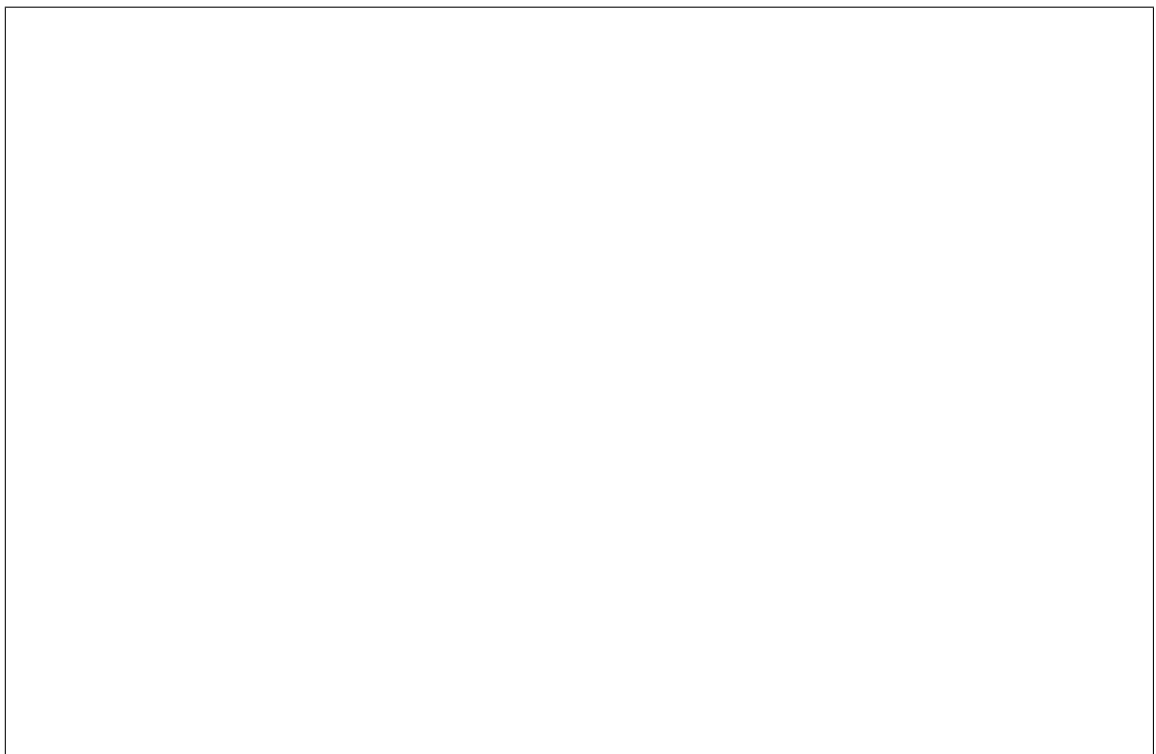




(continues on next page)

(continued from previous page)

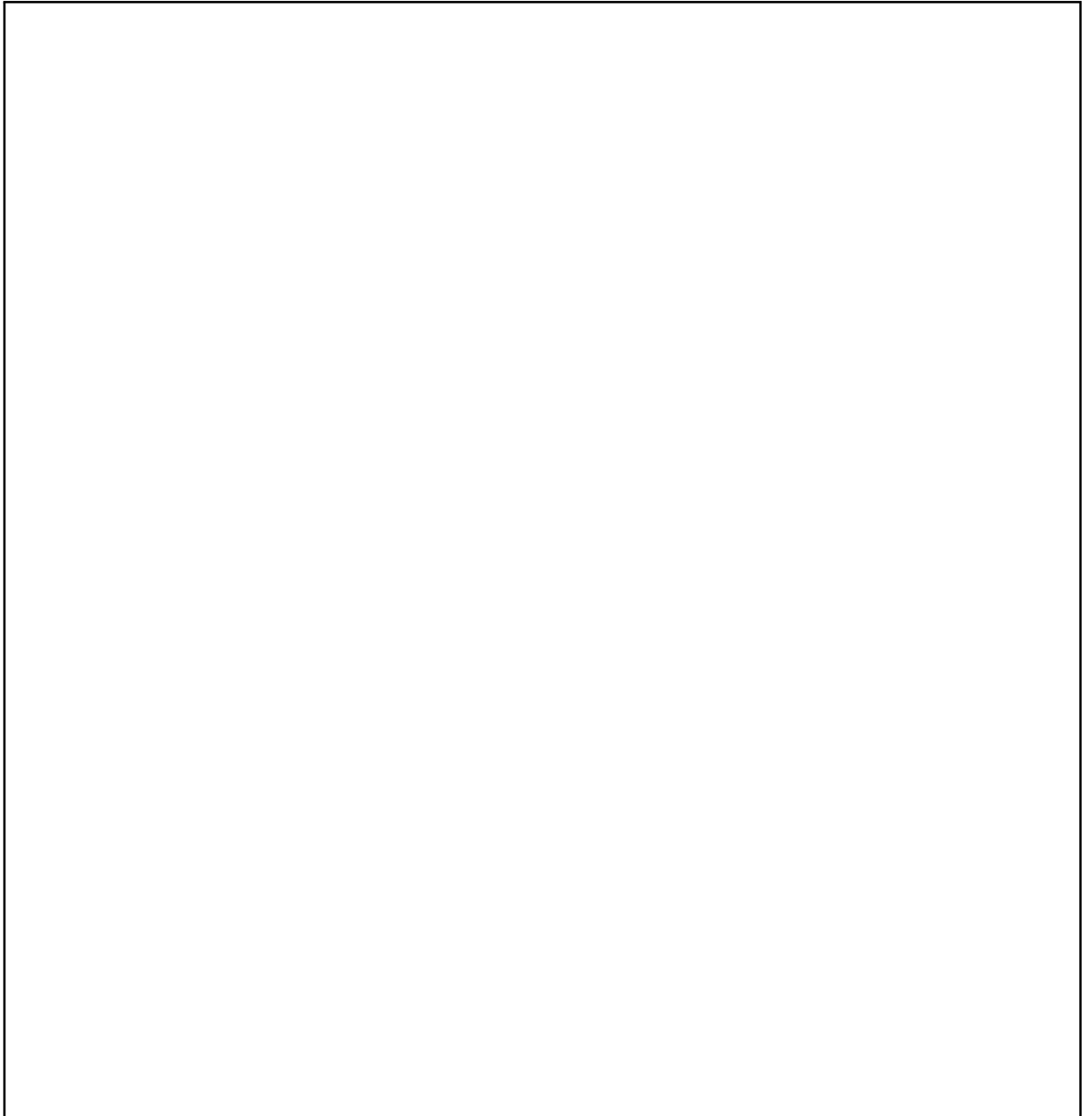


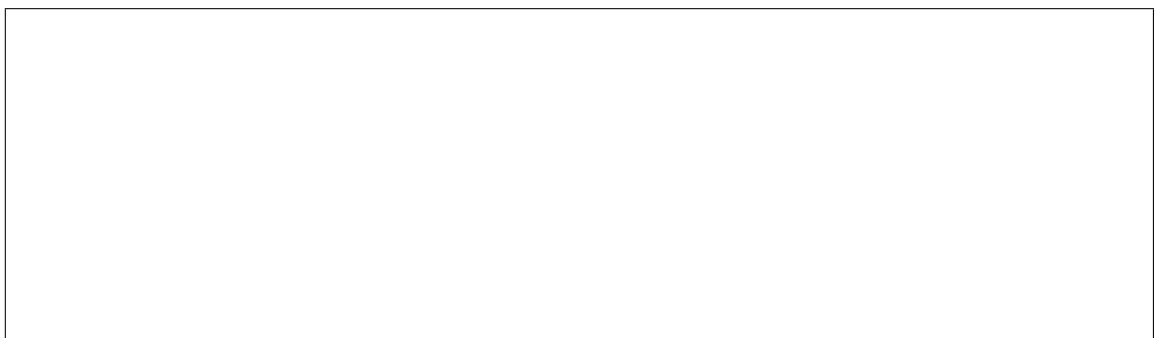
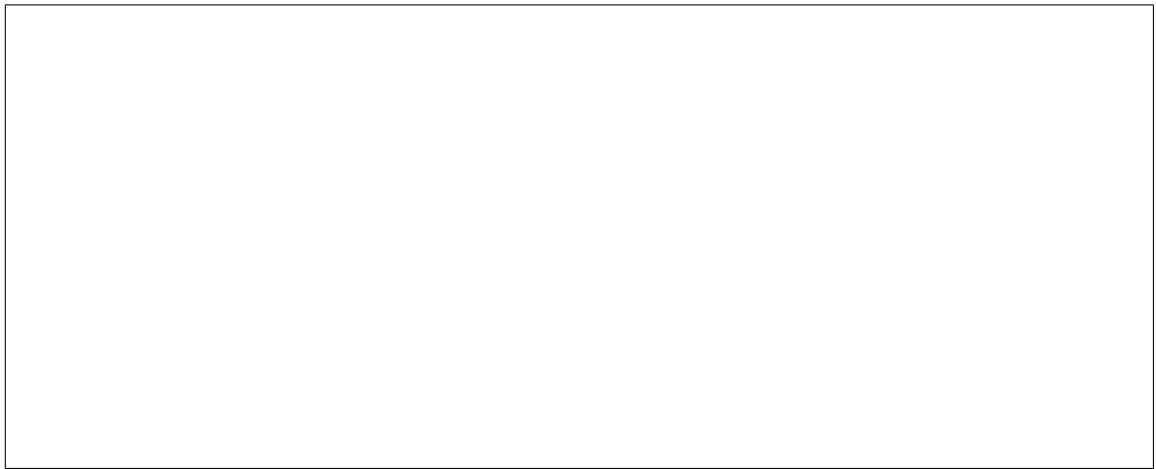


## **Static boot order configuration**

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)

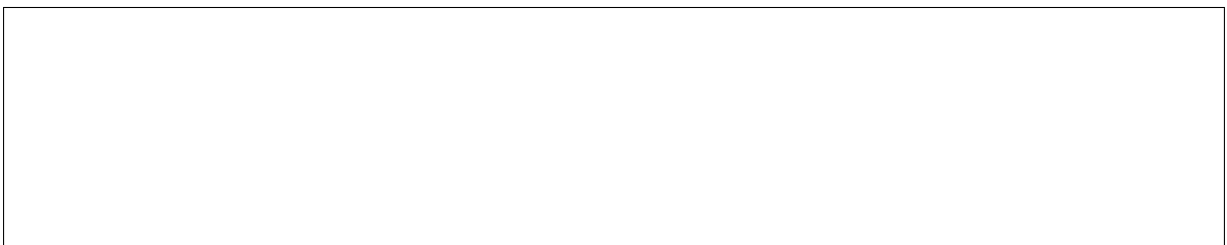
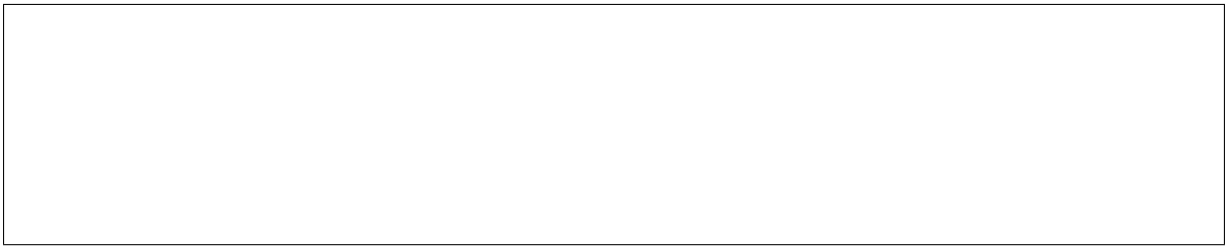
(continued from previous page)

--

## 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](#).



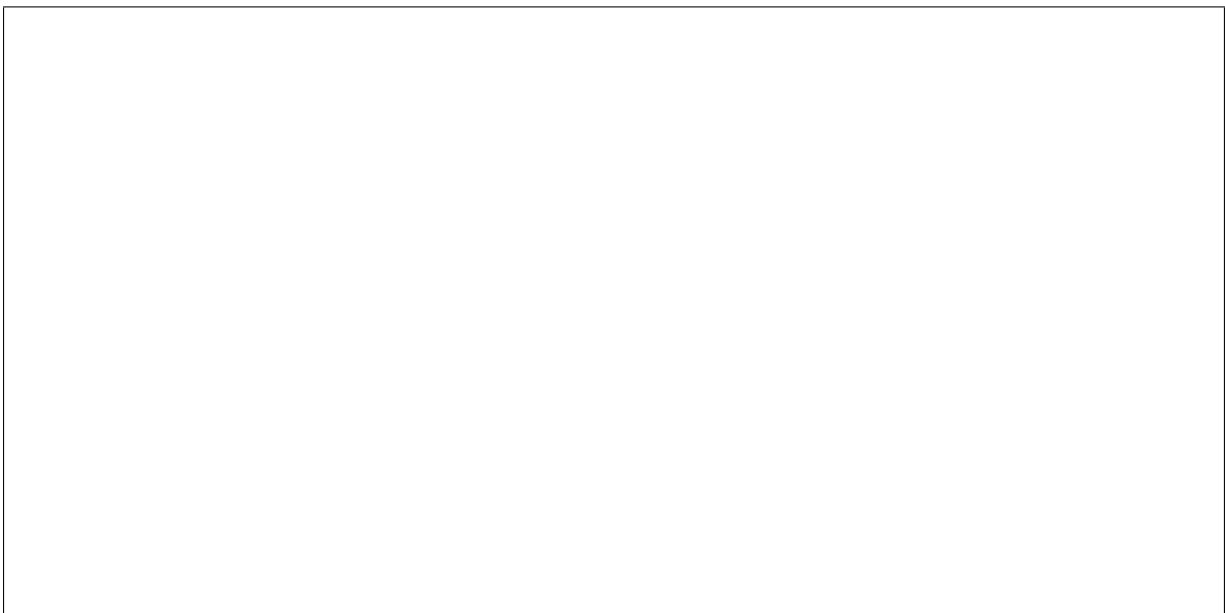


## **Capabilities discovery**



*Support.*





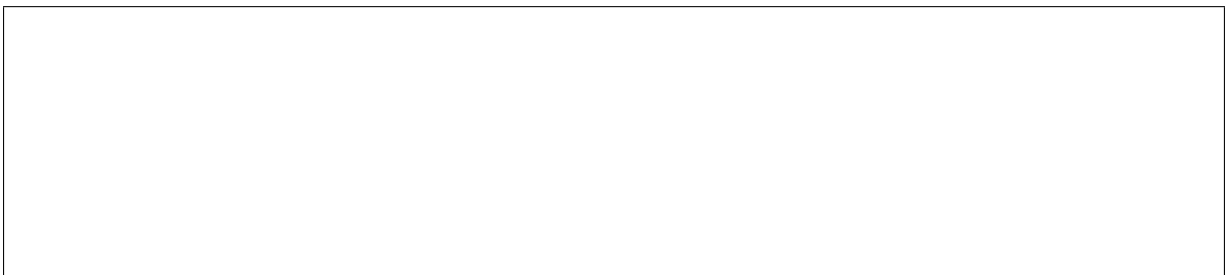
(continues on next page)

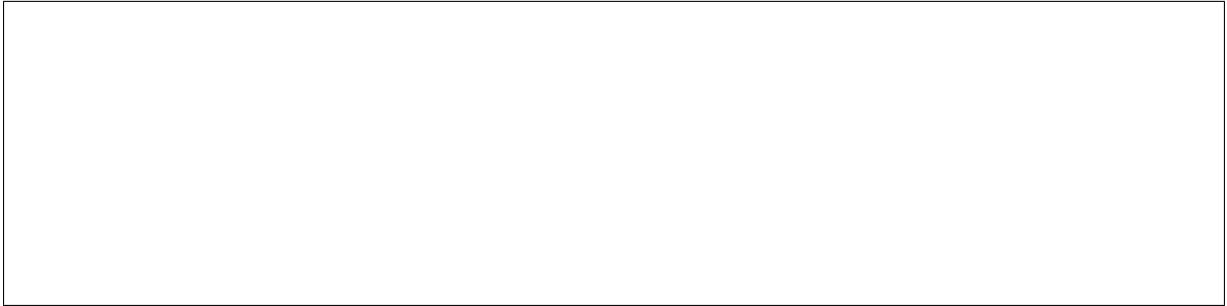
(continued from previous page)

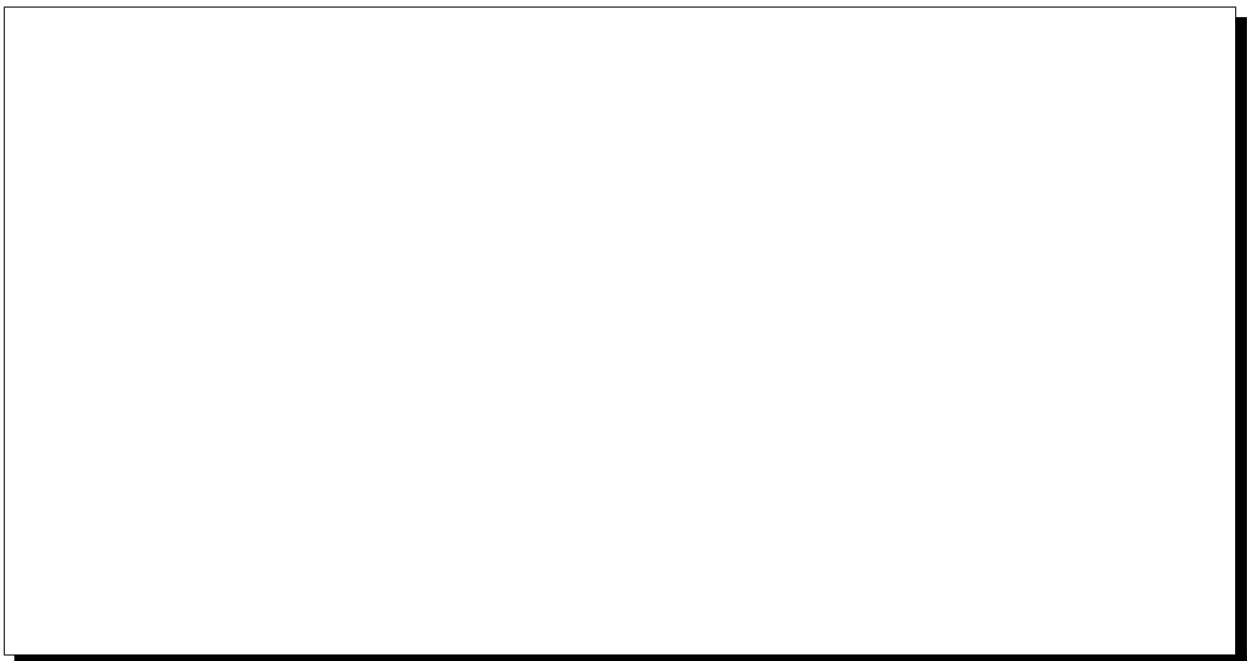
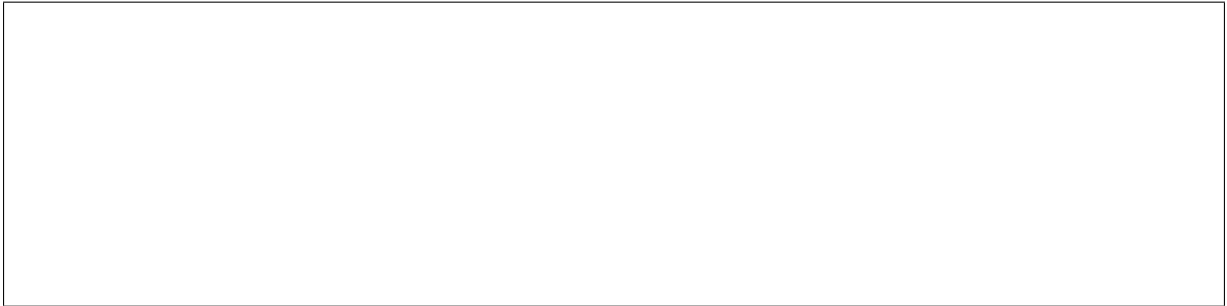
--

**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**













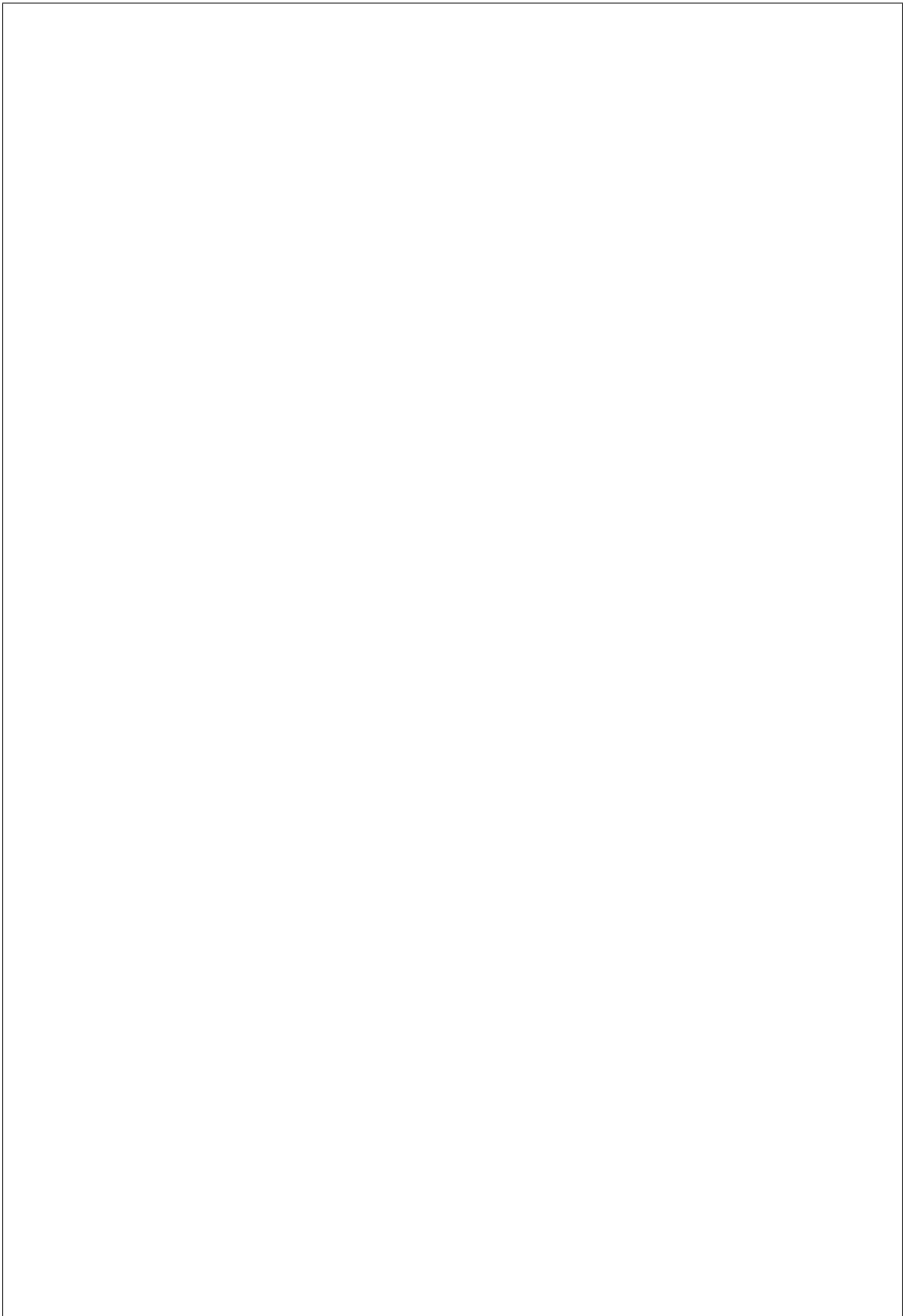


## **In-band steps**

## **Requesting steps**







(continues on next page)

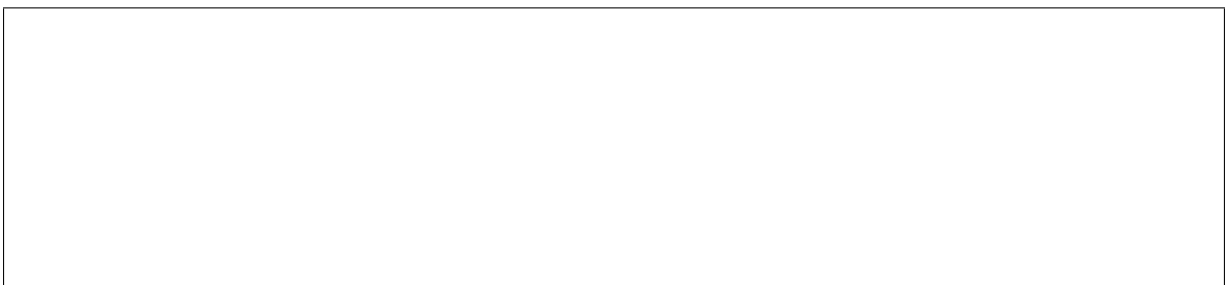
(continued from previous page)

## **Writing a Deploy Step**



## **FAQ**

**What deploy step is running?**



(continues on next page)

(continued from previous page)

--

## **Troubleshooting**



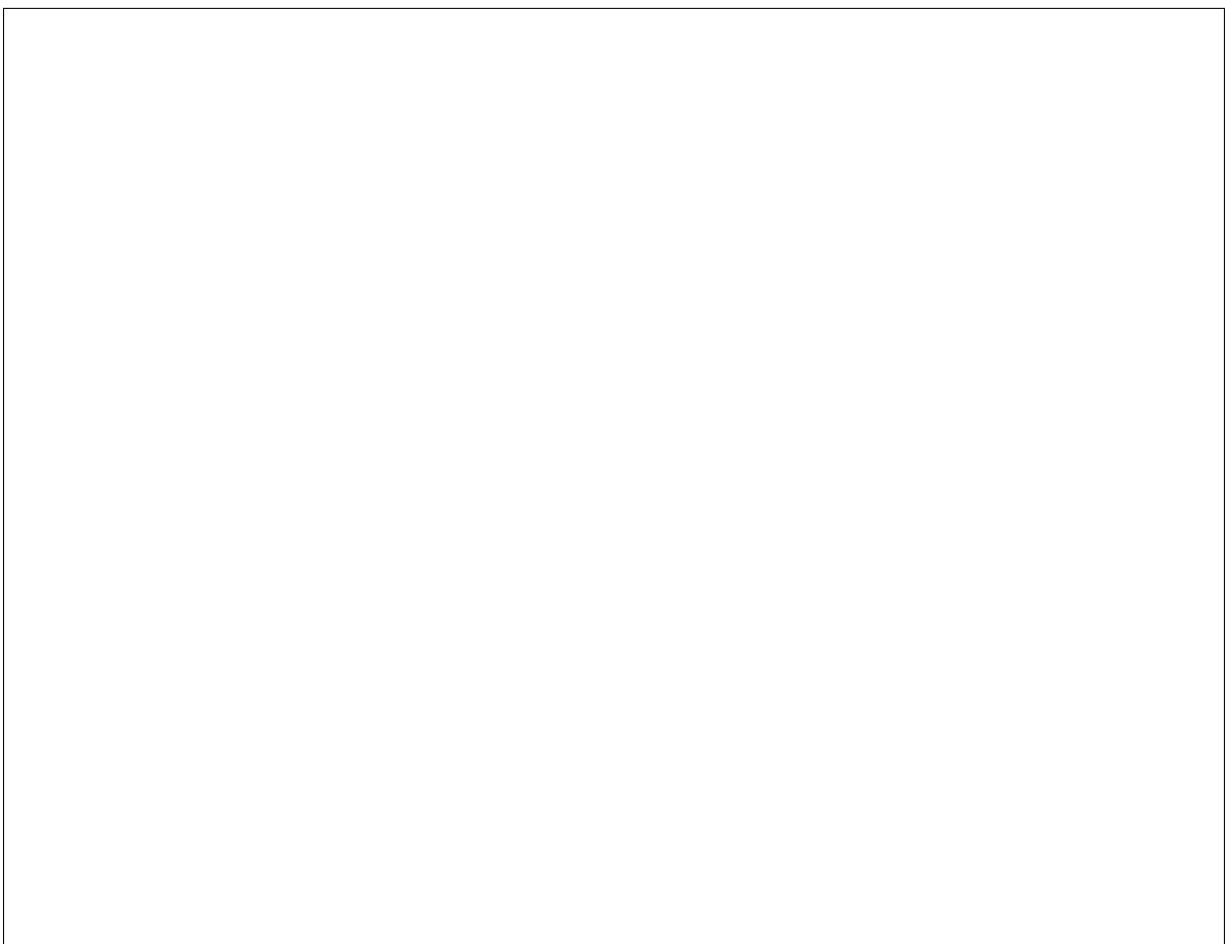
node.

## **Deploy Templates**





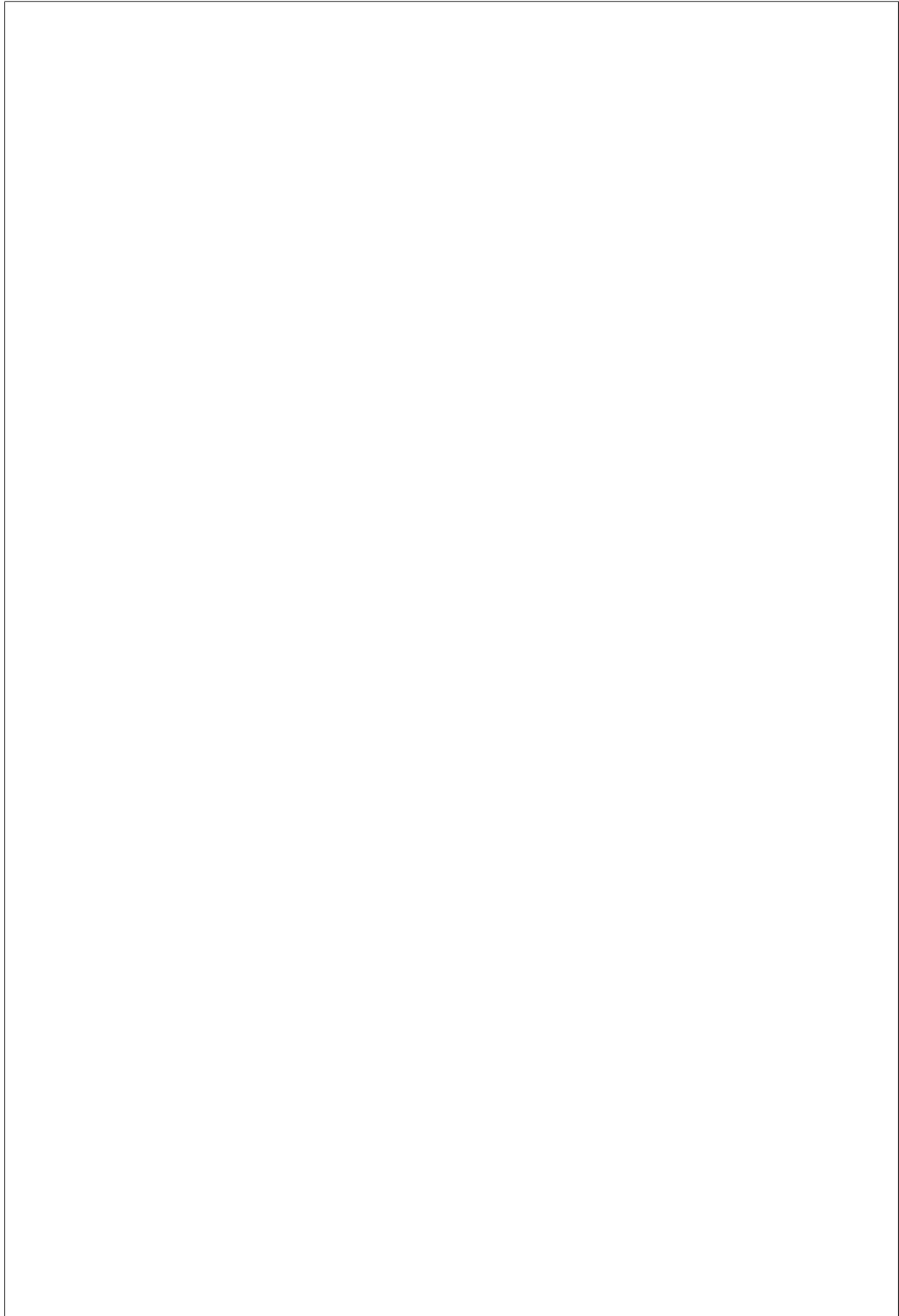
## **Deploy step format**



(continues on next page)



(continued from previous page)



(continues on next page)

(continued from previous page)



## **Matching deploy templates**



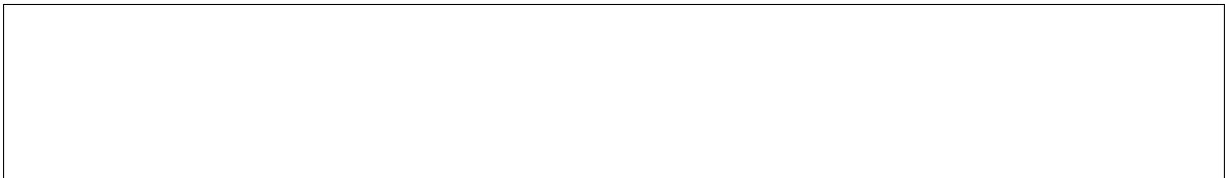
scheduling when the Bare Metal service is used with the Compute service.

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.

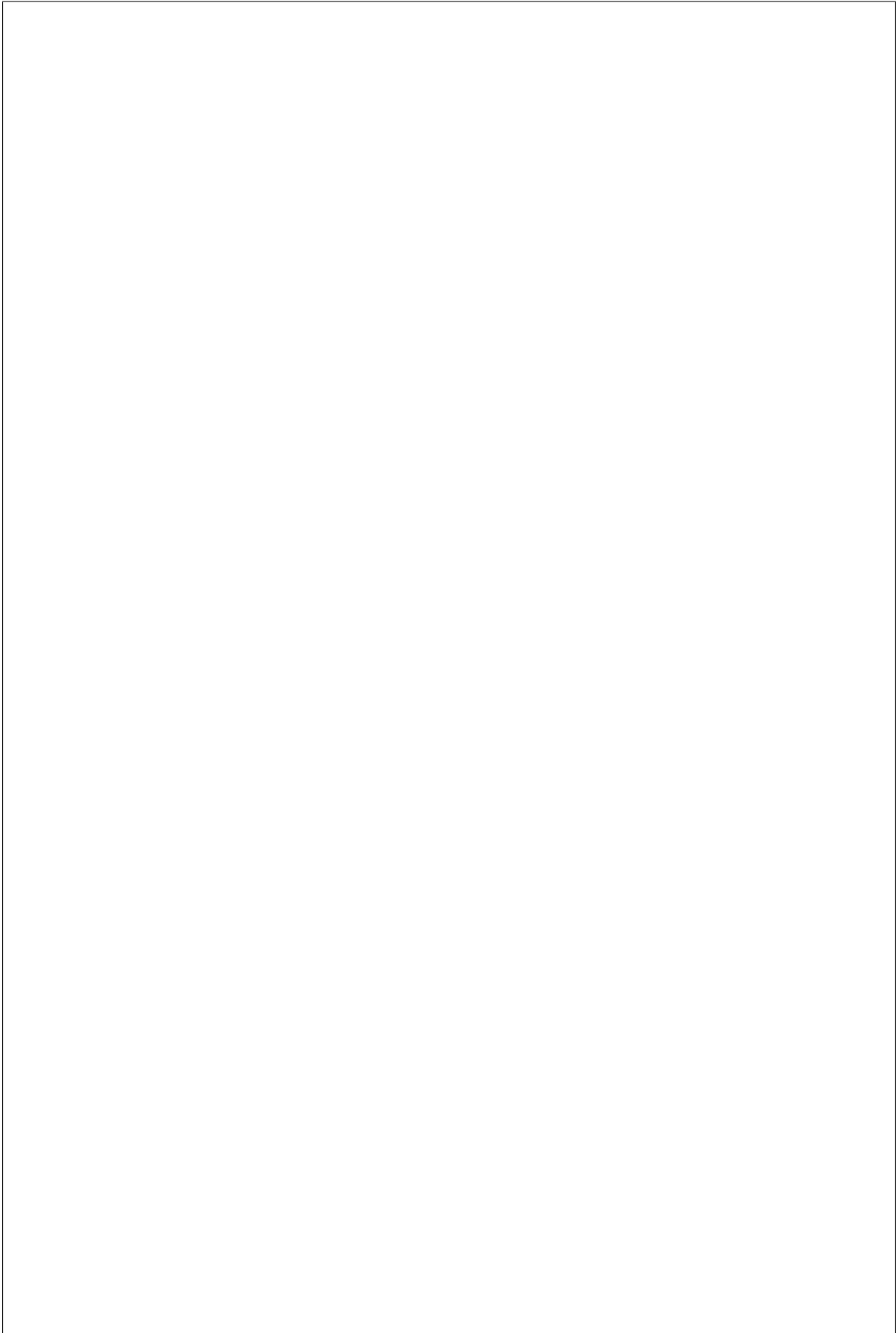
### **Overriding default deploy steps**

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**







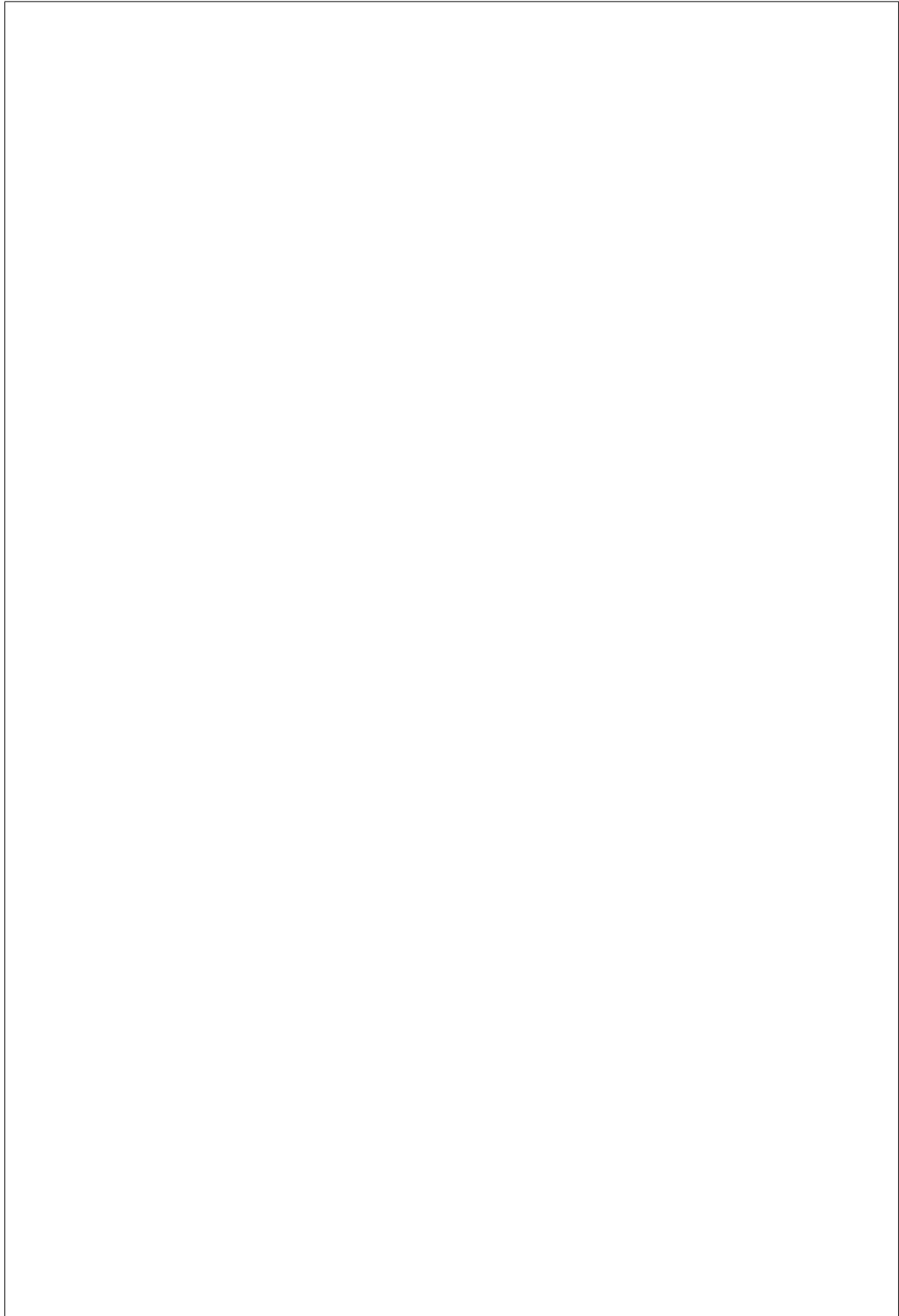
(continues on next page)

(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)



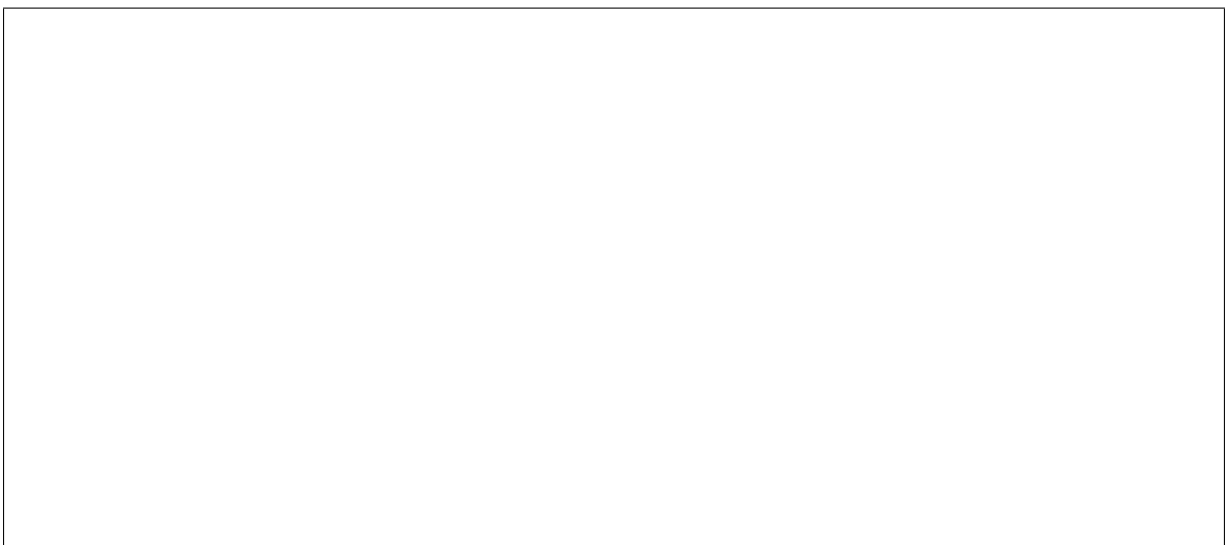
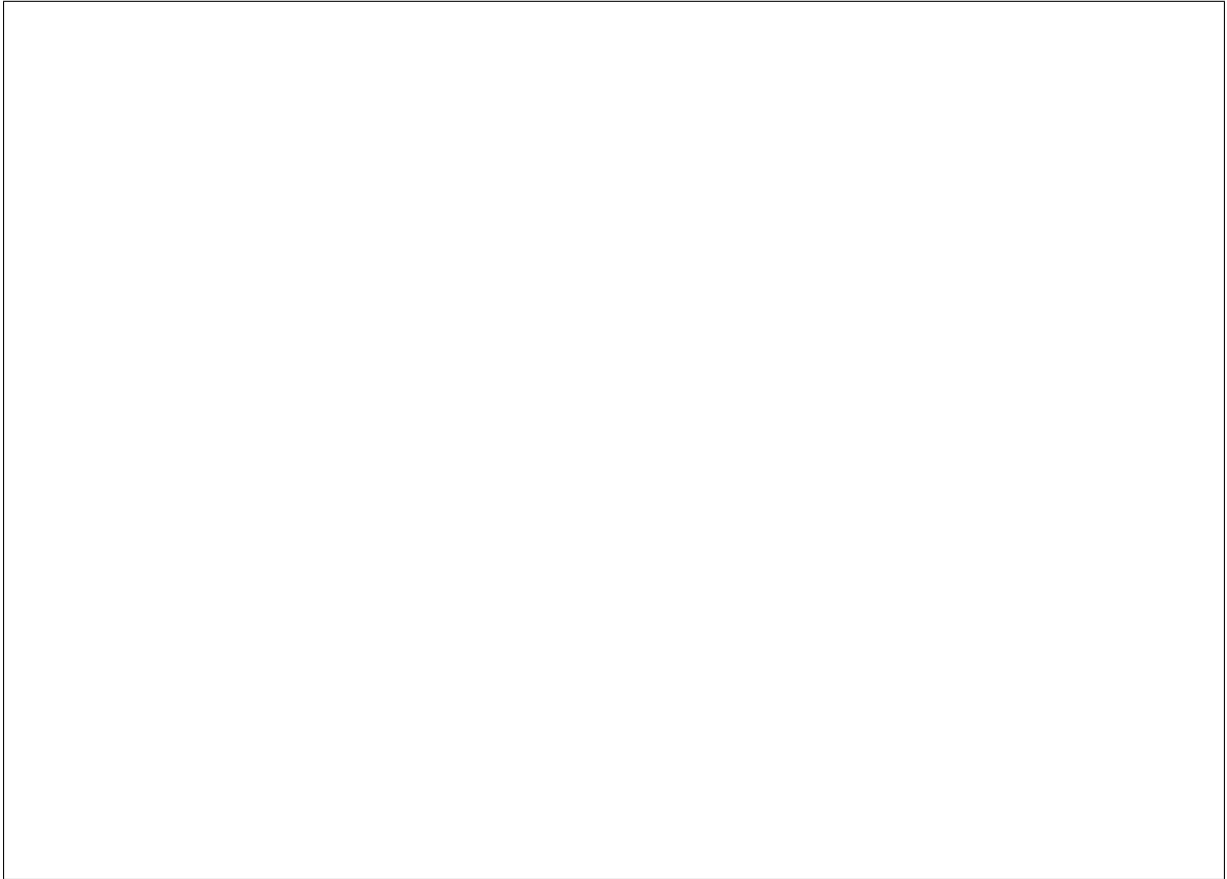
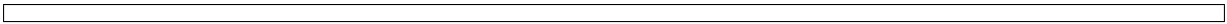
**Creating a deploy template via baremetal client**





(continues on next page)

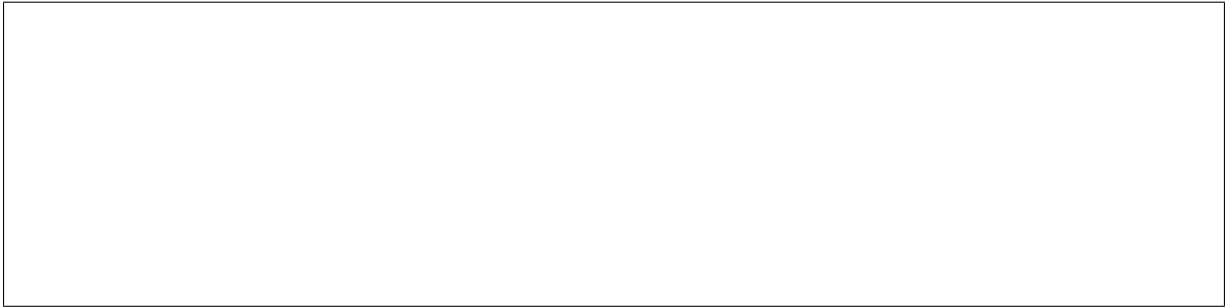
(continued from previous page)



(continues on next page)



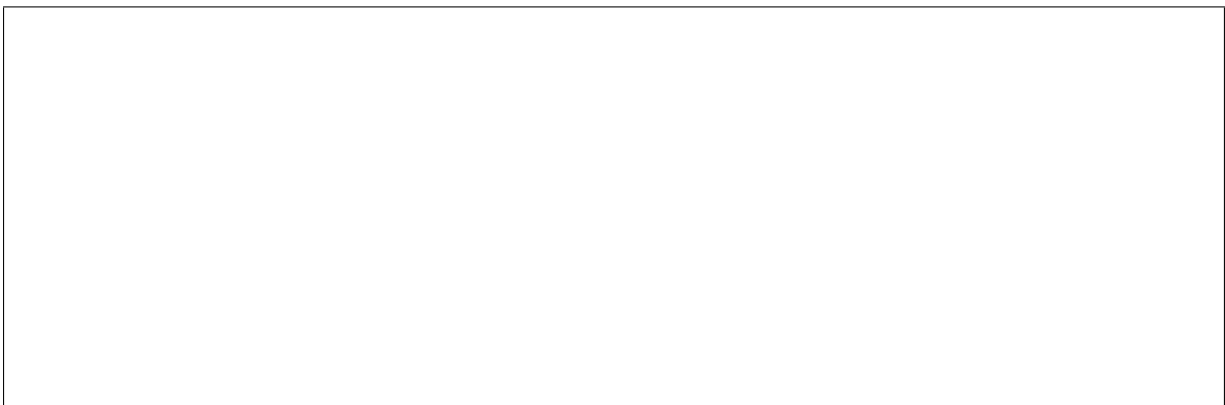
(continued from previous page)

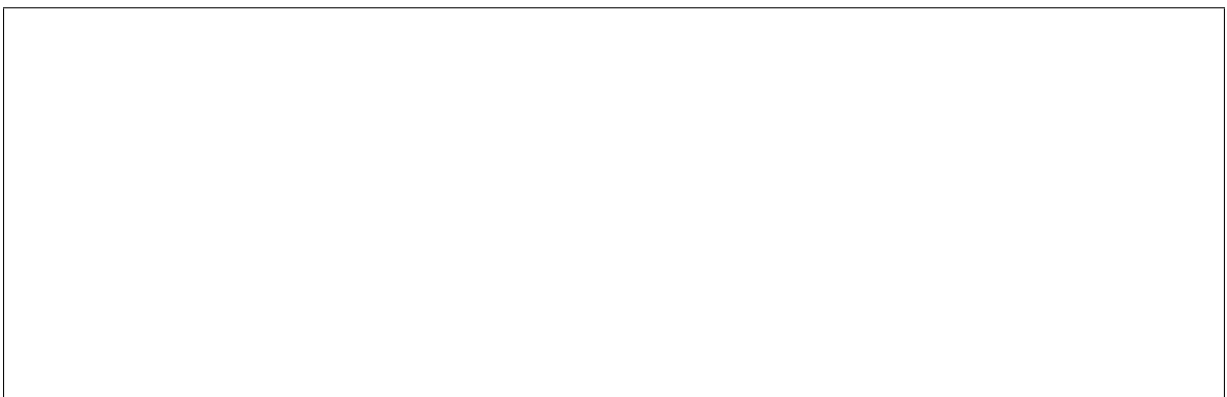


### **Example of use with the Compute service**

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







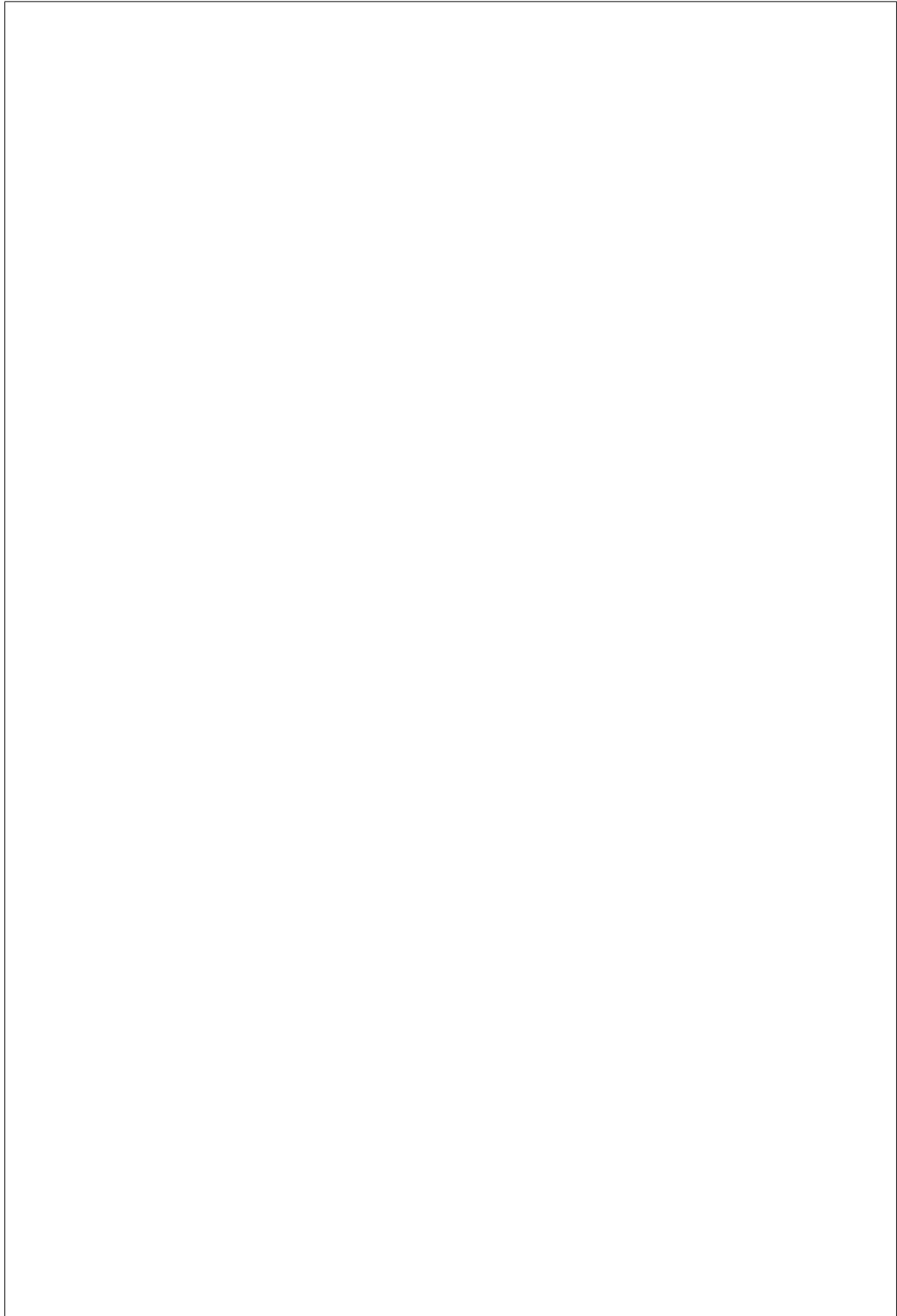
(continues on next page)

(continued from previous page)



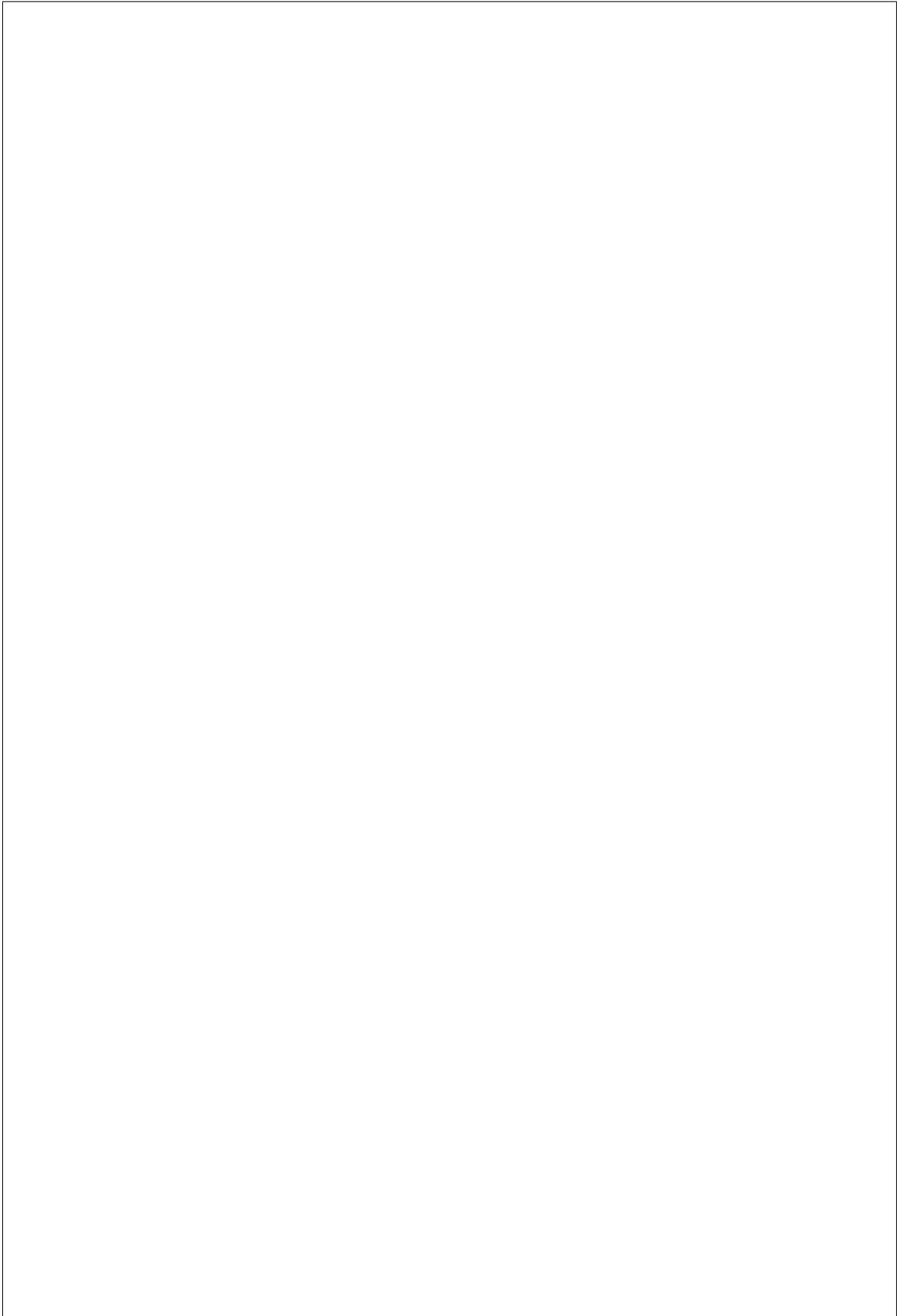
(continues on next page)

(continued from previous page)



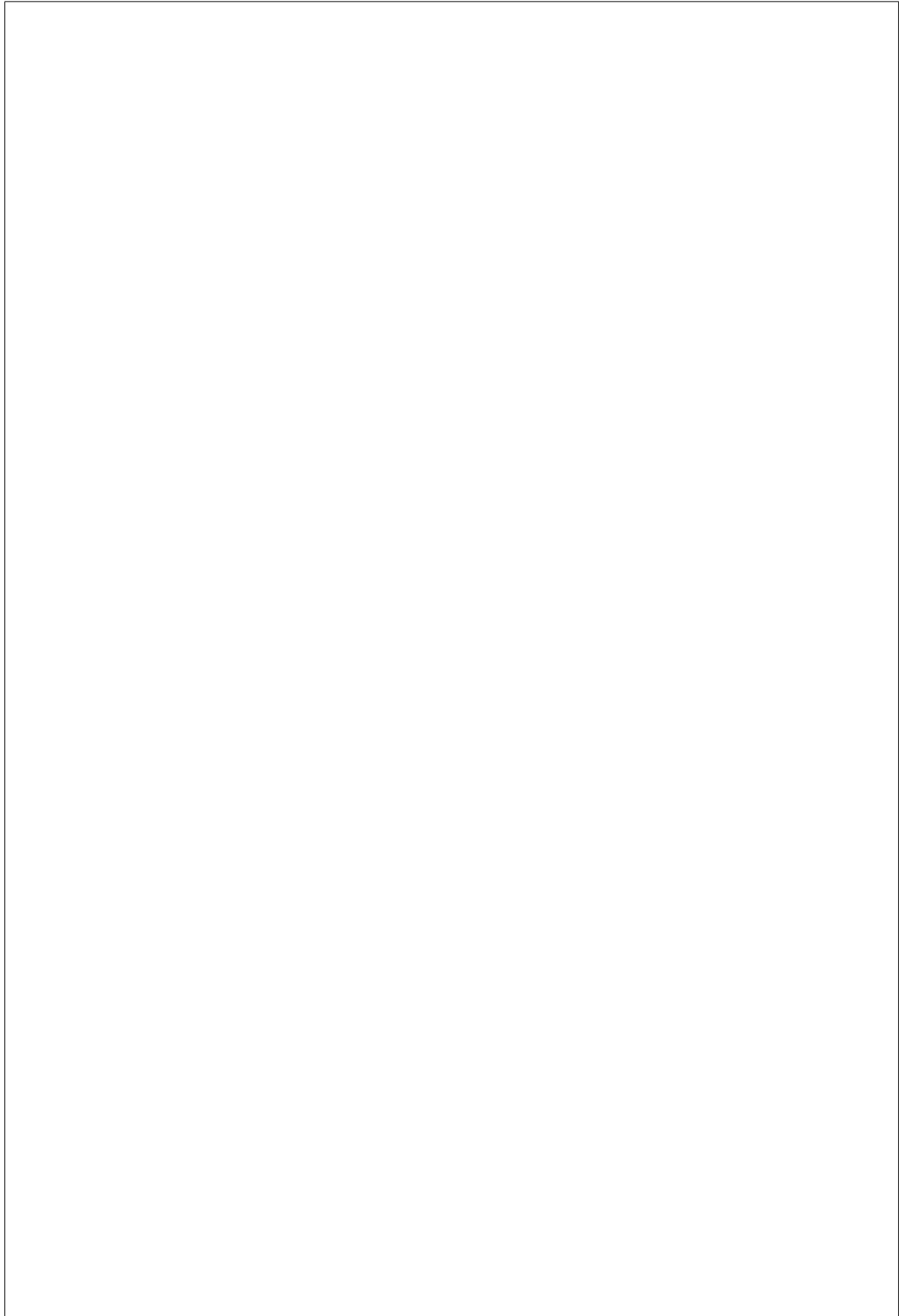
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

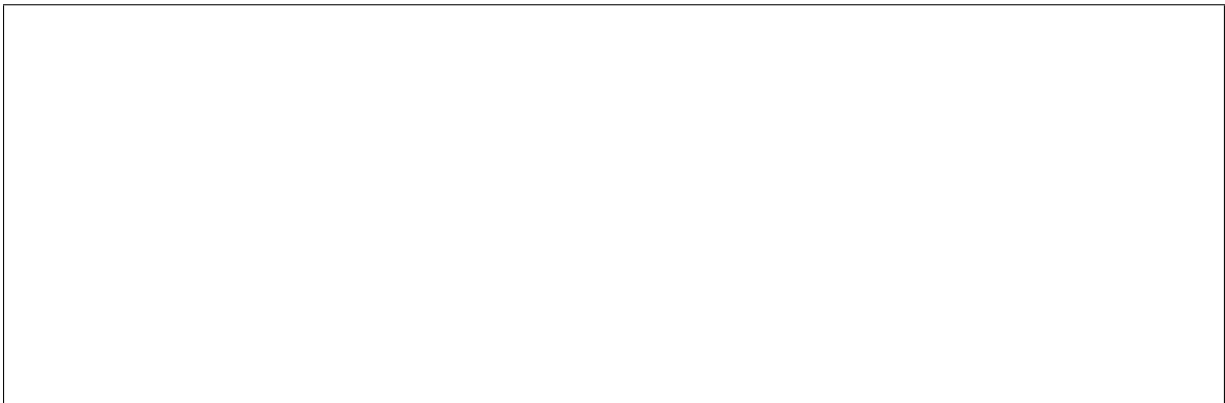
(continued from previous page)

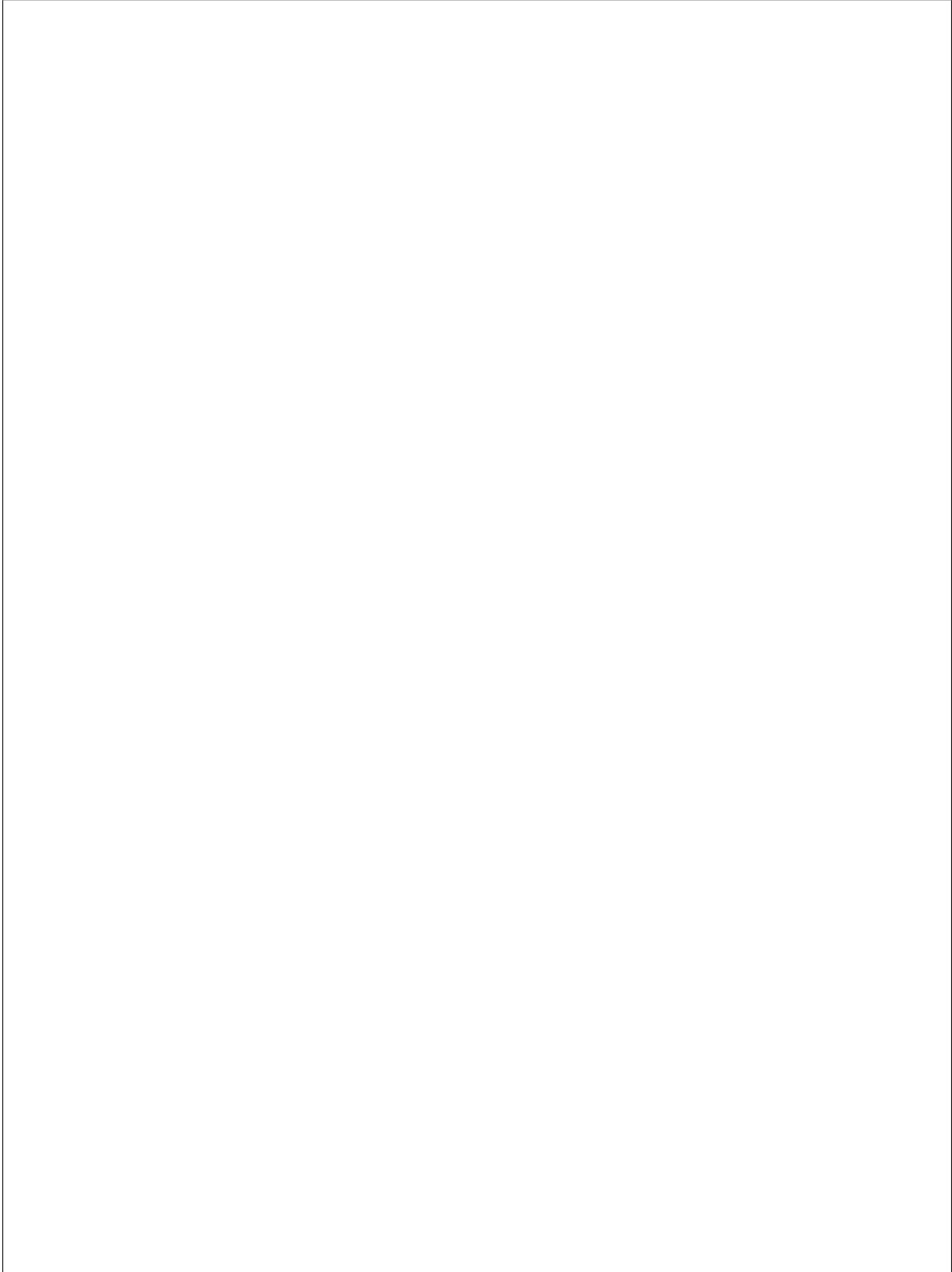




configuration.







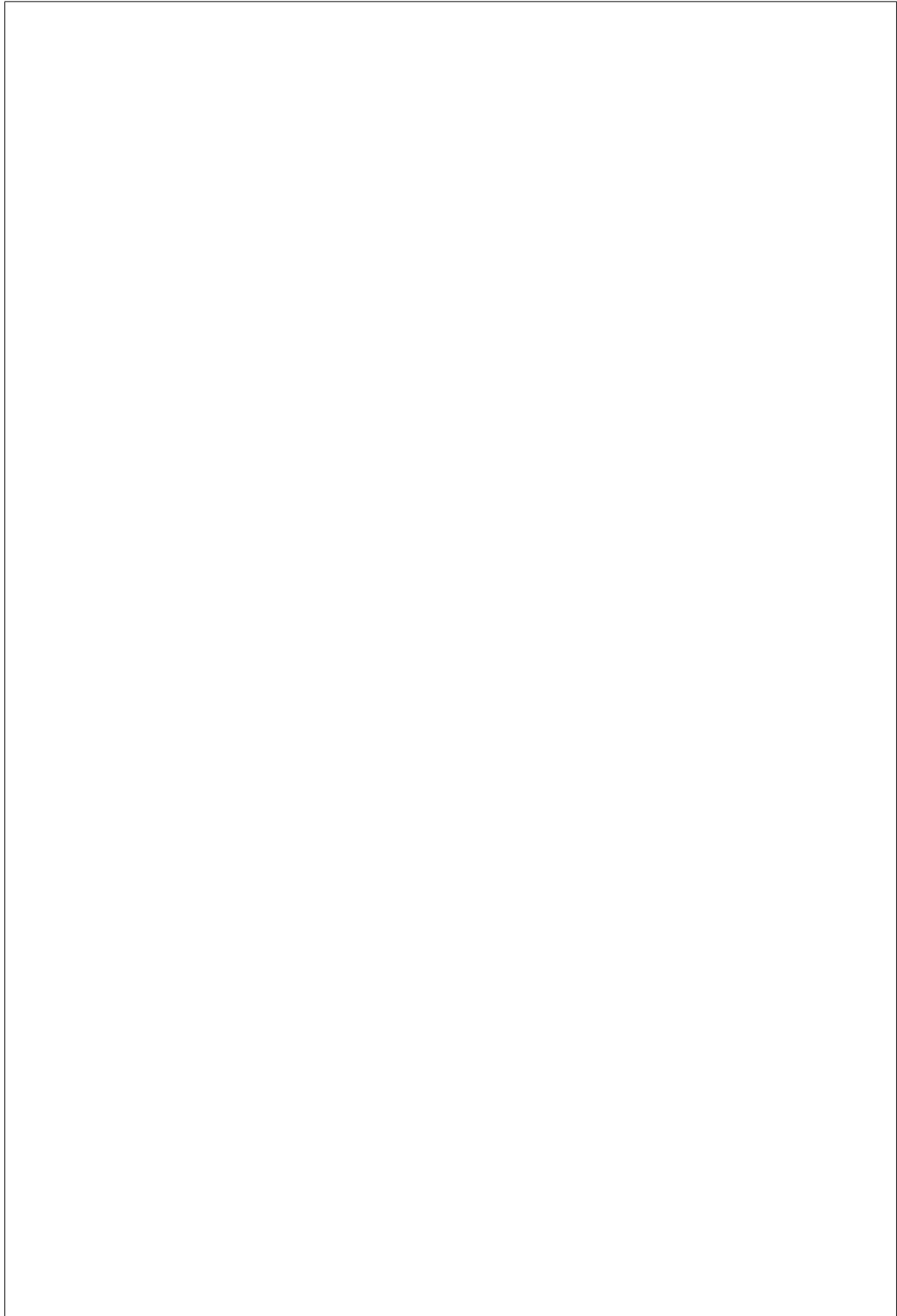
(continues on next page)

(continued from previous page)



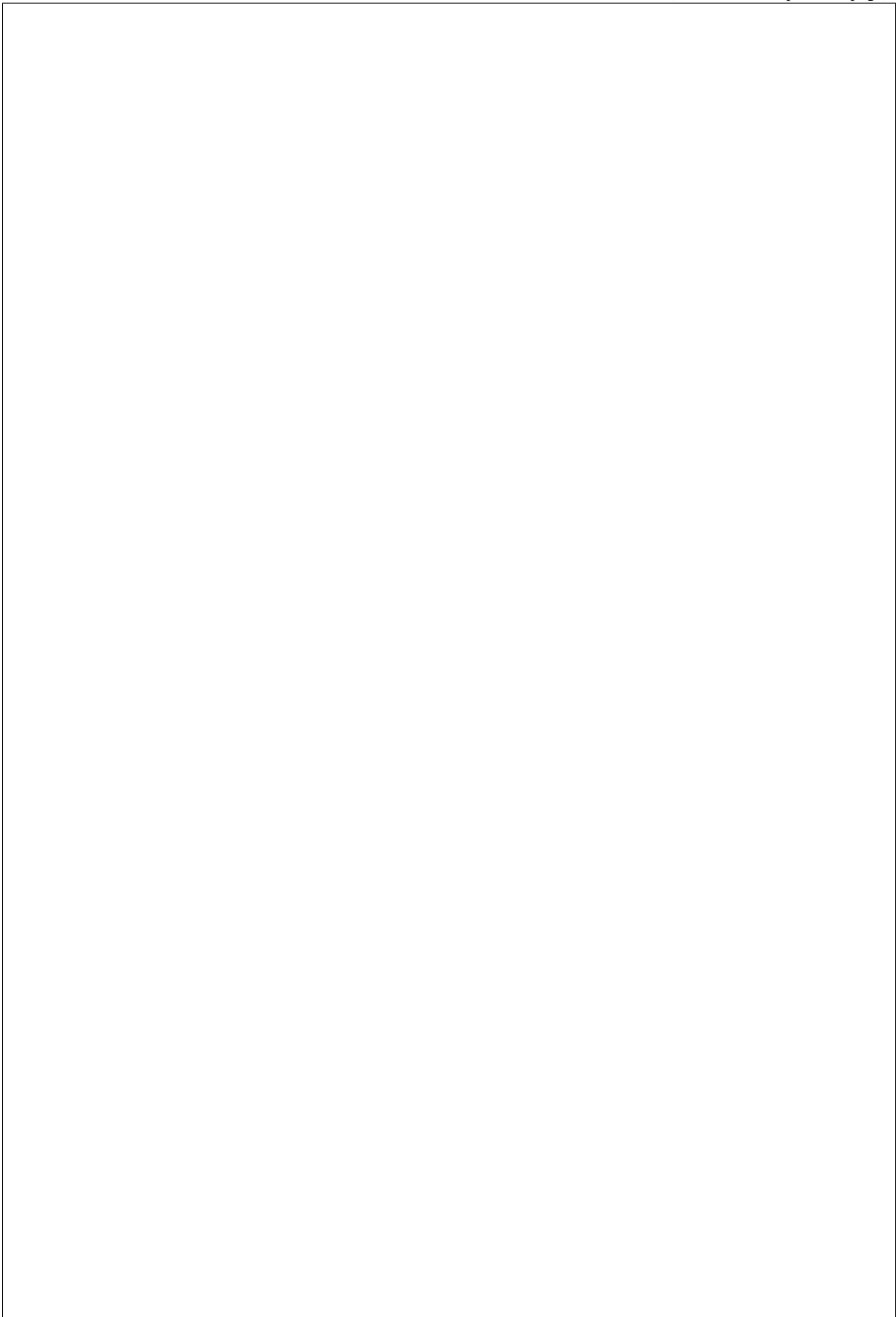
(continues on next page)

(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	Prior-ity	Stop-pable	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_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>re-quired</i> ) A list of firmware images to apply.

Name	Details	Prior-ity	Stop-pable	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
activate_license	Activates iLO Advanced license.	0	no	ilo_license_key ( <i>required</i> ) The HPE iLO Advanced license key to activate enterprise features.
clear_secure_boot	Clear all secure boot keys. Clears all the secure boot keys. This operation is supported only on HP Proliant Gen9 and above servers.	0	no	
reset_bios_to_defaults	Resets the BIOS settings to default values. Resets BIOS to default settings. This operation is currently supported only on HP Proliant Gen9 and above servers.	10	no	
reset_ilo	Resets the iLO.	0	no	
reset_ilo_credentials	Resets the iLO password.	30	no	
reset_secure_boot	Reset secure boot keys to manufacturing defaults. Resets the secure boot keys to manufacturing defaults. This operation is supported only on HP Proliant Gen9 and above servers.	20	no	
update_firmware	Updates the firmware.	0	no	firmware_images ( <i>required</i> ) This argument represents the ordered list of JSON dictionaries of firmware images. Each firmware image dictionary consists of three mandatory fields, namely url, checksum and component. These fields represent firmware image location URL, md5 checksum of image file and firmware component type respectively. The supported firmware URL schemes are file, http, https and swift. The supported values for firmware component are ilo, cpld, power_pic, bios and chassis. The firmware images will be applied (in the order given) one by one on the baremetal server. For more information, see <a href="https://docs.openstack.org/ironic/latest/admin/drivers/ilo.html#initiating-firmware-update-as-manual-clean">https://docs.openstack.org/ironic/latest/admin/drivers/ilo.html#initiating-firmware-update-as-manual-clean</a>

## 5.2. Administrators Guide

Name	Details	Prior-ity	Stop-pable	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	Chapter 5. Administrator Guide firmware_images ( <i>required</i> ) This argument represents the ordered list of JSON dictionaries of

Name	Details	Prior-ity	Stop-pable	Arguments
restore_ironic_bios	Restore BIOS config for a node.	0	no	

Name	Details	Prior-ity	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	Prior-ity	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	Prior-ity	Stop-pable	Arguments
apply_configuration	<p>Apply the BIOS configuration to the node</p> <p><b>param task</b> a TaskManager instance containing the node to act on</p> <p><b>param settings</b> List of BIOS settings to apply</p> <p><b>raises</b> DRA-COperationError 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	Prior-ity	Stop-pable	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	Prior-ity	Stop-pable	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	Prior-ity	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	

## Raid Interface

Name	Details	Prior-ity	Stop-pable	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	Prior-ity	Stop-pable	Arguments
create_configuration	Create a RAID configuration. This method creates a RAID configuration on the given node.	0	no	<div>create_nonroot_volumes</div> This specifies whether to create the non-root volumes. Defaults to <i>True</i> . <div>create_root_volume</div> This specifies whether to create the root volume. Defaults to <i>True</i> . <div>delete_existing</div> 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	Prior-ity	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	Prior-ity	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	Prior-ity	Stop-pable	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	

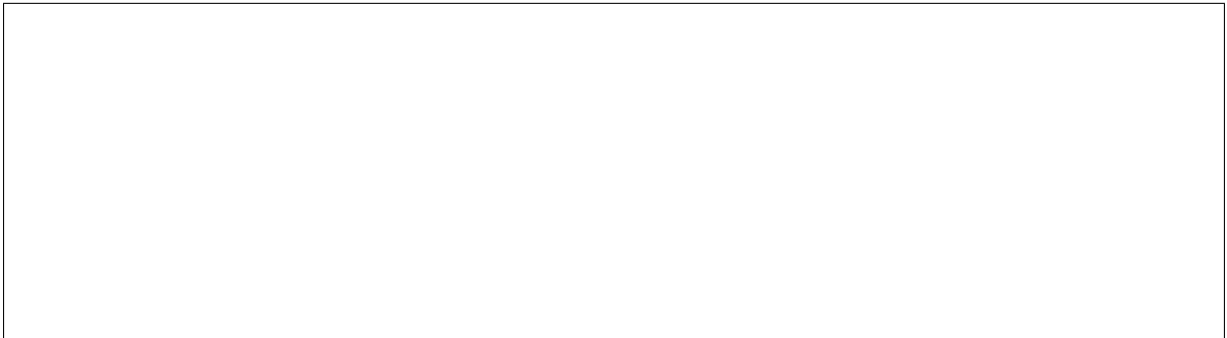
Name	Details	Prior-ity	Stop-pable	Arguments
create_configuration	Create the RAID configuration. This method creates the RAID configuration on the given node.	0	no	create_nonroot_volumes This specifies whether to create the non-root volumes. Defaults to <i>True</i> . create_root_volume This specifies whether to create the root volume. Defaults to <i>True</i> .
delete_configuration	Delete the RAID configuration.	0	no	

## Manual cleaning

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

## **Setup**

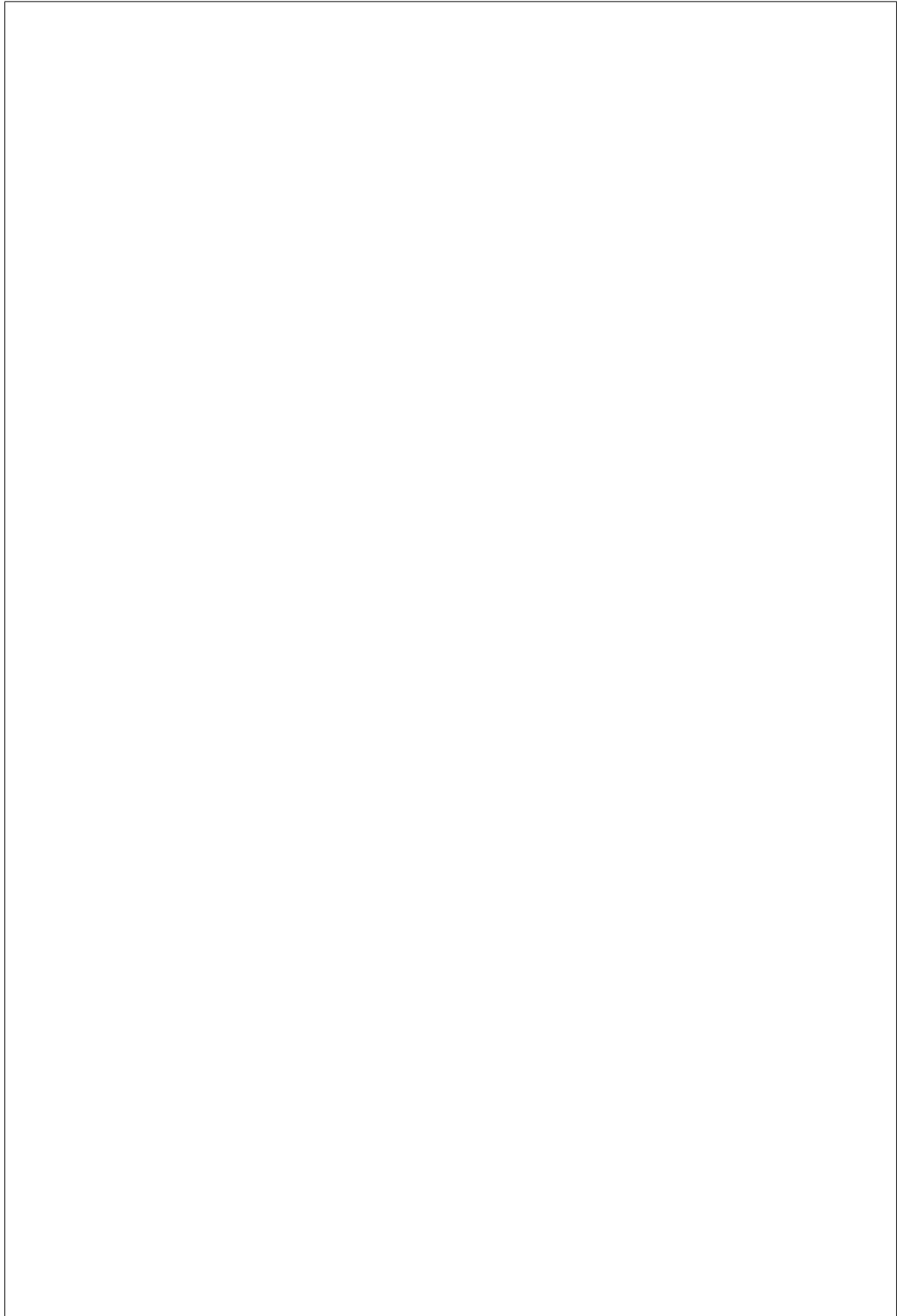
### **Starting manual cleaning via API**



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

(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

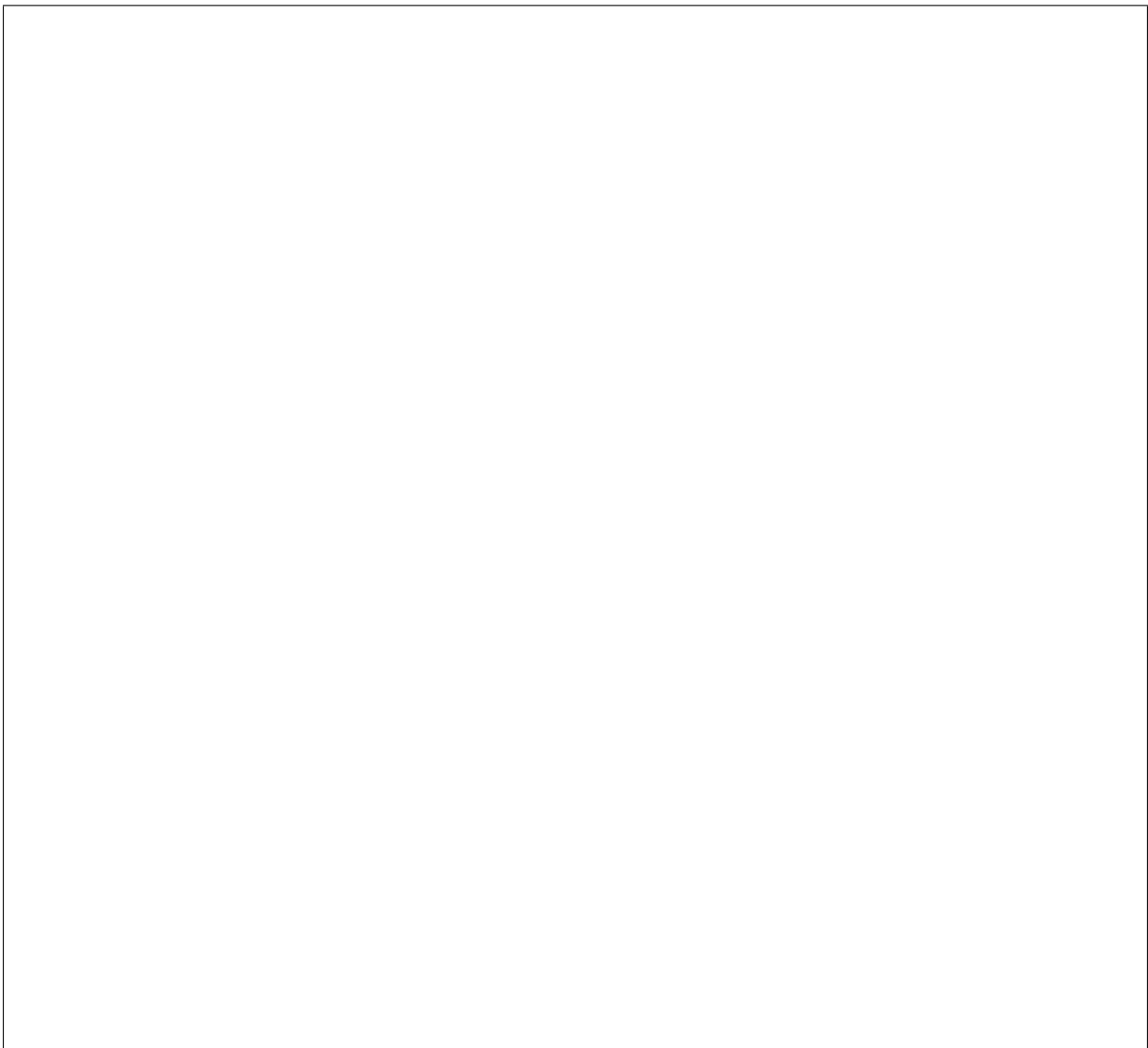




ing <name>: <value>.



error message.



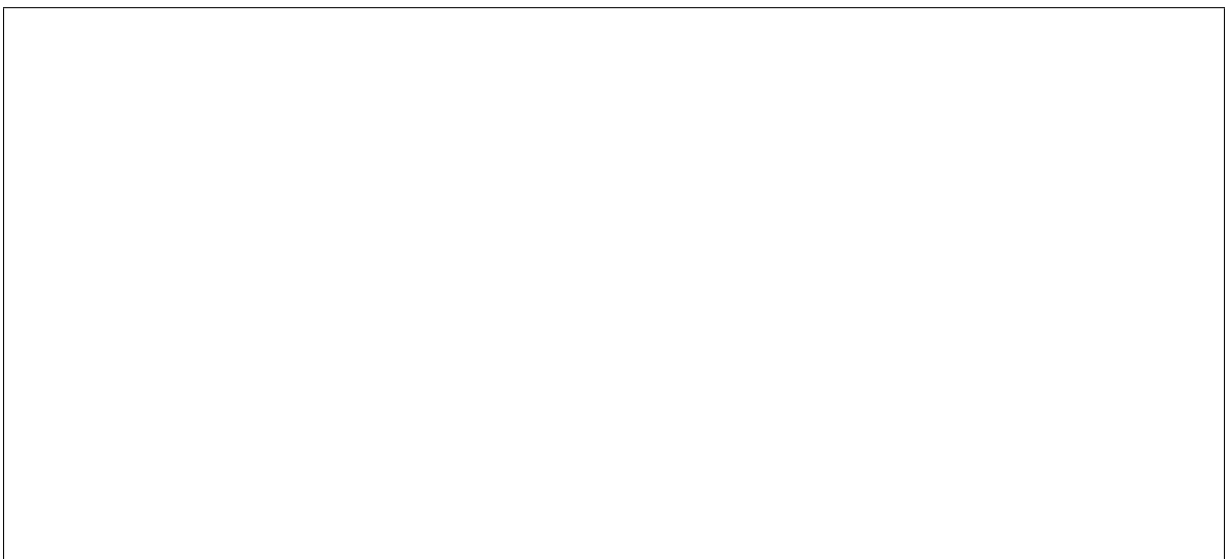
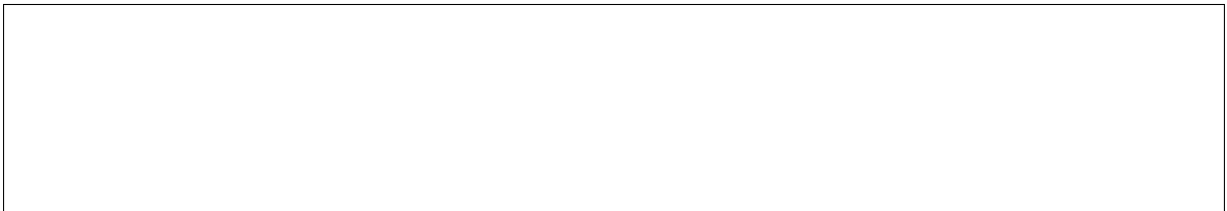
(continues on next page)

(continued from previous page)

**Starting manual cleaning via openstack metal CLI**



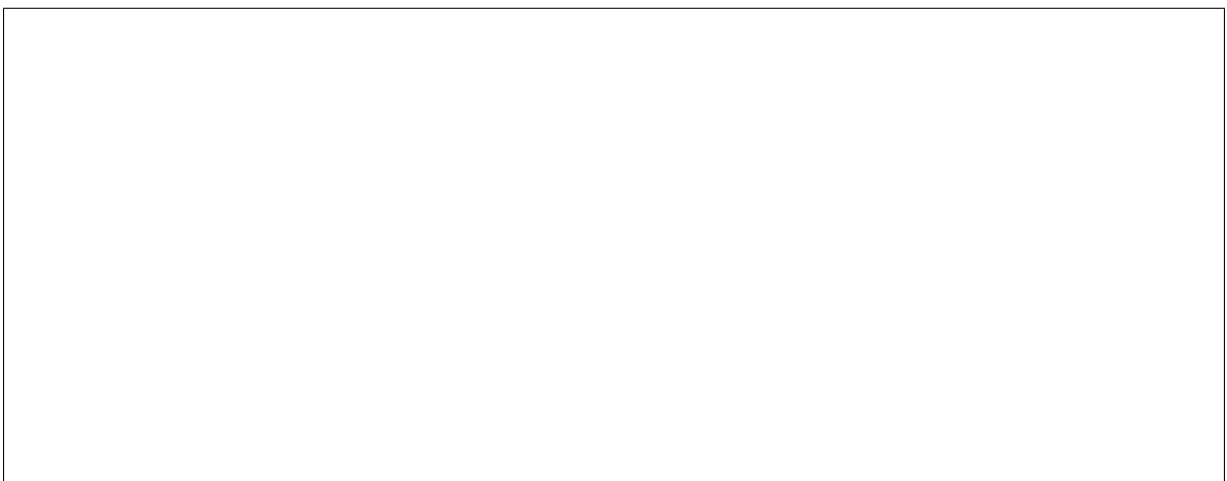




(continues on next page)



(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:

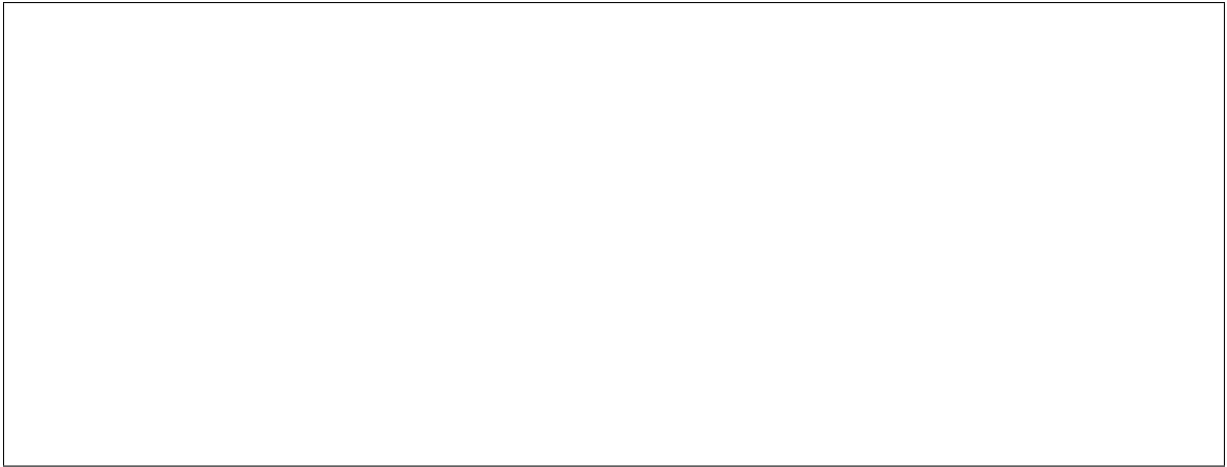
(continues on next page)

(continued from previous page)

--

by `shred` for software based disk erase is 1. To configure the number of iterations, use the following configuration option:

**What cleaning step is running?**



**Should I disable automated cleaning?**



time consuming process. To mitigate this, we suggest using disks with support for cryptographic ATA Security Erase, as typically the `erase_devices` step in the deploy interface takes the longest time to complete of all cleaning steps.

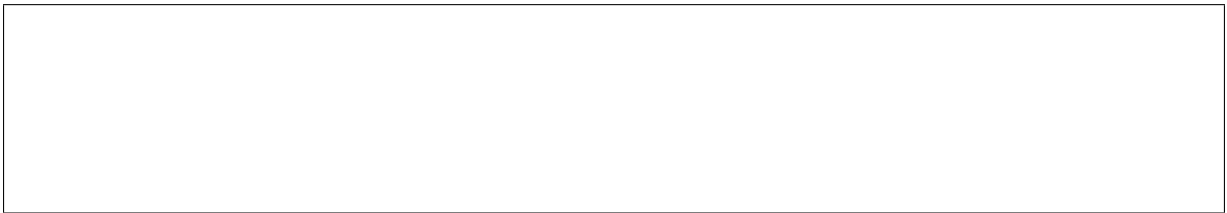
**Why cant I power on/off a node while its cleaning?**

## **Troubleshooting**

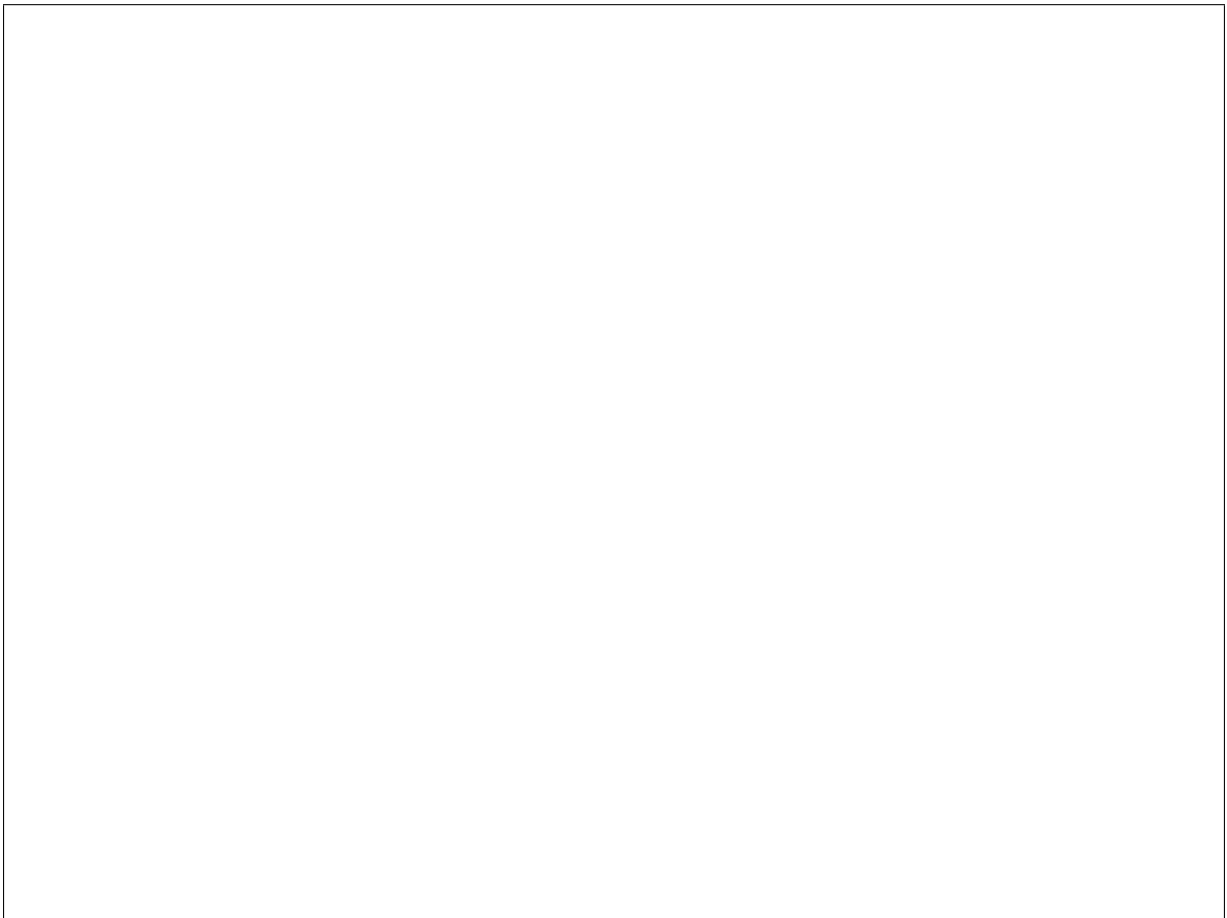
tor should validate that no permanent damage has been done to the node and no processes are still running on it before removing the maintenance mode.

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



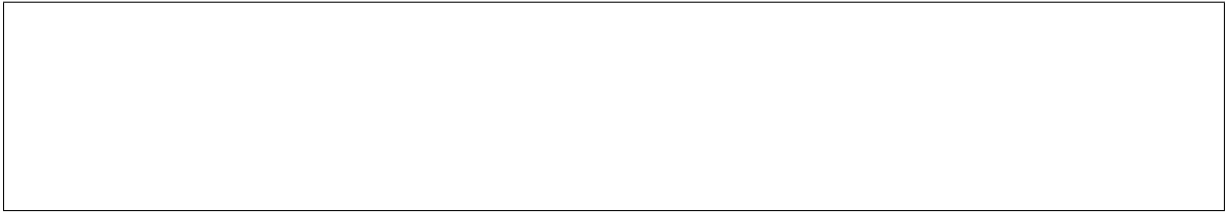






(continues on next page)

(continued from previous page)



## **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.

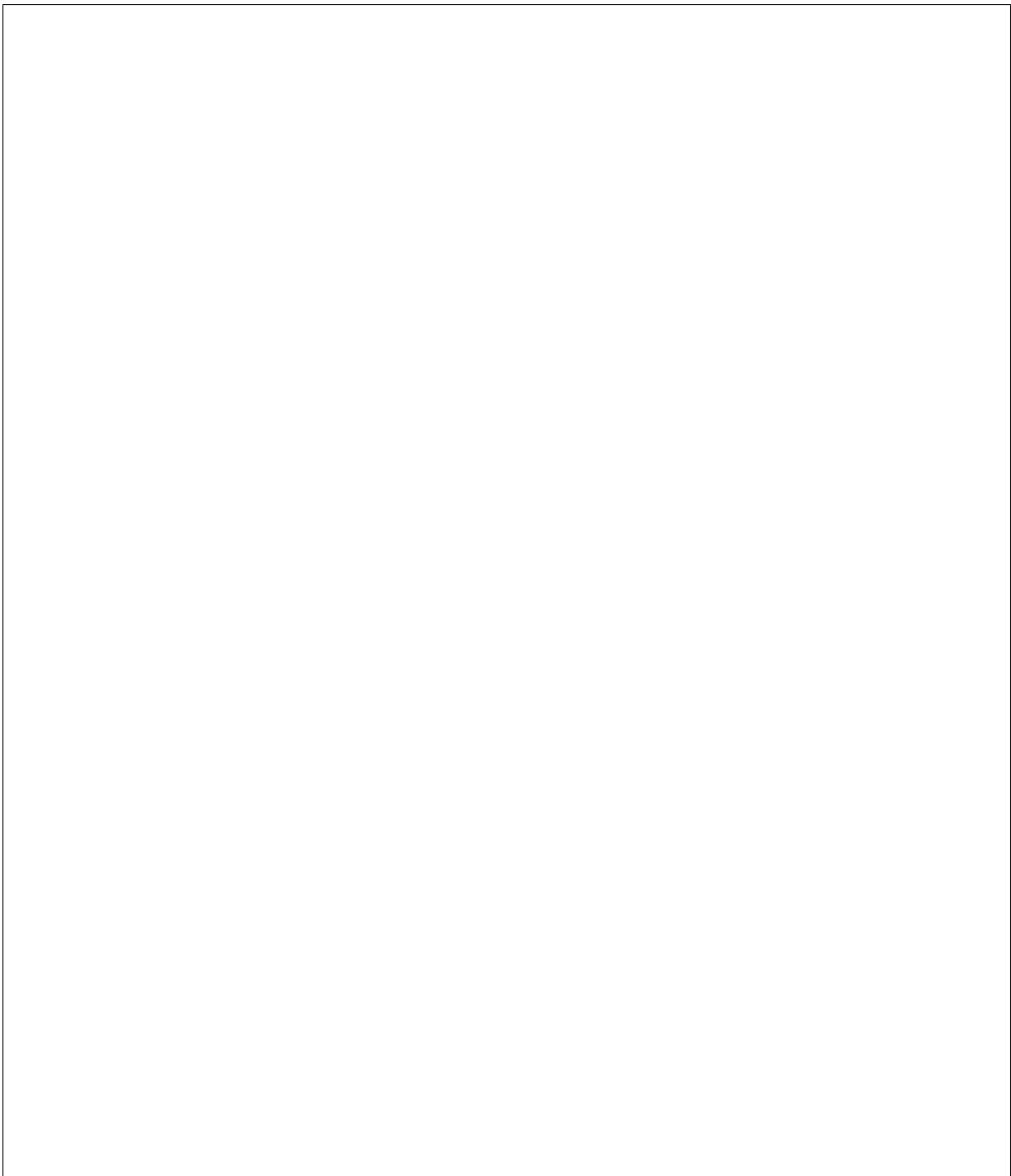
## **Requirements**





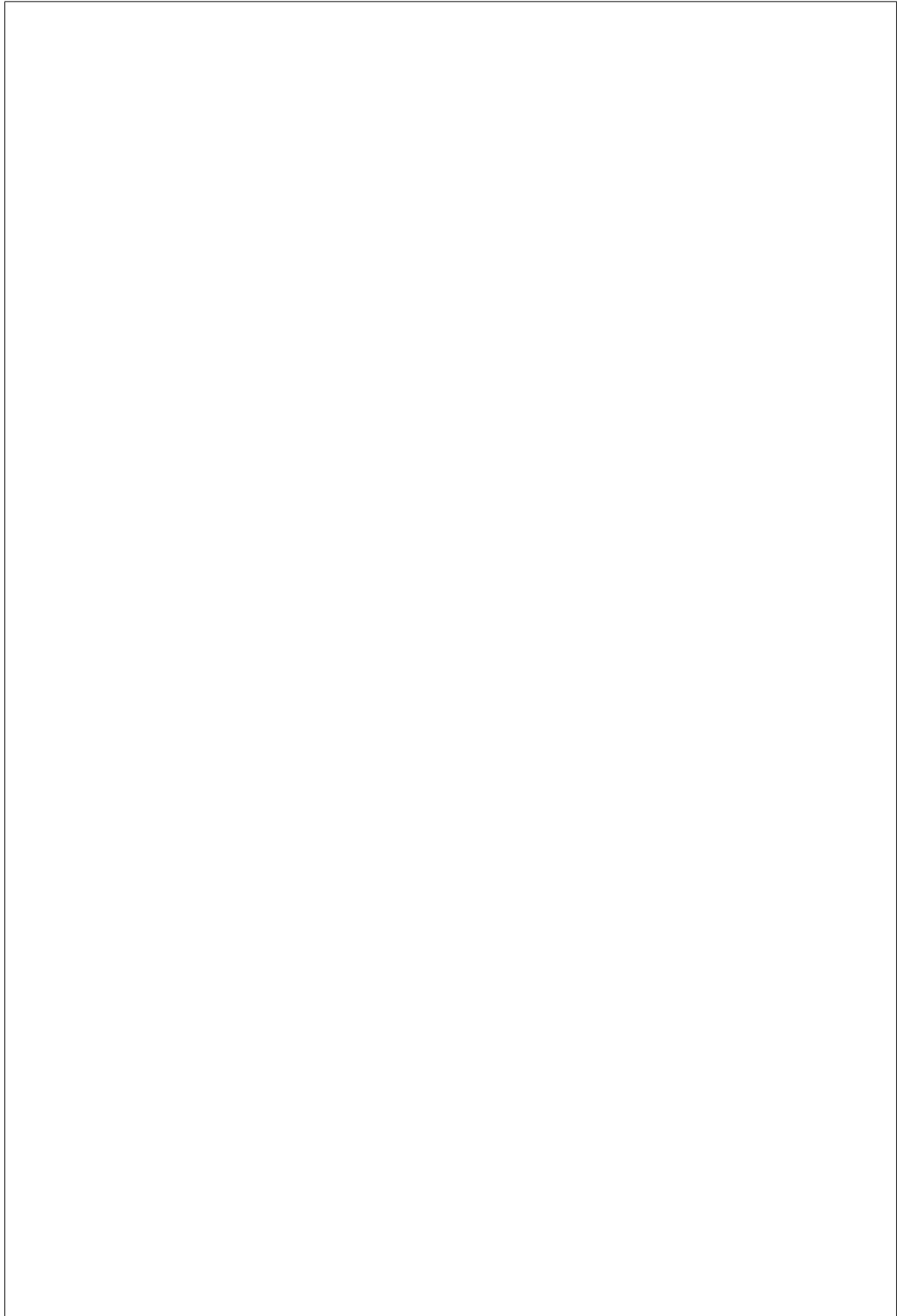
to retrieve the pertinent files. Inability to do so will result in the adoption failing, and the node being placed in the `adopt failed` state.

### **Example**



(continues on next page)

(continued from previous page)



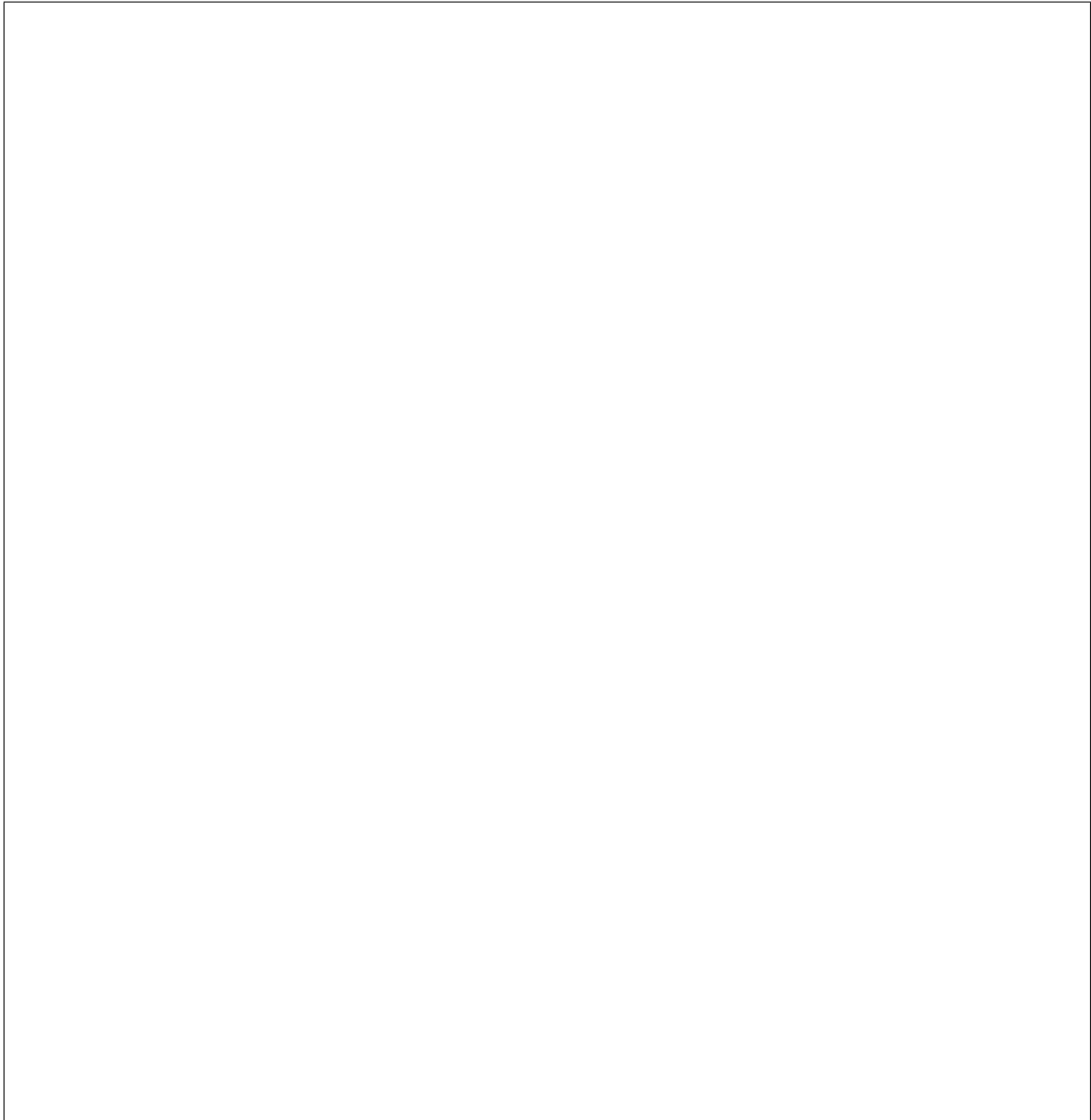
(continues on next page)

(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.

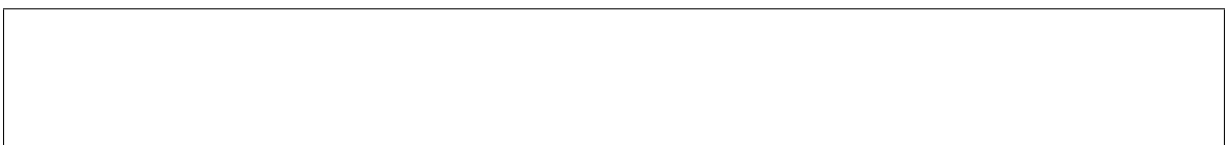
**Note:** The above example will fail a re-deployment as a fake image is defined and no `instance_info/image_checksum` value is defined. As such any actual attempt to write the image out will fail as the `image_checksum` value is only validated at time of an actual deployment operation.

**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.

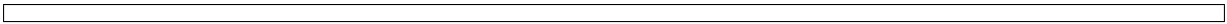
## Troubleshooting

dation step. Validation steps are dependent upon what driver is selected for the node.



(continues on next page)

(continued from previous page)





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



## **Overview**



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

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

### **How to use**

(continues on next page)

(continued from previous page)

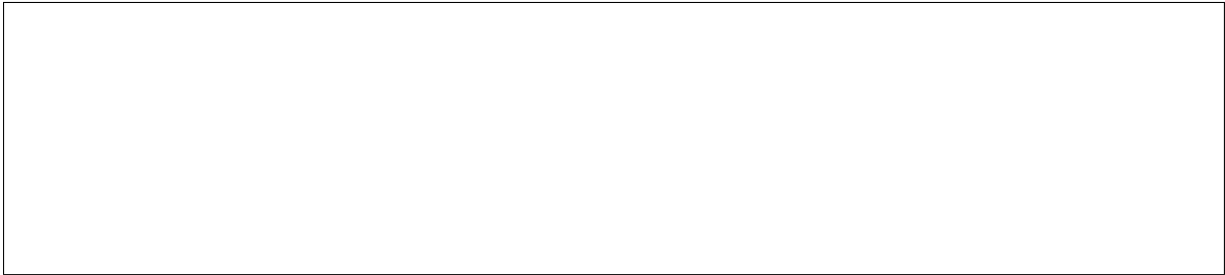
**Note:** An exception are nodes which are in `available`. For backwards compatibility reasons, these nodes need to be moved to `manageable` first. Trying to set the `retired` flag for `available` nodes will result in an error.







needs to be removed first. This can be done via:





## **Overview**

the bare metal during manual cleaning.

## **Prerequisites**

with some caveats - see *Software RAID* for details.



**Build agent ramdisk which supports RAID configuration**

should be used for HPE Proliant Servers.

**Note:** For in-band software RAID, the agent ramdisk does not need to be bundled with a hardware manager as the generic hardware manager in the IroniC Python Agent already provides (basic) support for software RAID.

### **RAID configuration JSON format**



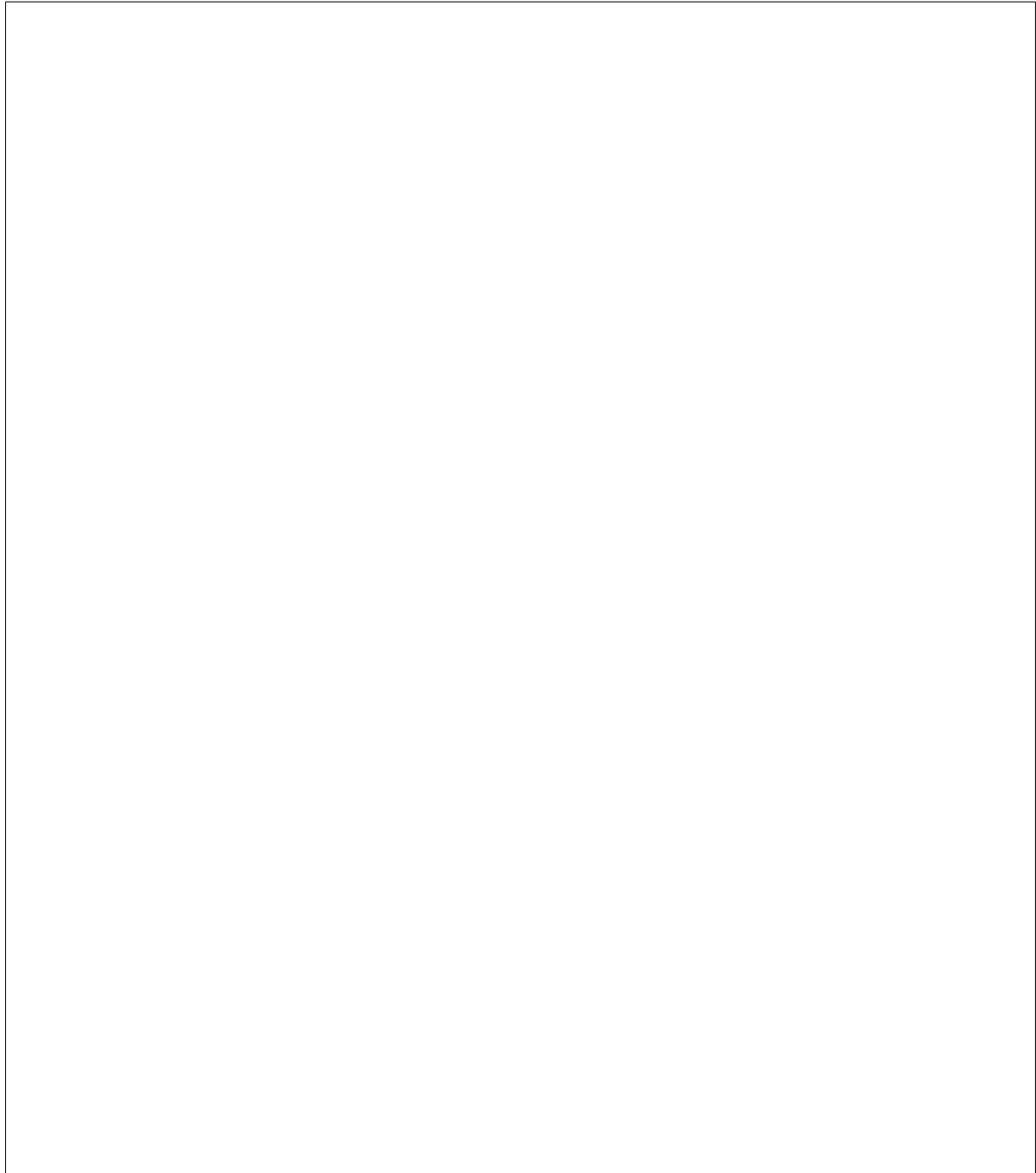
## **Target RAID configuration**

cleaning.



(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.

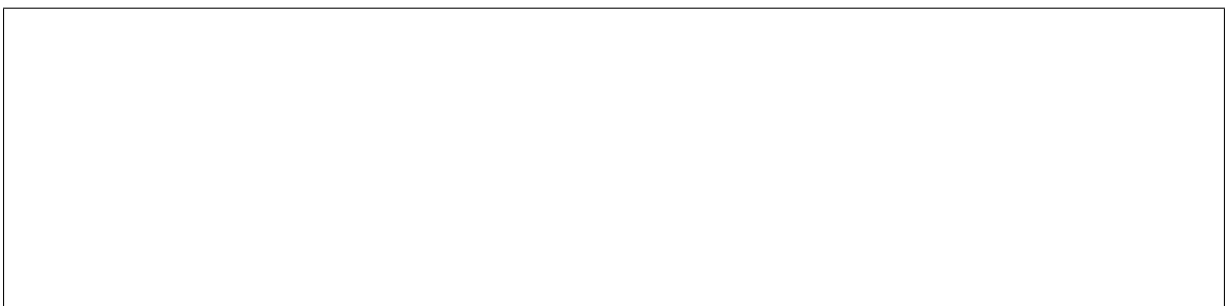




peat the same hint if necessary).

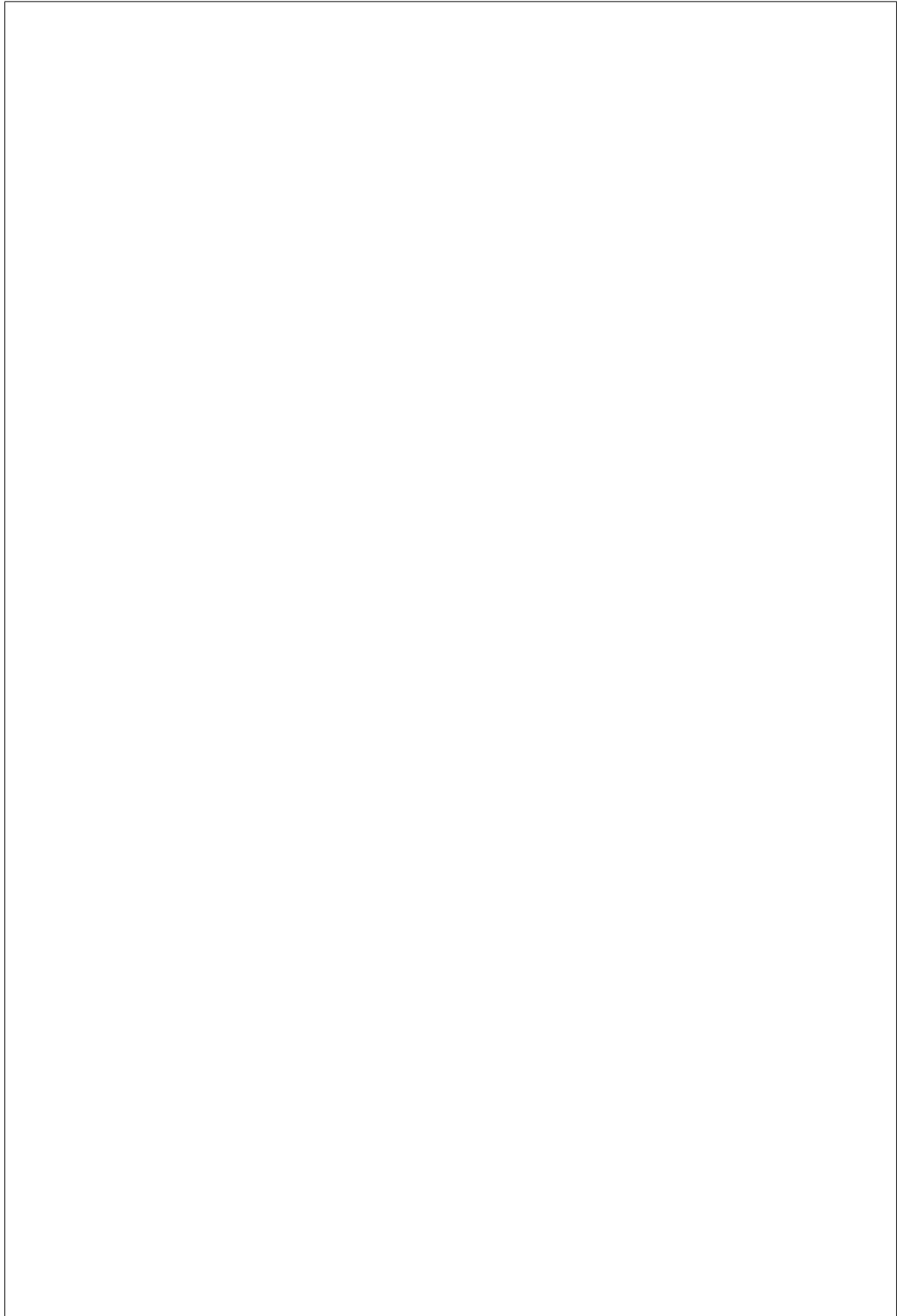
**Note:** If properties from both Backing physical disk hints or Backing physical disks are specified, they should be consistent with each other. If they are not consistent, then the RAID configuration will fail (because the appropriate backing physical disks could not be found).

### Examples for `target_raid_config`



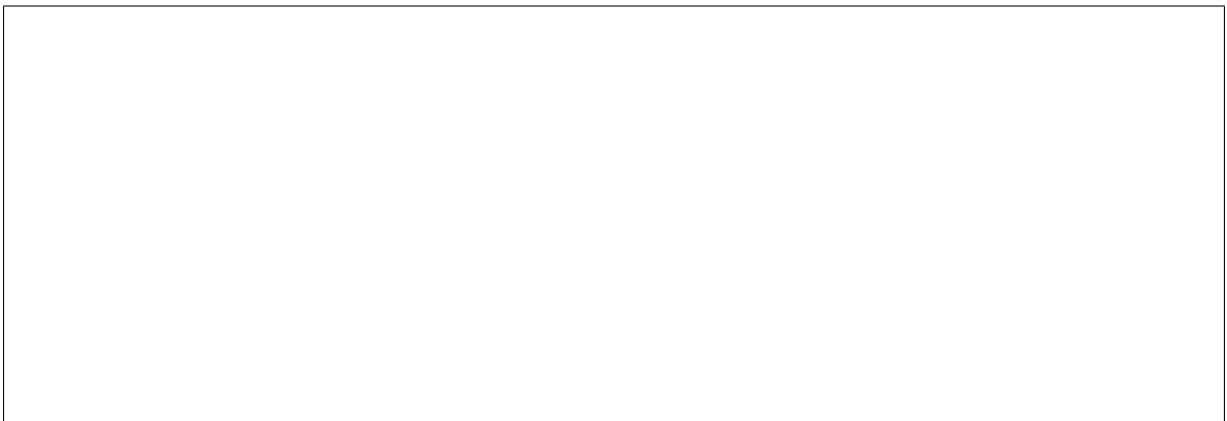
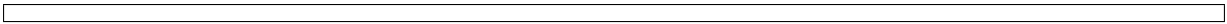
(continues on next page)

(continued from previous page)



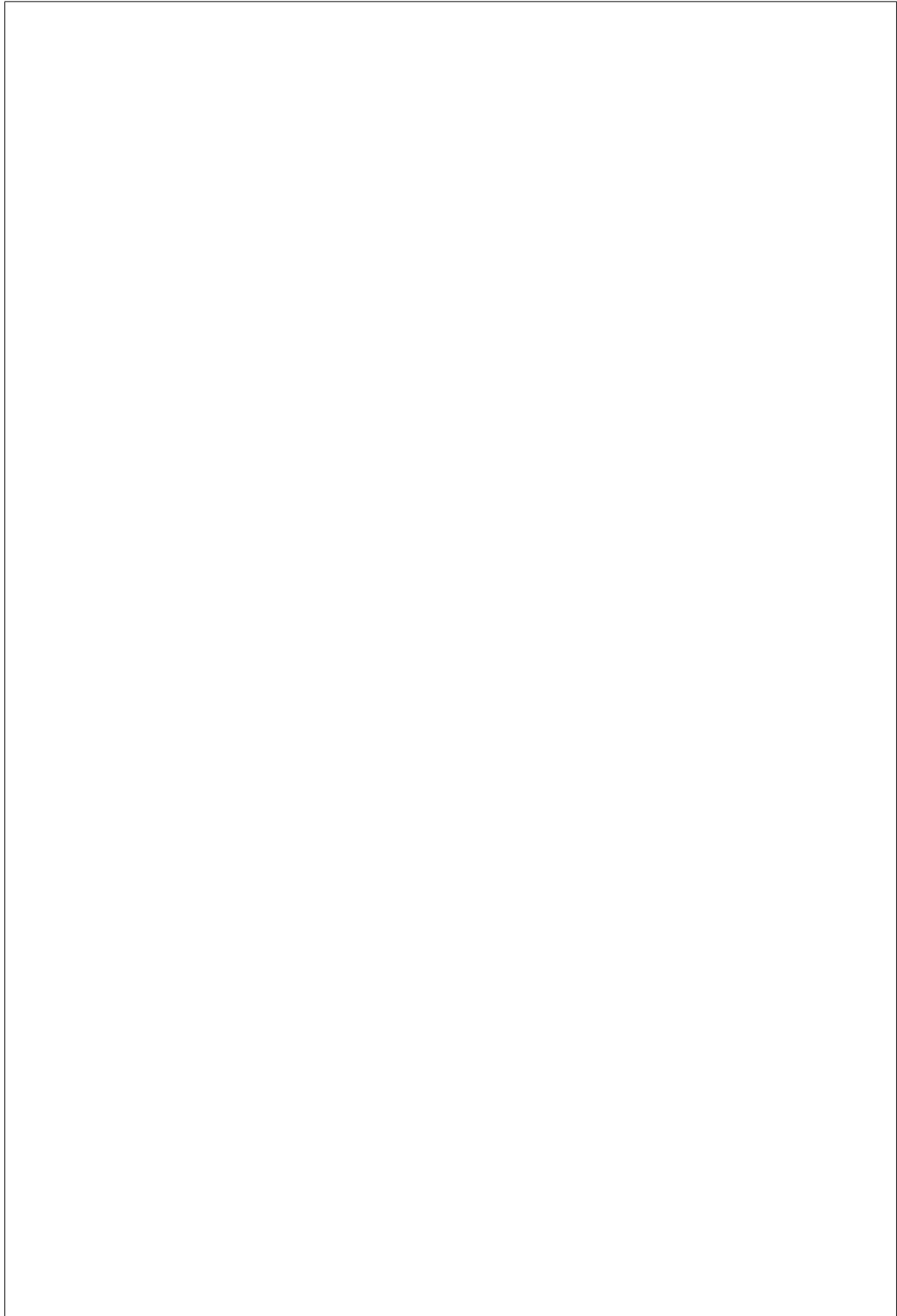
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



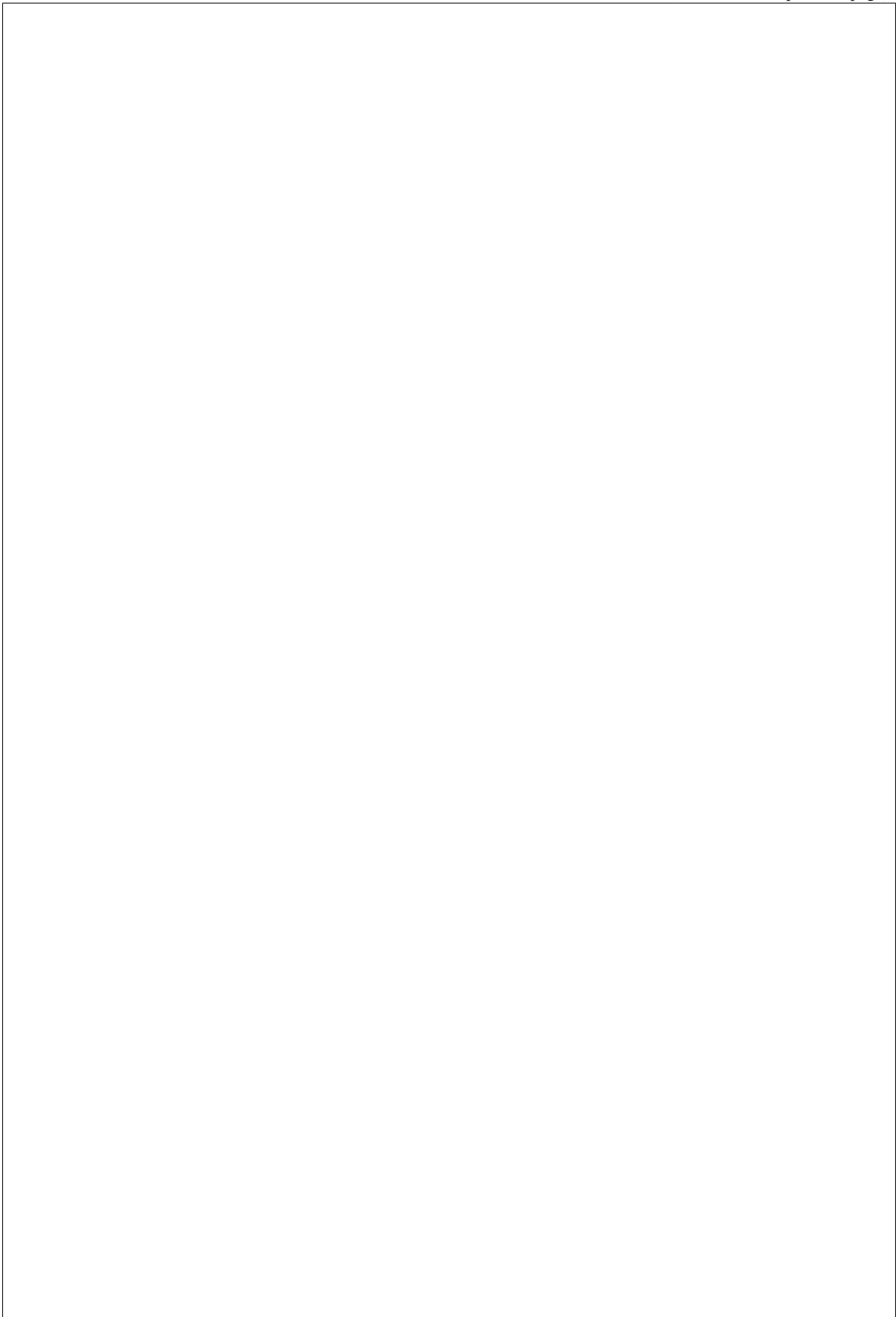
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)



(continued from previous page)



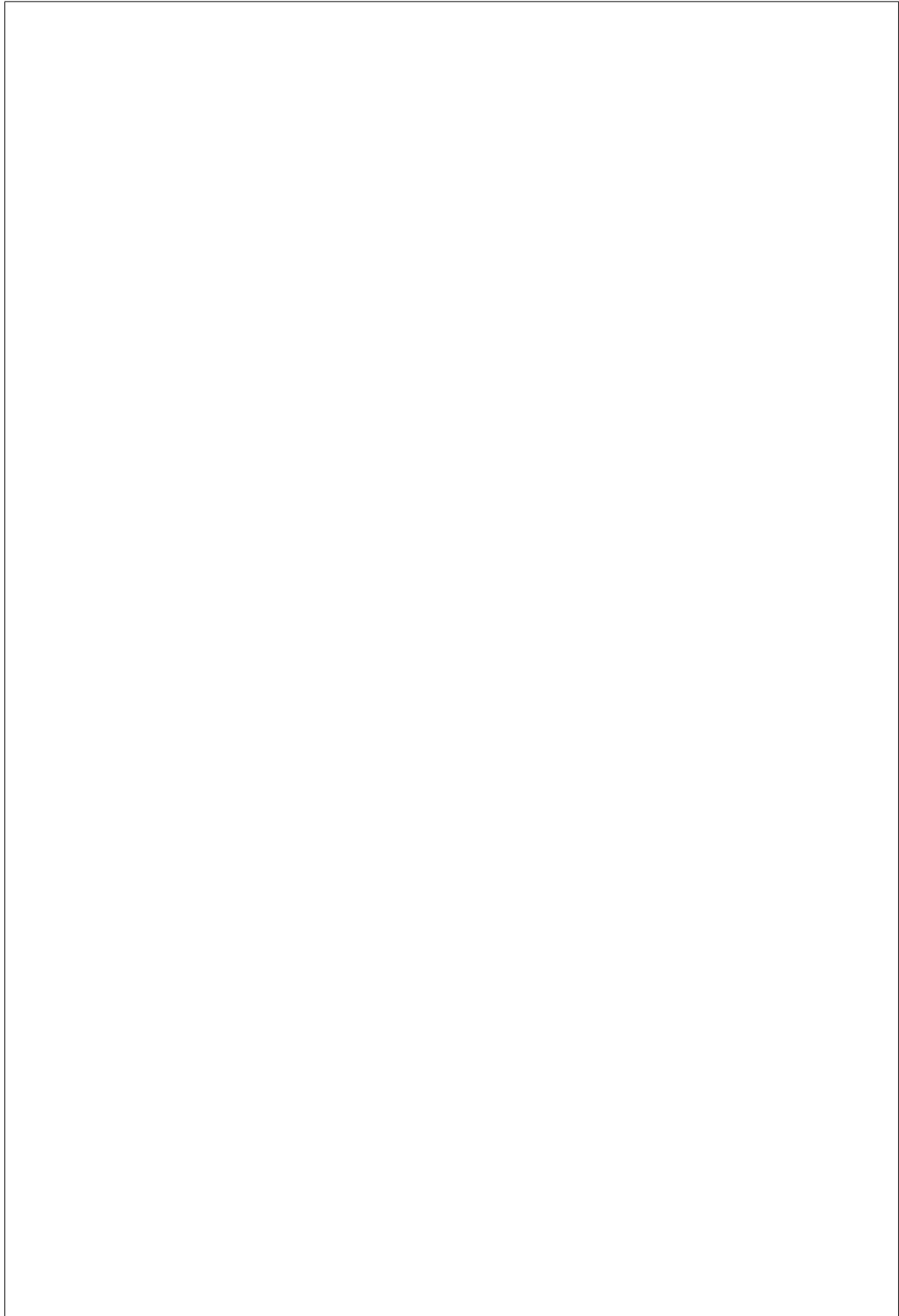
(continues on next page)

(continued from previous page)



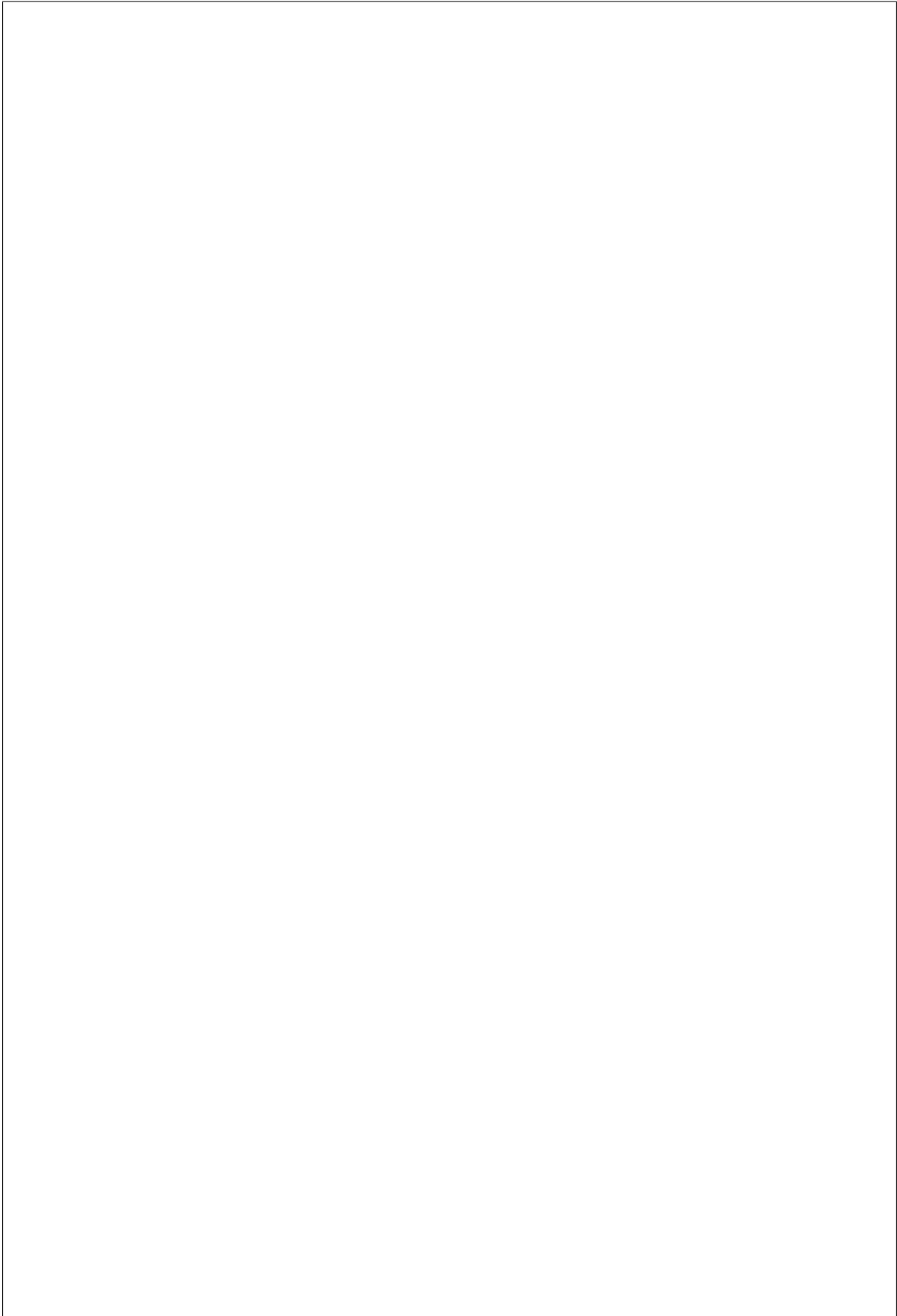
(continues on next page)

(continued from previous page)



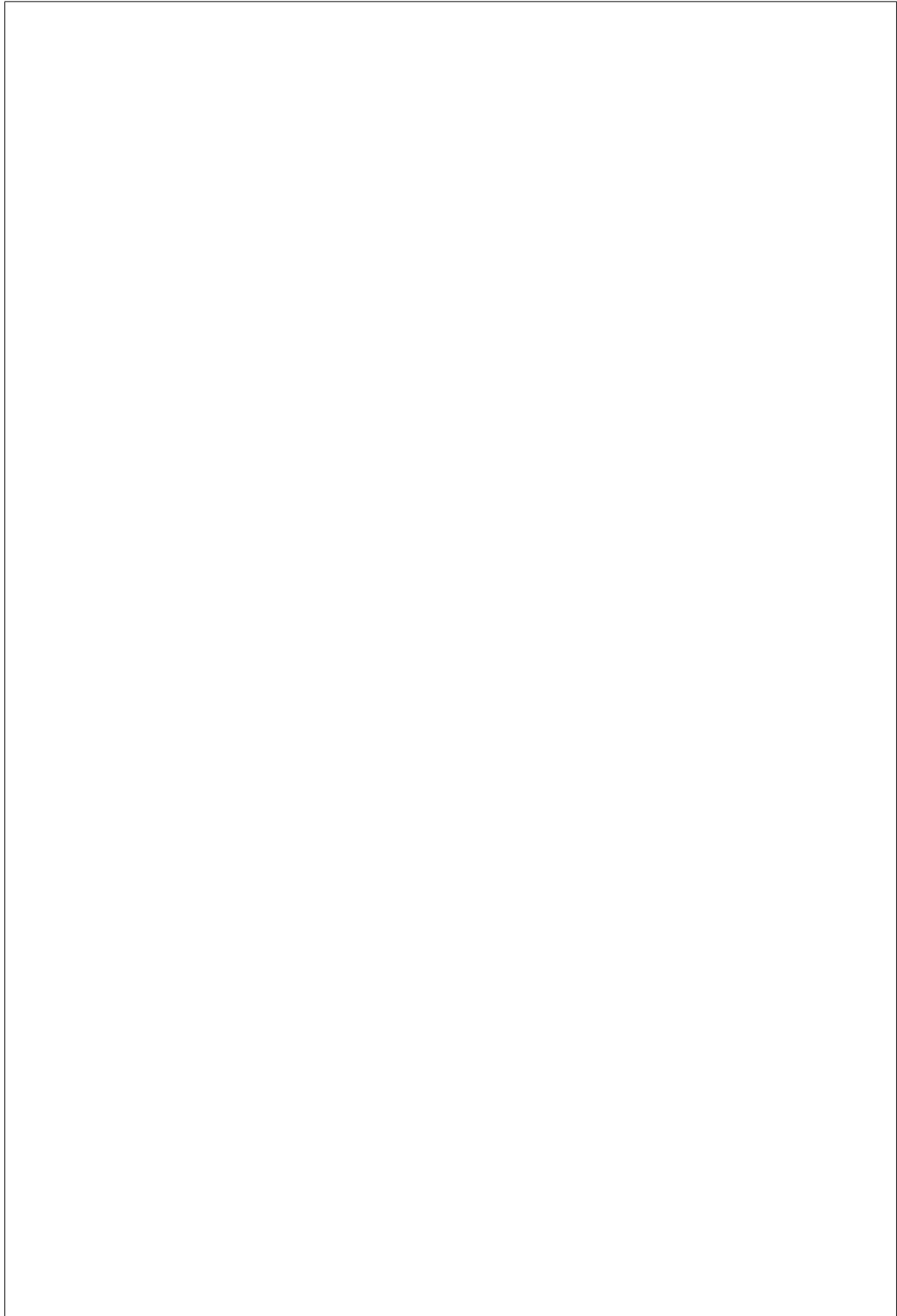
(continues on next page)

(continued from previous page)



(continues on next page)

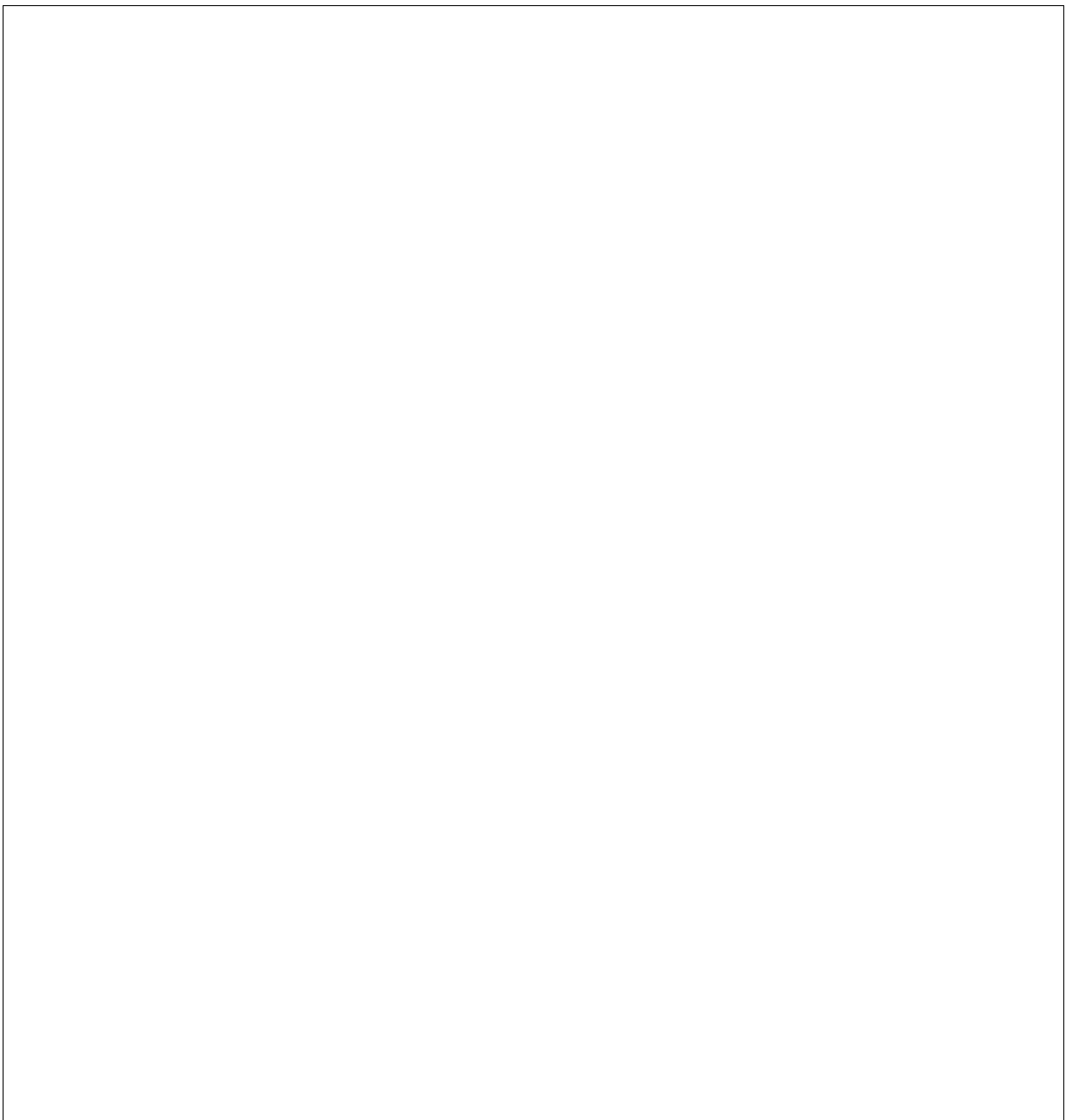
(continued from previous page)



(continues on next page)

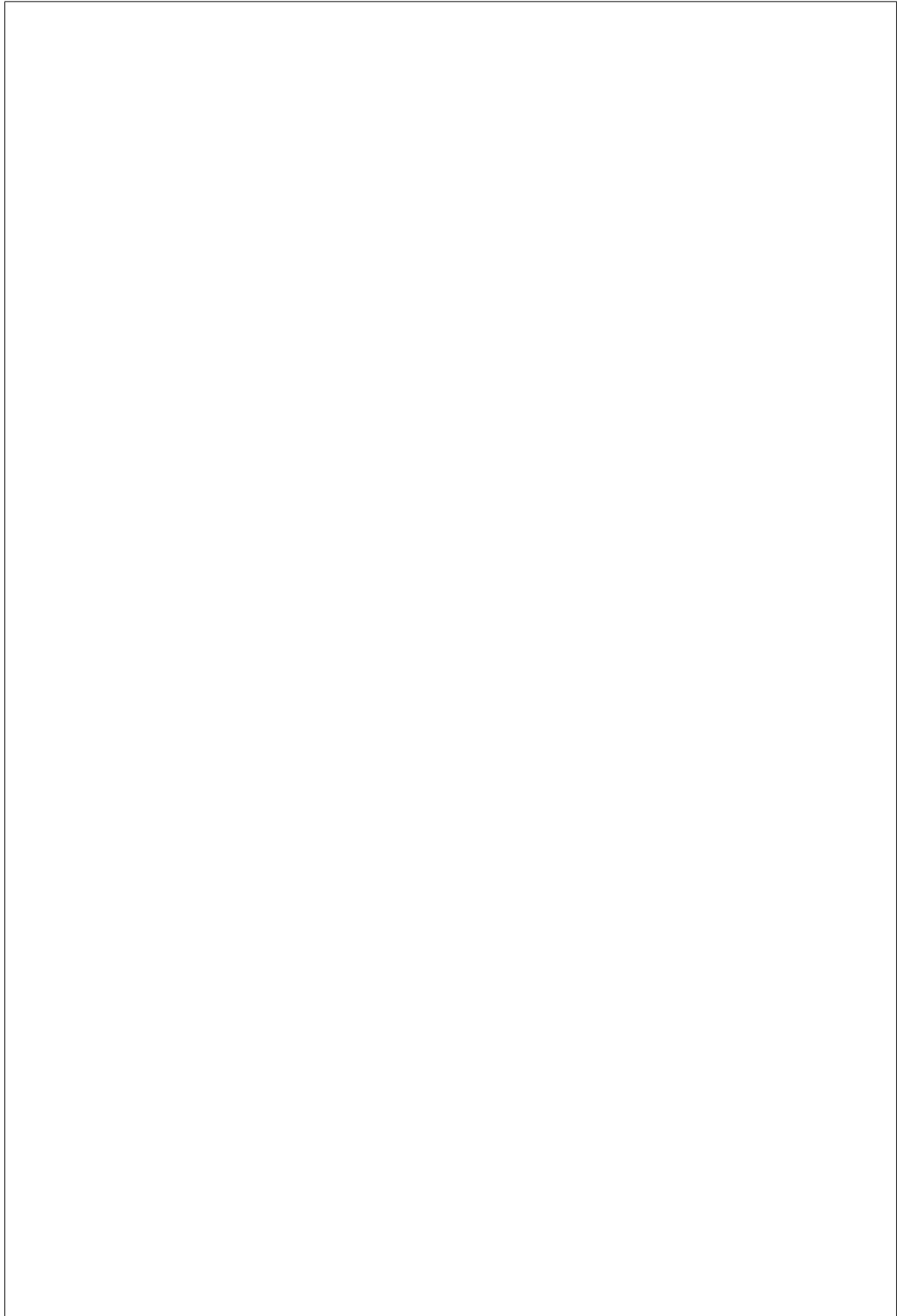
(continued from previous page)

---



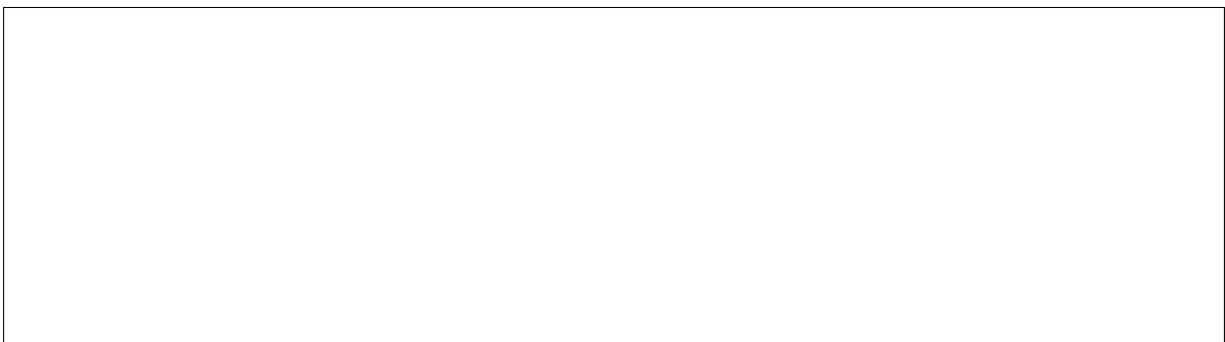
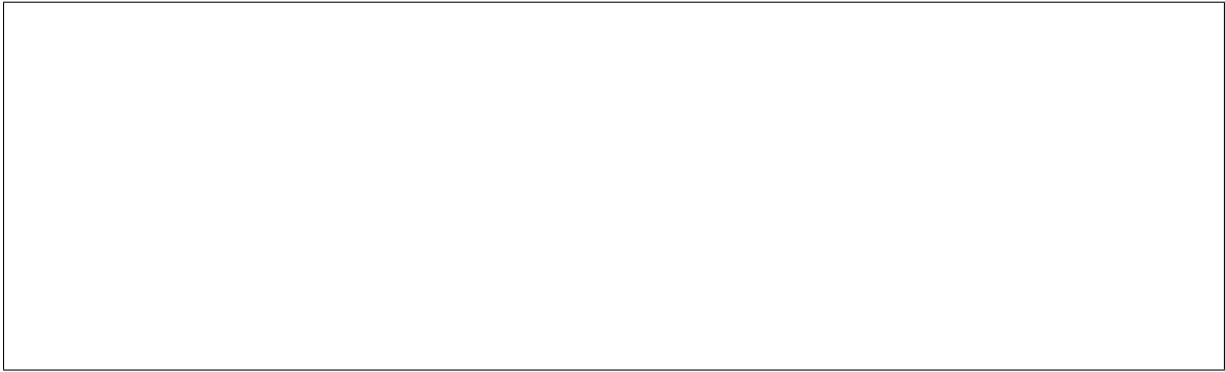
(continues on next page)

(continued from previous page)



(continues on next page)

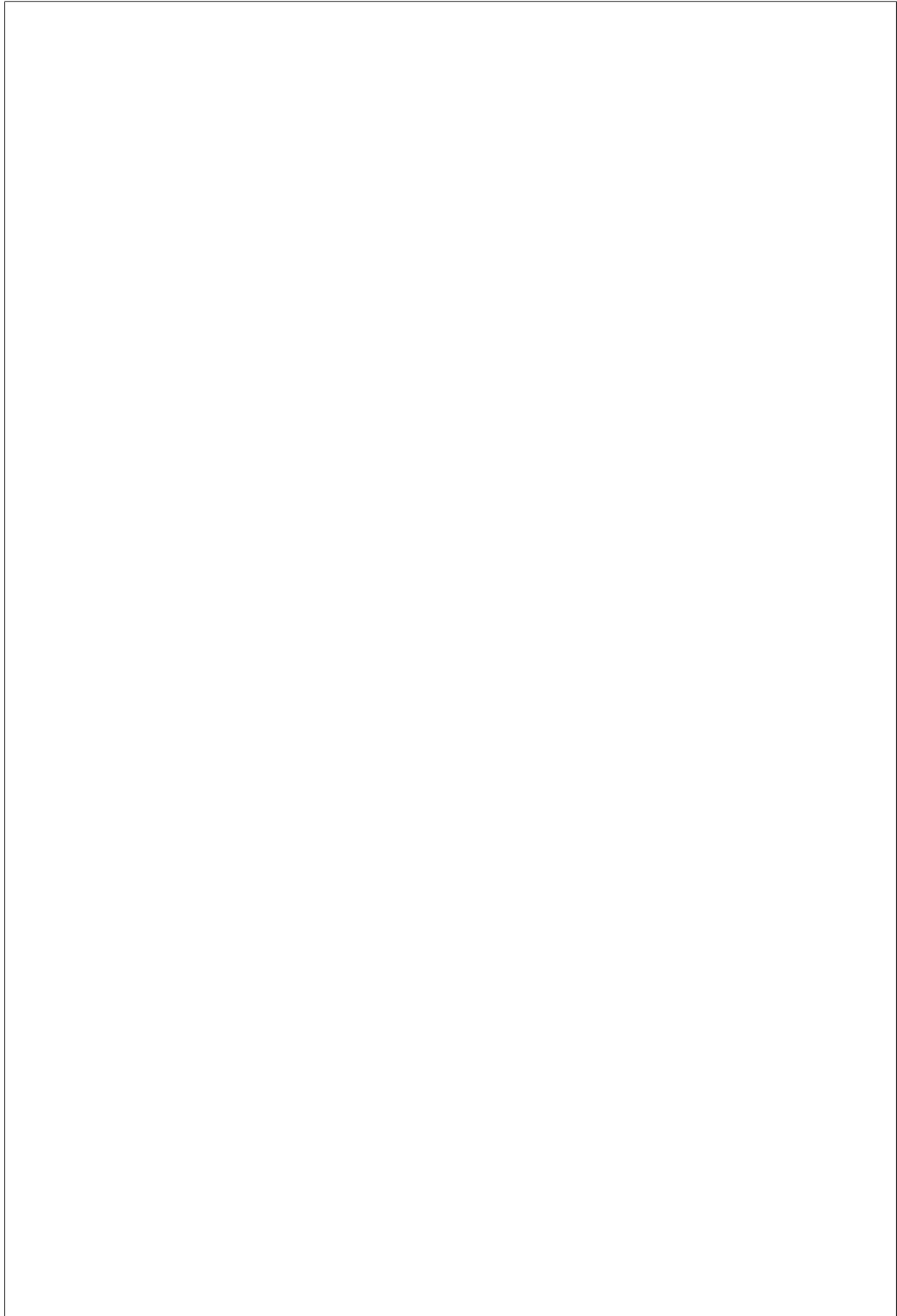
(continued from previous page)



(continues on next page)



(continued from previous page)



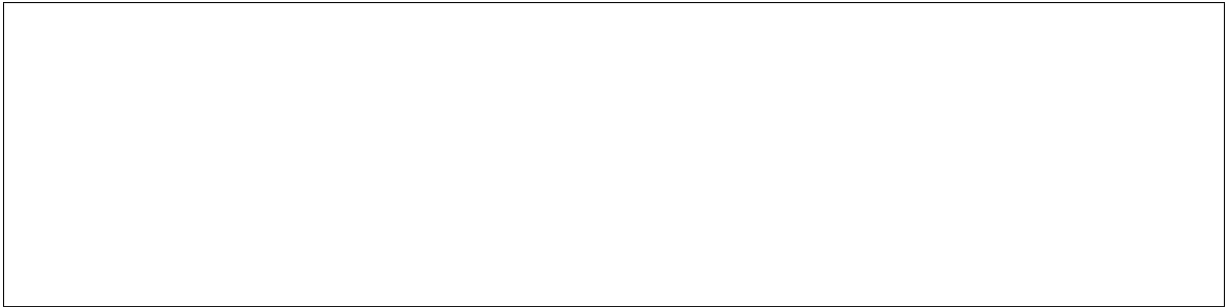
(continues on next page)

(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**

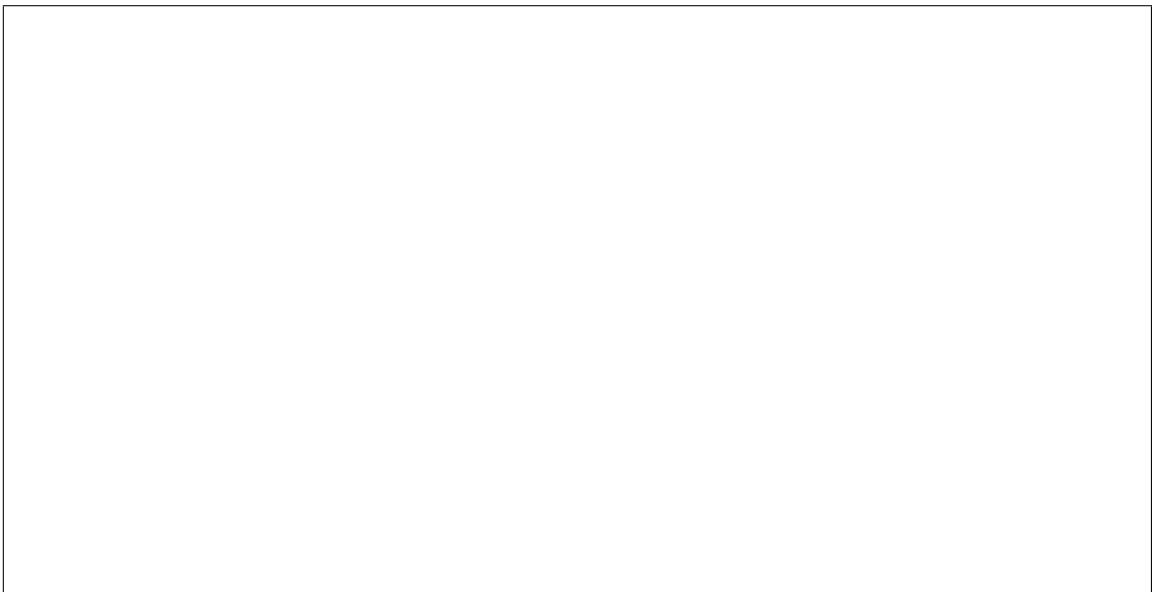
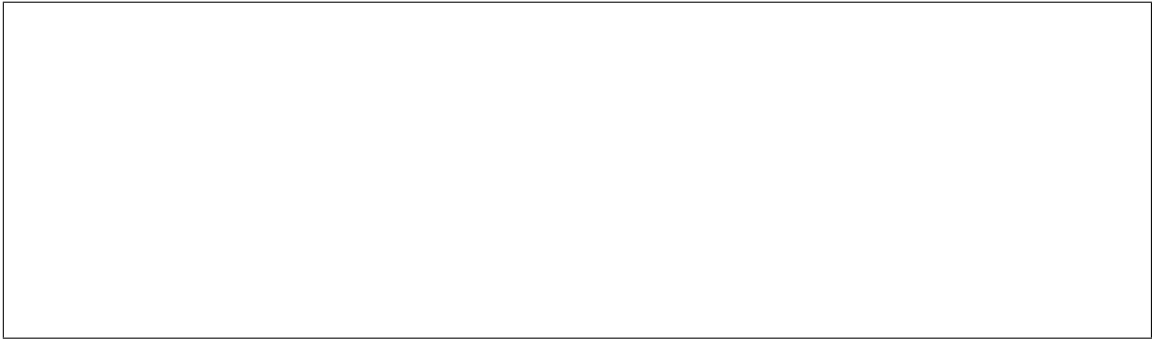


mation.



(continues on next page)

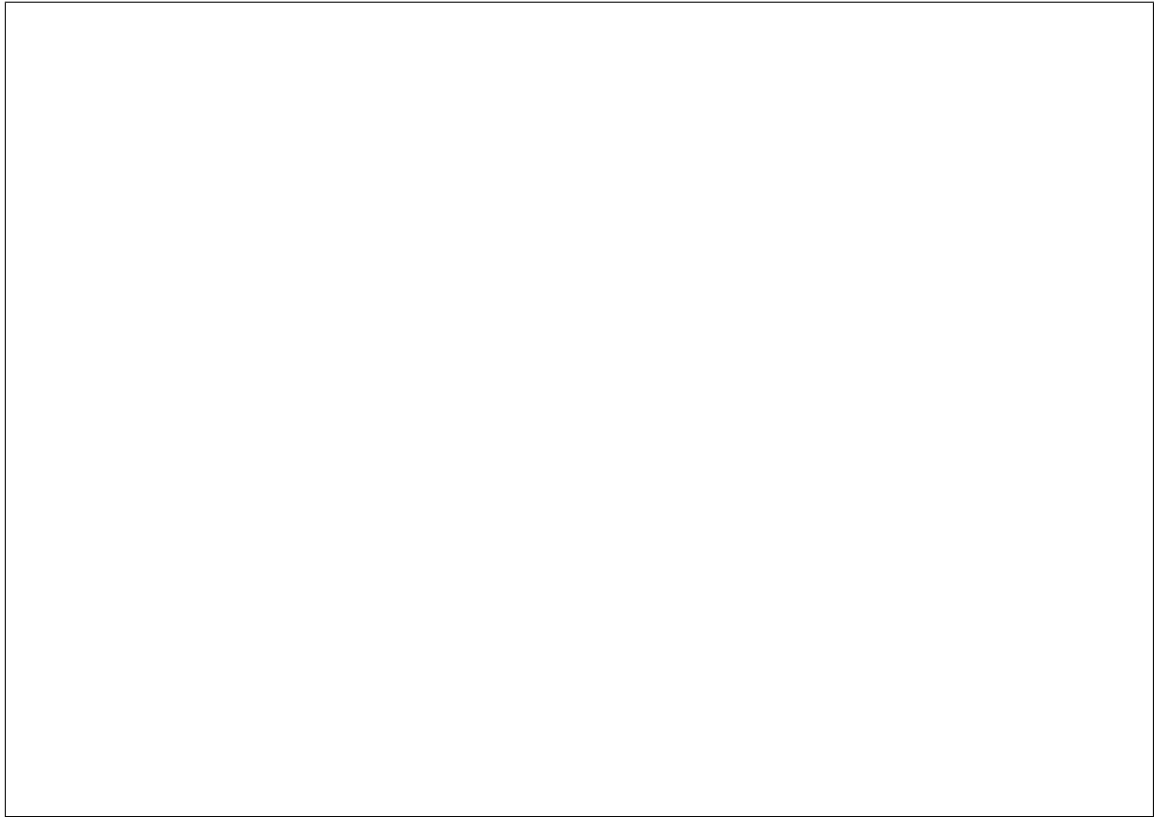
(continued from previous page)

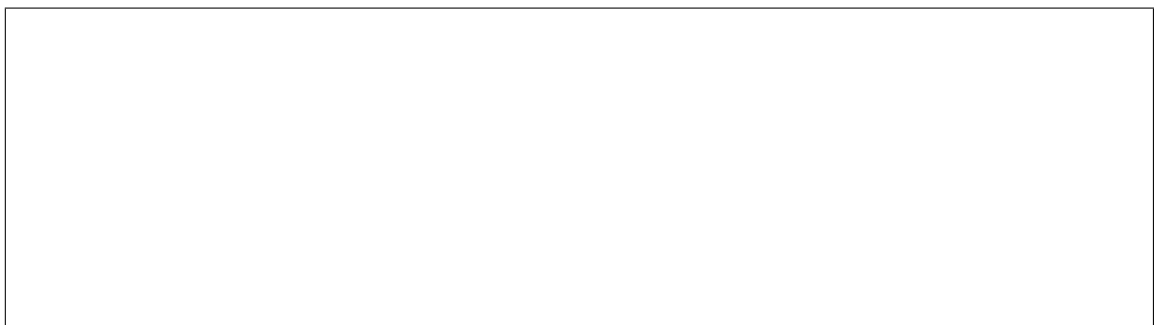


(continues on next page)



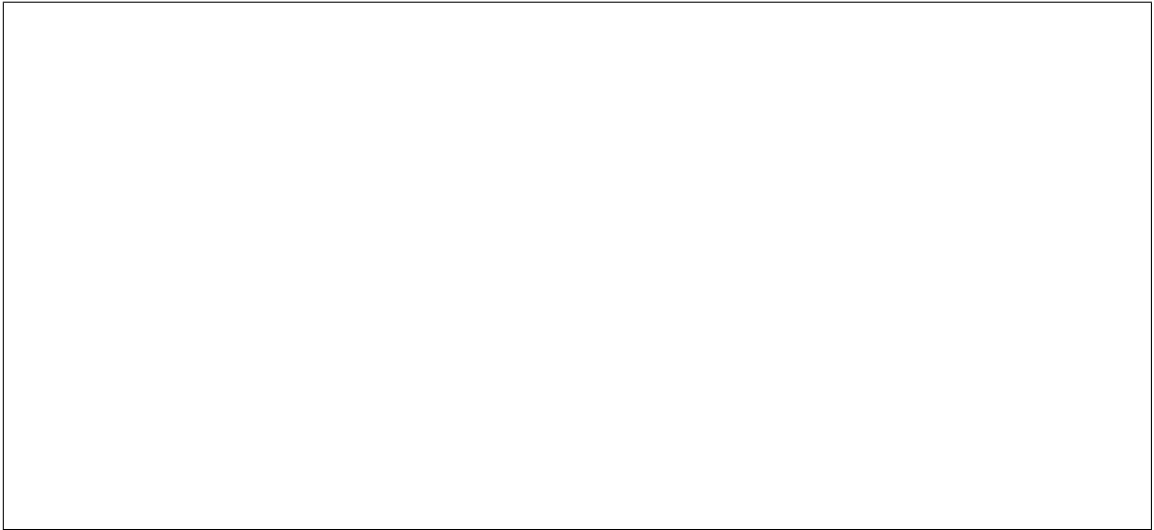
(continued from previous page)





(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

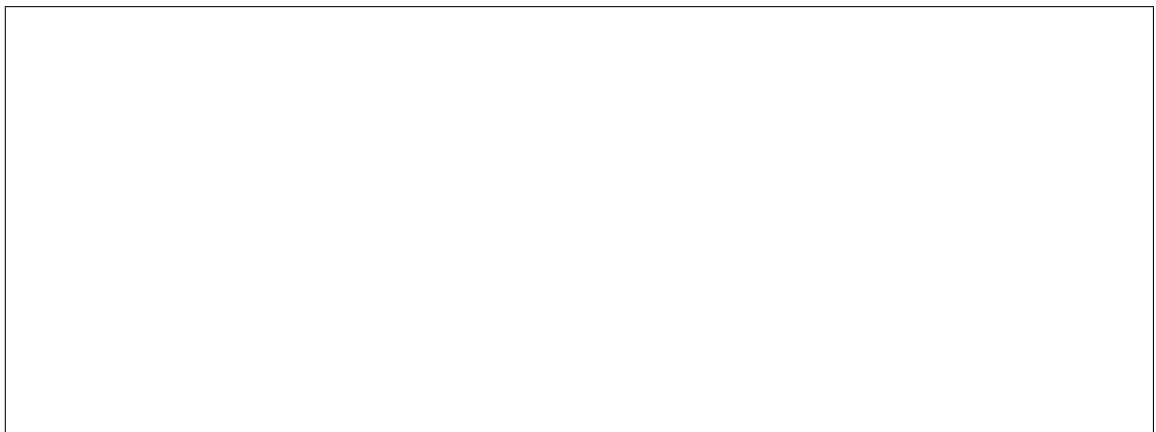
---

## **Software RAID**

a software RAID configuration example in *Examples for target\_raid\_config*.

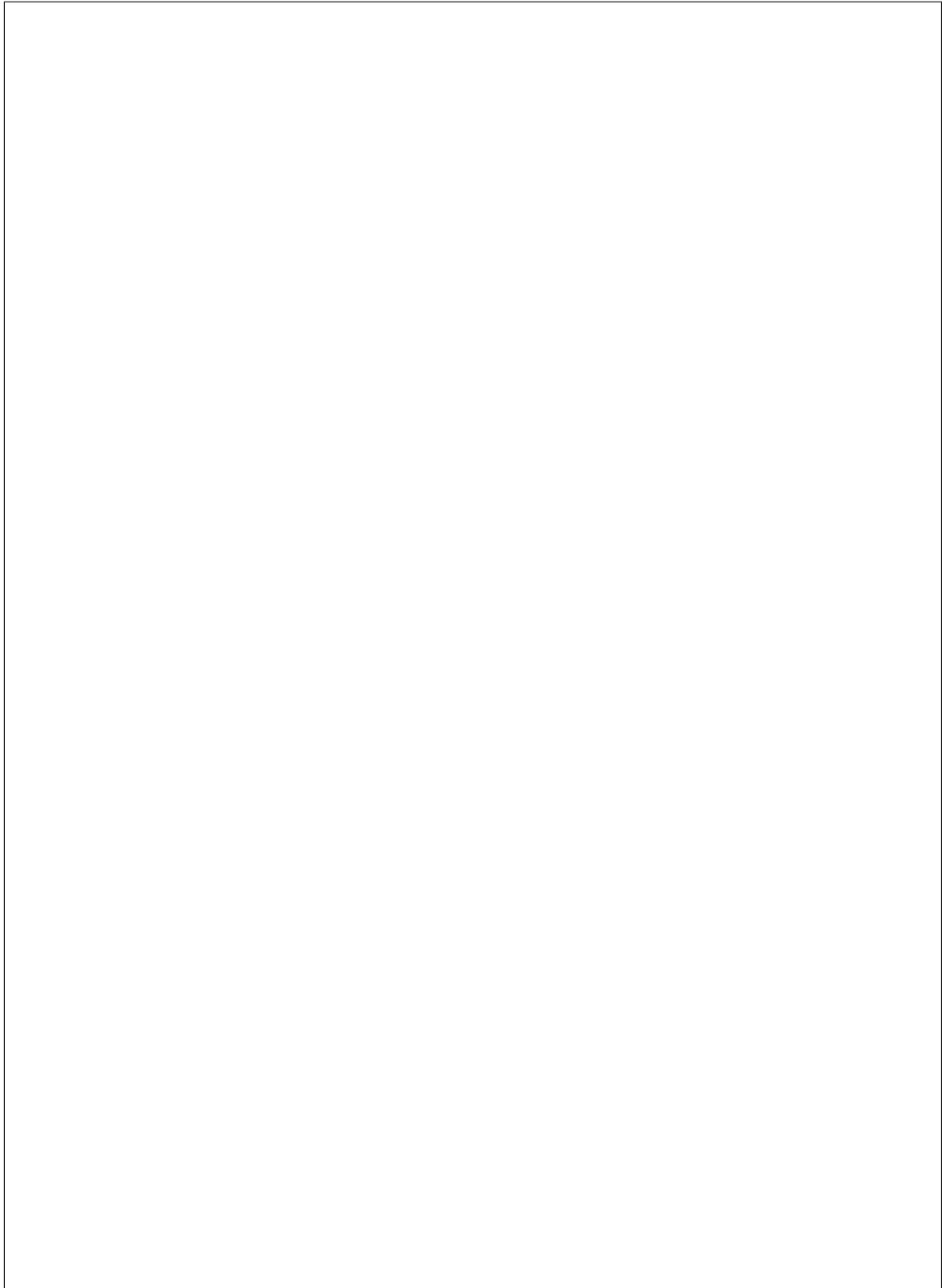


ond one can 0, 1, or 1+0. As the first RAID device will be the deployment device, enforcing a RAID-1 reduces the risk of ending up with a non-booting node in case of a disk failure.



(continues on next page)

(continued from previous page)



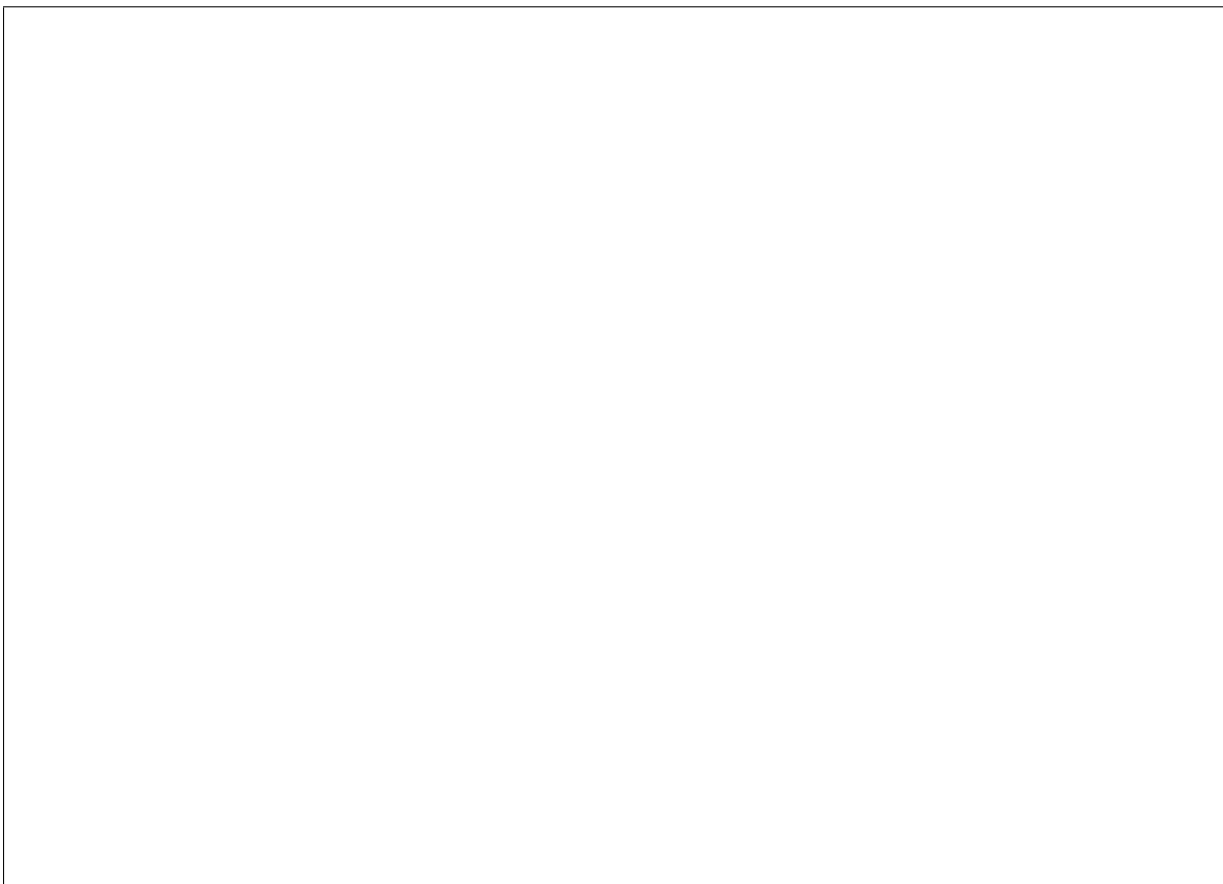


bedded in the images initrd).



## **Image requirements**

tem on the first partition. Starting with Ussuri, the image can also have additional metadata to point IroniC to the partition with the root file system: for this, the image needs to set the `rootfs_uuid` property with the file system UUID of the root file system. One way to extract this UUID from an existing image is to download the image, mount it as a loopback device, and use `blkid`:

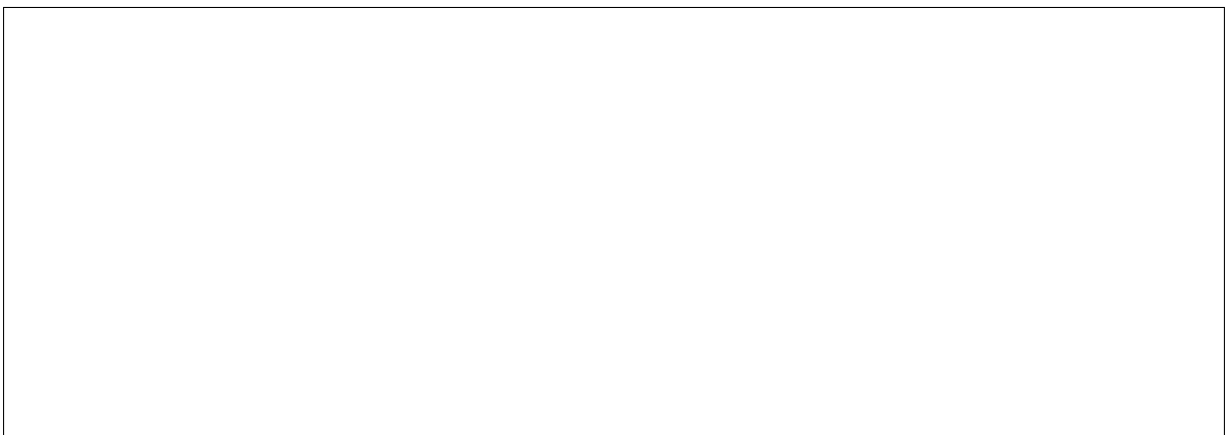


(continues on next page)

(continued from previous page)

--

## **Using RAID in nova flavor for scheduling**



## **Developer documentation**

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

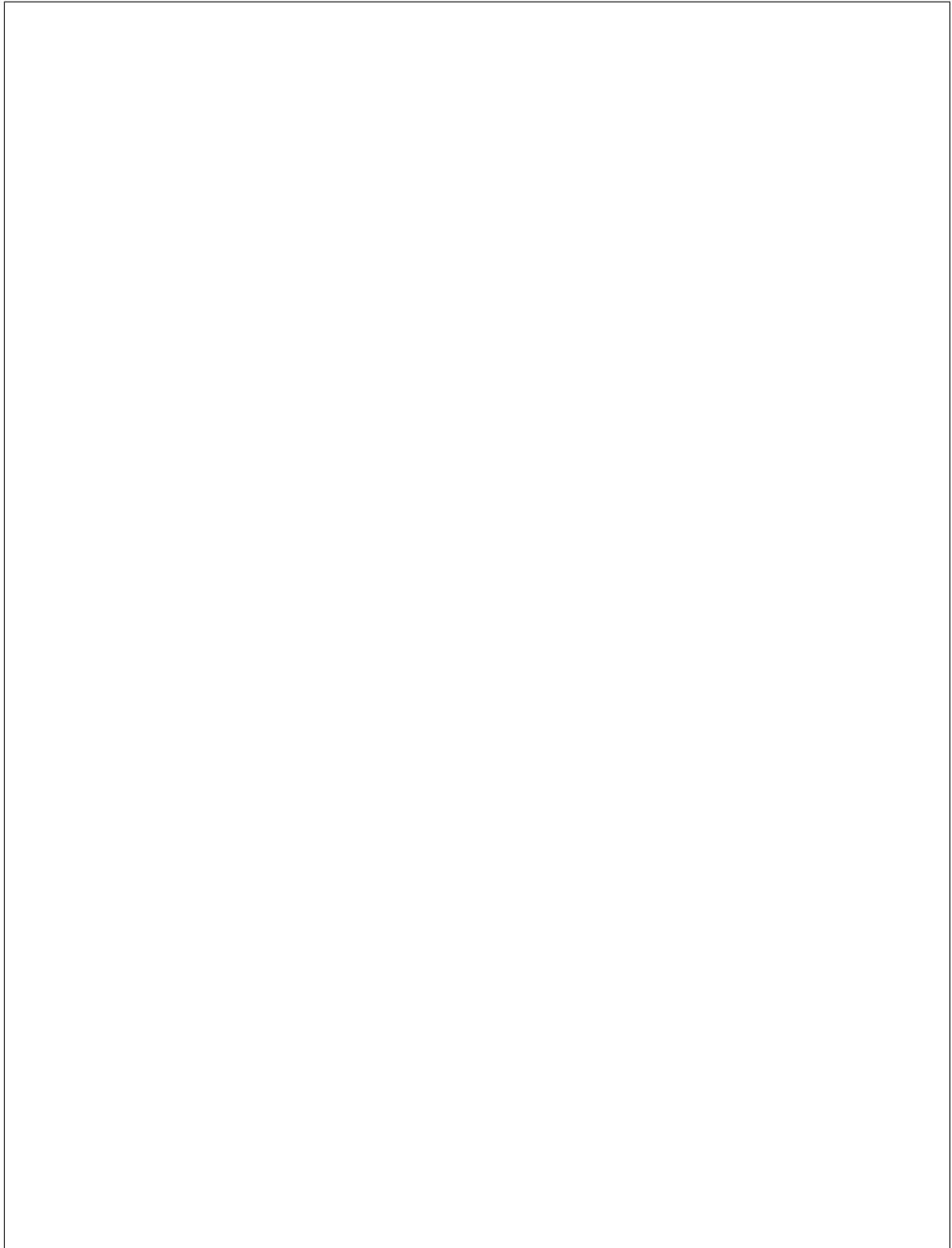


`set node.raid_config.`



(continues on next page)

(continued from previous page)



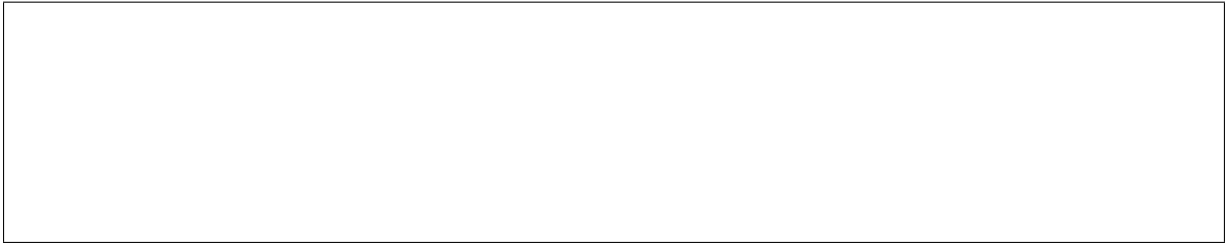
## **Overview**

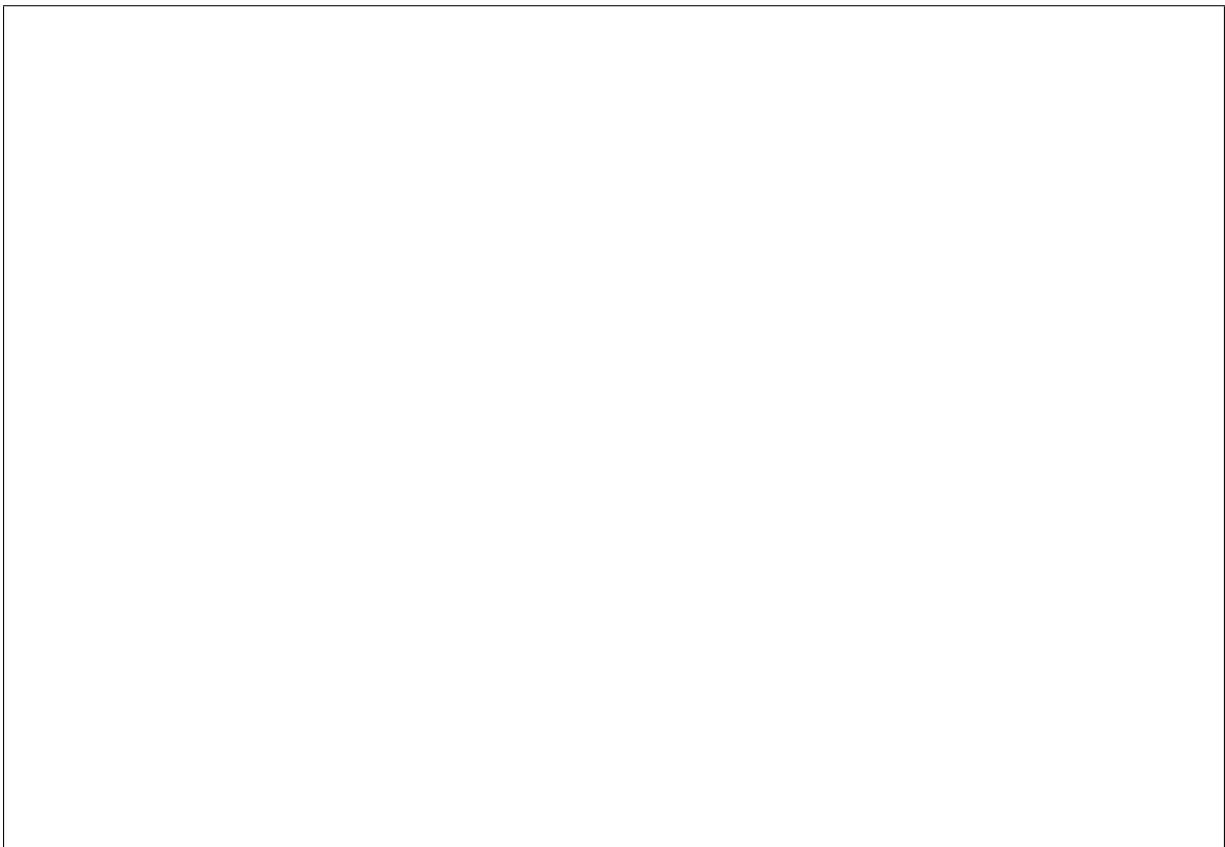
ual cleaning.

### **Prerequisites**

## **Enabling hardware types**

## **Enabling hardware interface**





**Retrieve BIOS settings**



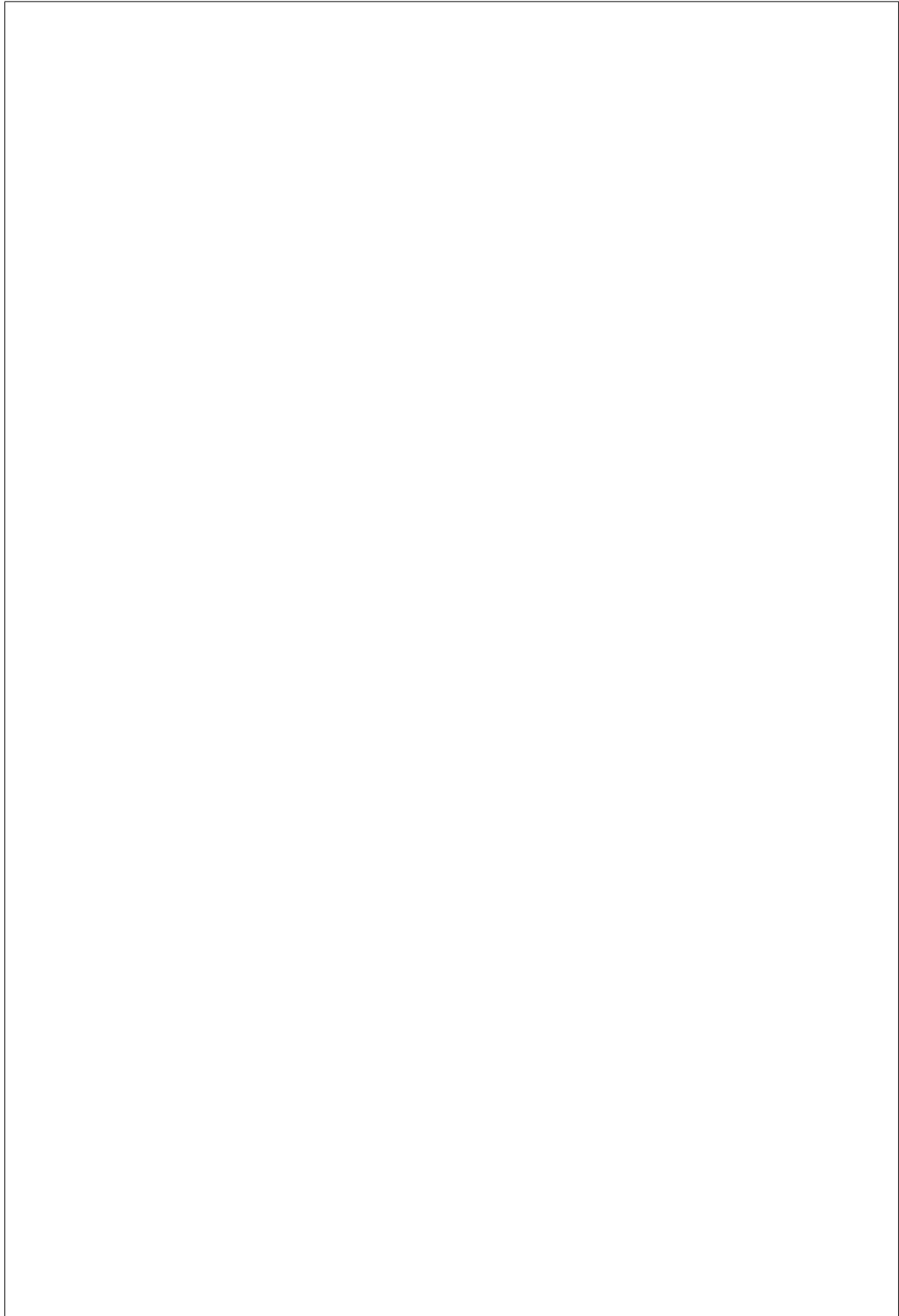


`json` is added as suffix to above command, it returns BIOS settings as following:



(continues on next page)

(continued from previous page)



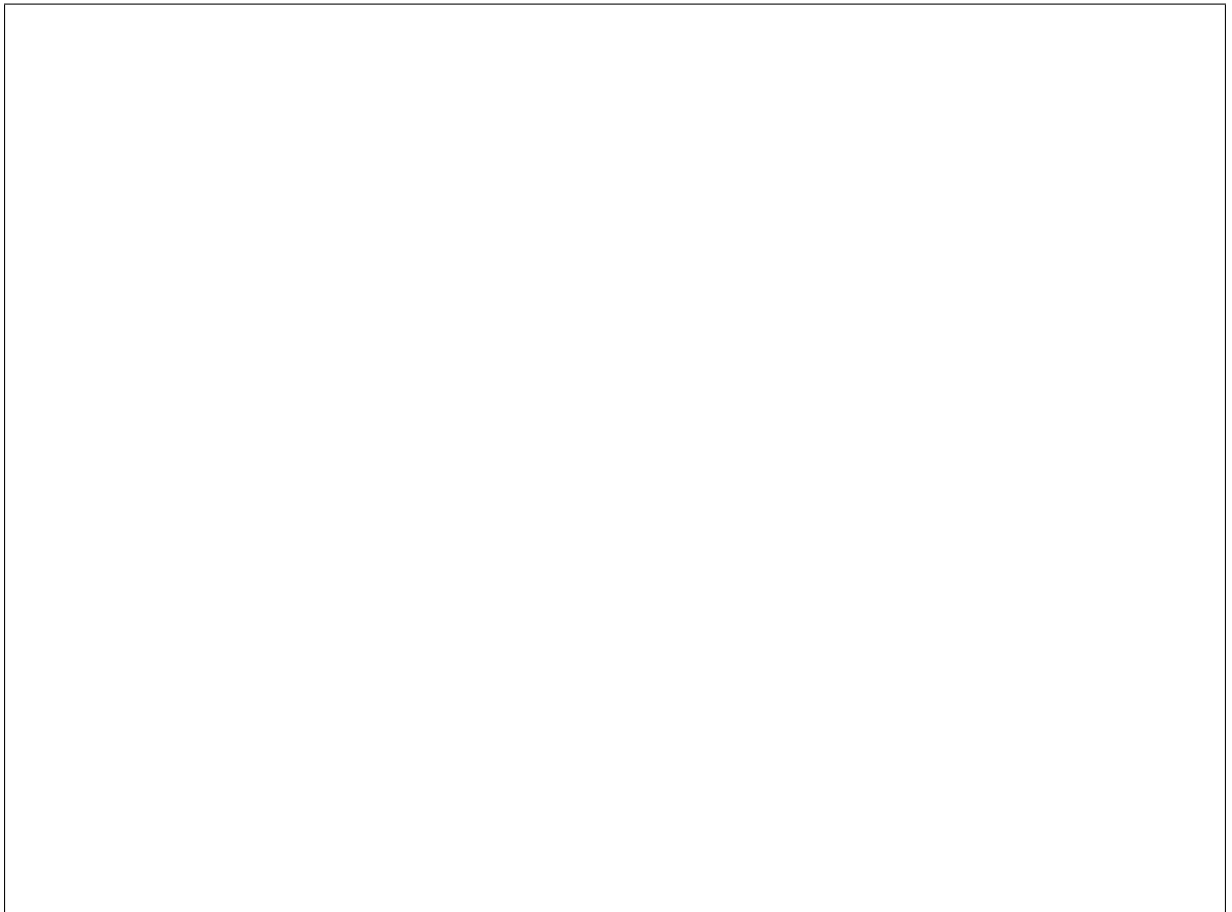
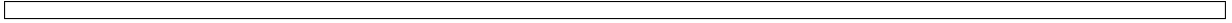
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



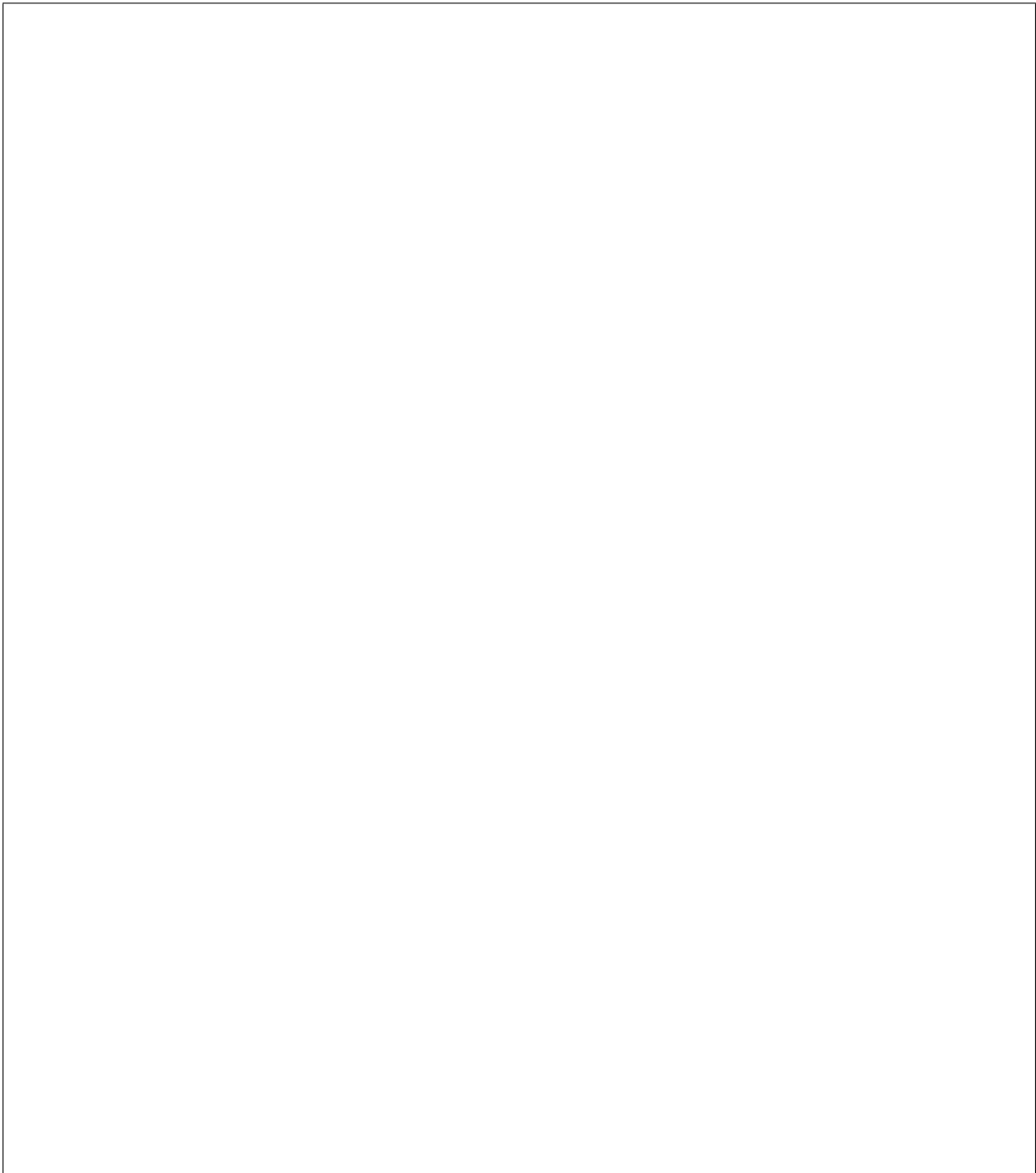
(continues on next page)

(continued from previous page)



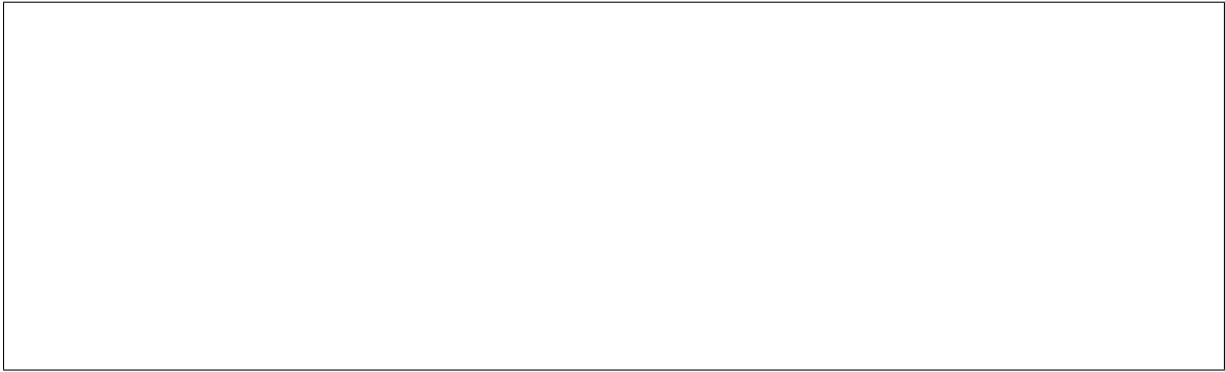
### **Configure BIOS settings**

### **Factory reset**



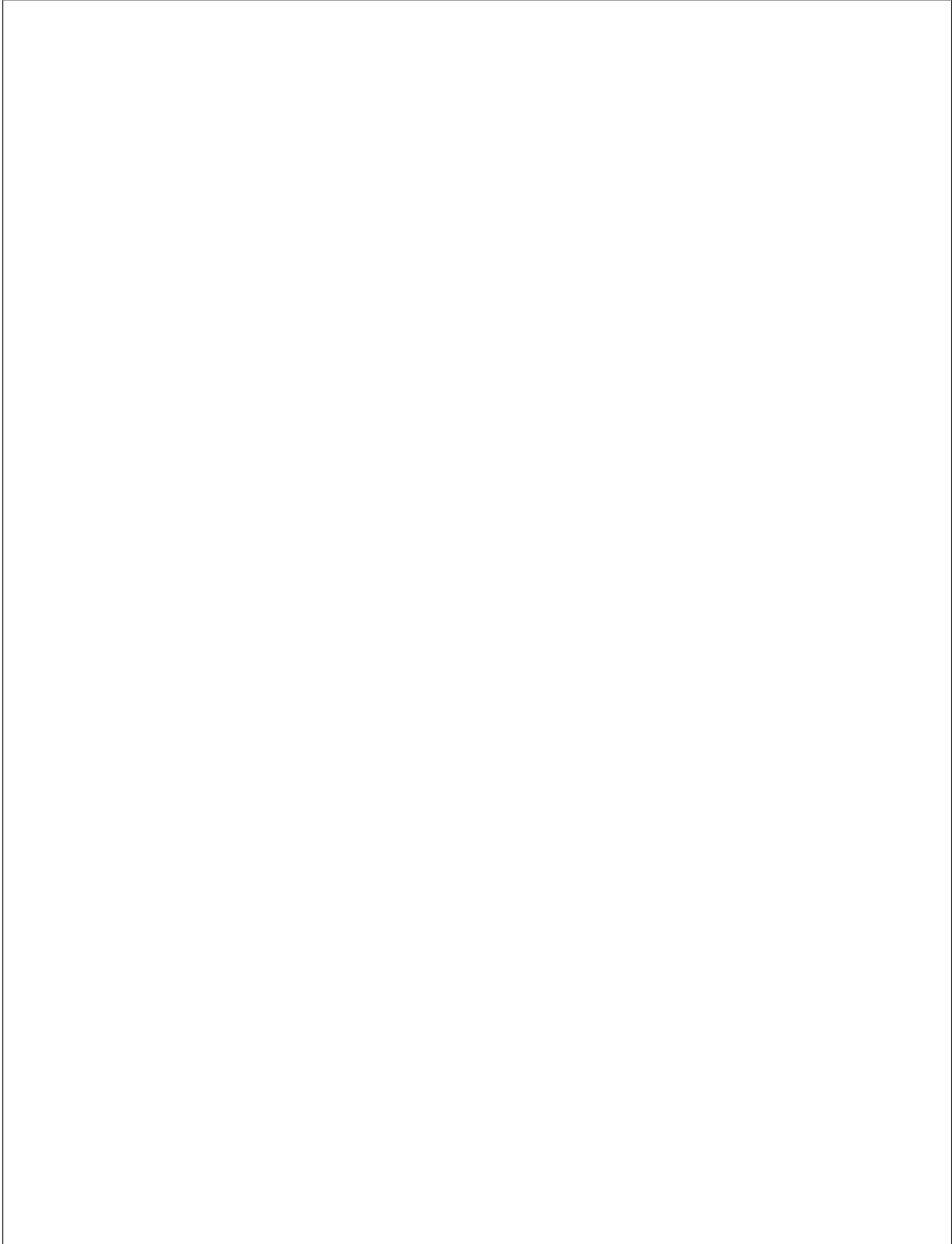
(continues on next page)

(continued from previous page)



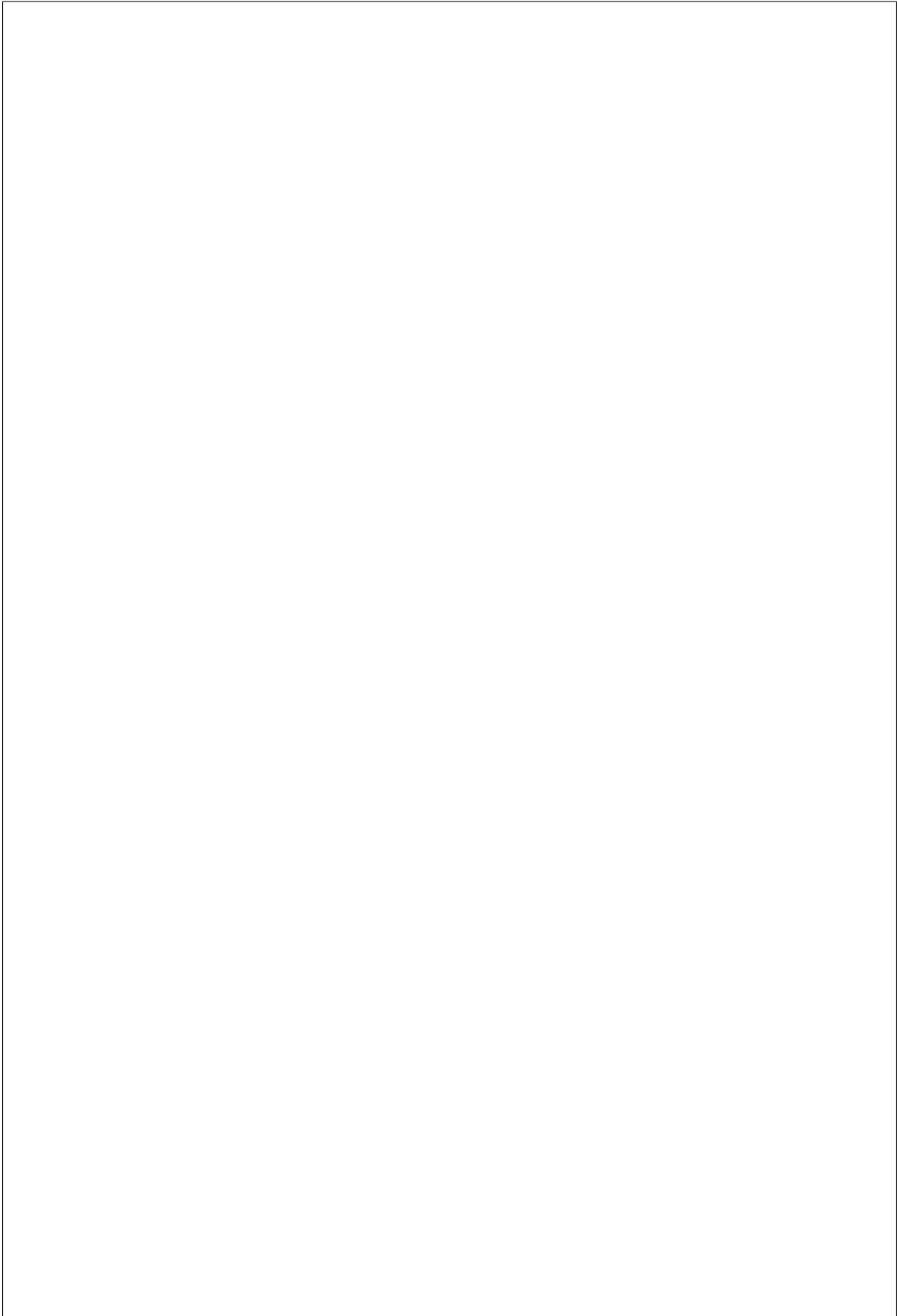
**Apply BIOS configuration**





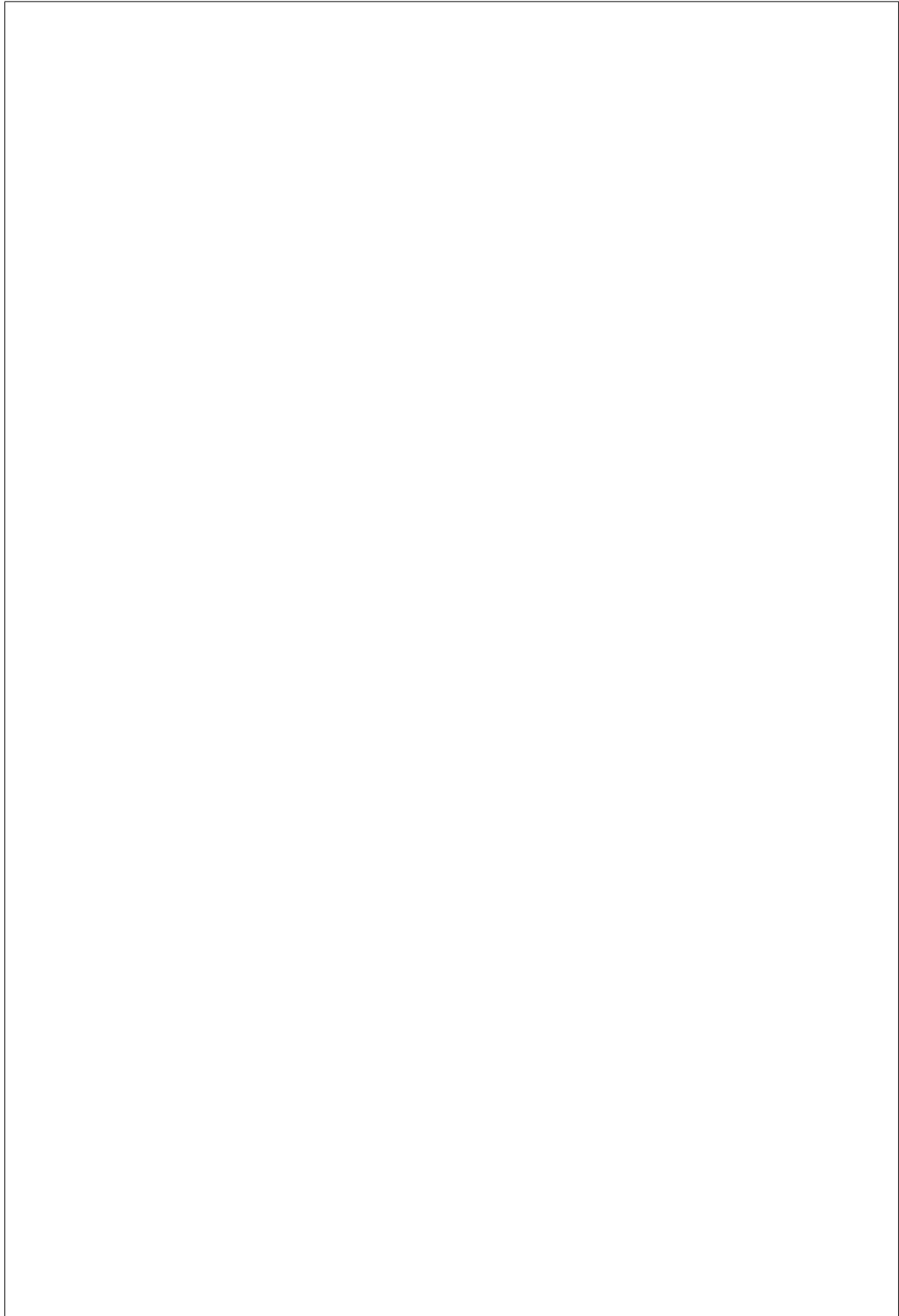
(continues on next page)

(continued from previous page)



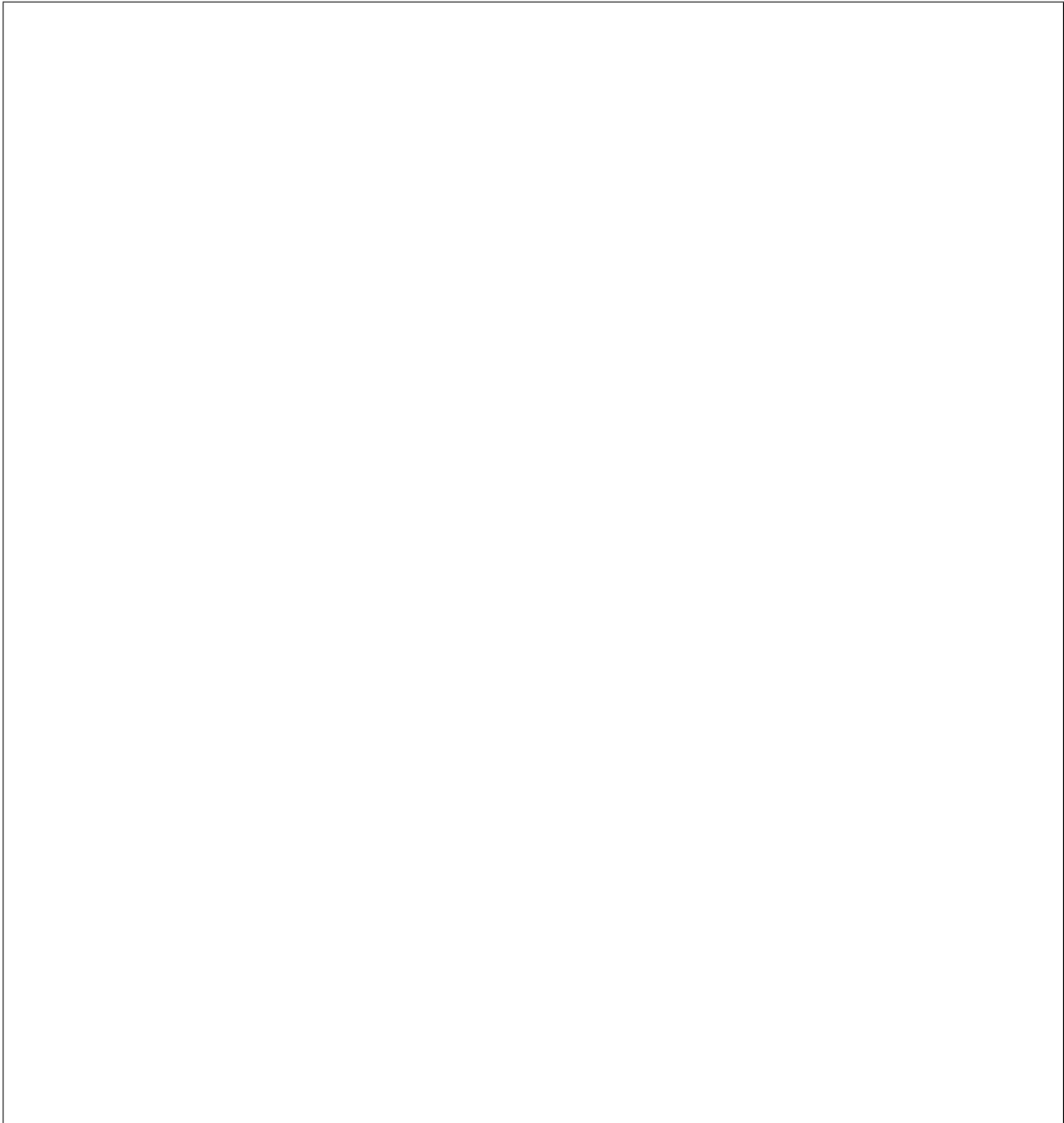
(continues on next page)

(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)

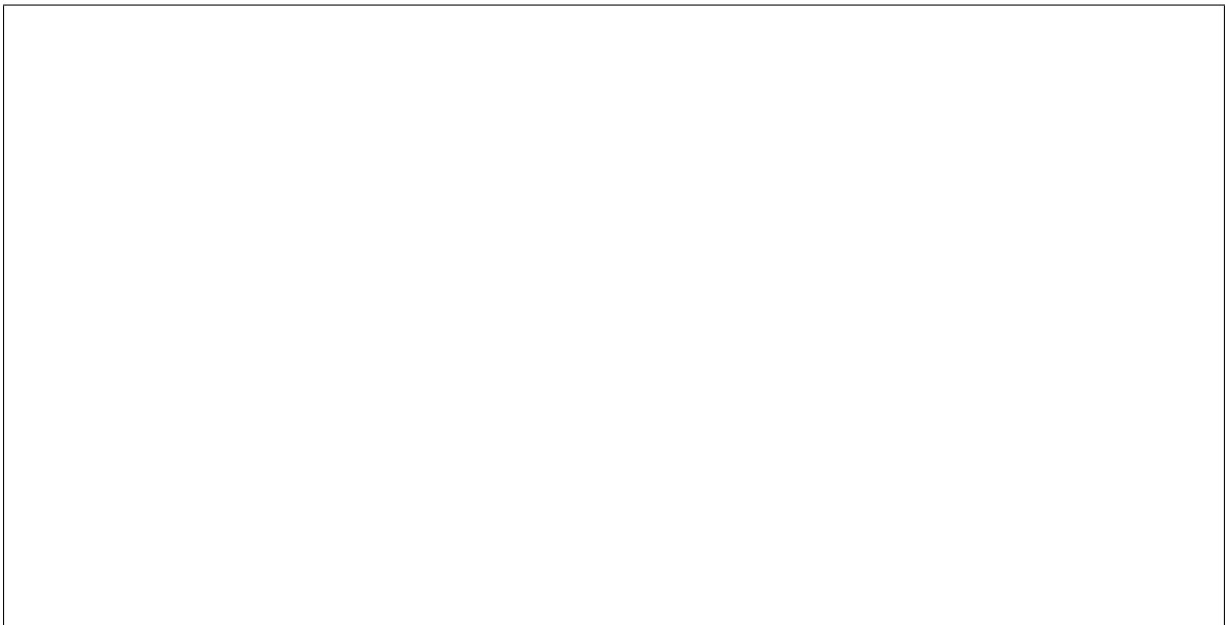












## **Overview**

duced.



## **Prerequisites**





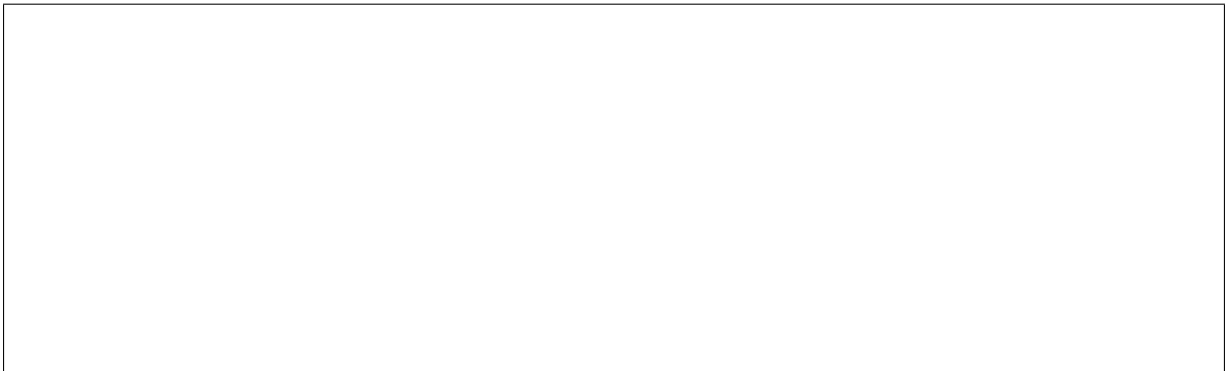
## **Conductor Configuration**





**Node Configuration**

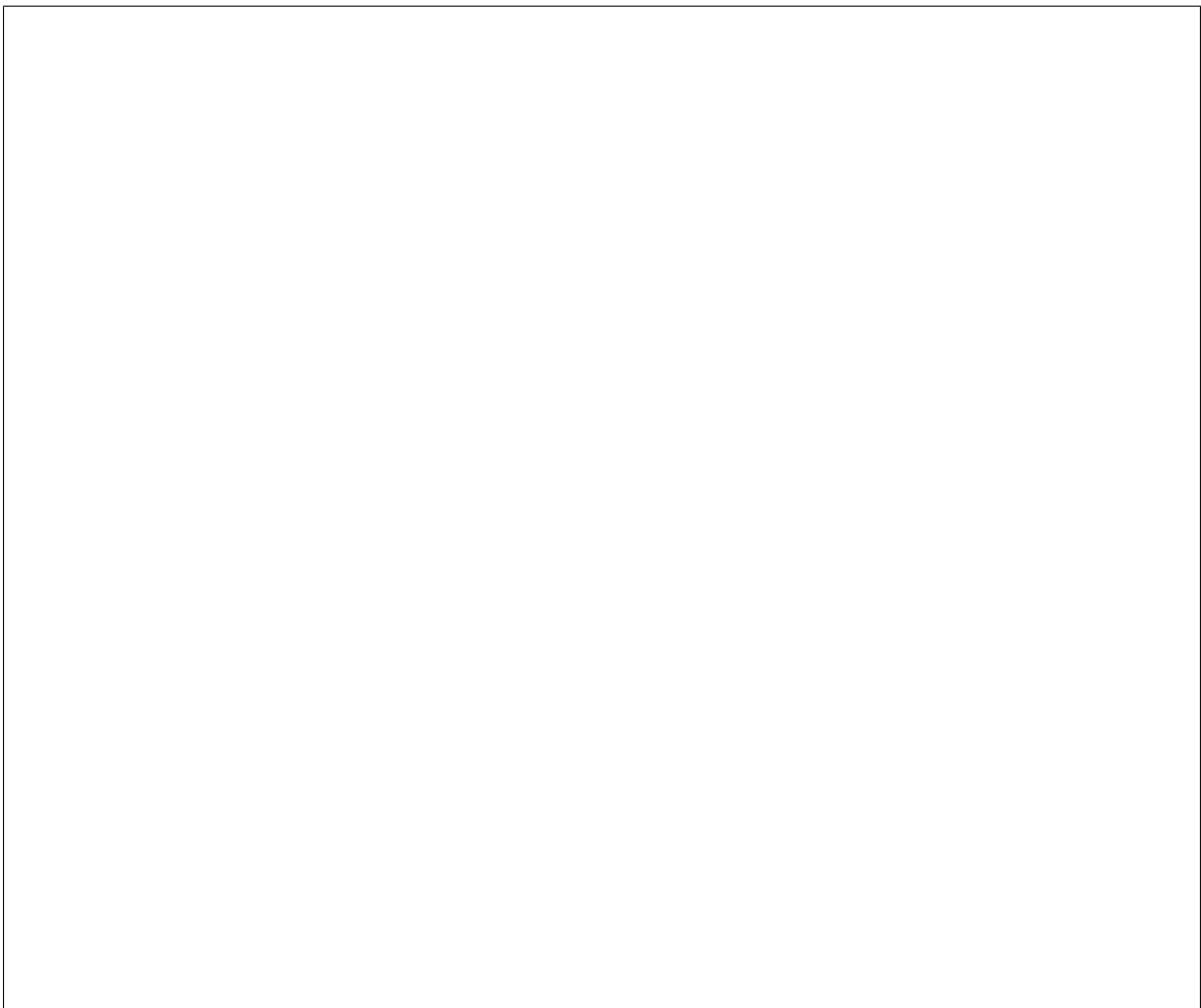
**Storage Interface**



## **iSCSI Configuration**



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

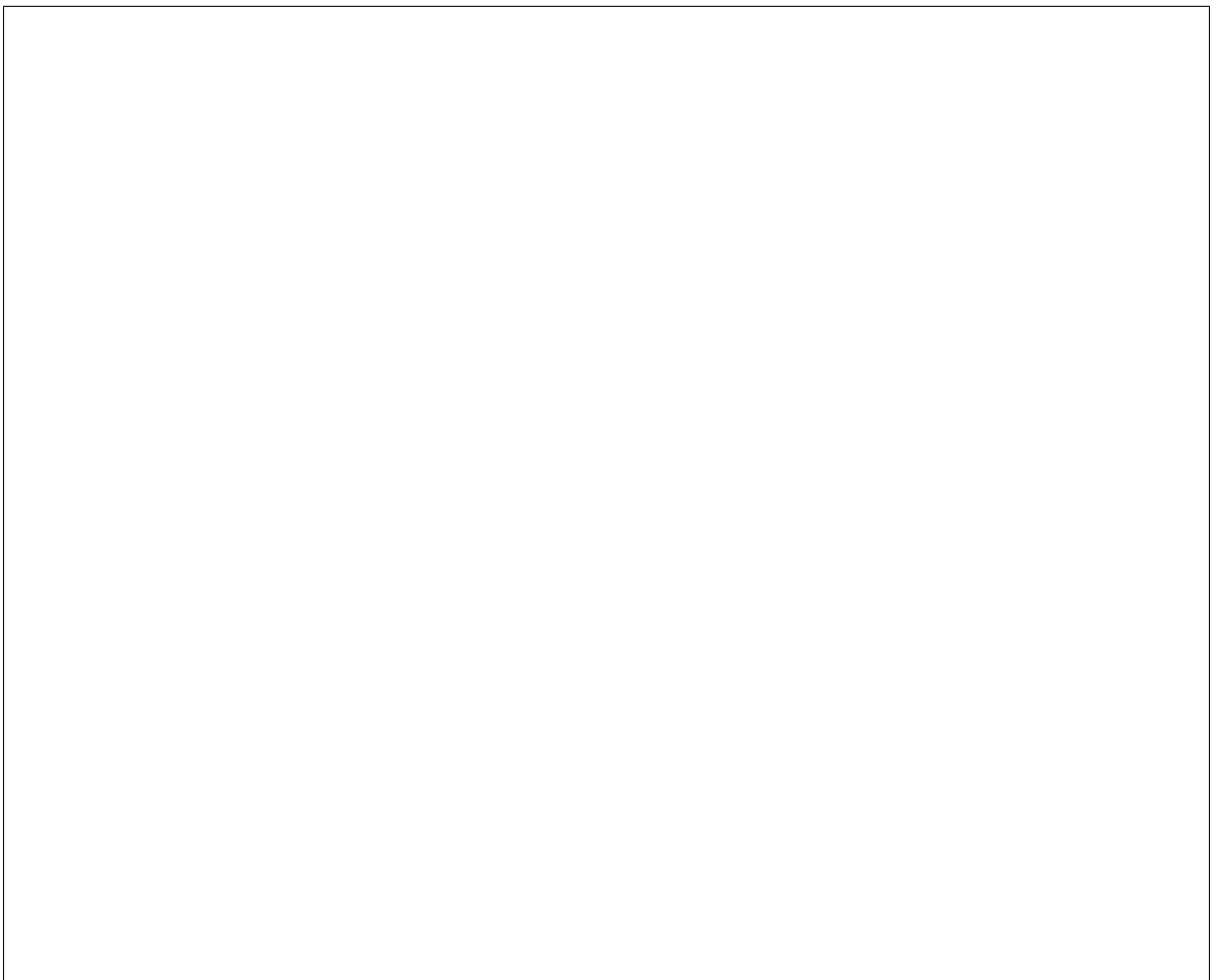


(continues on next page)

(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

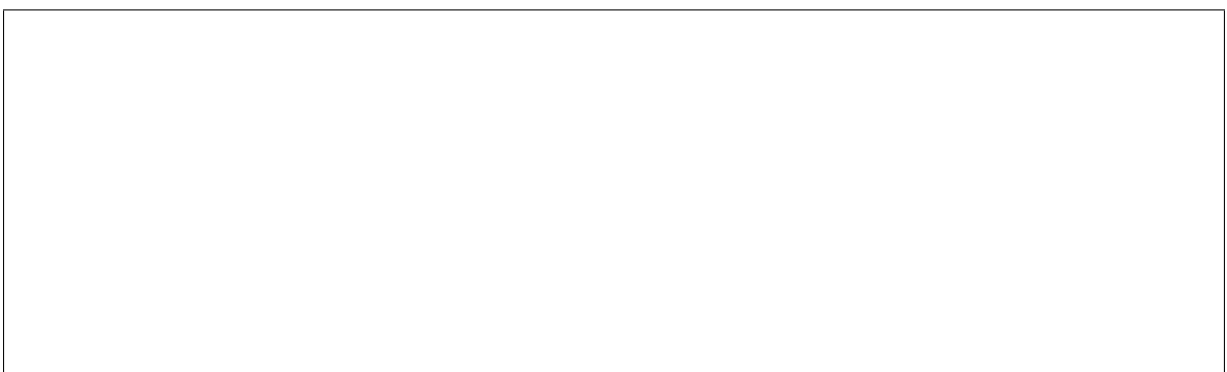


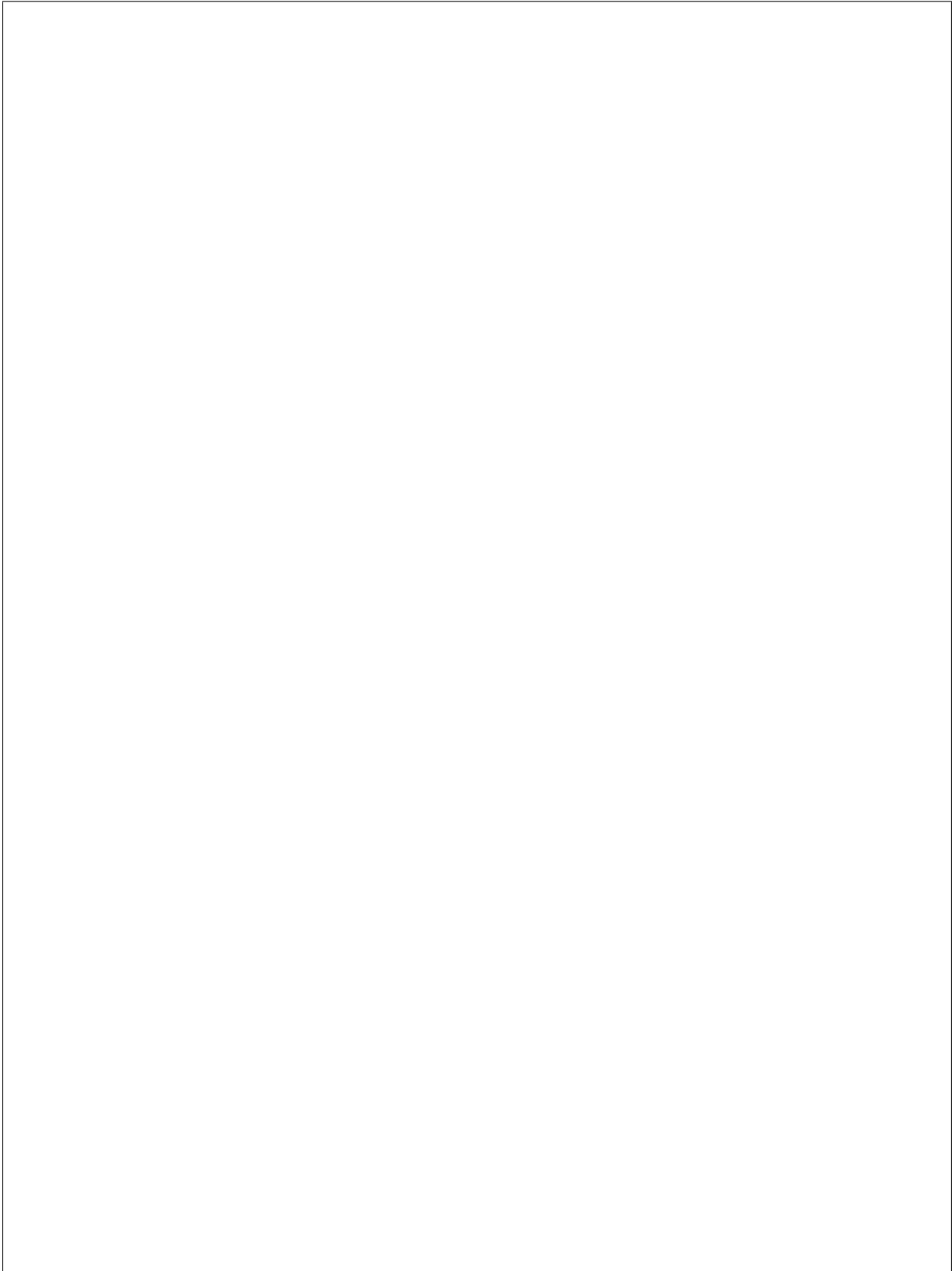


age interface which does not contain logic to determine if the node should or could boot from a remote volume.



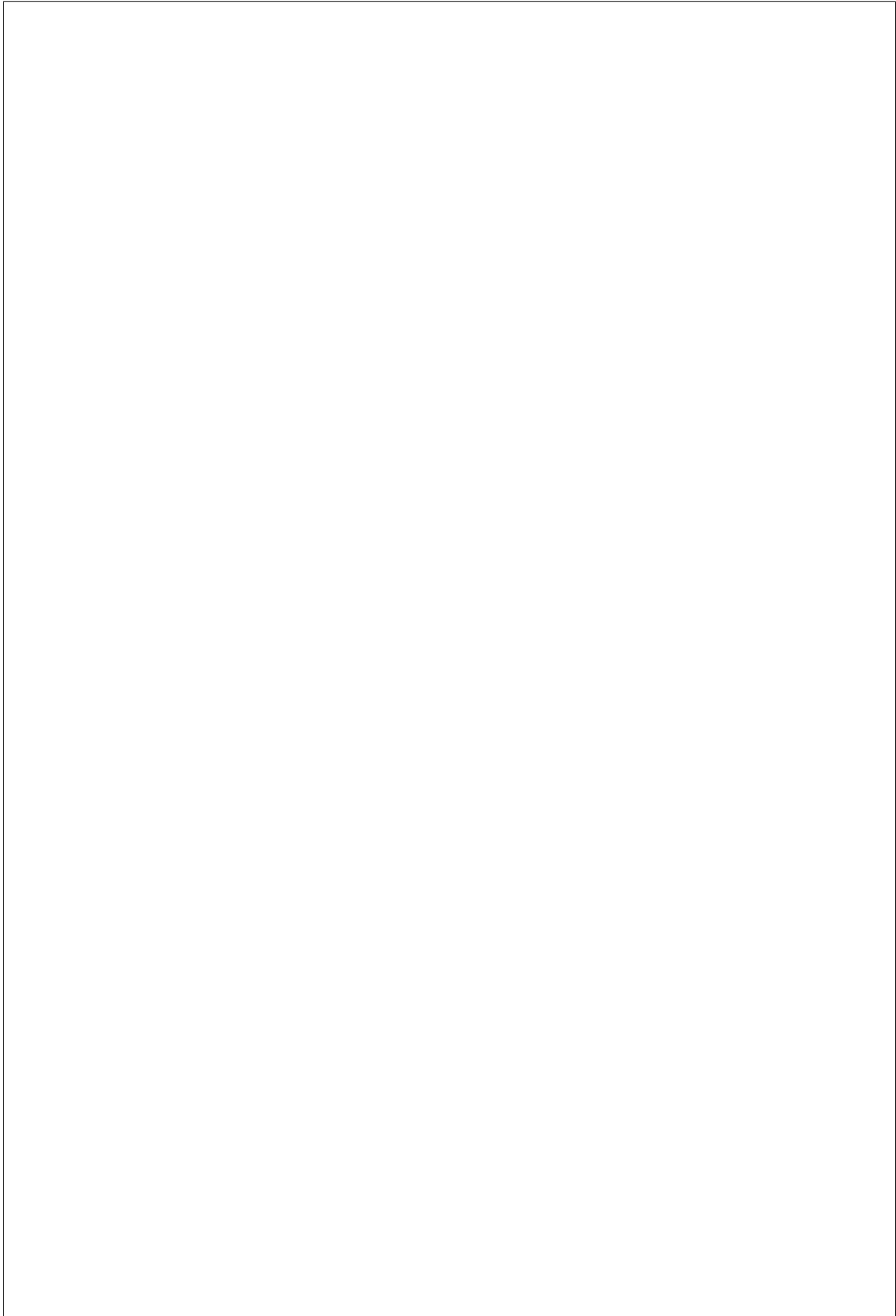
scenario.





(continues on next page)

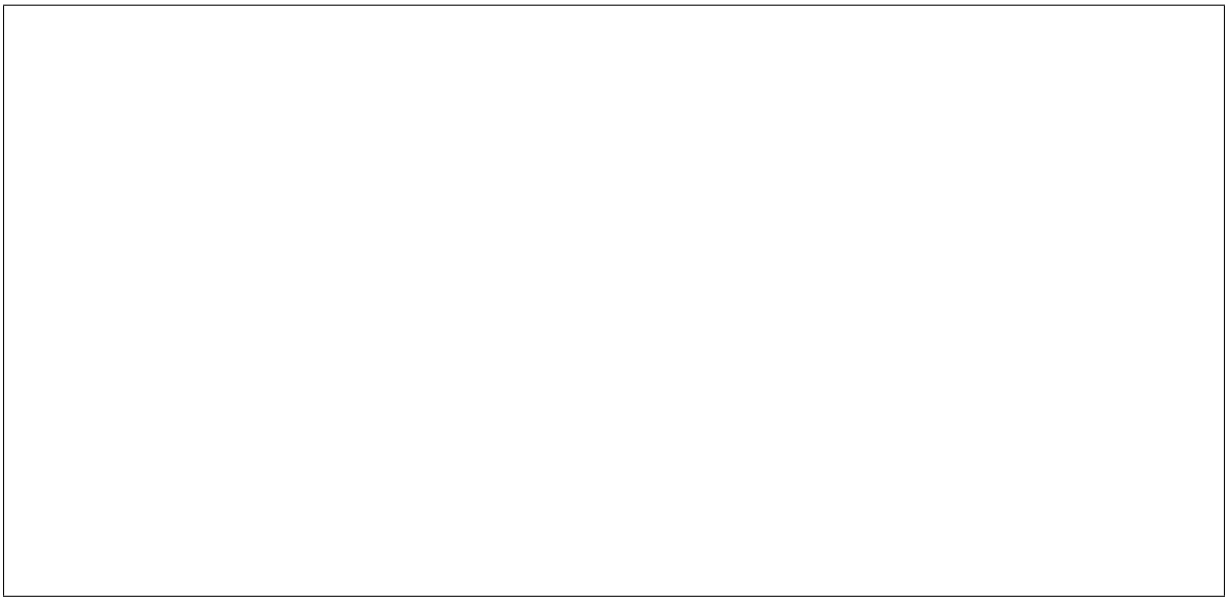
(continued from previous page)



(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**

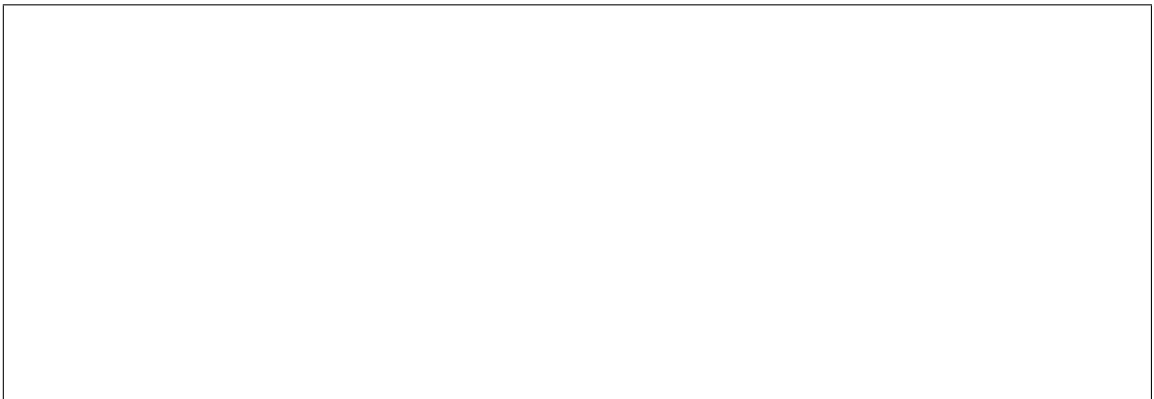




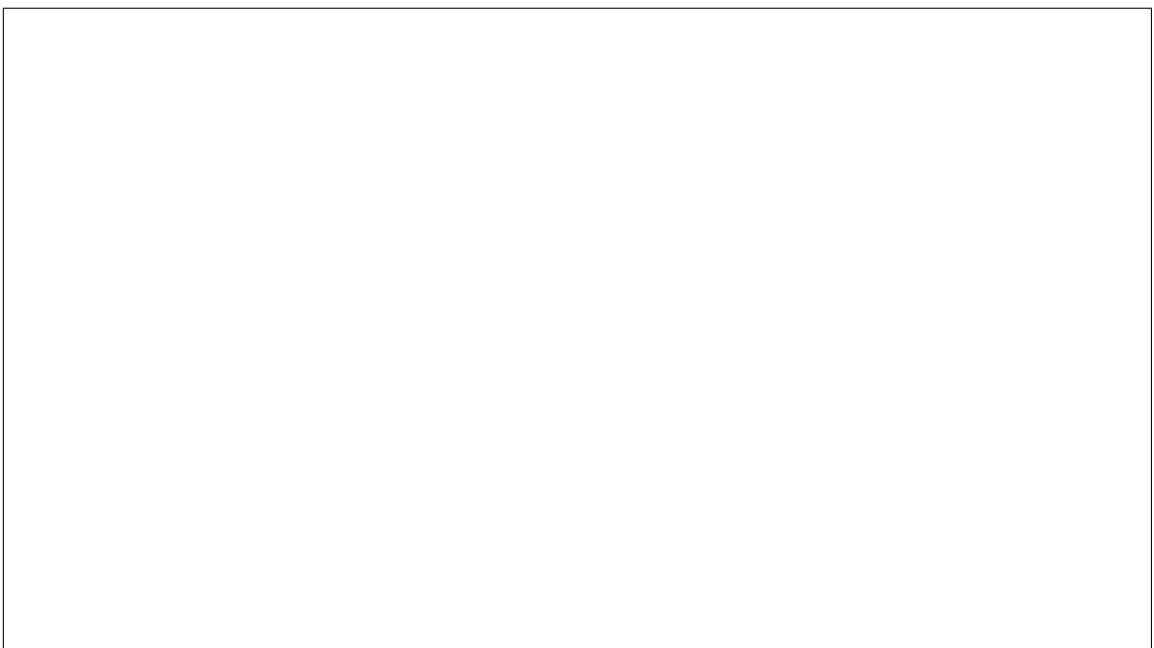




vices ML2 driver:

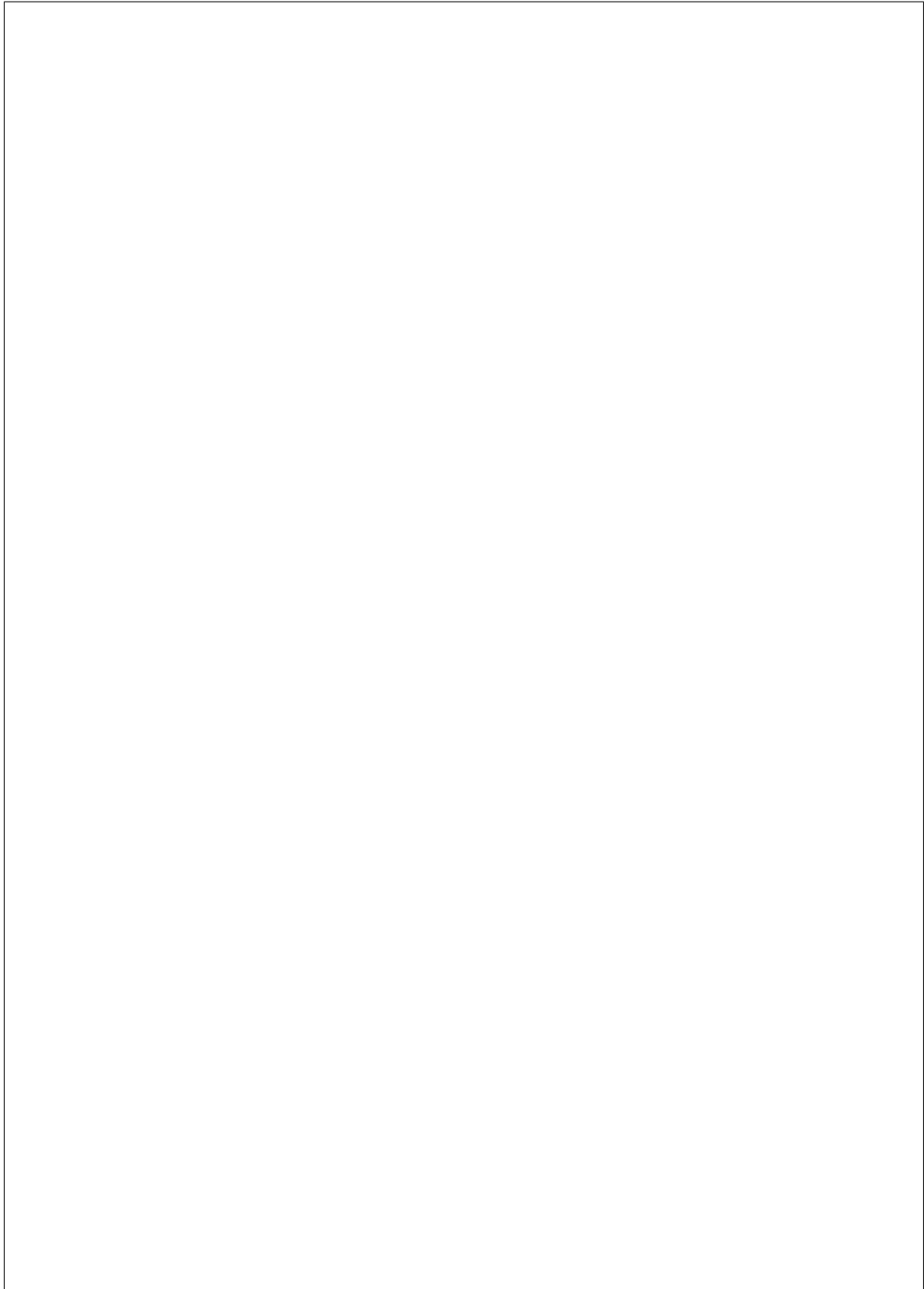






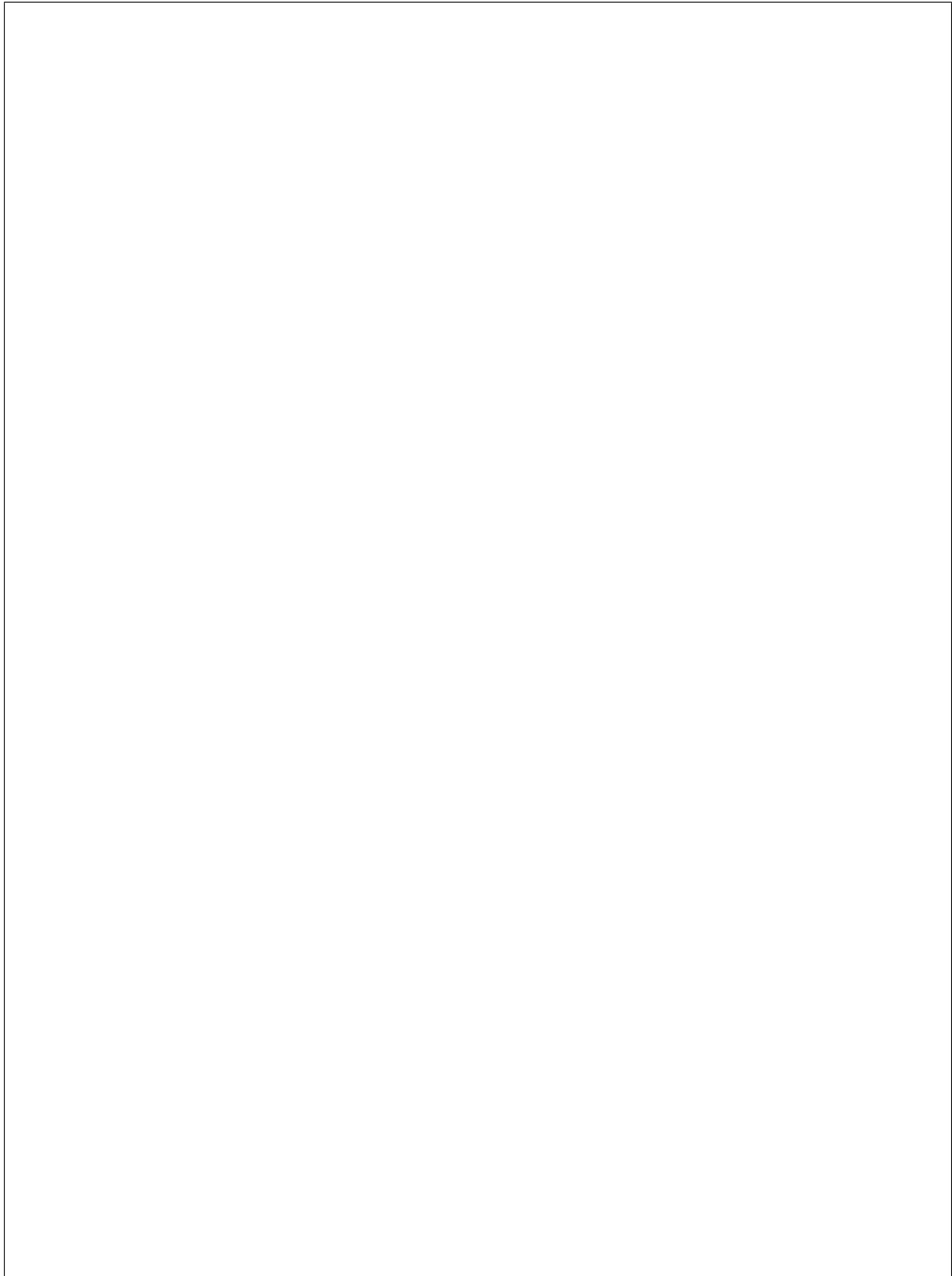
(continues on next page)

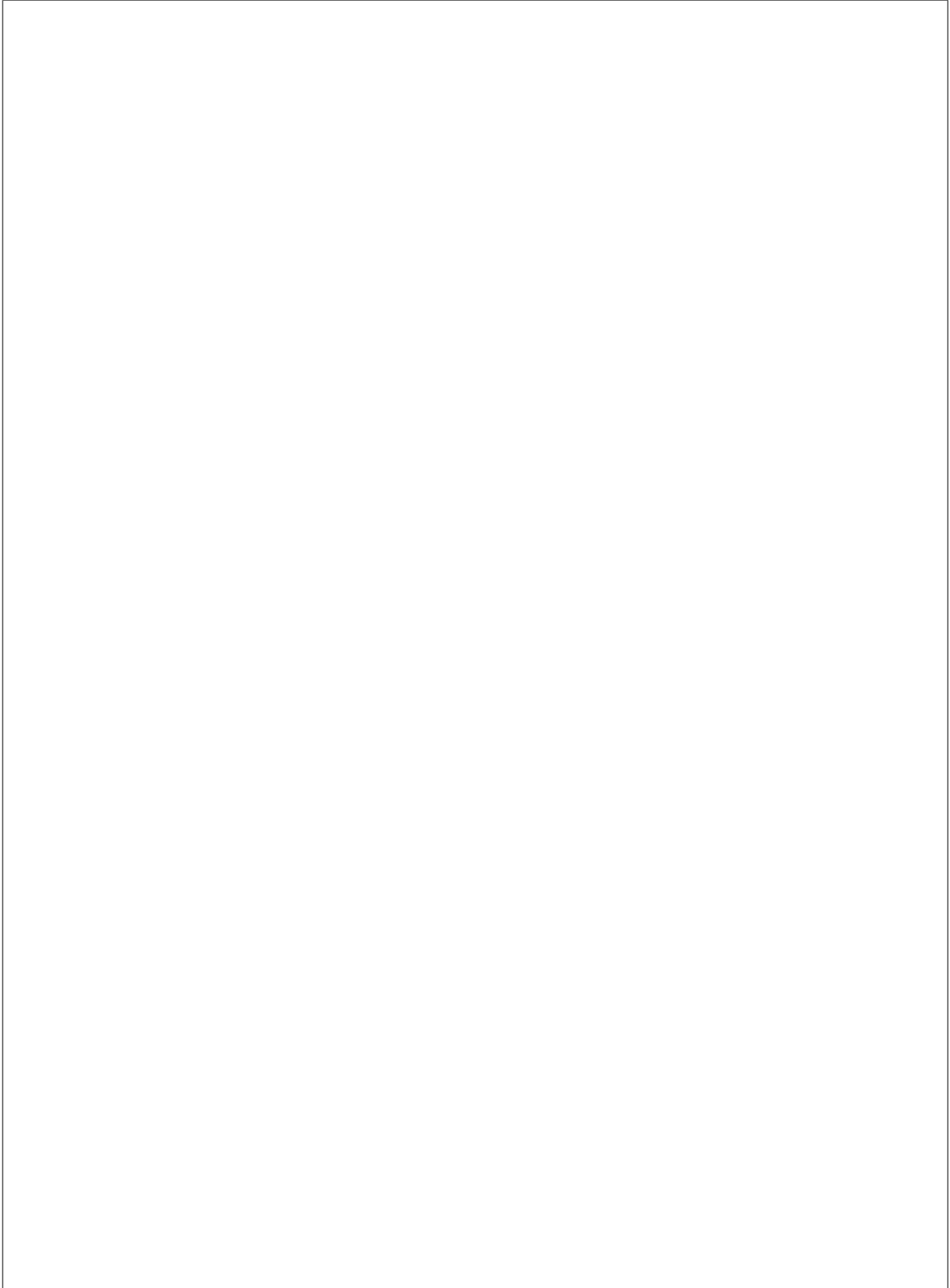
(continued from previous page)



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)







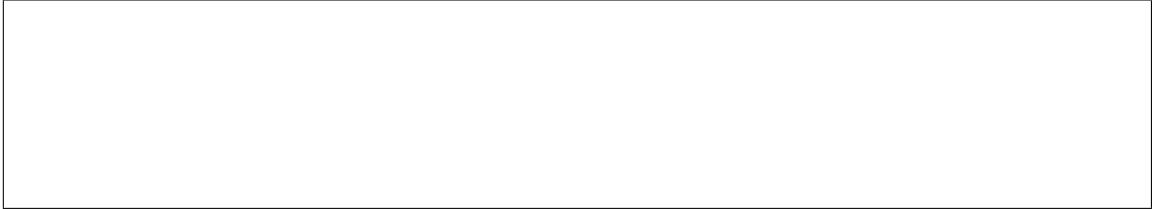
## **Configuring the Networking service**

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.

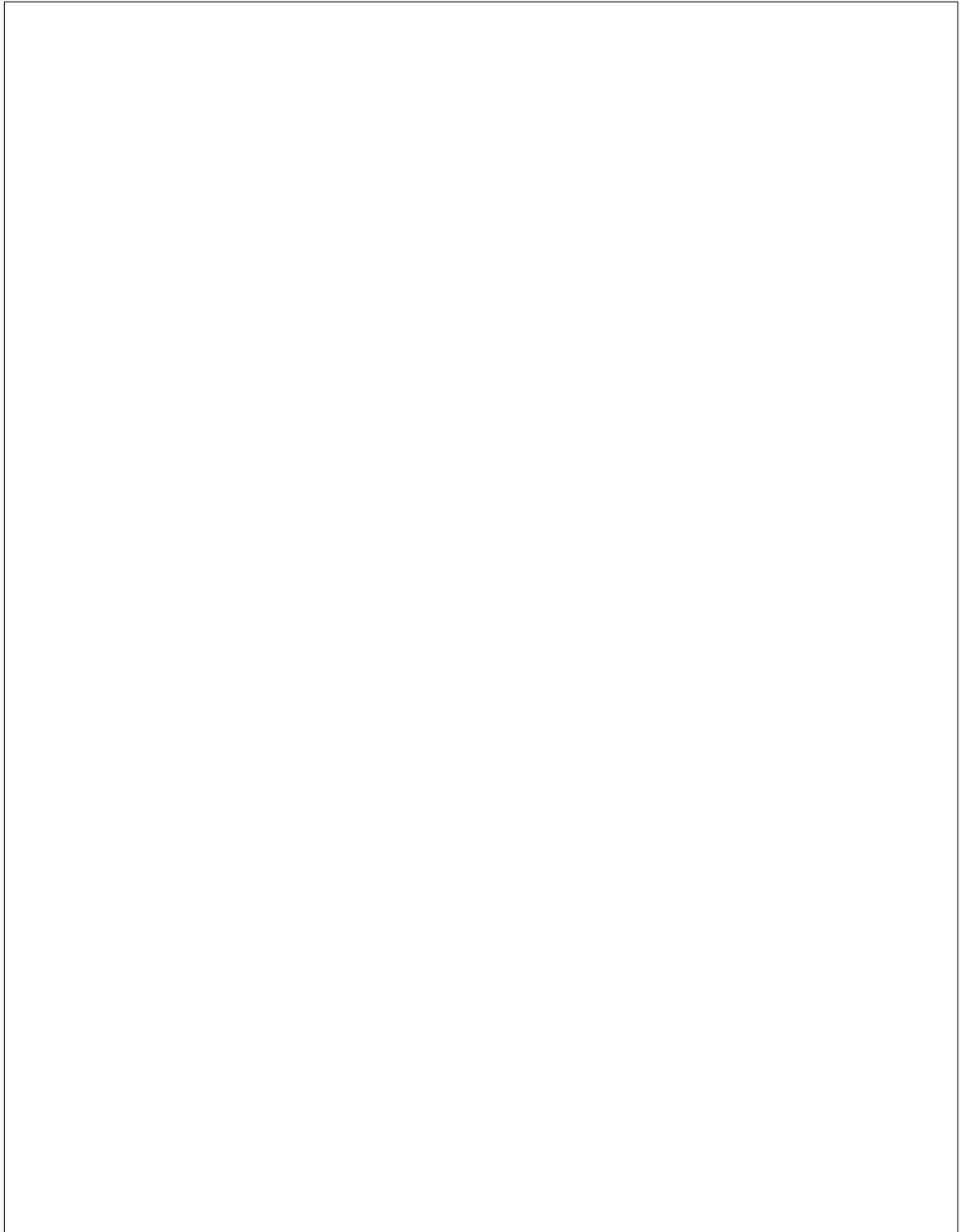
bers to be used by themselves, you need to set port groups `standalone_ports_supported` value to be `False` in `ironic`, as it is `True` by default.

## **Physical networks**



## **Port groups configuration in the Bare Metal service**



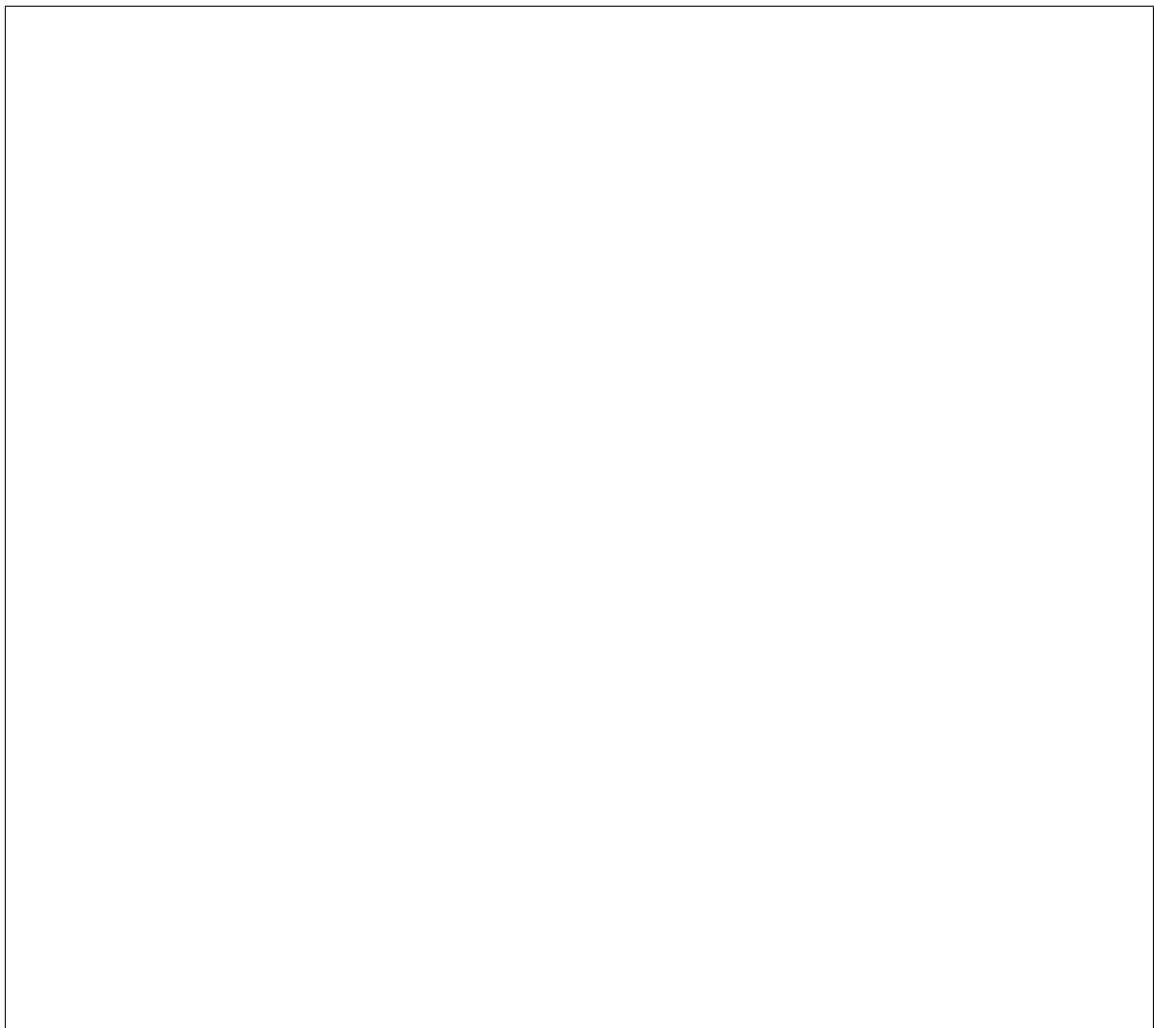






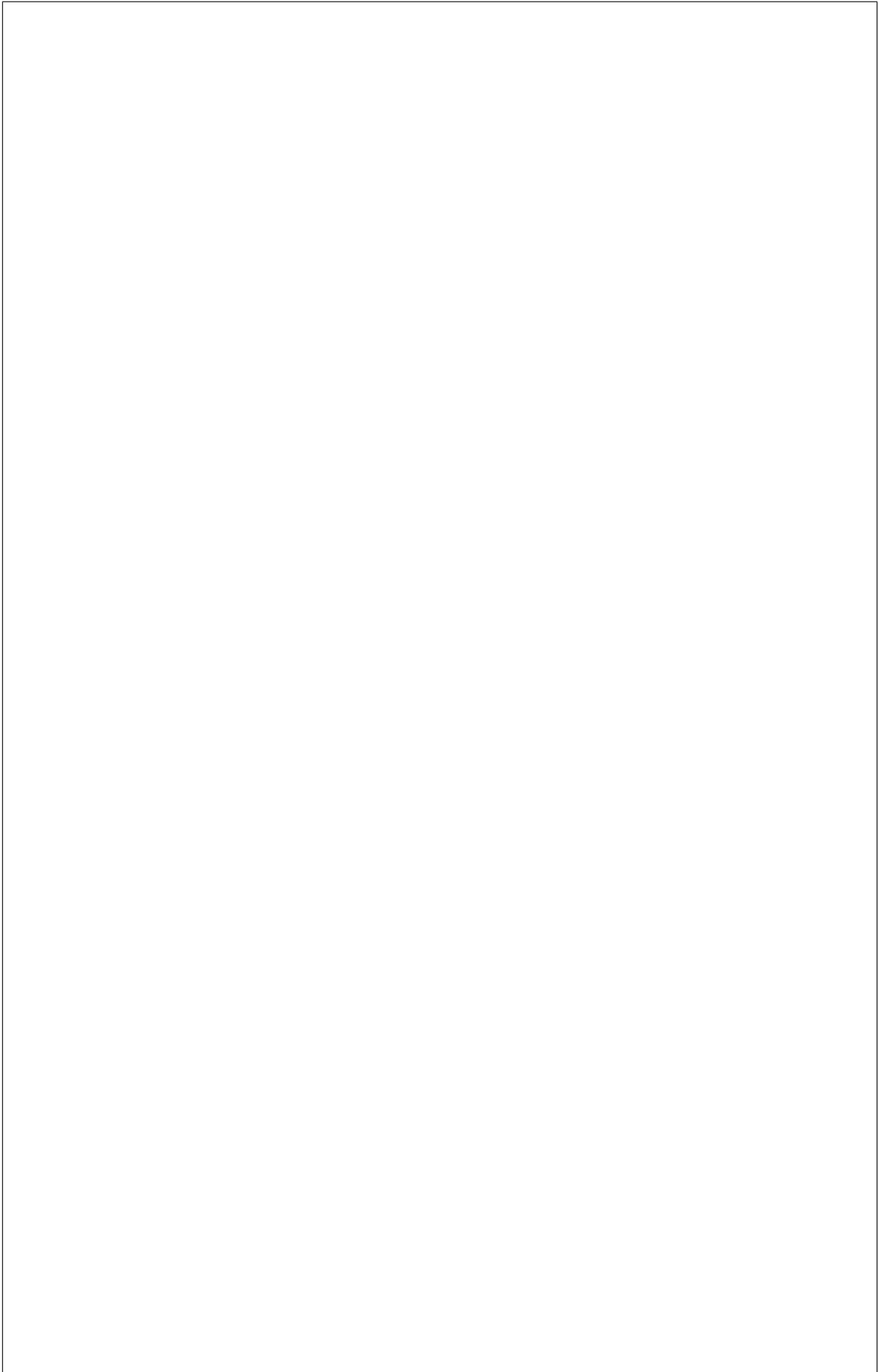
not be configured.

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.



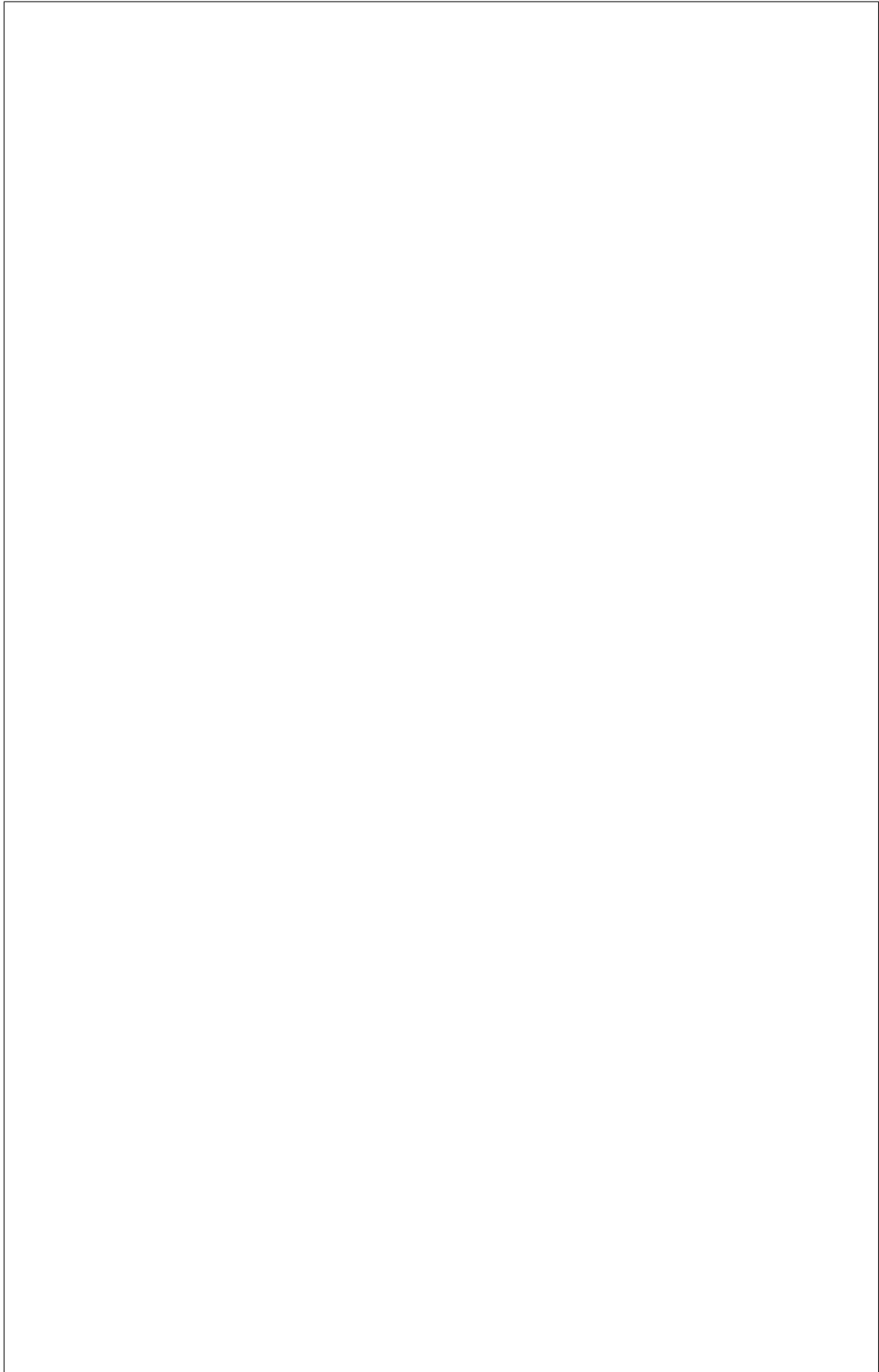
(continues on next page)

(continued from previous page)



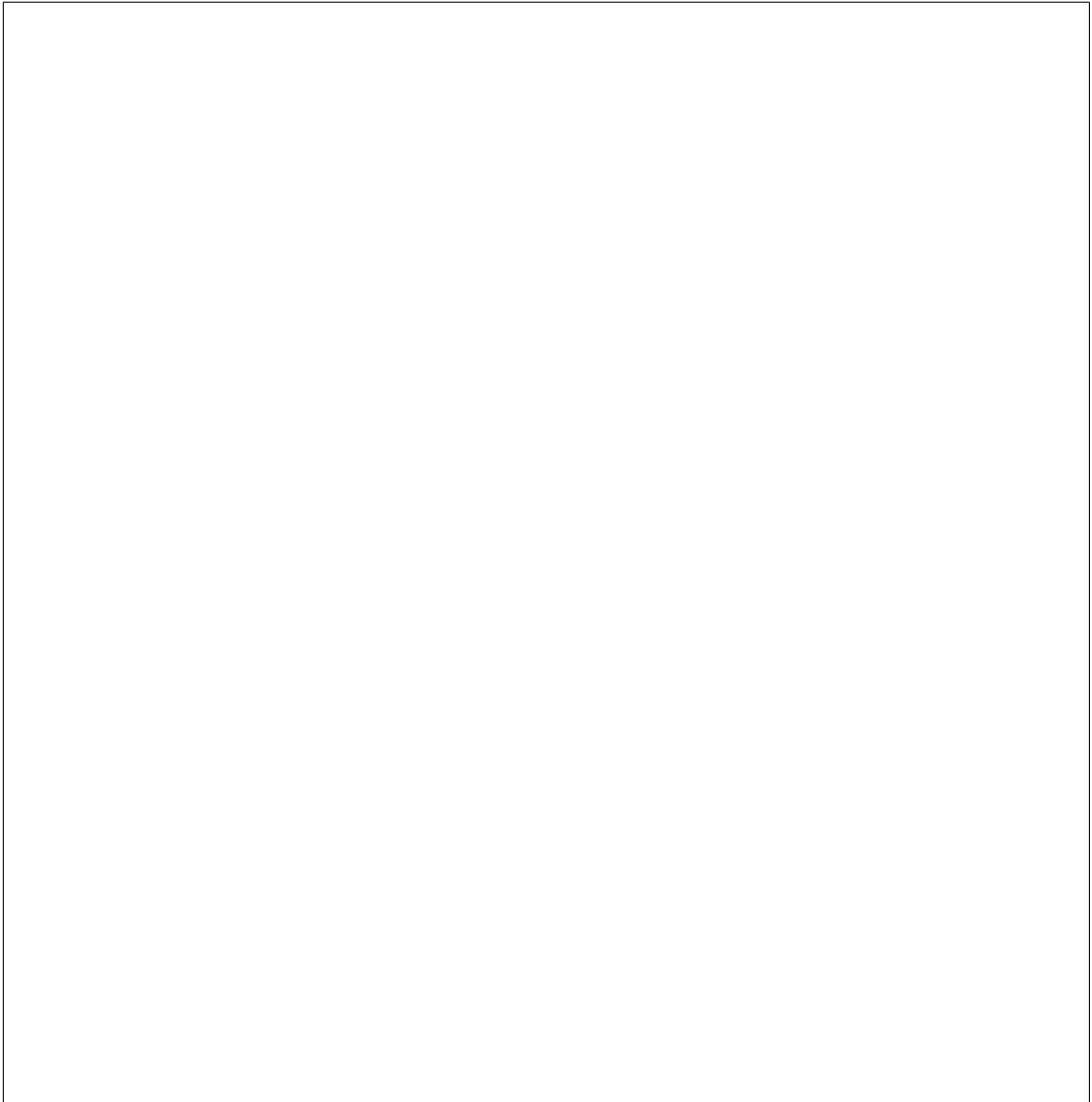
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

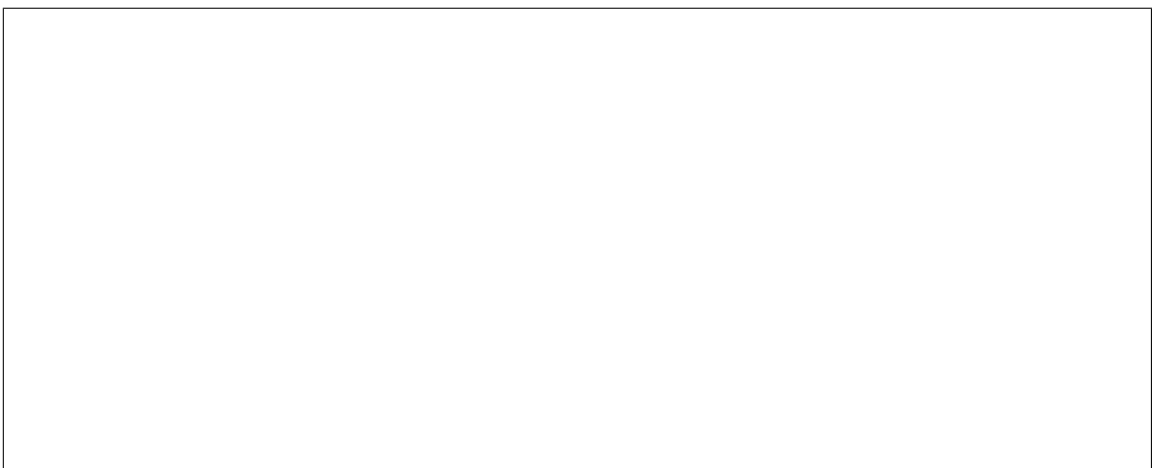






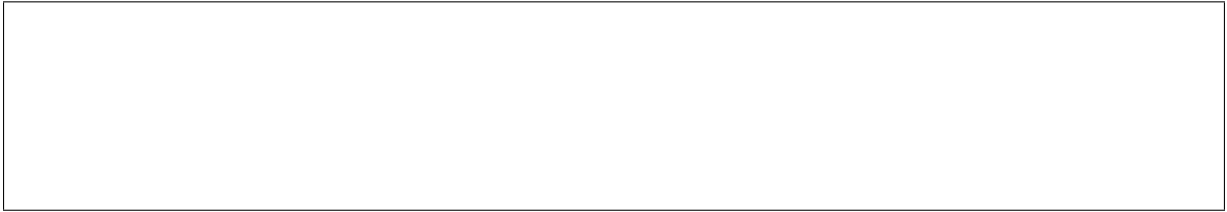
Port groups that dont have any ports will be ignored.

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**



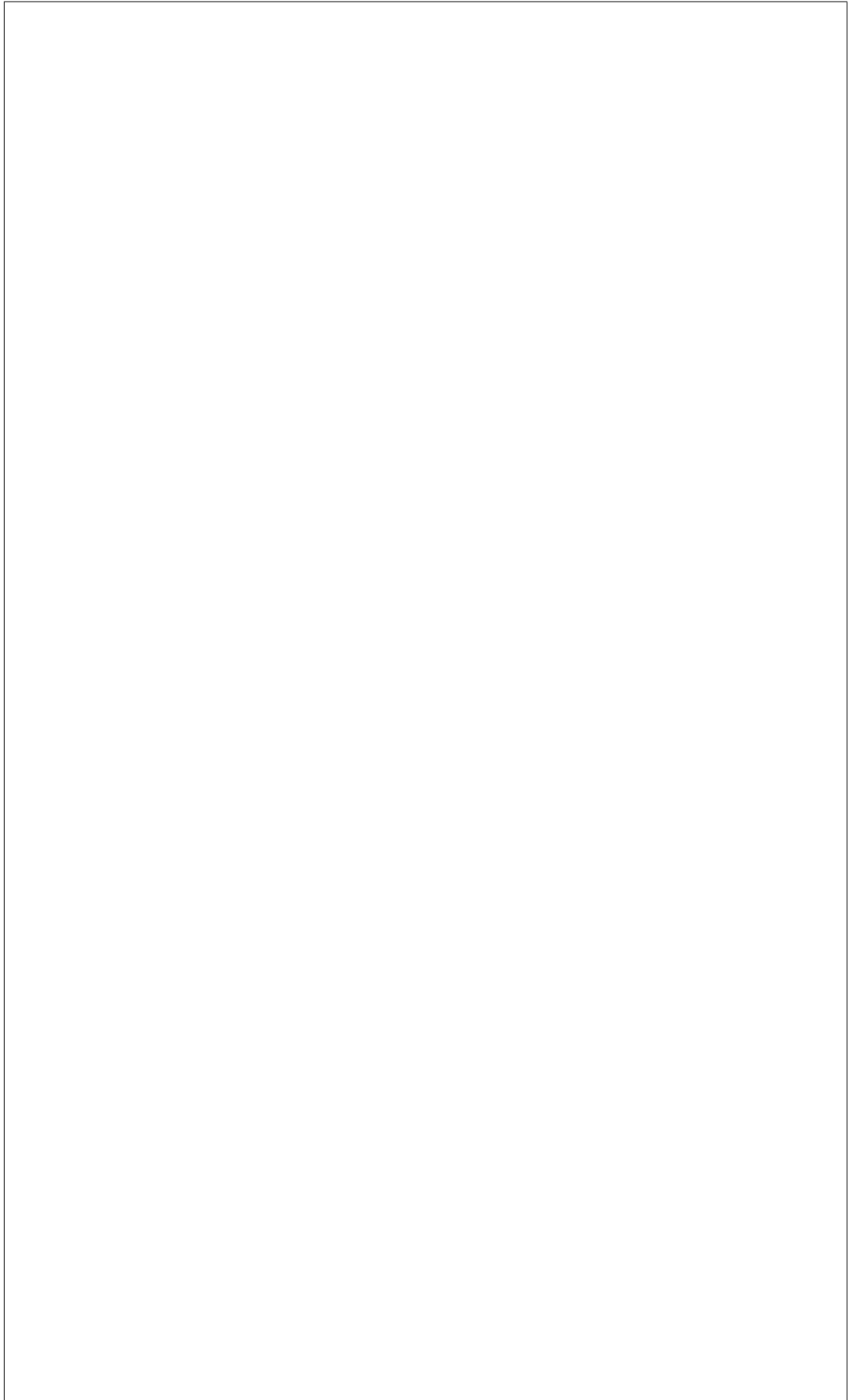






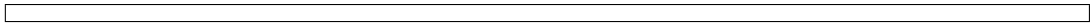
(continues on next page)

(continued from previous page)



(continues on next page)

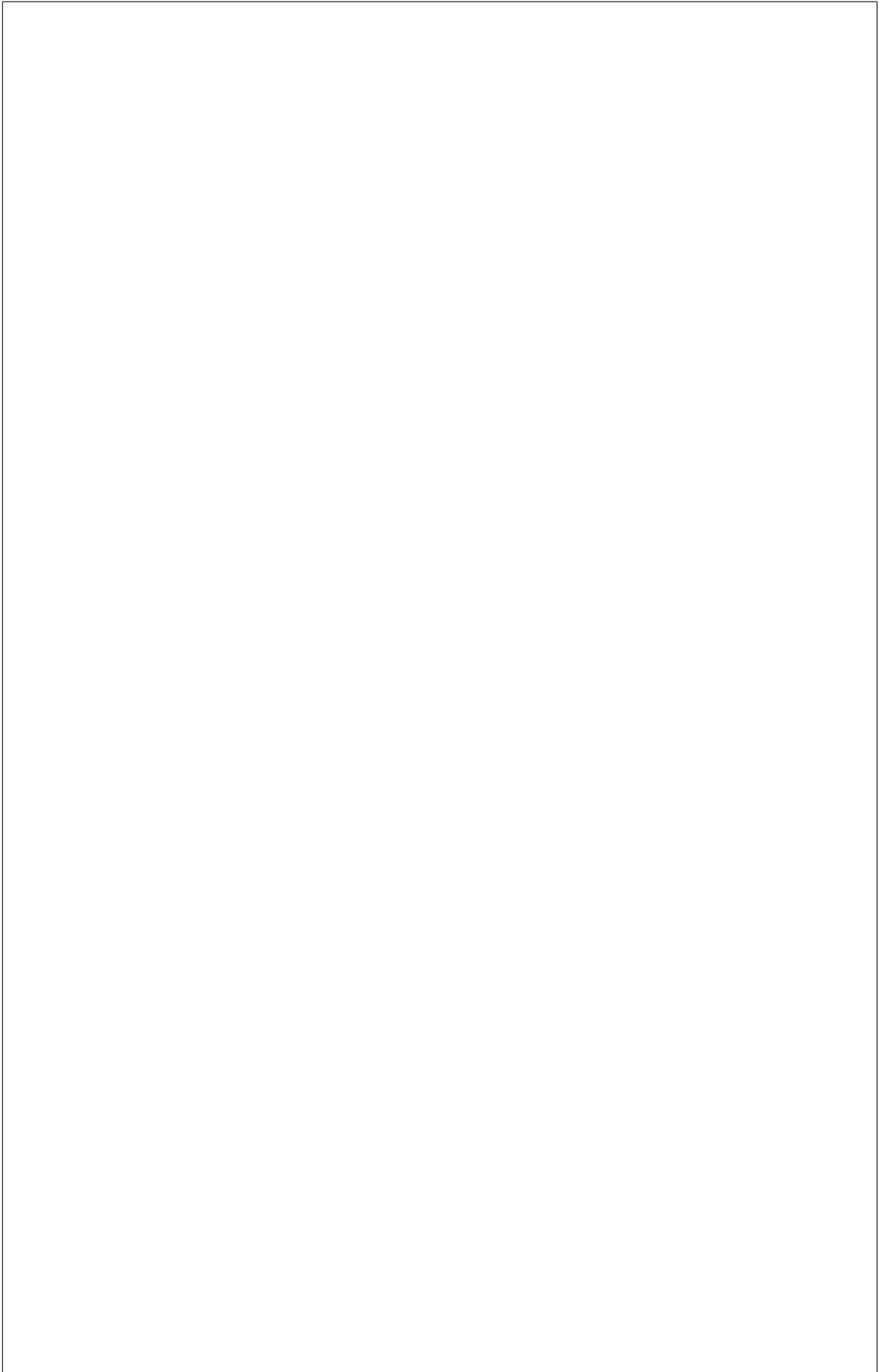
(continued from previous page)



(continues on next page)

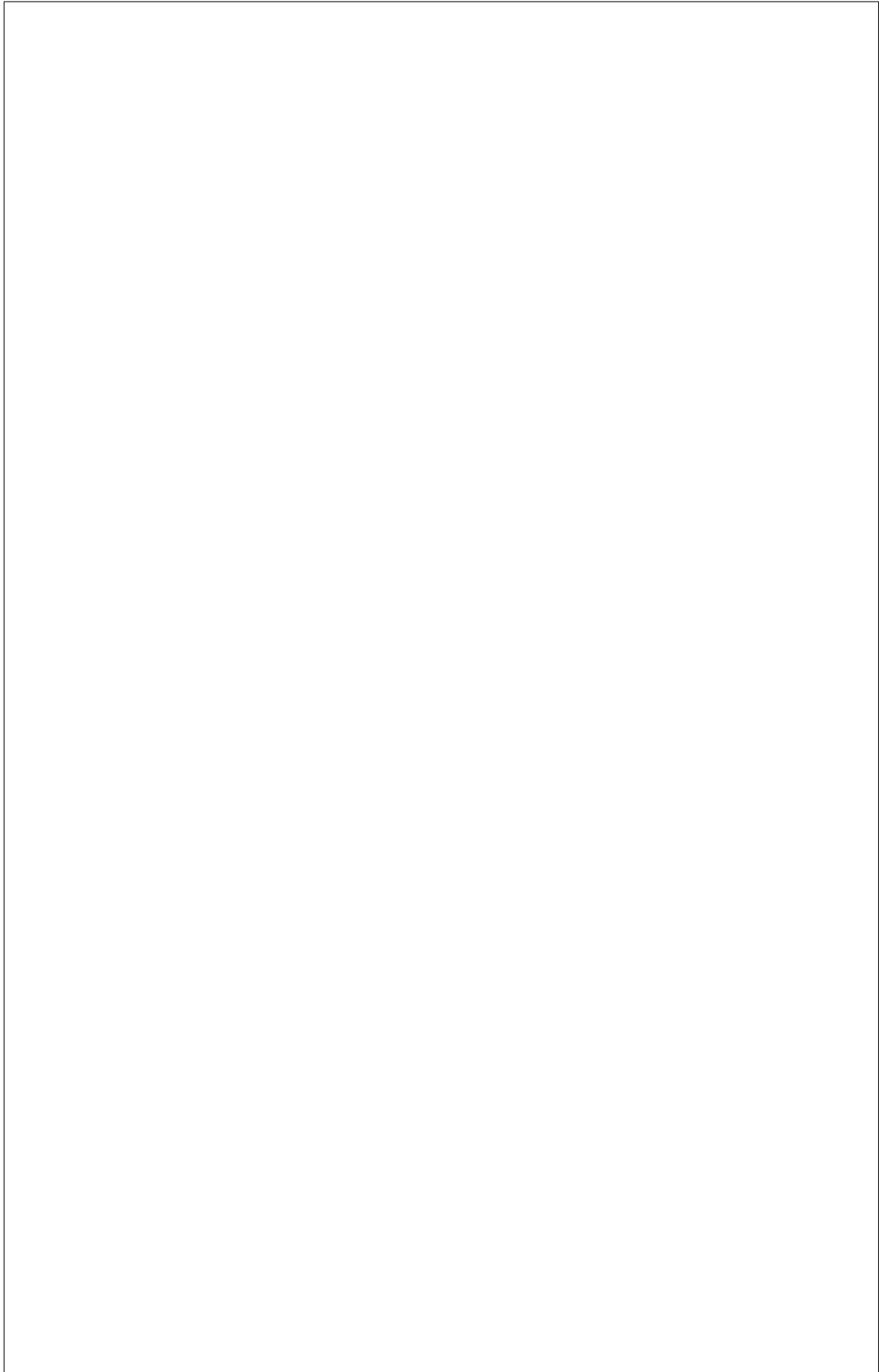


(continued from previous page)



(continues on next page)

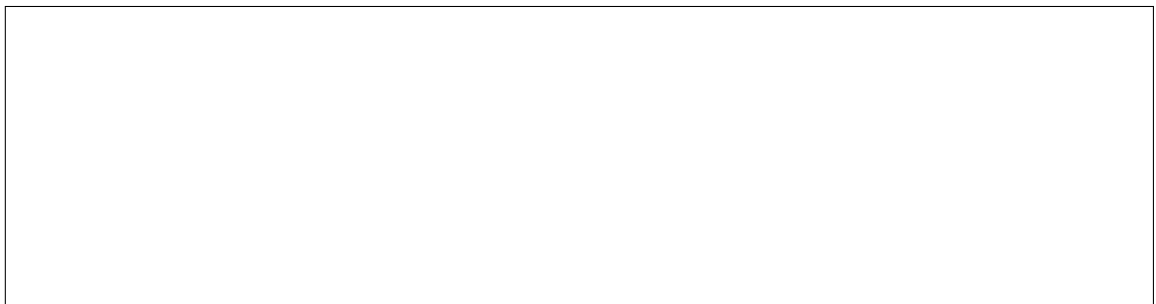
(continued from previous page)

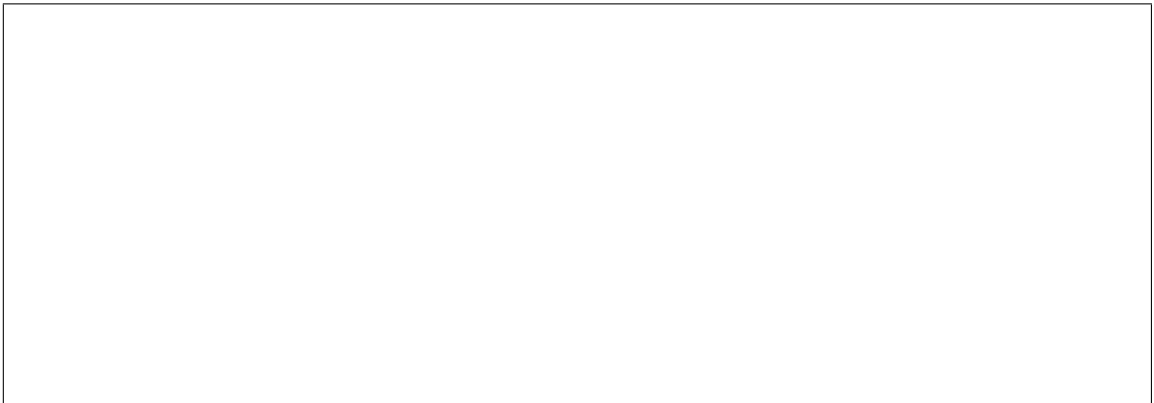


(continues on next page)

(continued from previous page)



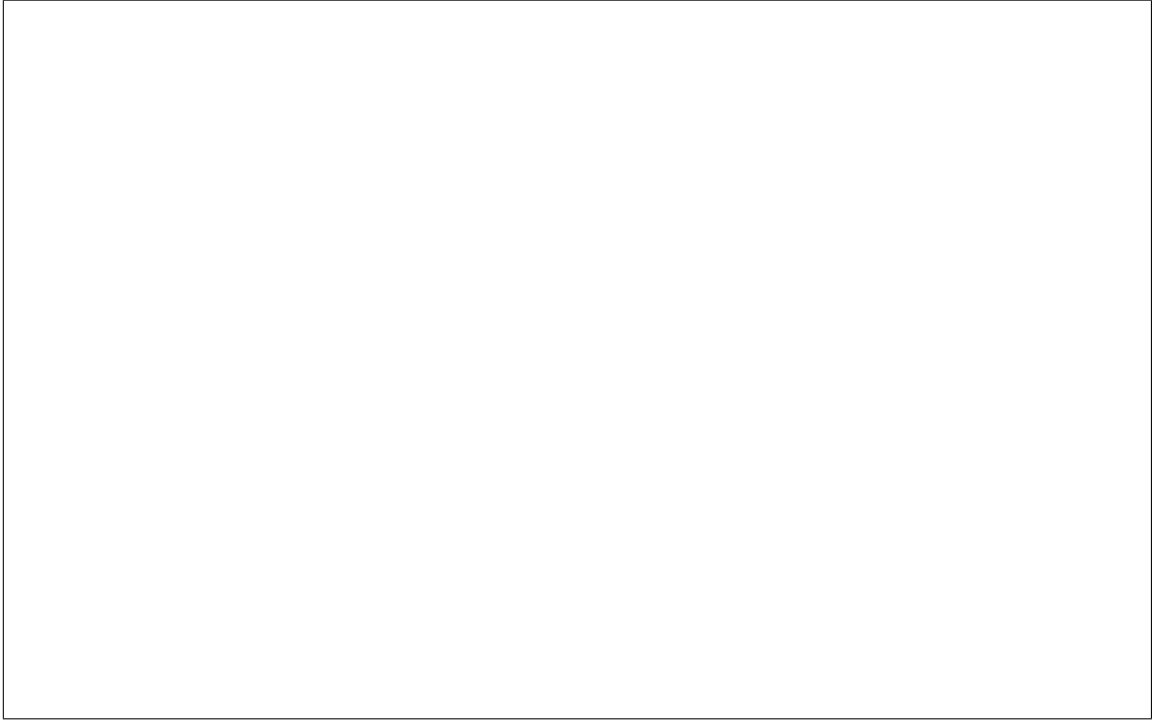




(continues on next page)

(continued from previous page)



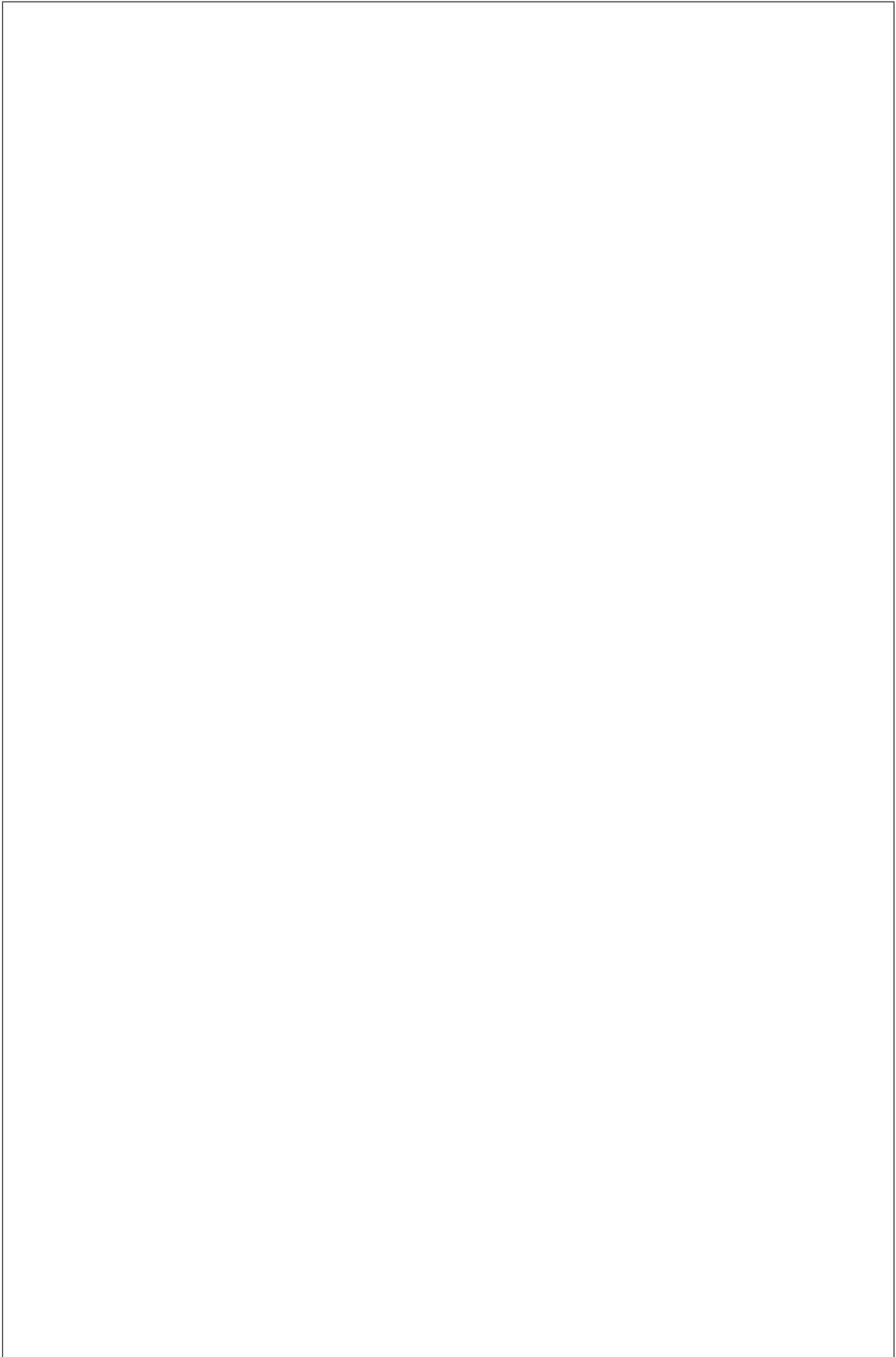


(continues on next page)

(continued from previous page)

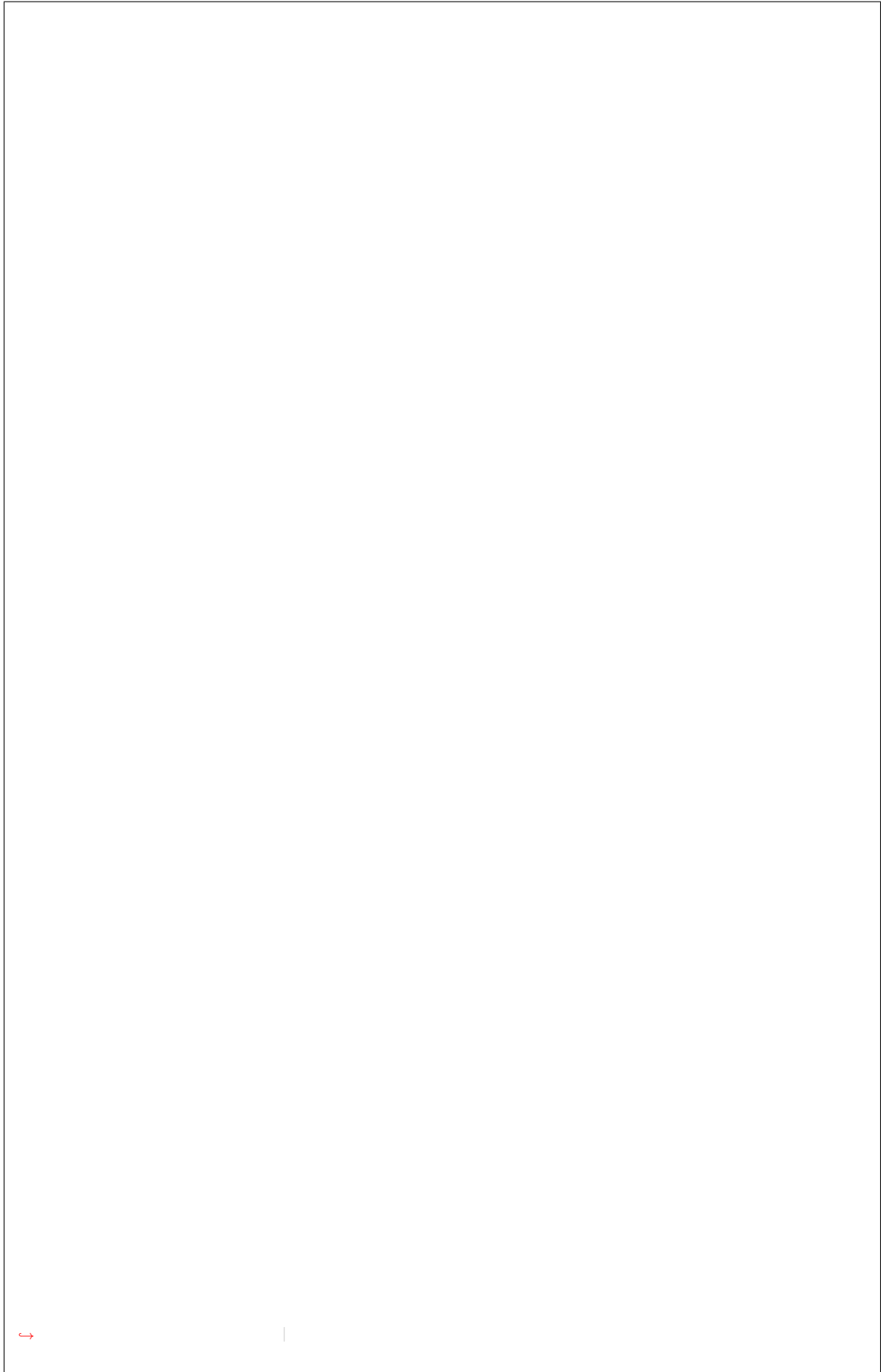
--





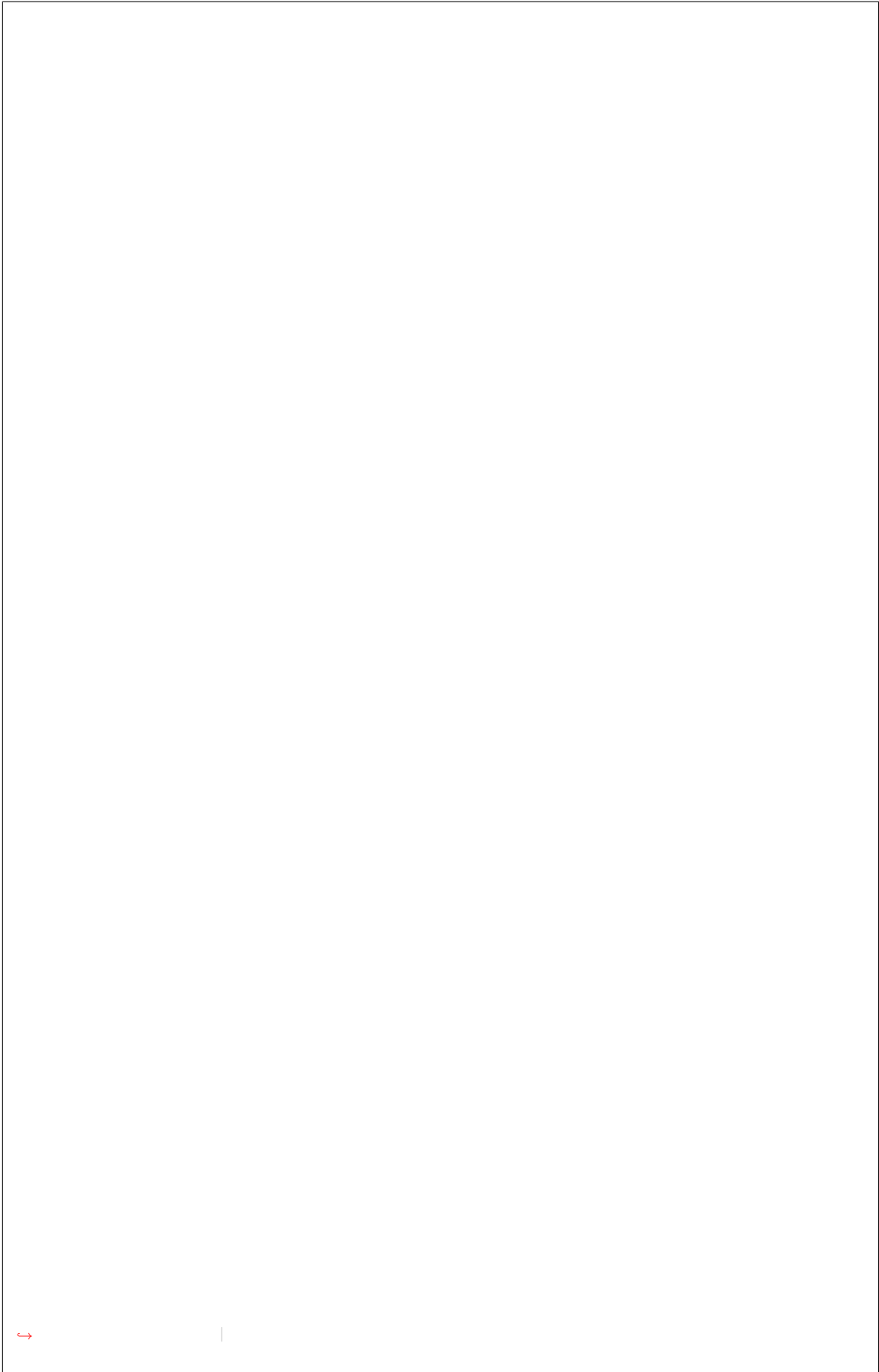
(continues on next page)

(continued from previous page)



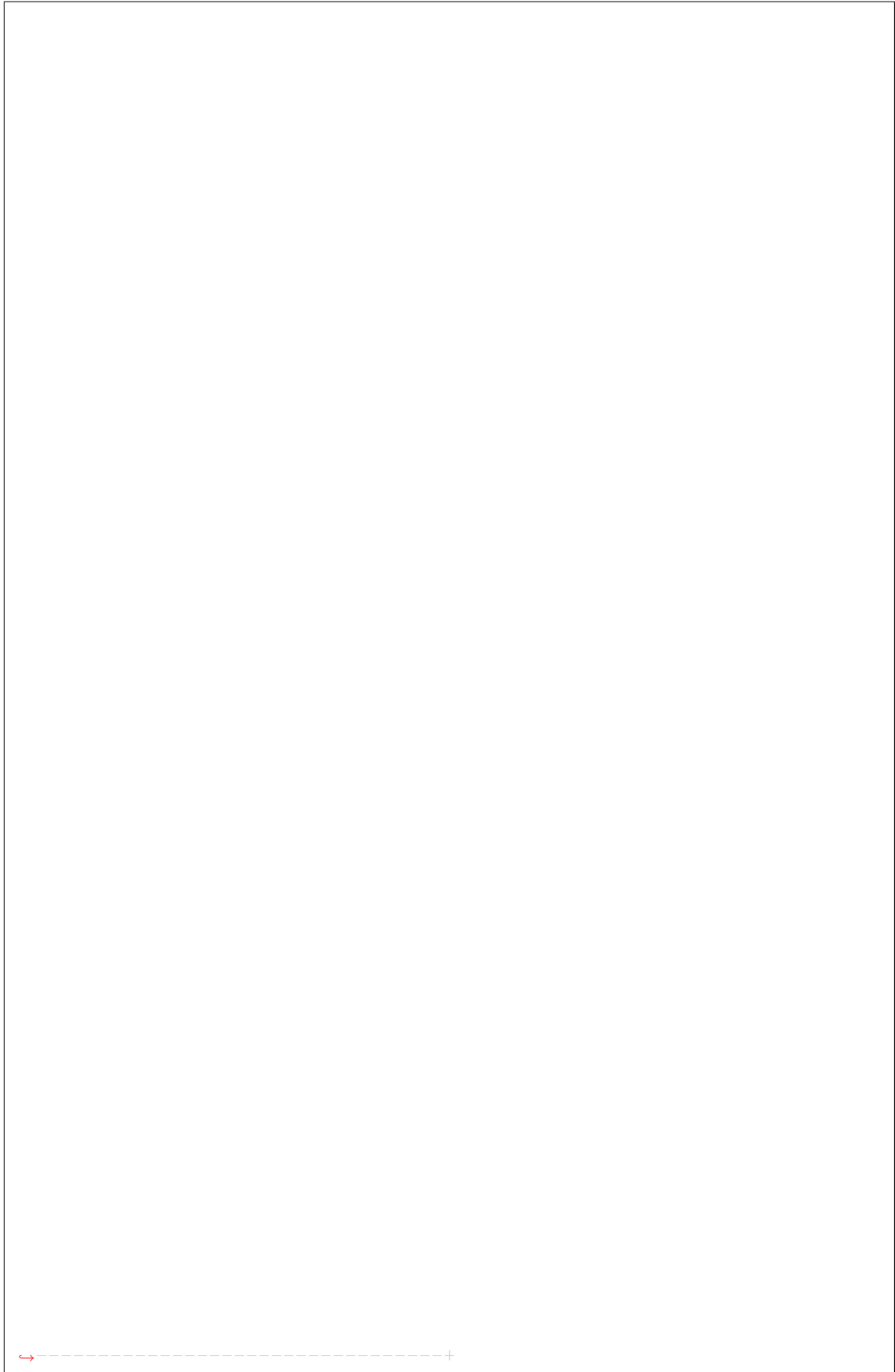
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)





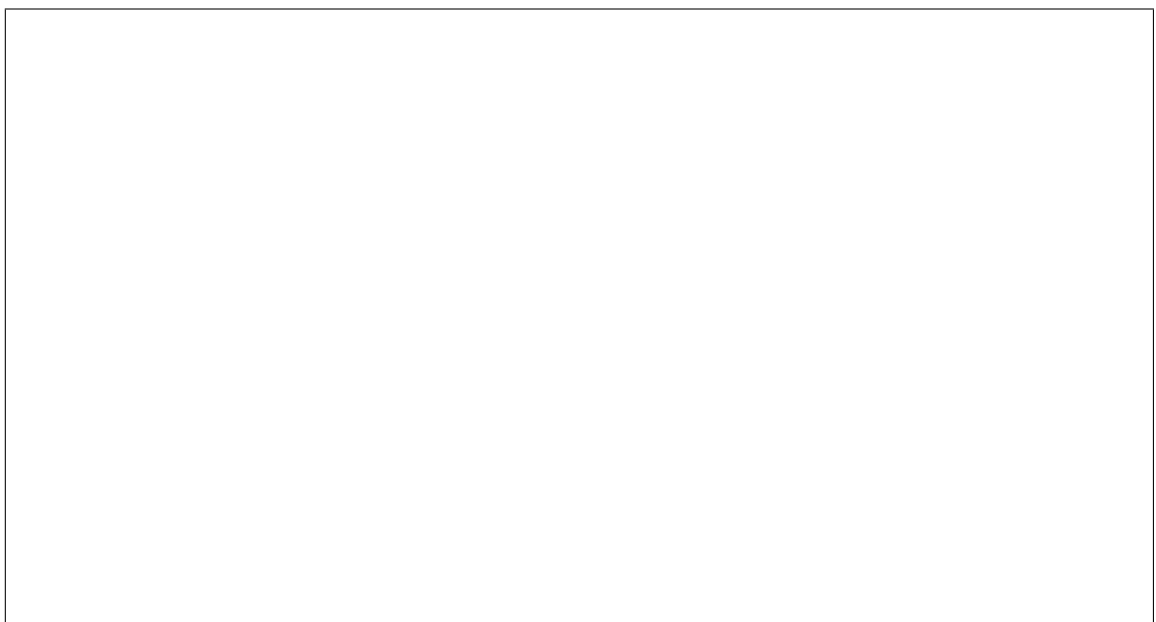
## **Node serial console**





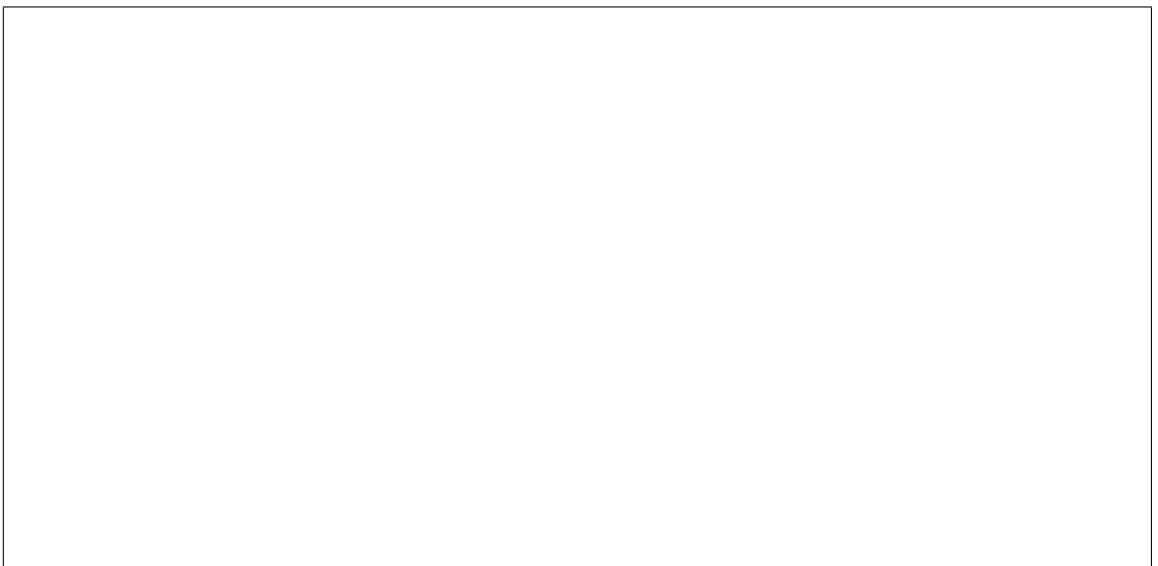
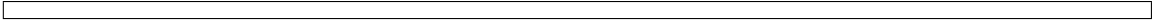






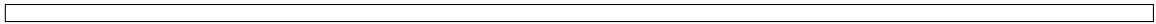
(continues on next page)

(continued from previous page)



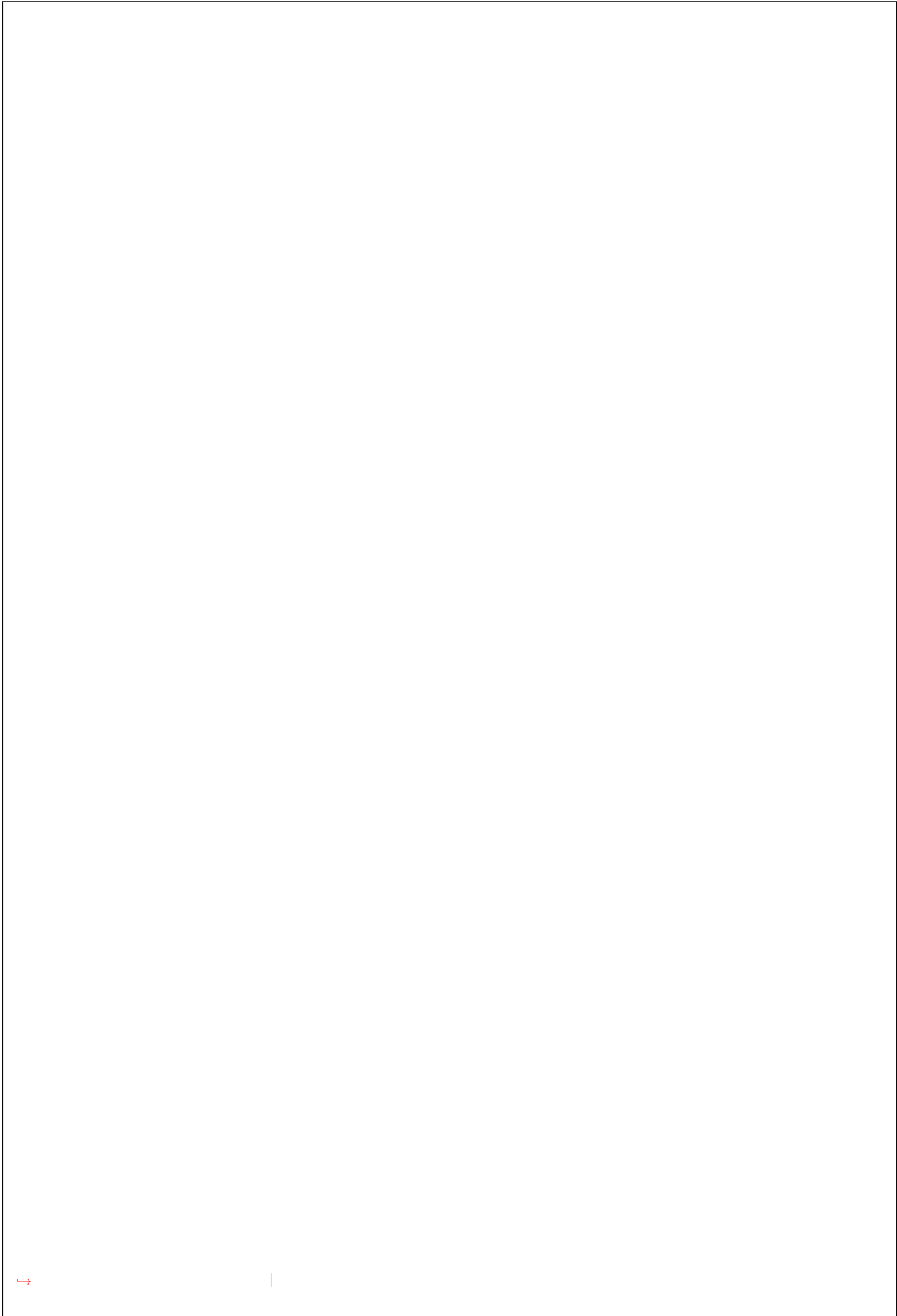
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



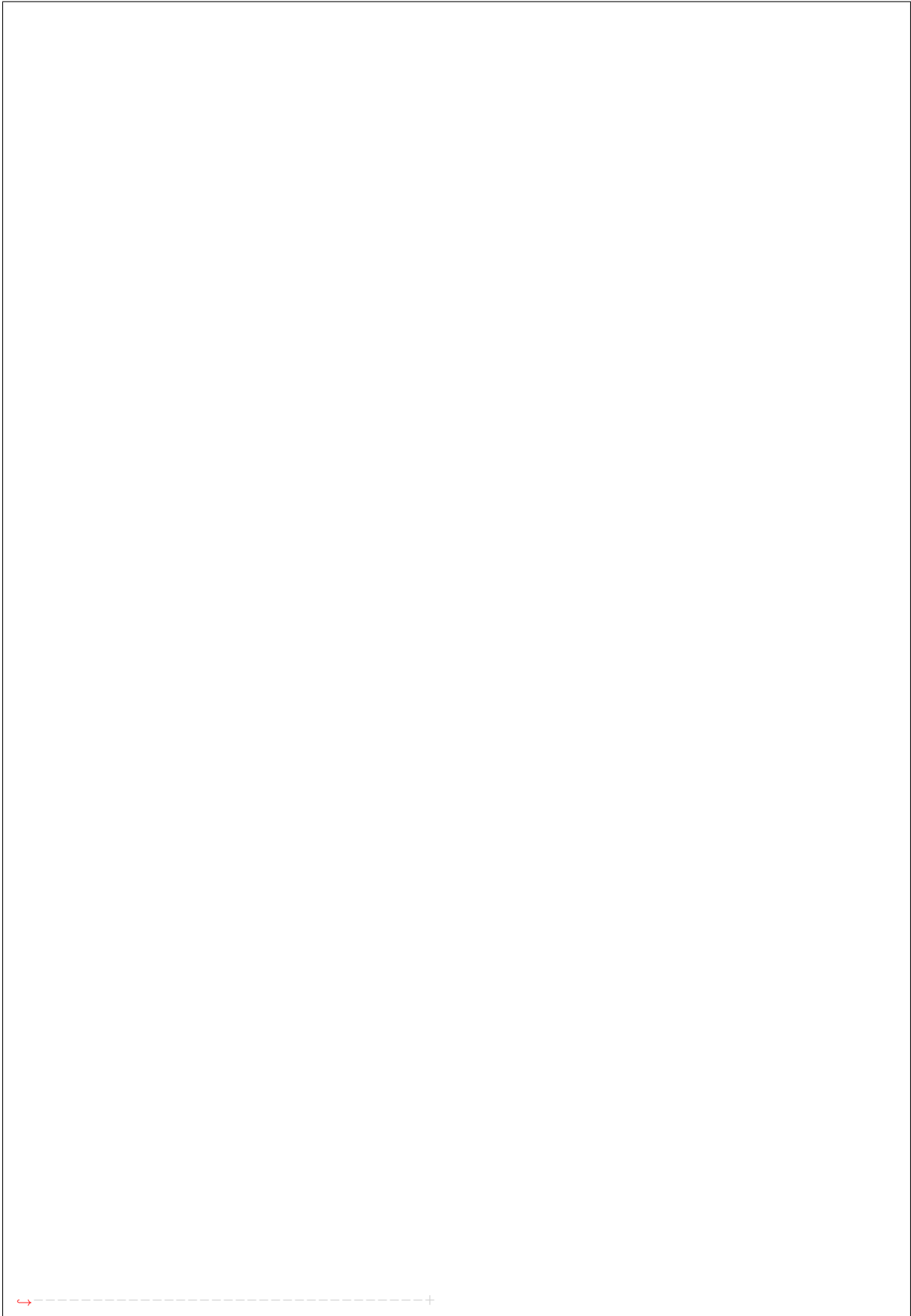
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



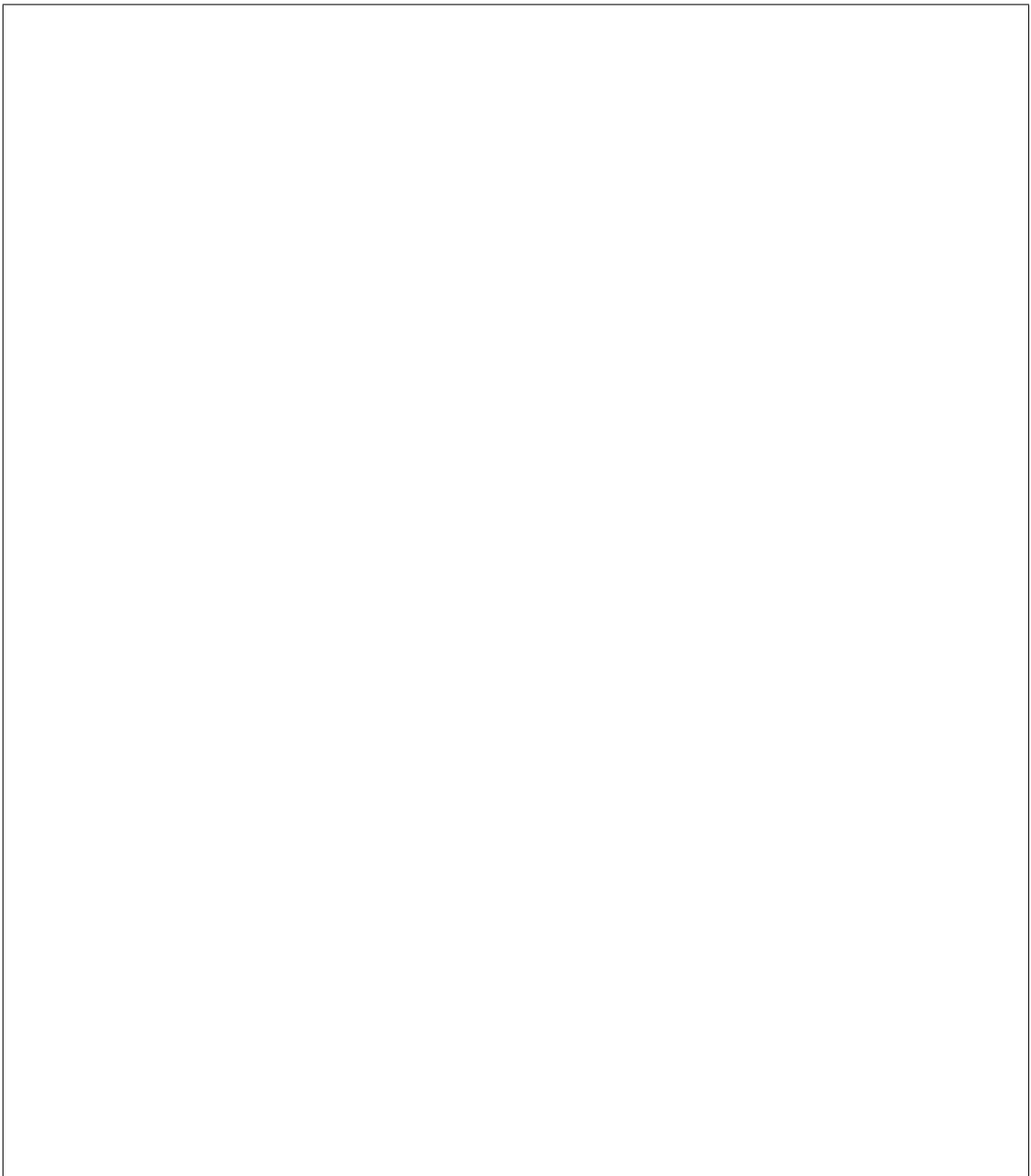




to achieve that, you need to follow the documentation for [Serial Console](#) from the Compute service.

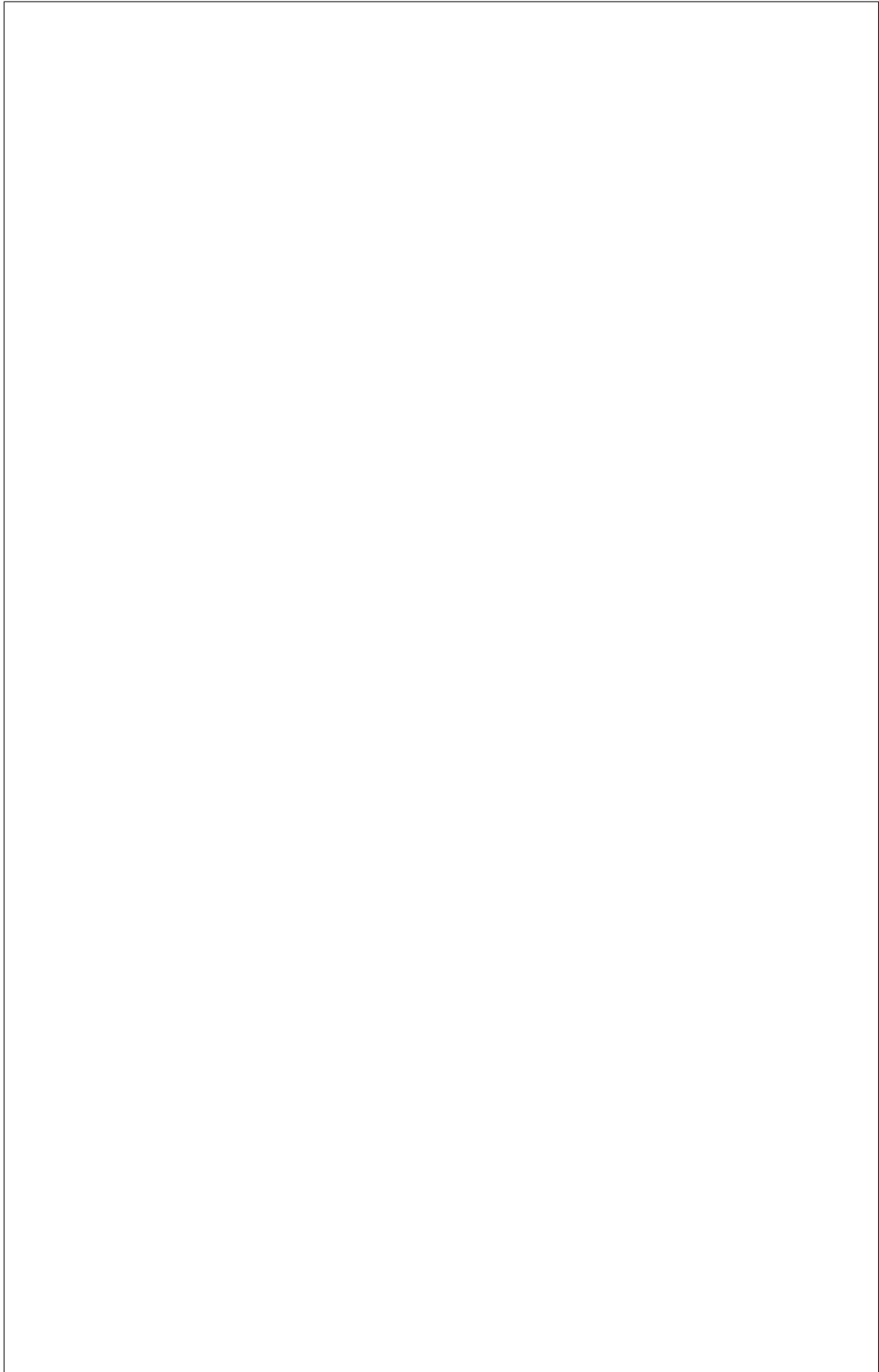
## **Configuring HA**





(continues on next page)

(continued from previous page)



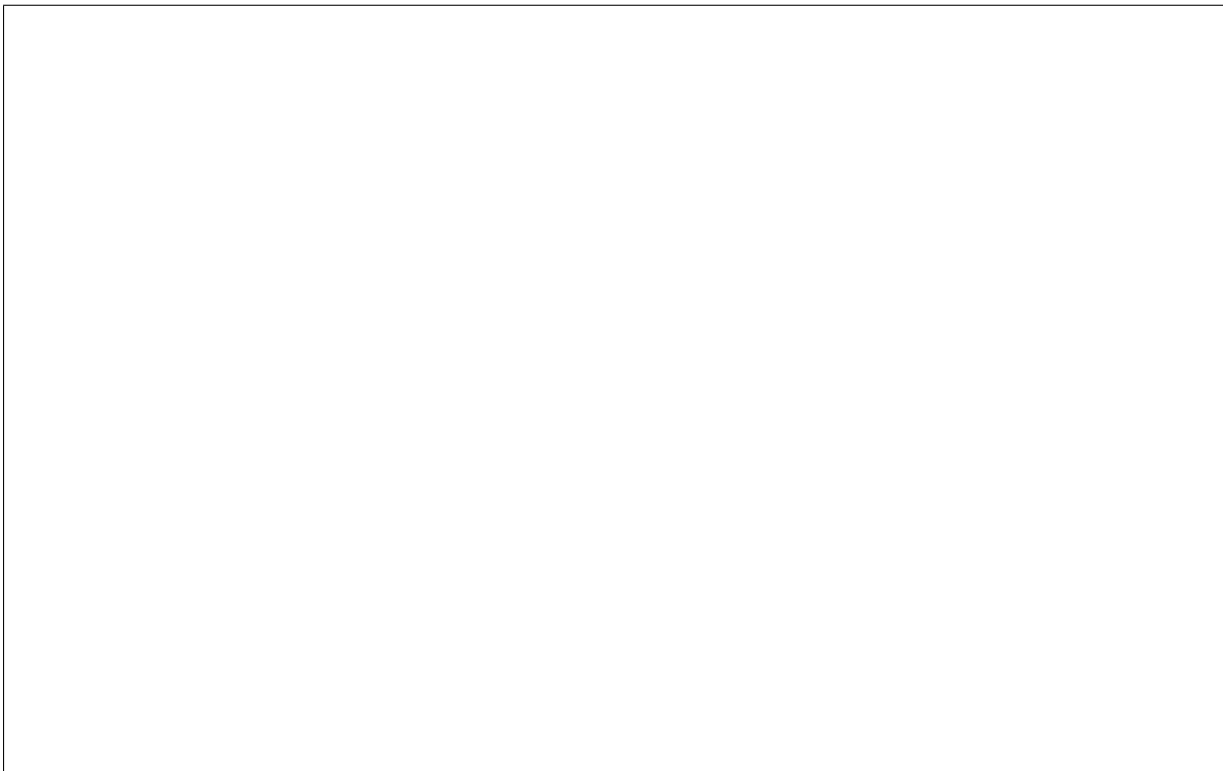
(continues on next page)

(continued from previous page)

--

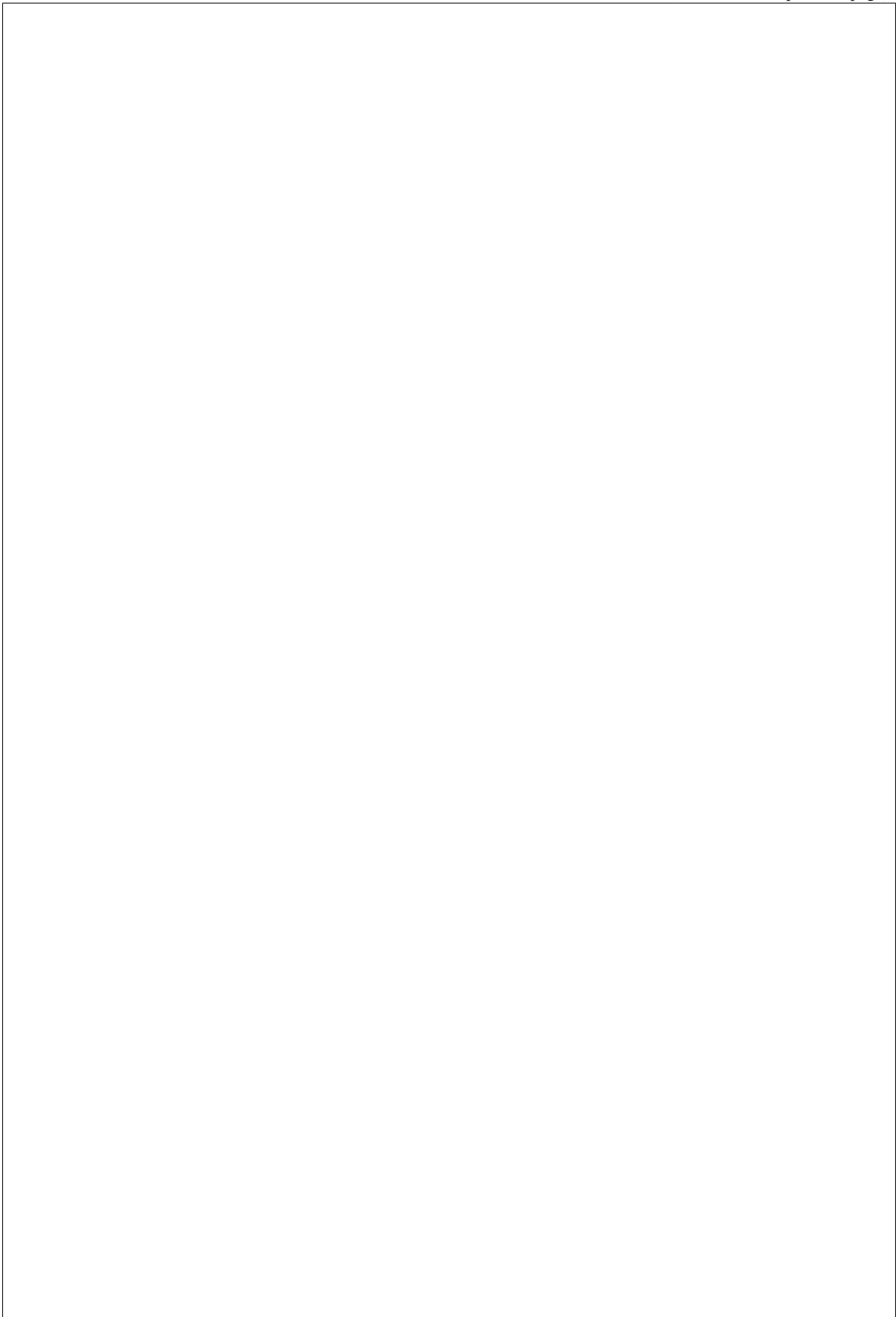


billing or usage system, a monitoring data store, or other OpenStack services. This page describes how to enable notifications and the different kinds of notifications that ironiC may emit. The external consumer will see notifications emitted by ironiC as JSON objects structured in the following manner:



(continues on next page)

(continued from previous page)



(continues on next page)



(continued from previous page)

--

## **Configuration**

fications are emitted. For example, if the option is set to warning, all notifications with priority level warning, error, or critical are emitted, but not notifications with priority level debug or info. For information about the semantics of each log level, see the OpenStack logging standards<sup>1</sup>. If this option is unset, no notifications will be emitted. The priority level of each available notification is documented below.

---

<sup>1</sup> [https://wiki.openstack.org/wiki/LoggingStandards#Log\\_level\\_definitions](https://wiki.openstack.org/wiki/LoggingStandards#Log_level_definitions)





mation, see the documentation of your chosen message bus, such as the RabbitMQ documentation<sup>2</sup>.

## Versioning

---

<sup>2</sup> <https://www.rabbitmq.com/documentation.html>



## **Available notifications**

**ironic-api notifications**

**Resources CRUD notifications**

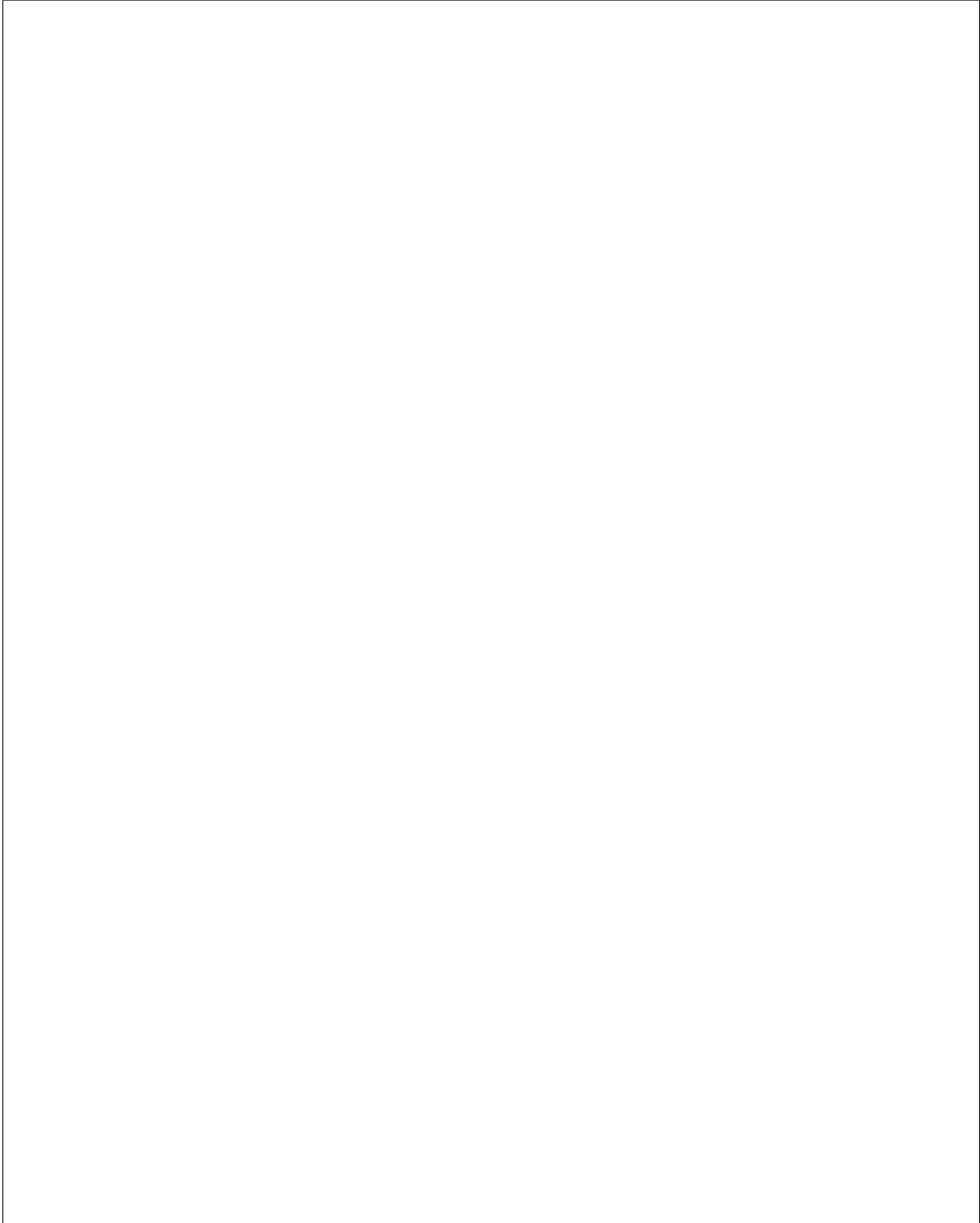


that is emitted at ERROR level.

---

<sup>3</sup> [https://en.wikipedia.org/wiki/Create,\\_read,\\_update\\_and\\_delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete)





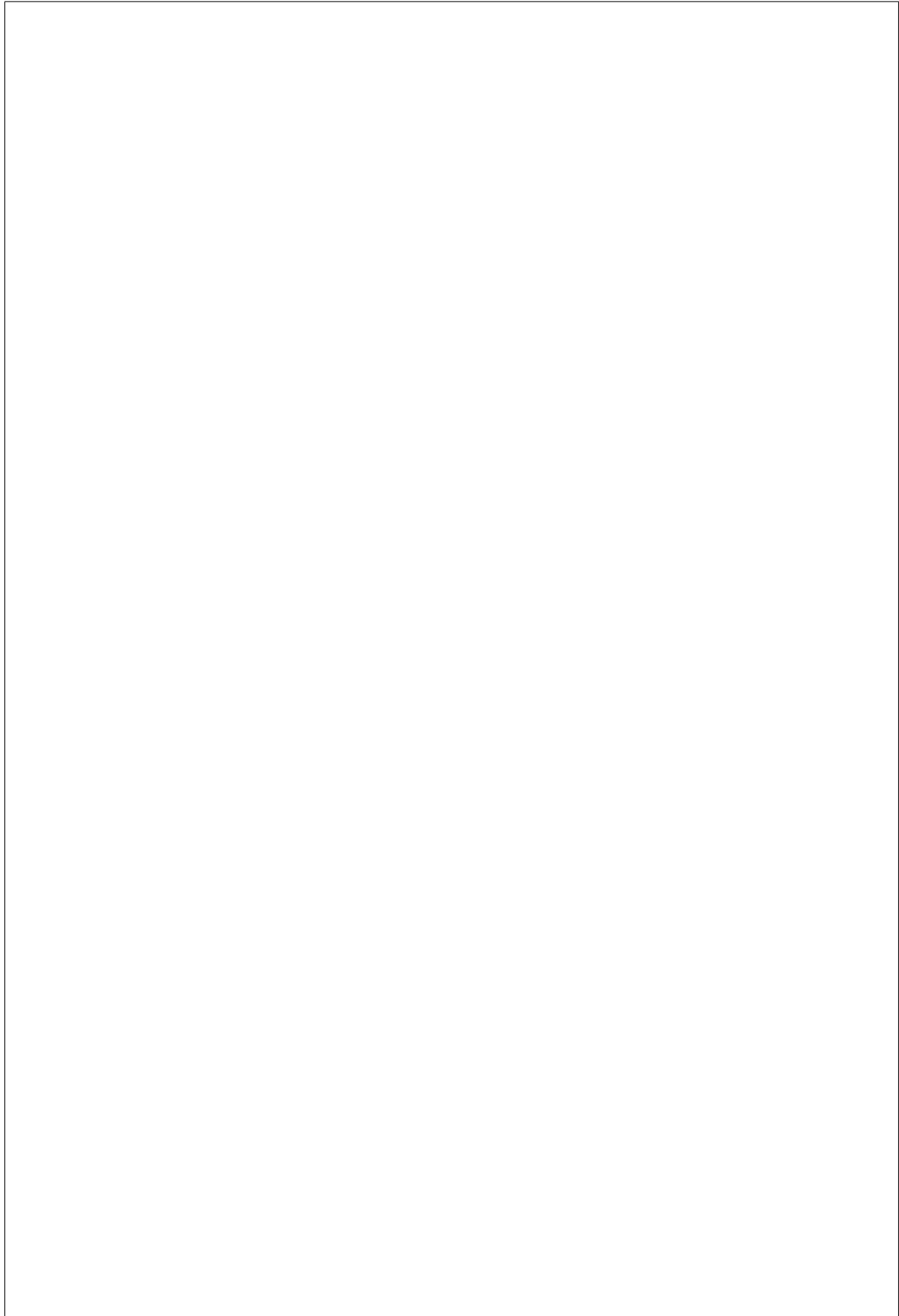
(continues on next page)

(continued from previous page)



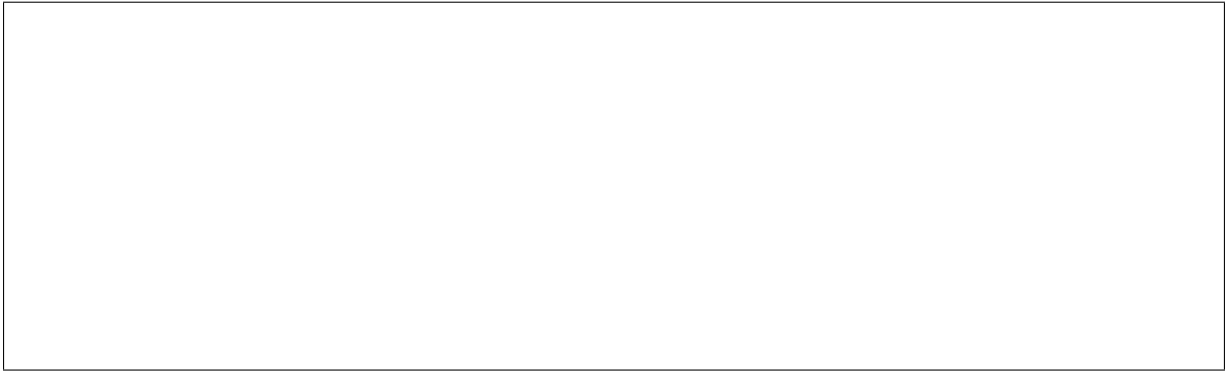
(continues on next page)

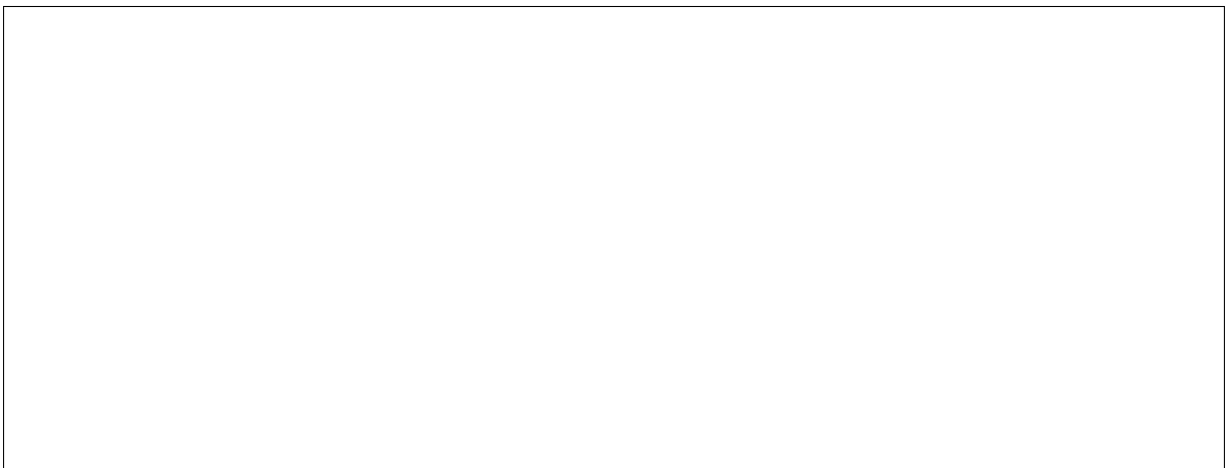
(continued from previous page)



(continues on next page)

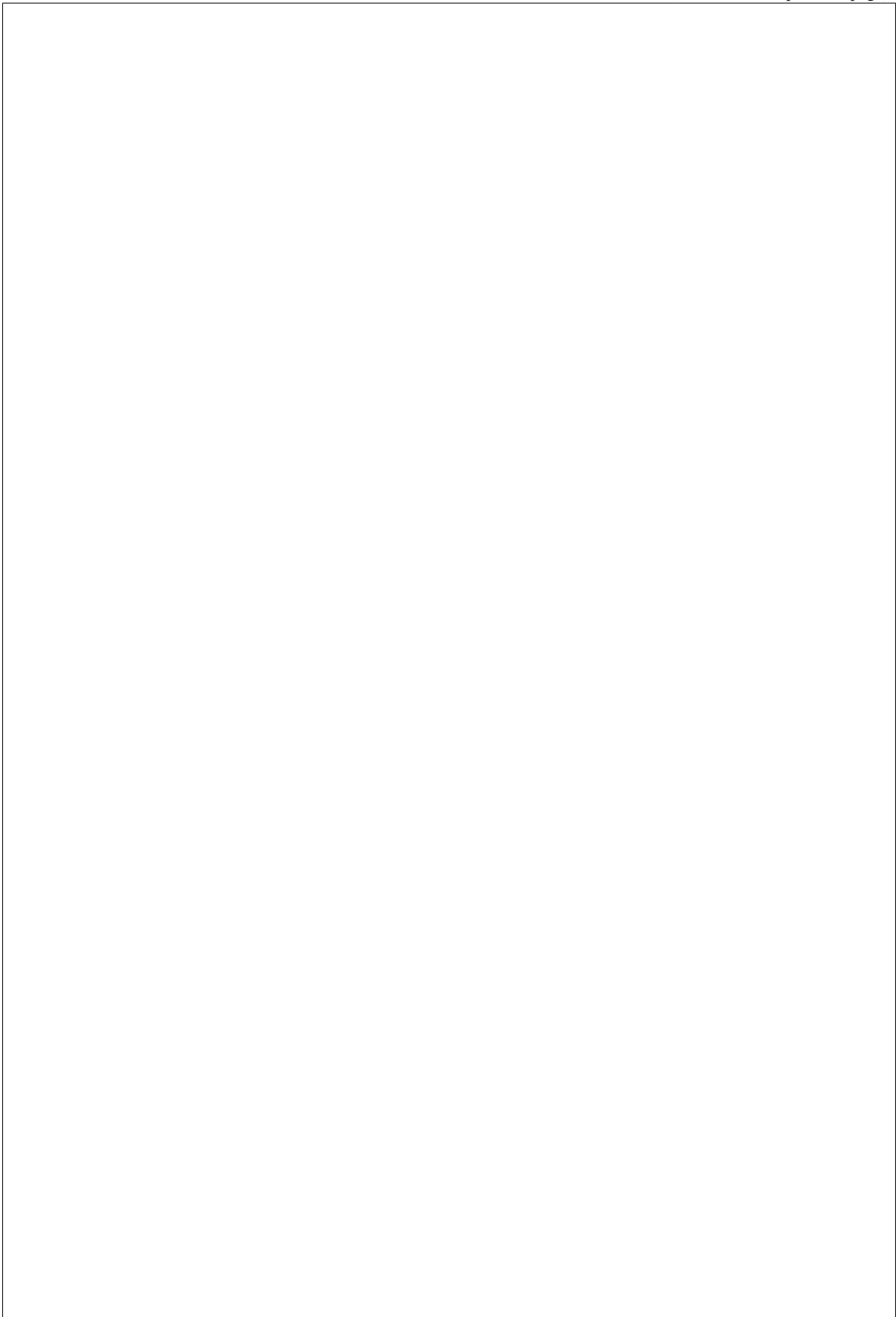
(continued from previous page)





(continues on next page)

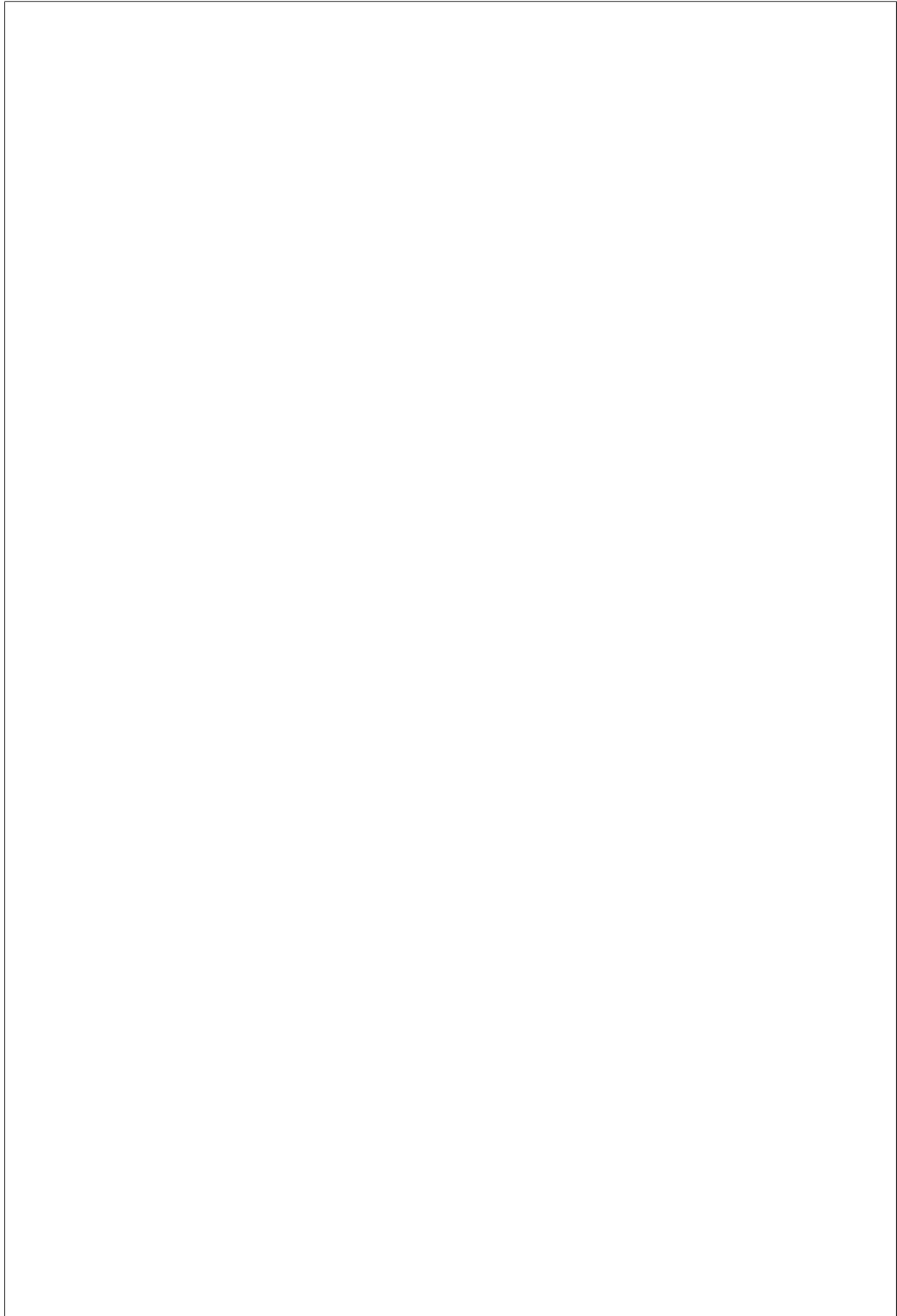
(continued from previous page)



(continues on next page)



(continued from previous page)



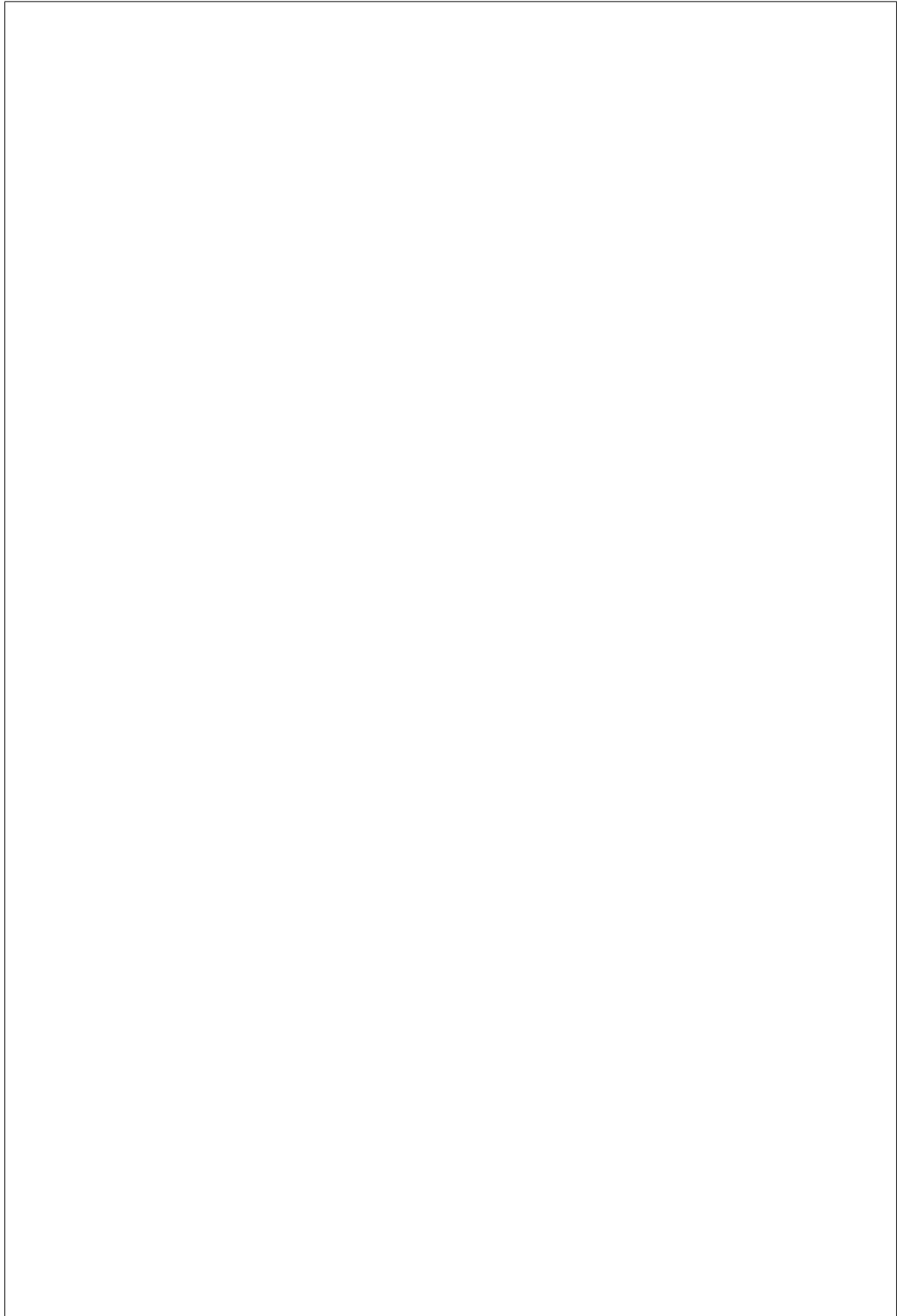
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



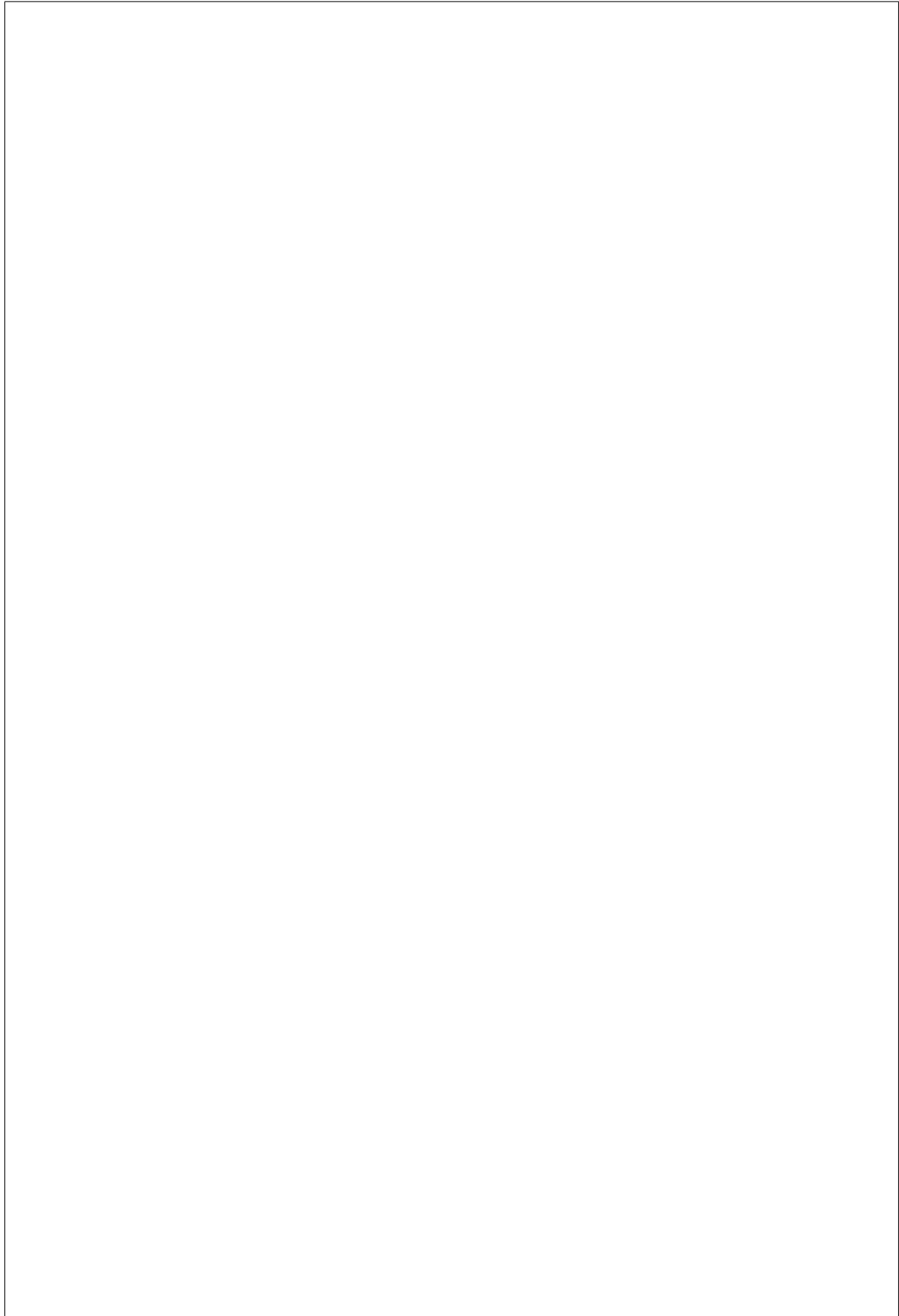
(continues on next page)

(continued from previous page)



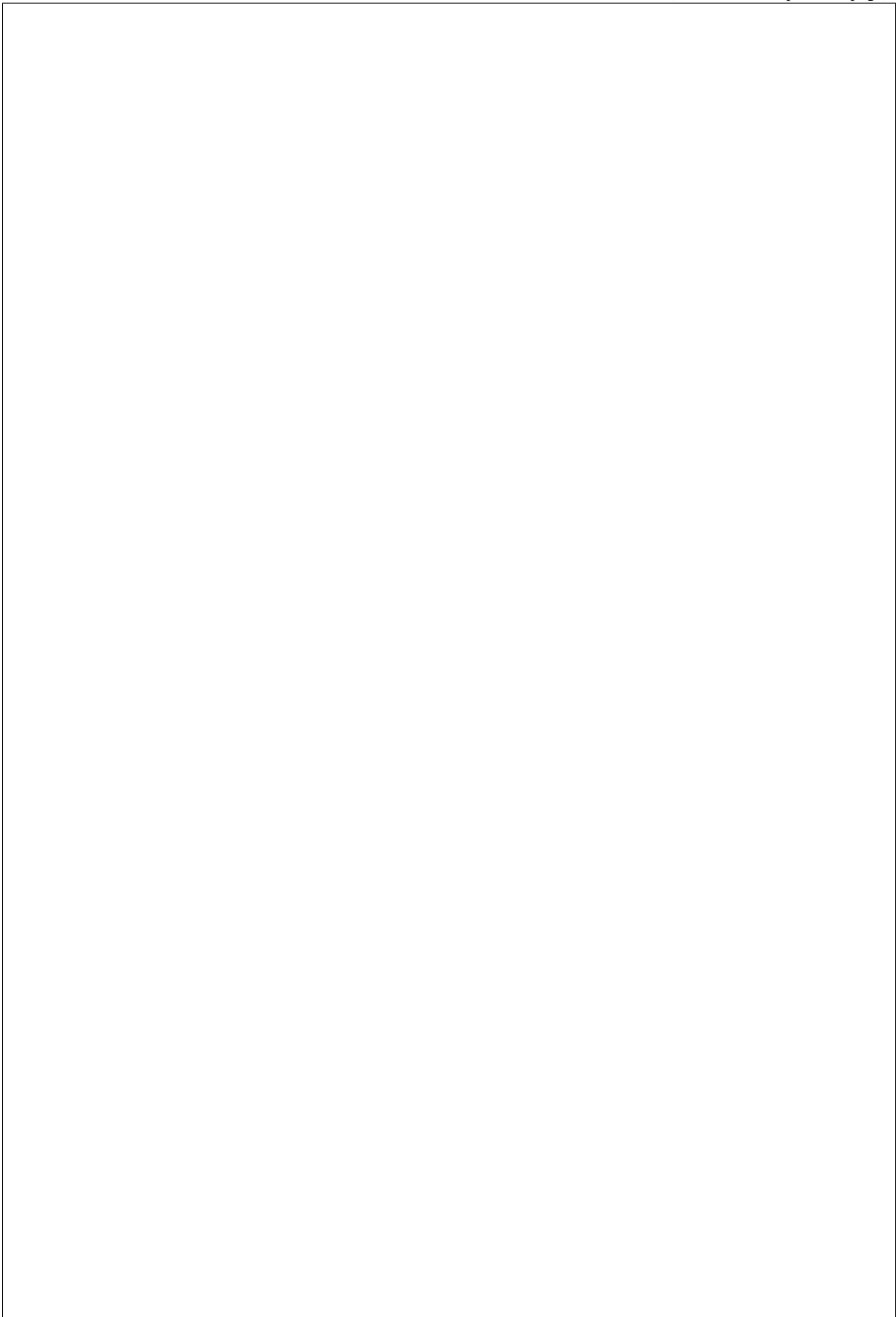
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



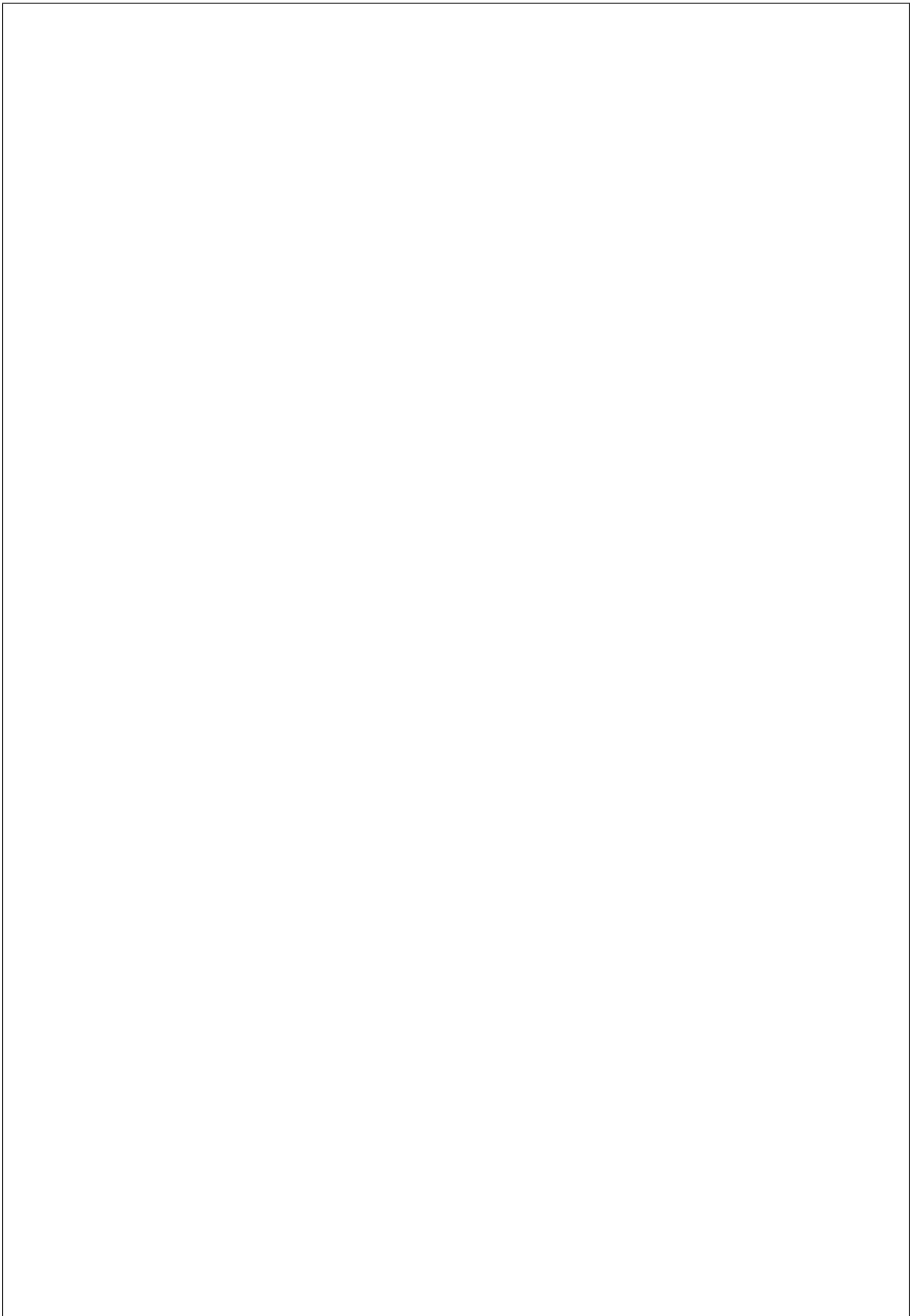
(continues on next page)

(continued from previous page)









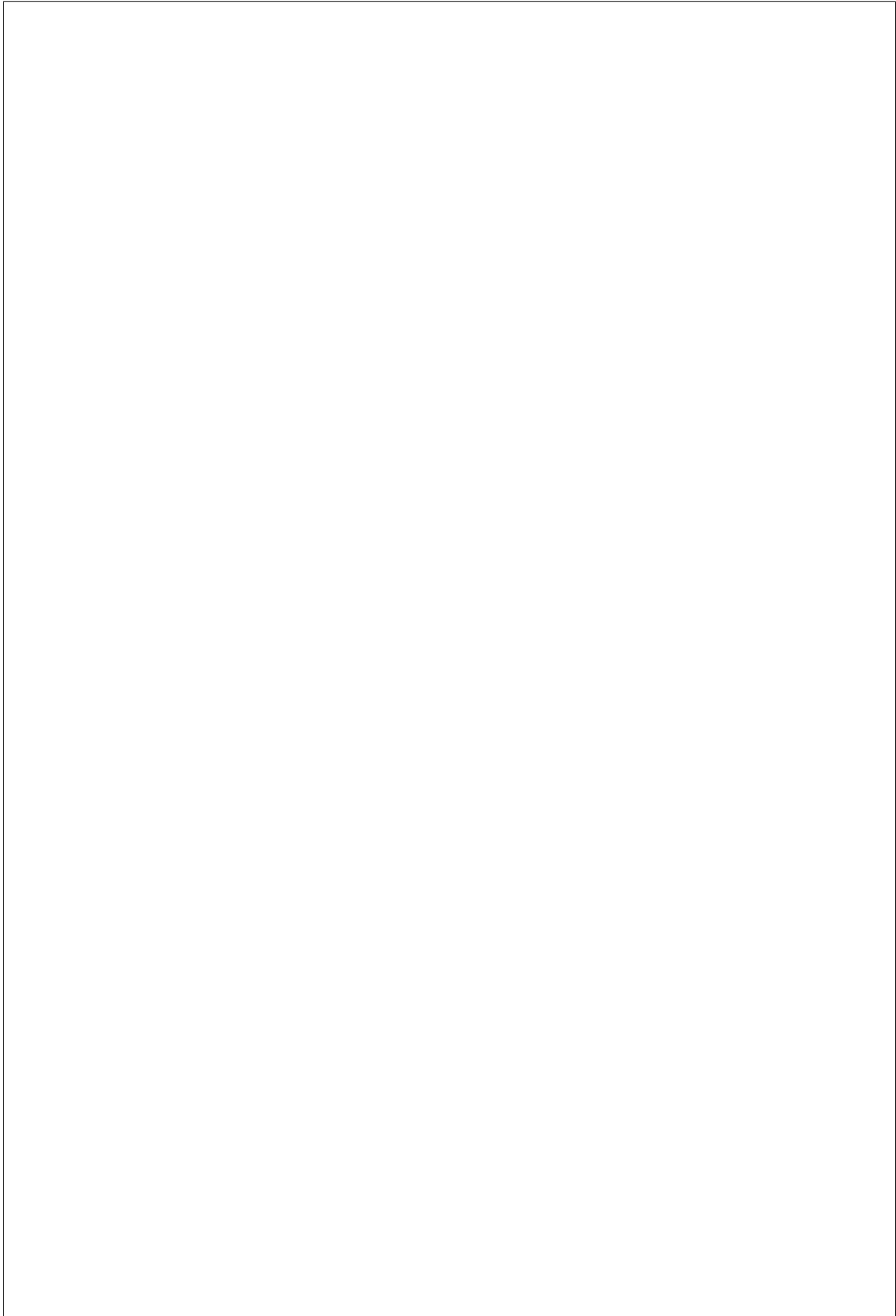
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



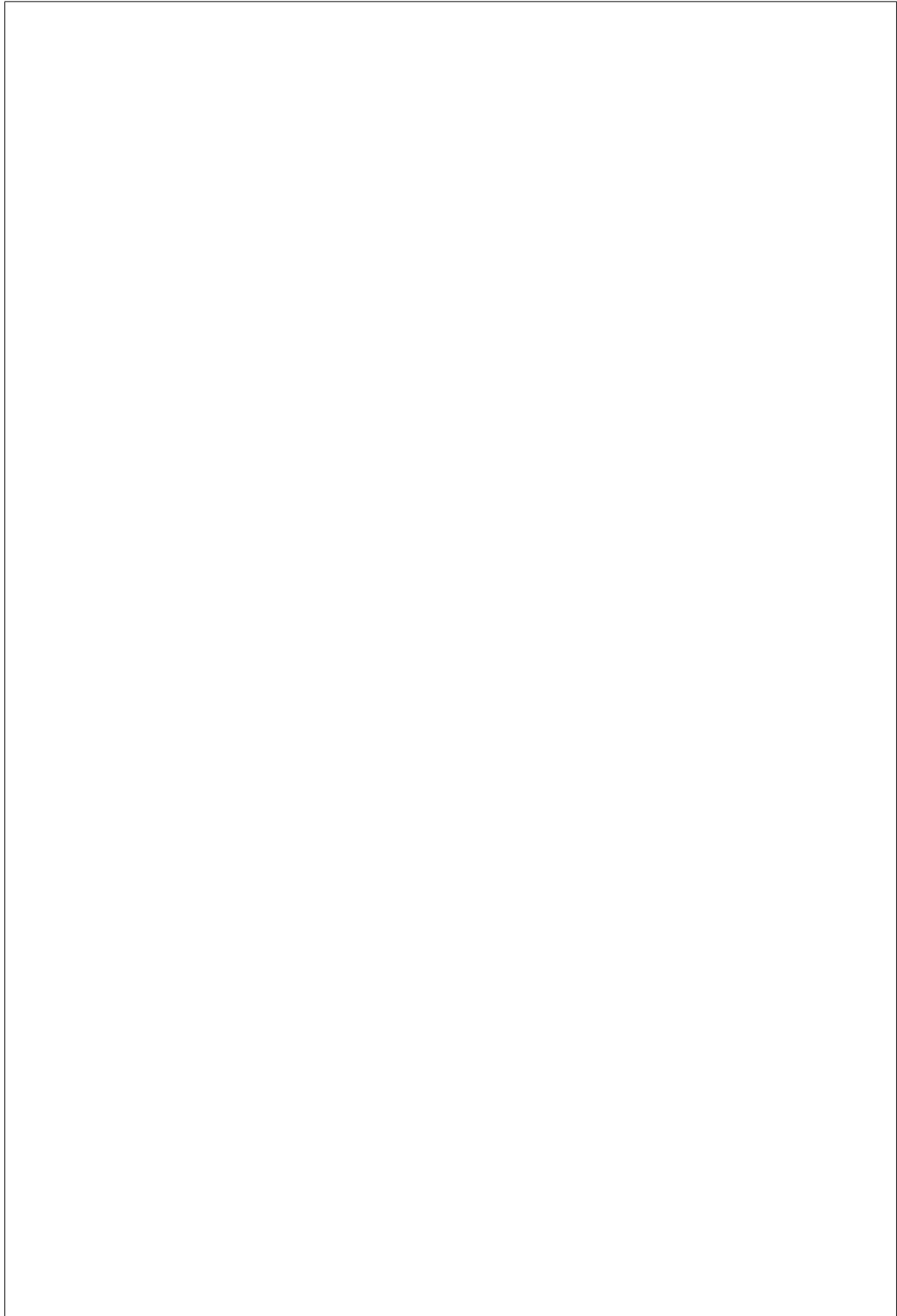
(continues on next page)

(continued from previous page)



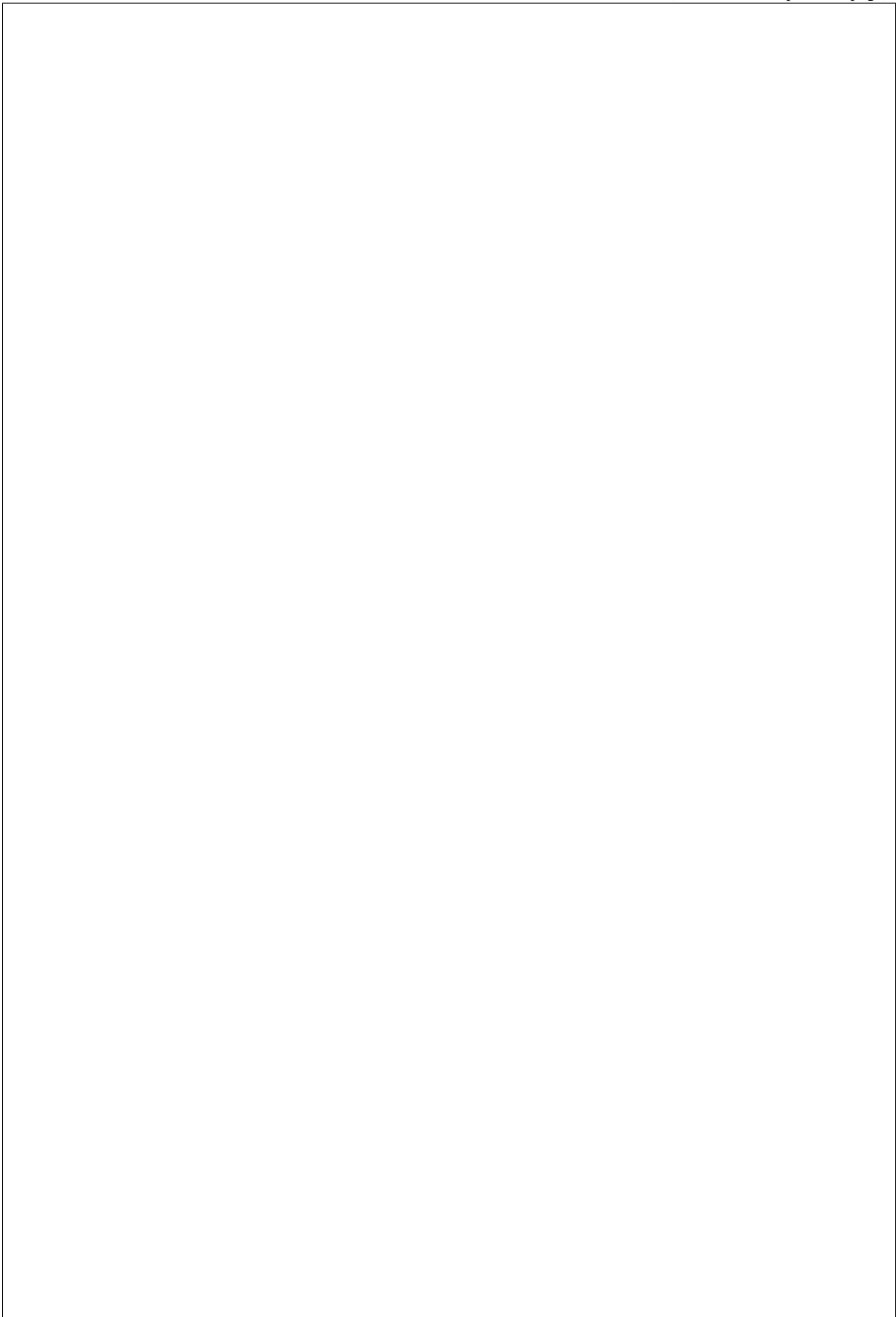
(continues on next page)

(continued from previous page)



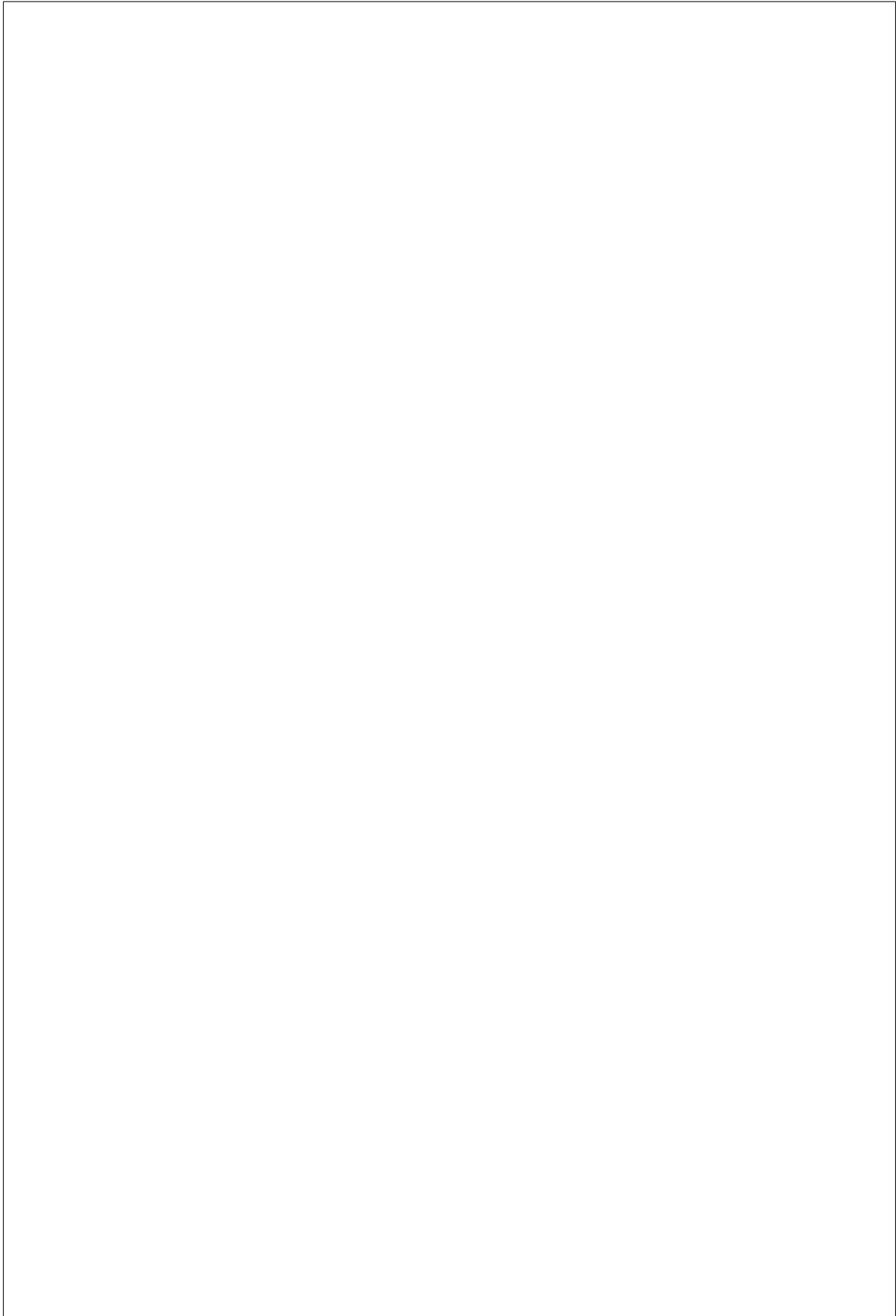
(continues on next page)

(continued from previous page)



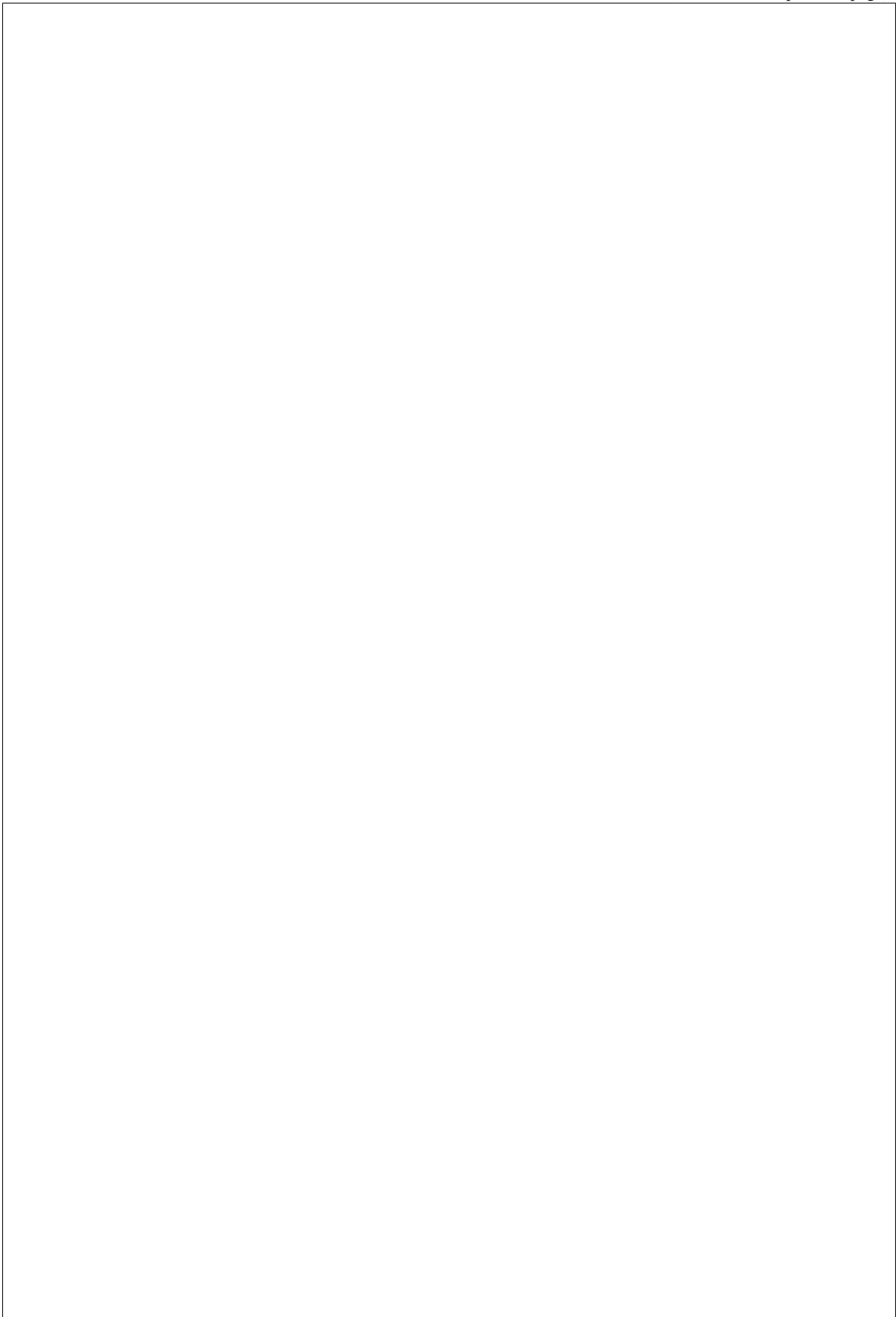
(continues on next page)

(continued from previous page)



(continues on next page)

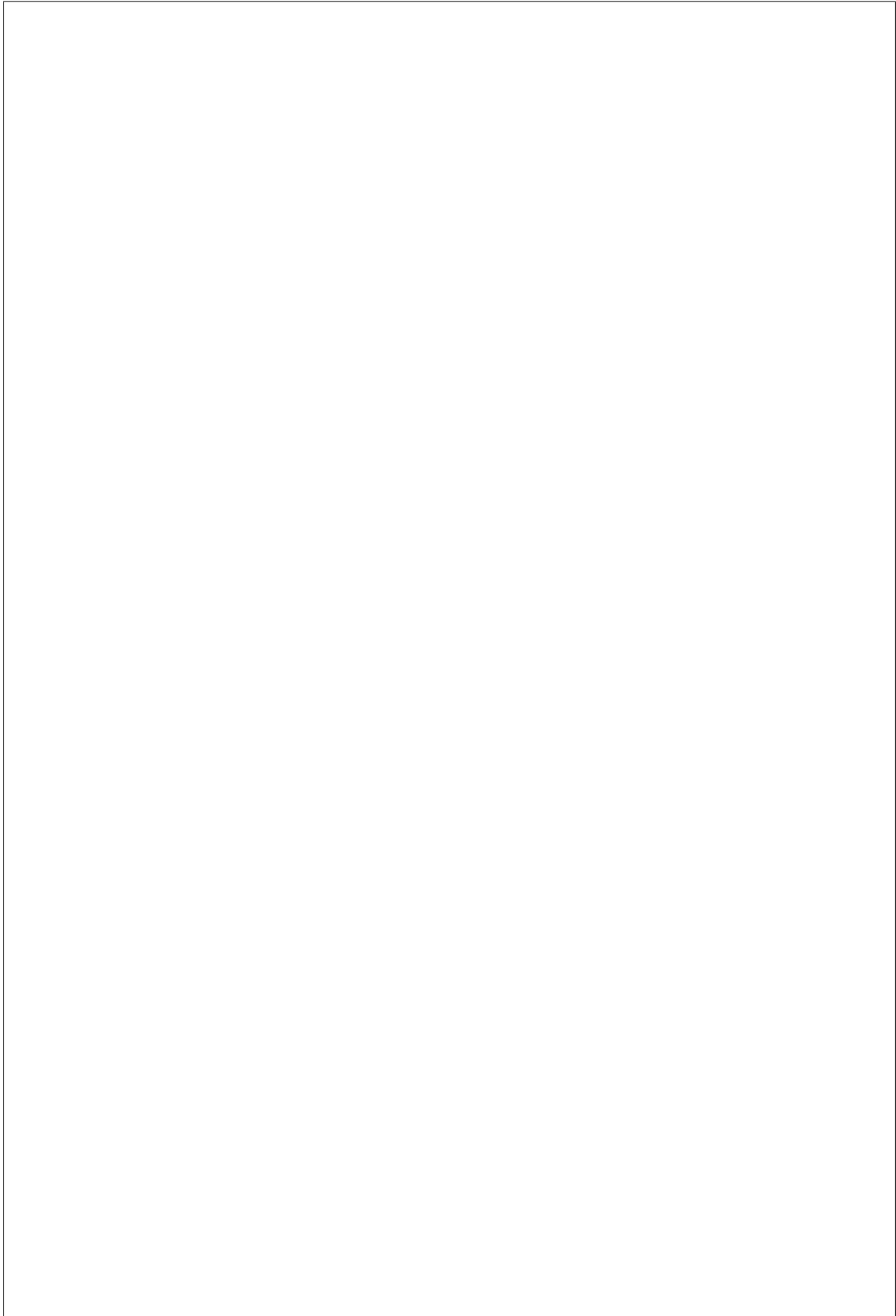
(continued from previous page)



(continues on next page)



(continued from previous page)



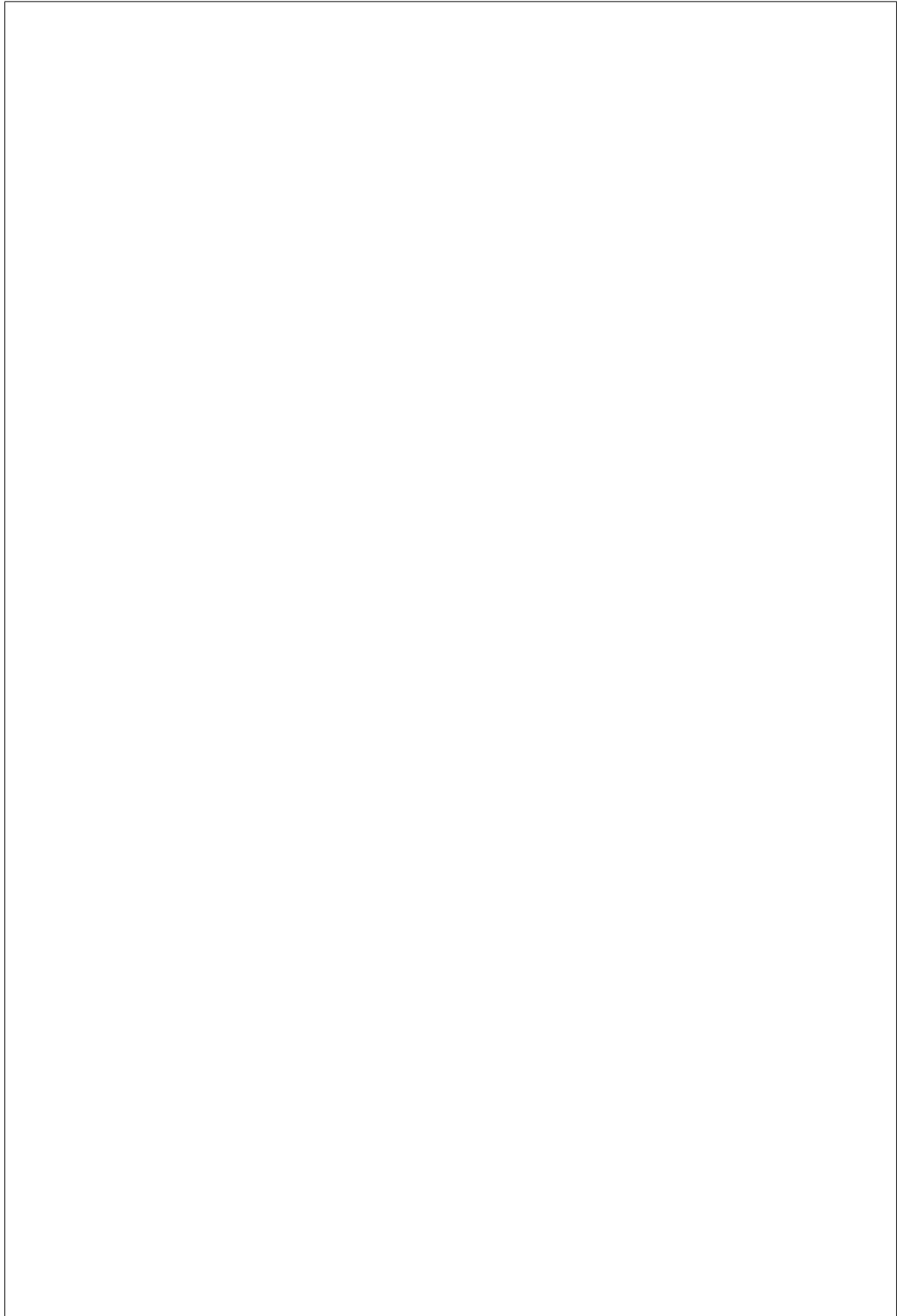
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



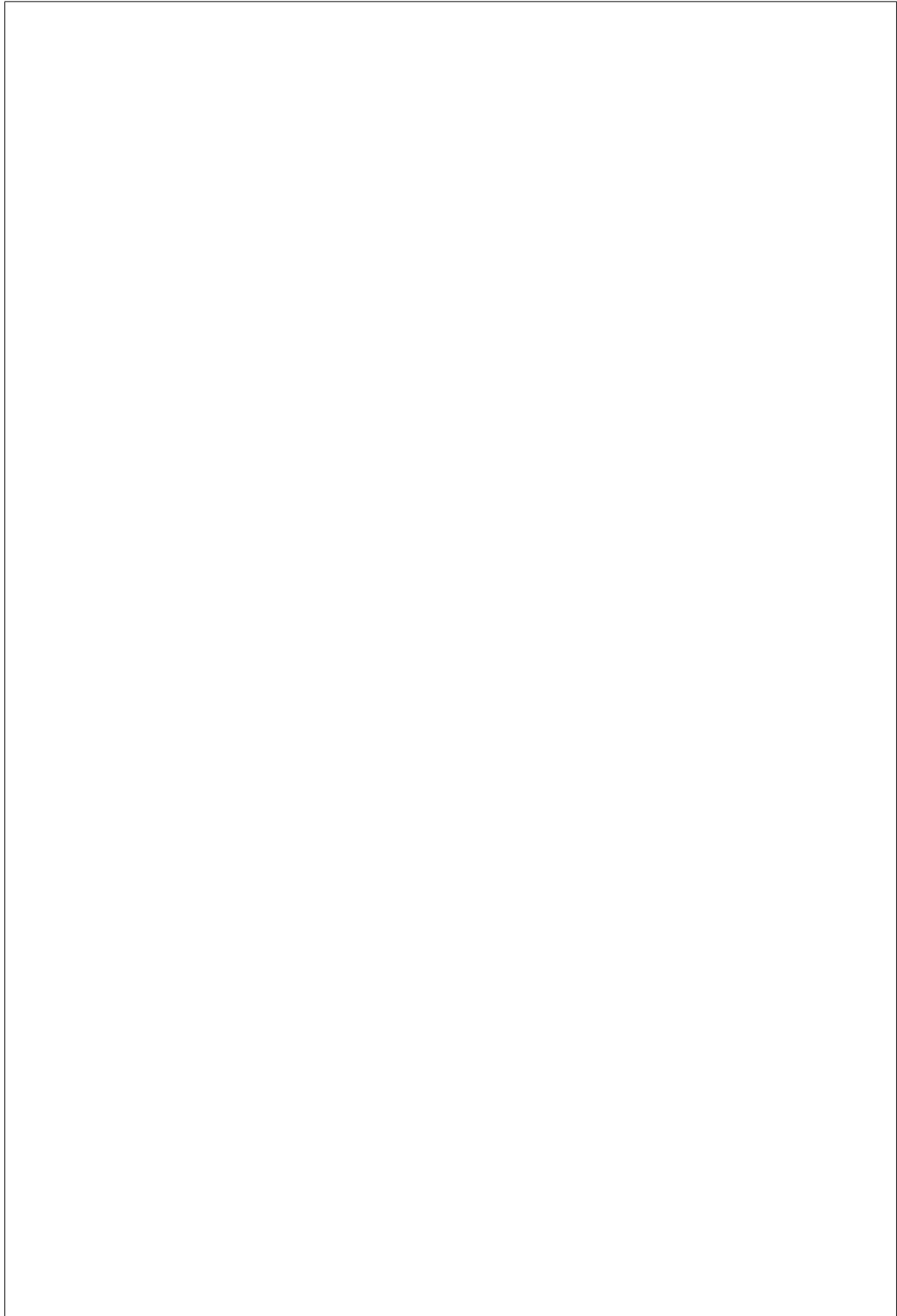
(continues on next page)

(continued from previous page)



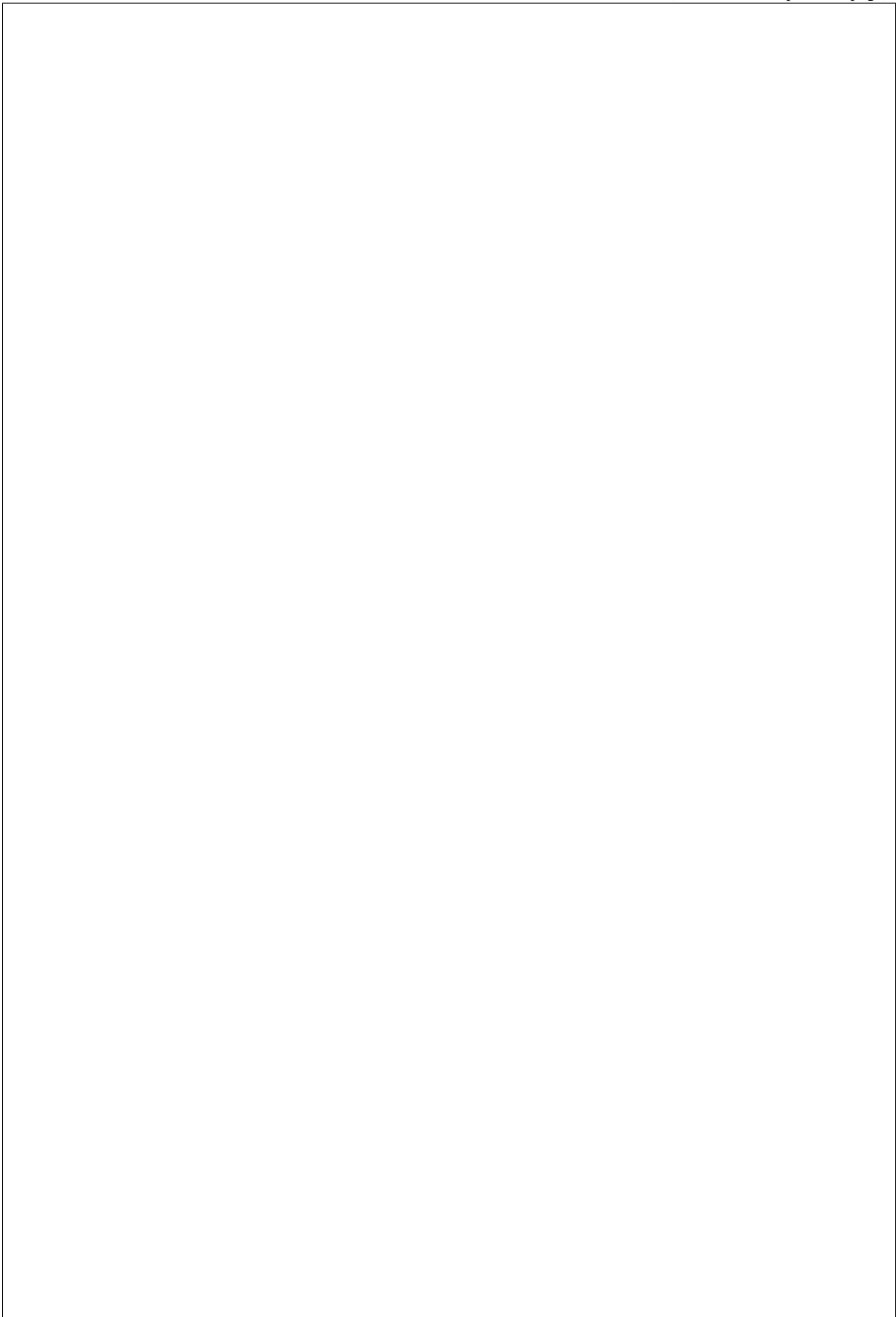
(continues on next page)

(continued from previous page)



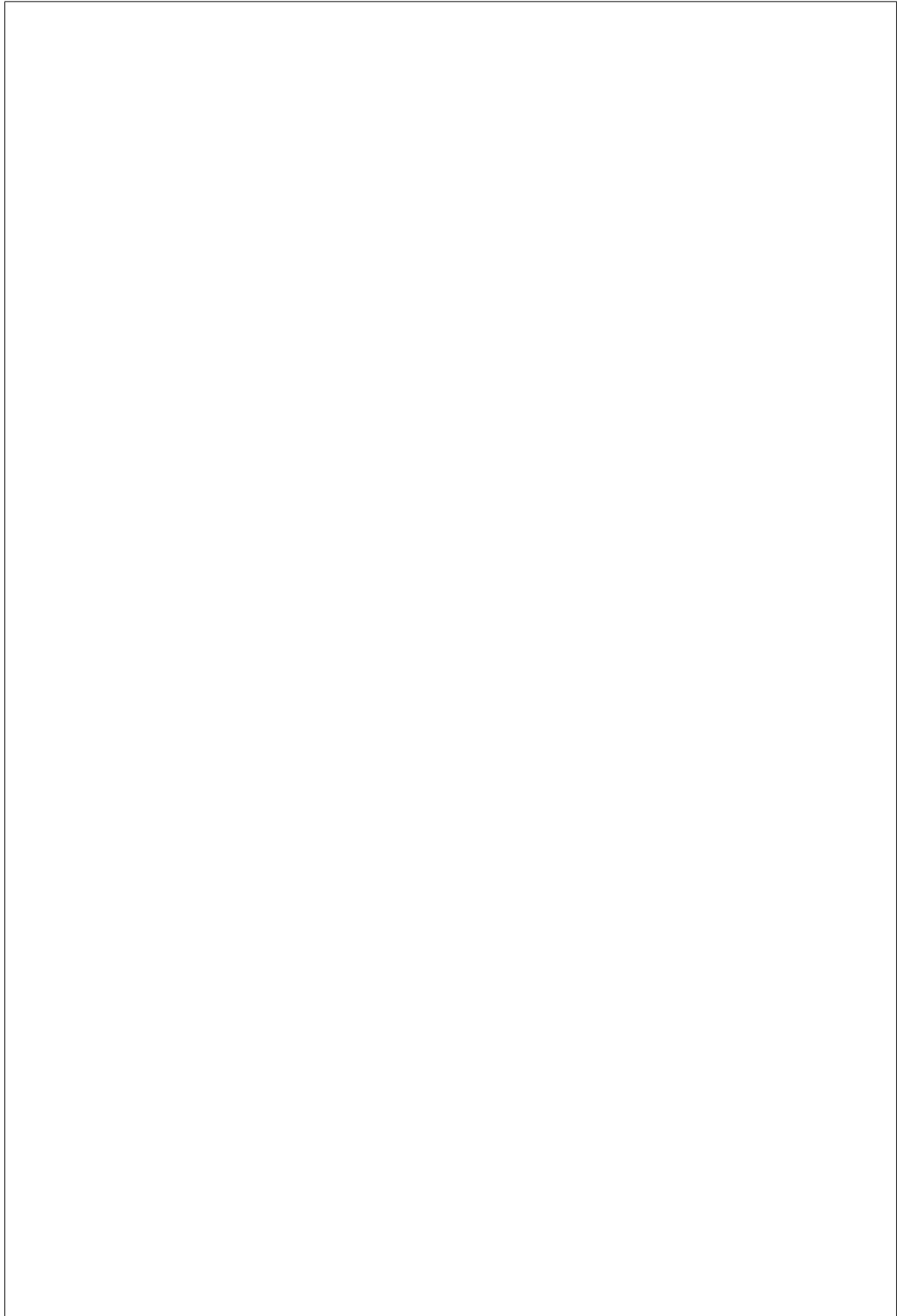
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

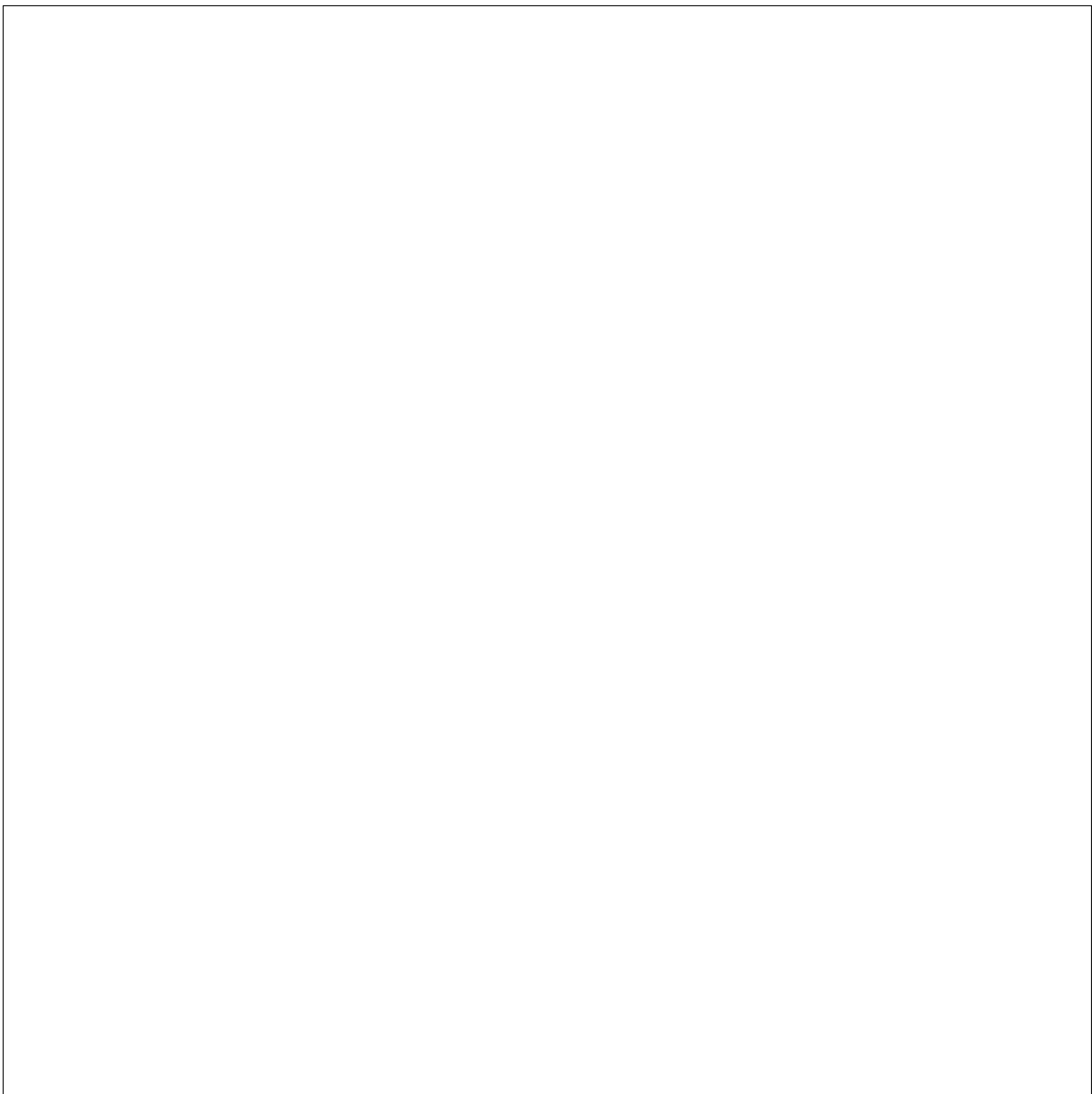


(continues on next page)

(continued from previous page)

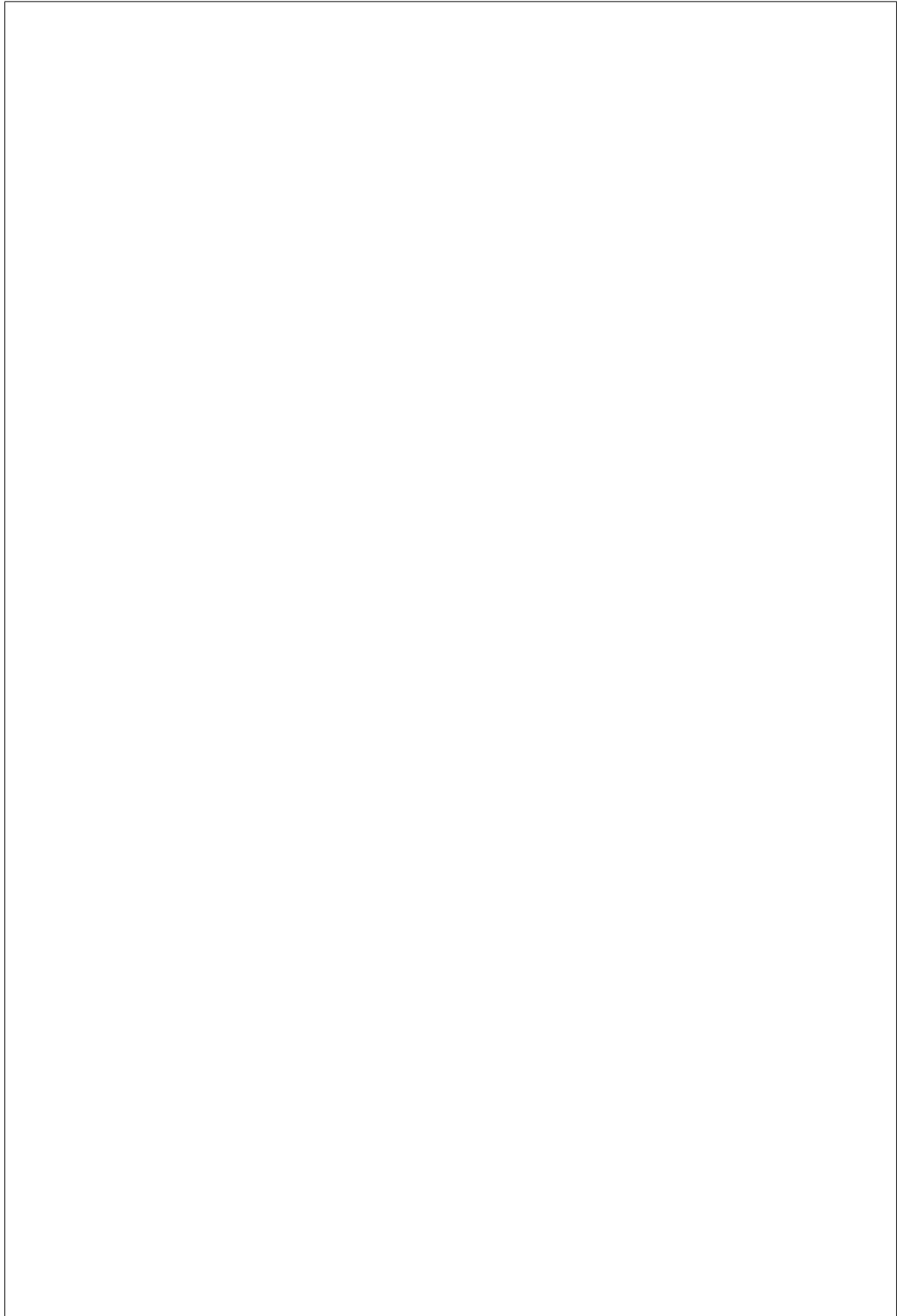






(continues on next page)

(continued from previous page)



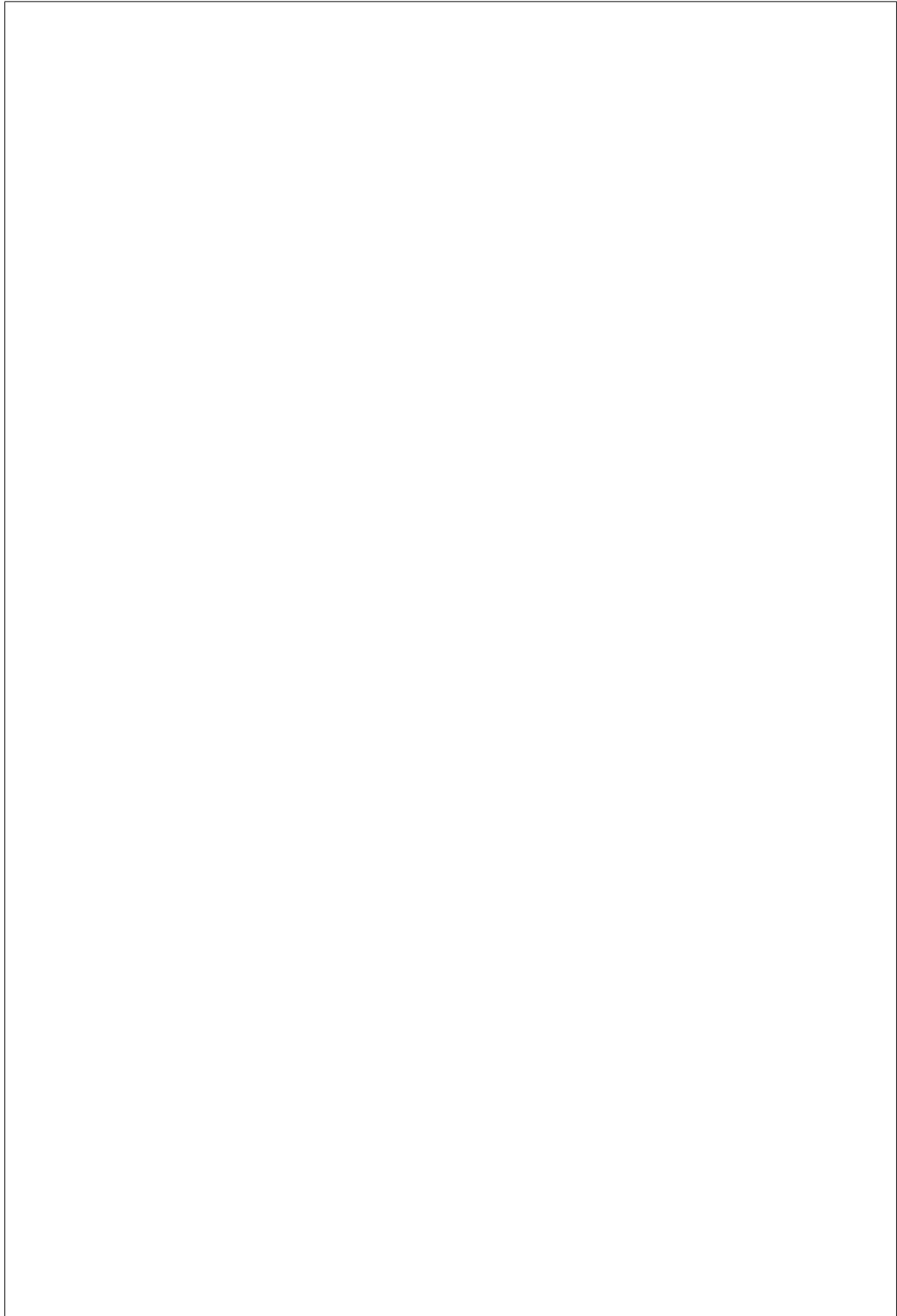
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

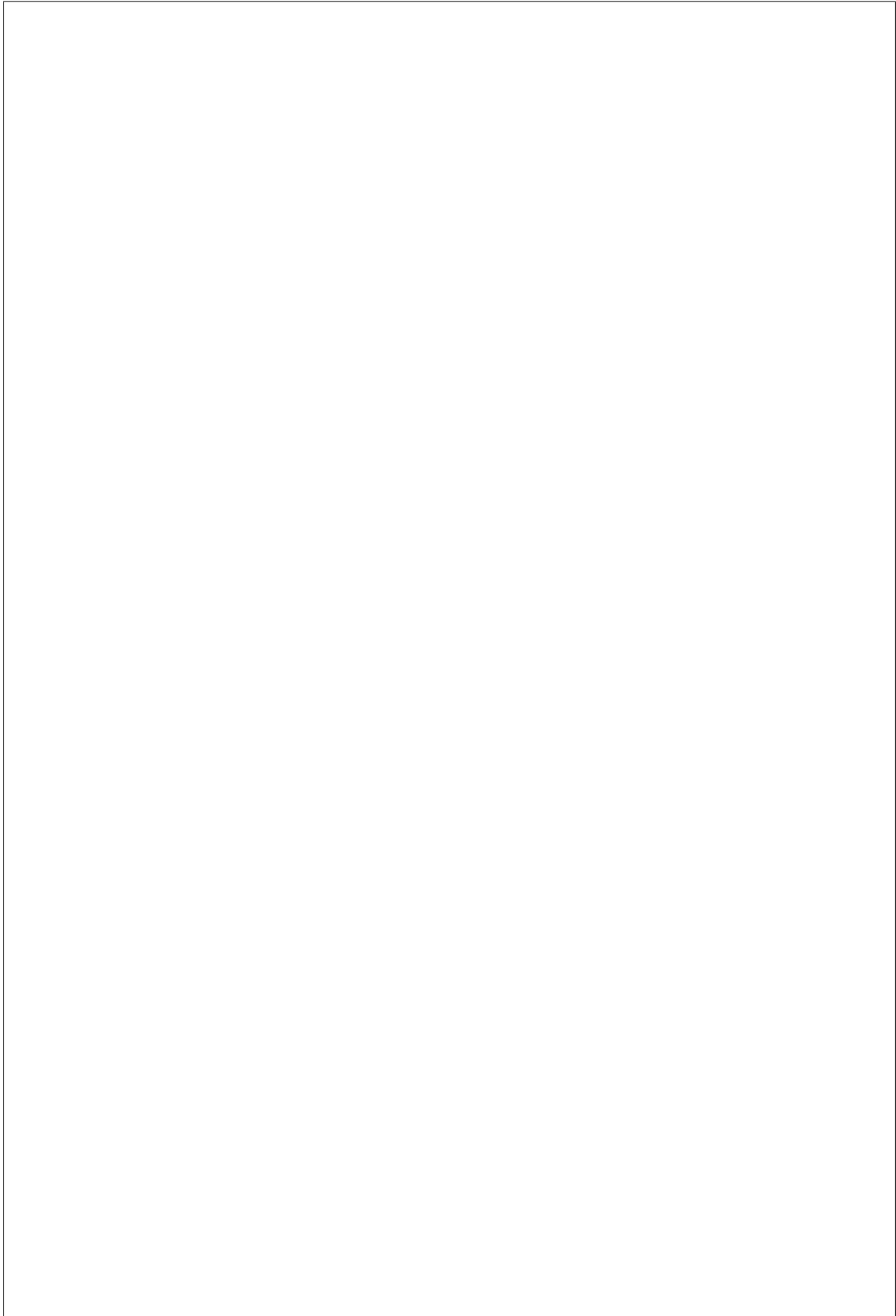




(continues on next page)



(continued from previous page)



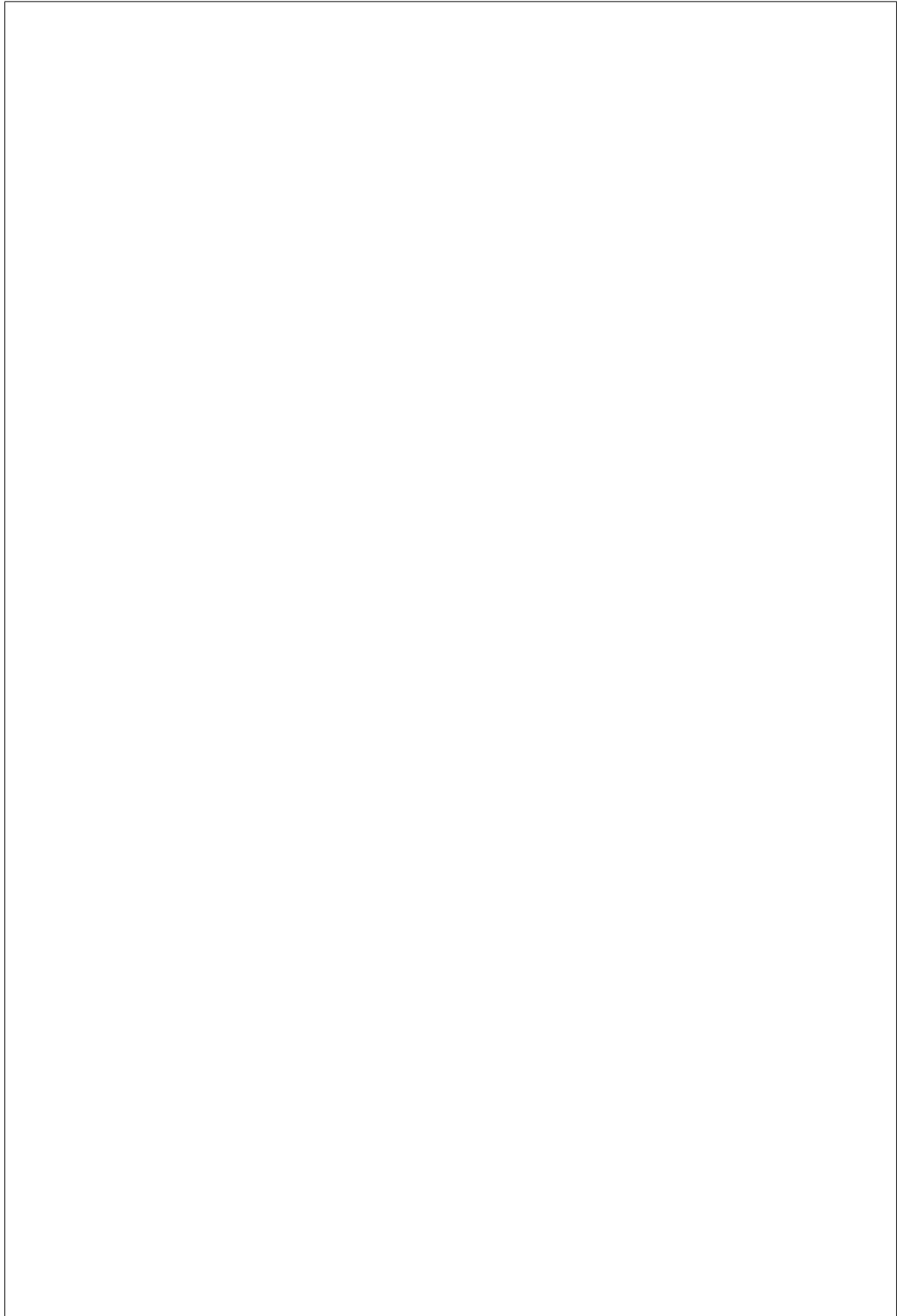
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

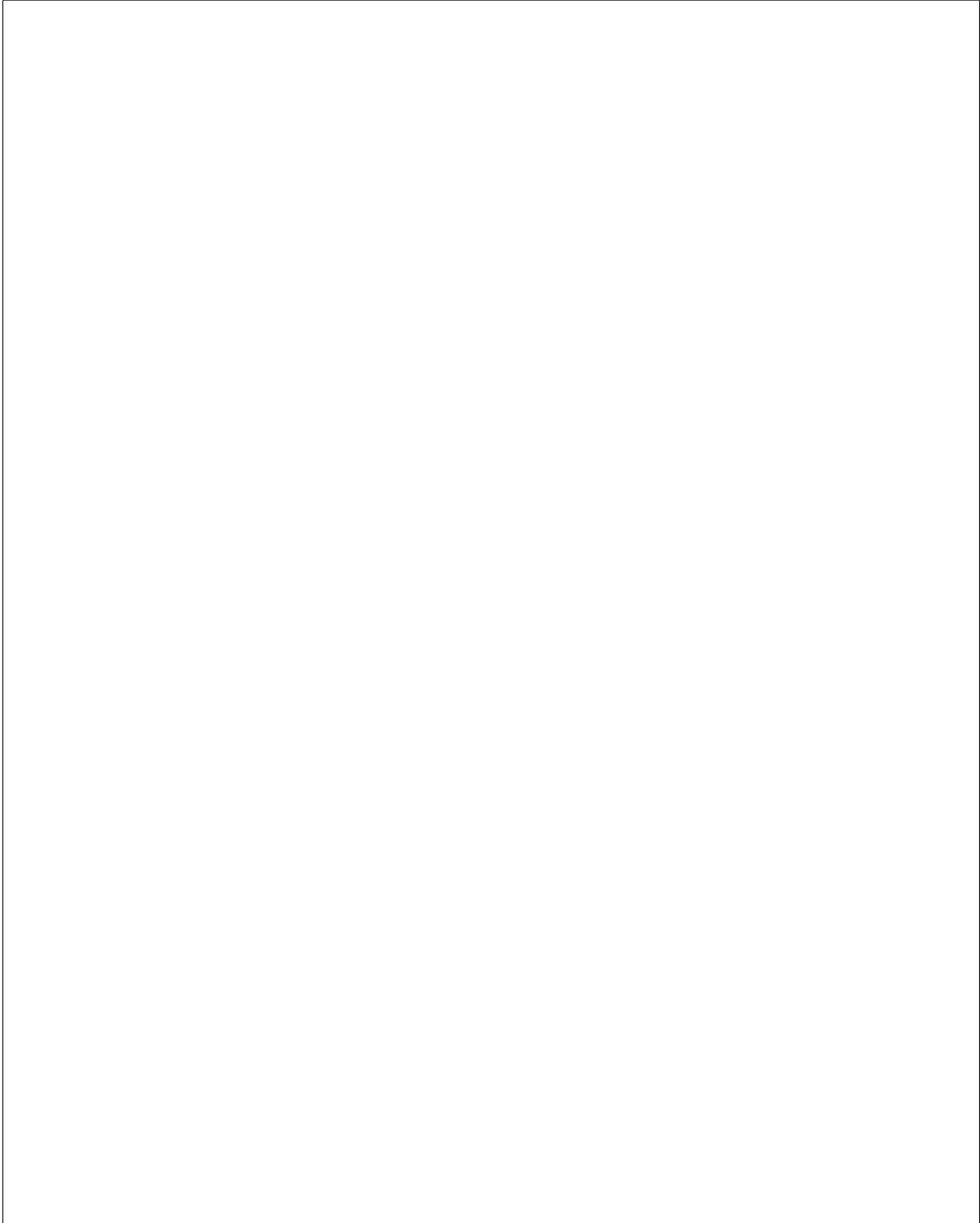


(continues on next page)

(continued from previous page)

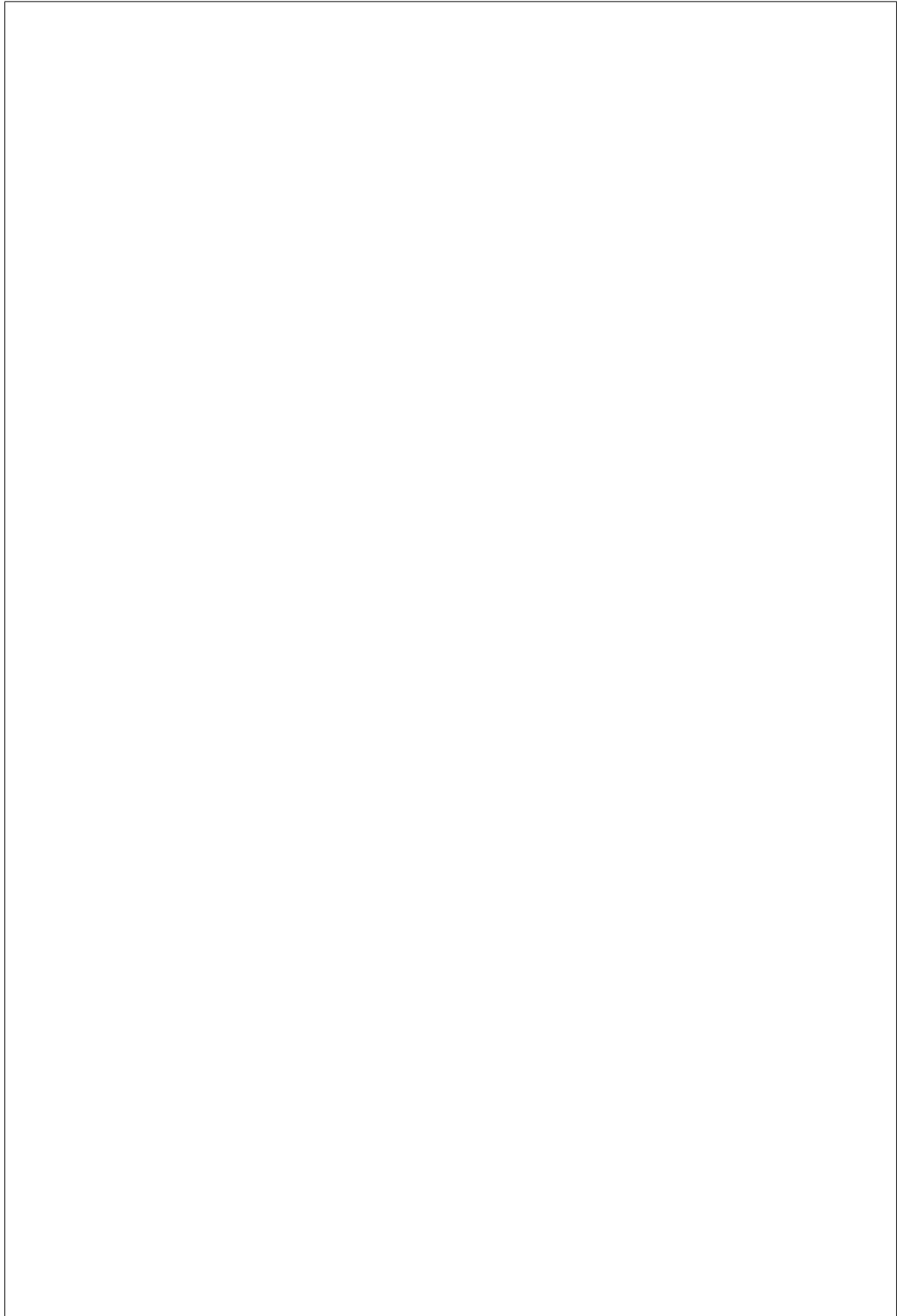






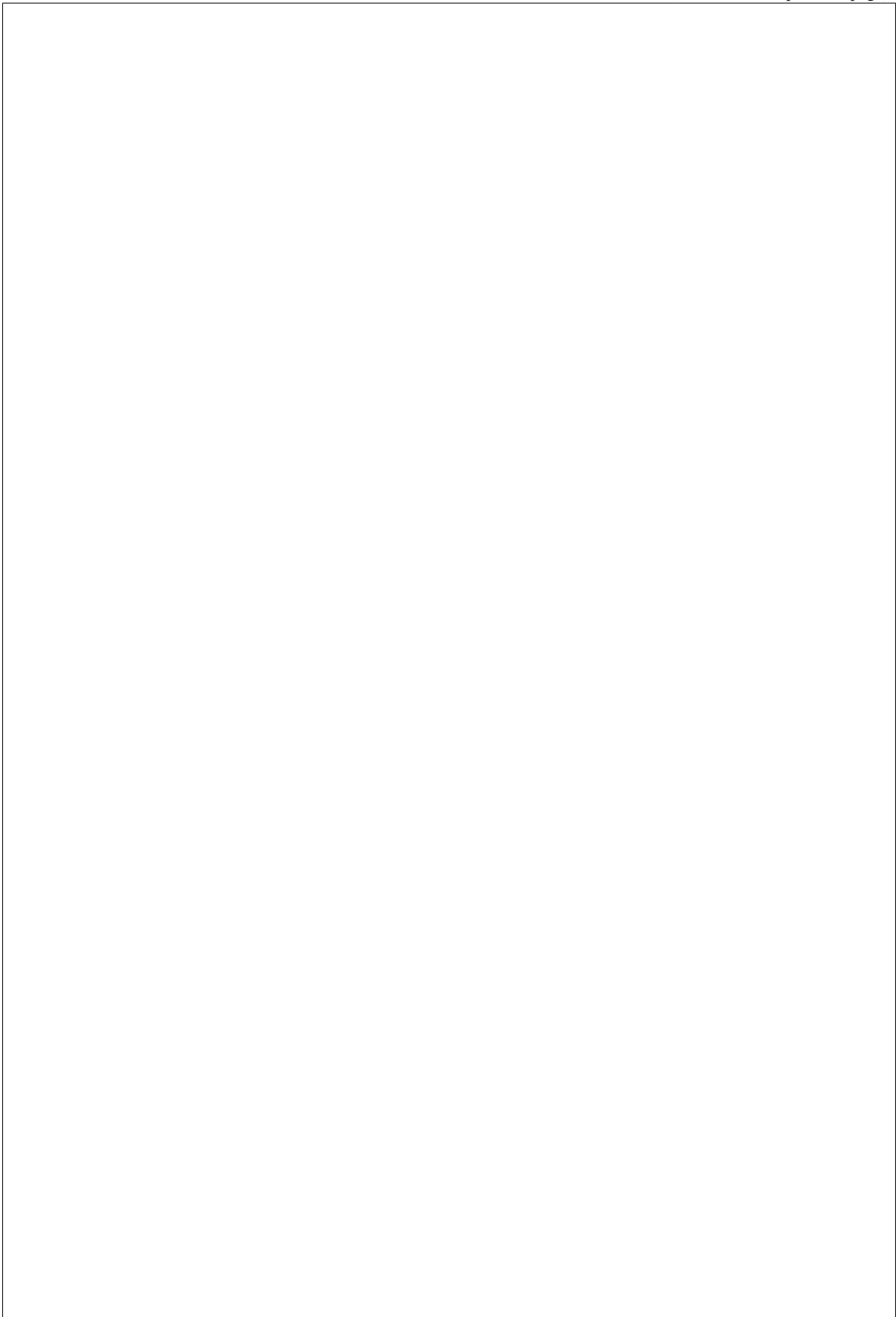
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



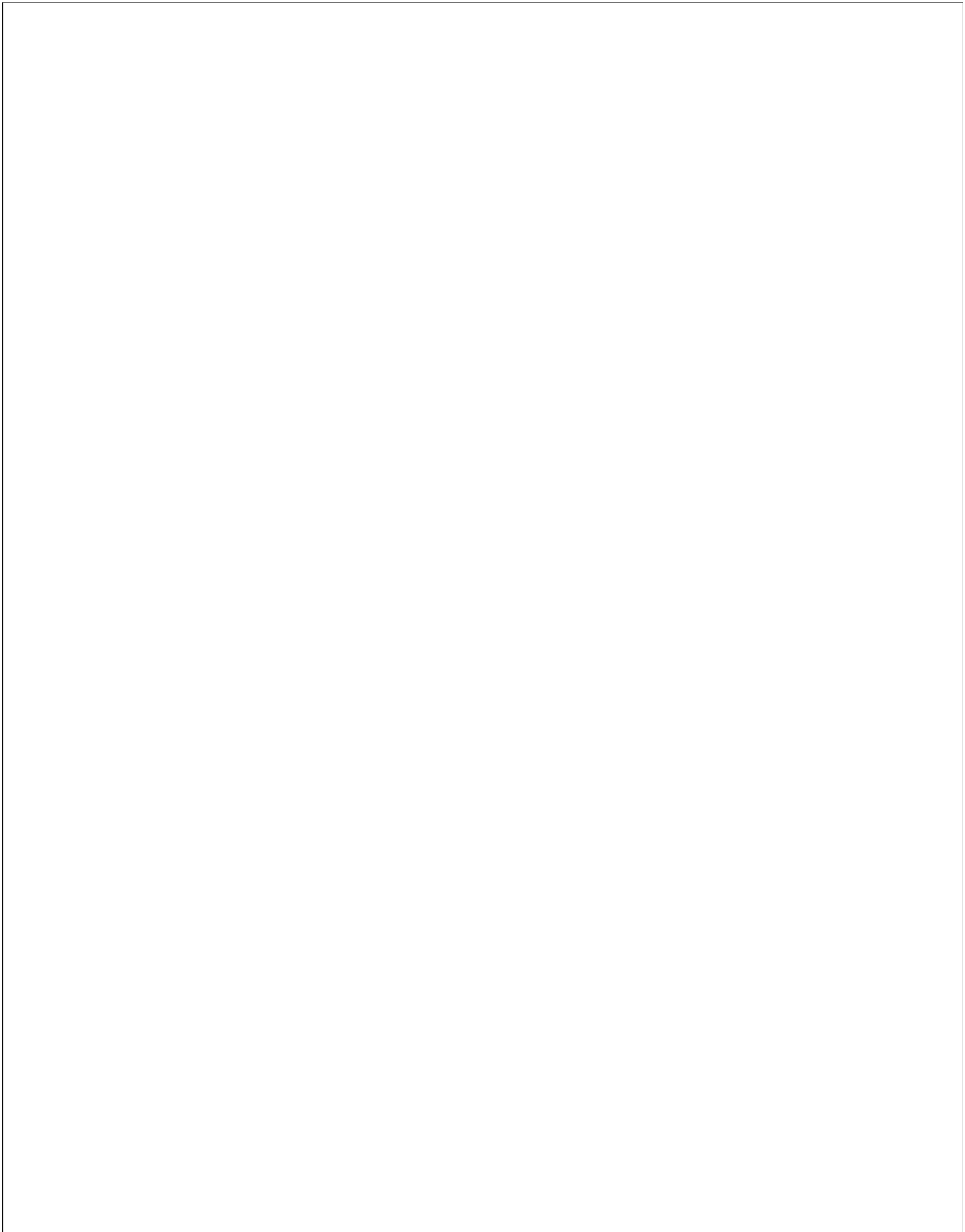
(continues on next page)



(continued from previous page)







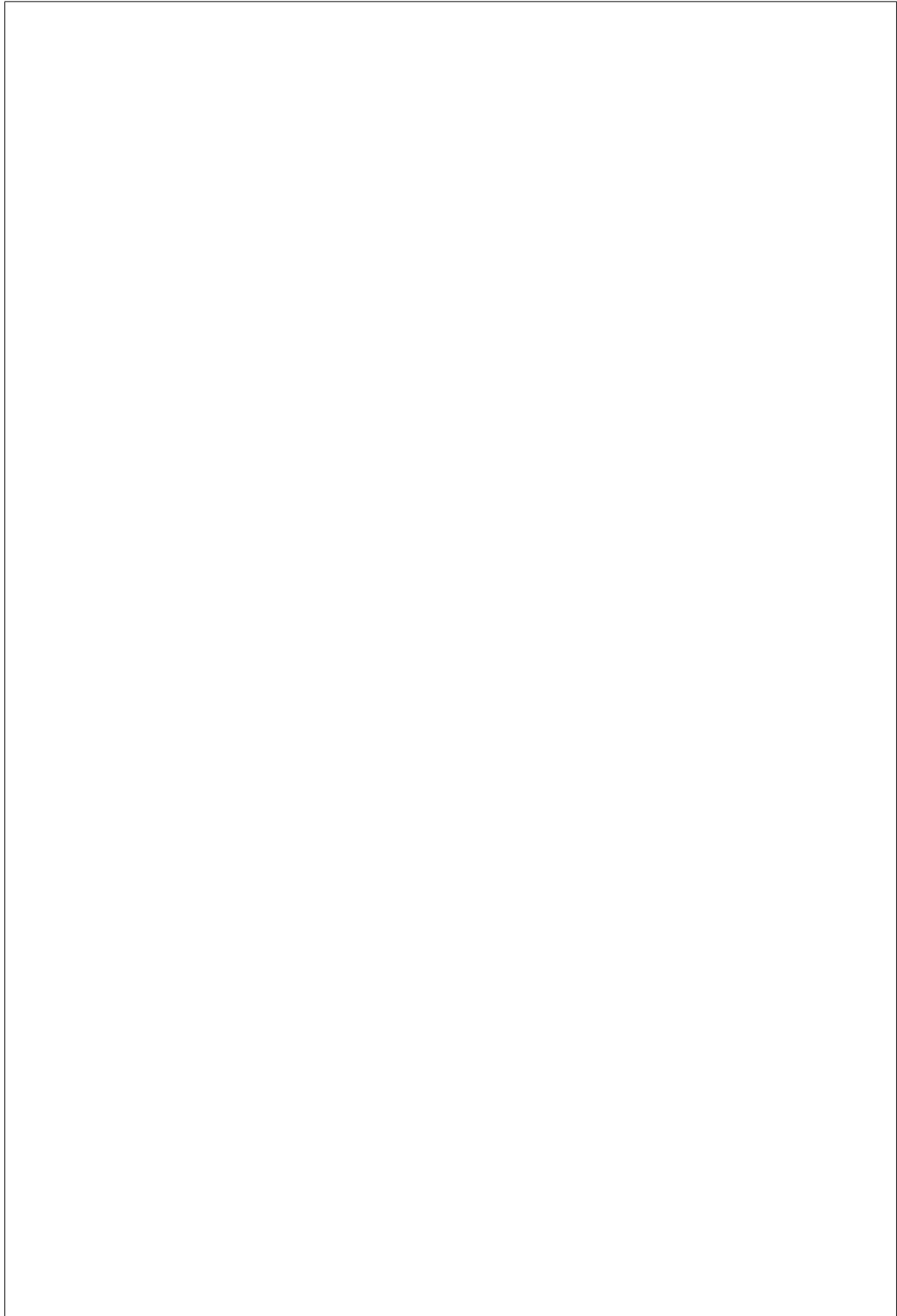
(continues on next page)

(continued from previous page)



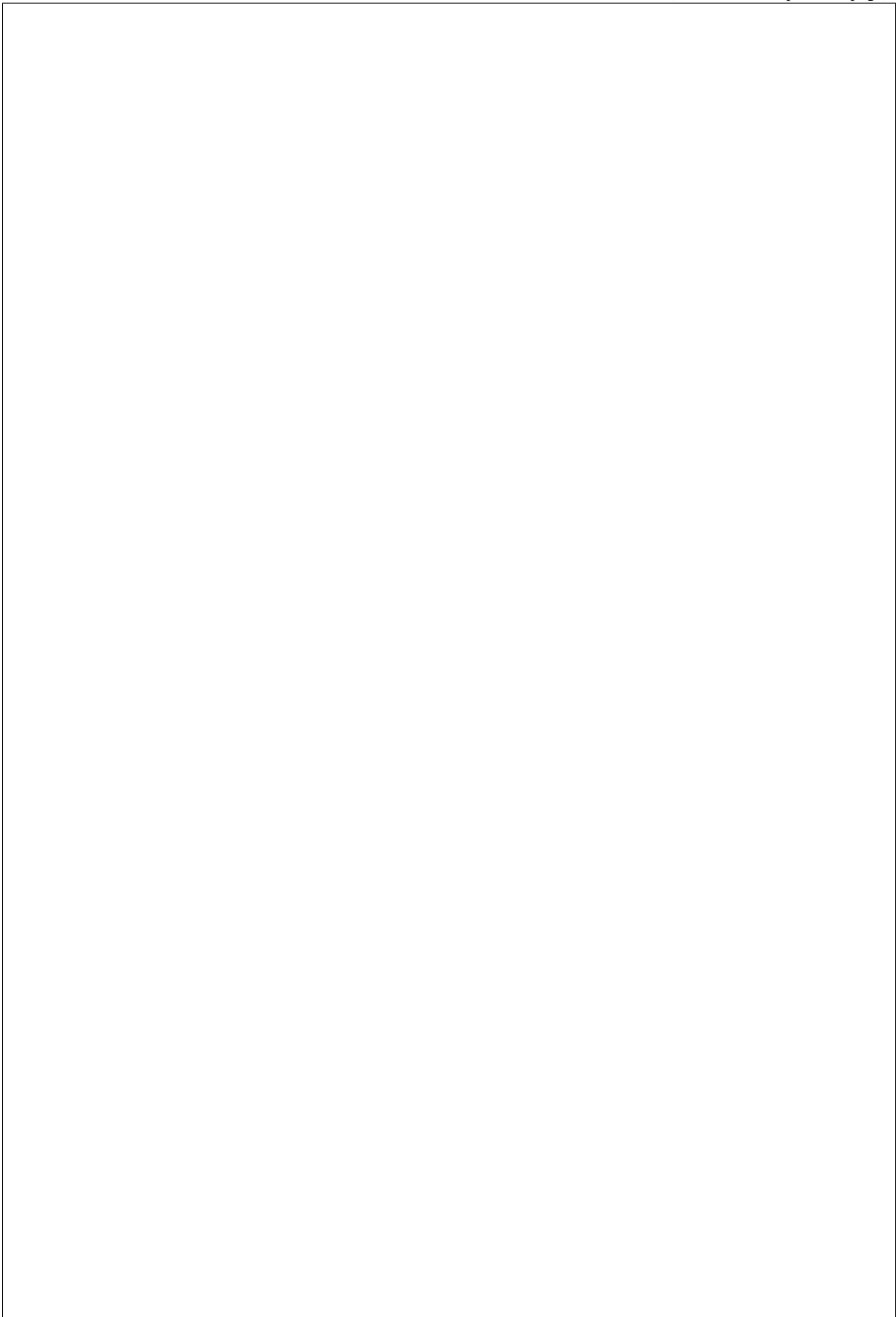
(continues on next page)

(continued from previous page)



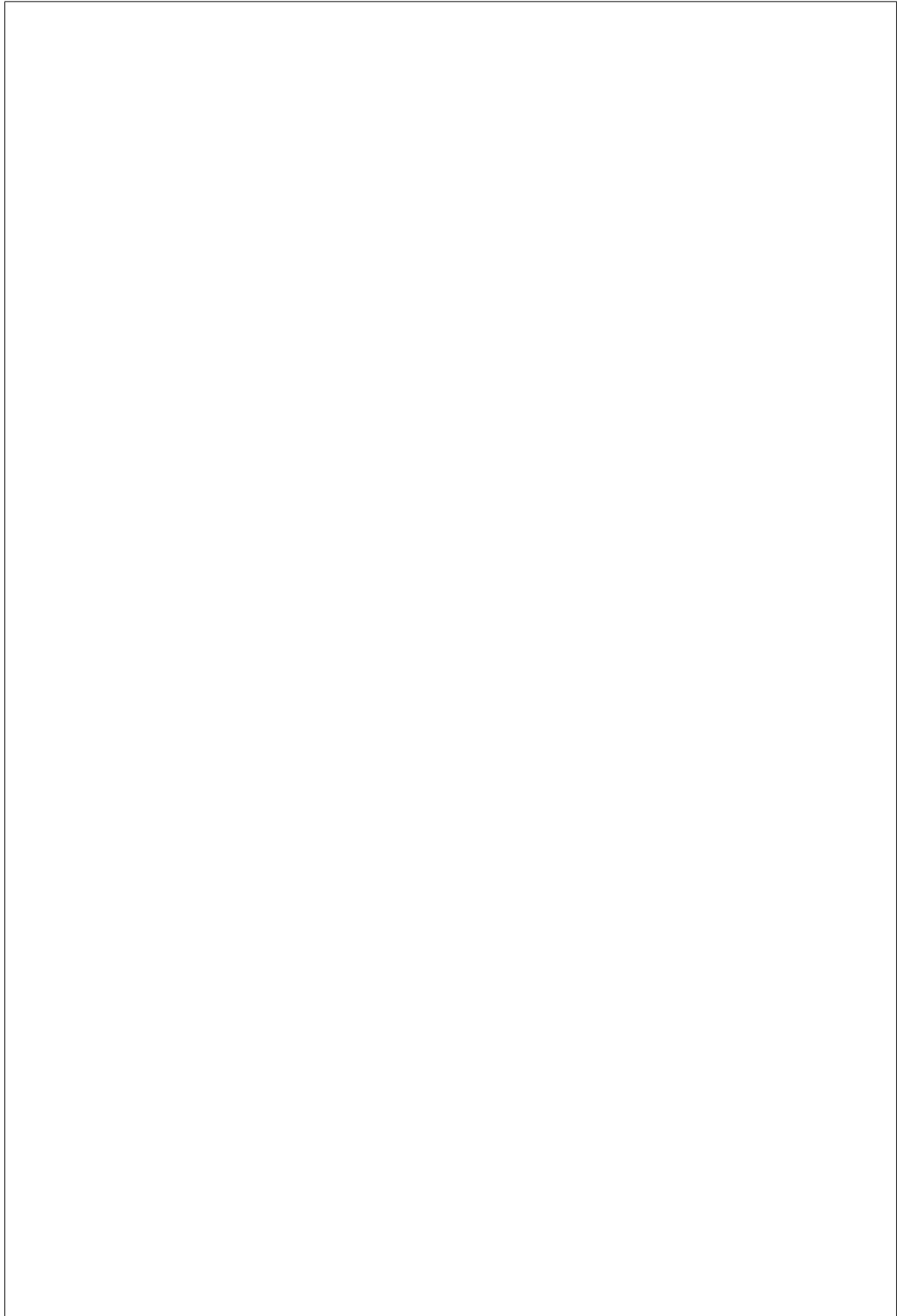
(continues on next page)

(continued from previous page)



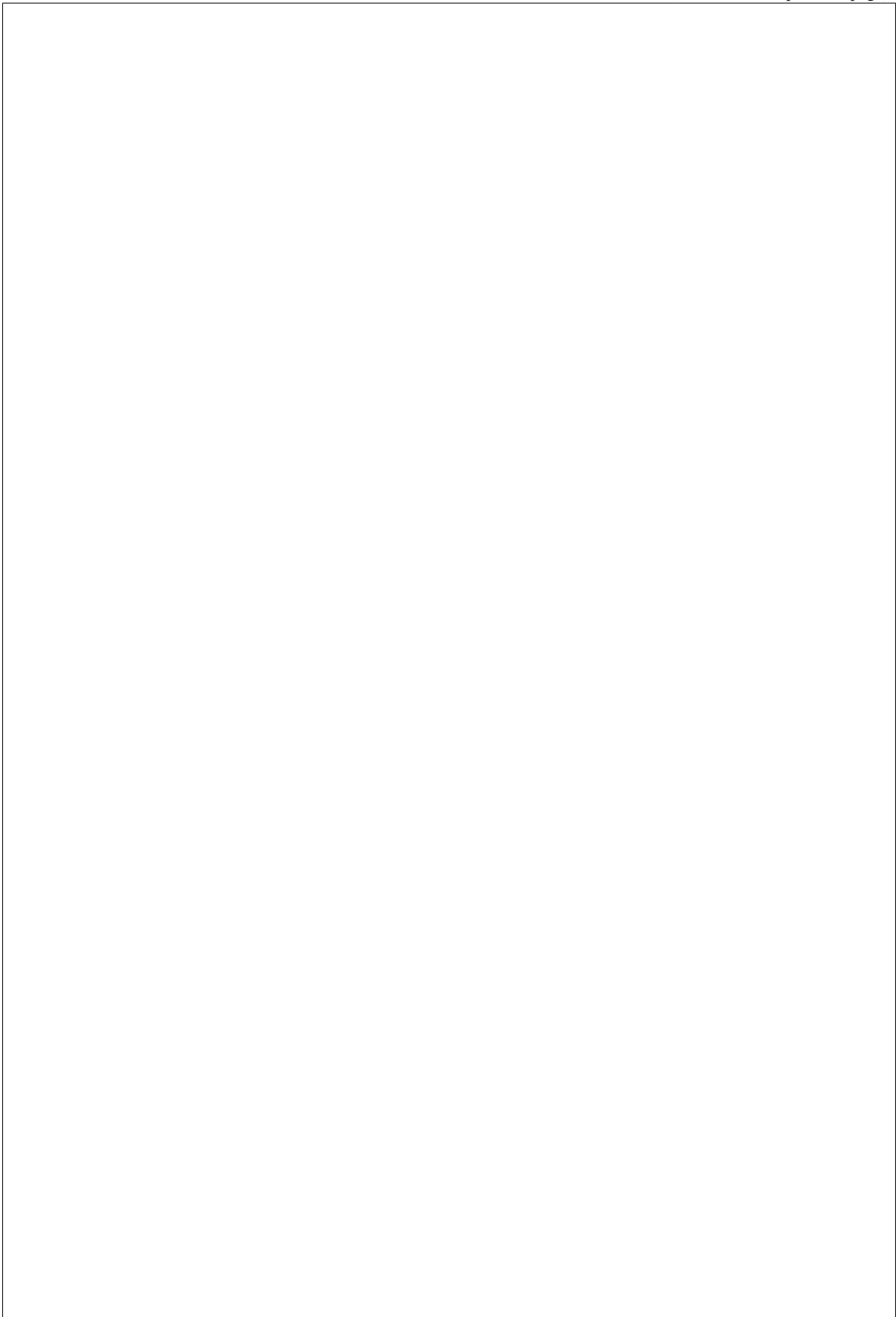
(continues on next page)

(continued from previous page)



(continues on next page)

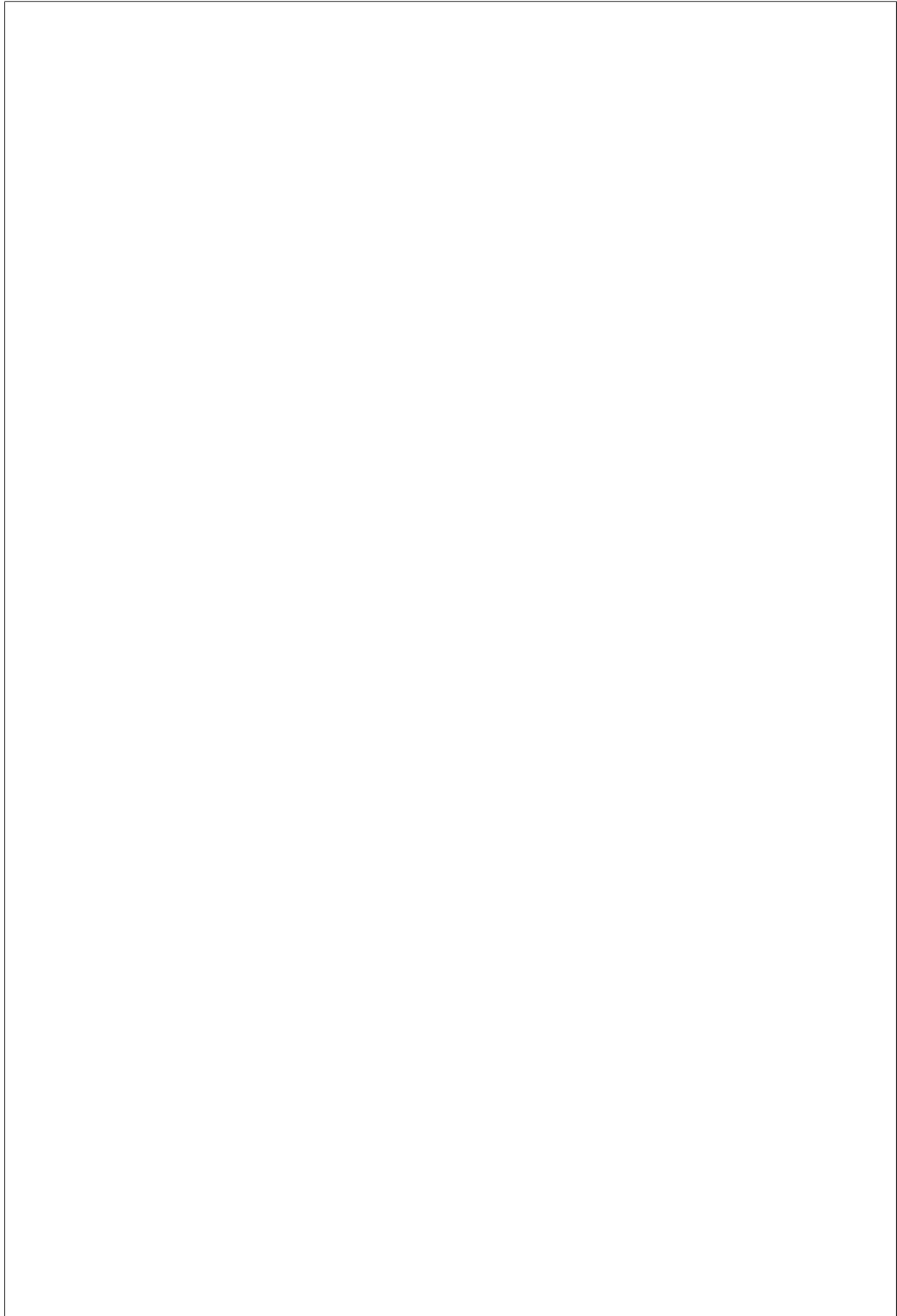
(continued from previous page)



(continues on next page)



(continued from previous page)



(continues on next page)

(continued from previous page)



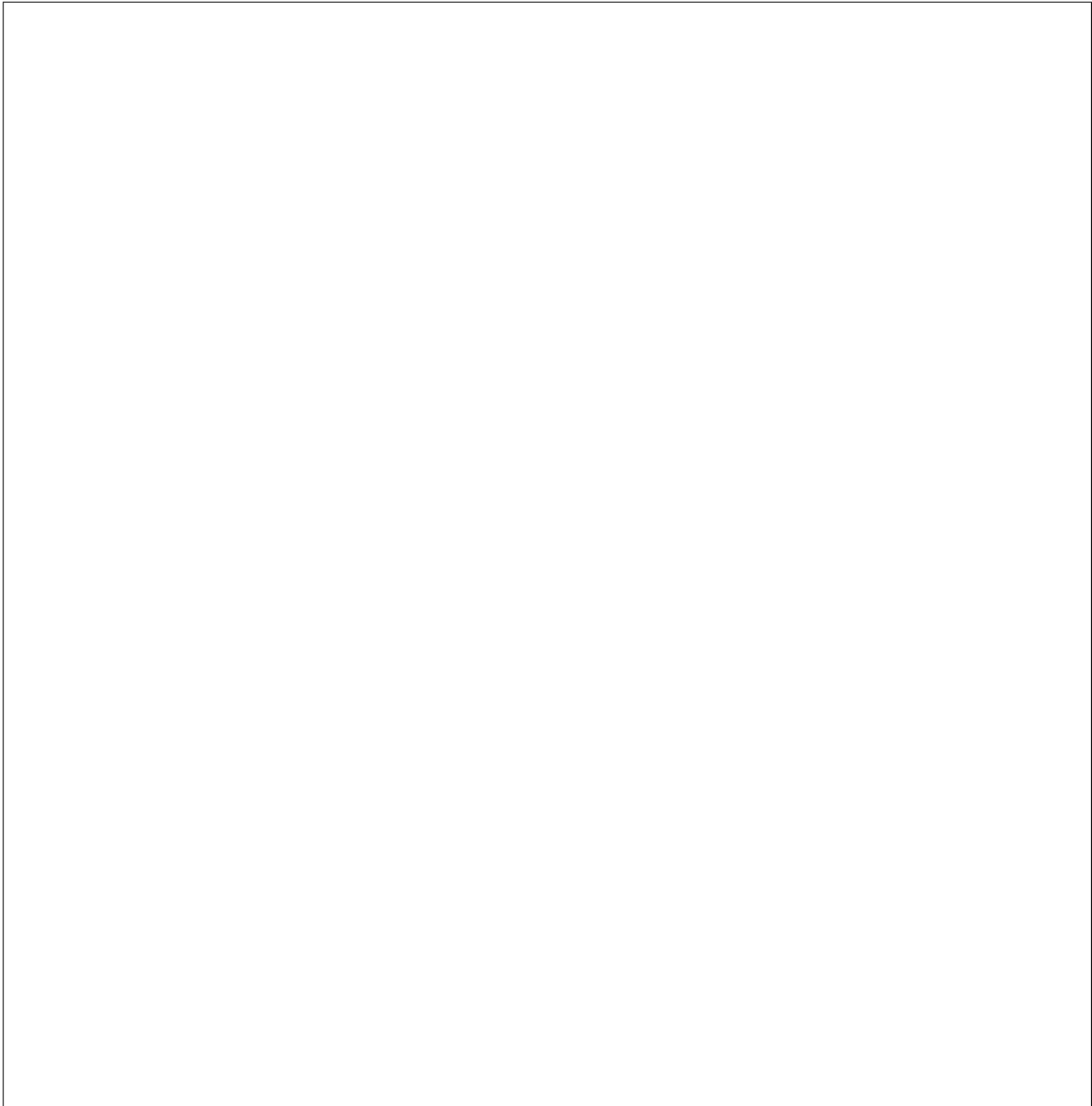
(continues on next page)

(continued from previous page)



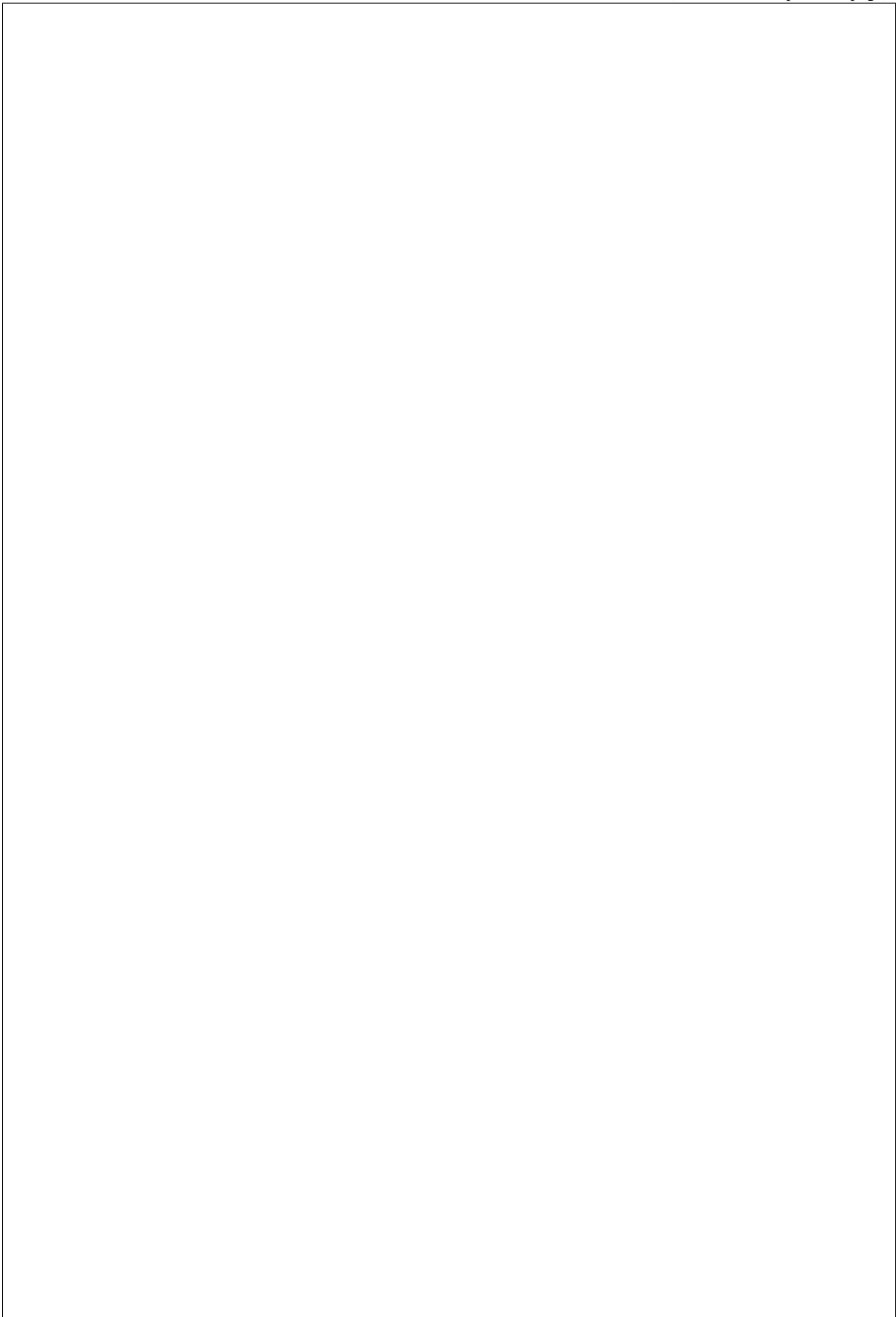
## **Node maintenance notifications**





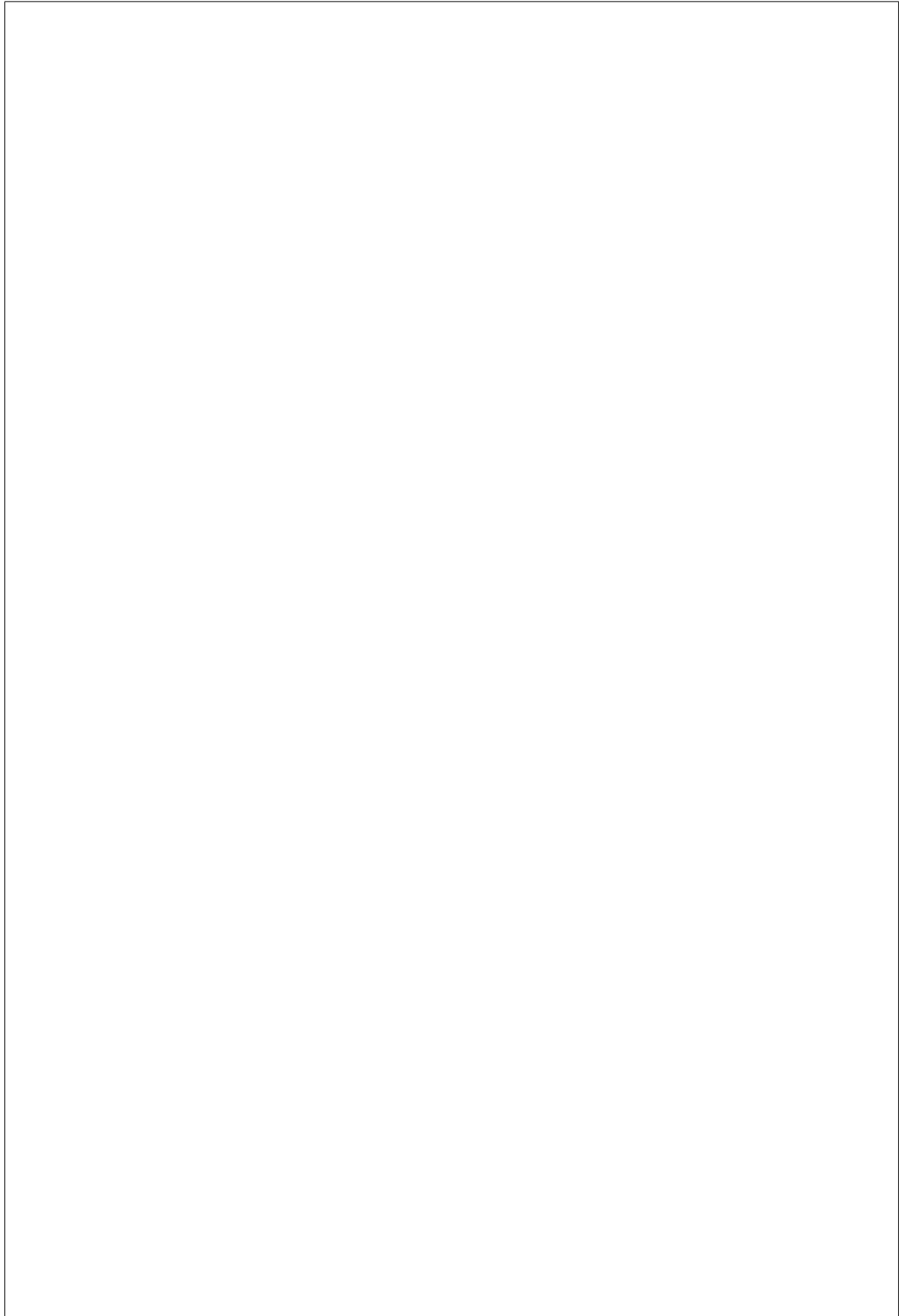
(continues on next page)

(continued from previous page)



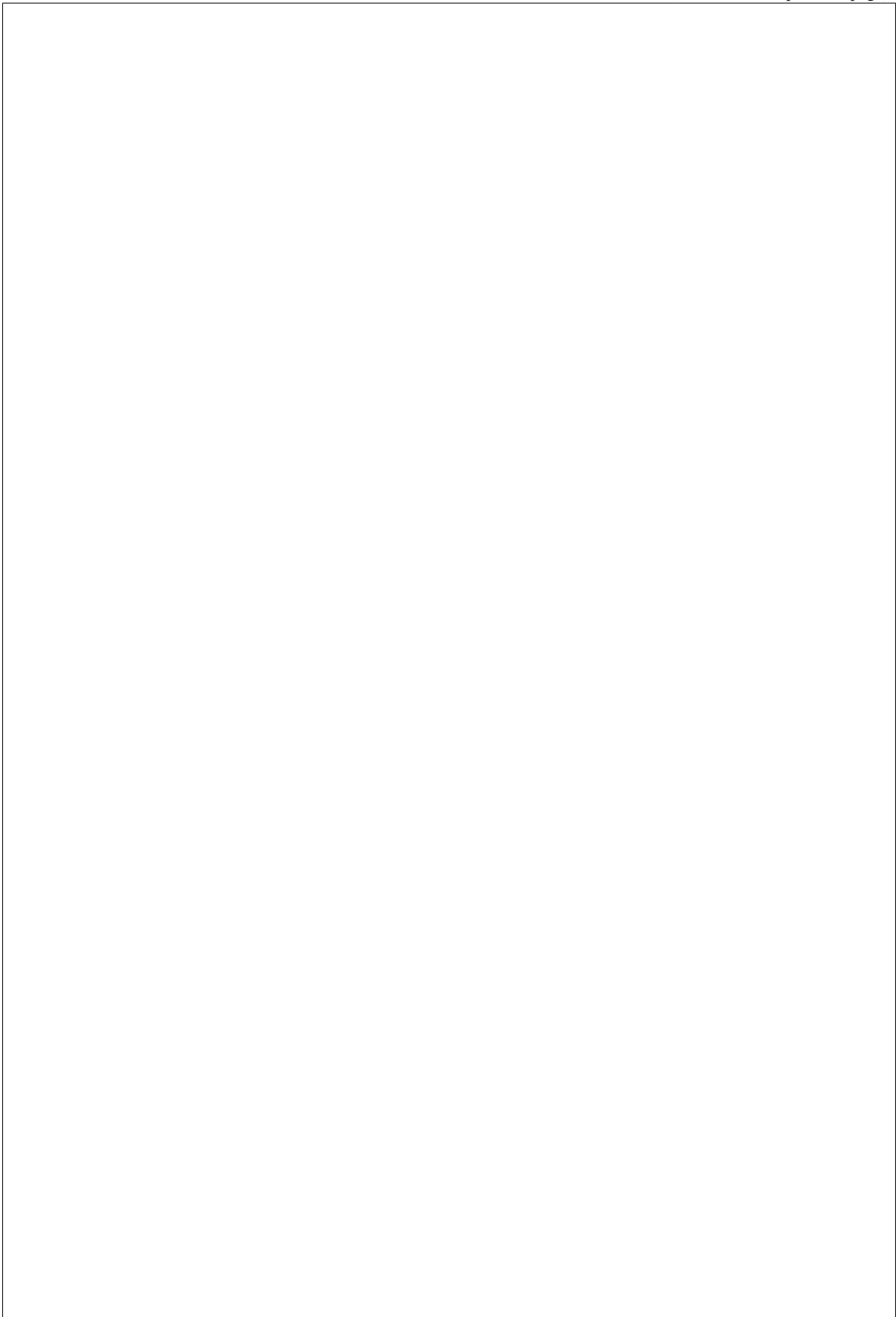
(continues on next page)

(continued from previous page)



(continues on next page)

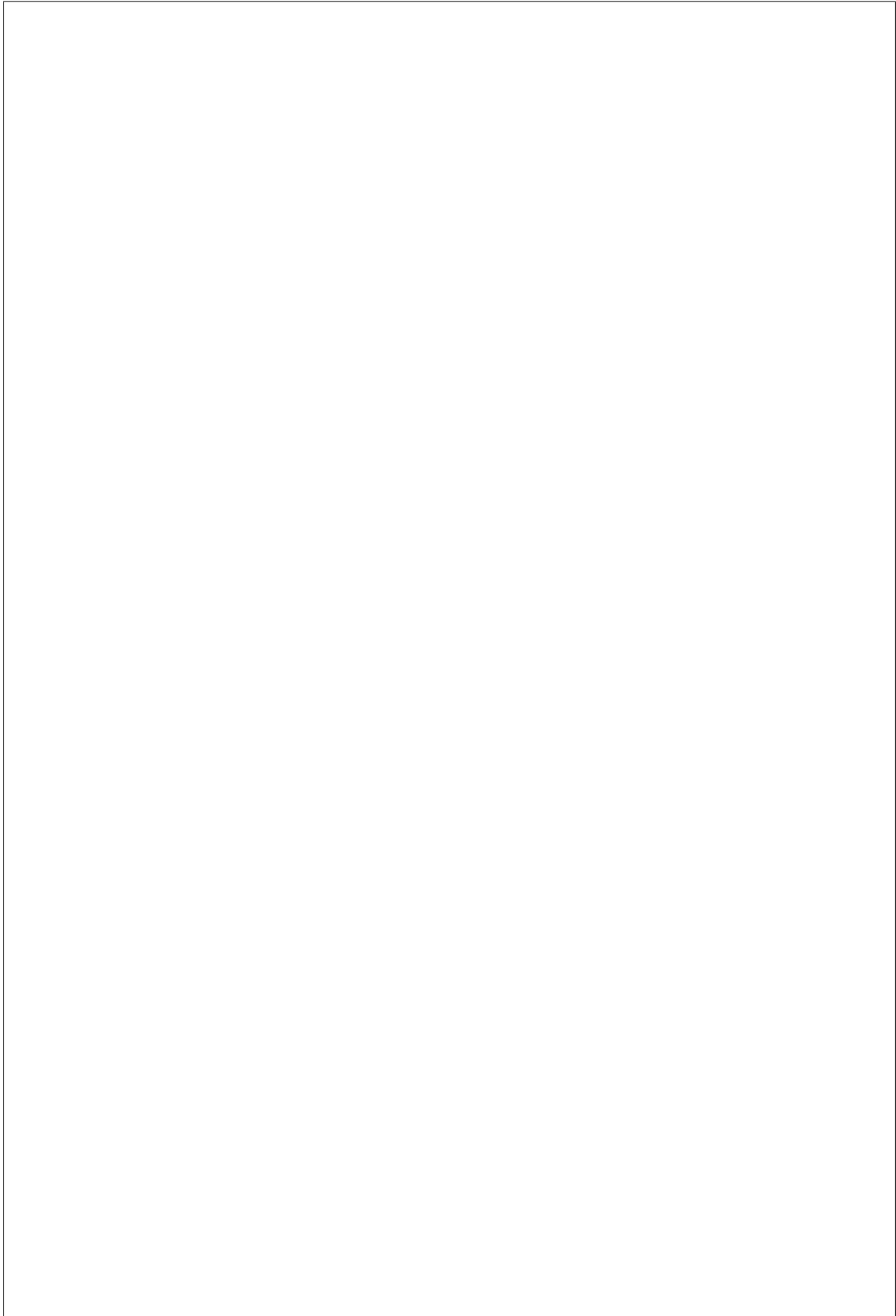
(continued from previous page)



(continues on next page)



(continued from previous page)



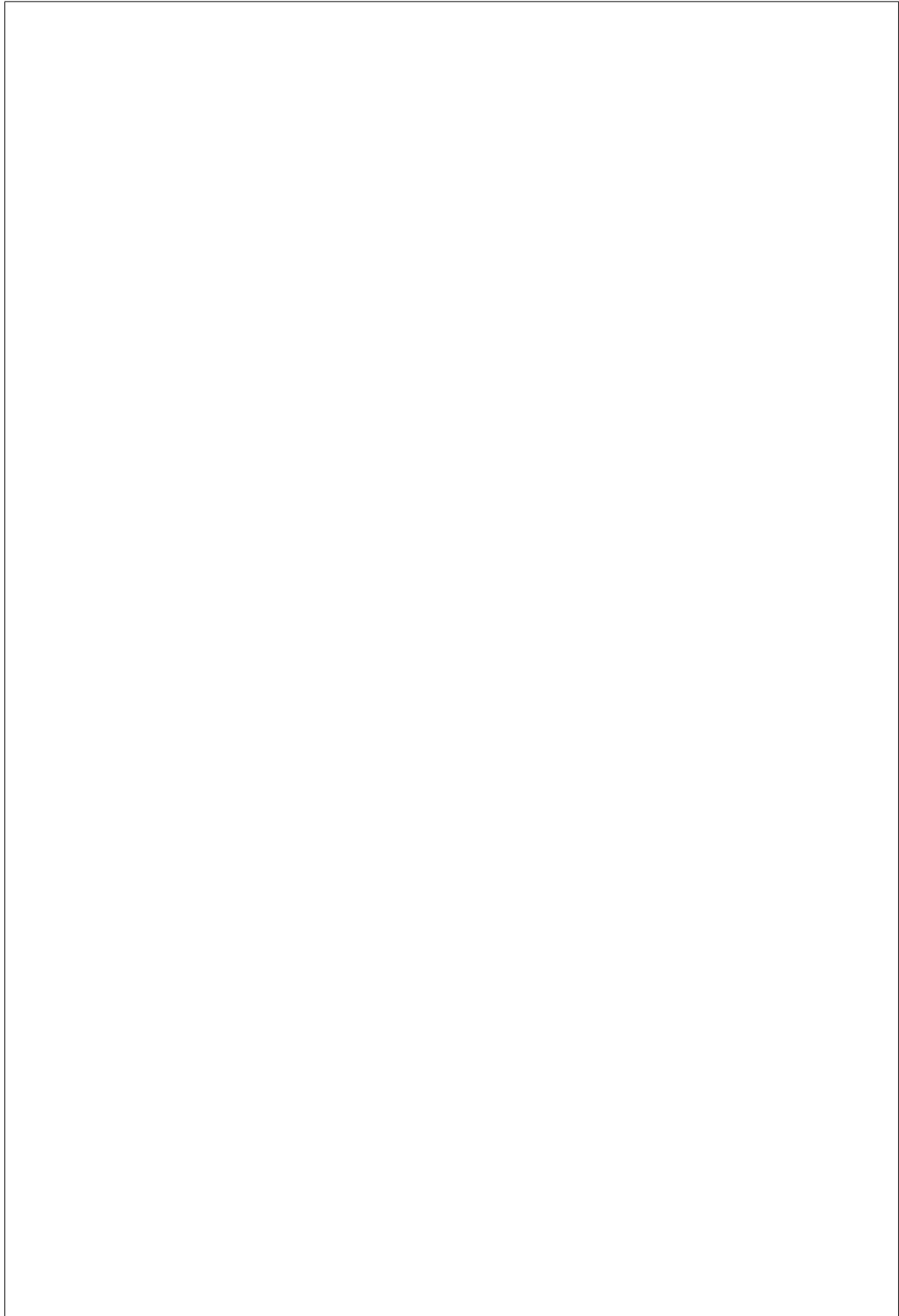
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



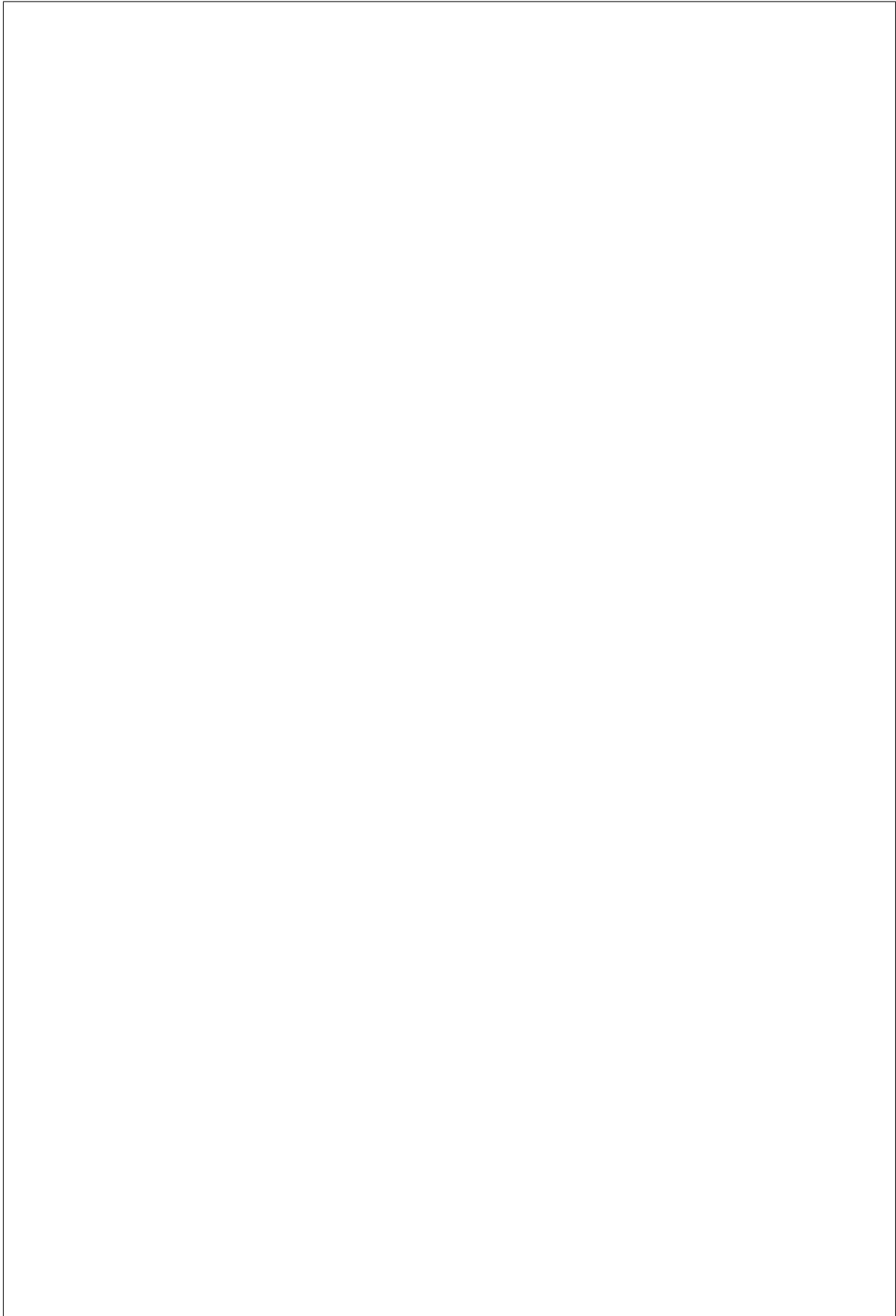
(continues on next page)

(continued from previous page)



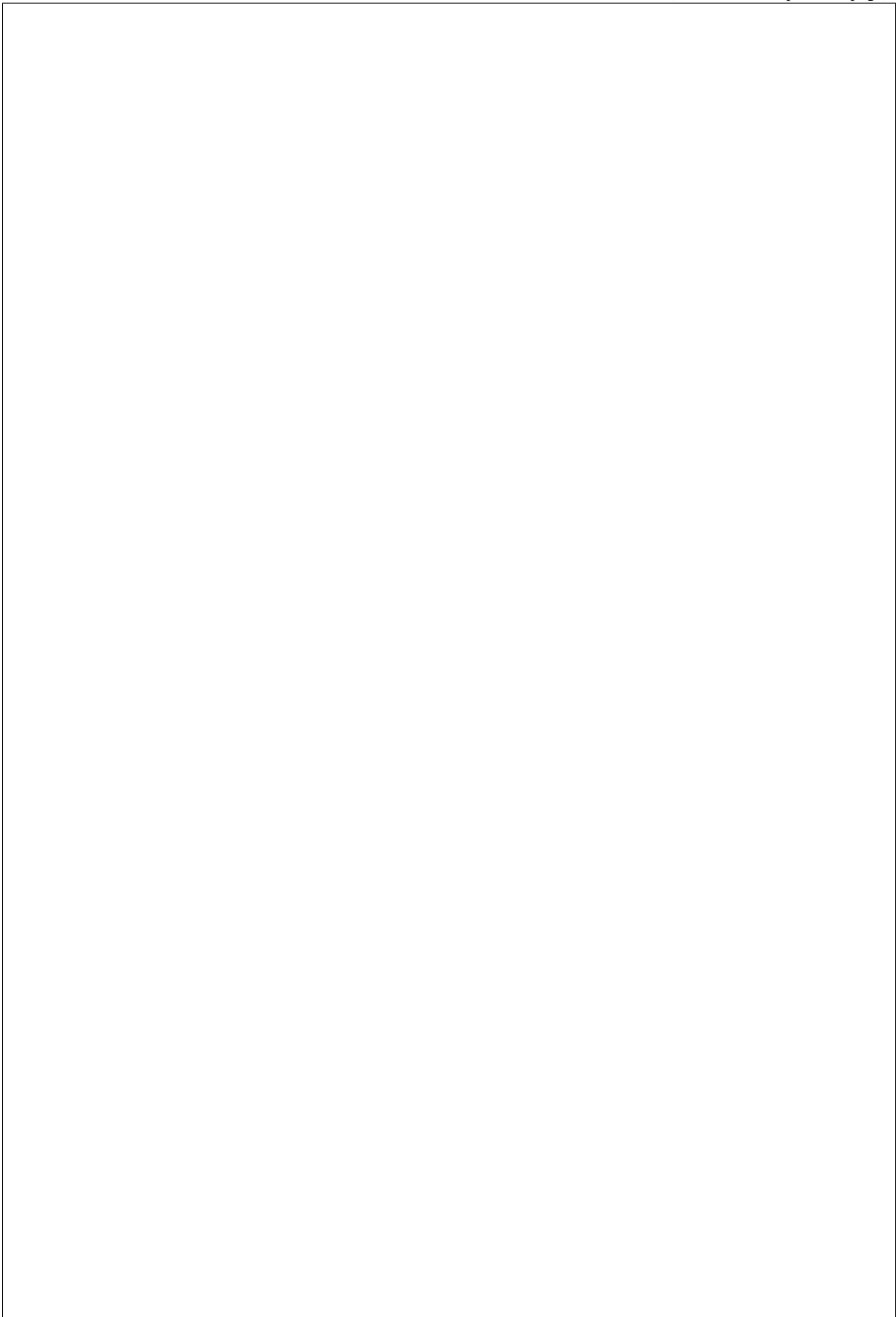
(continues on next page)

(continued from previous page)



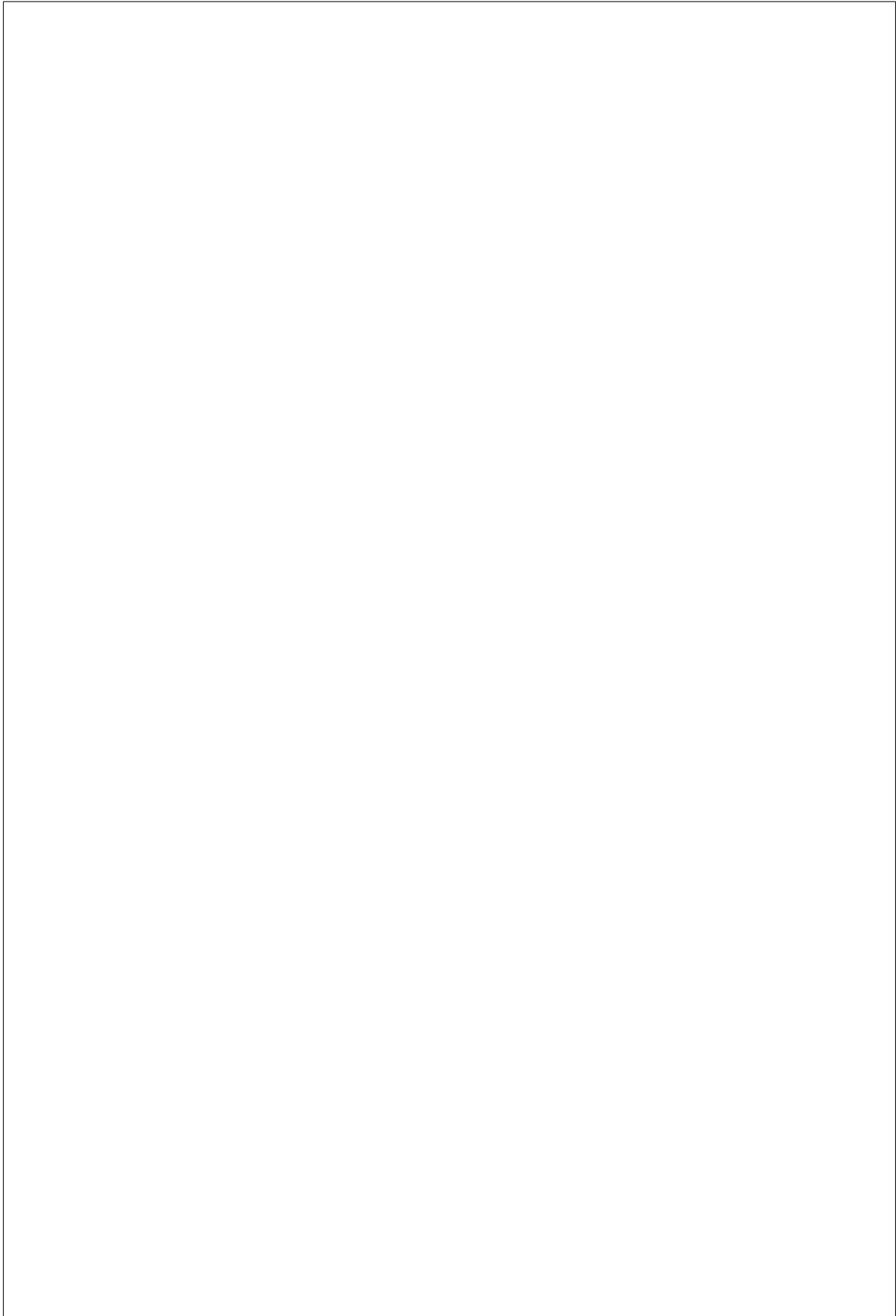
(continues on next page)

(continued from previous page)



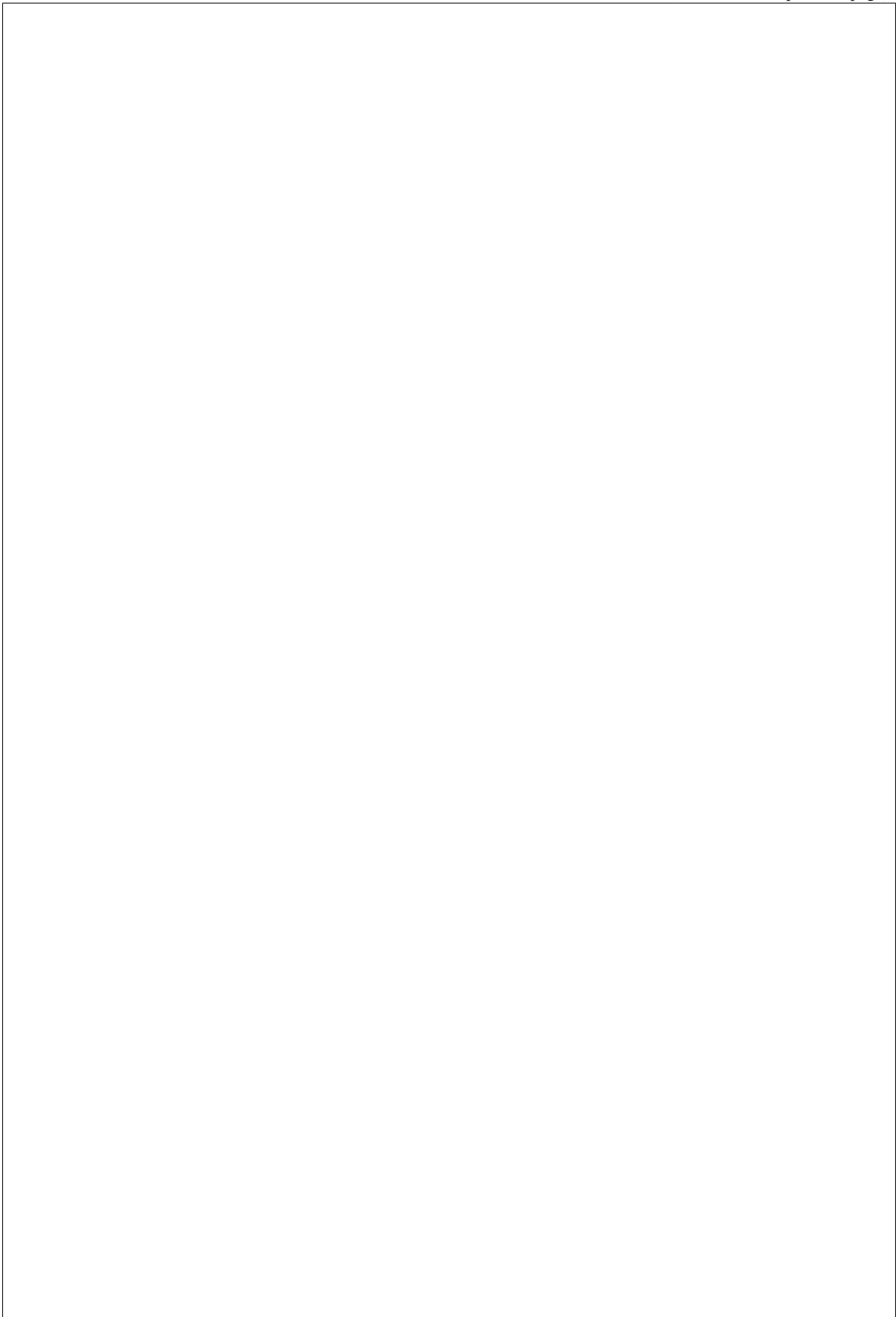
(continues on next page)

(continued from previous page)



(continues on next page)

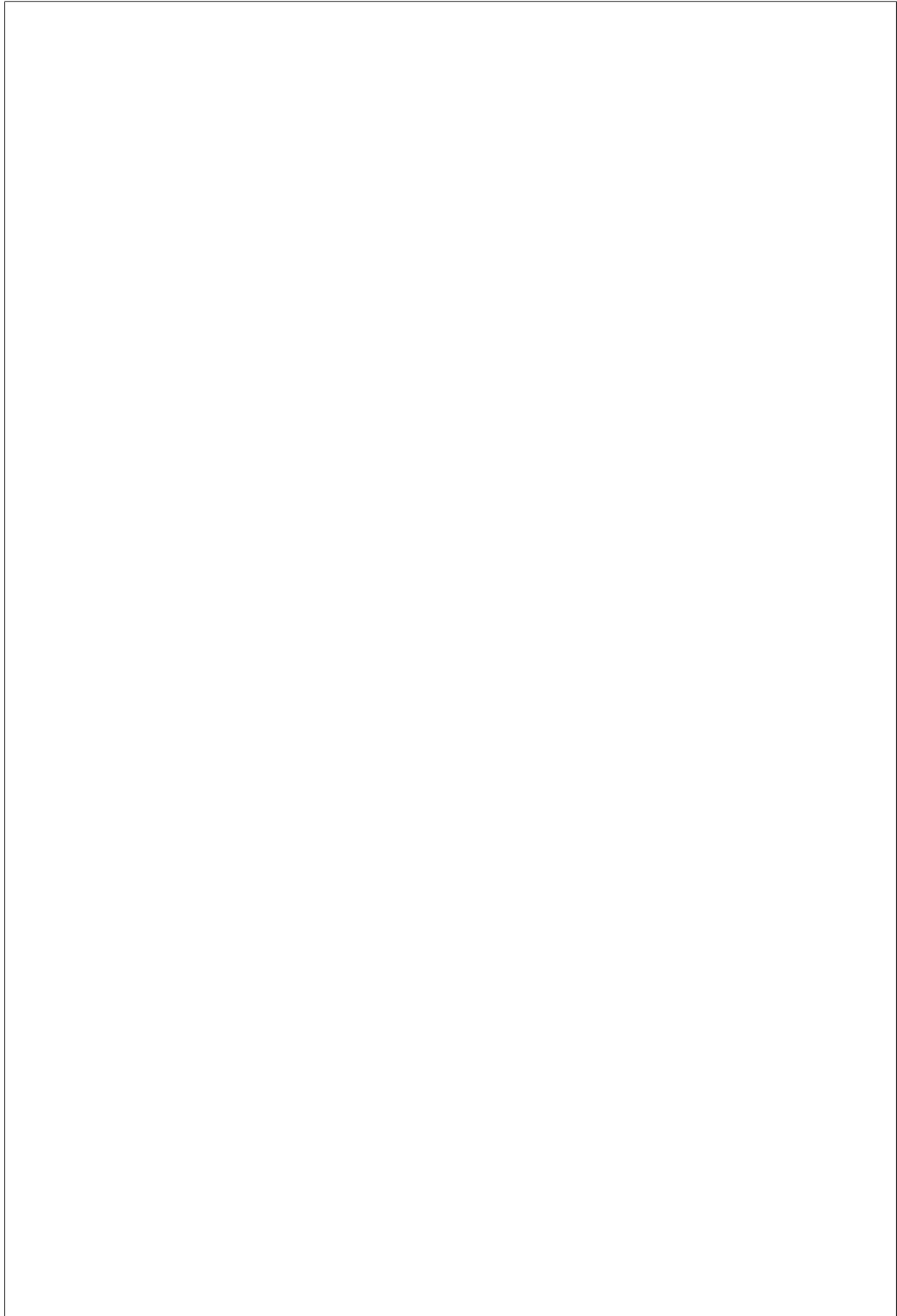
(continued from previous page)



(continues on next page)



(continued from previous page)



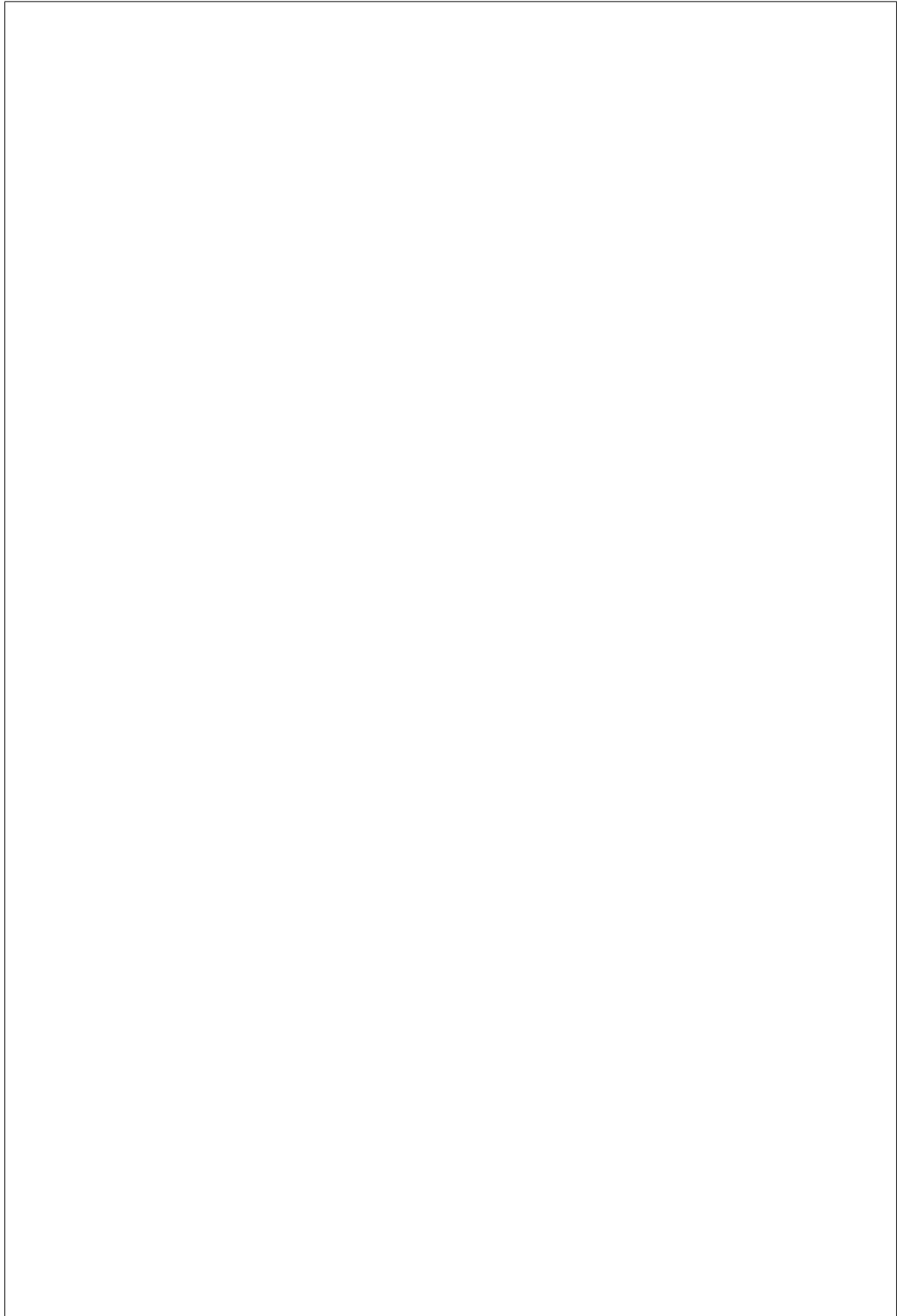
(continues on next page)

(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**

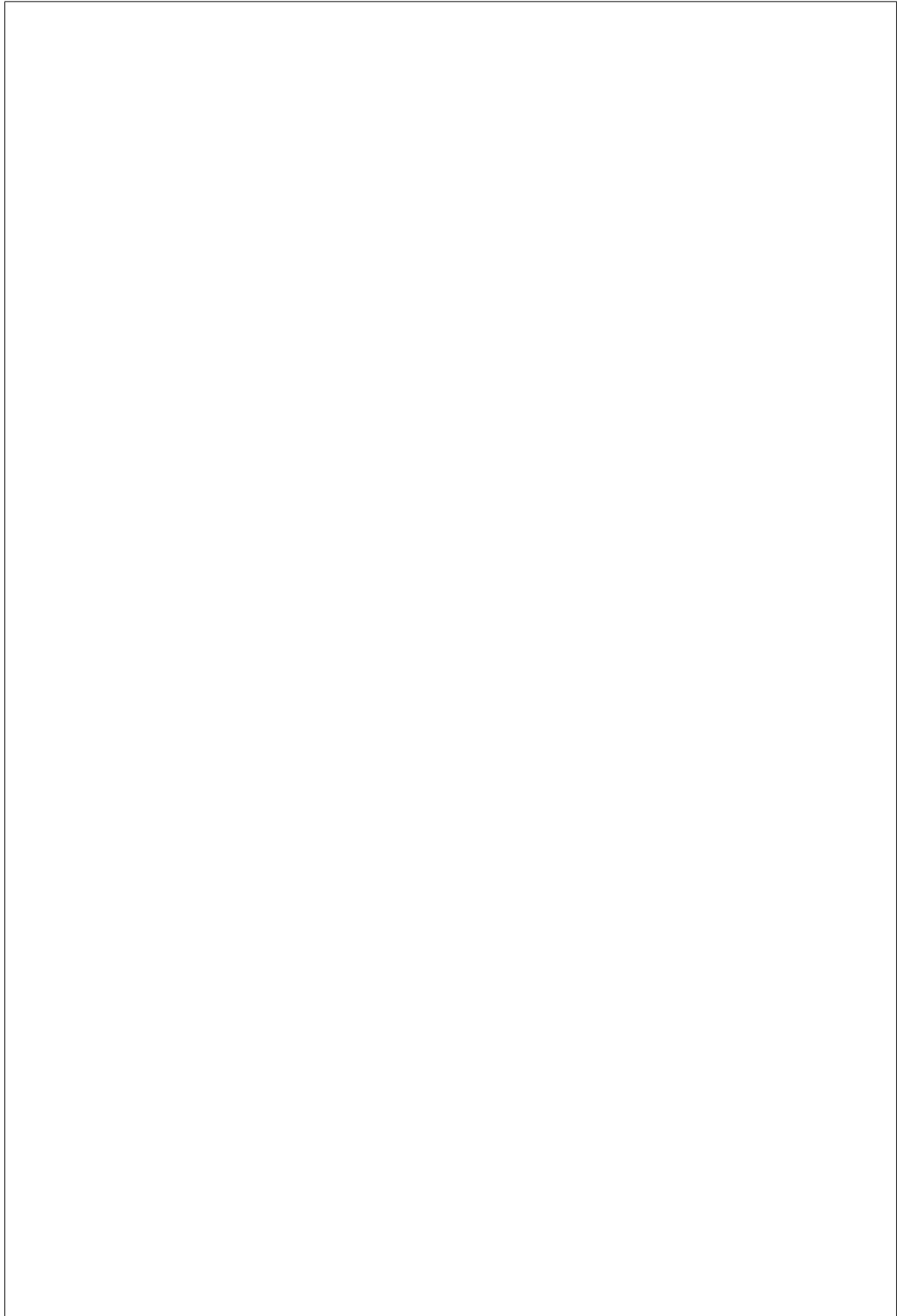


takes over a node that was being managed by another ironic-conductor. start and end notifications have INFO level, error has ERROR. Example of node console notification:



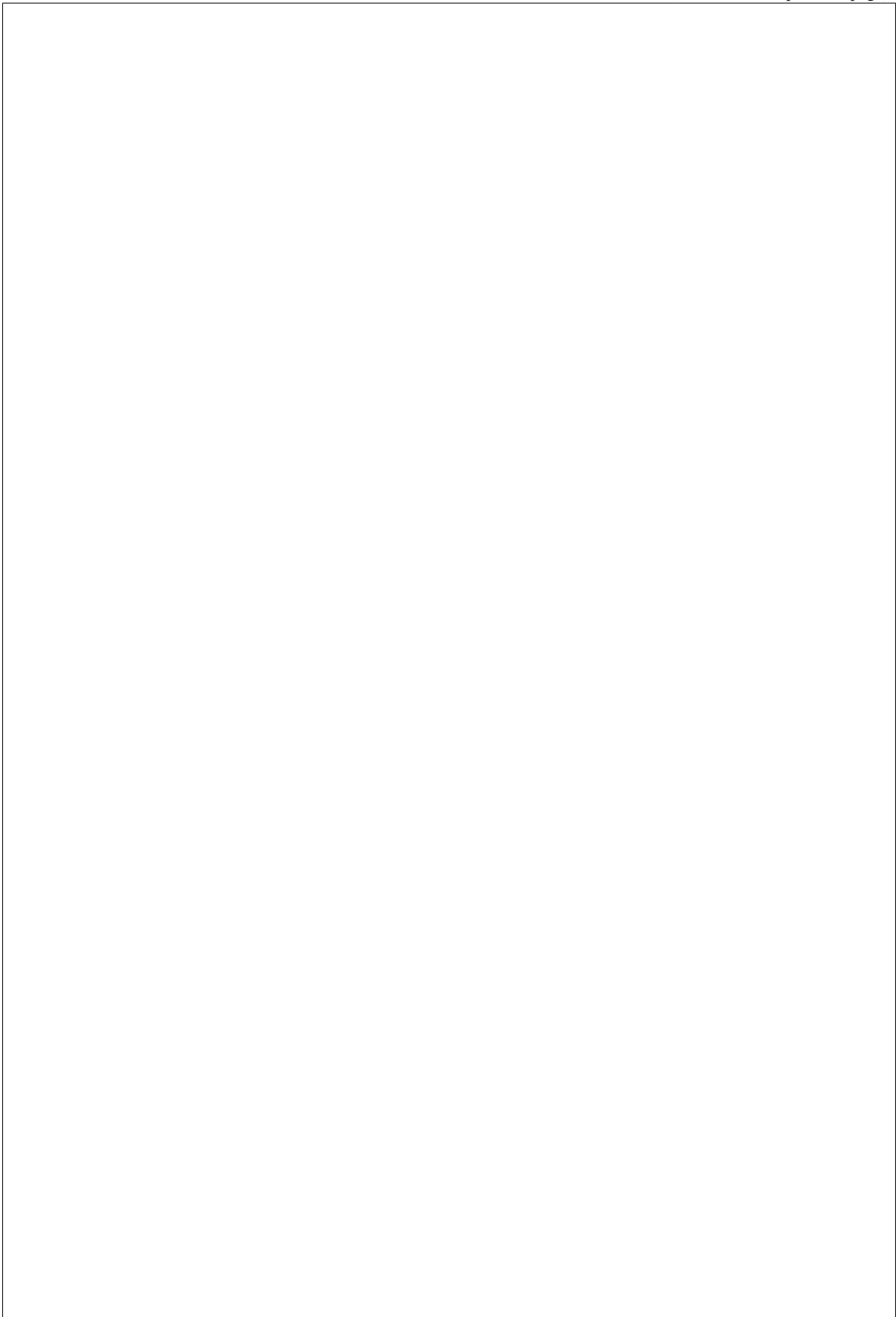
(continues on next page)

(continued from previous page)



(continues on next page)

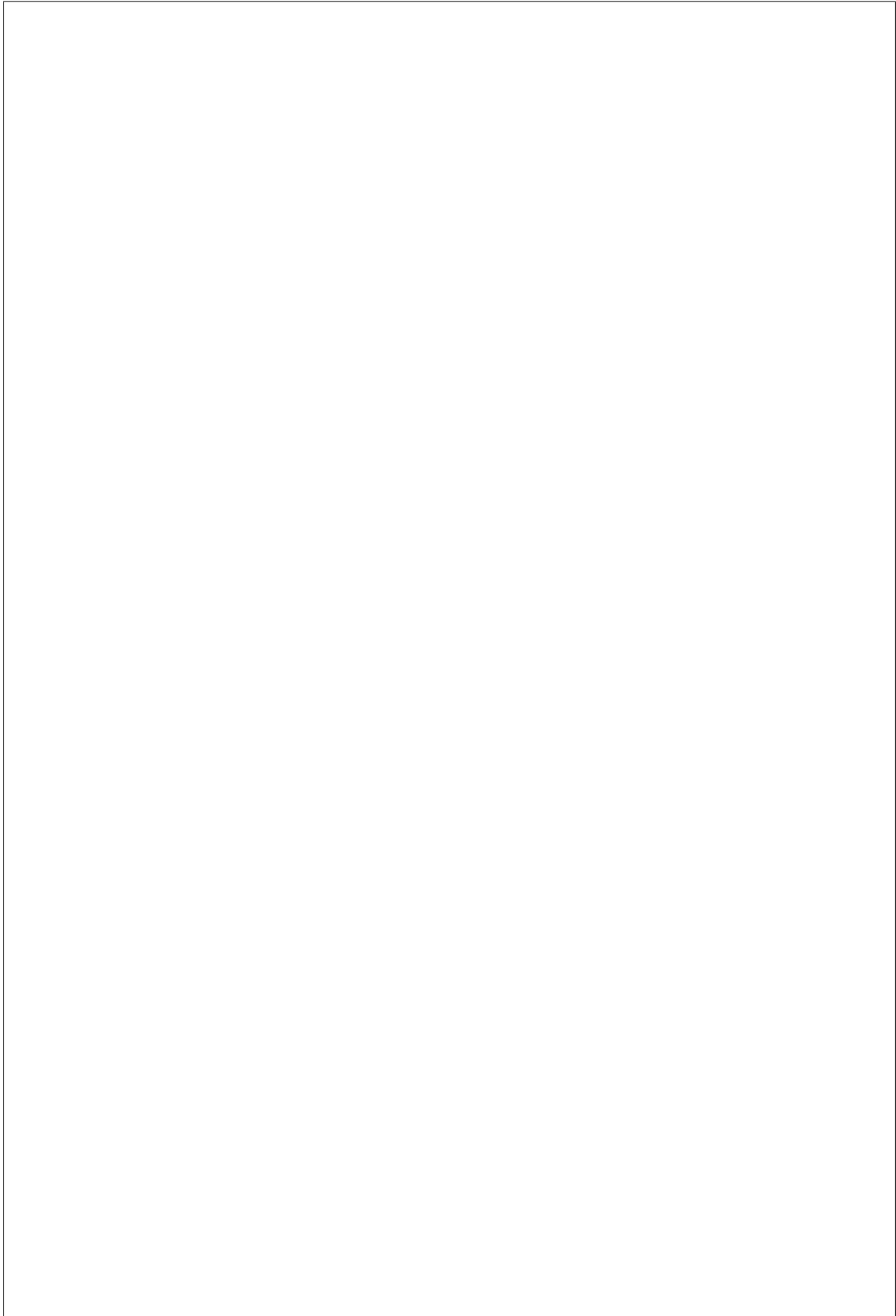
(continued from previous page)



(continues on next page)



(continued from previous page)



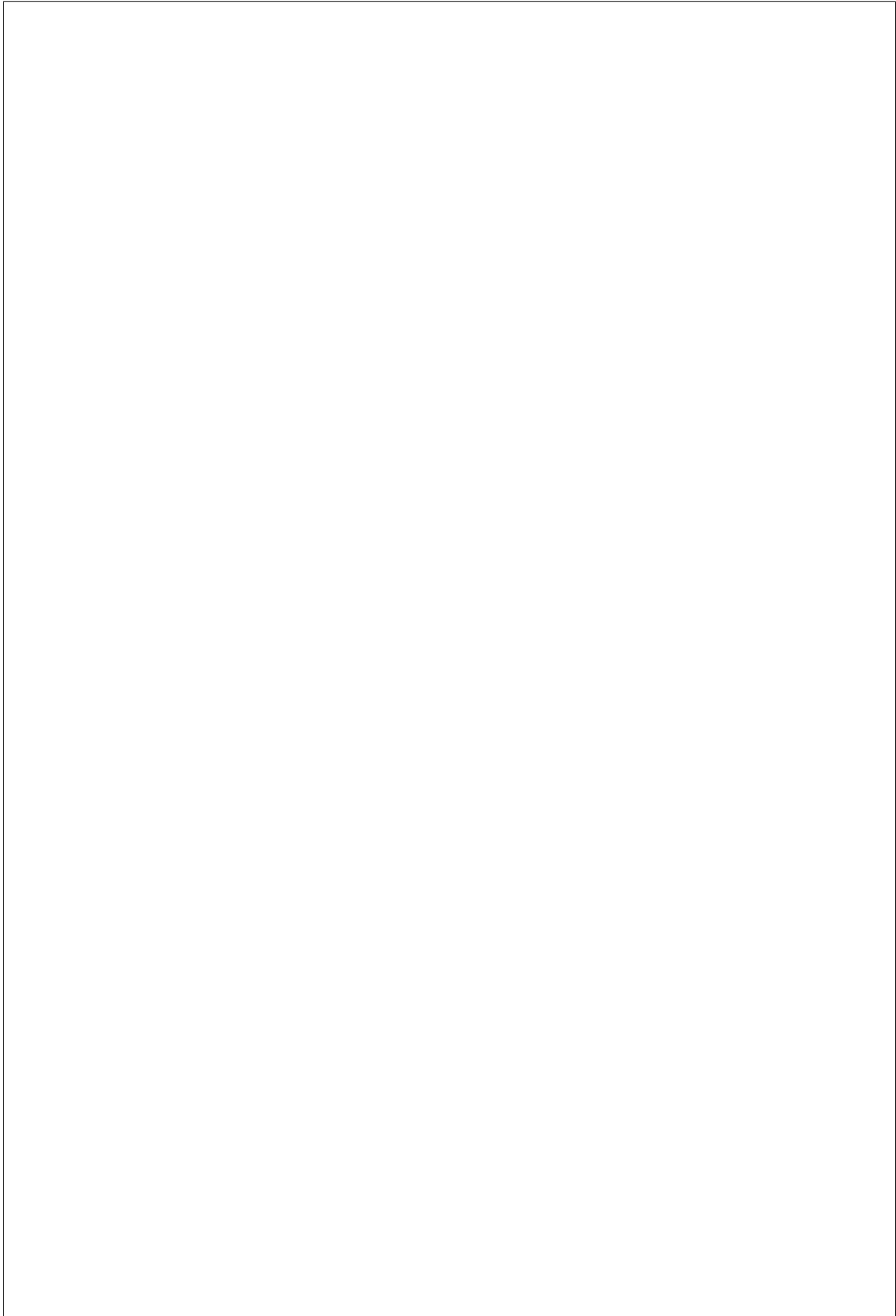
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



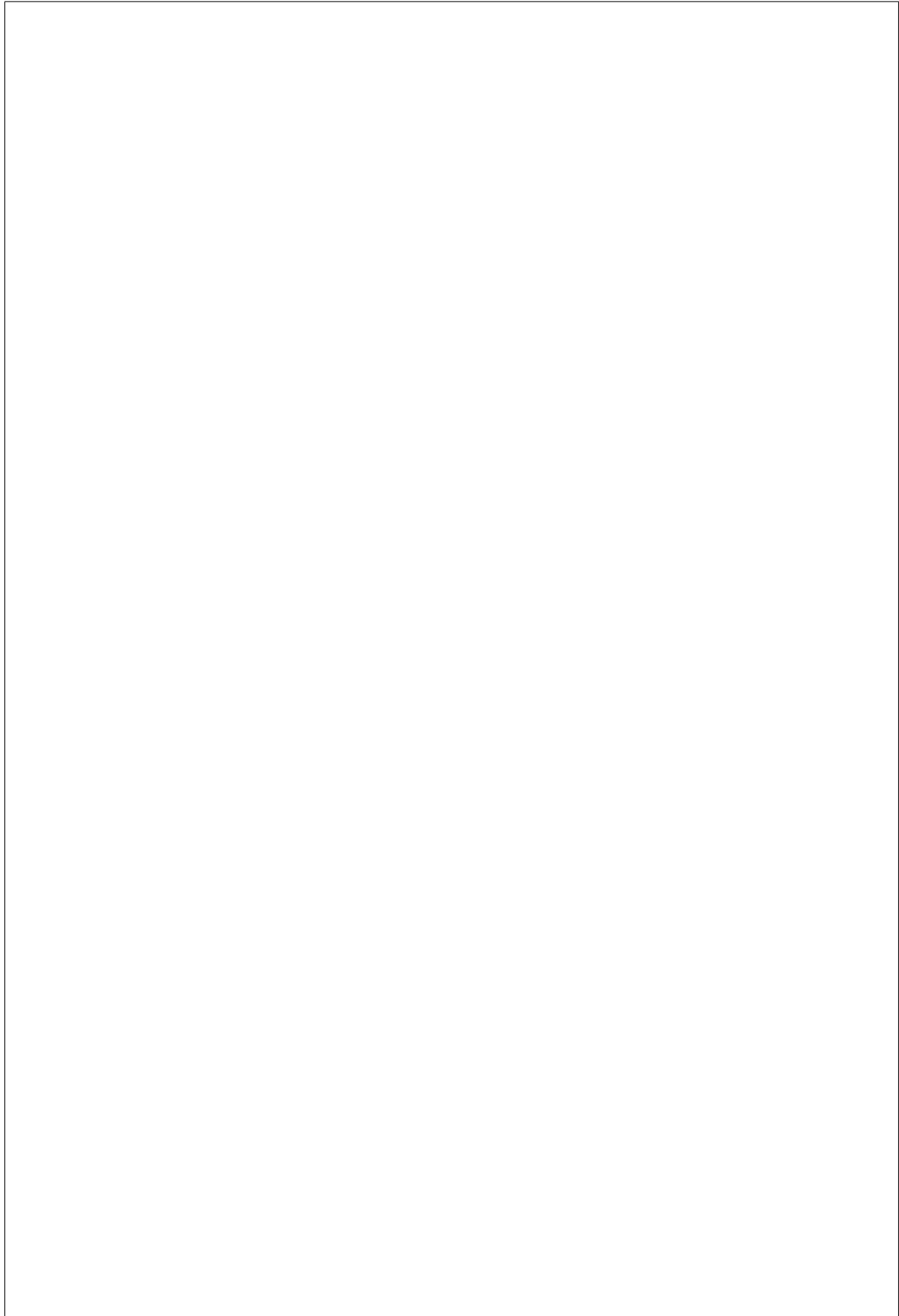
(continues on next page)

(continued from previous page)



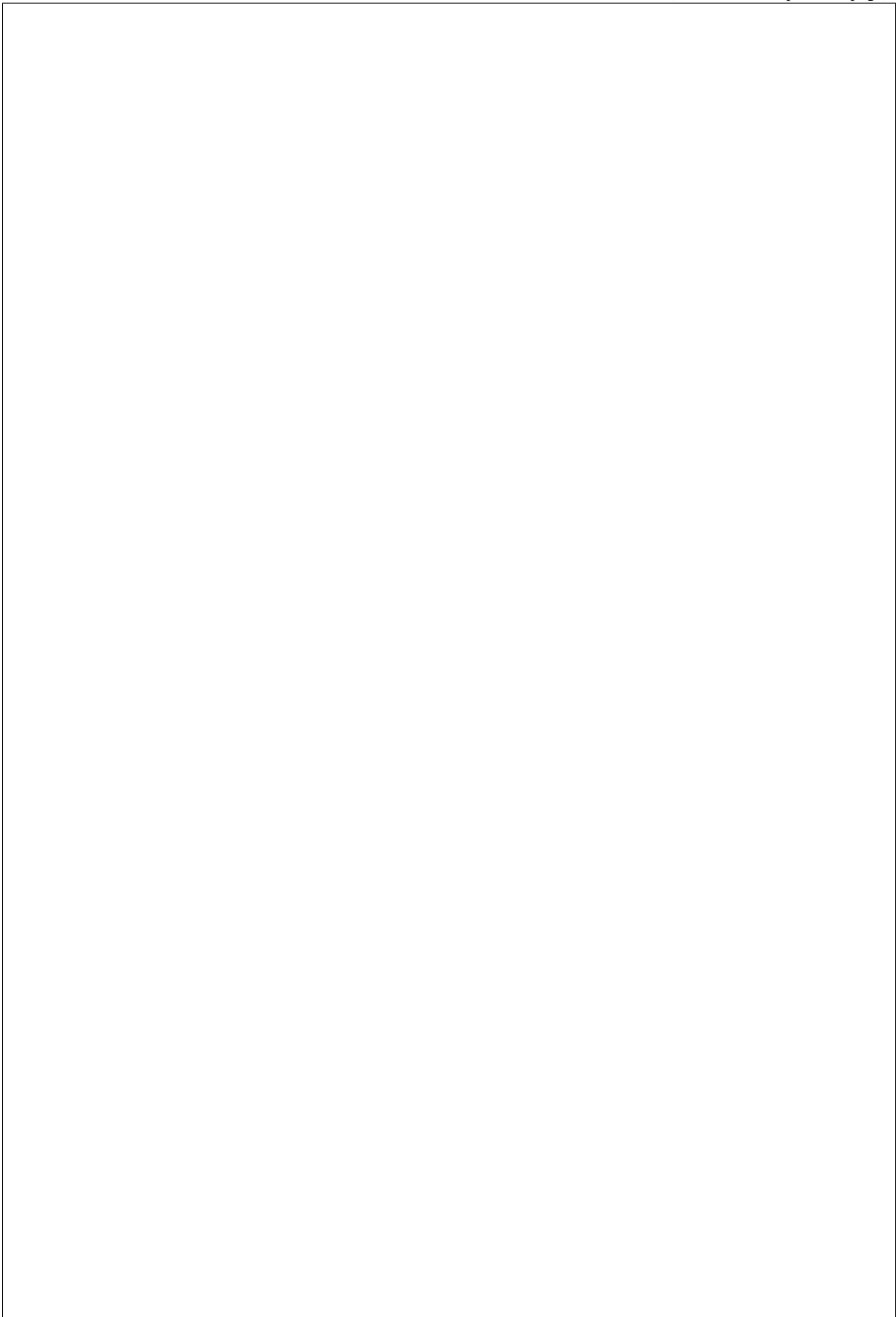
(continues on next page)

(continued from previous page)



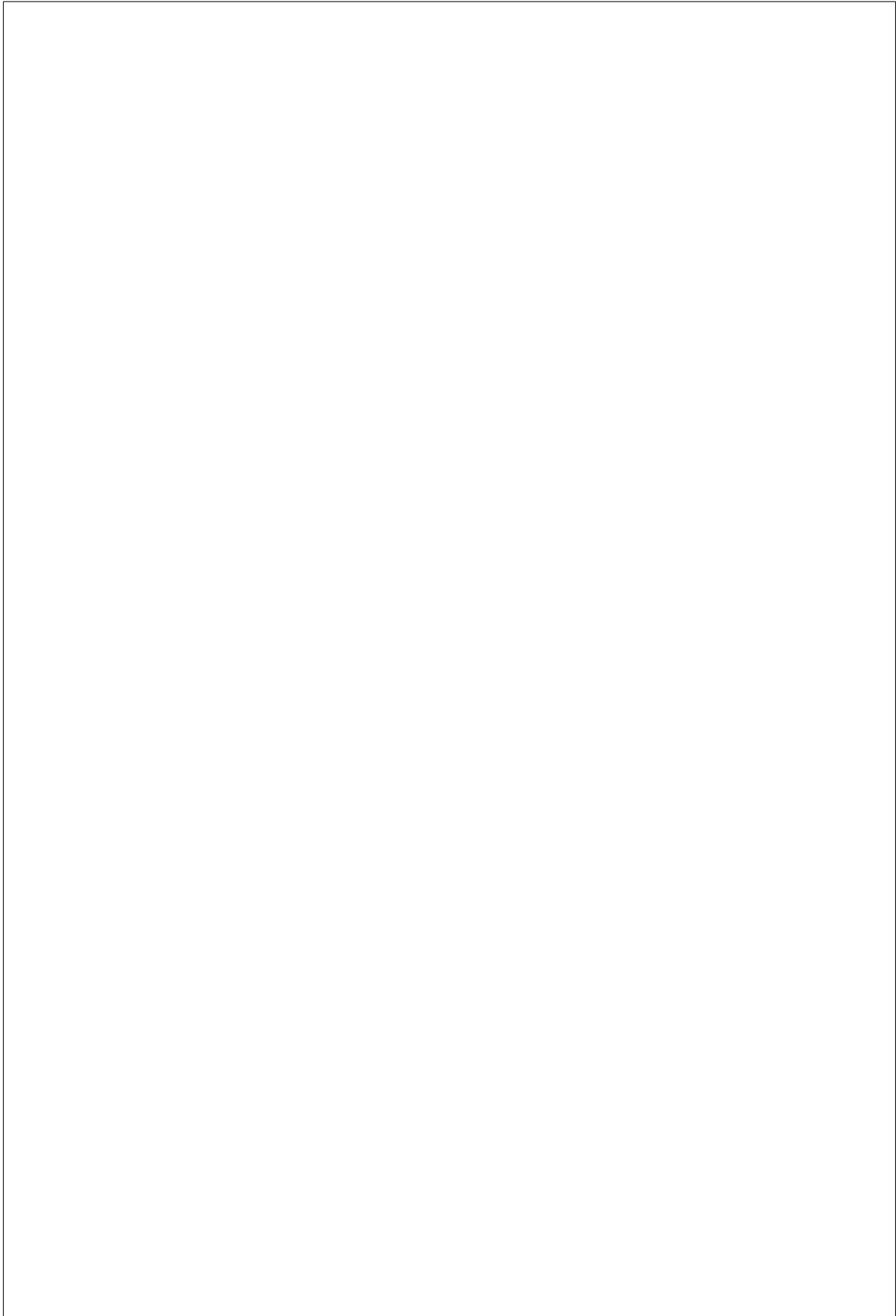
(continues on next page)

(continued from previous page)



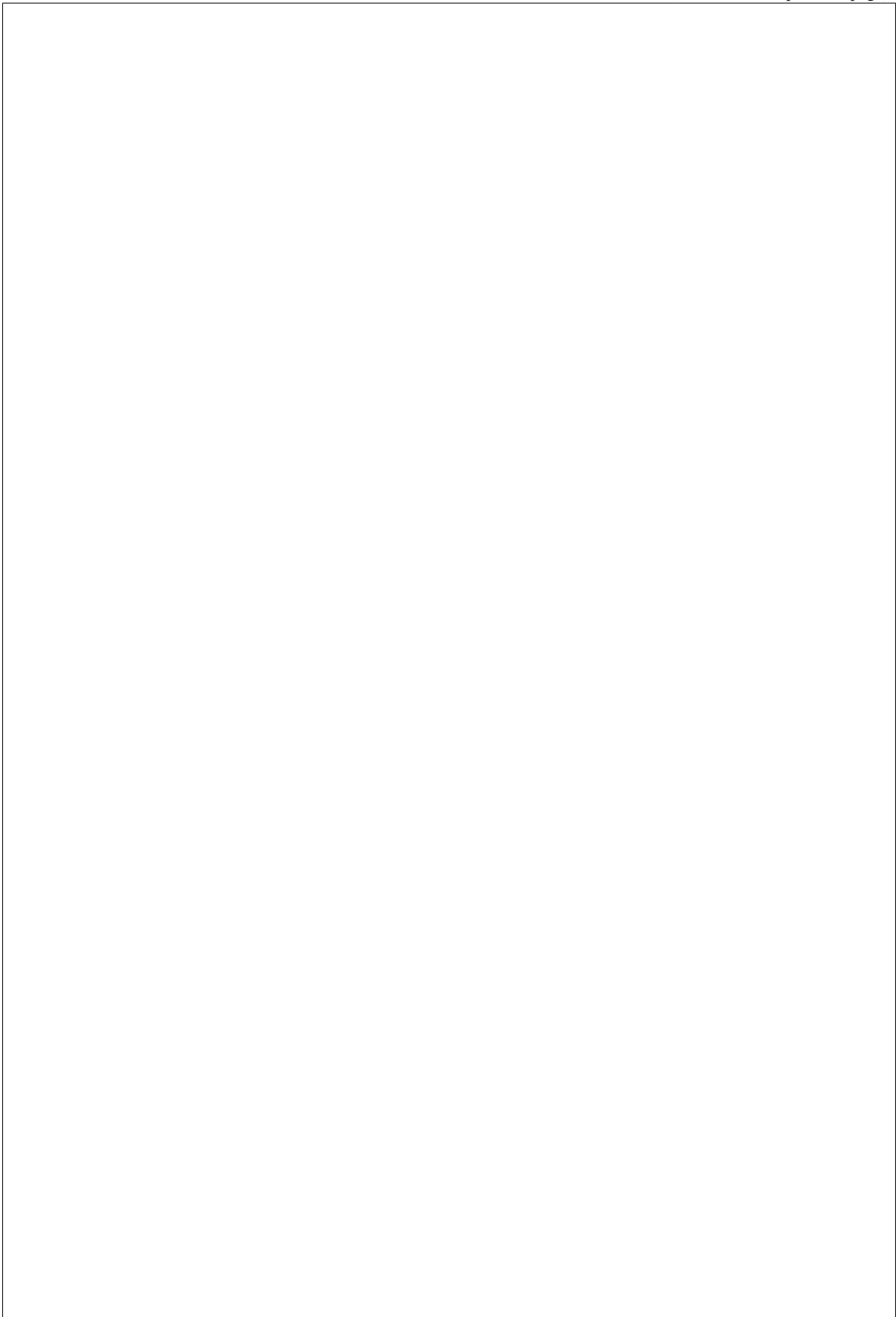
(continues on next page)

(continued from previous page)



(continues on next page)

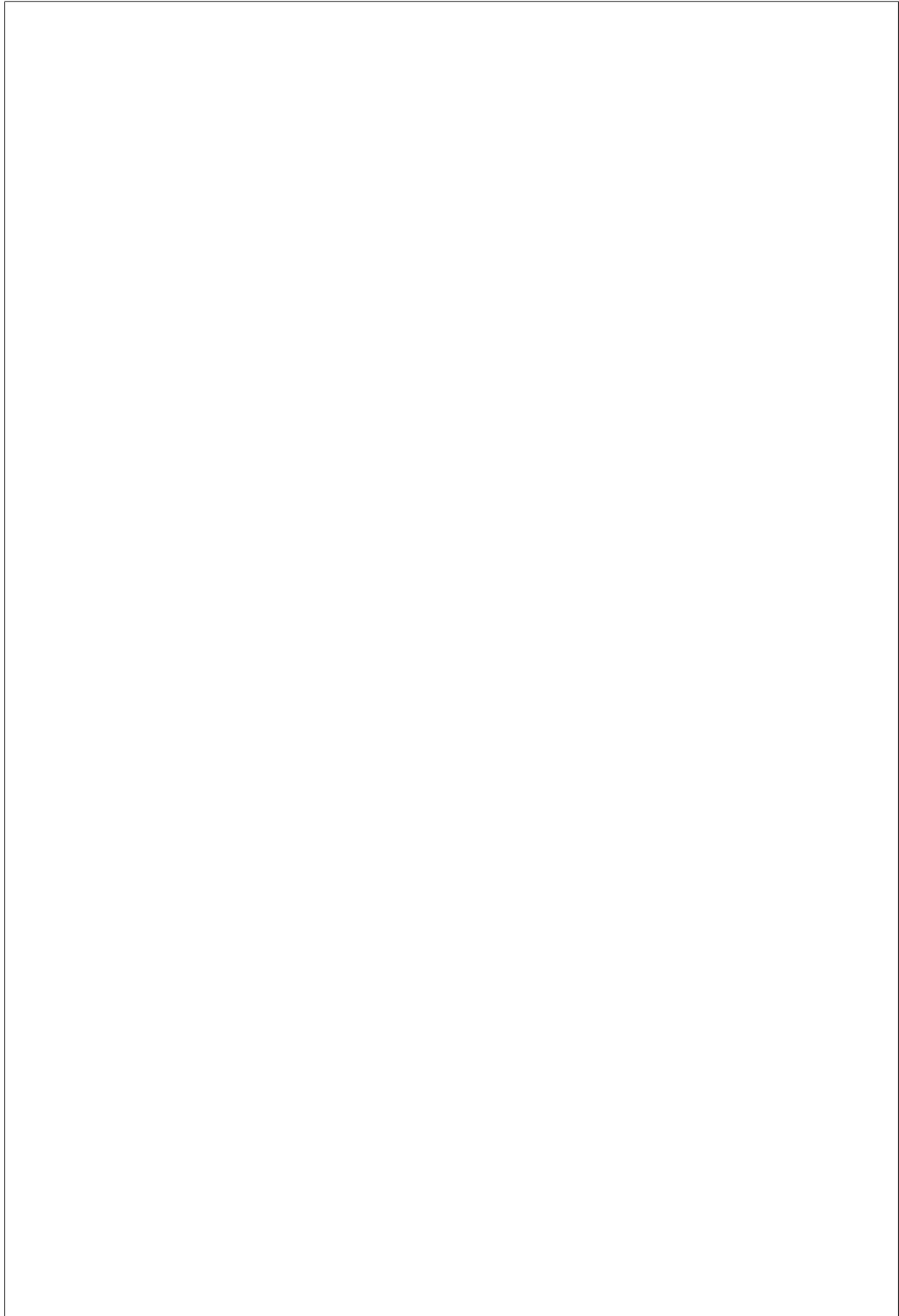
(continued from previous page)



(continues on next page)



(continued from previous page)



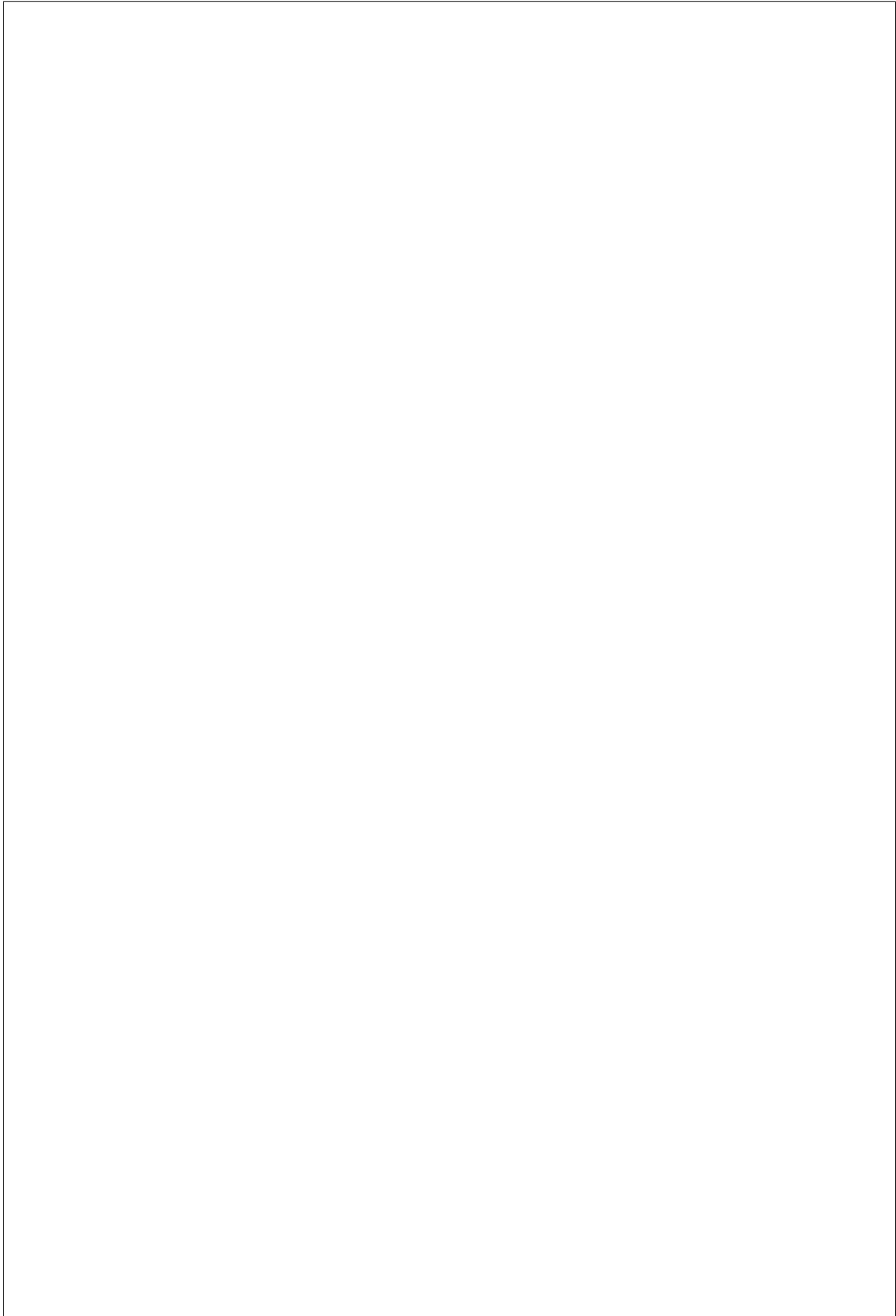
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



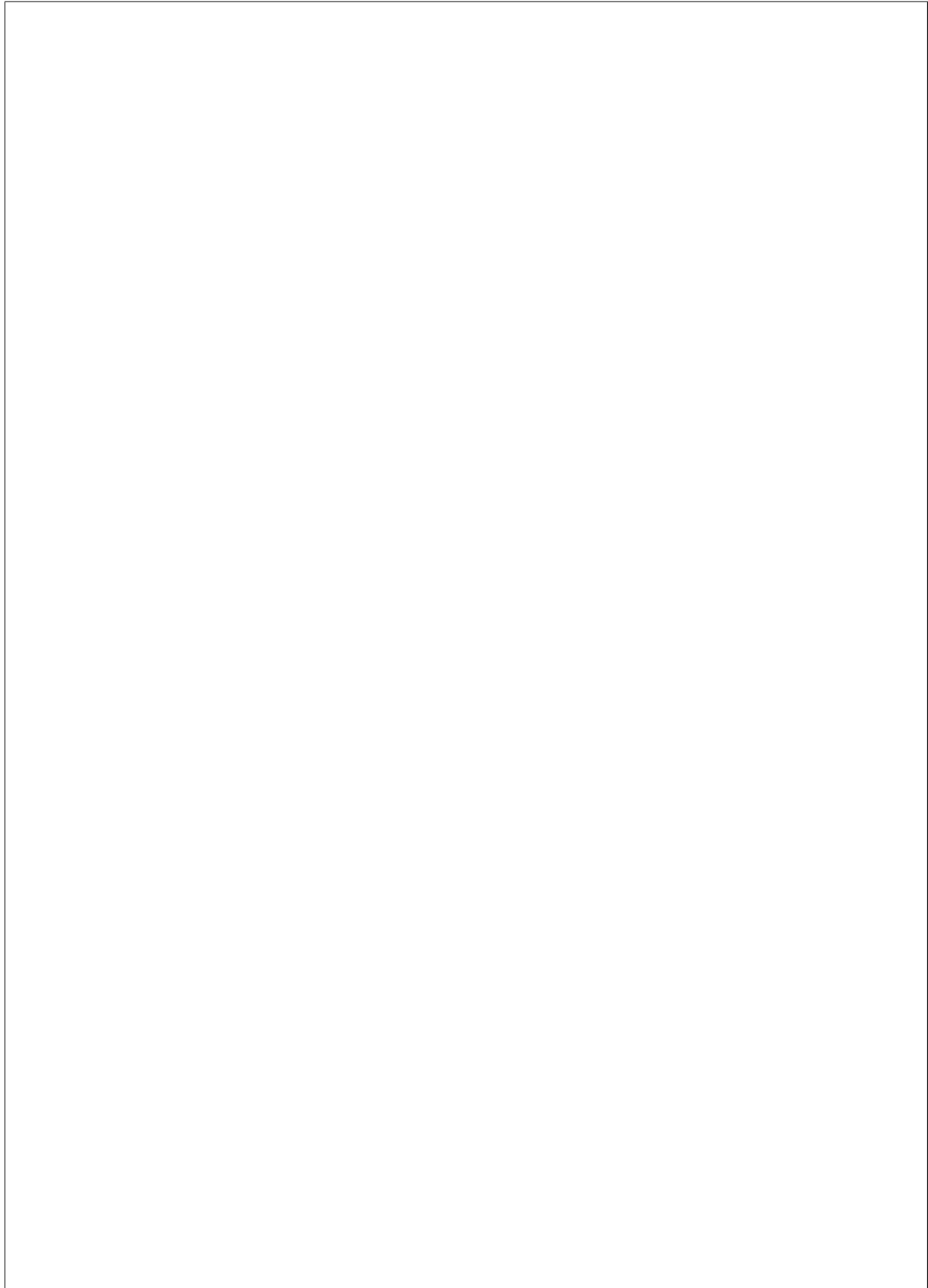
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

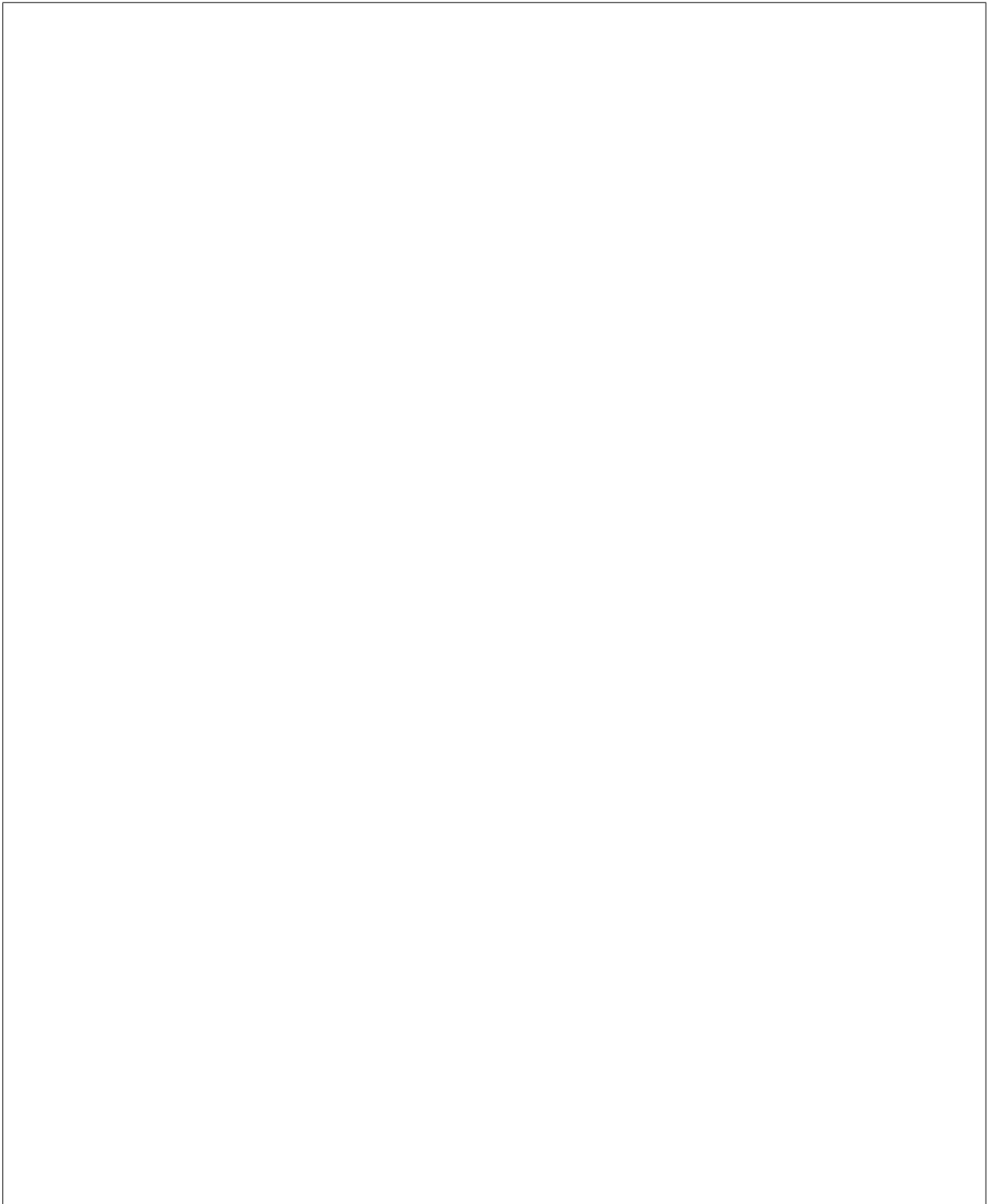


`baremetal.node.power_set`



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





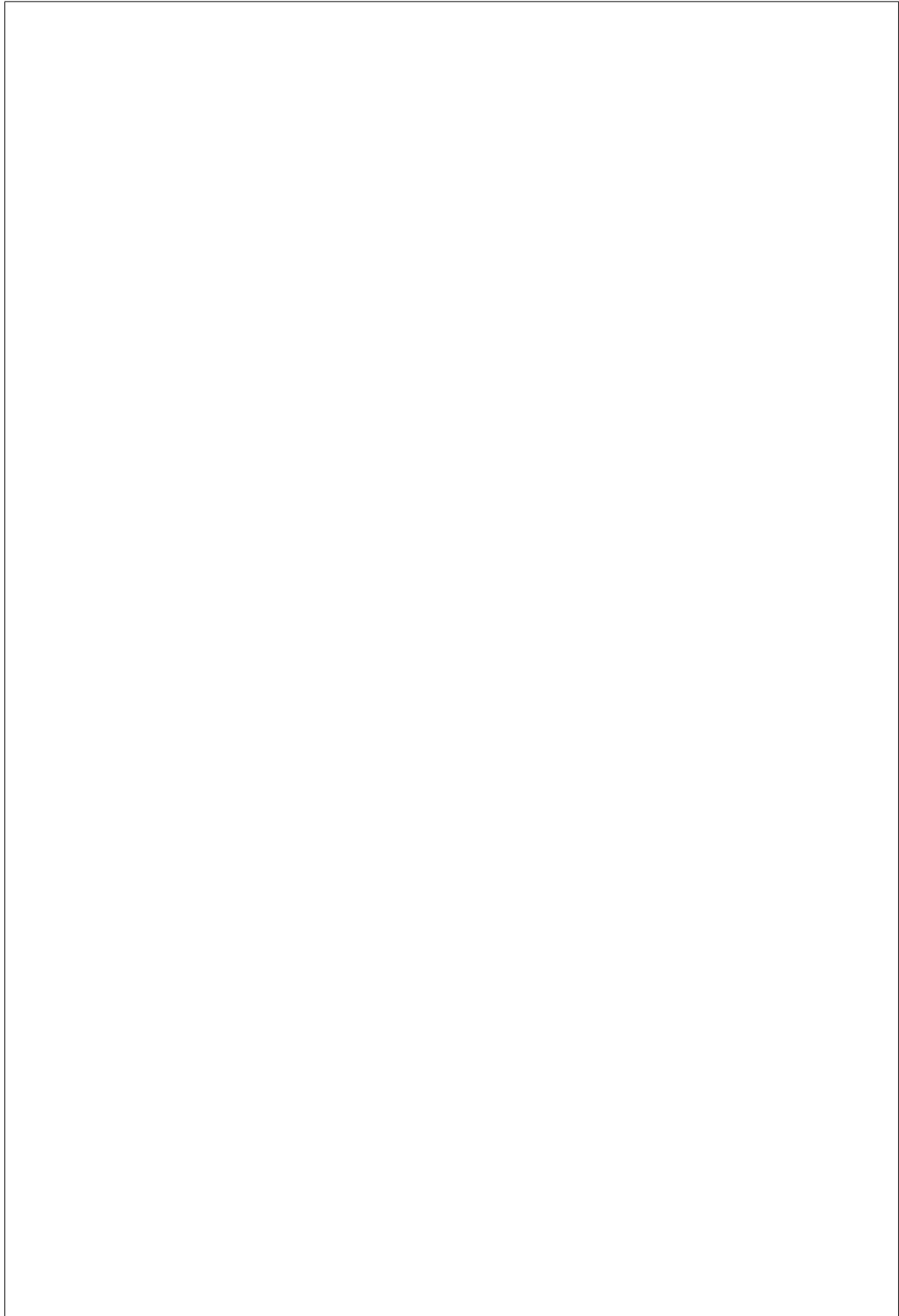
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



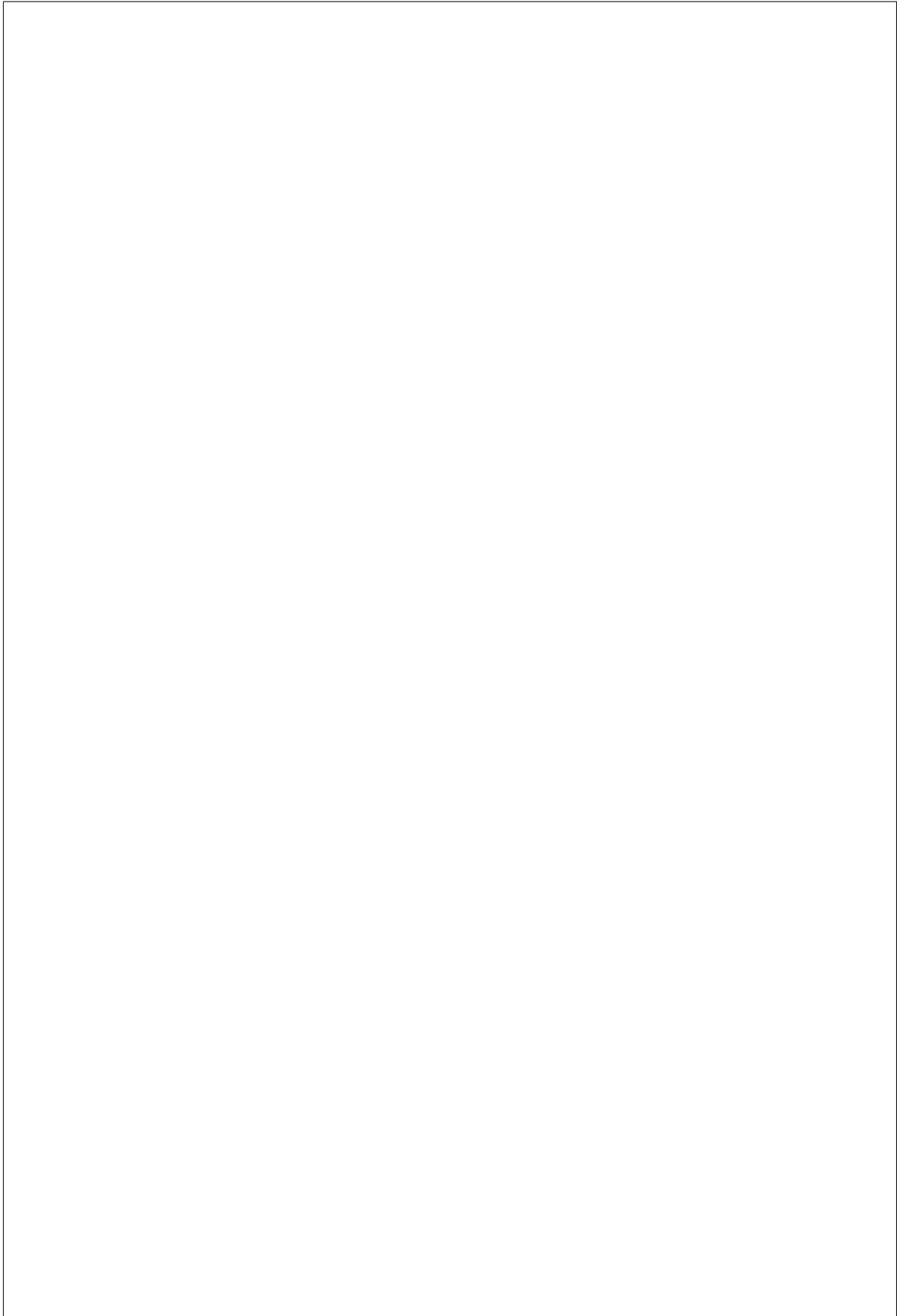
(continues on next page)

(continued from previous page)



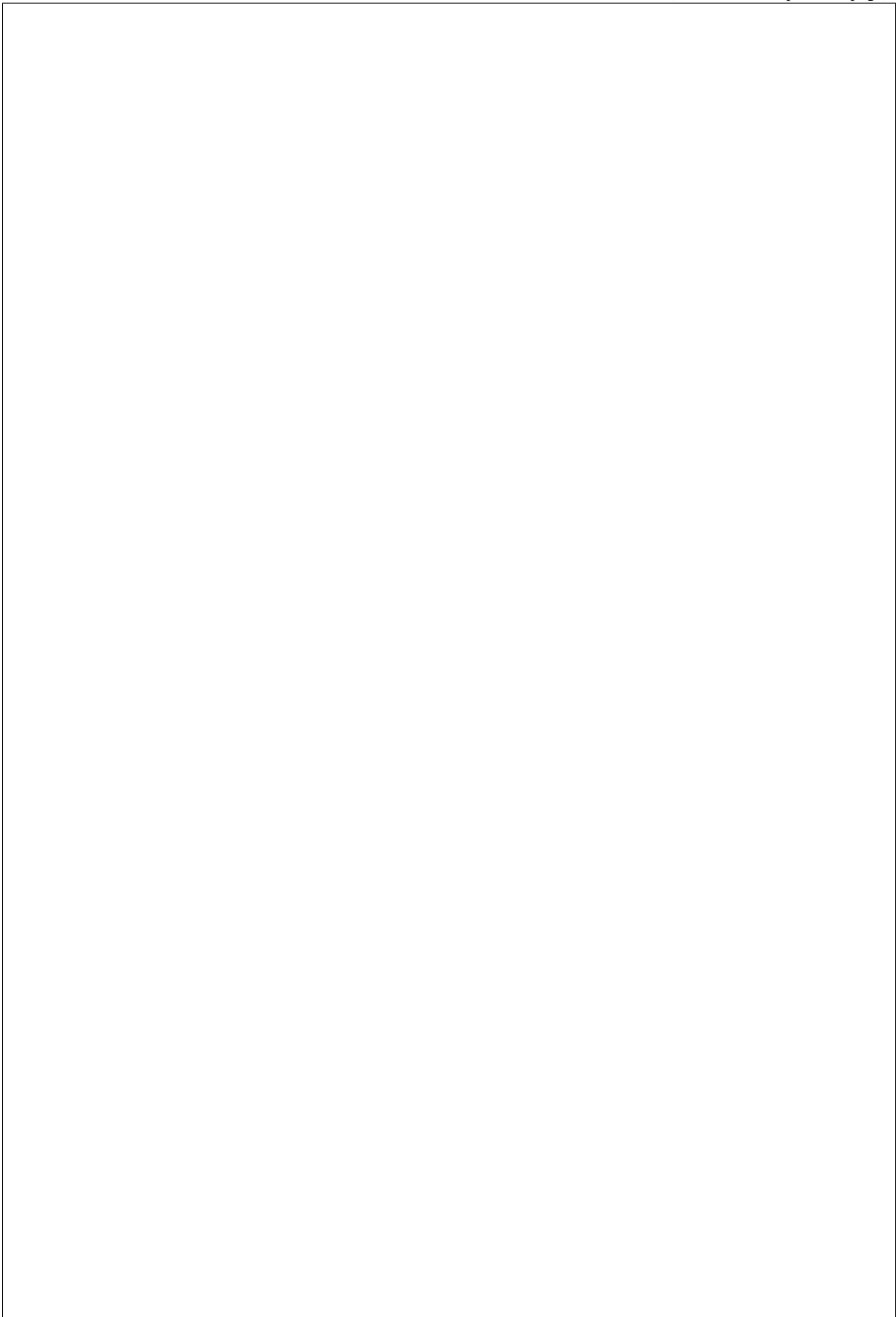
(continues on next page)

(continued from previous page)



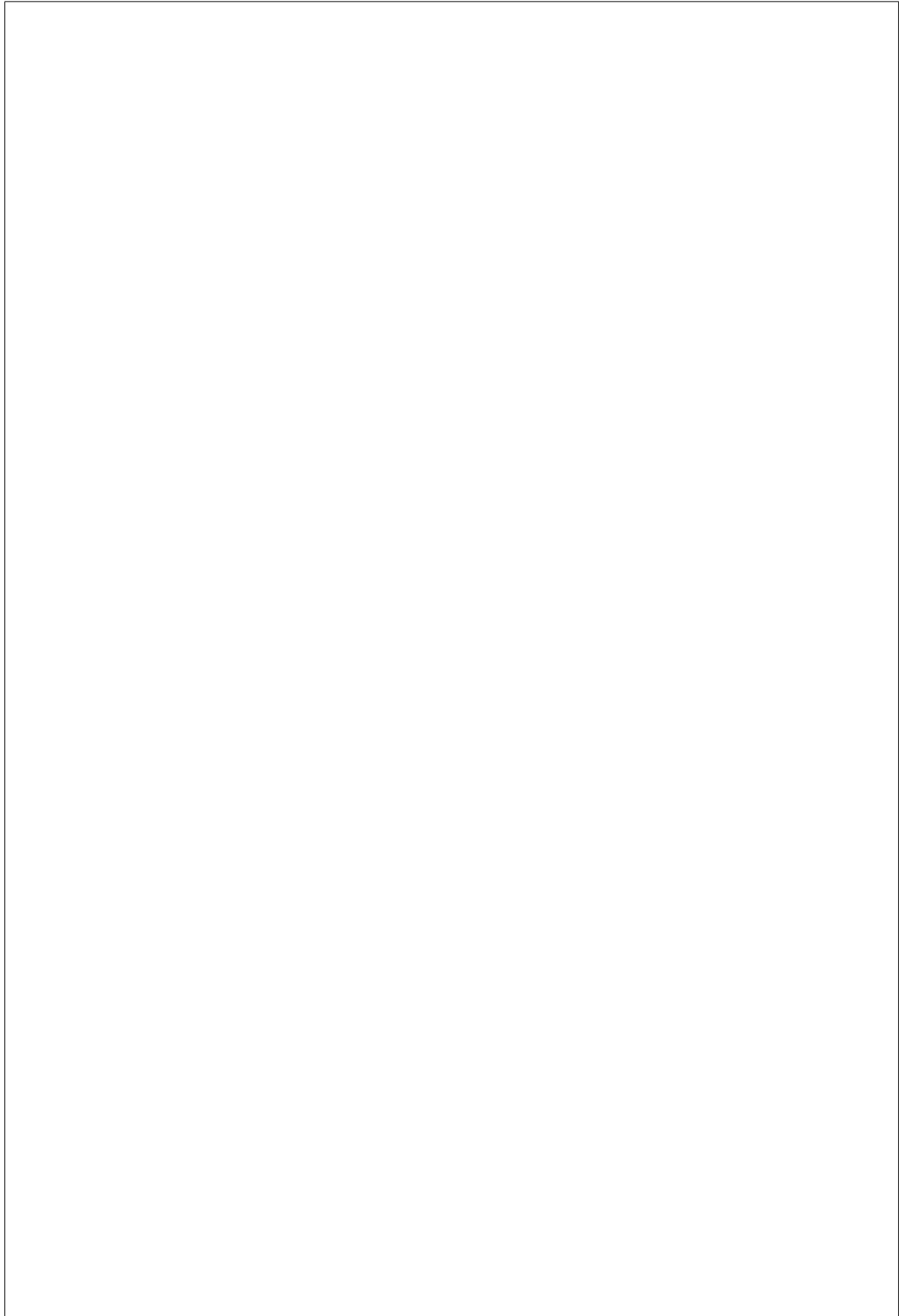
(continues on next page)

(continued from previous page)



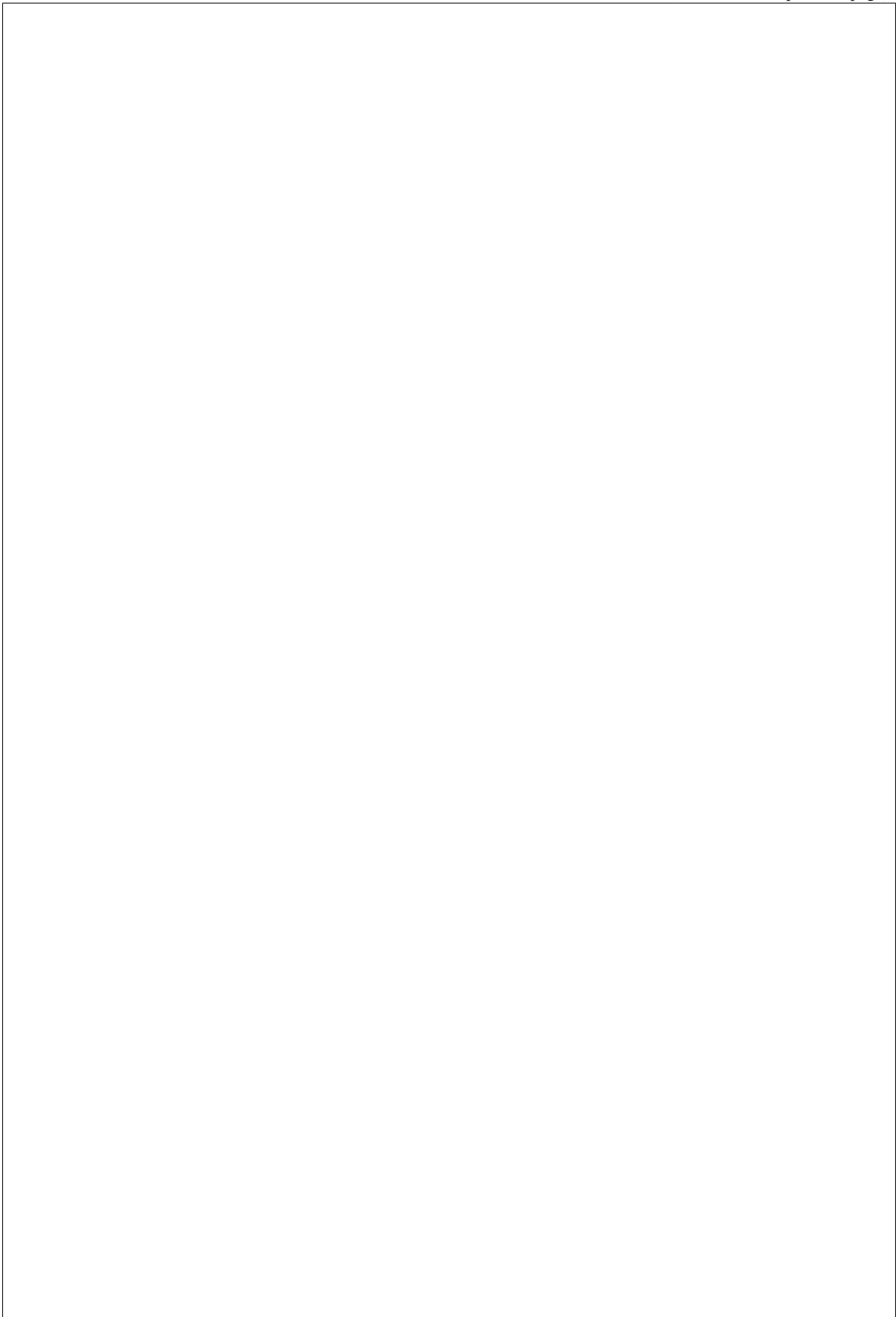
(continues on next page)

(continued from previous page)



(continues on next page)

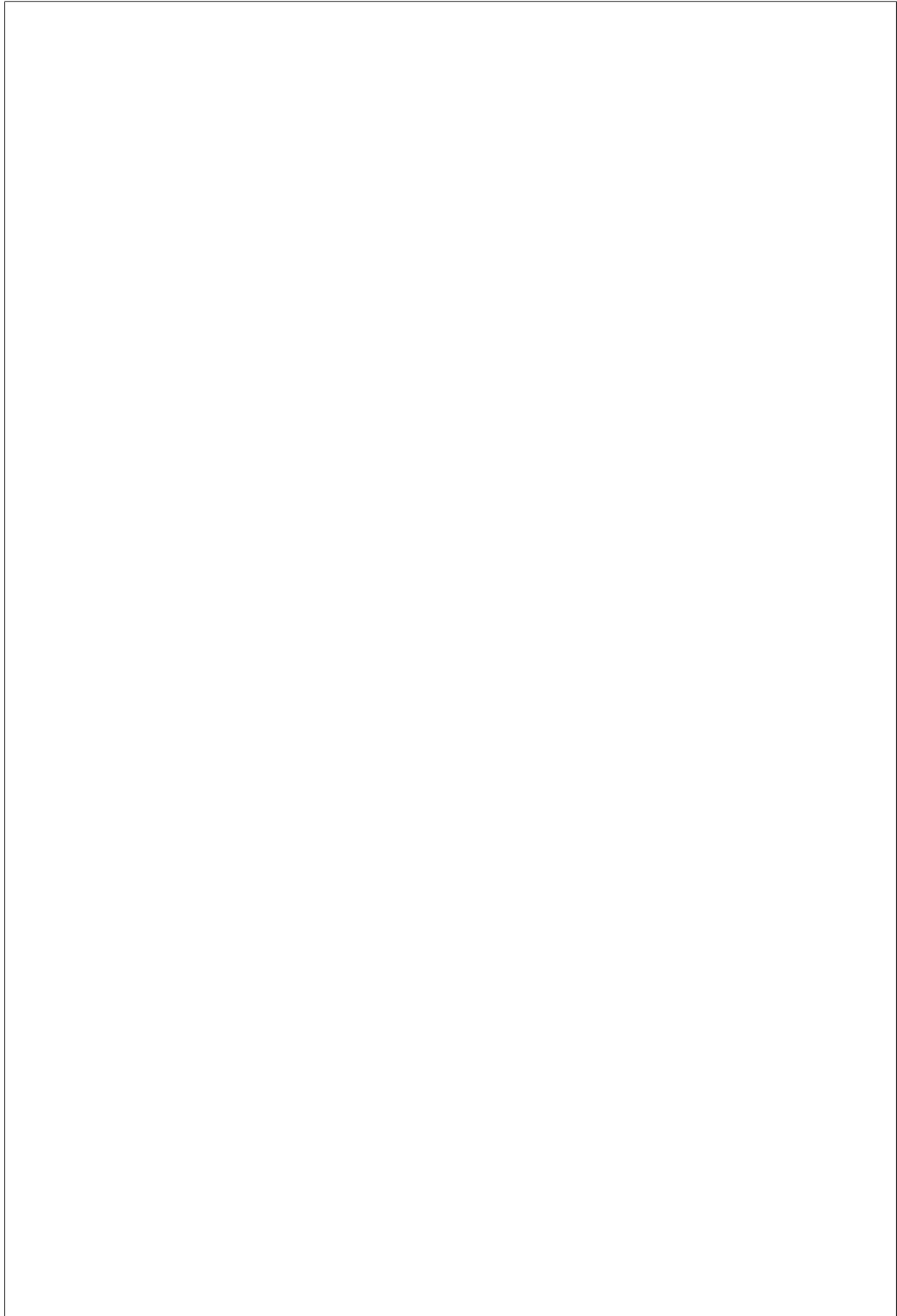
(continued from previous page)



(continues on next page)



(continued from previous page)



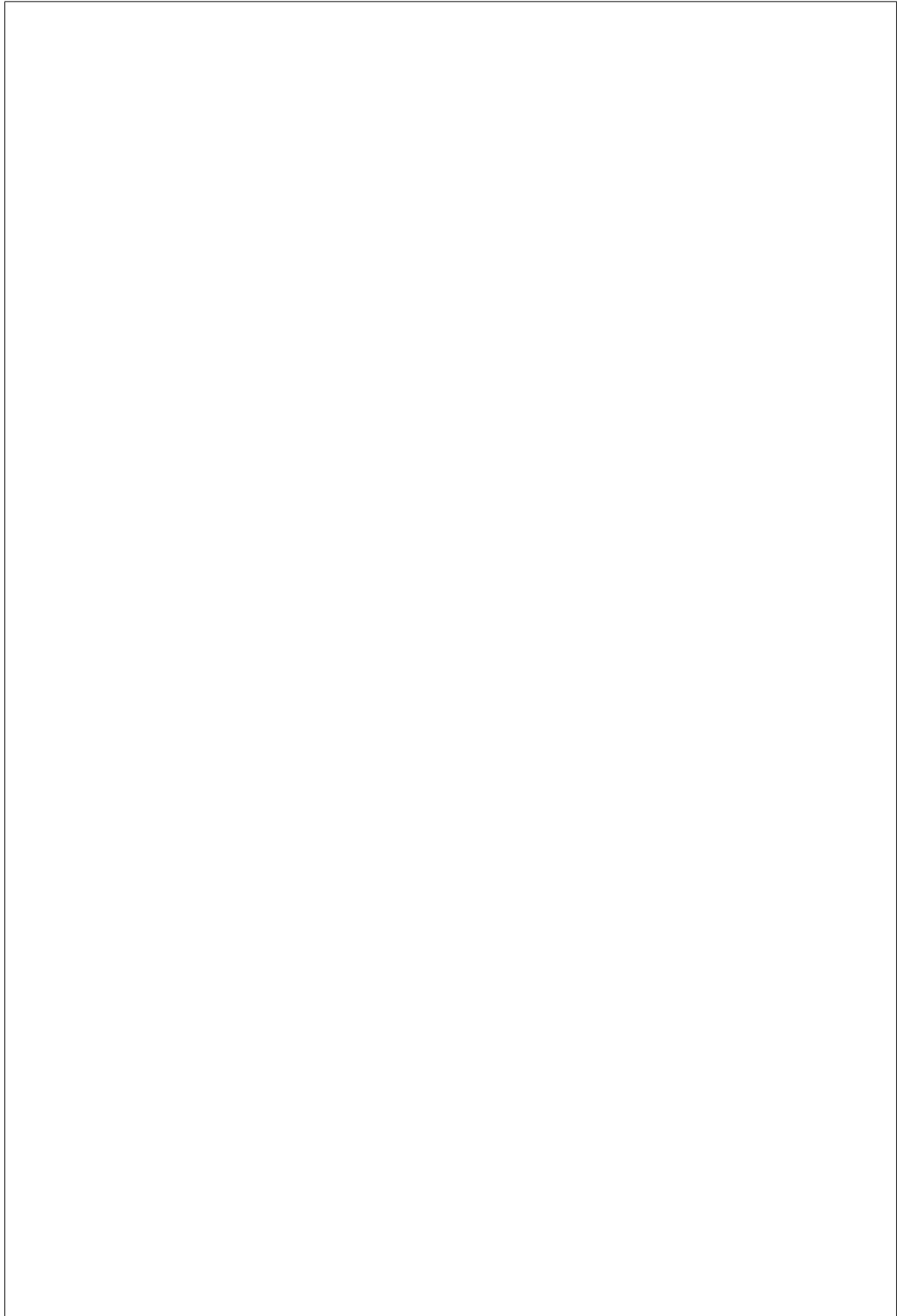
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



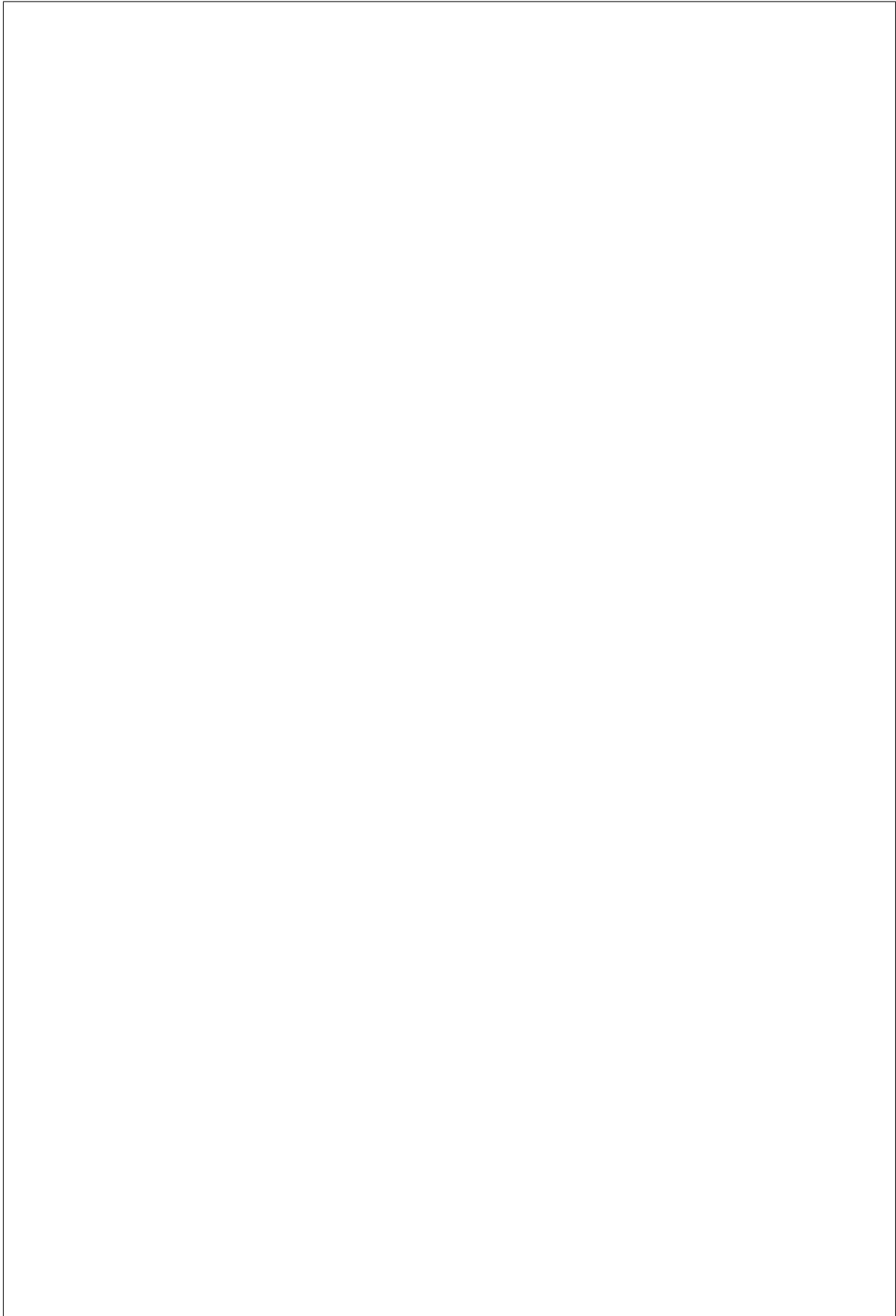
(continues on next page)

(continued from previous page)



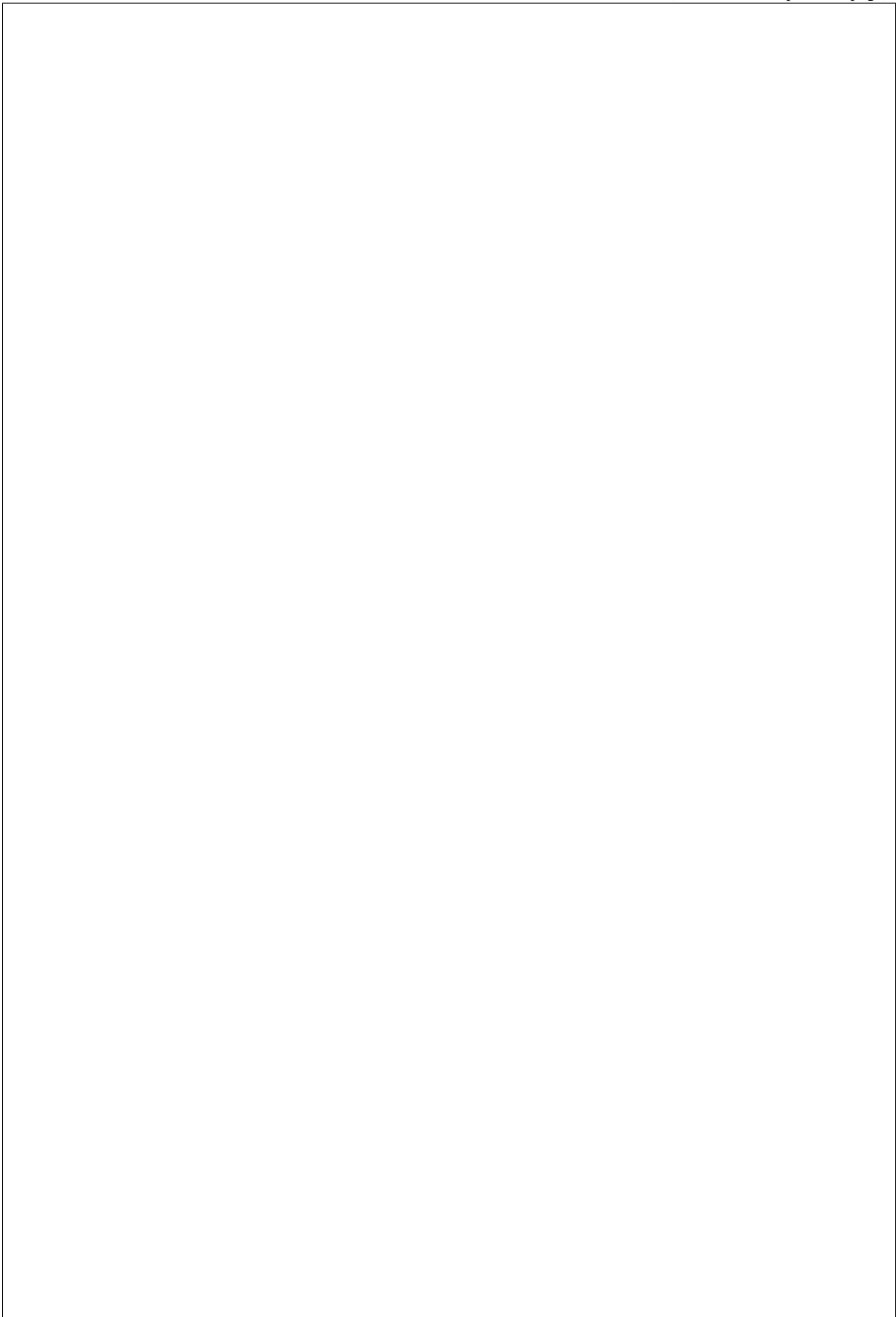
(continues on next page)

(continued from previous page)



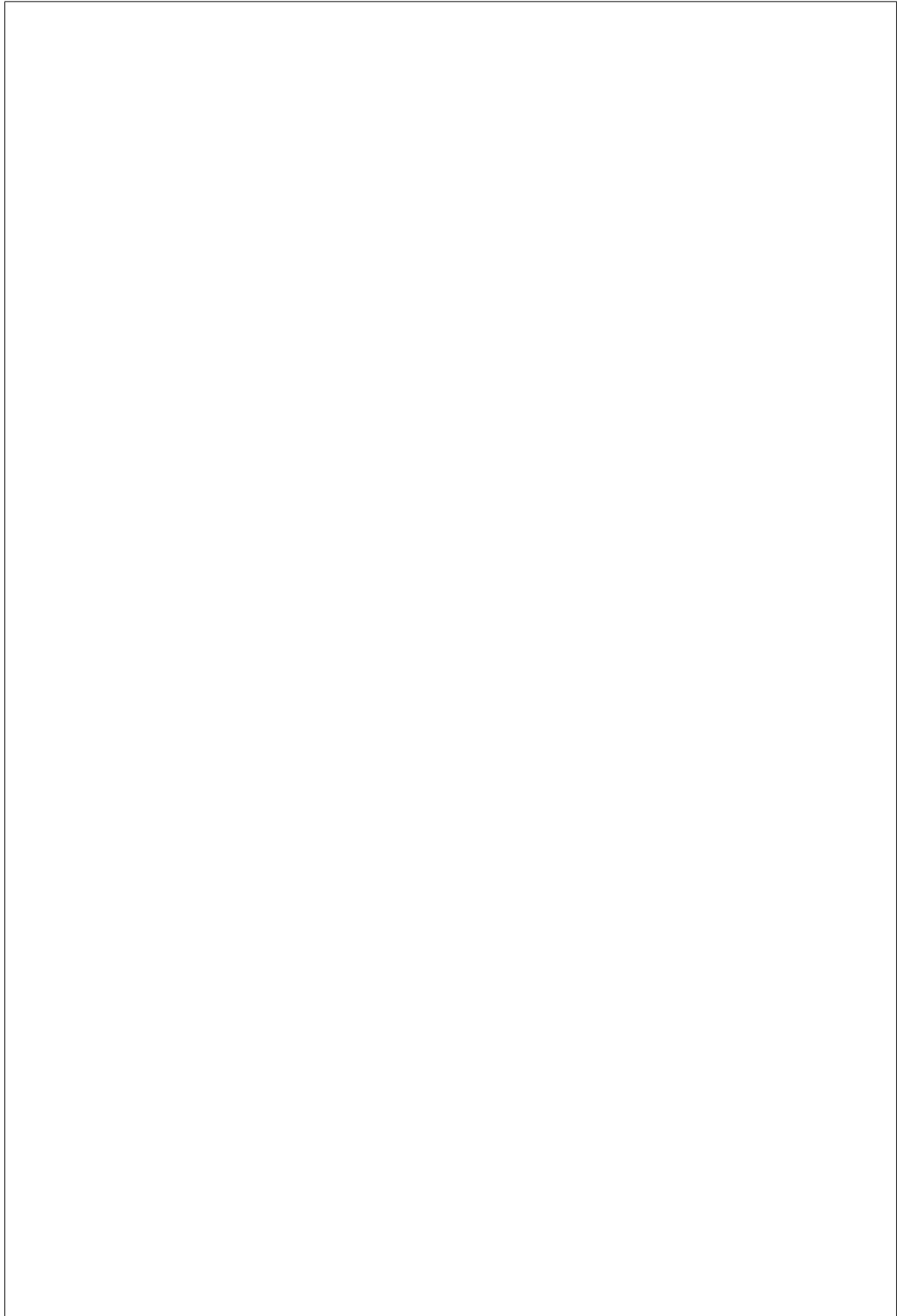
(continues on next page)

(continued from previous page)



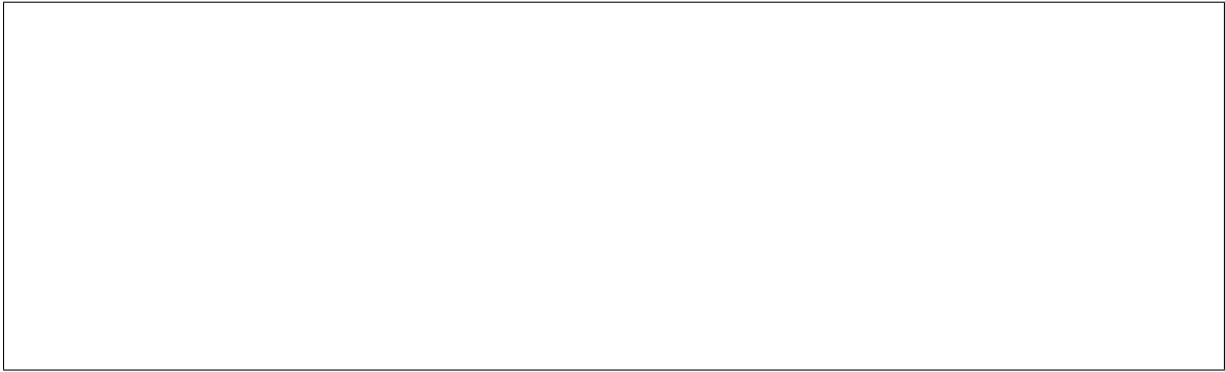
(continues on next page)

(continued from previous page)



(continues on next page)

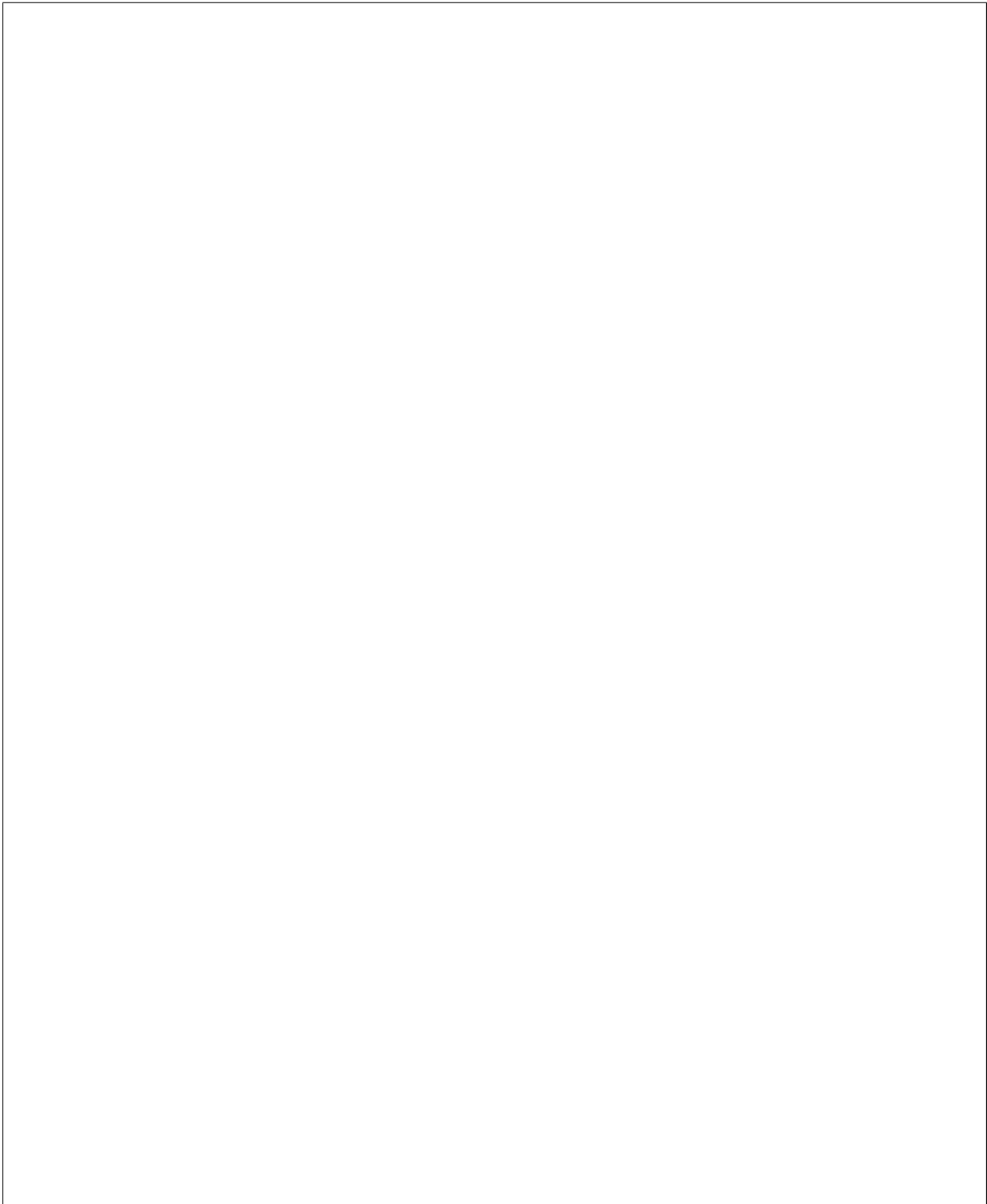
(continued from previous page)



**baremetal.node.power\_state\_corrected**

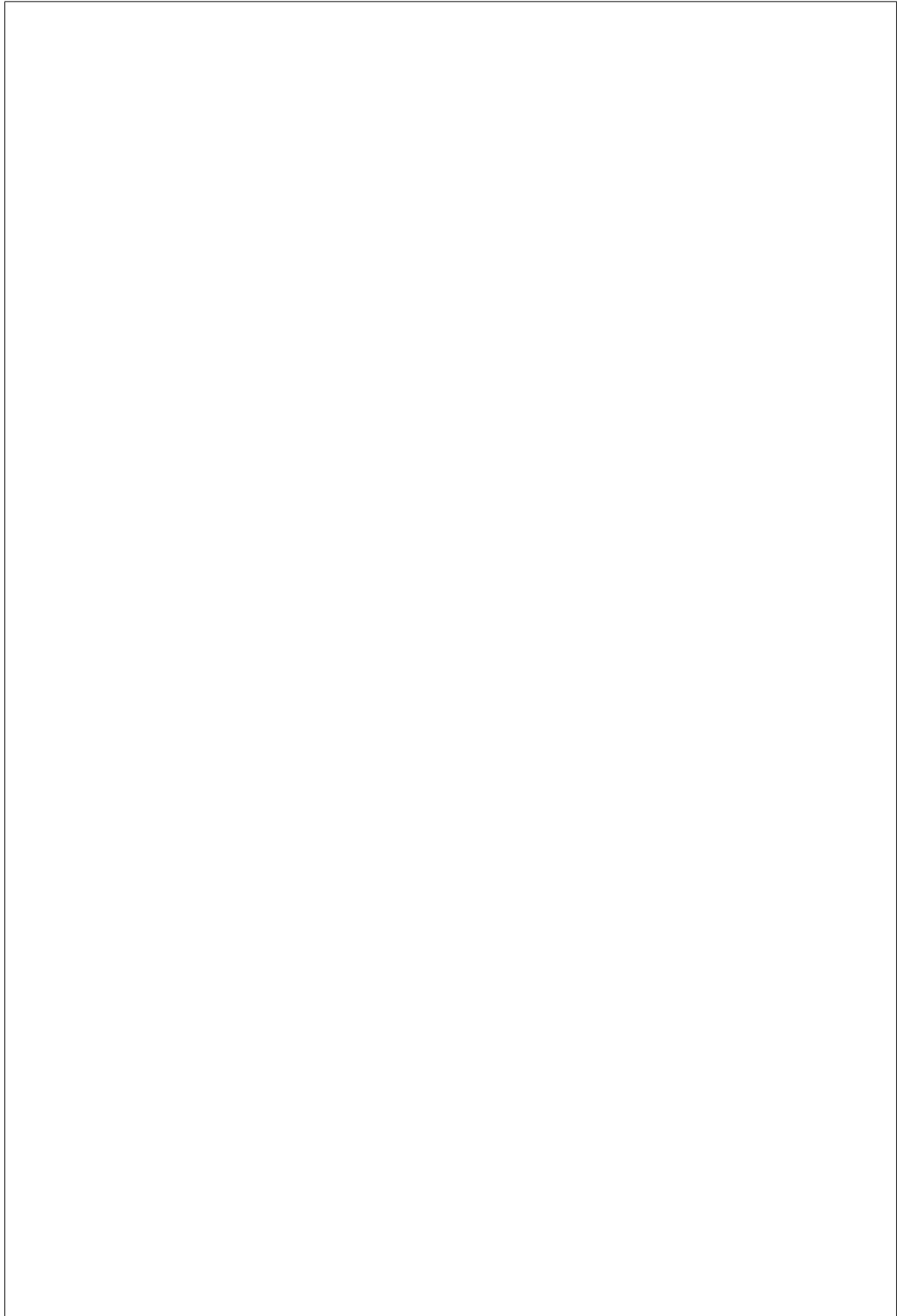


tion level info.



(continues on next page)

(continued from previous page)



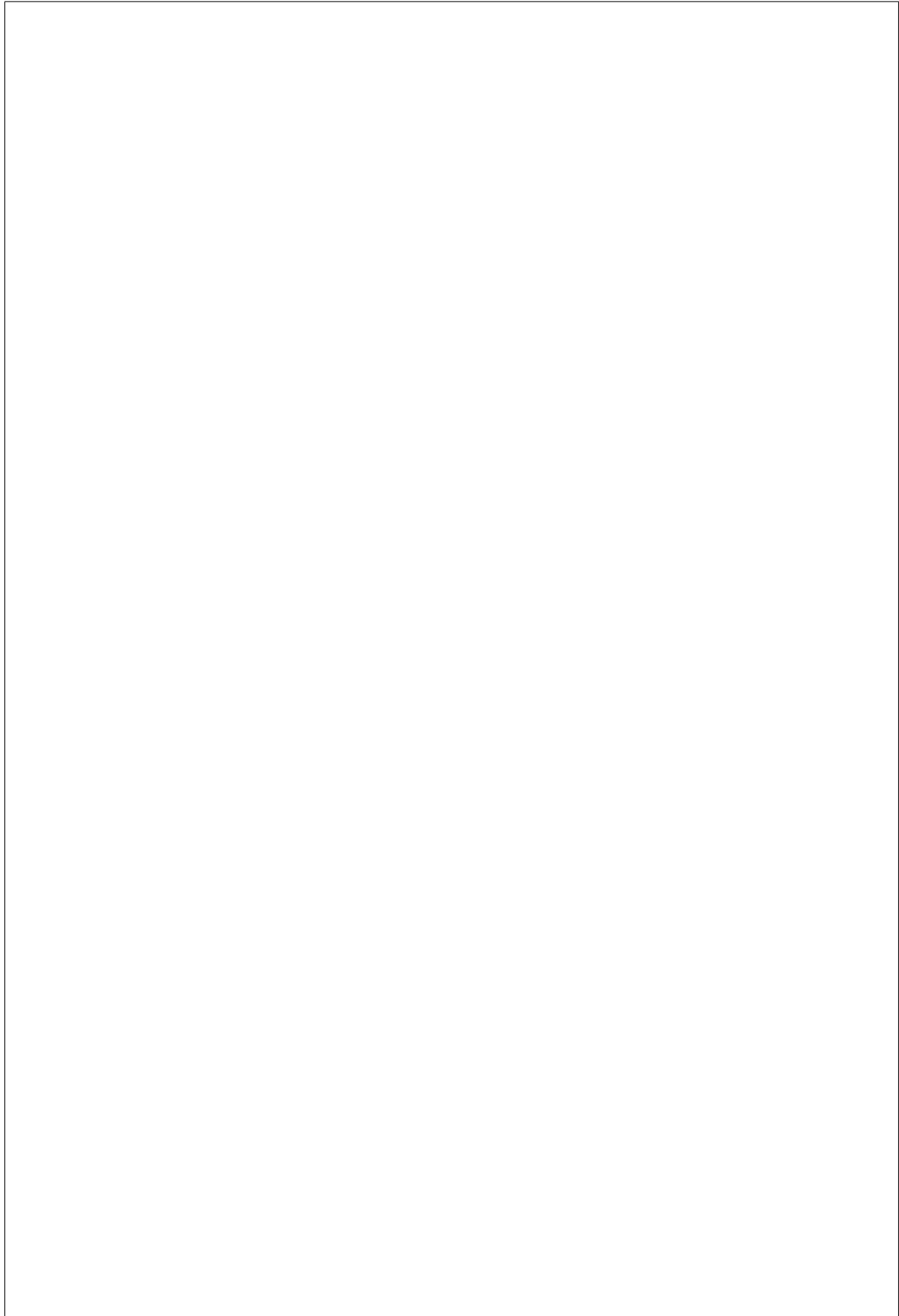
(continues on next page)

(continued from previous page)



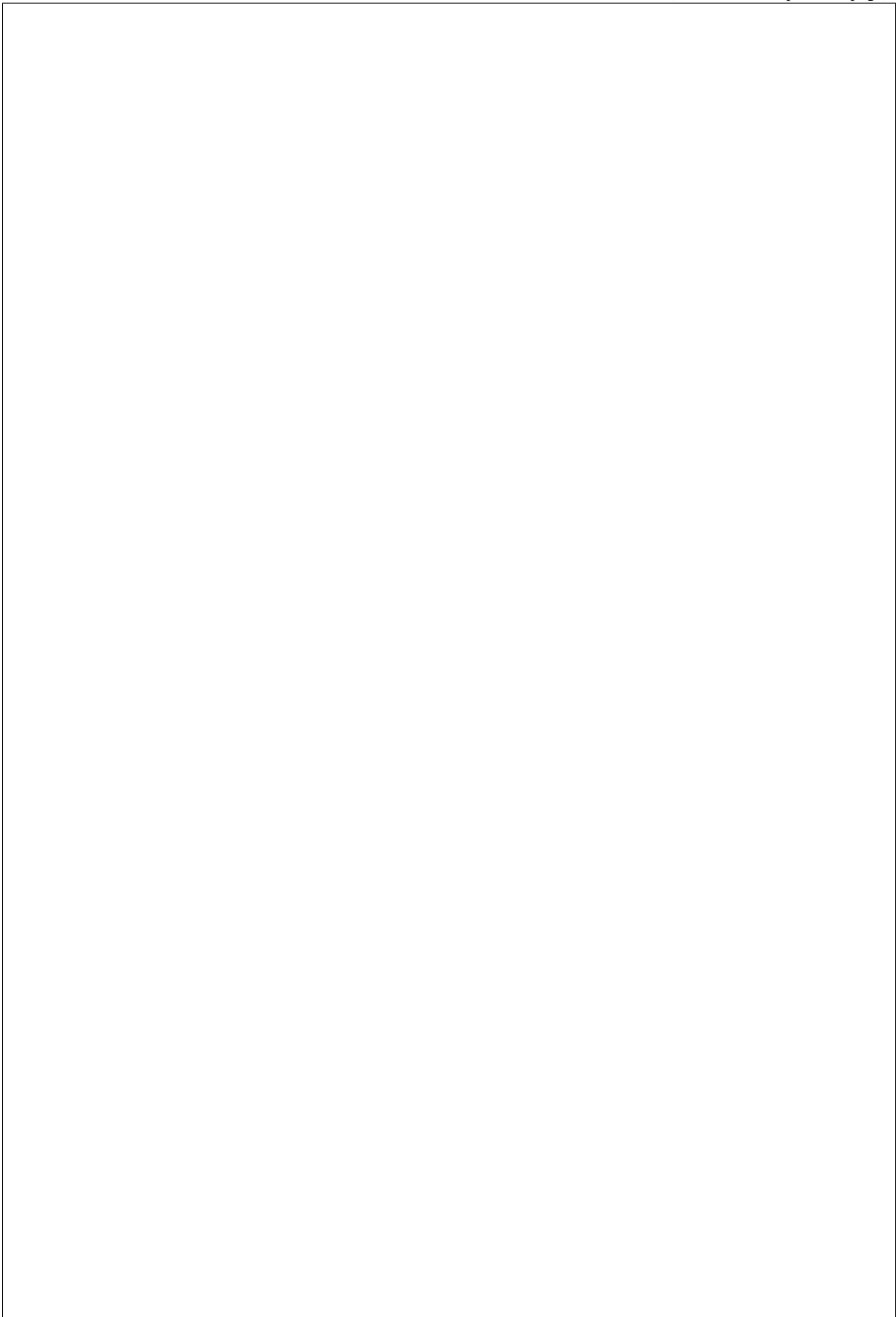
(continues on next page)

(continued from previous page)



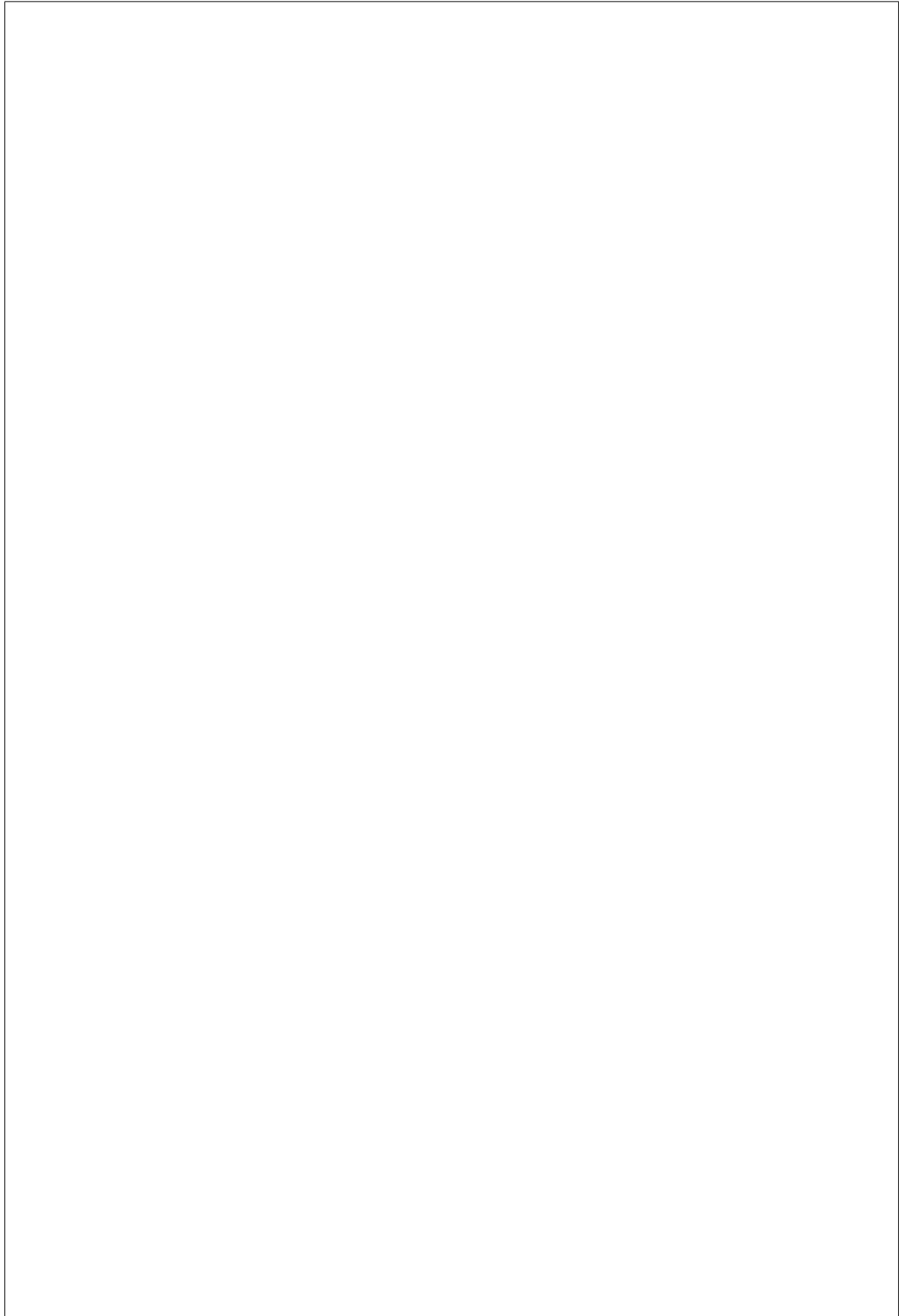
(continues on next page)

(continued from previous page)



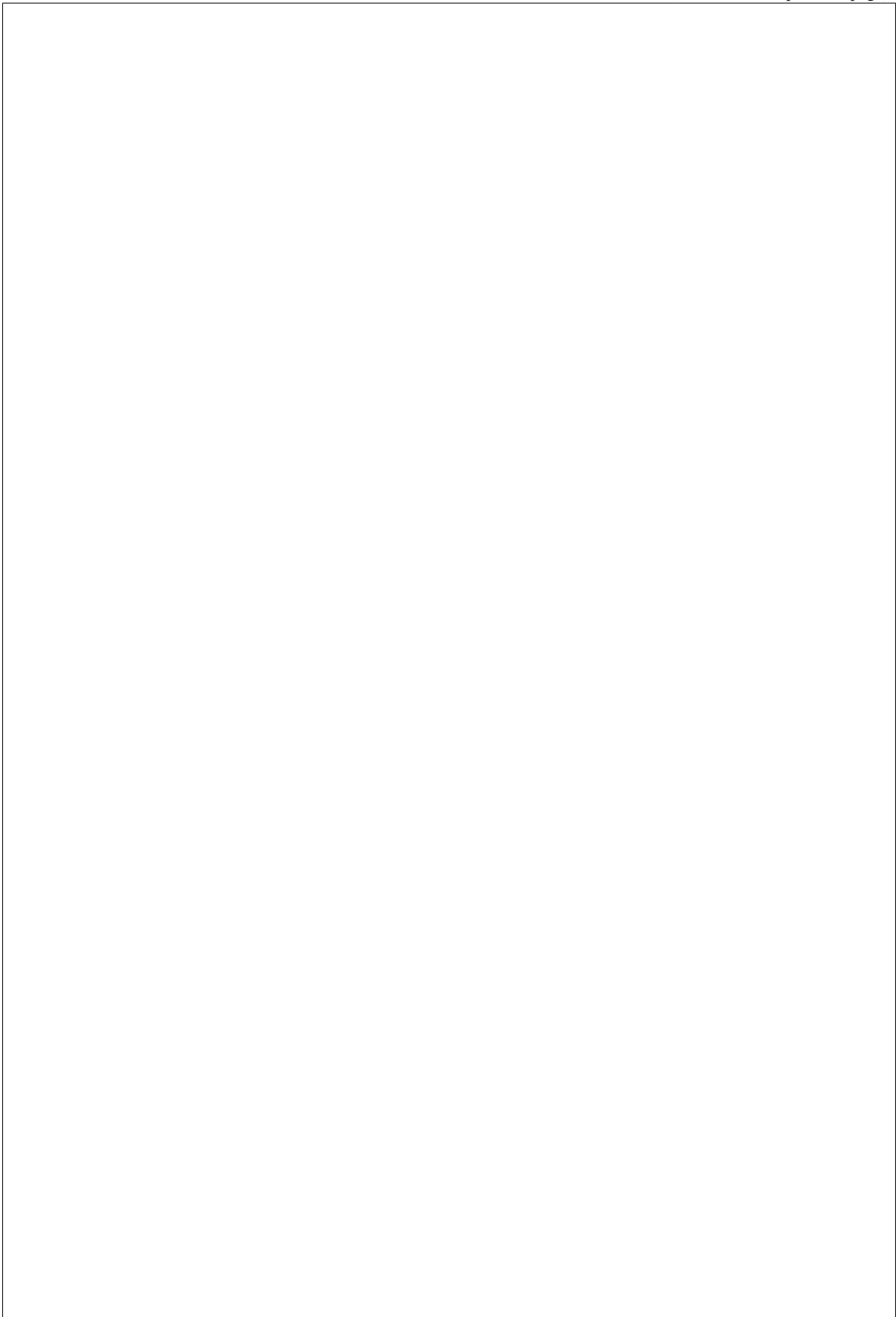
(continues on next page)

(continued from previous page)



(continues on next page)

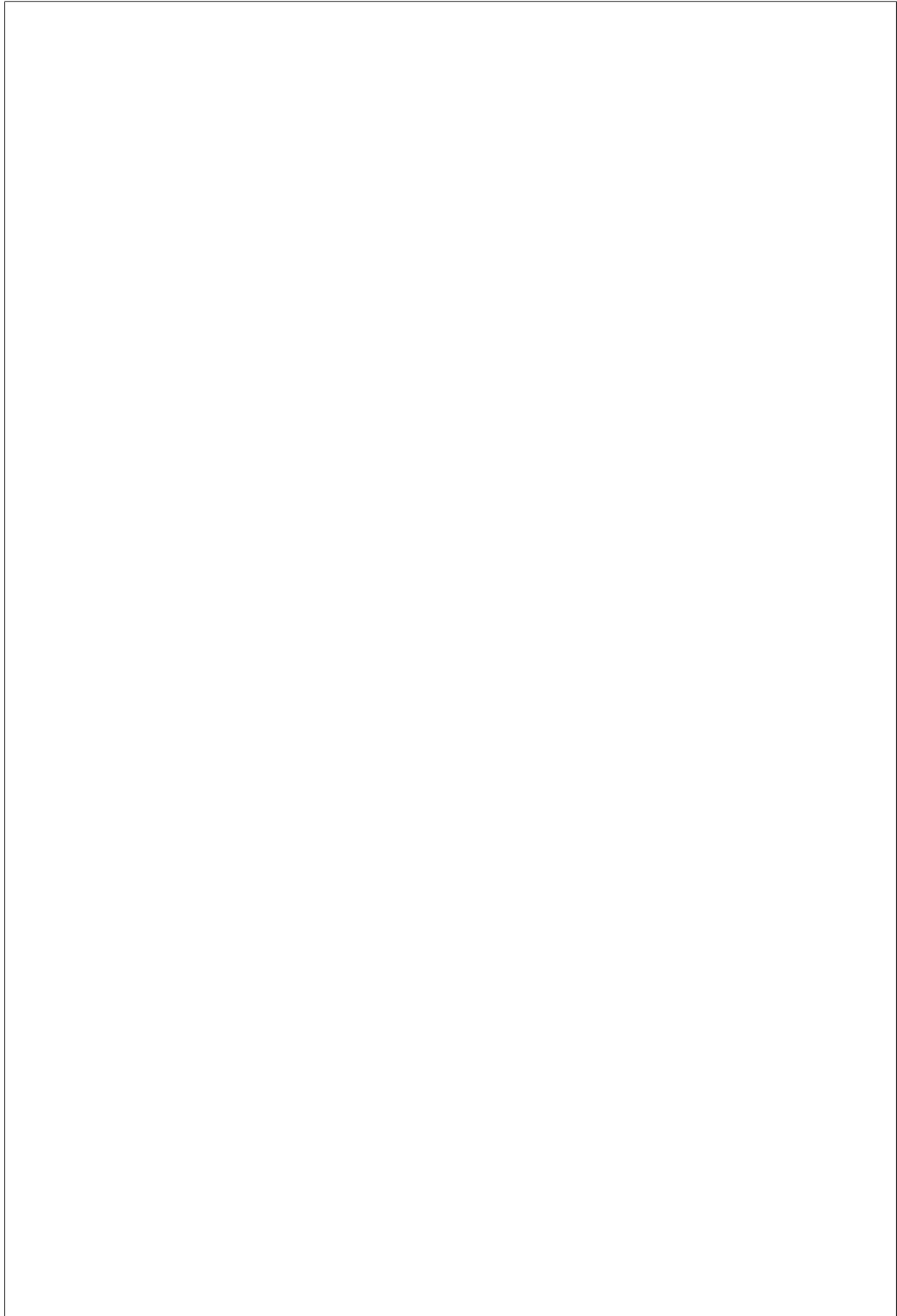
(continued from previous page)



(continues on next page)



(continued from previous page)



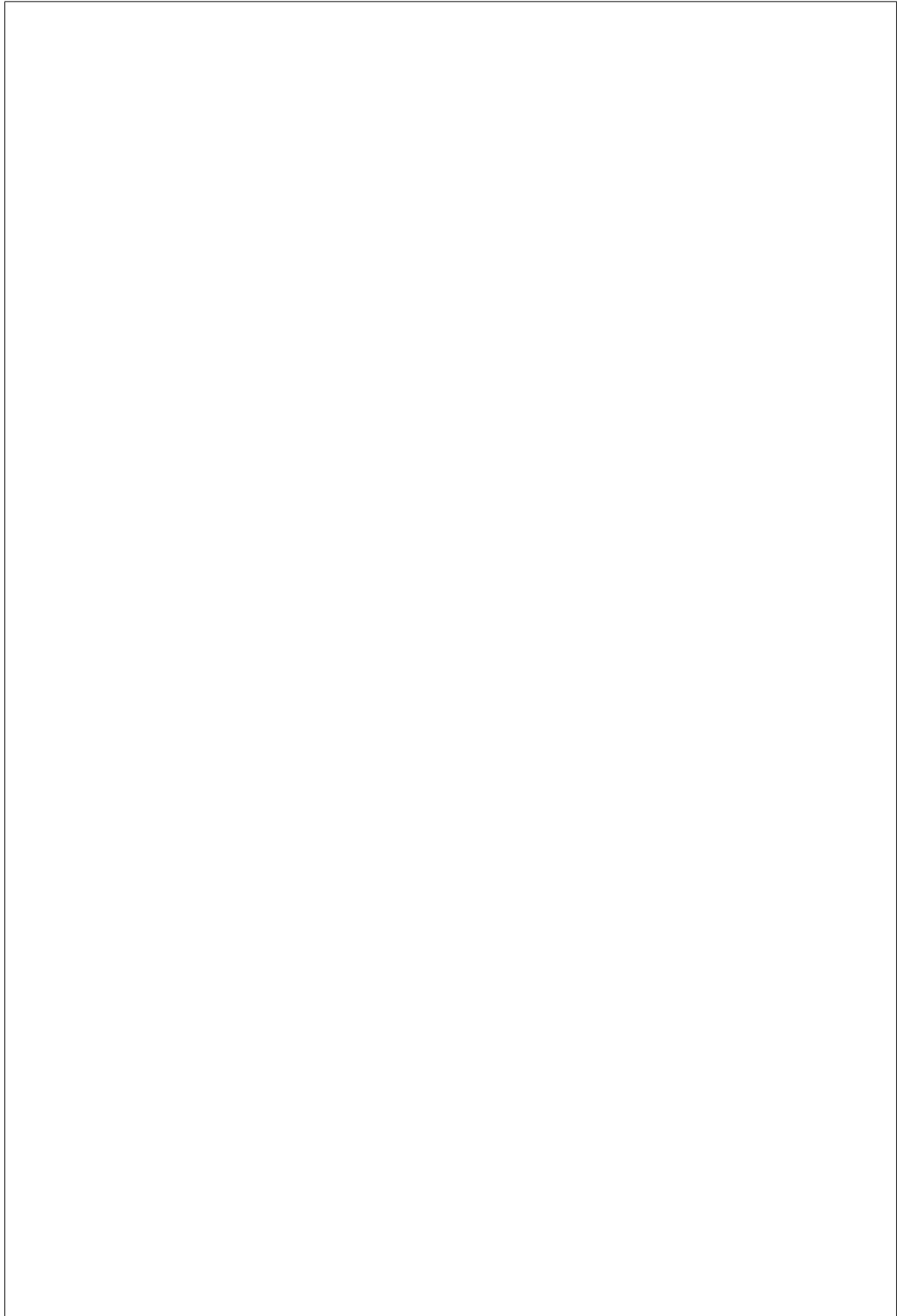
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



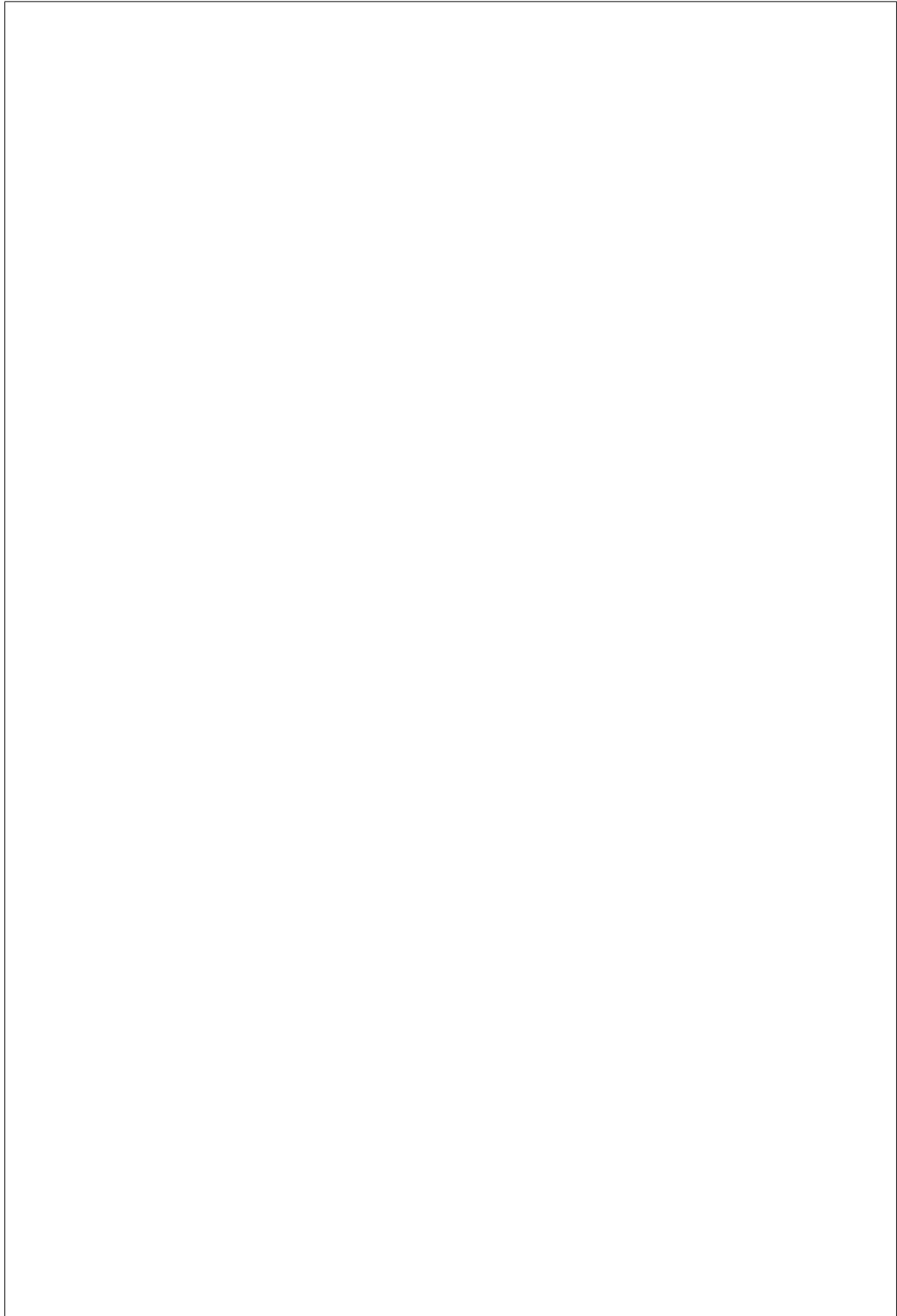
(continues on next page)

(continued from previous page)



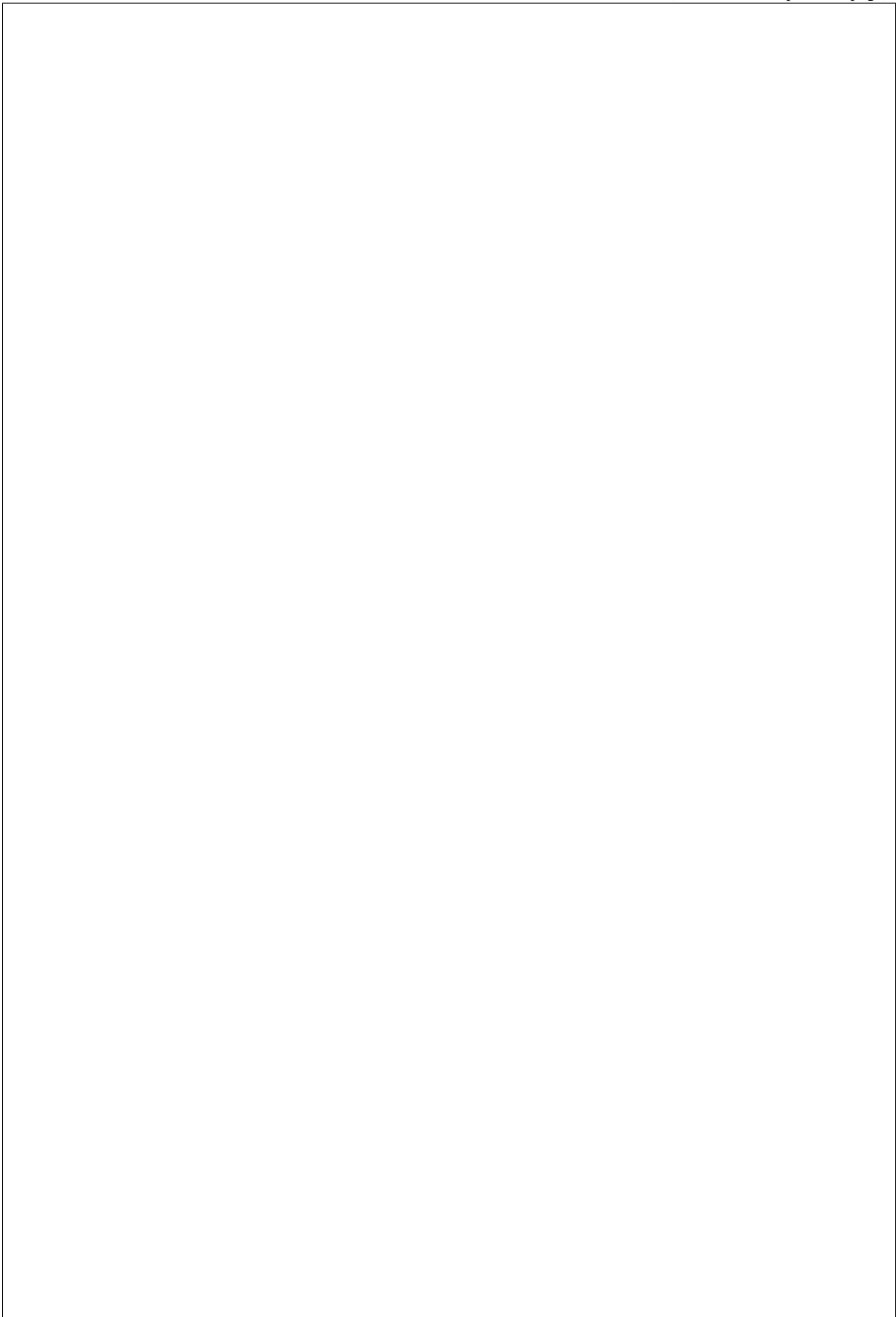
(continues on next page)

(continued from previous page)



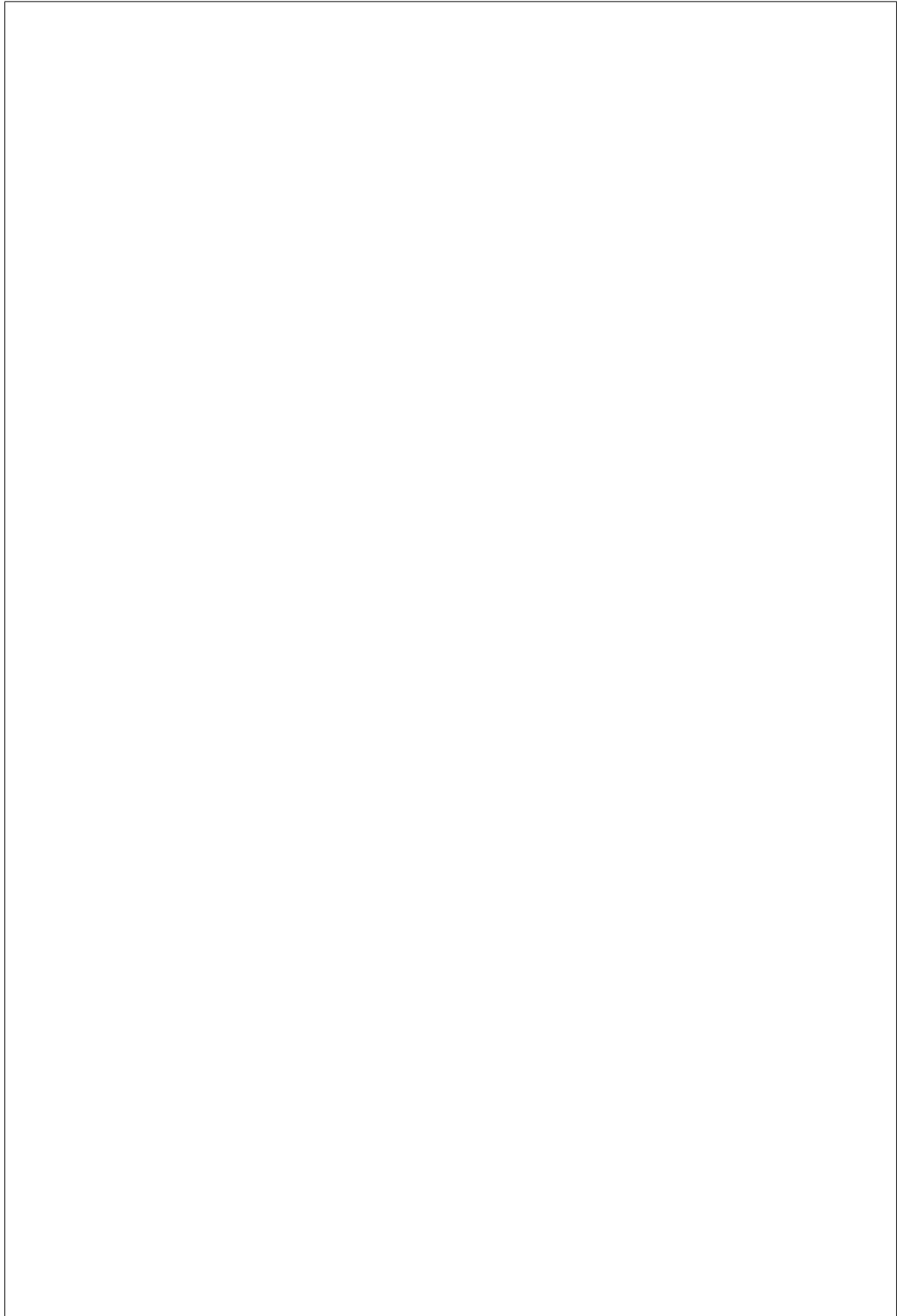
(continues on next page)

(continued from previous page)



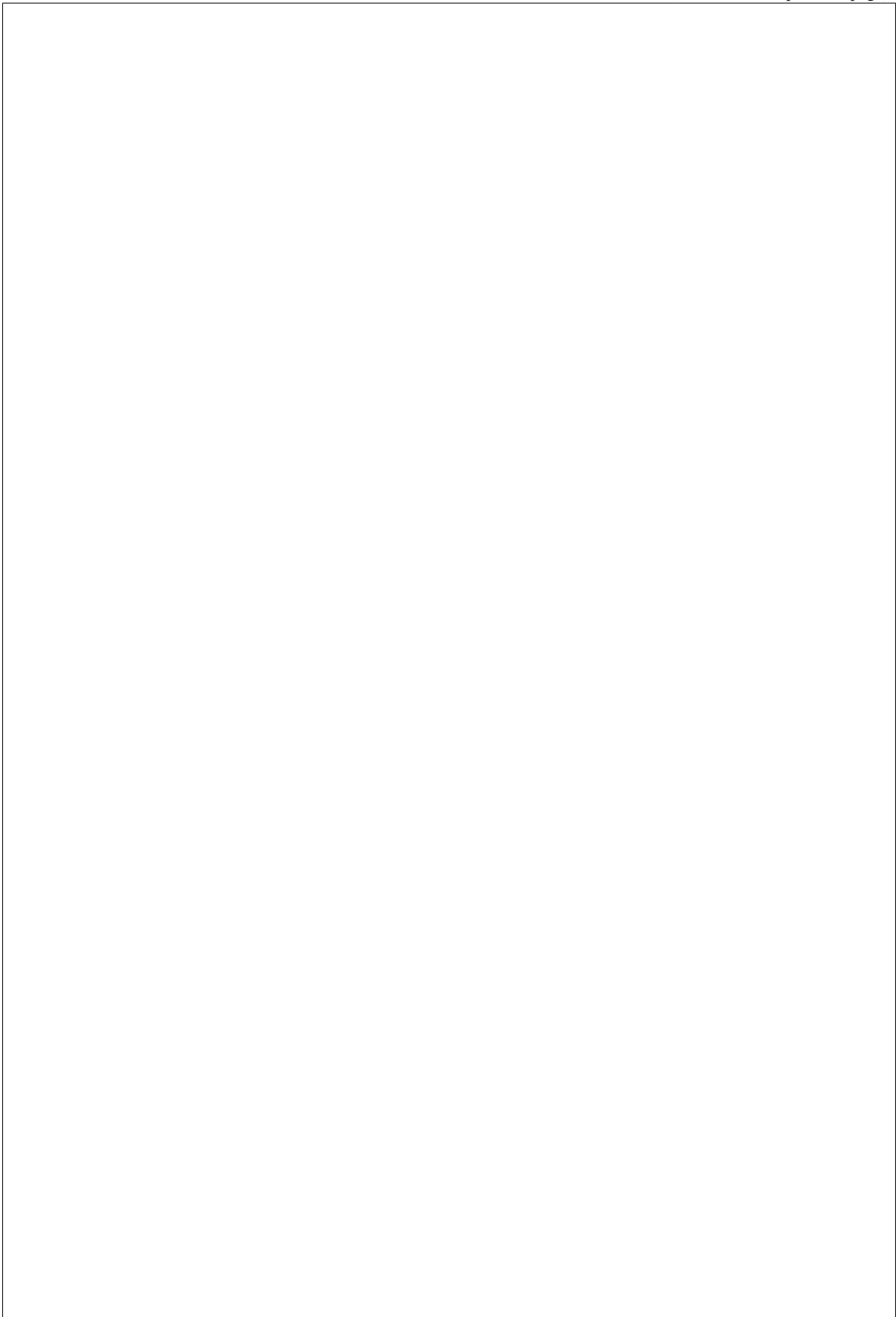
(continues on next page)

(continued from previous page)



(continues on next page)

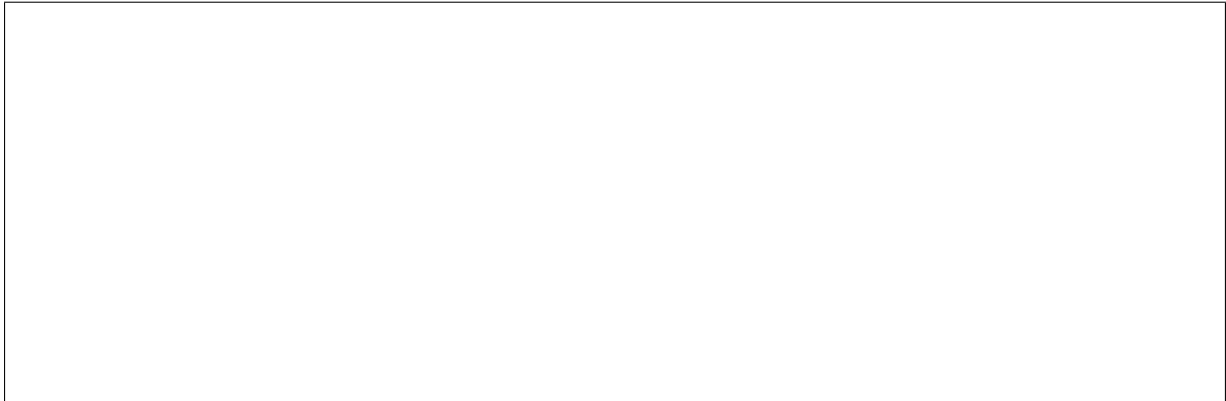
(continued from previous page)



(continues on next page)



(continued from previous page)



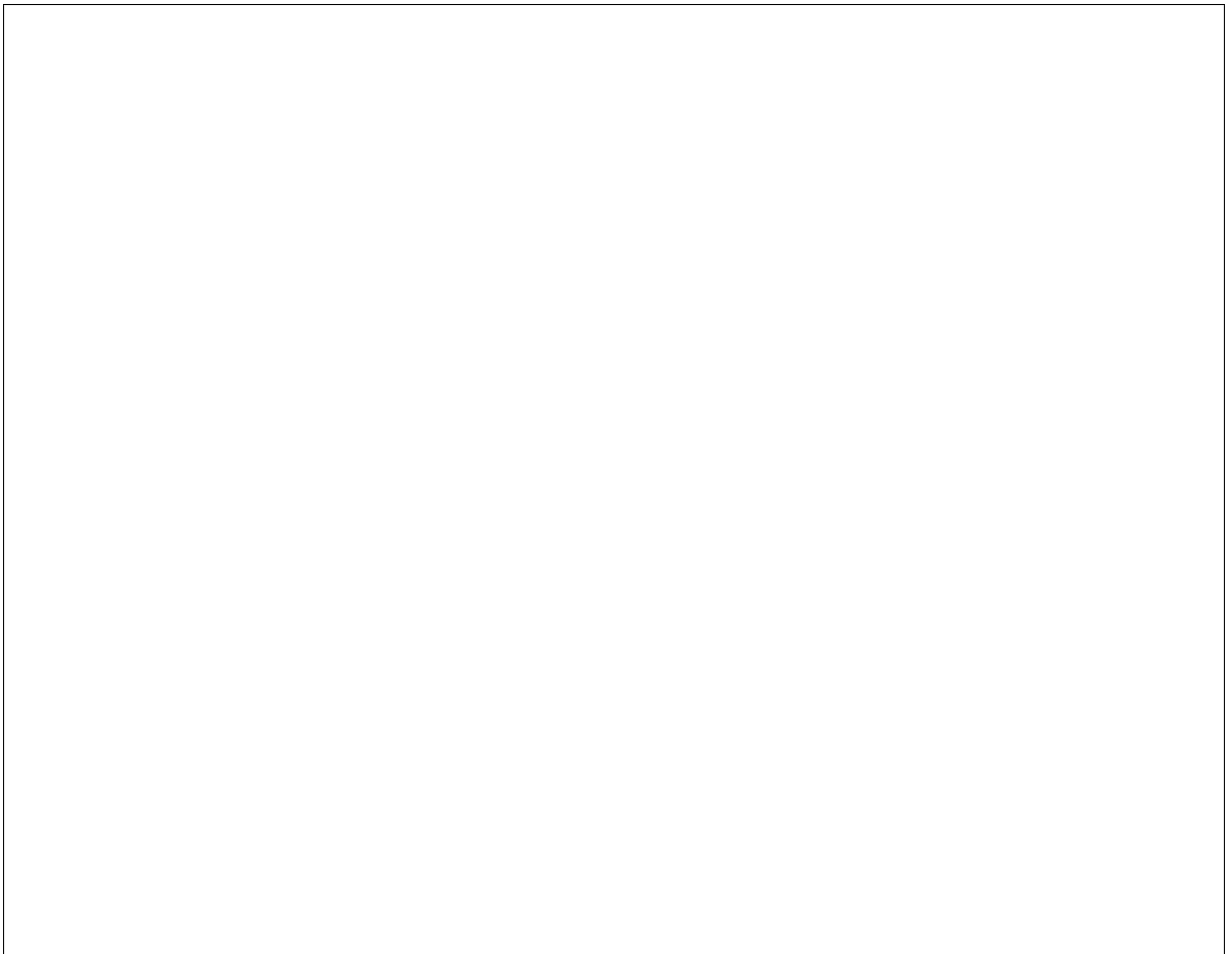
**baremetal.node.provision\_set**





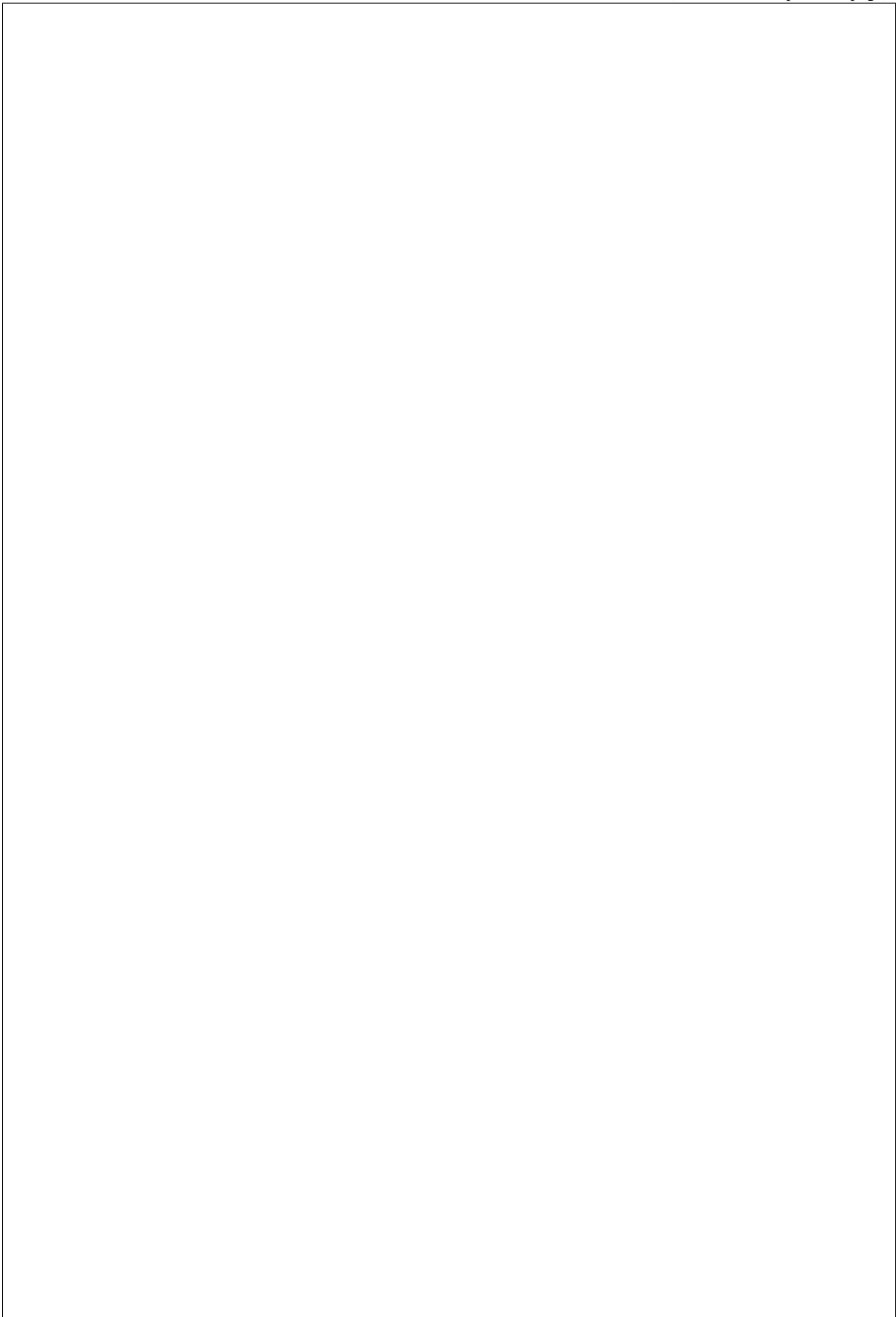


triggered the state change:



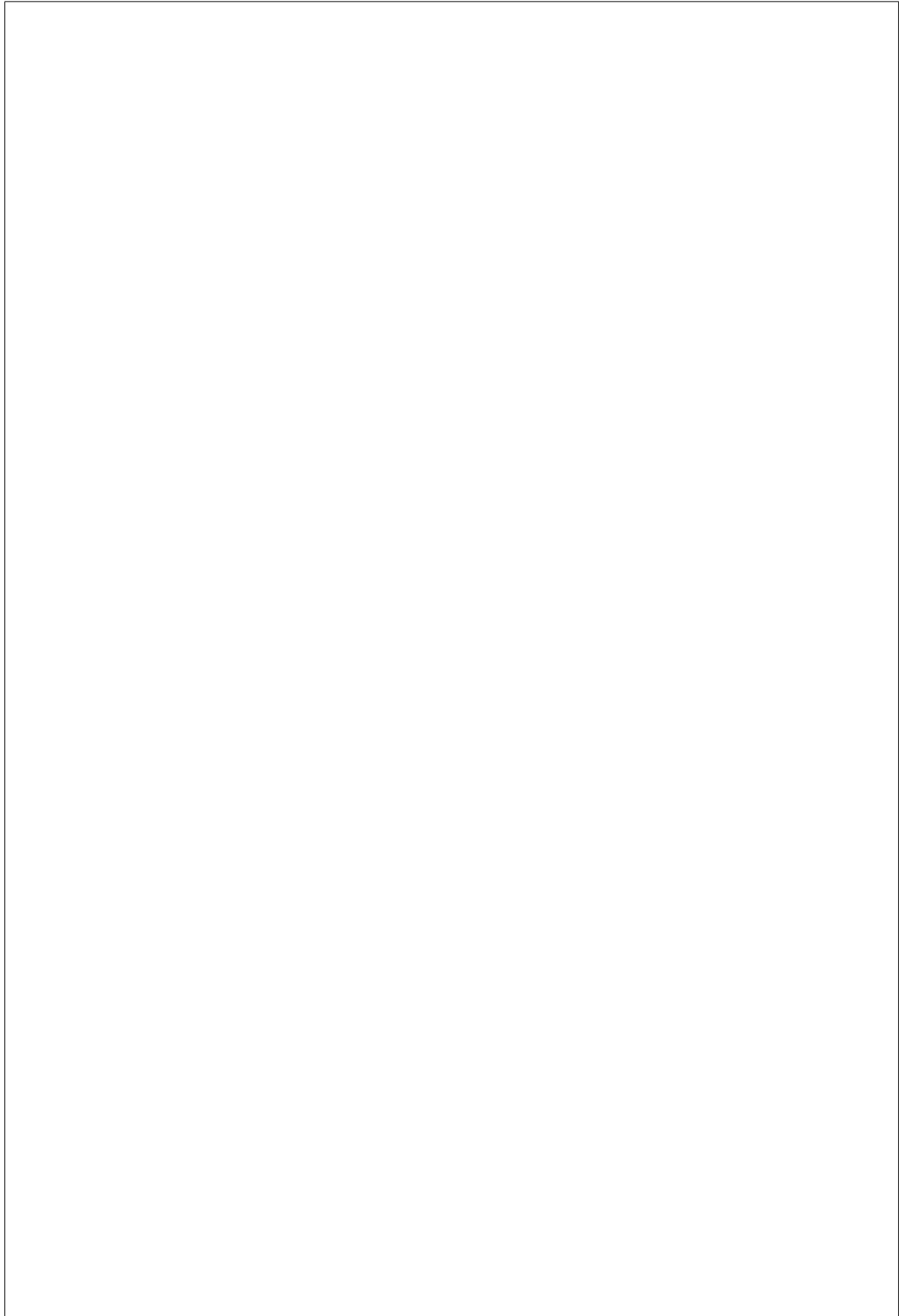
(continues on next page)

(continued from previous page)



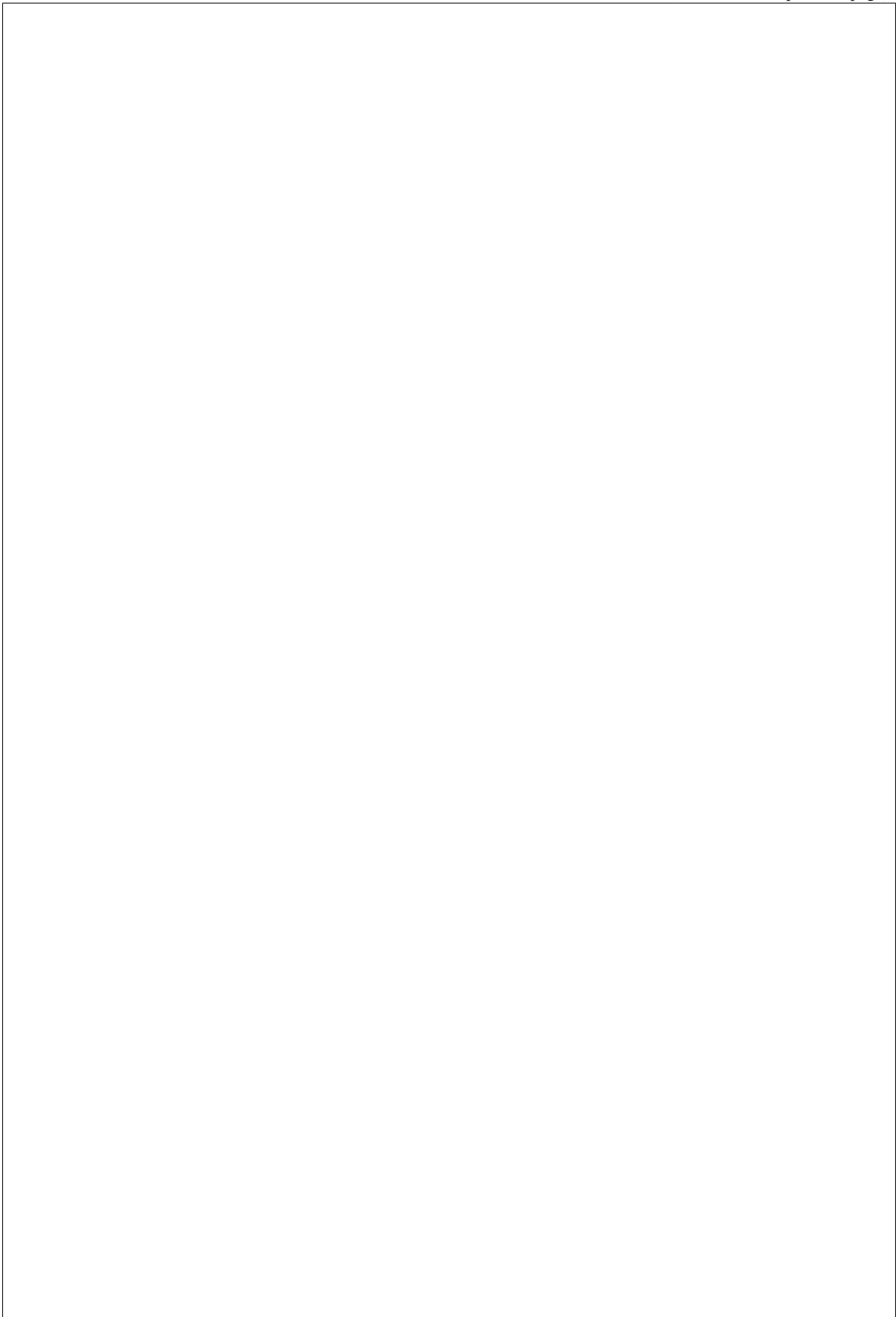
(continues on next page)

(continued from previous page)



(continues on next page)

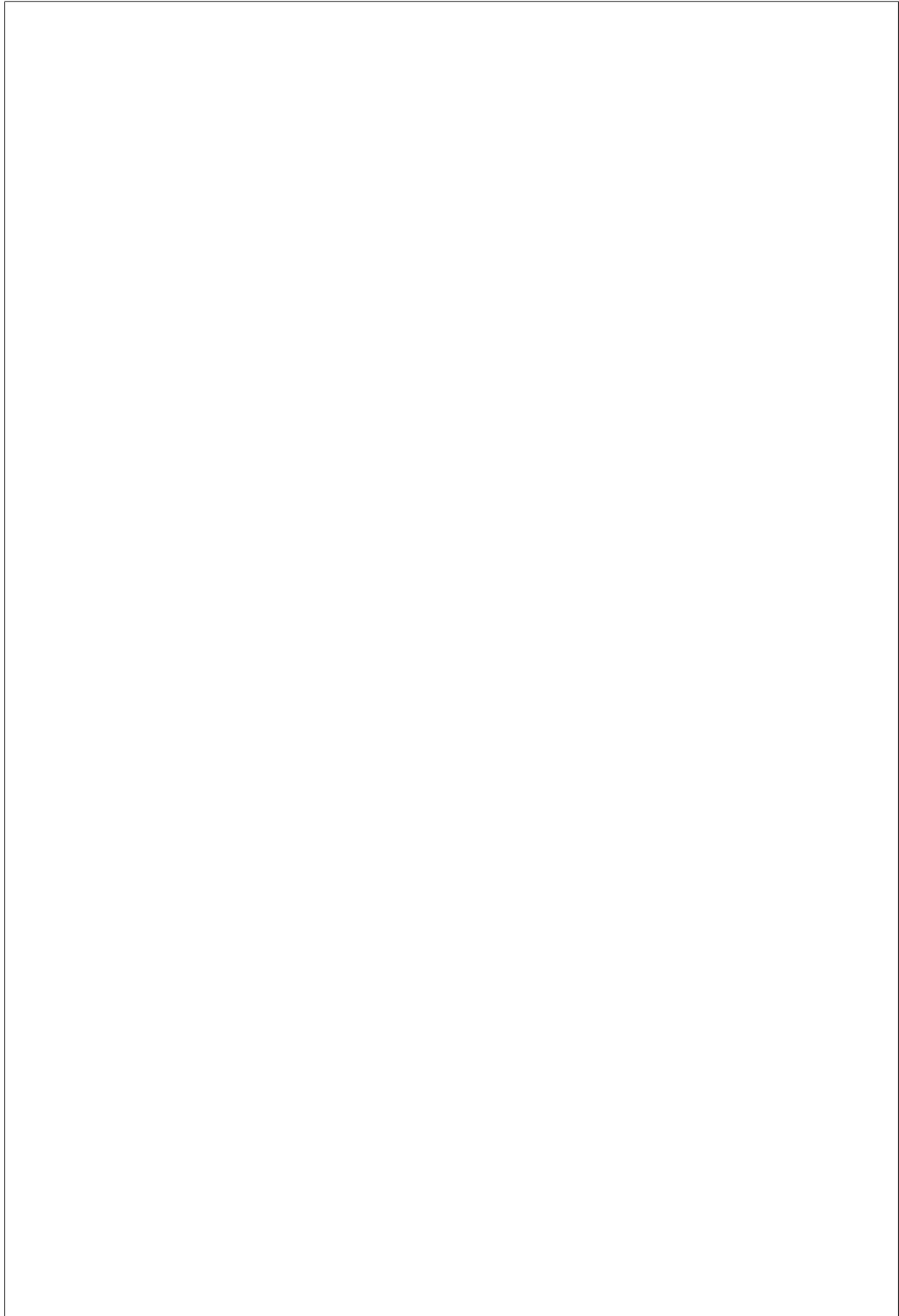
(continued from previous page)



(continues on next page)



(continued from previous page)



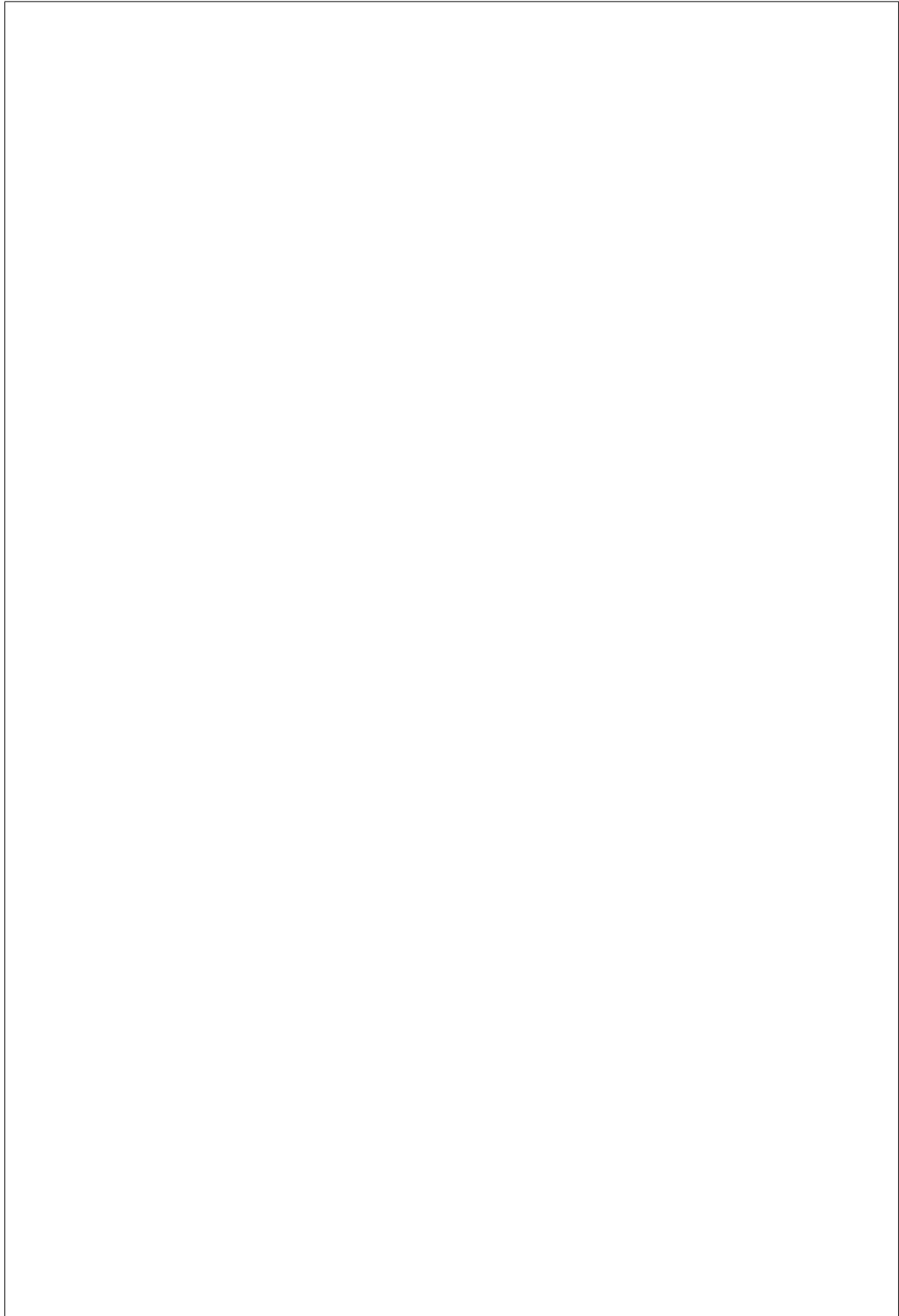
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



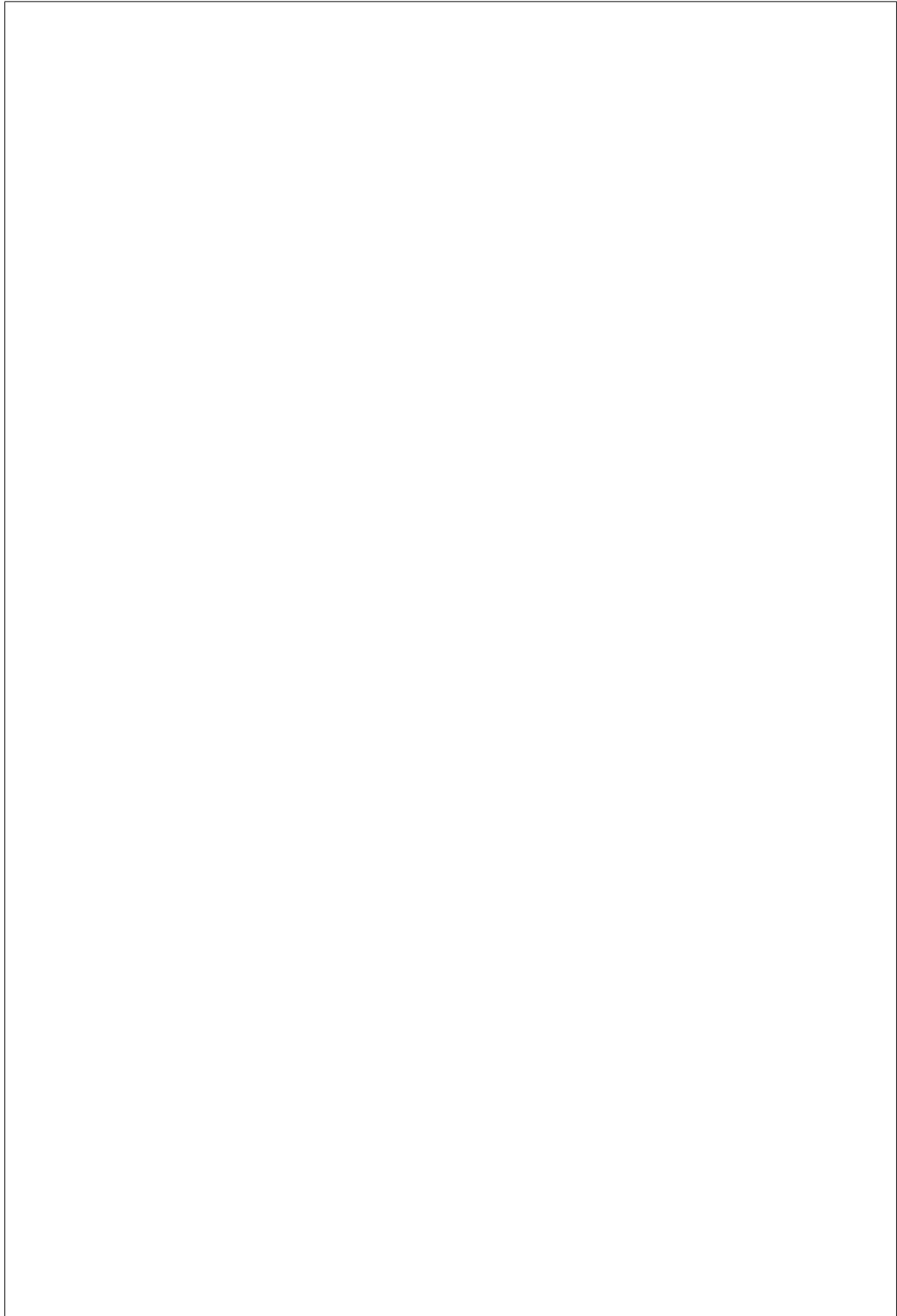
(continues on next page)

(continued from previous page)



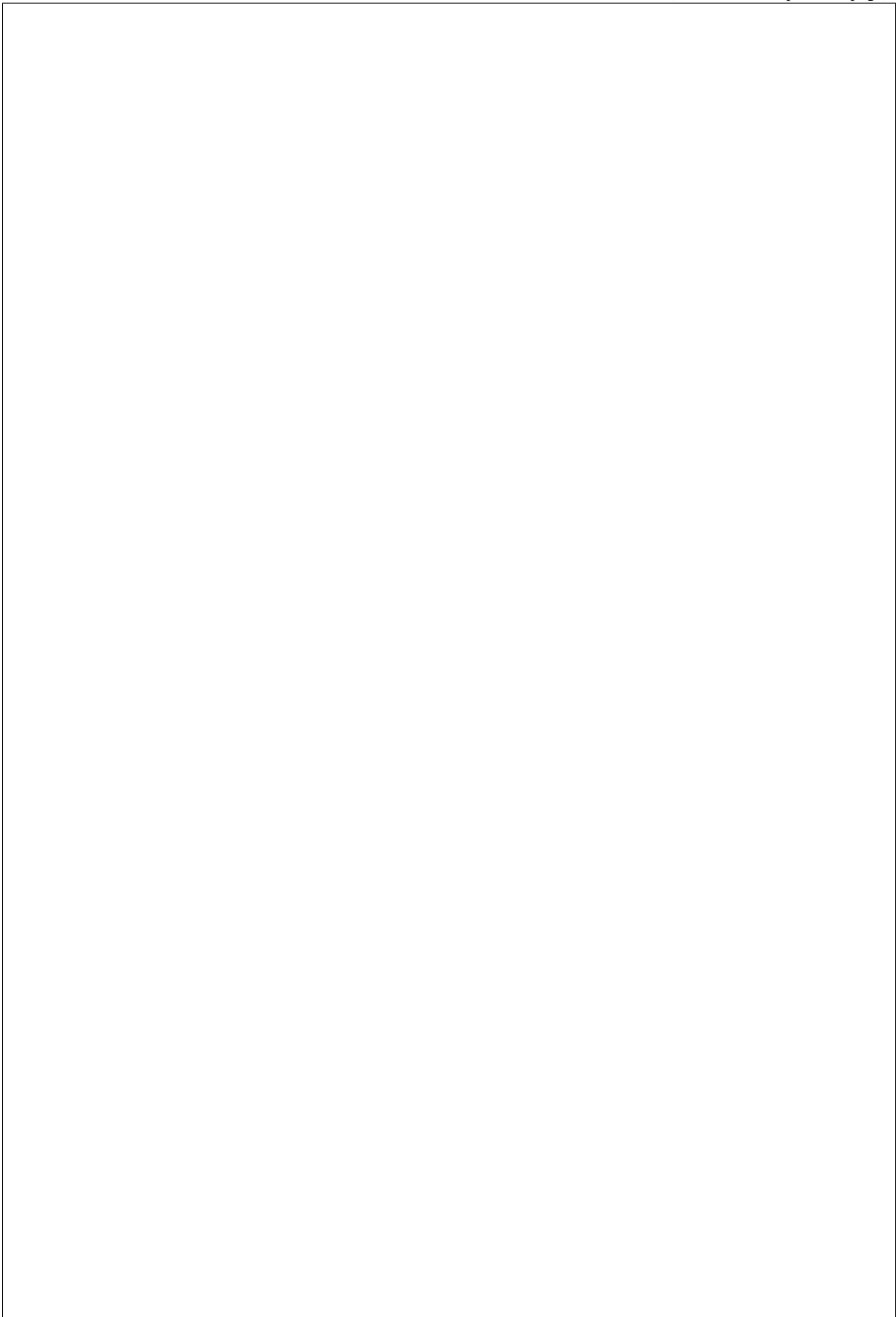
(continues on next page)

(continued from previous page)



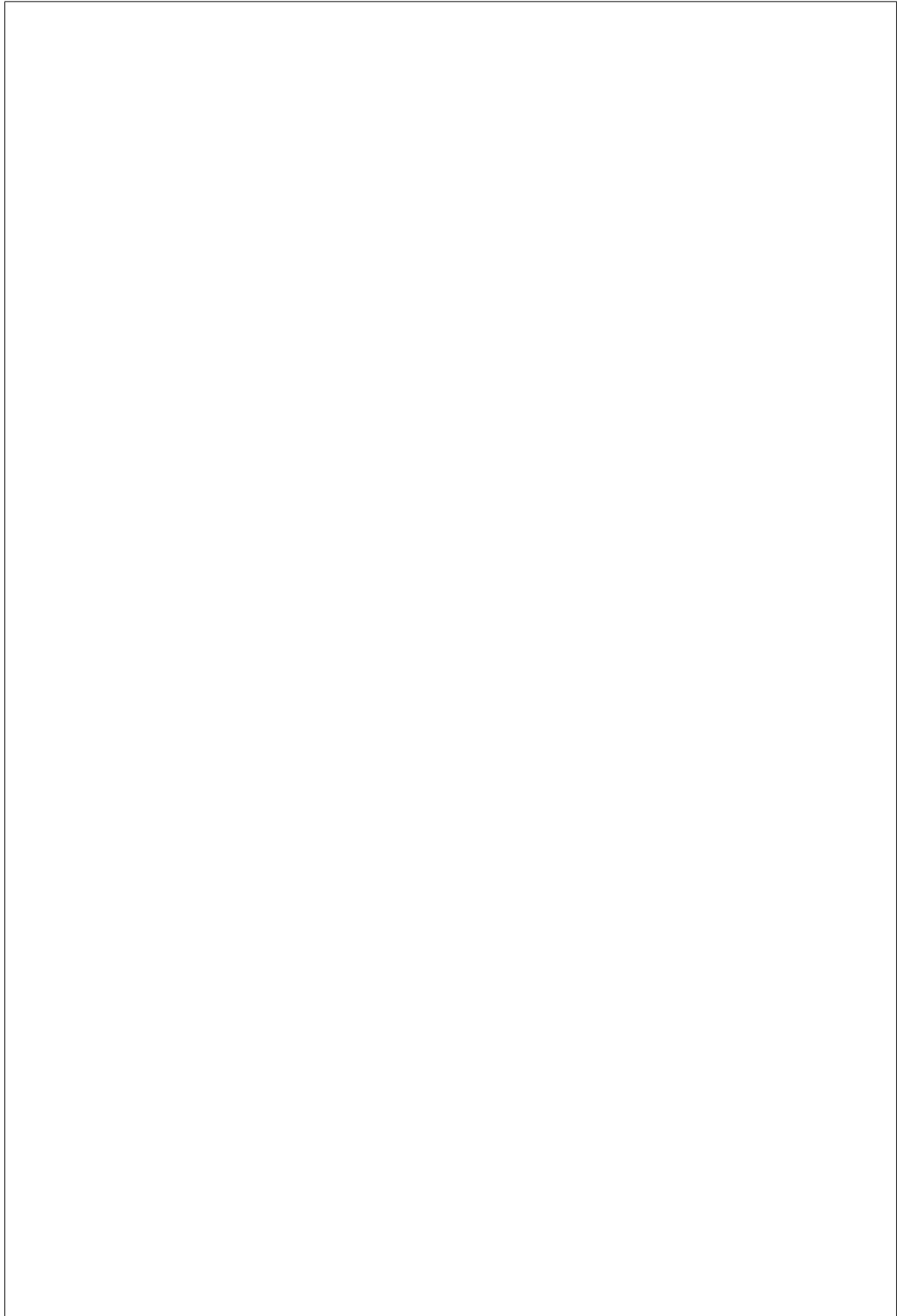
(continues on next page)

(continued from previous page)



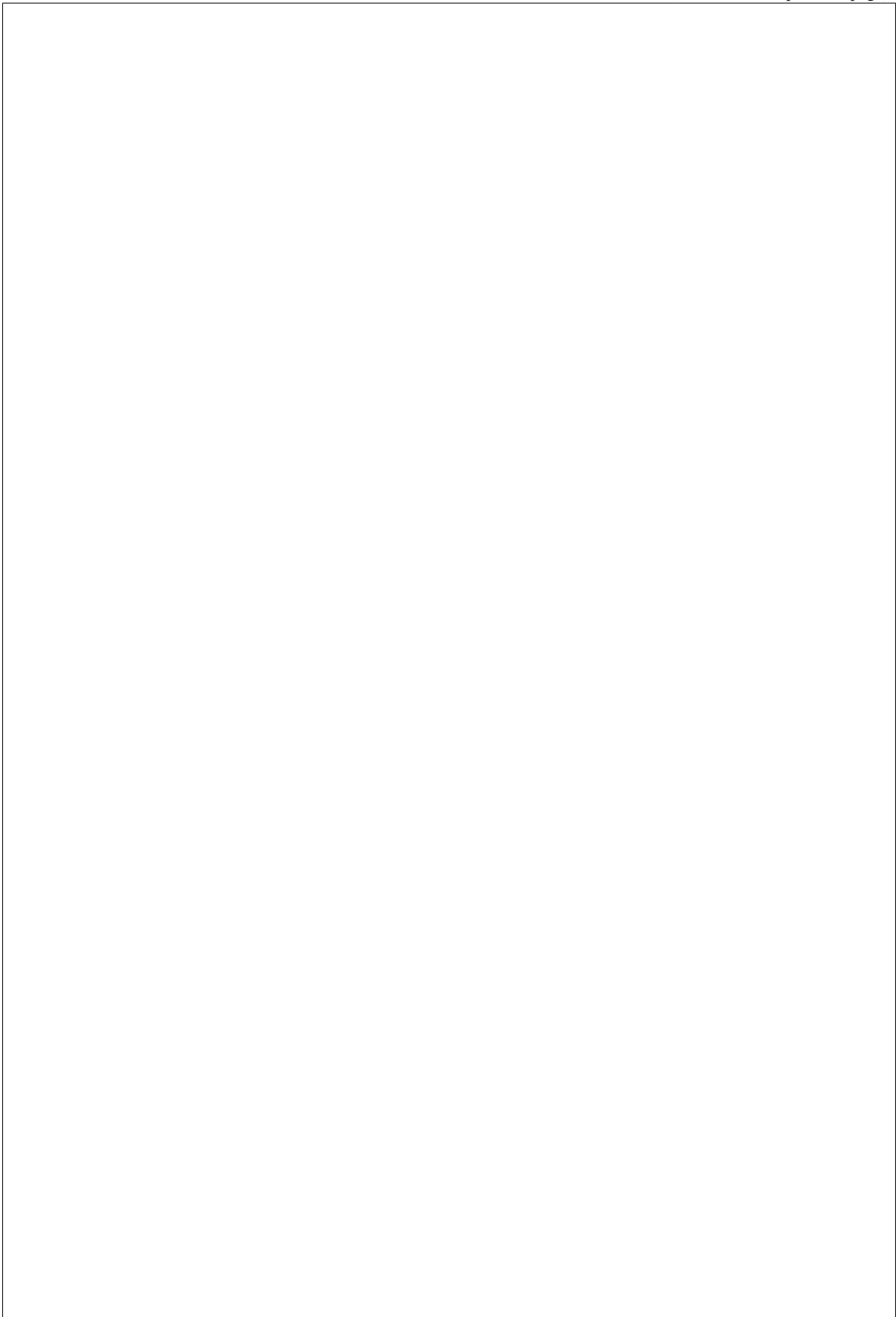
(continues on next page)

(continued from previous page)



(continues on next page)

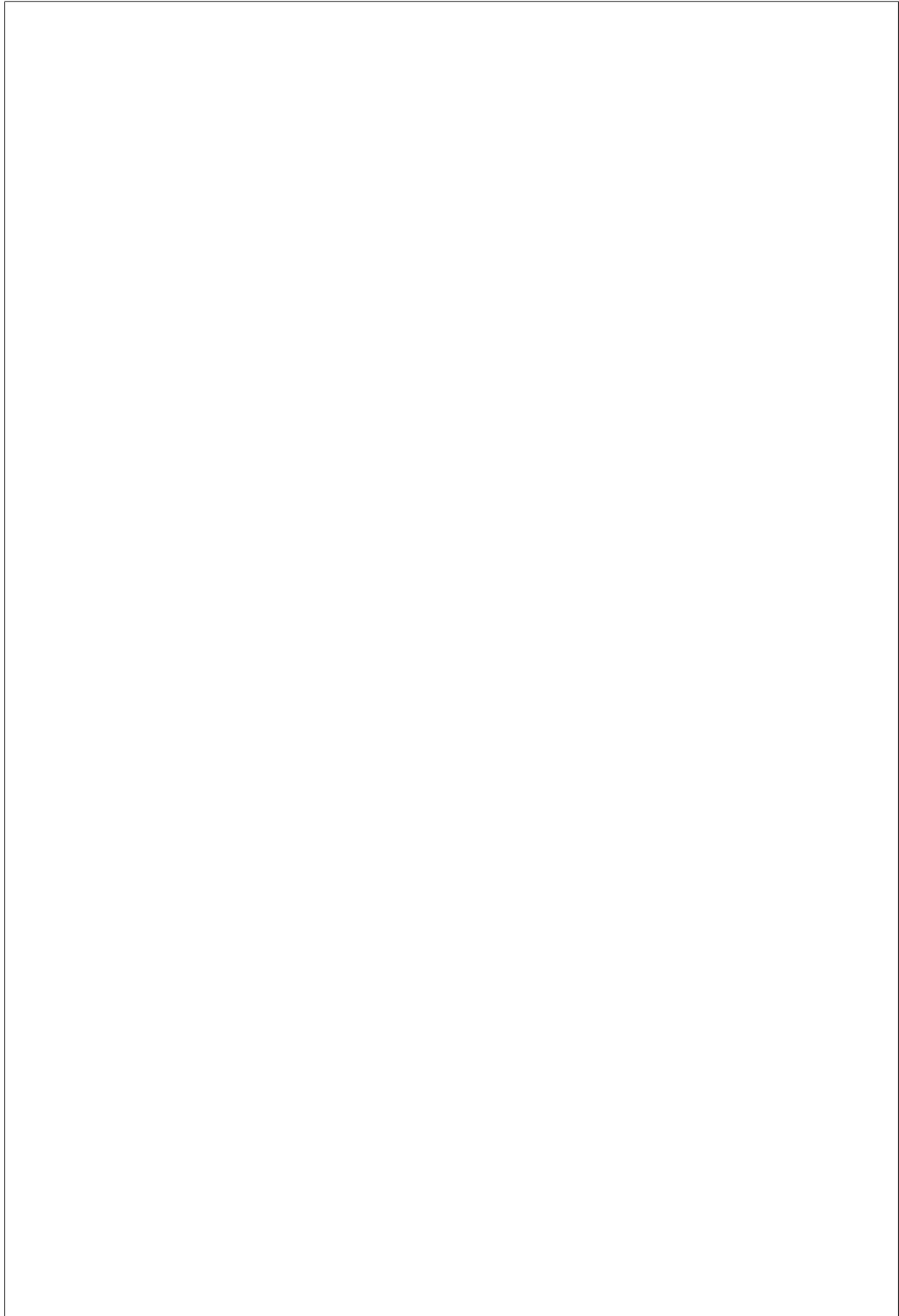
(continued from previous page)



(continues on next page)



(continued from previous page)



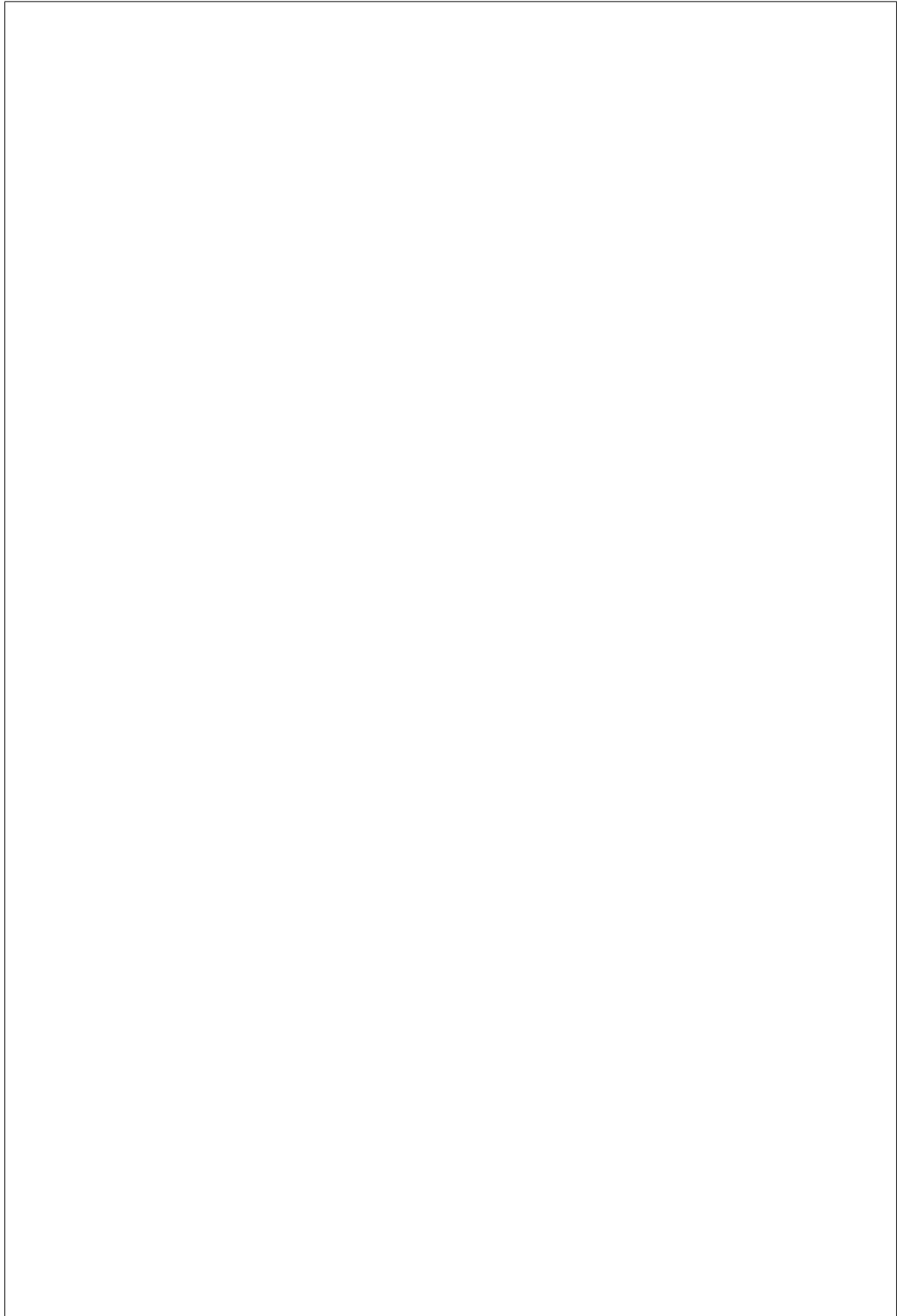
(continues on next page)

(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.

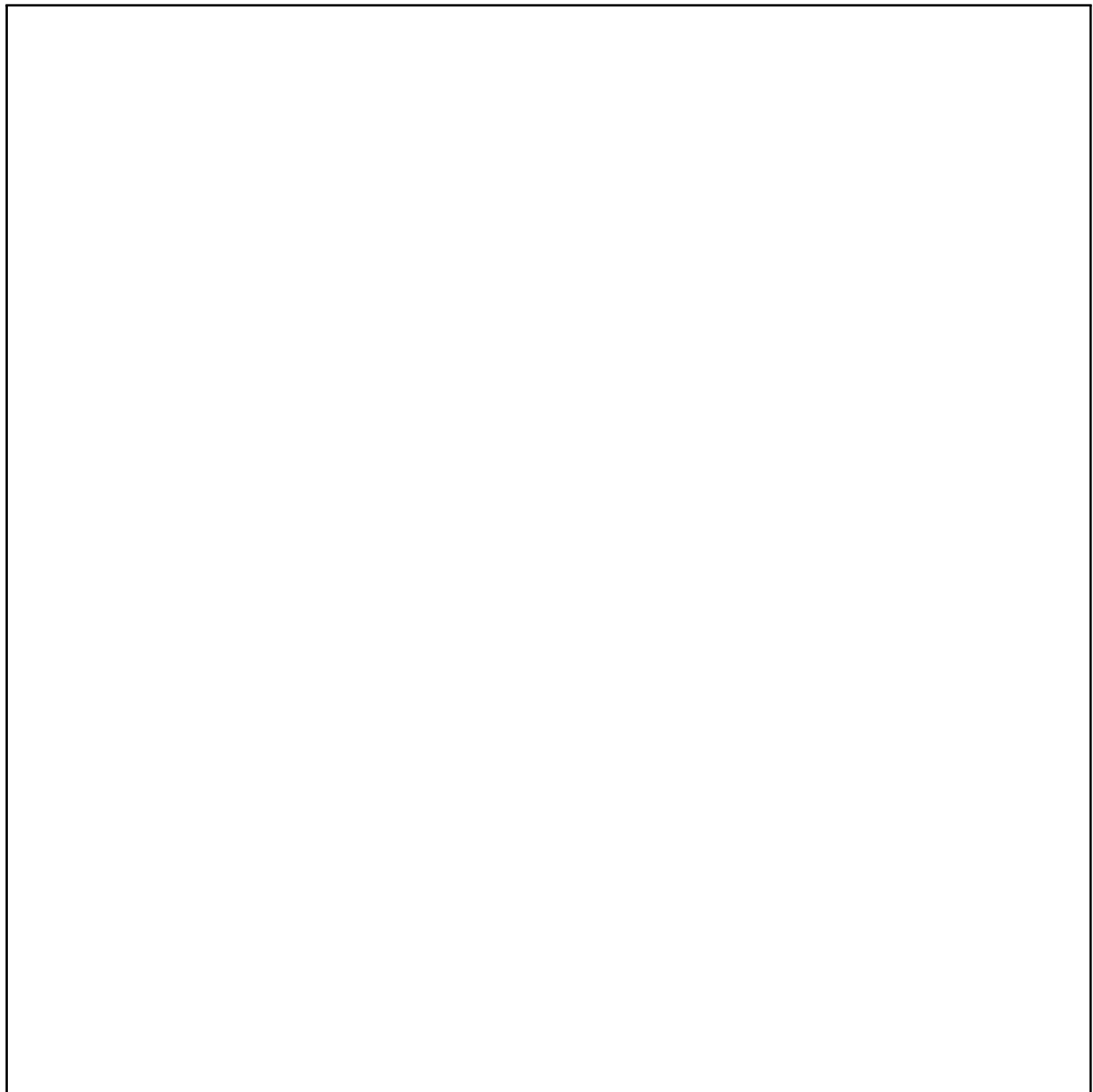
New York City.

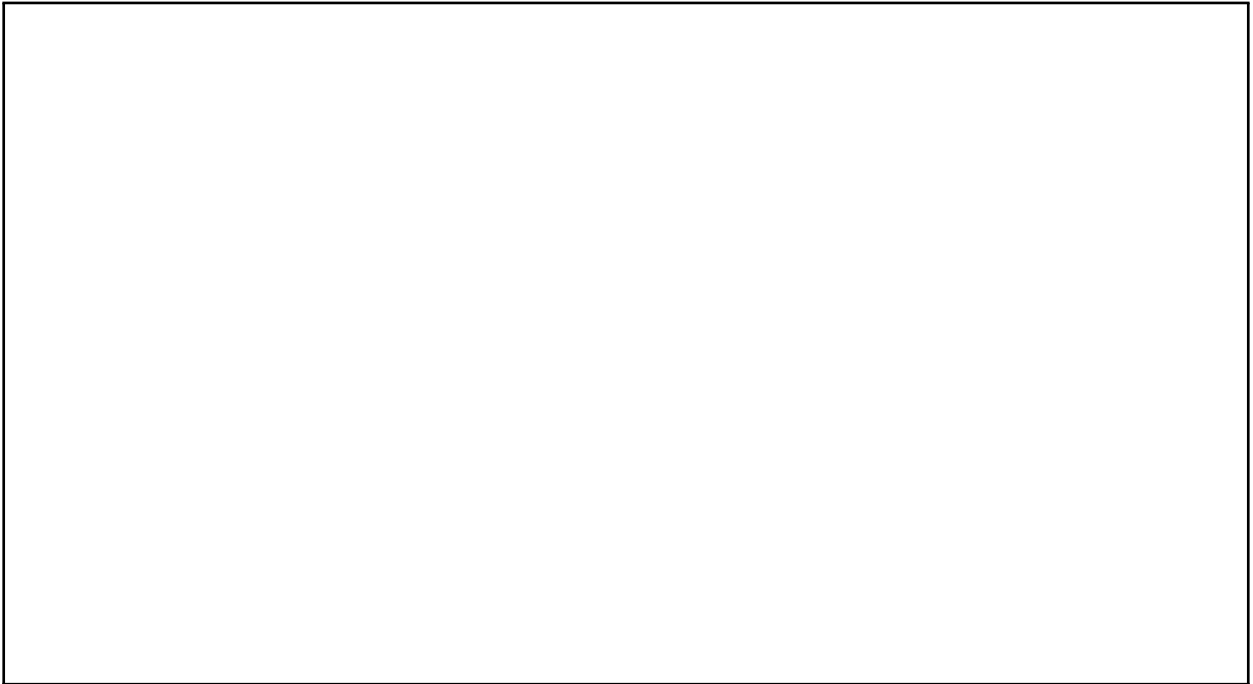
## **How it works**



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.

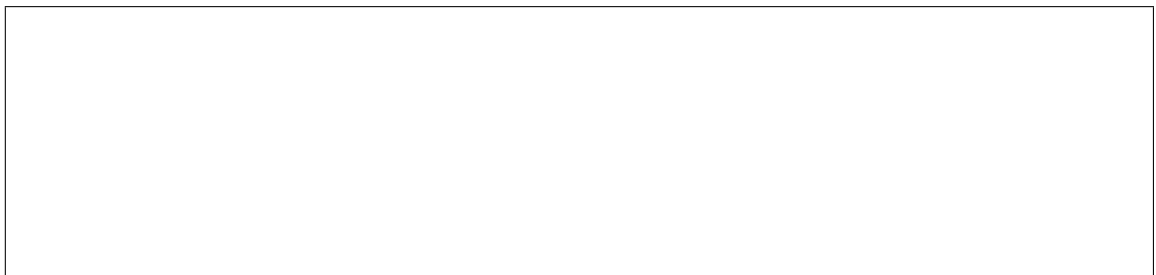
### **REST API: user roles and policy settings**







`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.

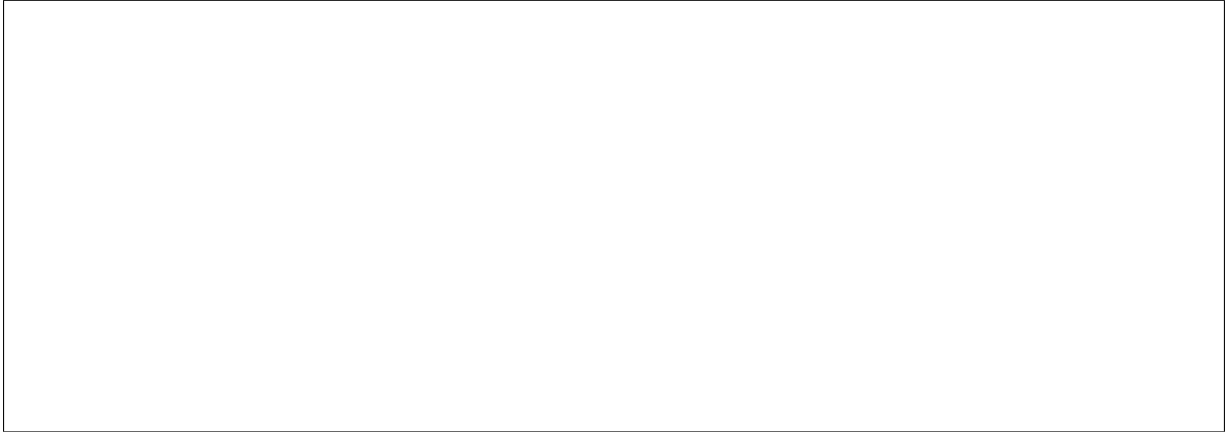


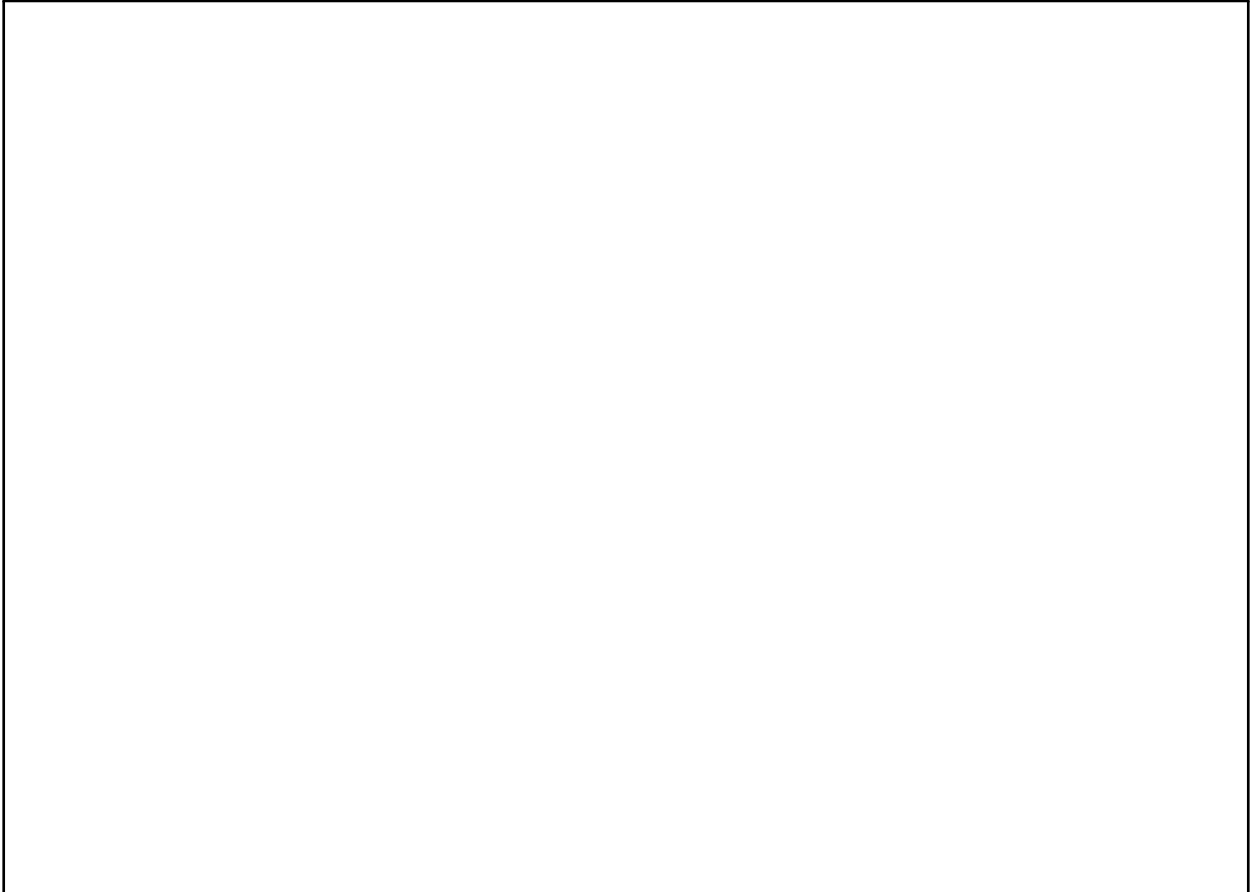




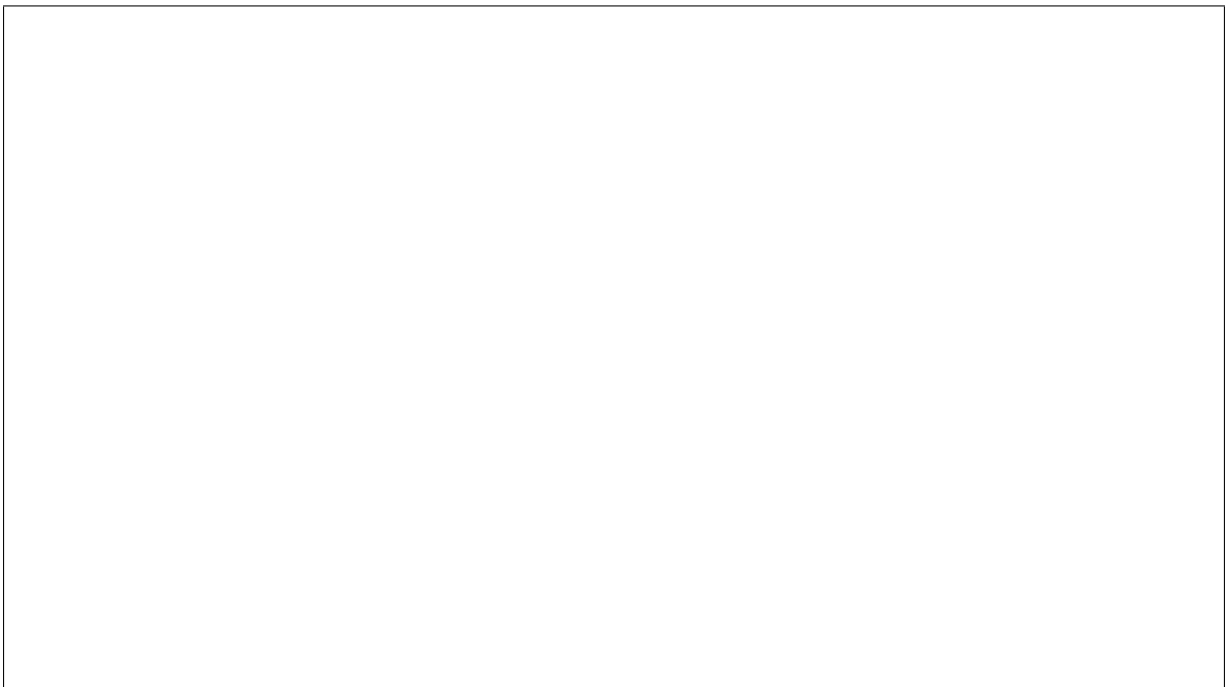
**UEFI secure boot mode**







**Compatible images**



## **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 deploy 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**

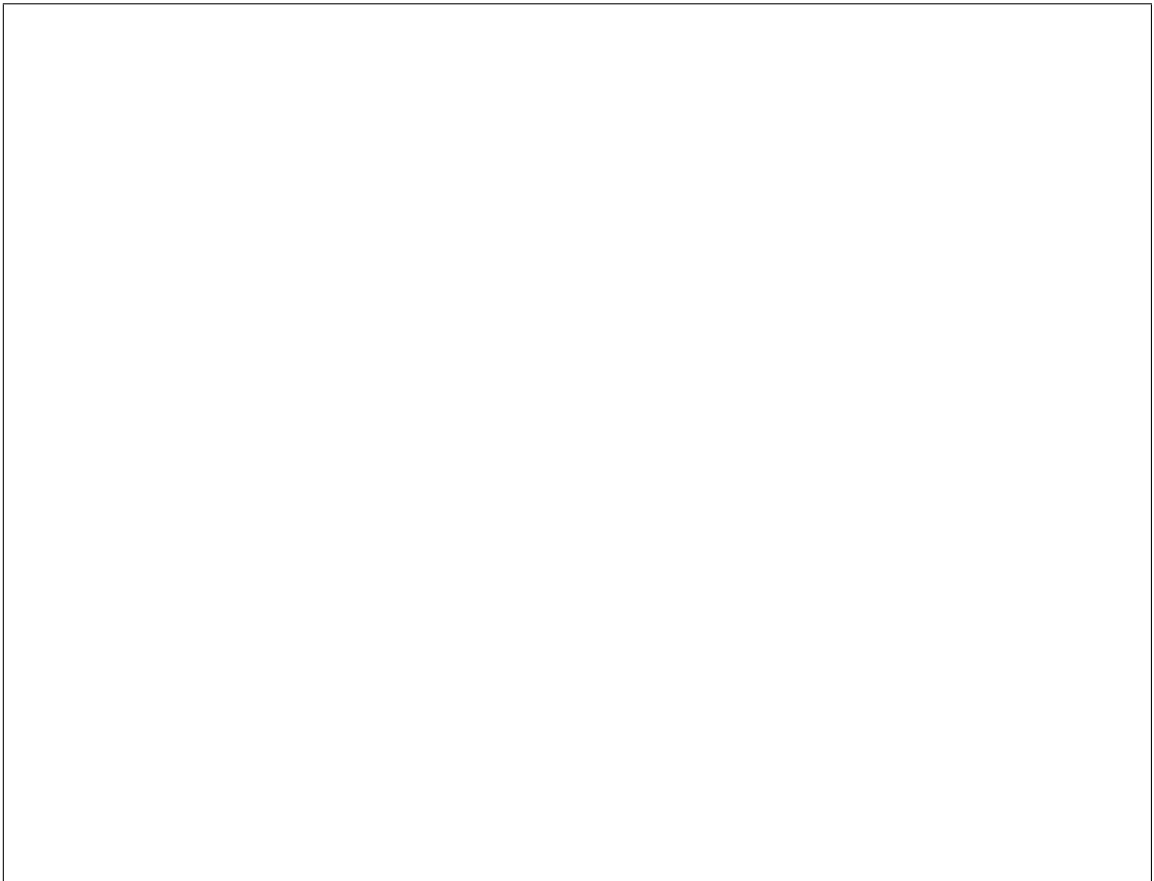


**API endpoints for RAM disk use**







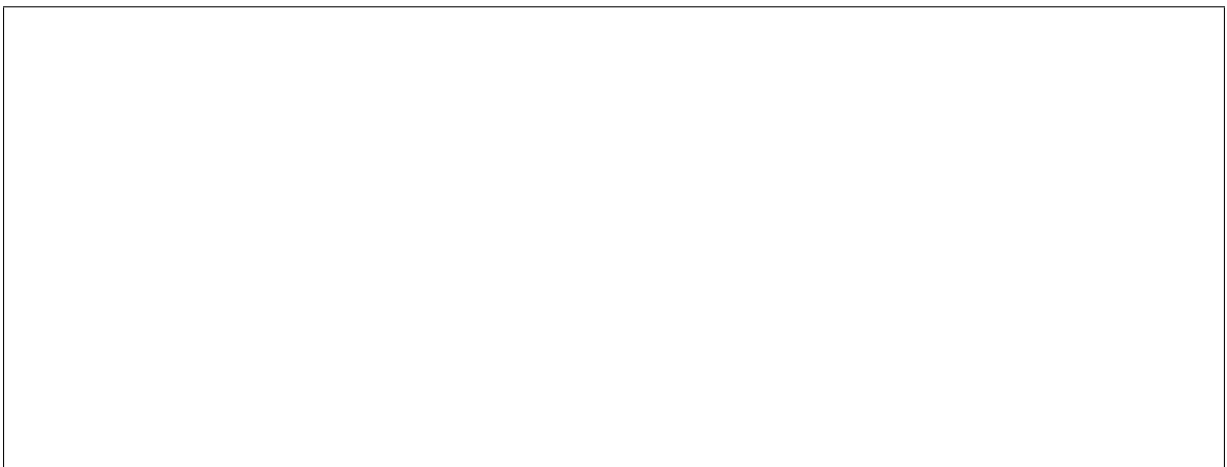


(continues on next page)

(continued from previous page)

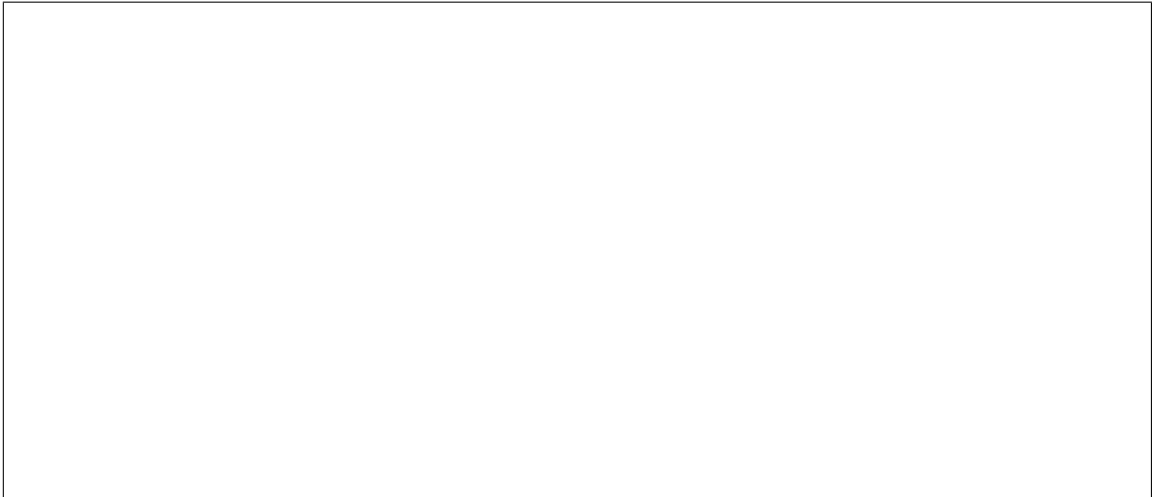
---

**Nova returns No valid host was found Error**









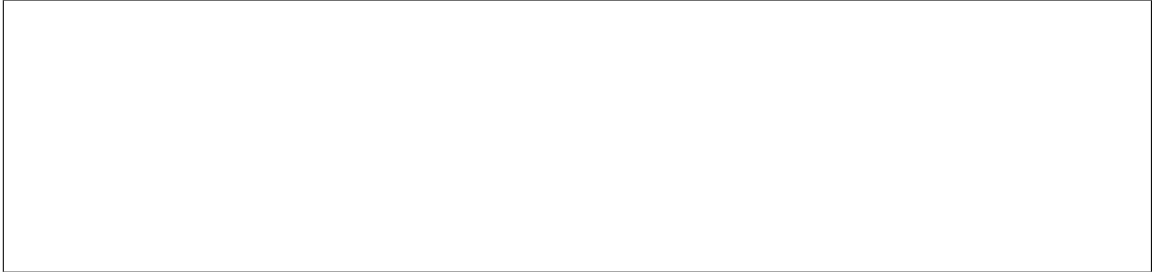
(continues on next page)

(continued from previous page)

--

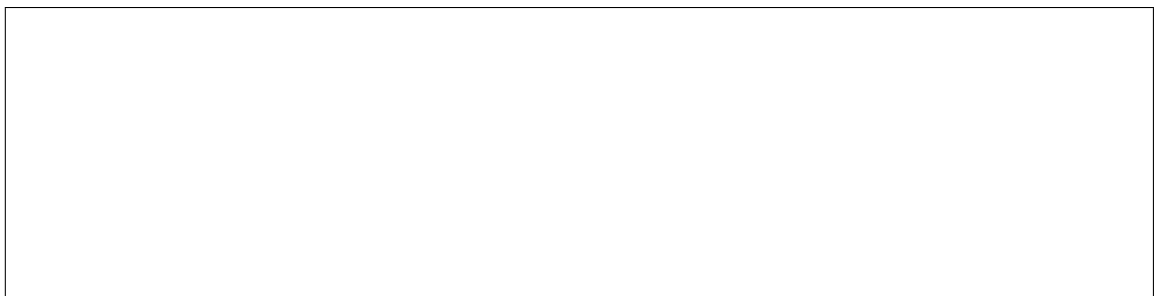
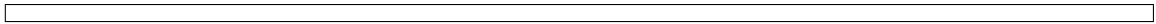


nance mode:



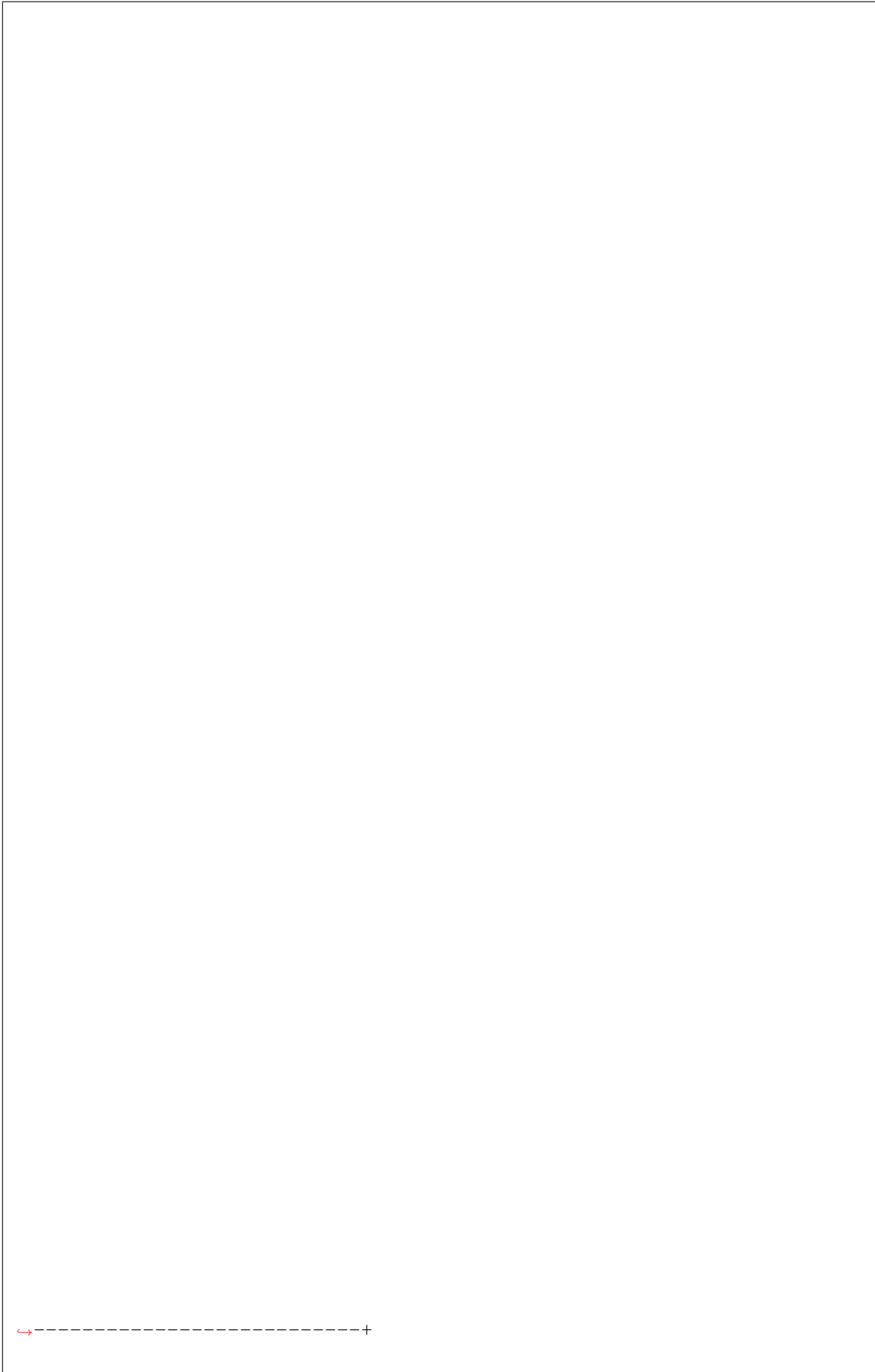
(continues on next page)

(continued from previous page)



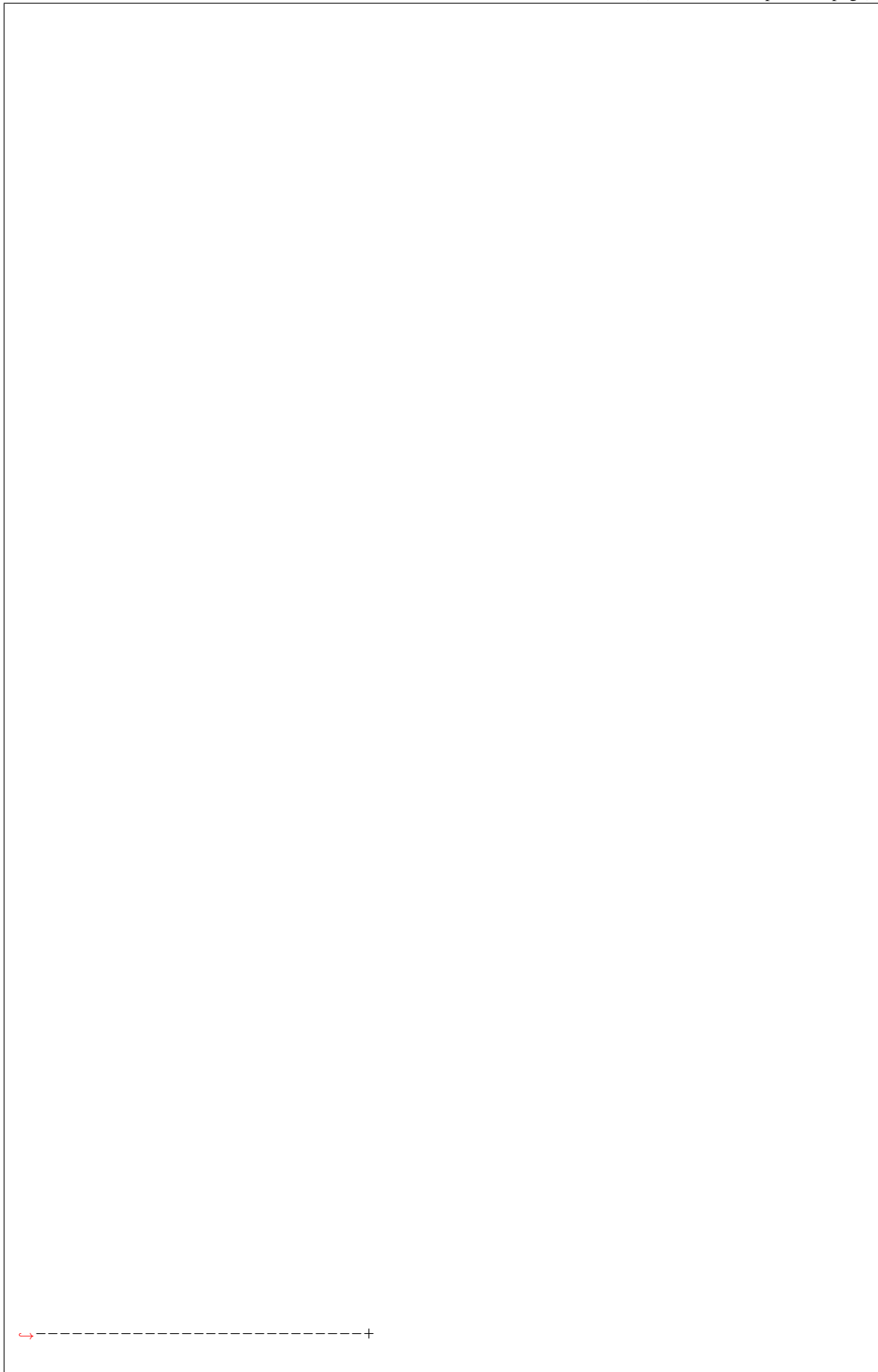
(continues on next page)

(continued from previous page)



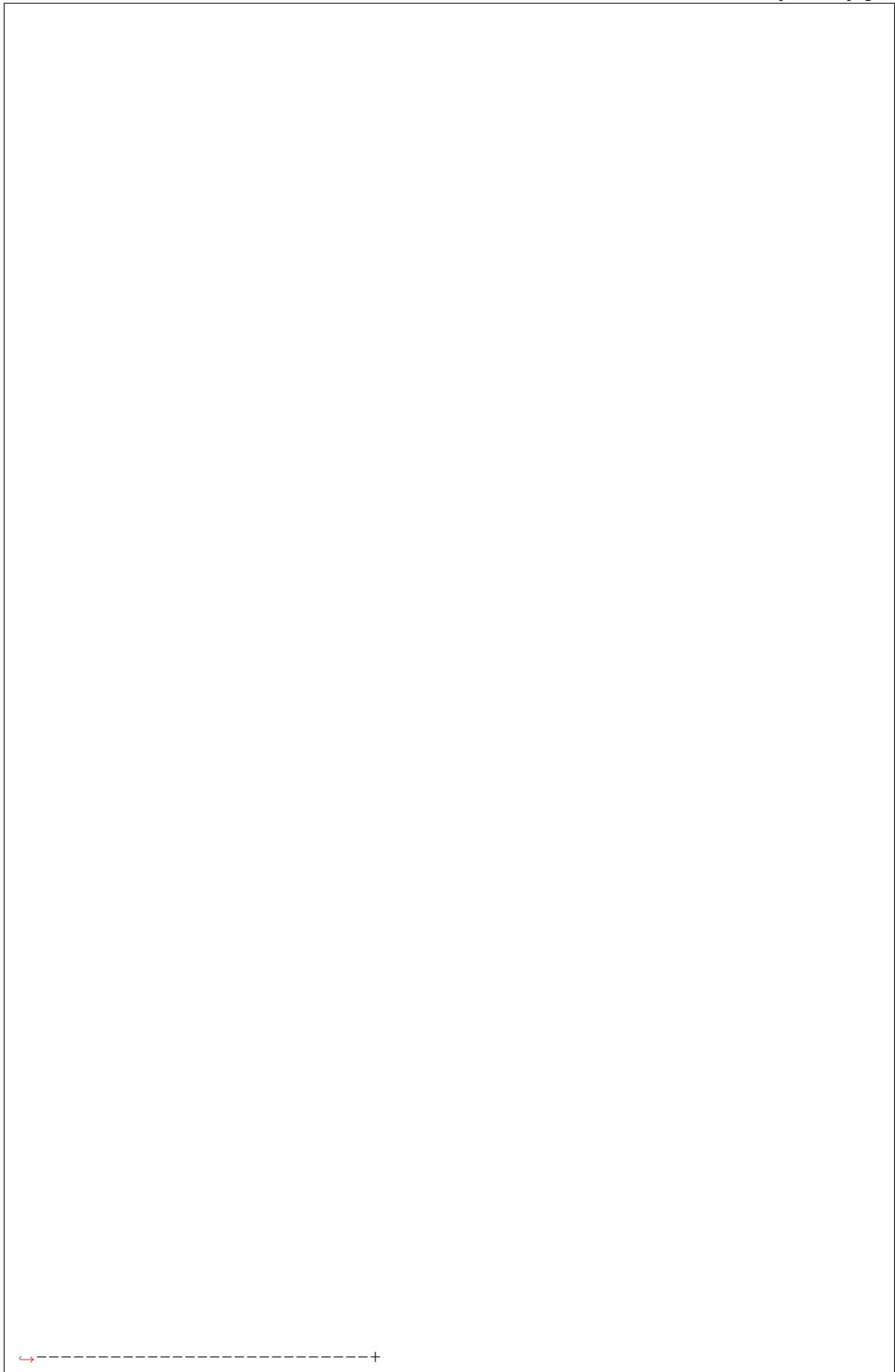
(continues on next page)

(continued from previous page)



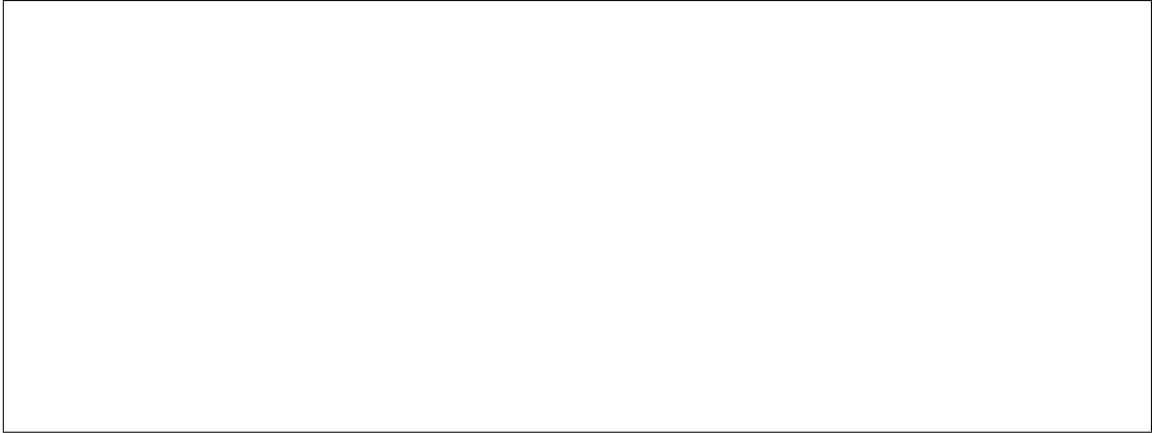
(continues on next page)

(continued from previous page)



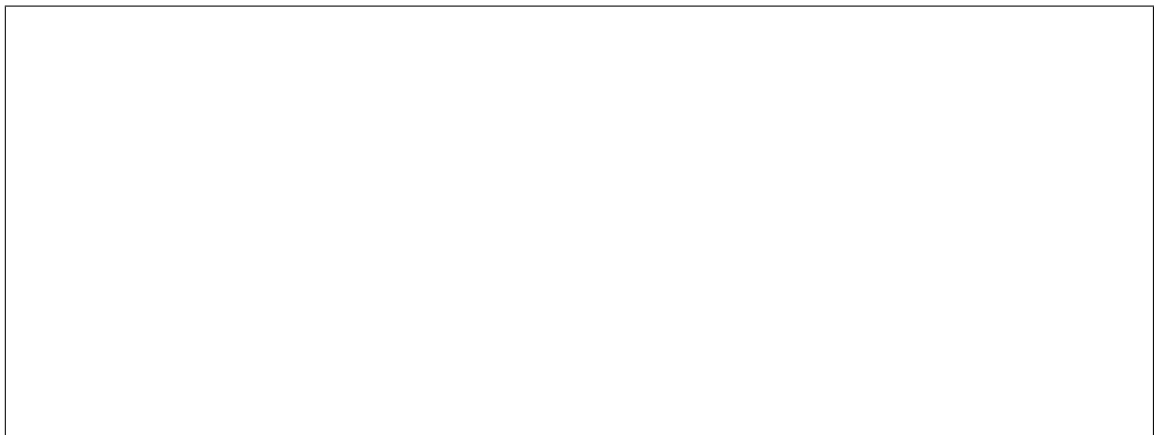
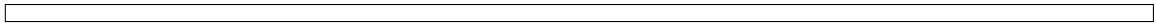


source of the failures, then re-enable it:



(continues on next page)

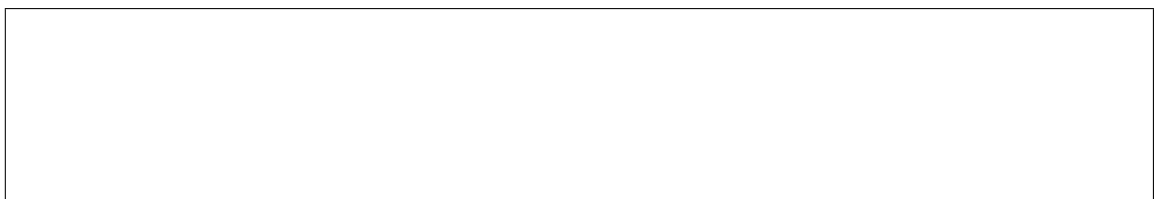
(continued from previous page)





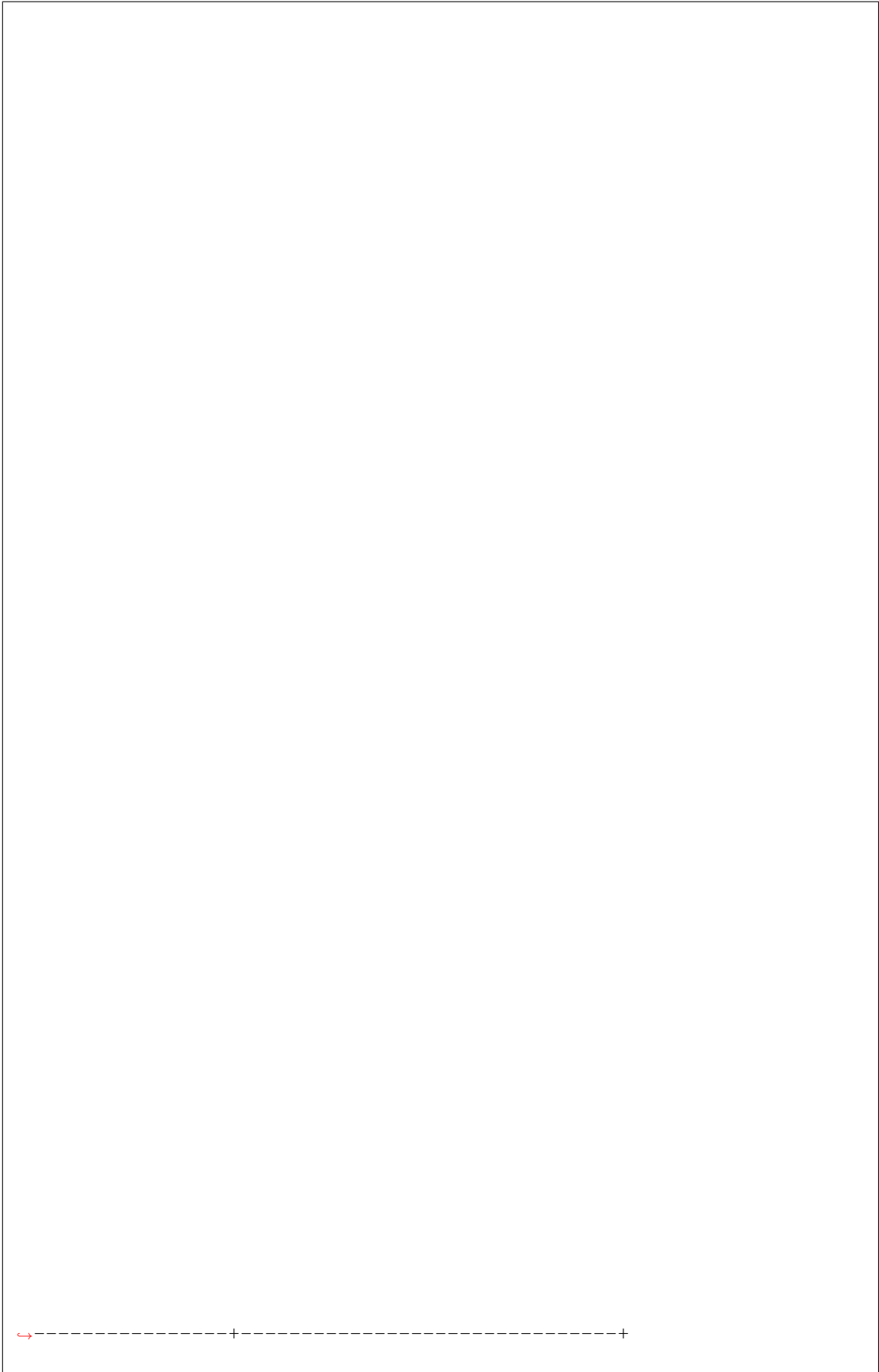


request will result in a No valid host was found error. It is hence sensible to check if Placement is aware of resource providers (nodes) for the requested resource class with:



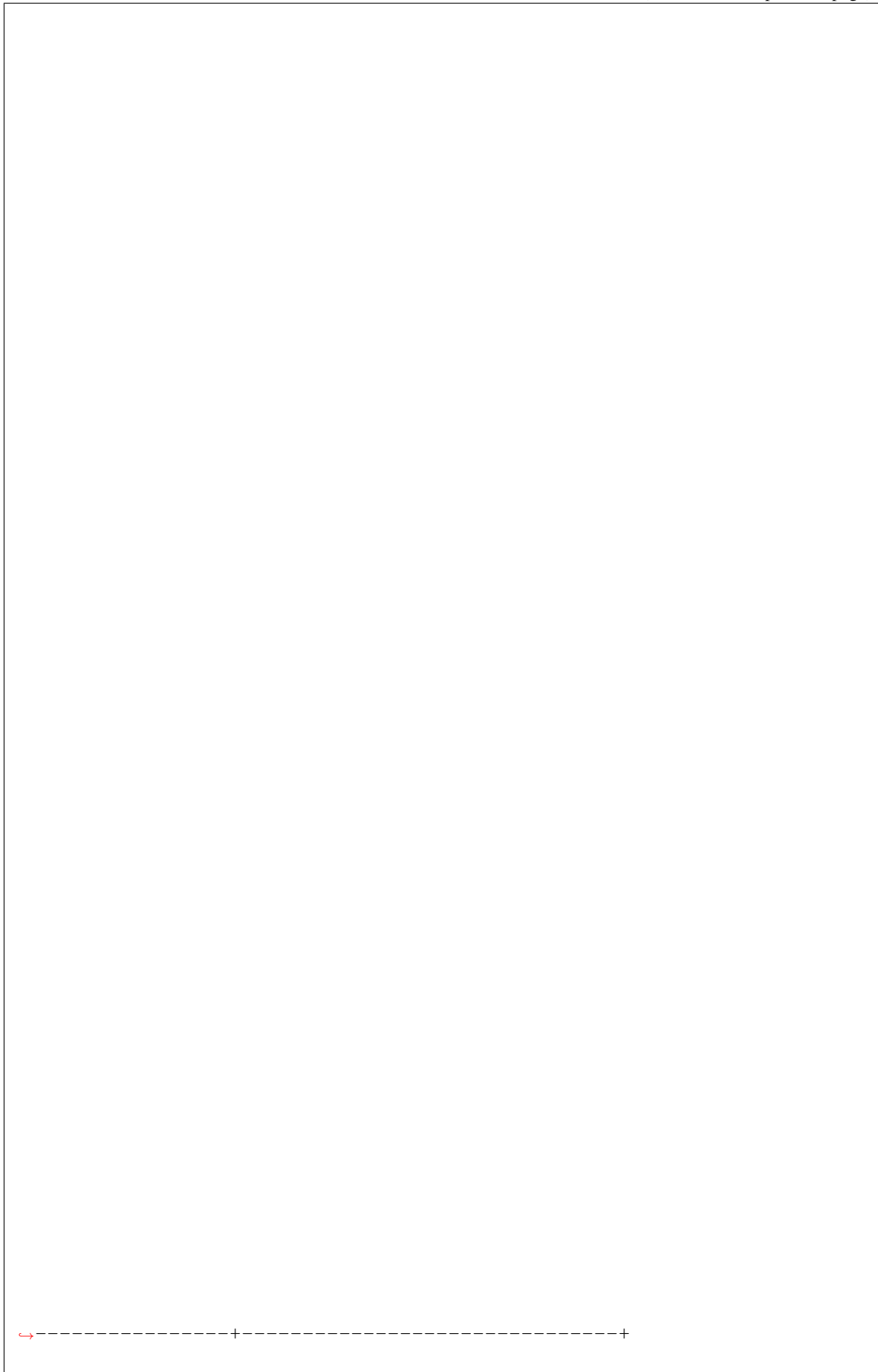
(continues on next page)

(continued from previous page)



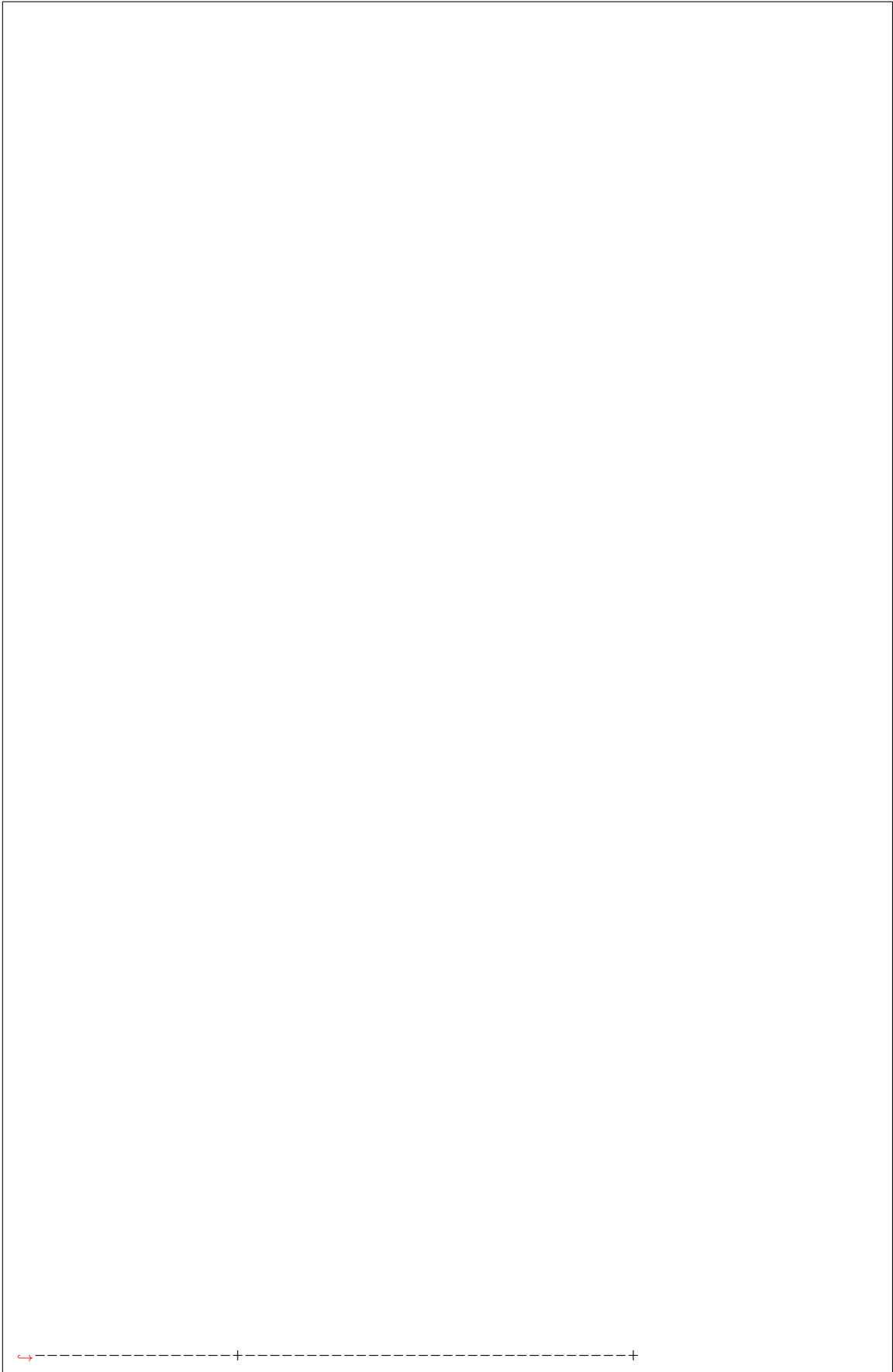
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)





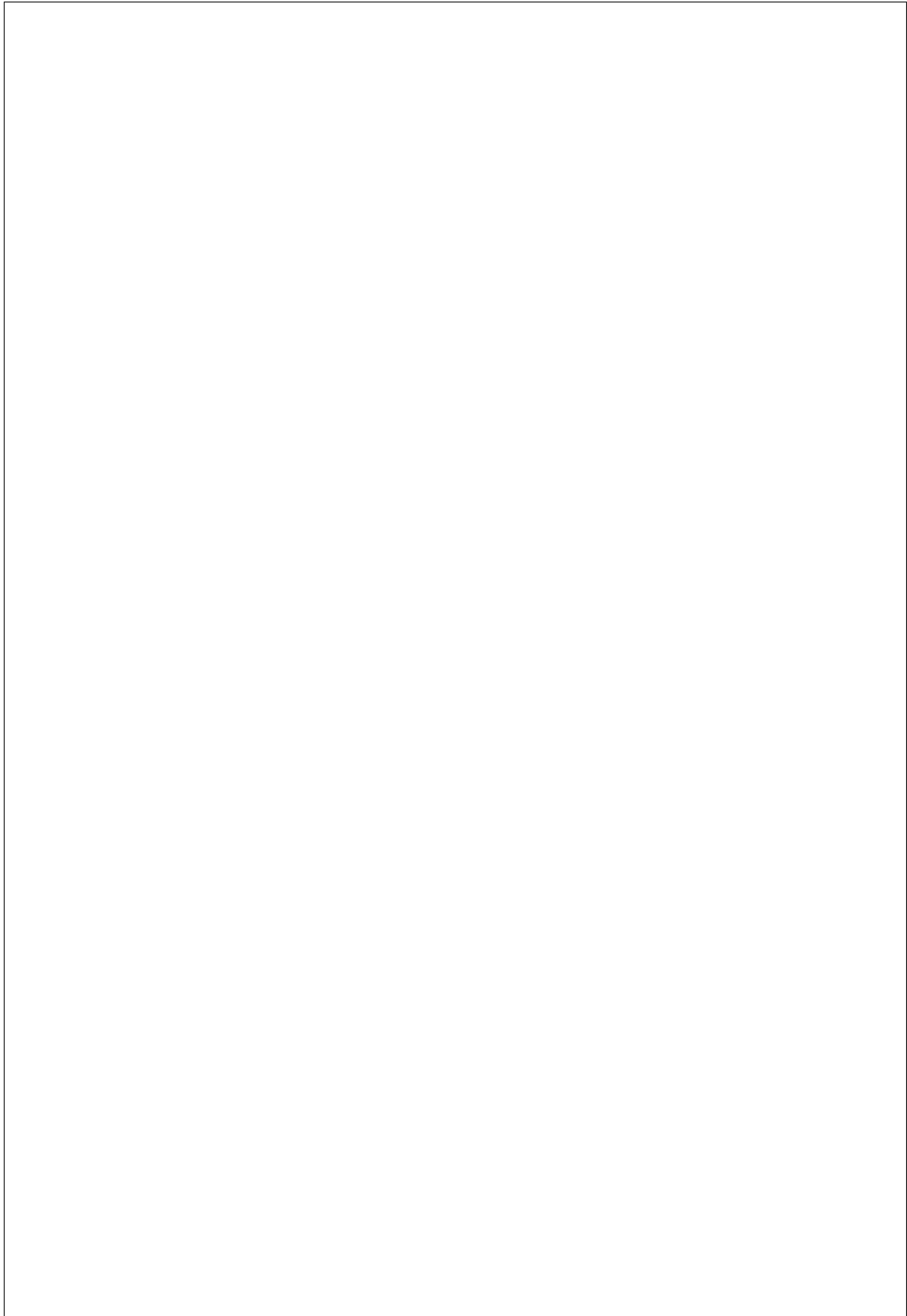
ported this provider to placement. Potential explanations include:





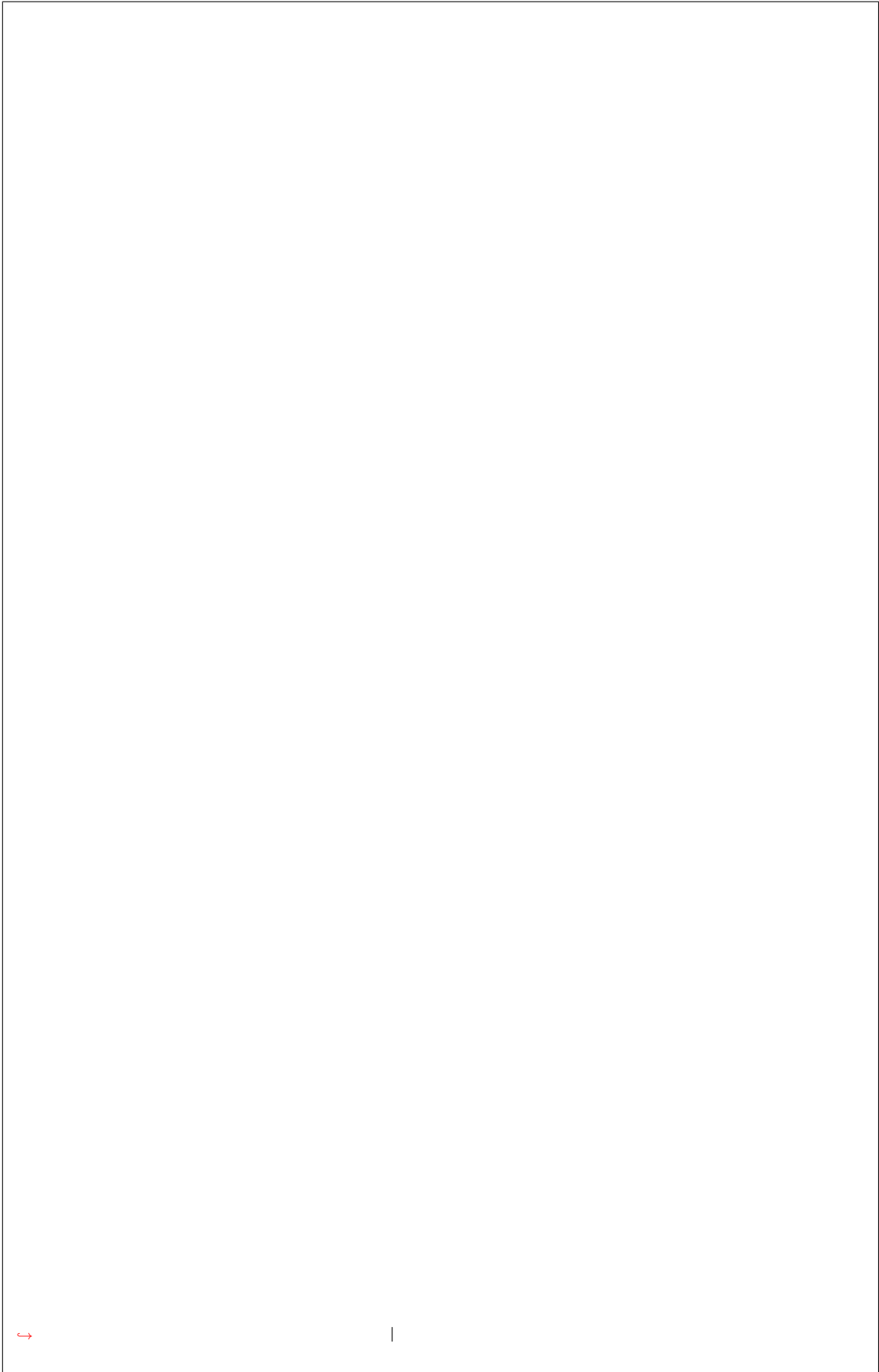


`memory_mb` and `local_gb`. Example of valid properties:



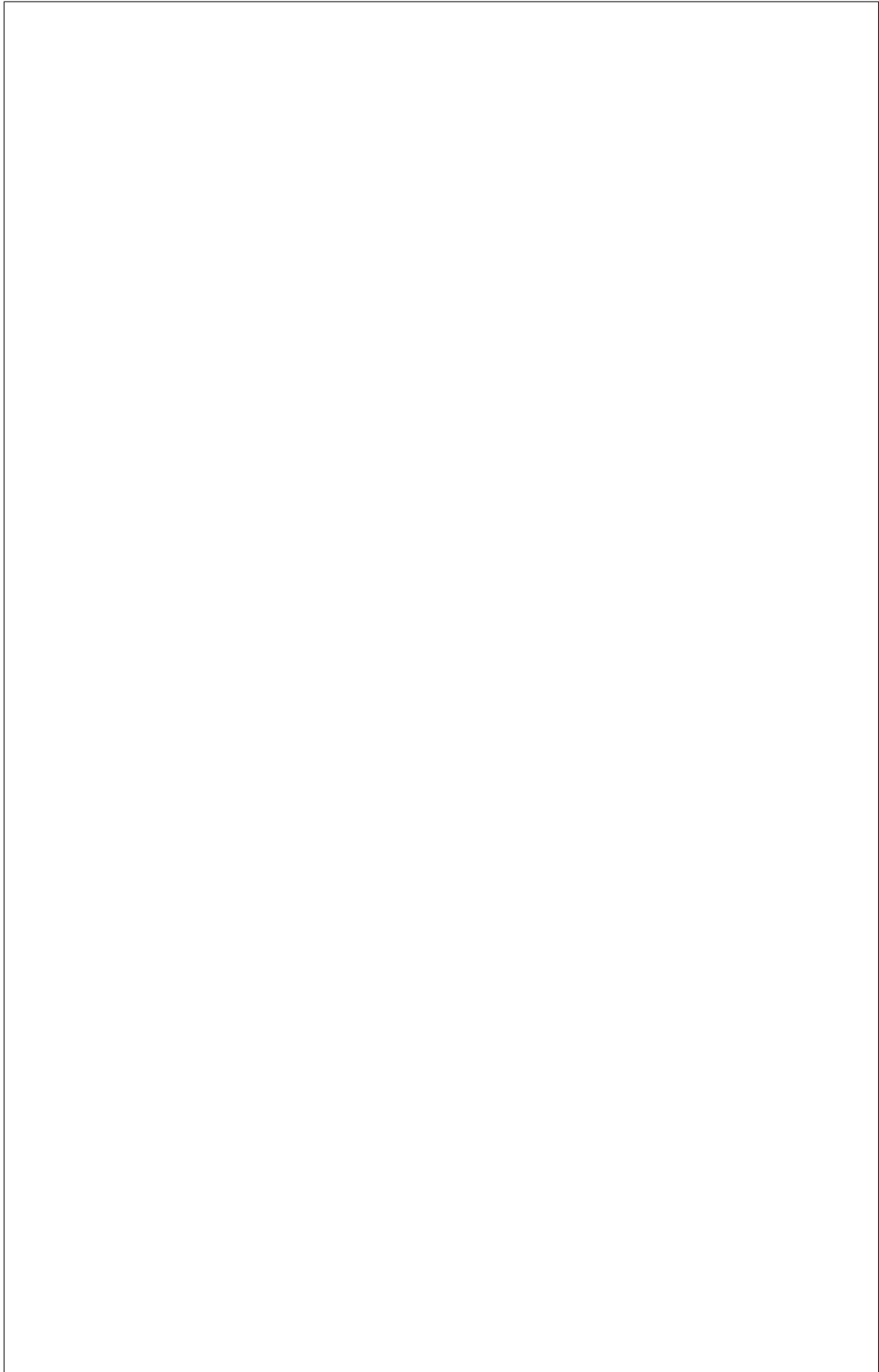
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



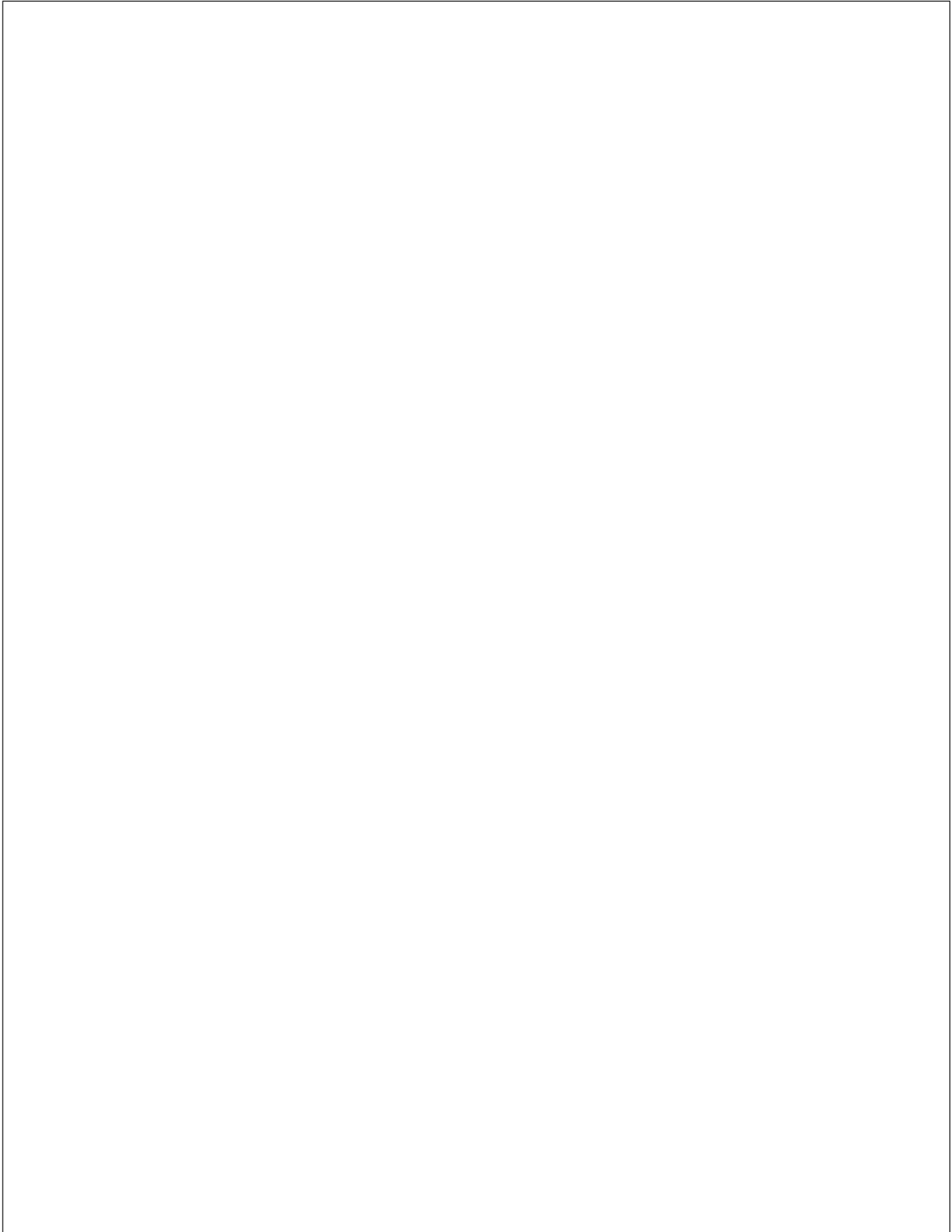
(continues on next page)

(continued from previous page)

---

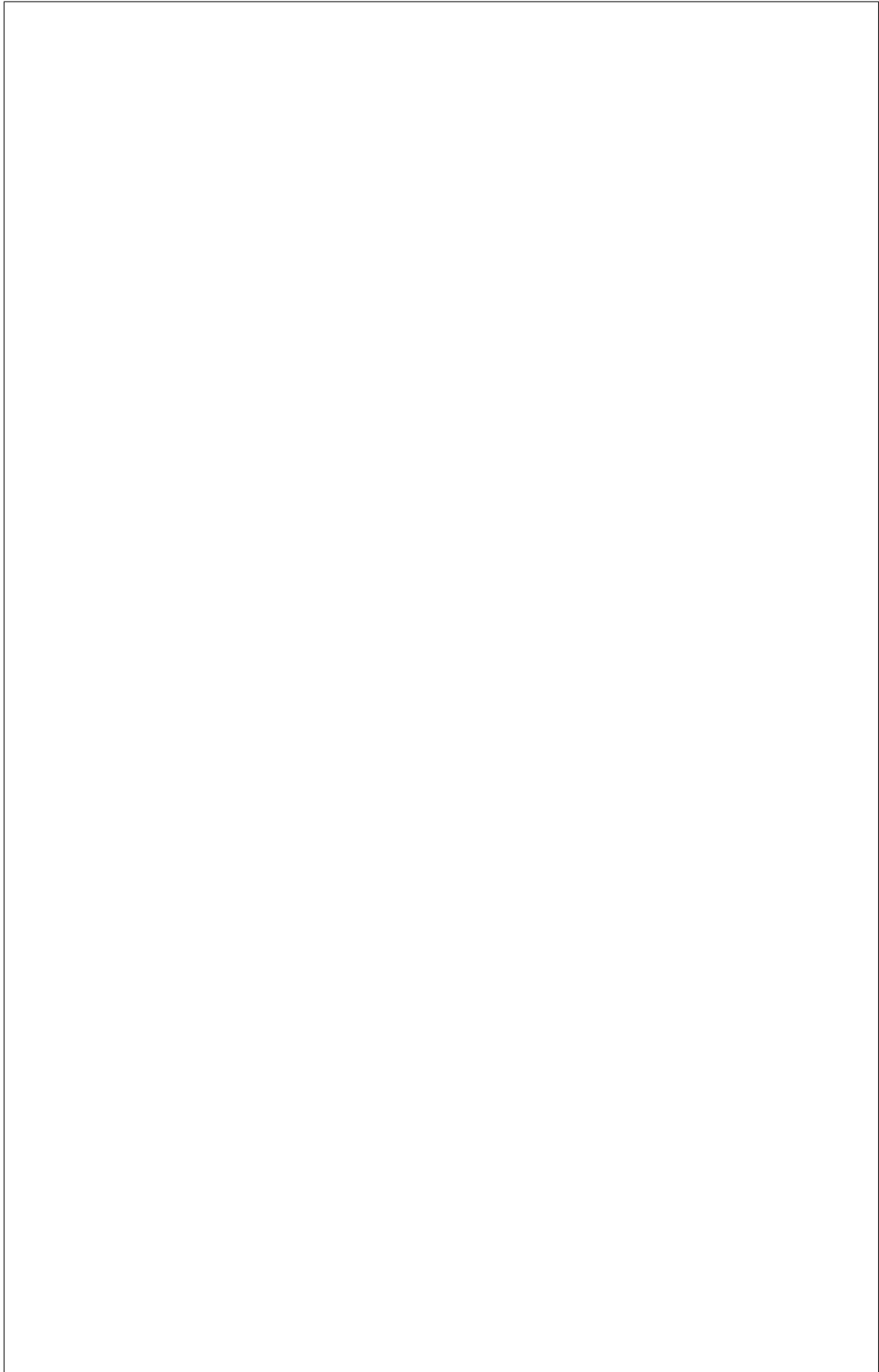






(continues on next page)

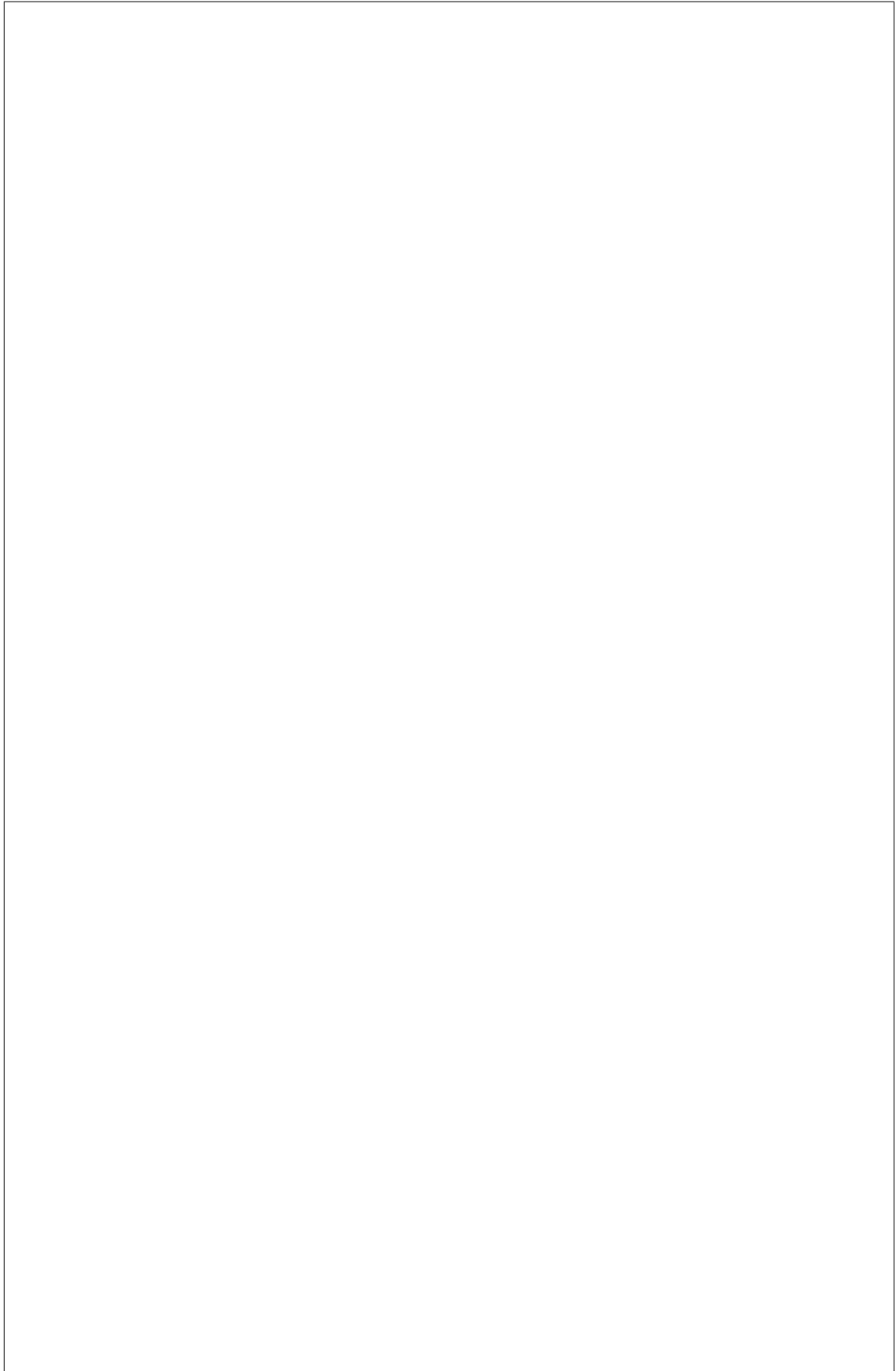
(continued from previous page)

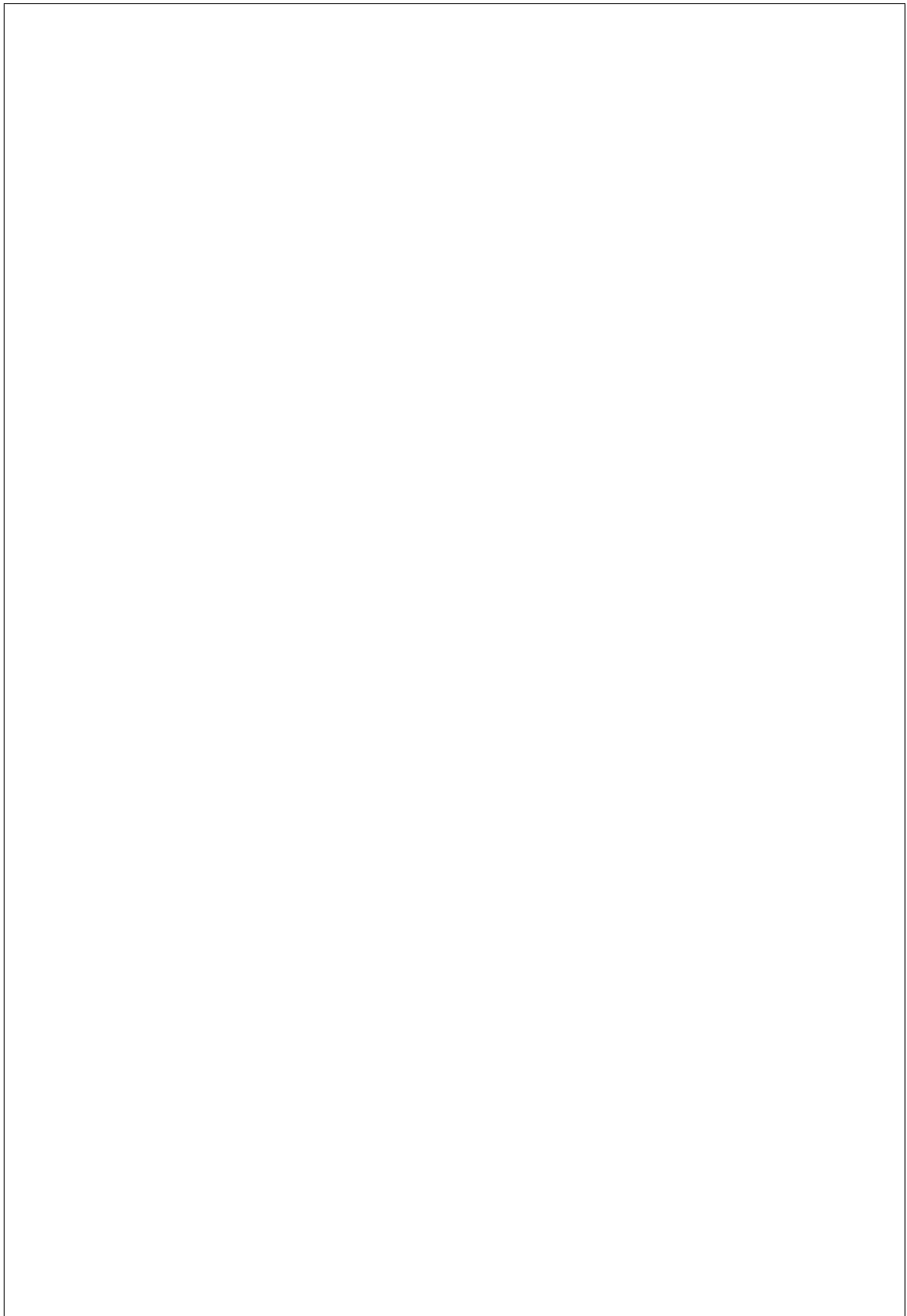


(continues on next page)



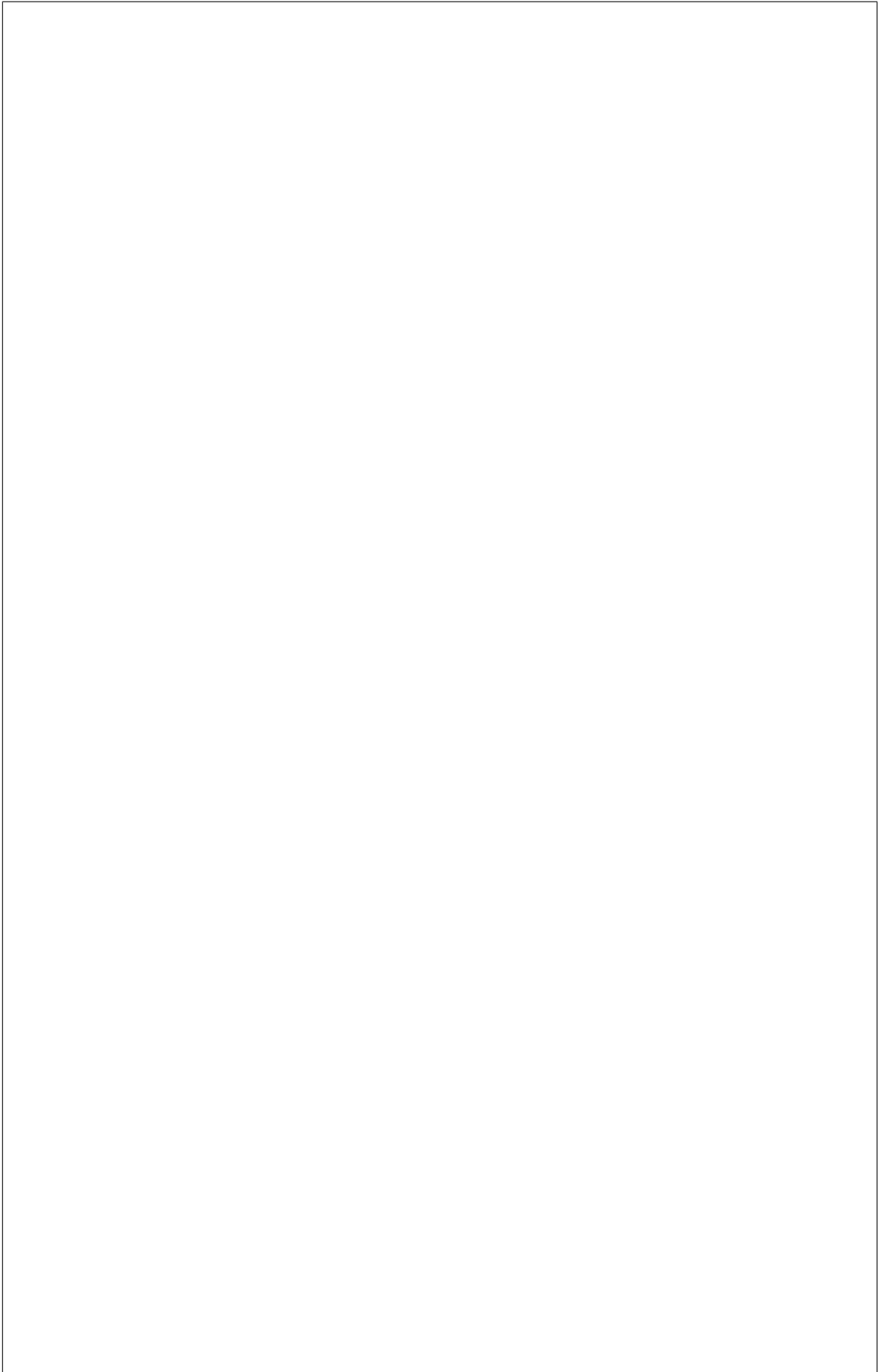
(continued from previous page)





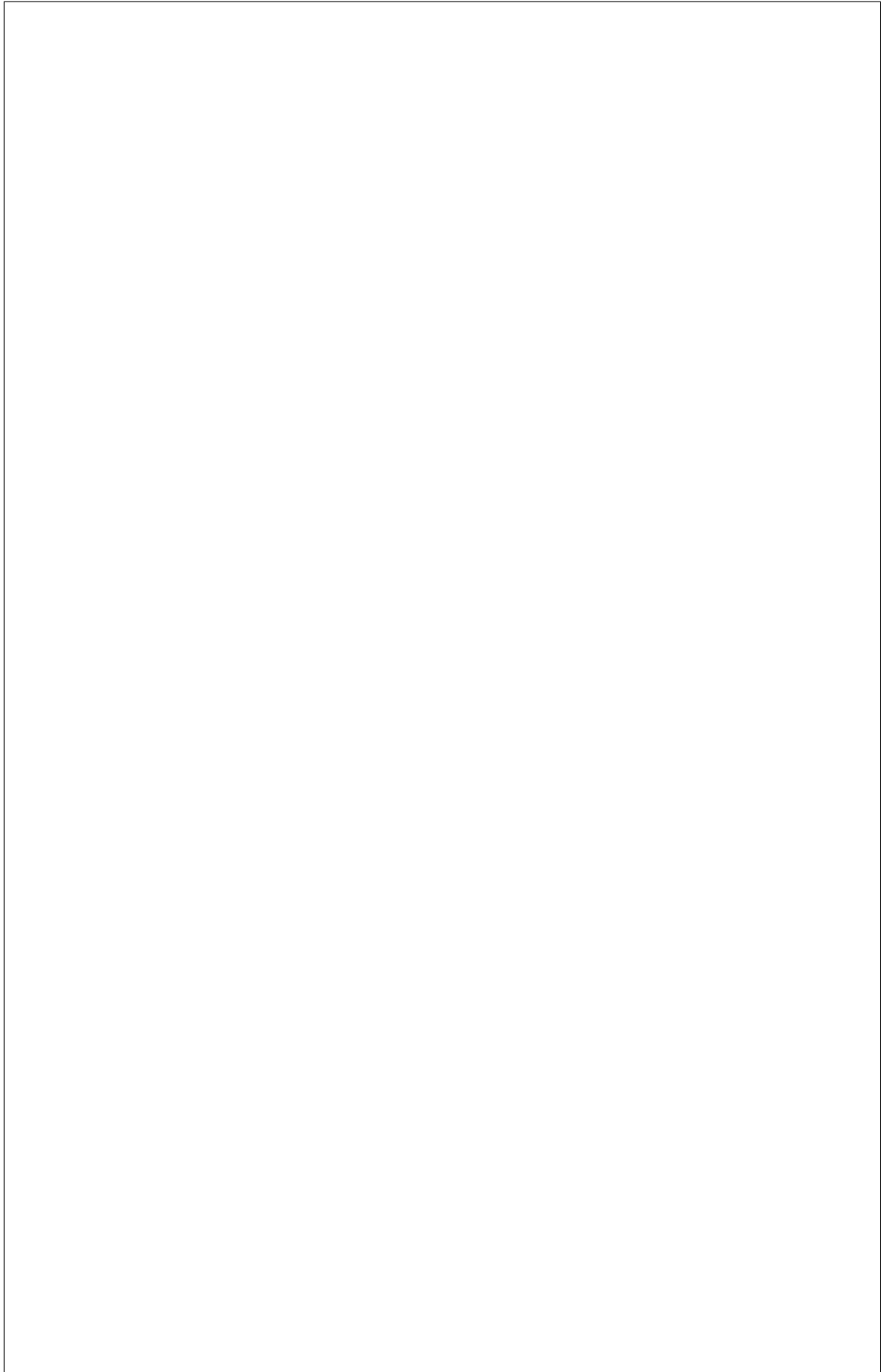
(continues on next page)

(continued from previous page)



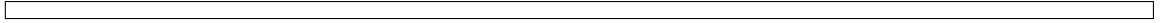
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)





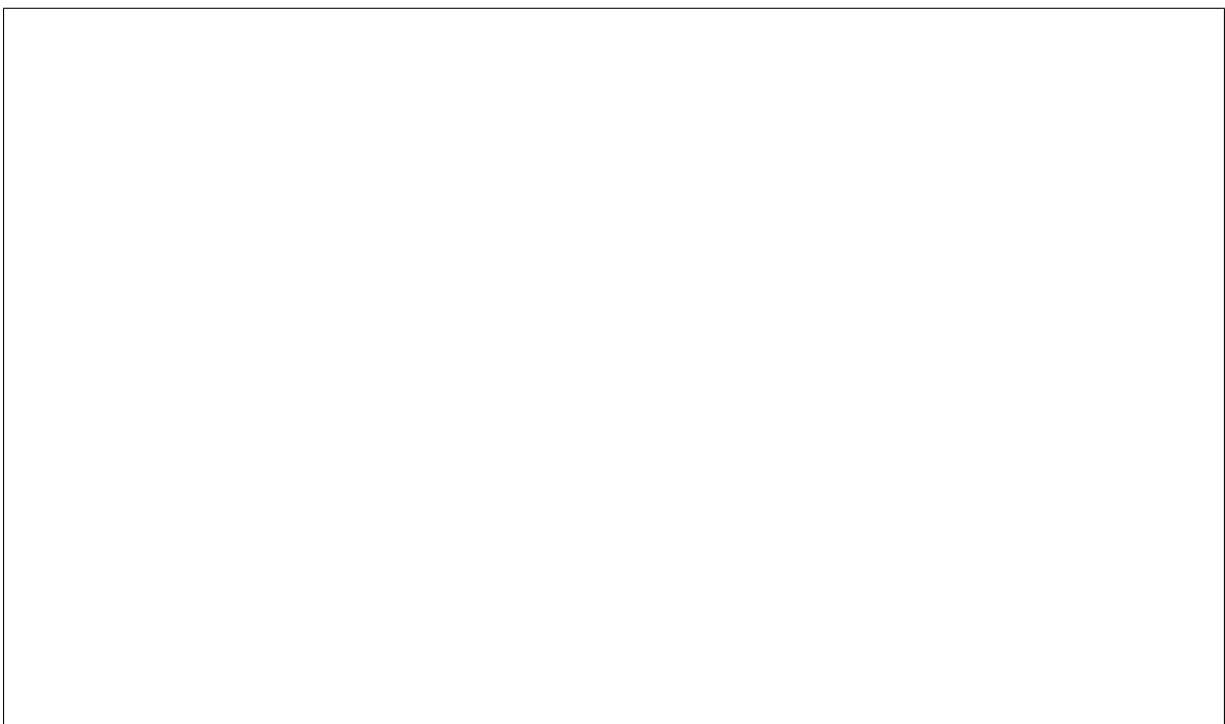


ror messages in Ironic conductor log, it means the conductor run into a special error during deployment. So you can check the log carefully to fix or work around and then try again.



## **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)

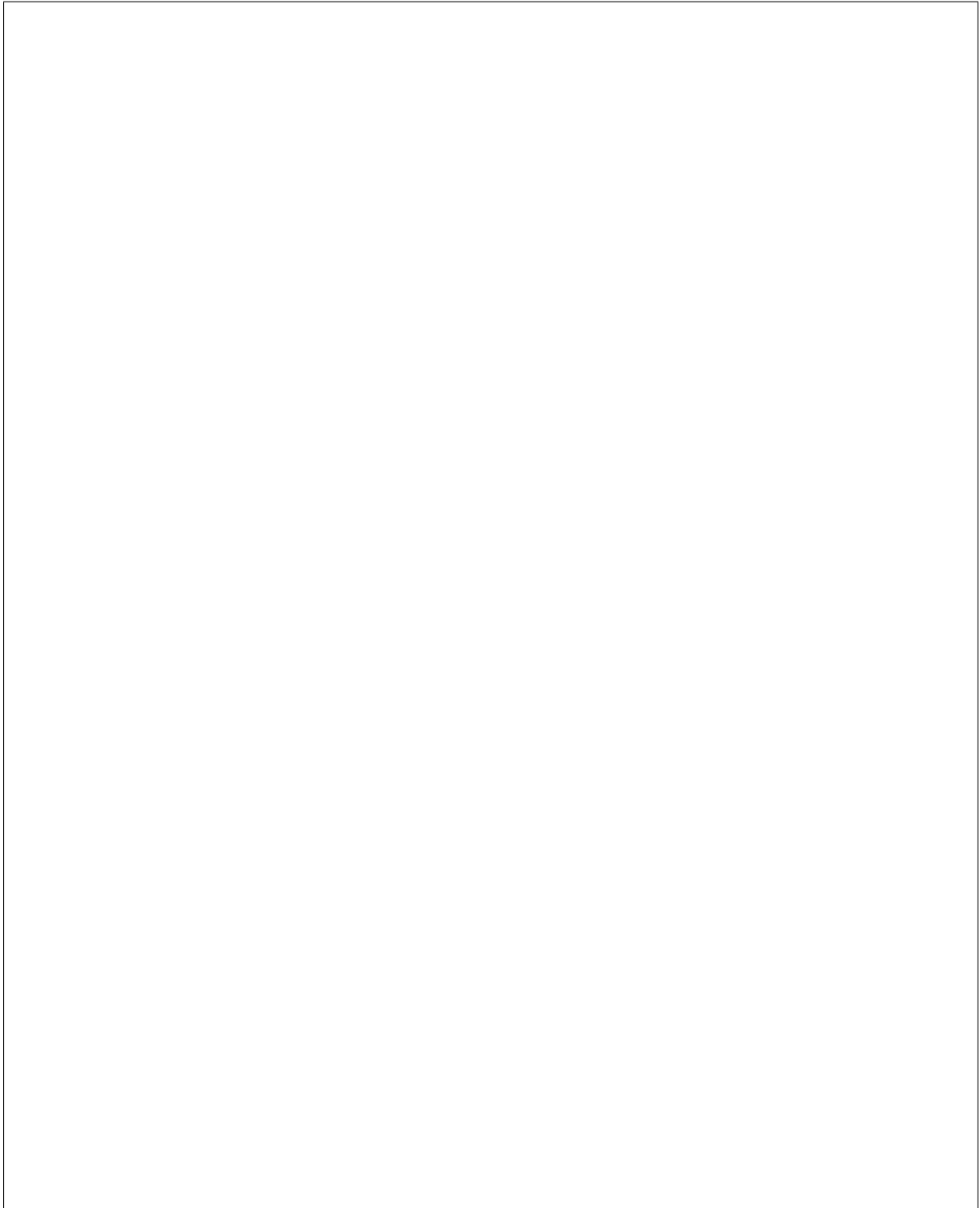
(continued from previous page)

---

**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**

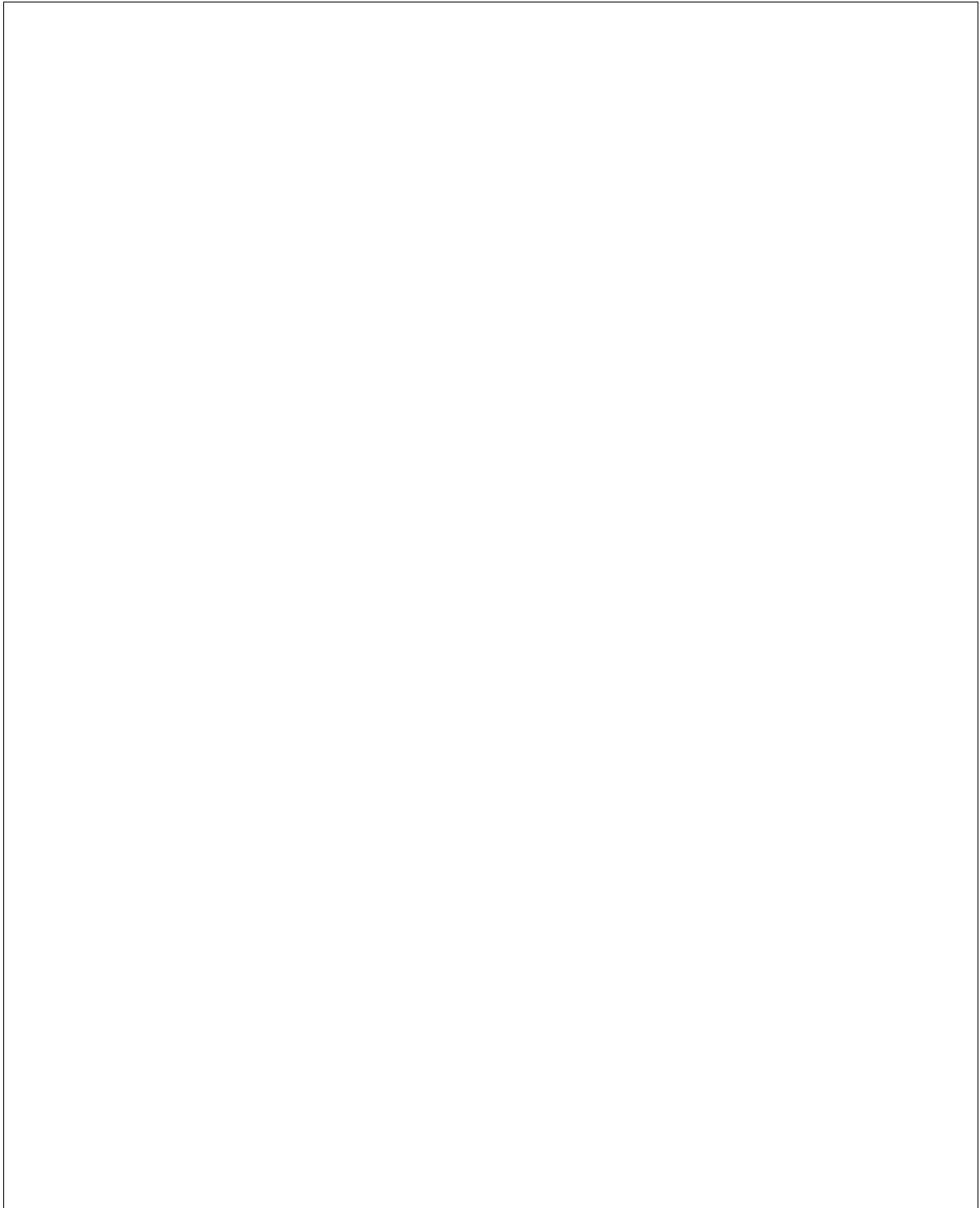
### **When storing in the local filesystem**

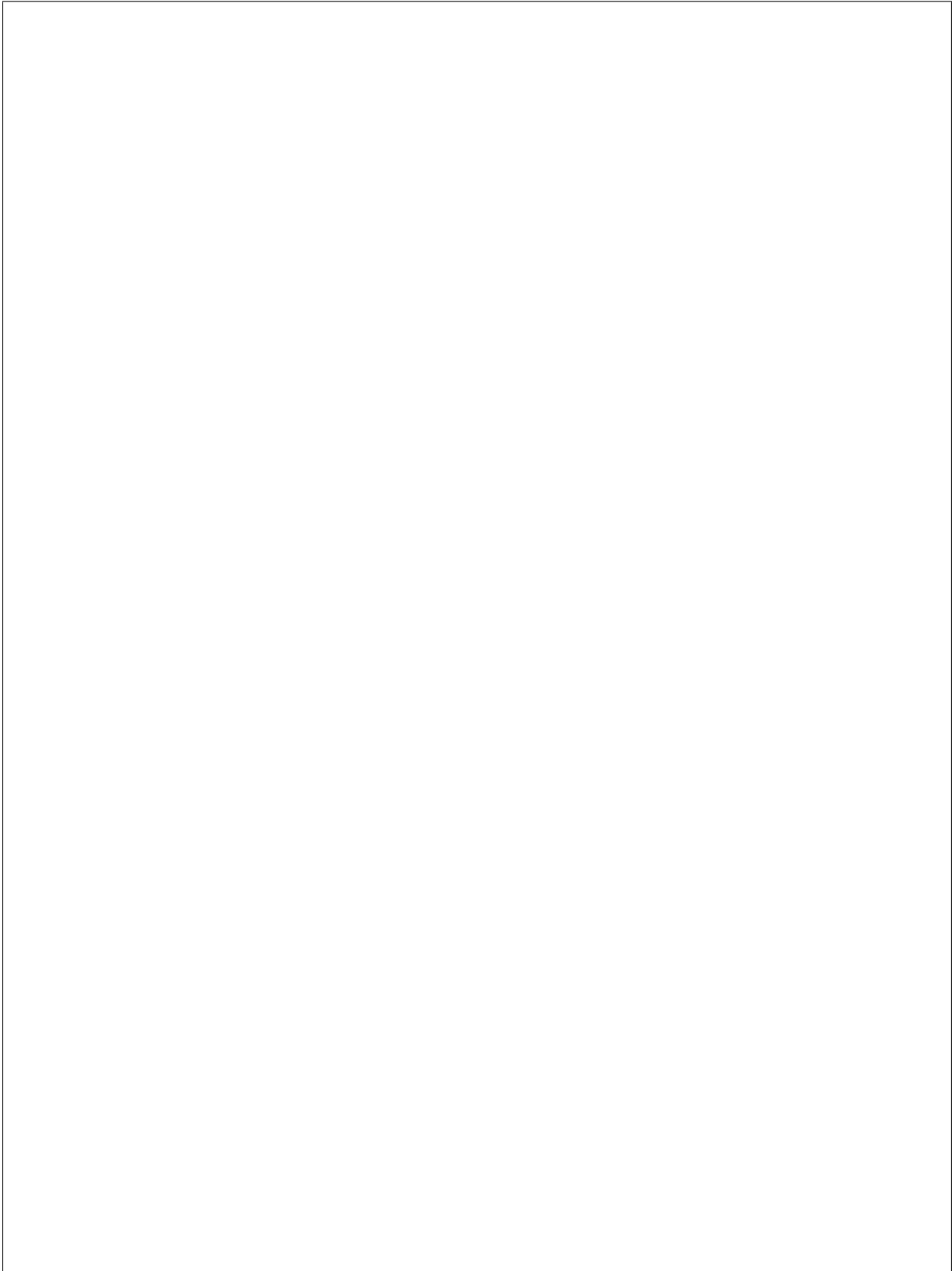


**Note:** When saving the logs to the filesystem, operators may want to enable some form of rotation for the logs to avoid disk space problems.

### **When storing in Swift**







(continues on next page)

(continued from previous page)

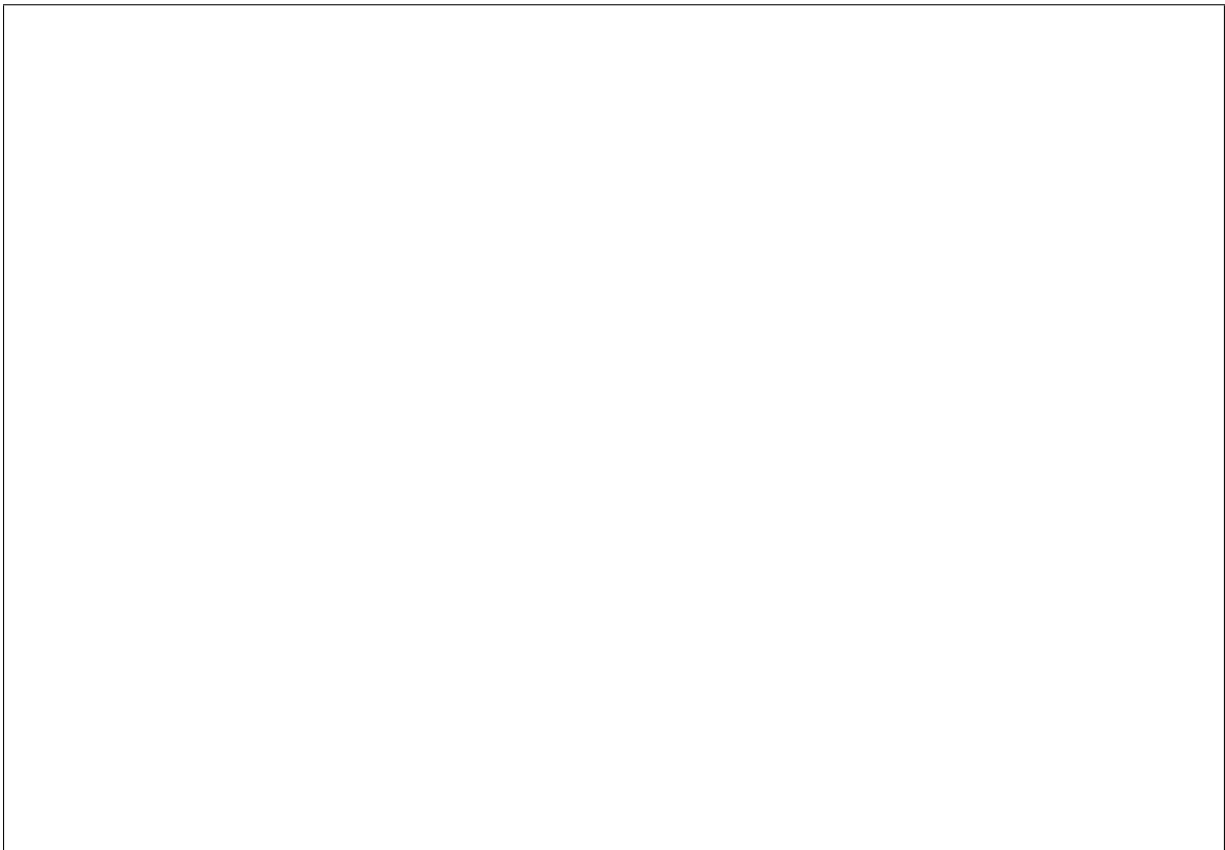
---

### **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.

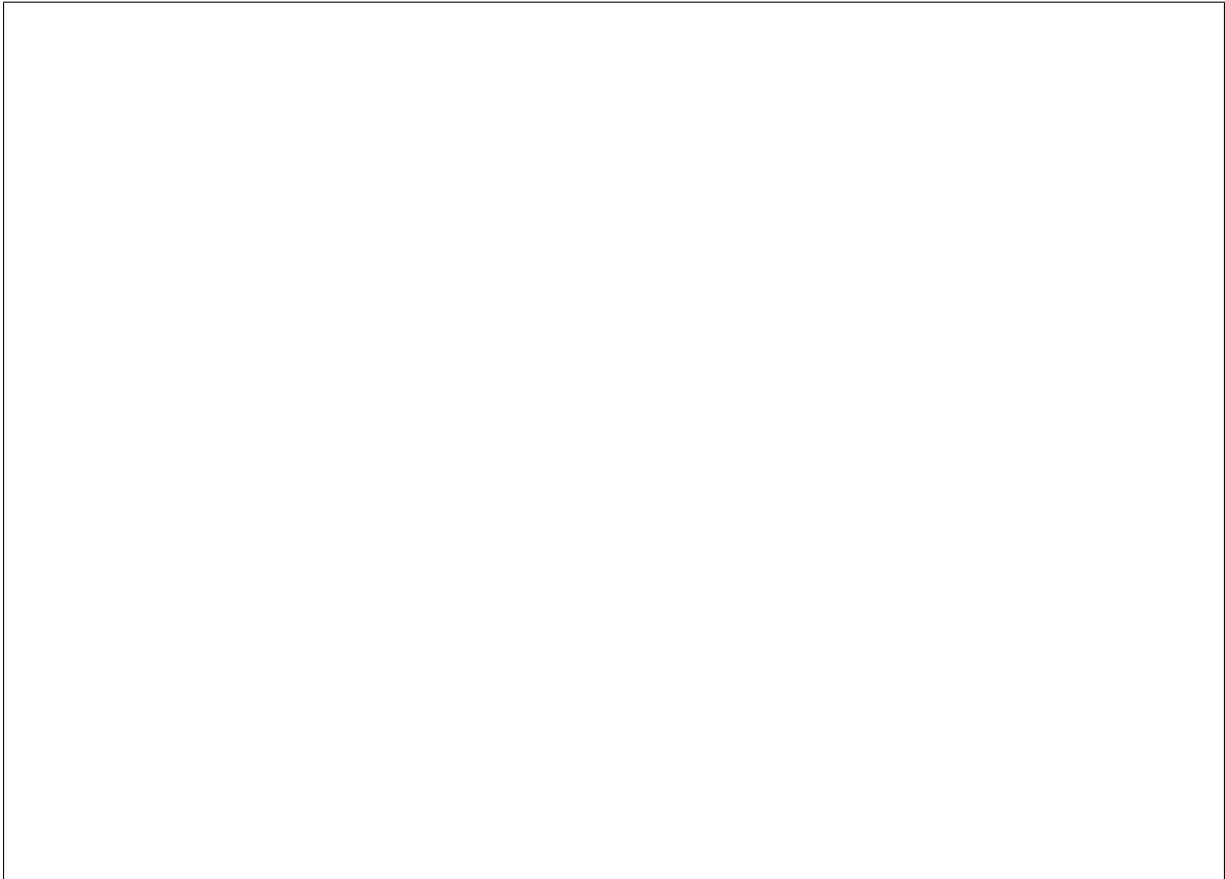
attempt to navigate the worst-known default hold-down timers to help ensure a deployment does not fail

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.

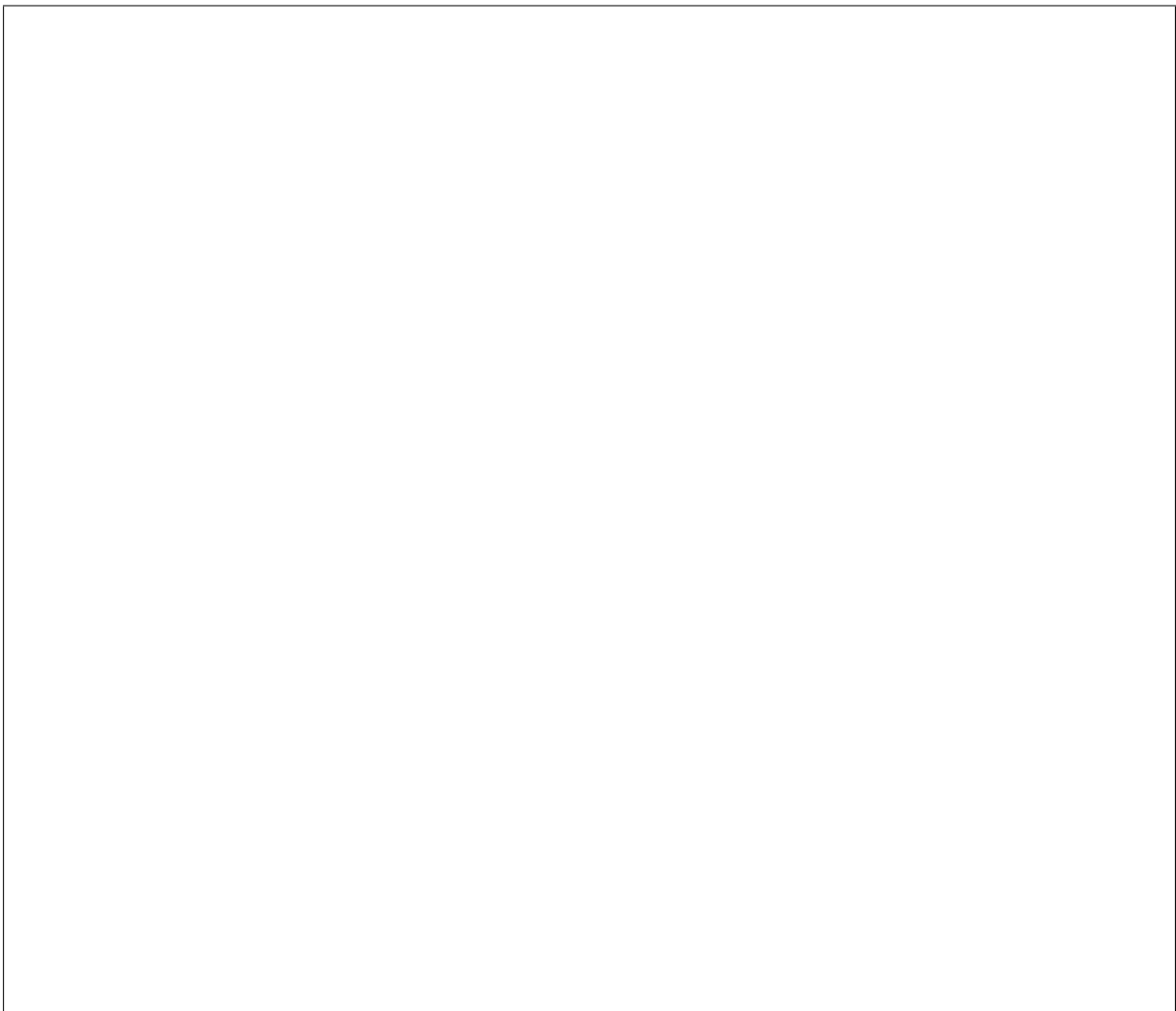
## **IPMI errors**

### **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:** IroniC 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?**

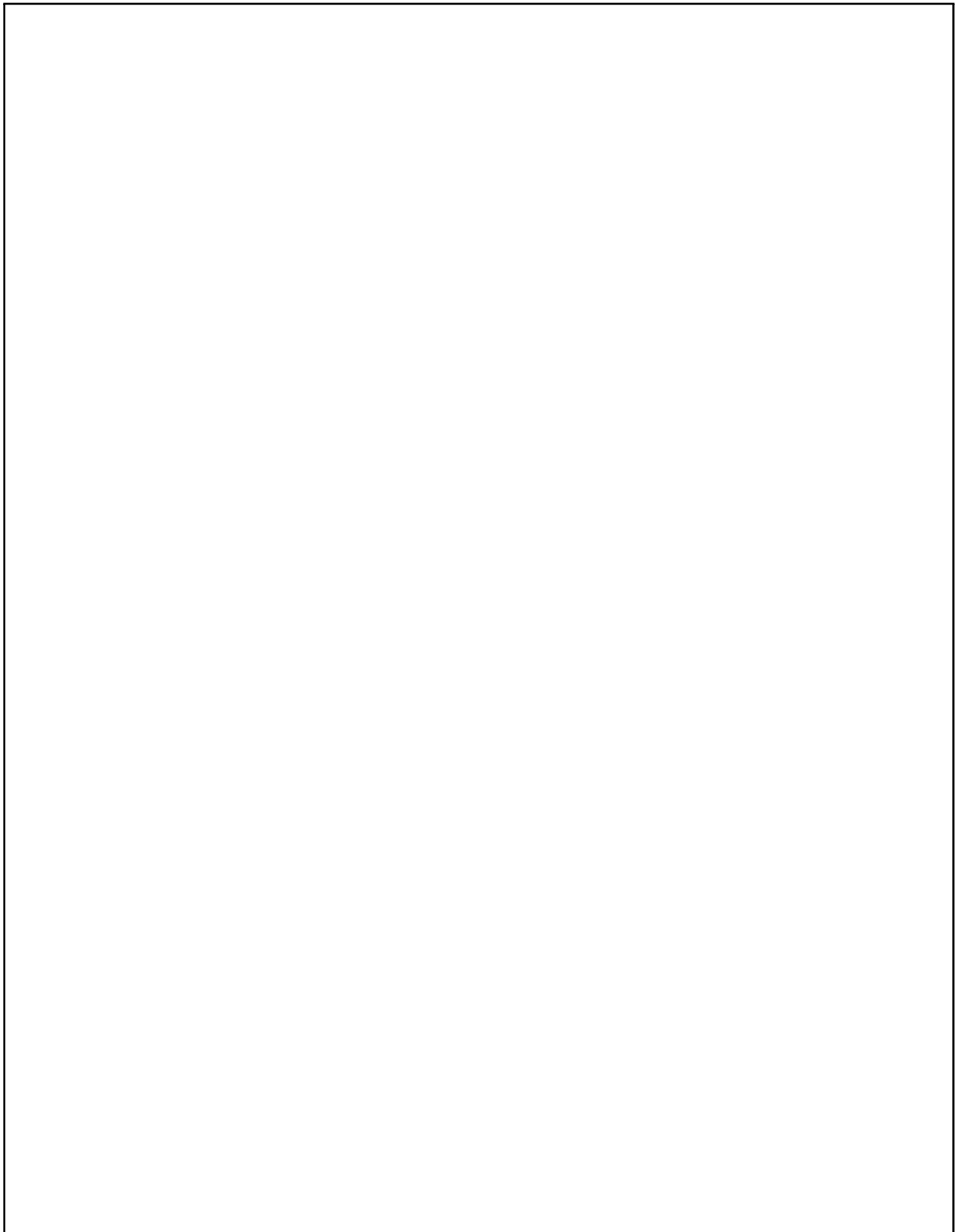


in the Bare Metal service.

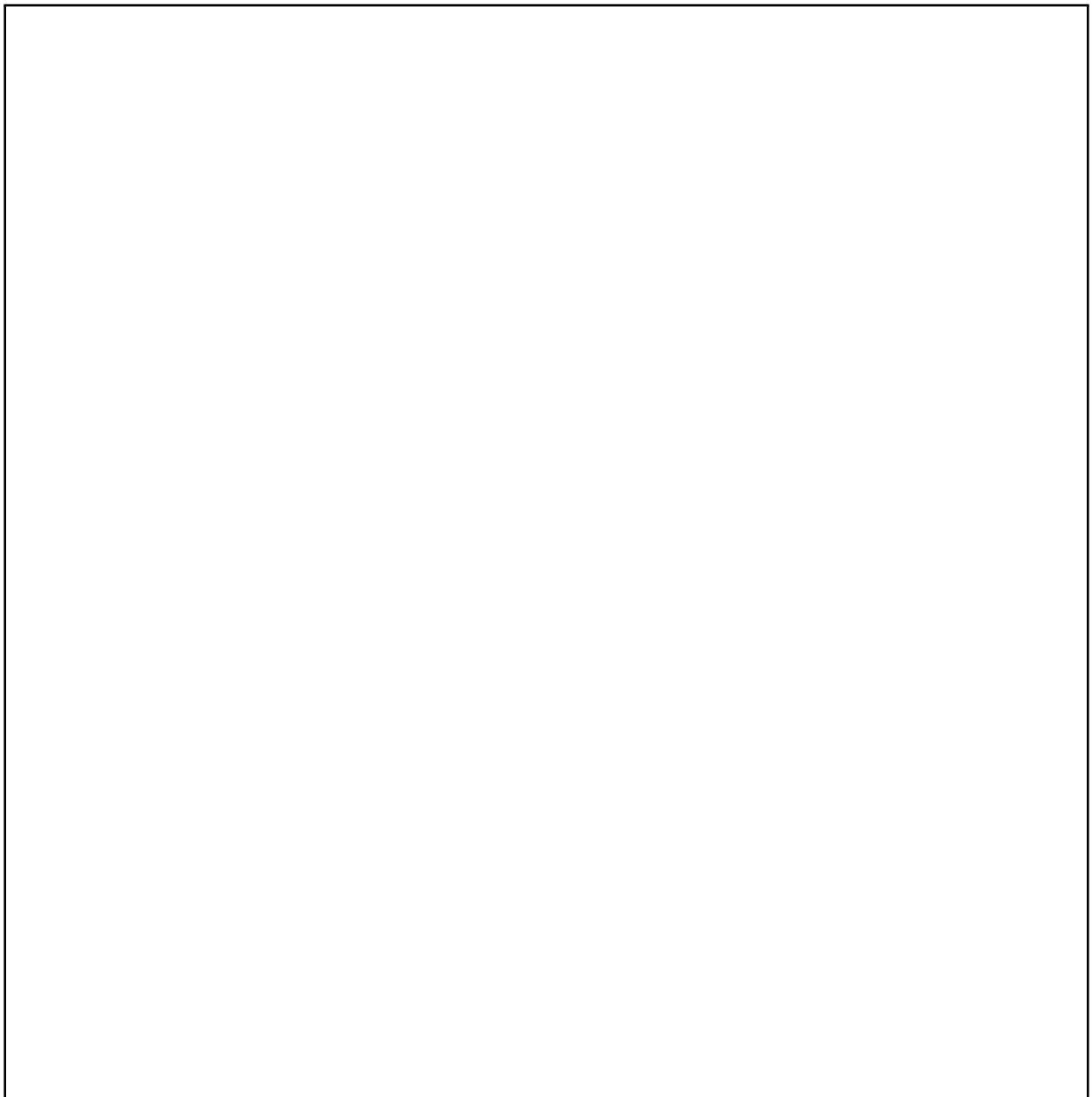
sult in the VIF not being cleaned up from Neutron.

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

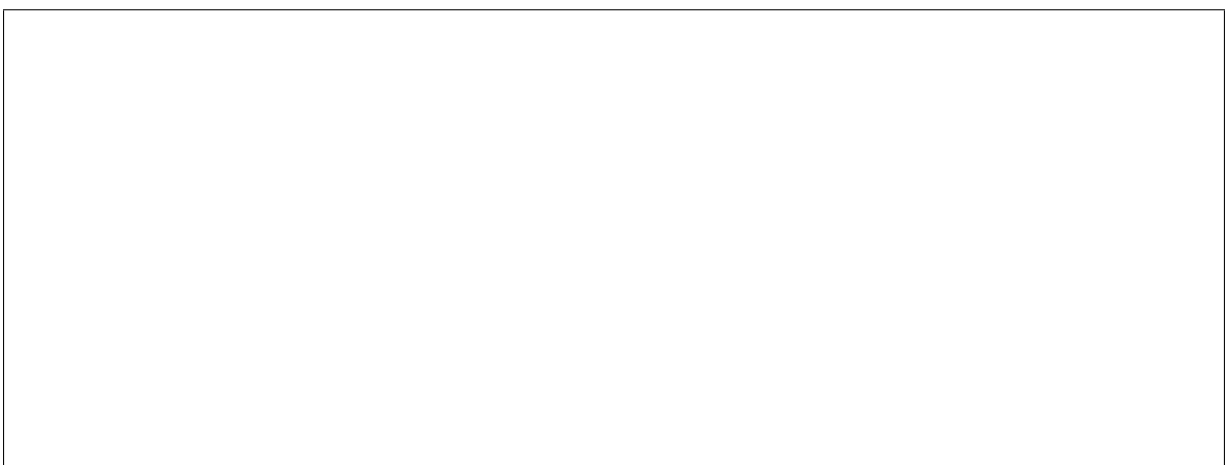
unexpected internal structure.

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?**

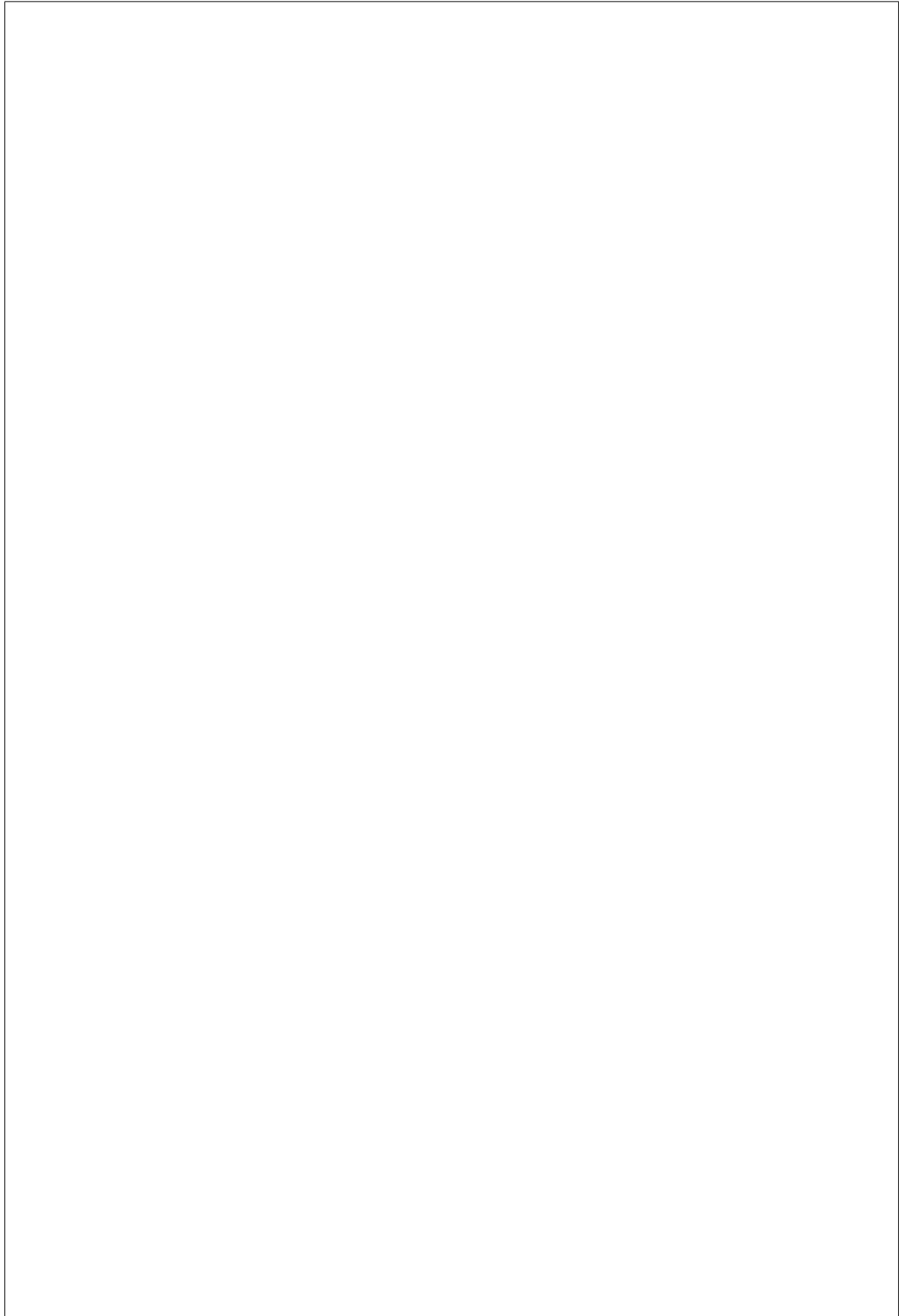
of these published cloud images, also support auto-configuration of networking AND population of user keys.

### **Issues with autoconfigured TLS**



(continues on next page)

(continued from previous page)



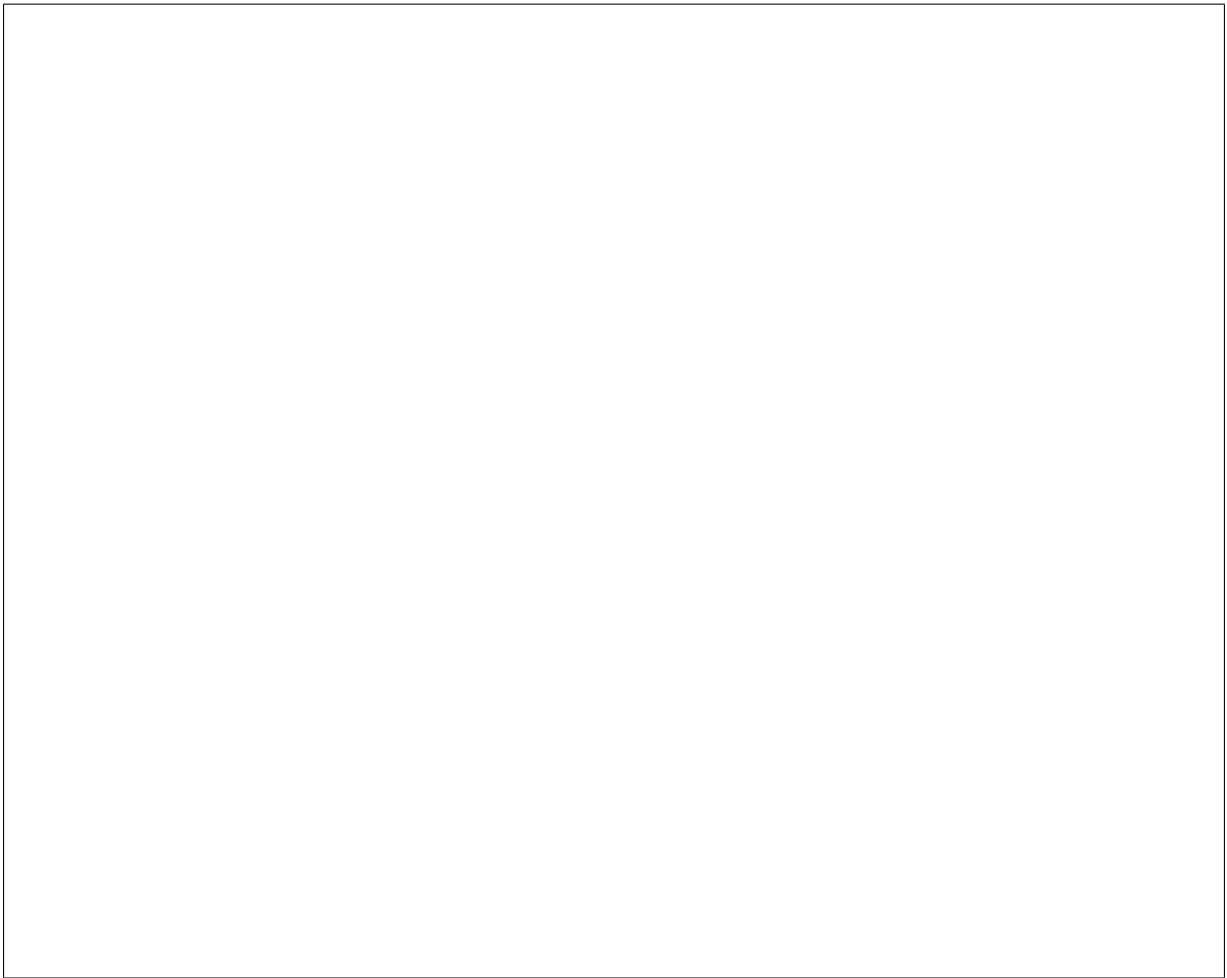
(continues on next page)

(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.

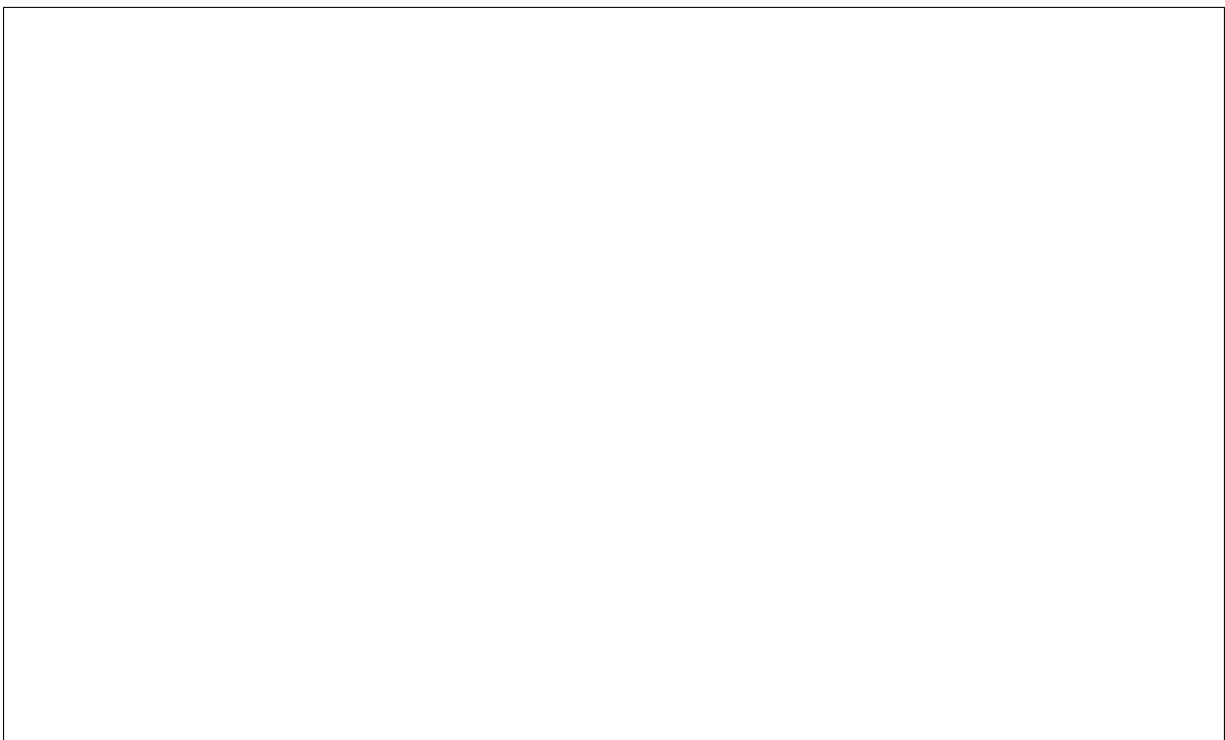
**Note:** Although an exclusive lock is used when sending notifications to the compute service, there can still be a race condition if the compute-baremetal power sync happens to happen a nano-second before the power state change event is received from the baremetal service in which case the power state from compute services database will be forced on the node.



## **Setting the Owner and Lessee**

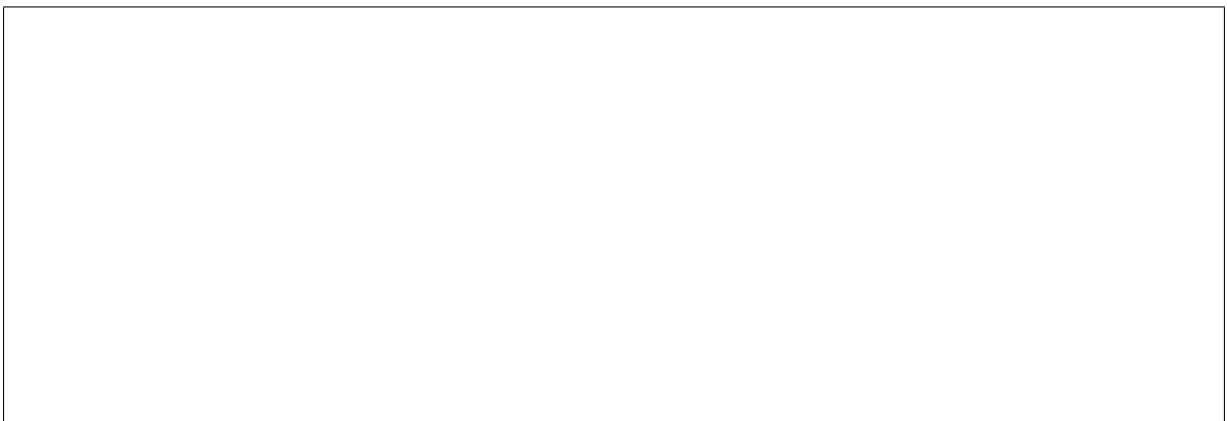
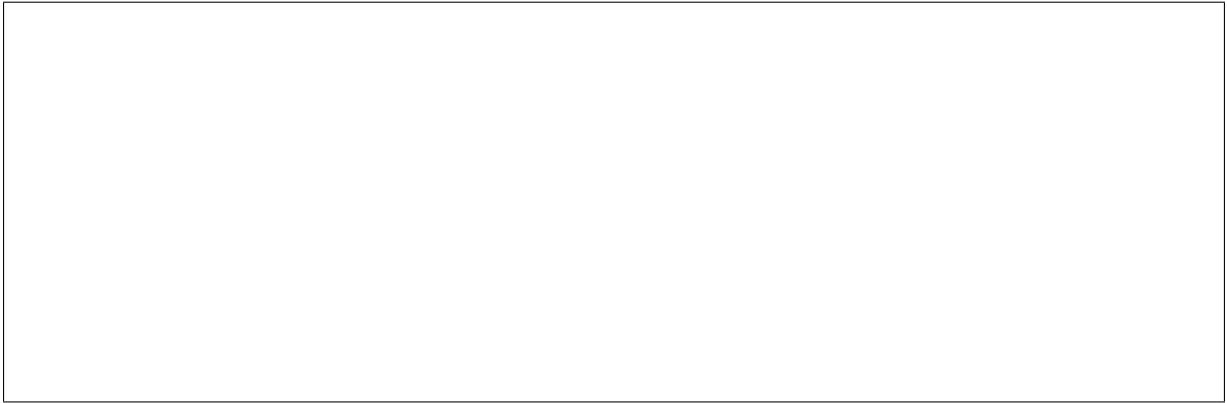


## **Configuring the Bare Metal Service Policy**



(continues on next page)

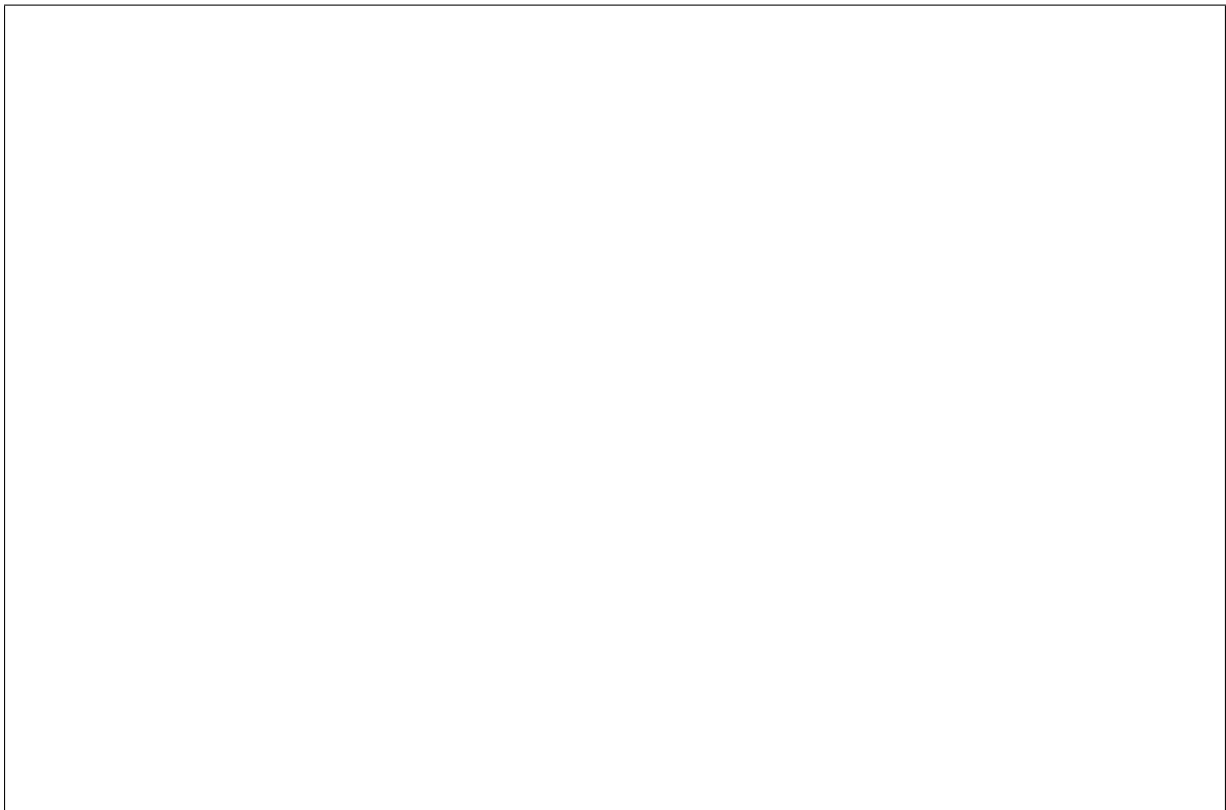
(continued from previous page)



(continues on next page)

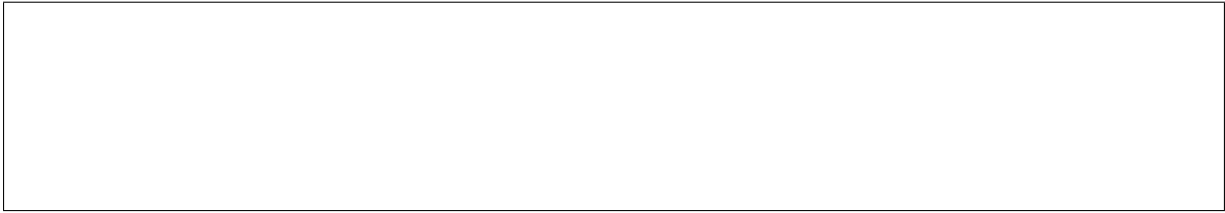
(continued from previous page)



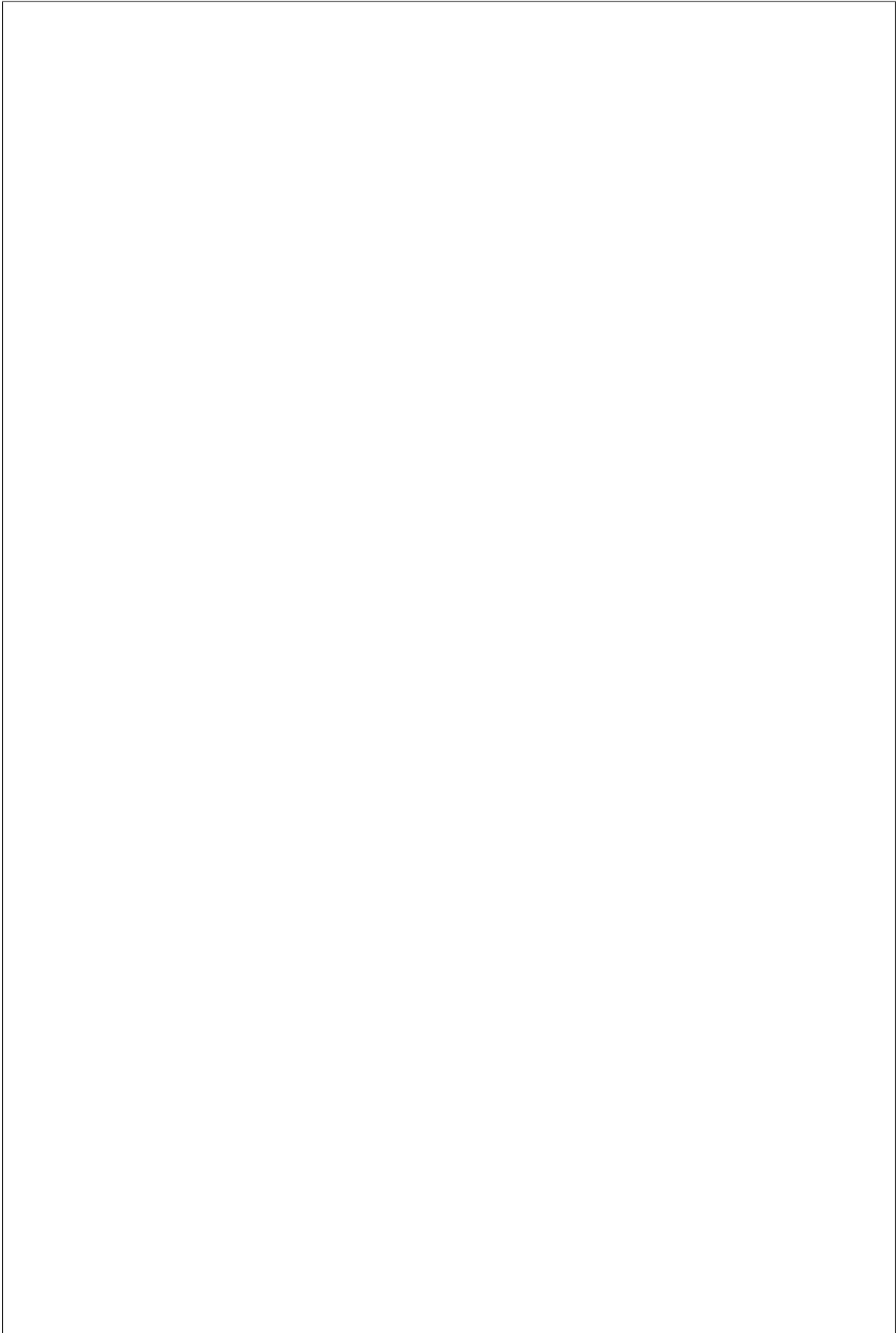


(continues on next page)

(continued from previous page)



## **Ports**



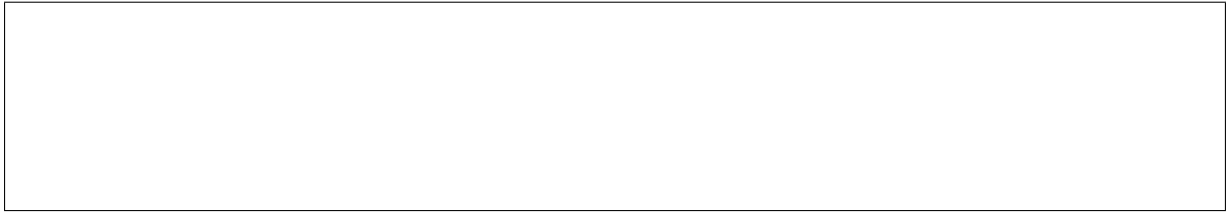
(continues on next page)

(continued from previous page)



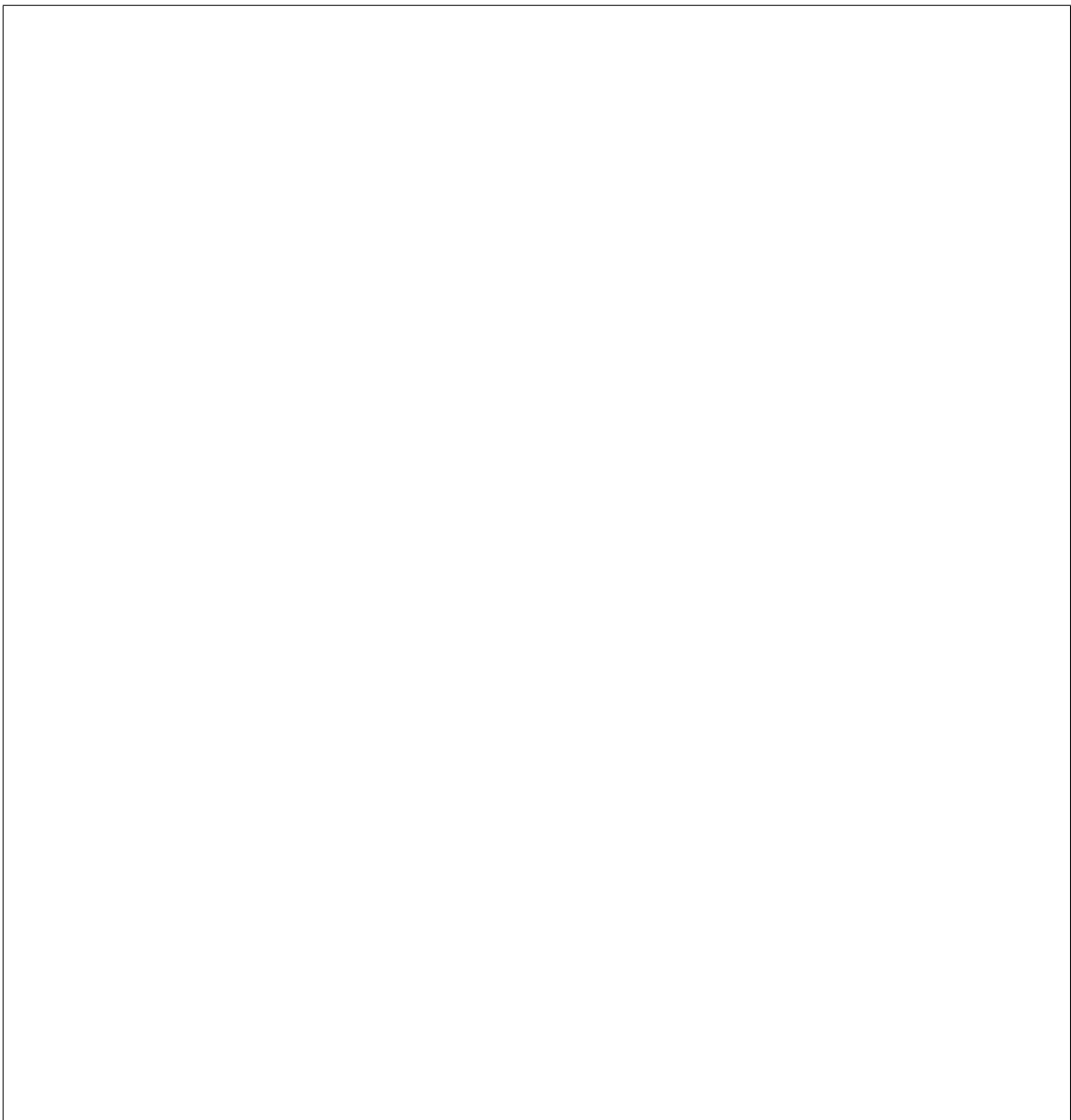
(continues on next page)

(continued from previous page)



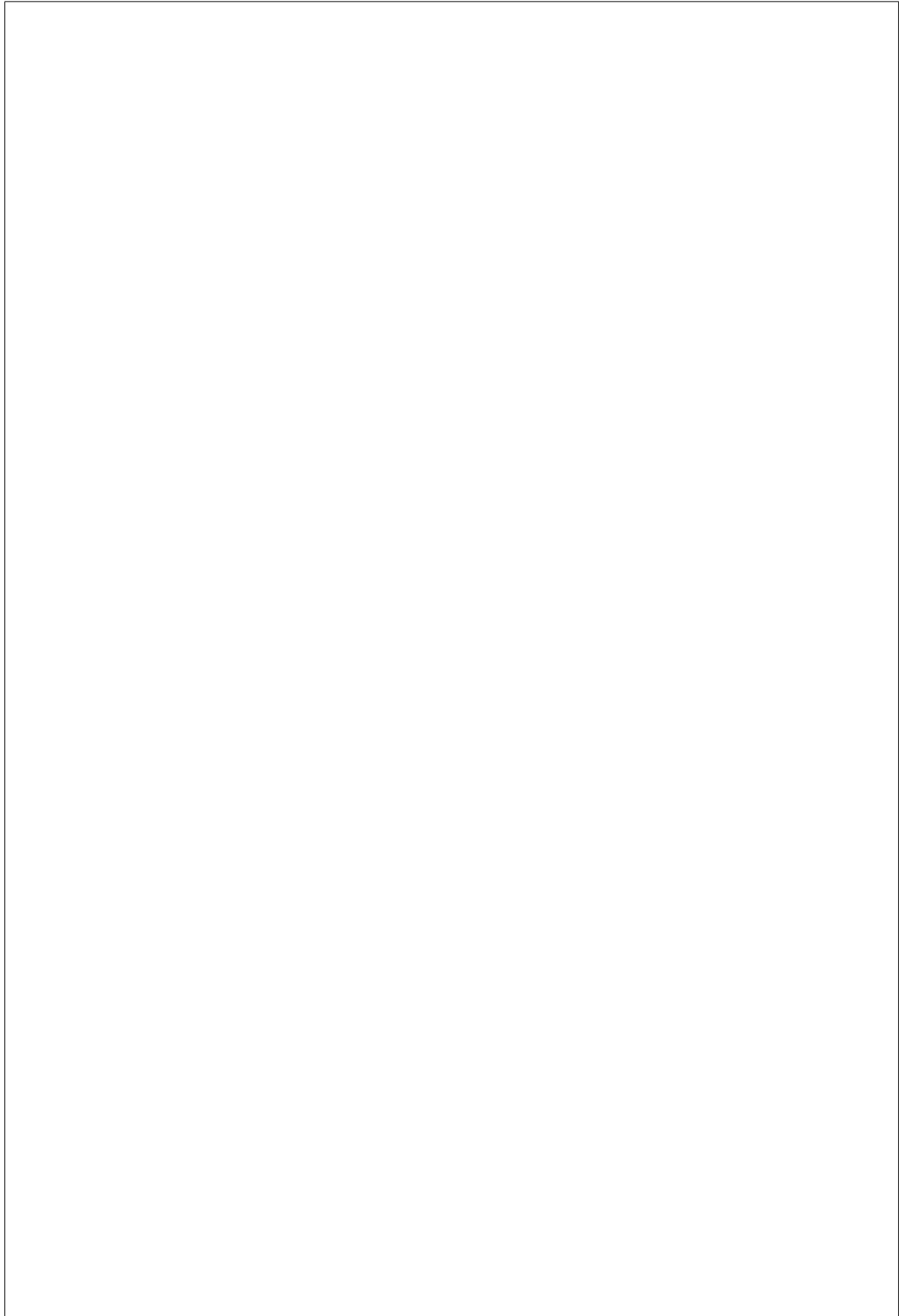
## **Allocations**

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



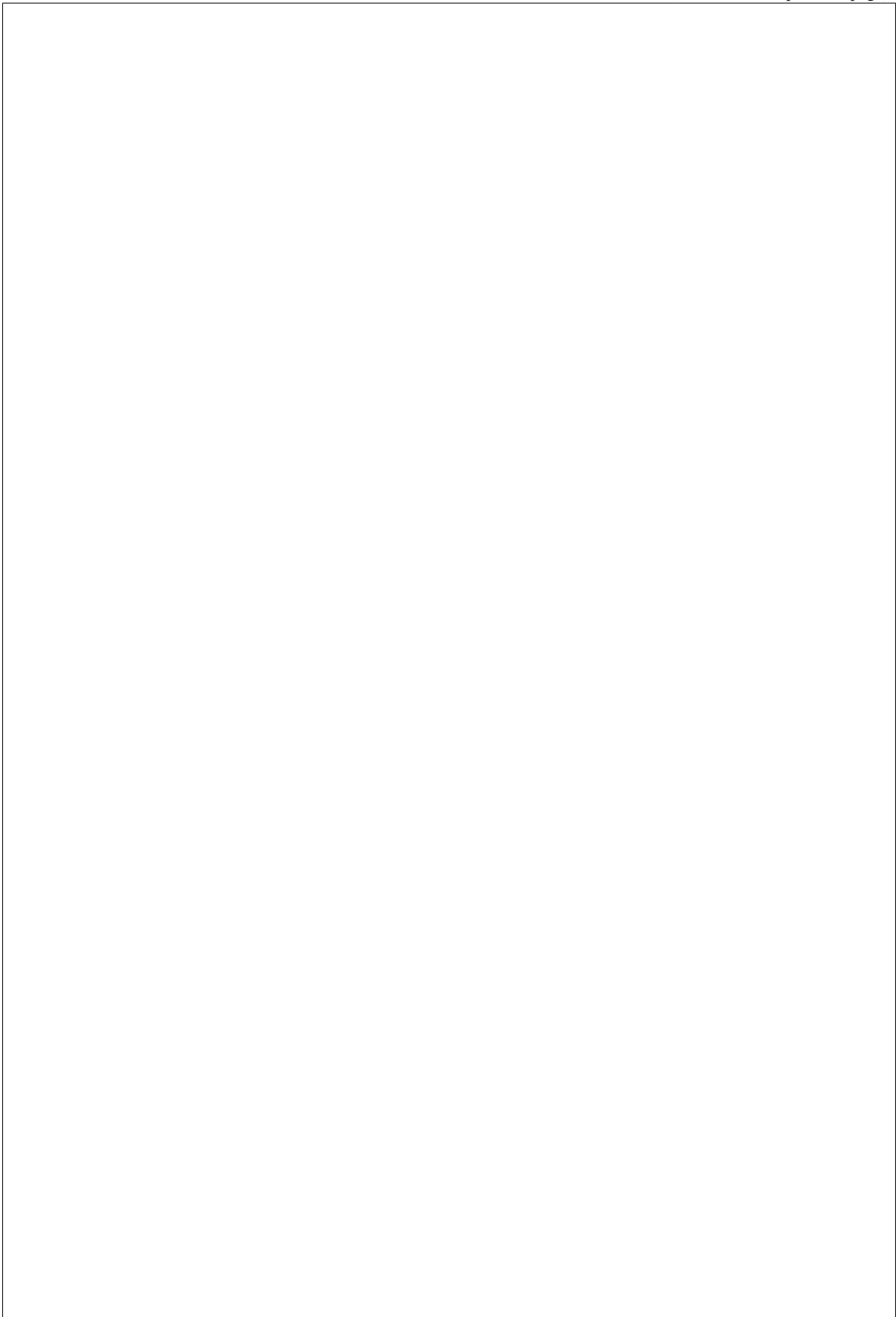
(continues on next page)

(continued from previous page)



(continues on next page)

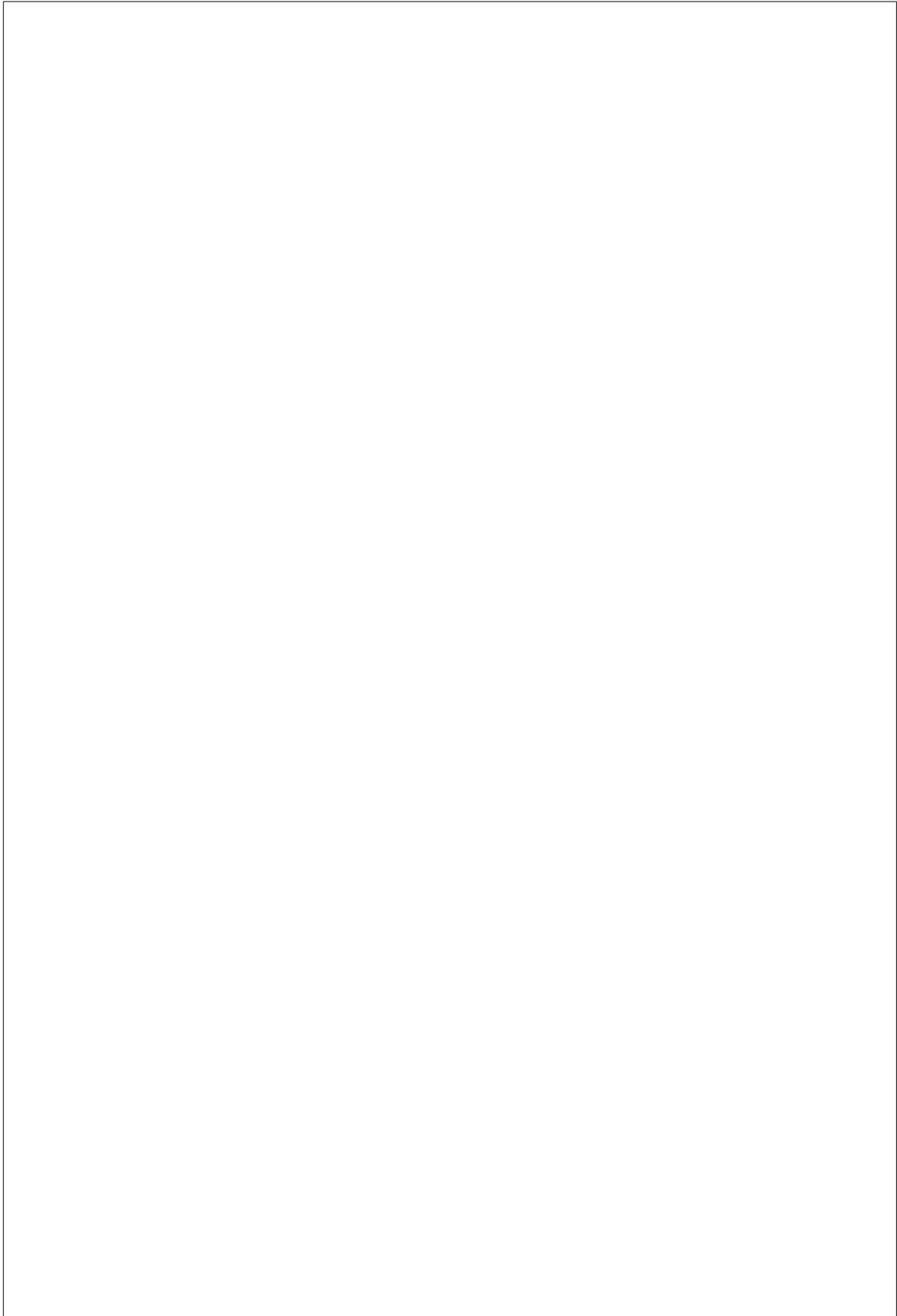
(continued from previous page)



(continues on next page)

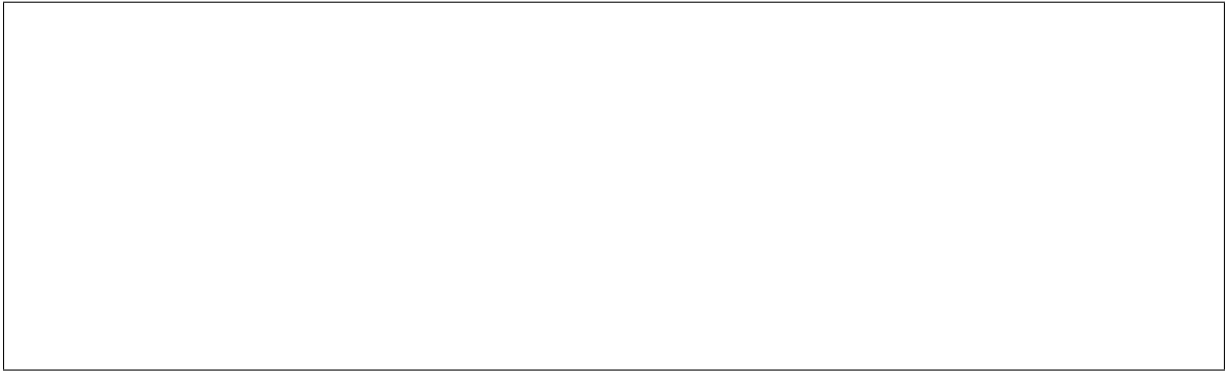


(continued from previous page)

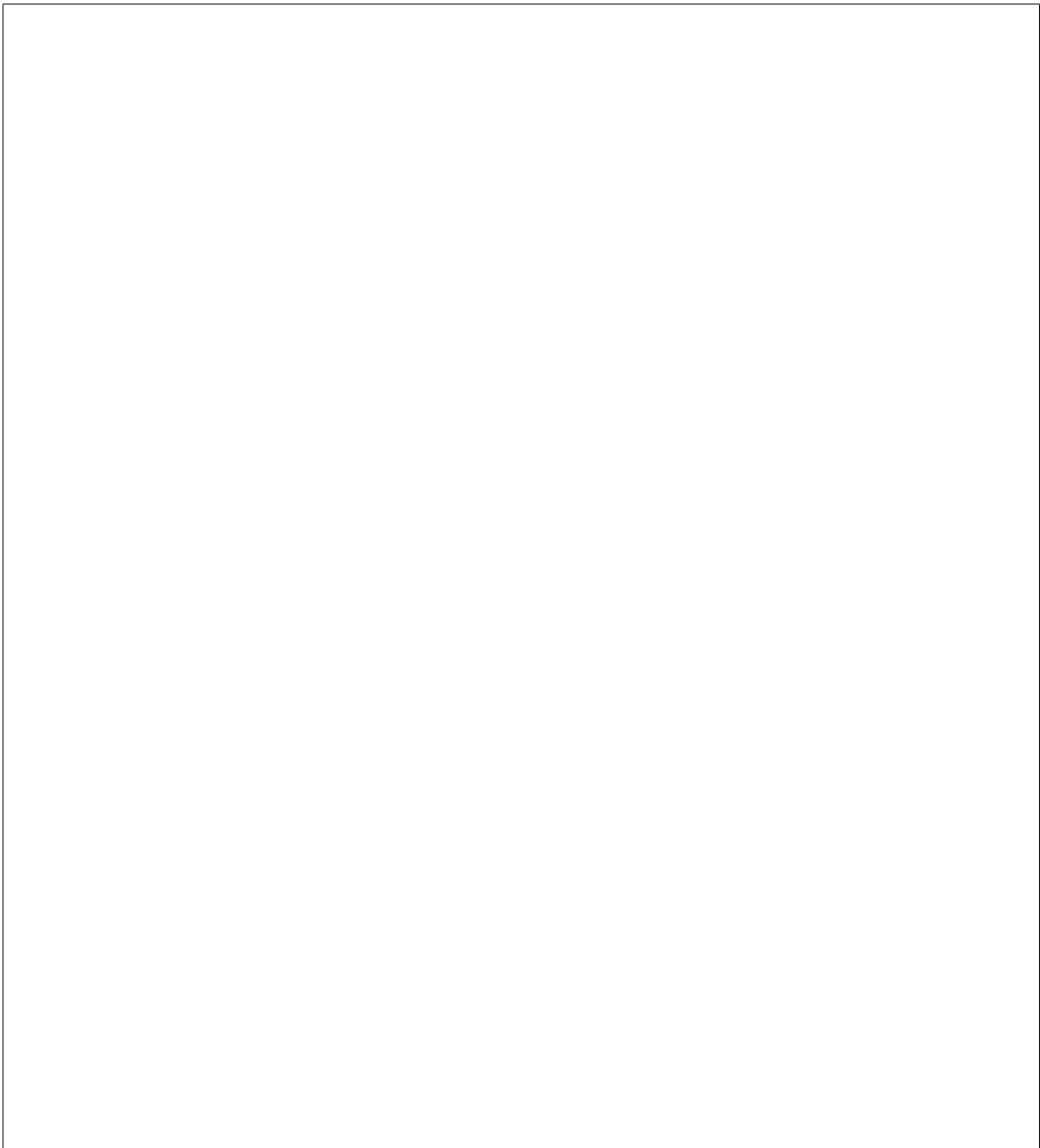


(continues on next page)

(continued from previous page)



## **Deployment and Metalsmith**



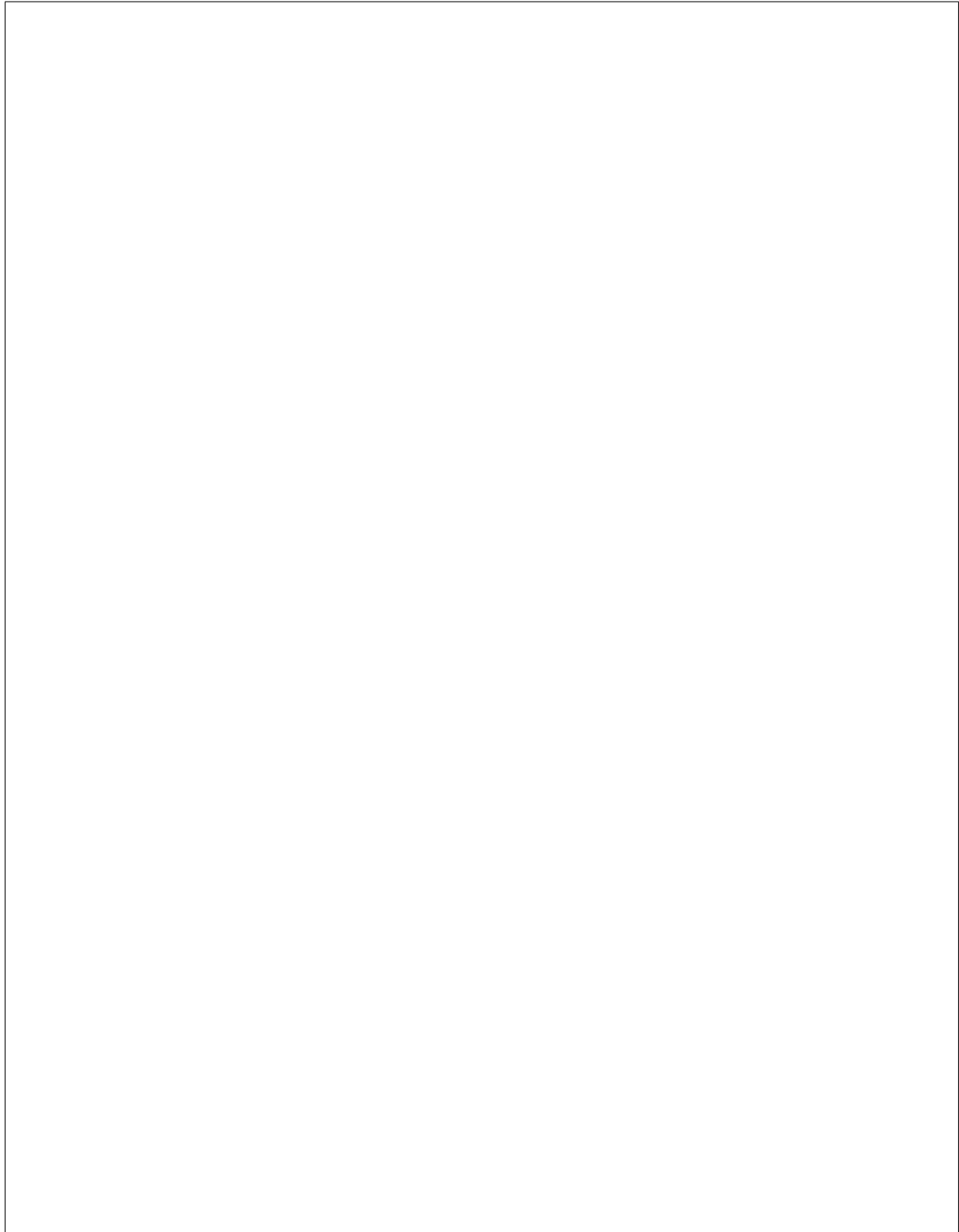
(continues on next page)

(continued from previous page)



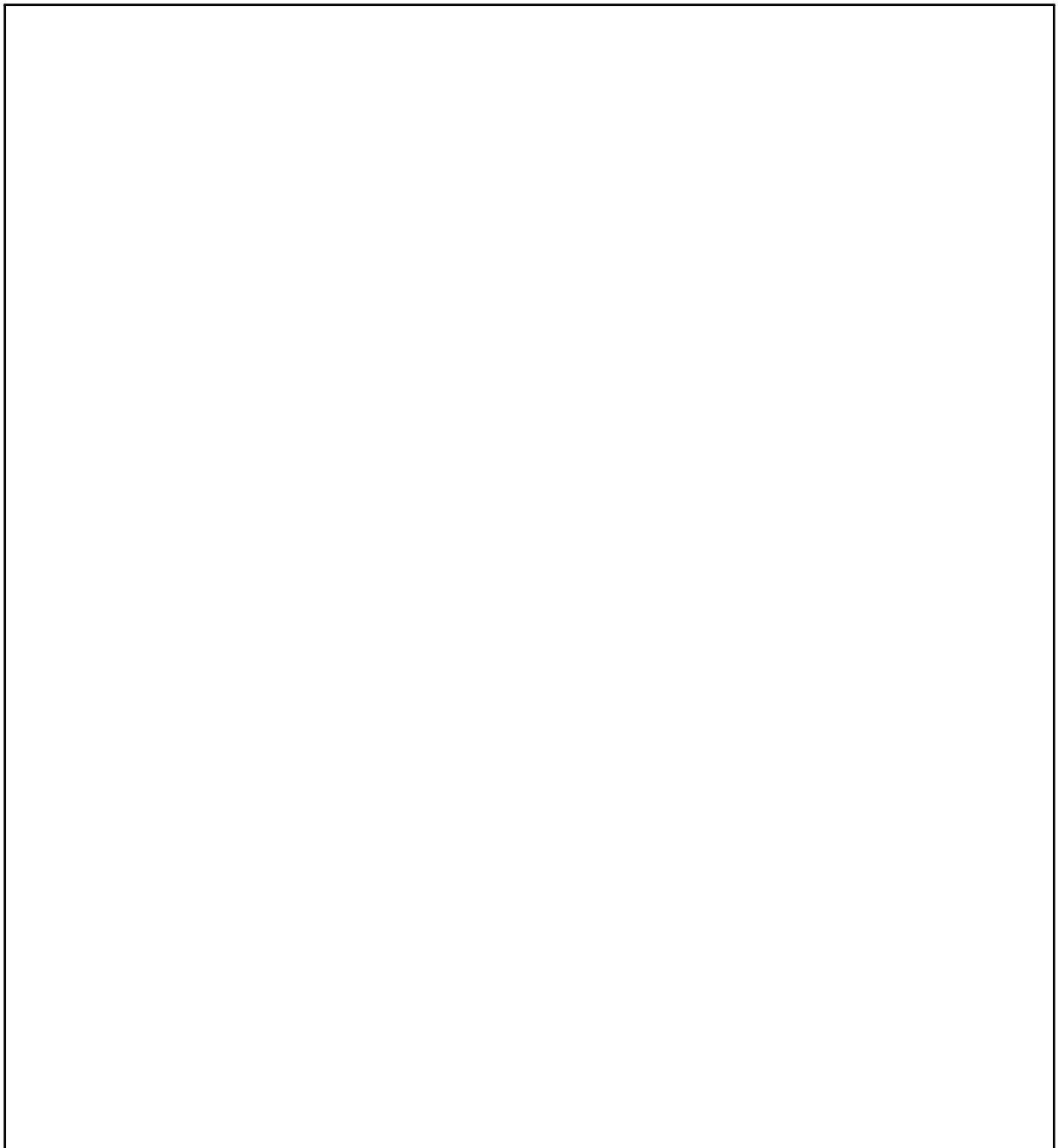
(continues on next page)

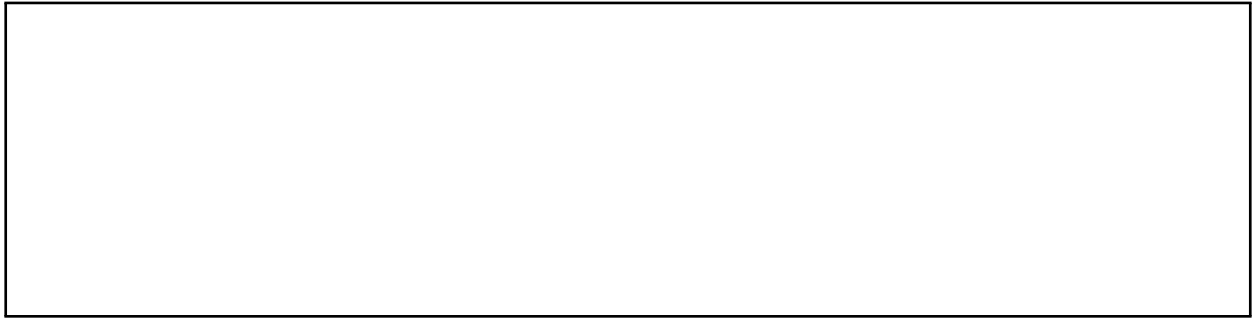
(continued from previous page)





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)

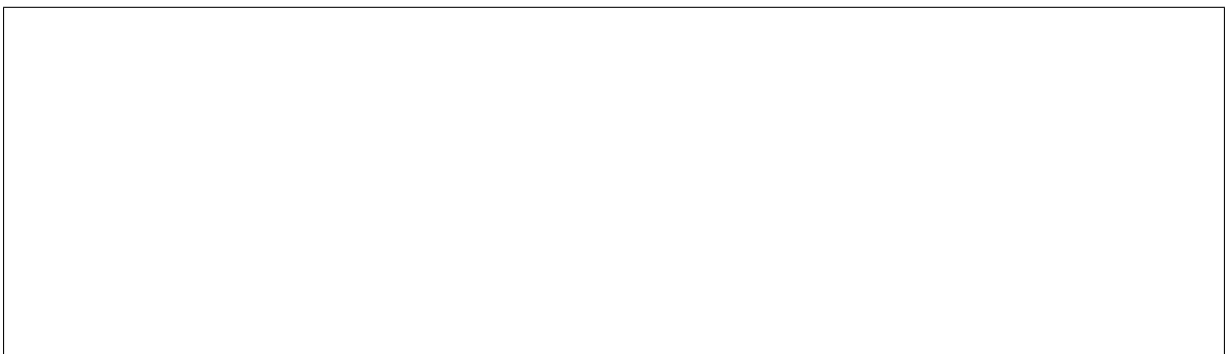
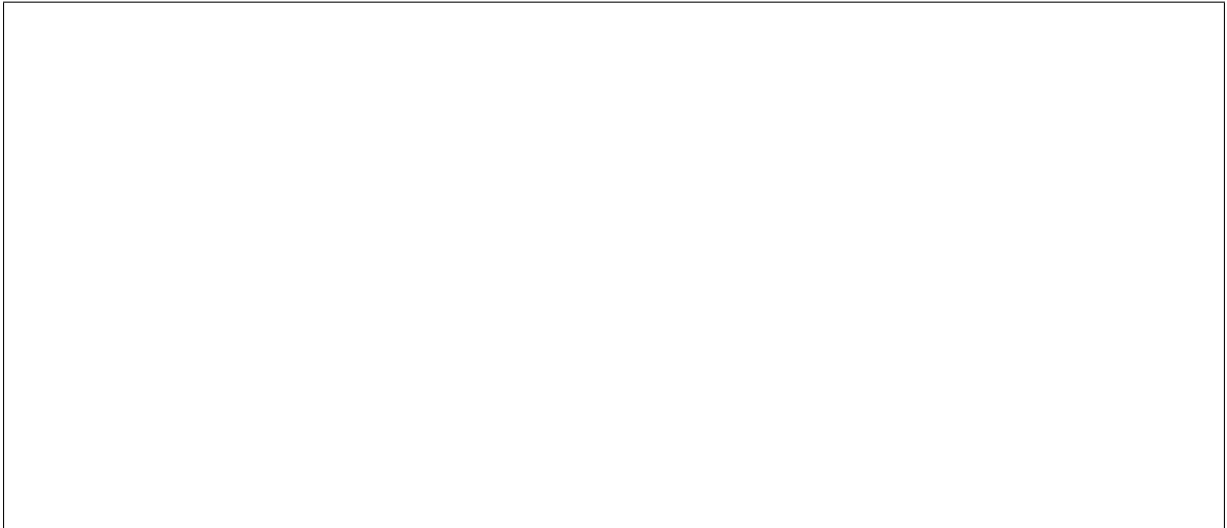
(continued from previous page)

--



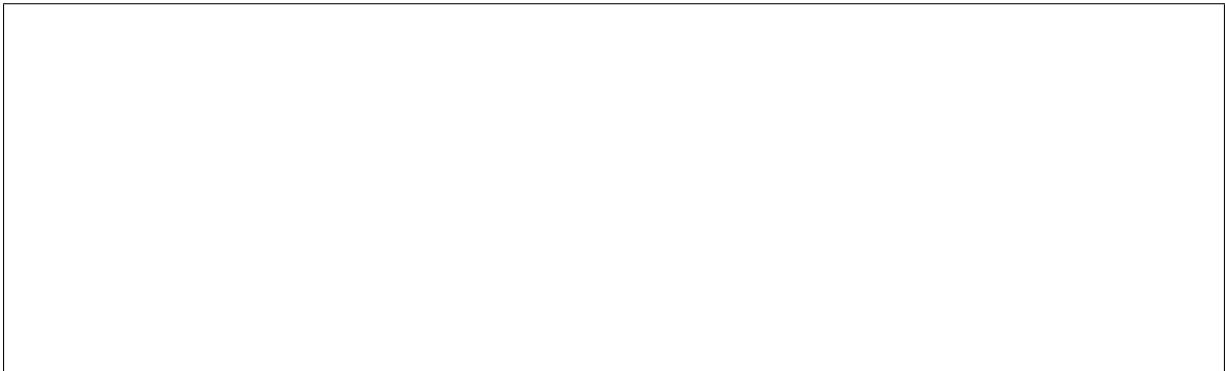
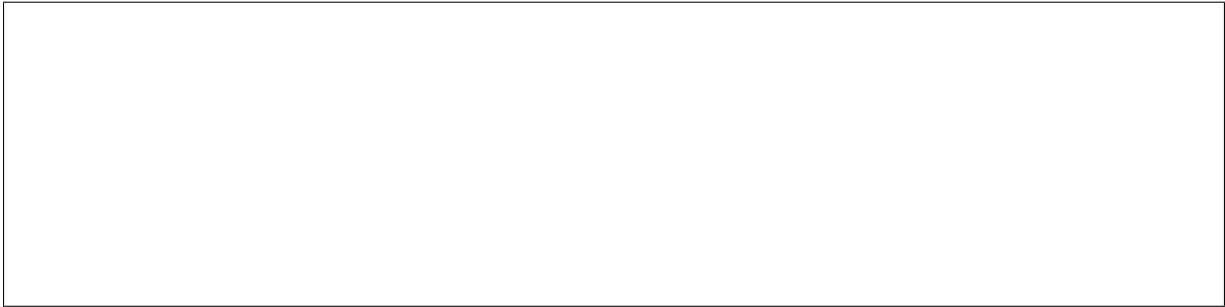
the only local storage contents being those in memory. It is supported by pxe, ipxe, redfish-virtual-media and ilo-virtual-media boot interfaces.

## **Configuration**

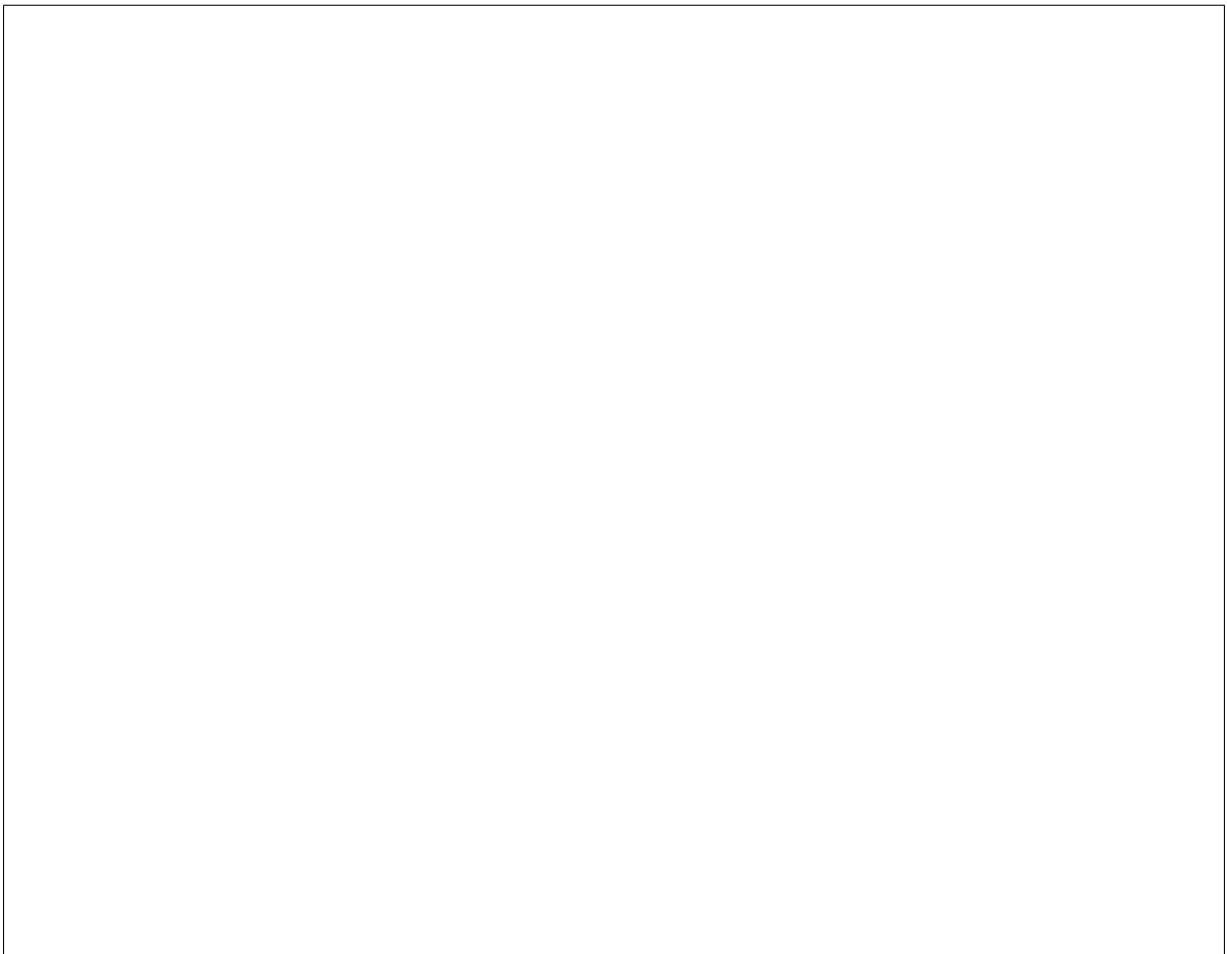


(continues on next page)

(continued from previous page)



## **Creating a ramdisk**



(continues on next page)

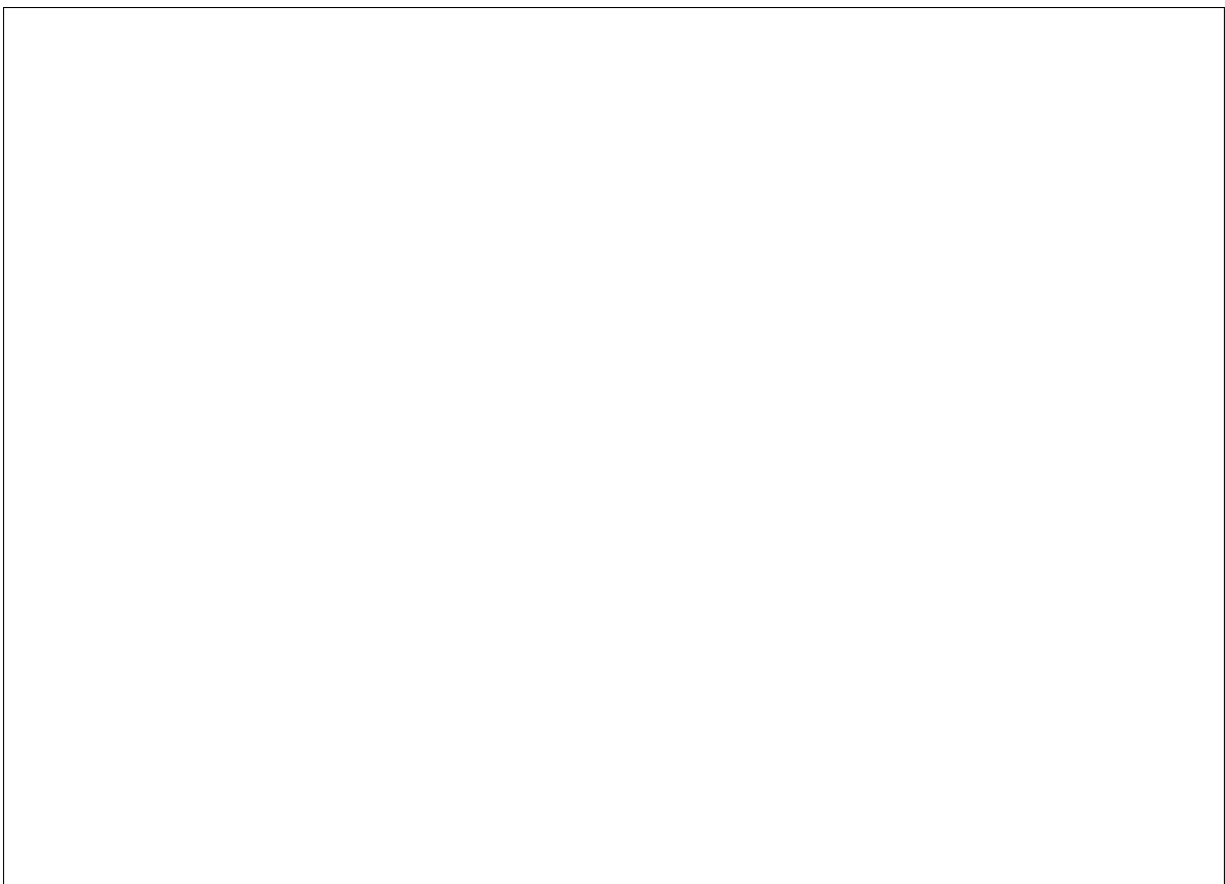
(continued from previous page)

--





## **Booting a ramdisk**



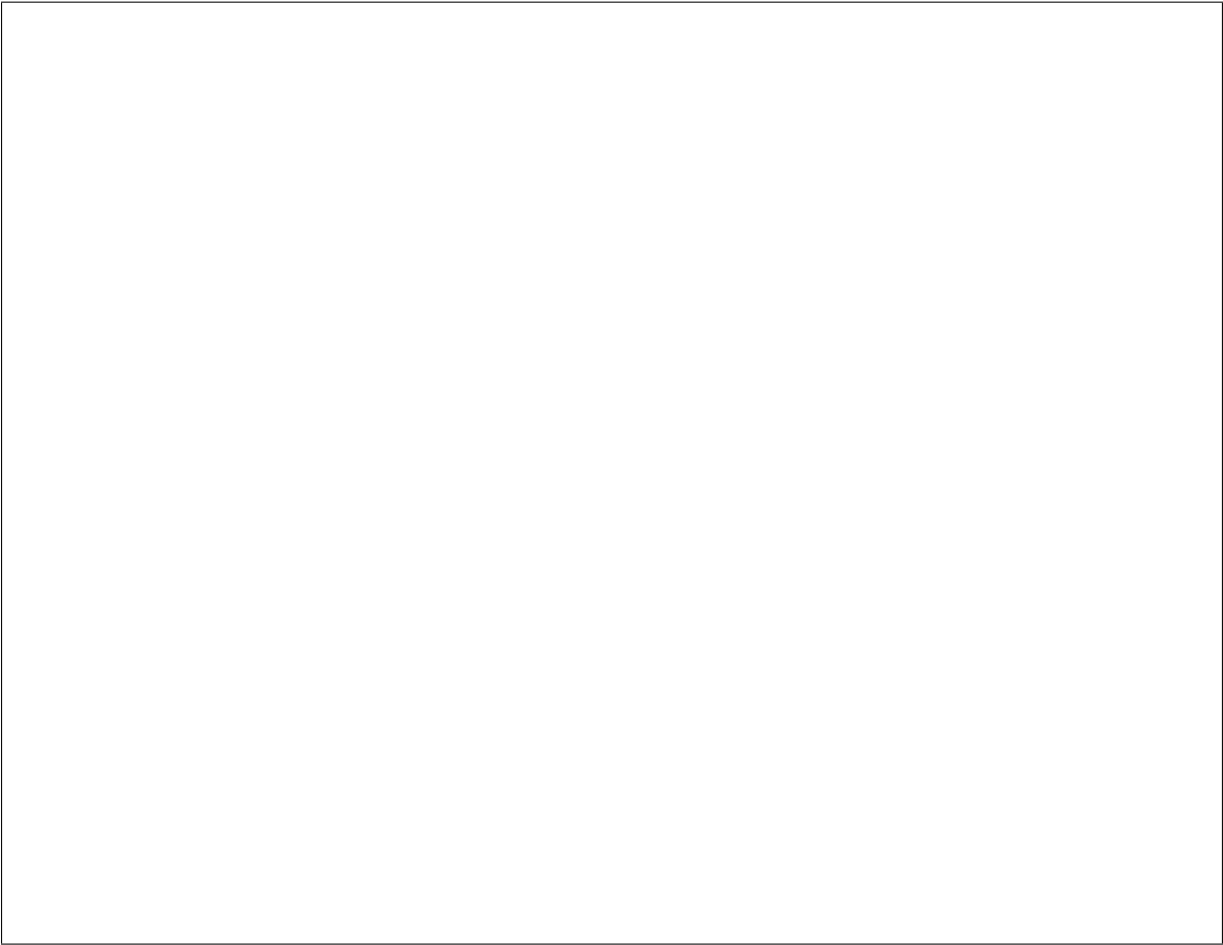
(continues on next page)

(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**



## **Limitations**







visioning and cleaning networks



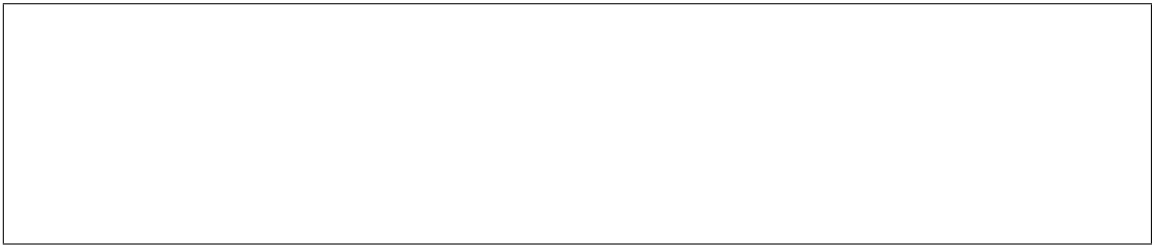
## **Ceph Object Gateway support**

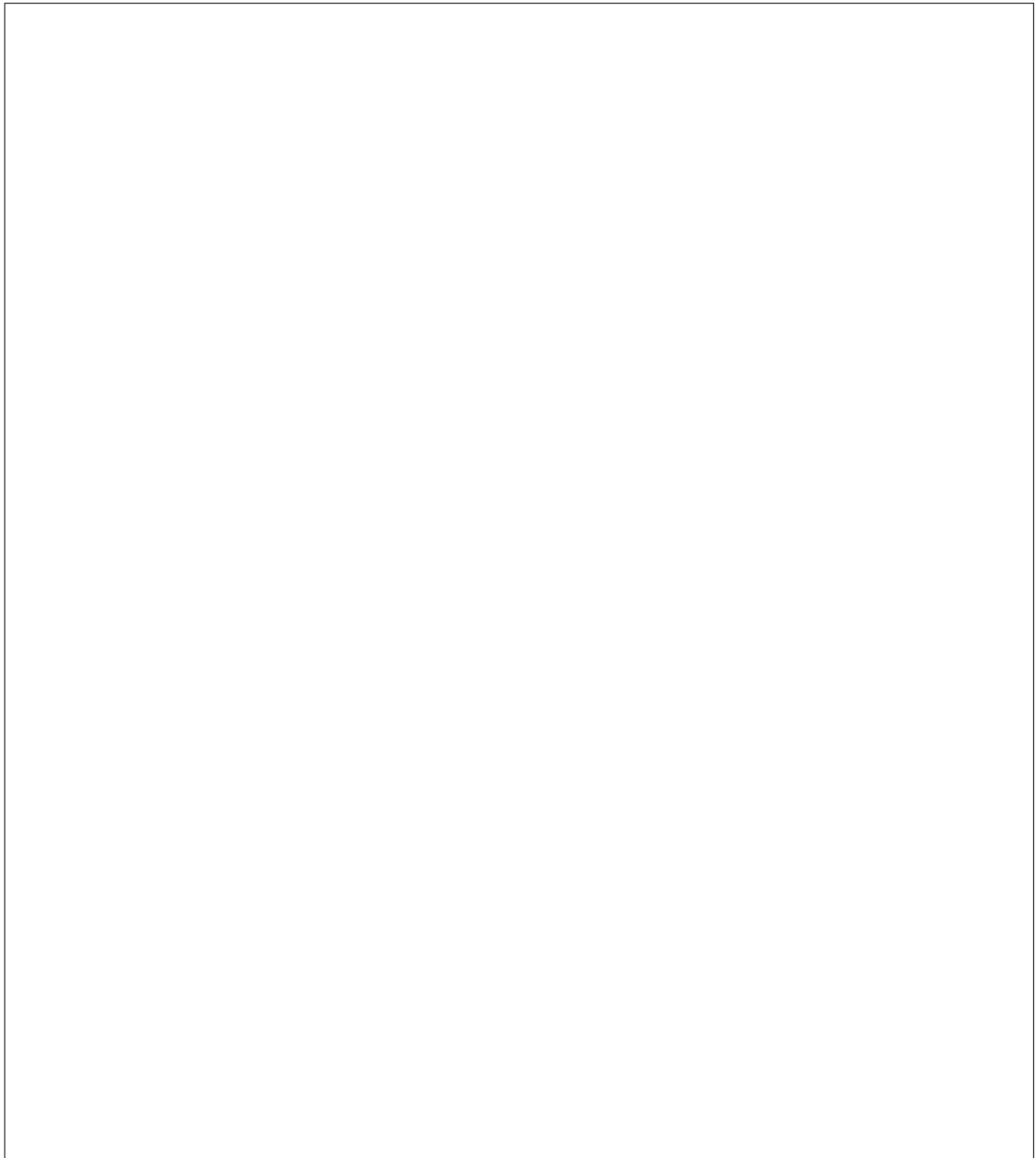
### **Overview**

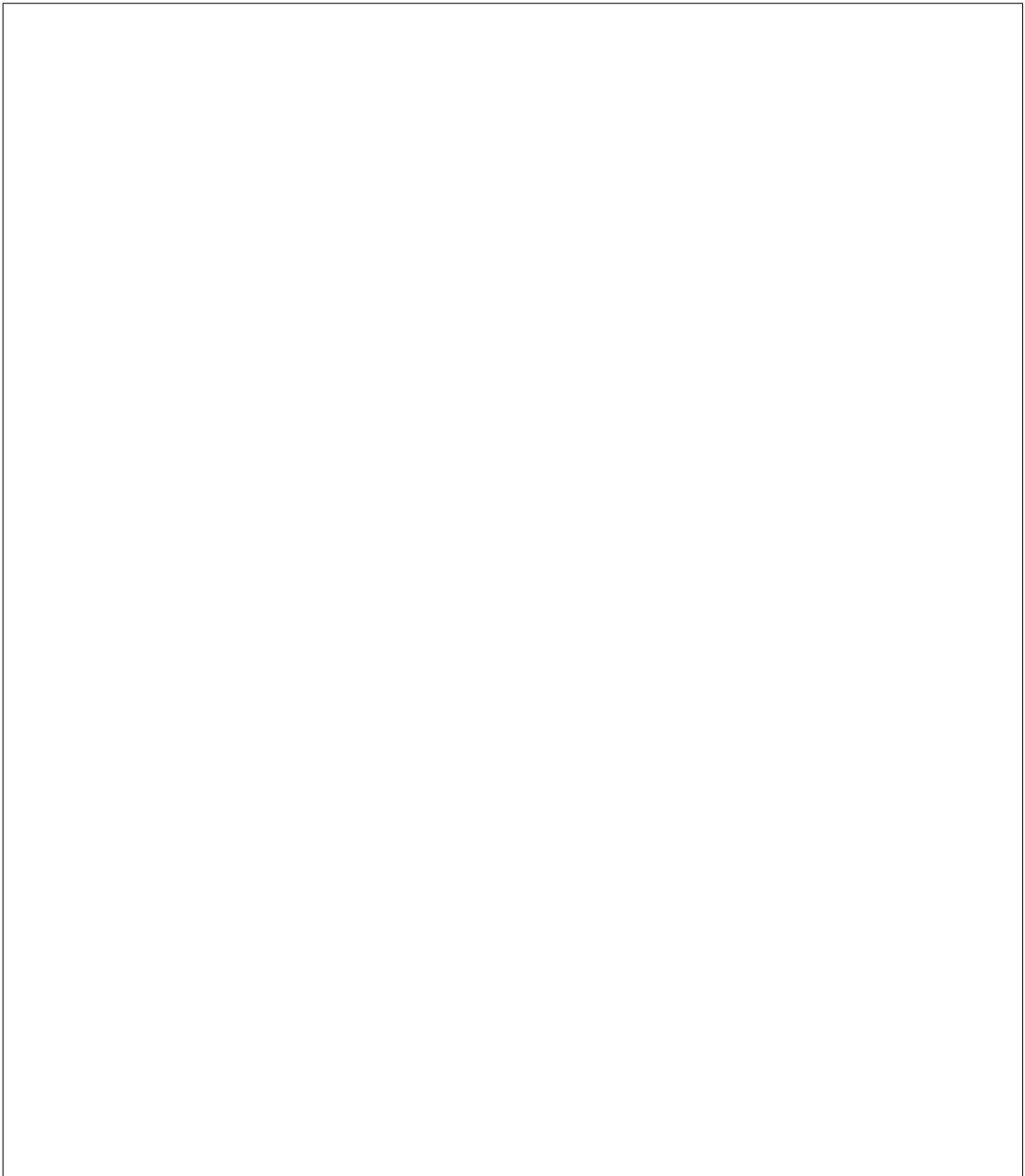


## **Configure IroniC and Glance with RADOS Gateway**

















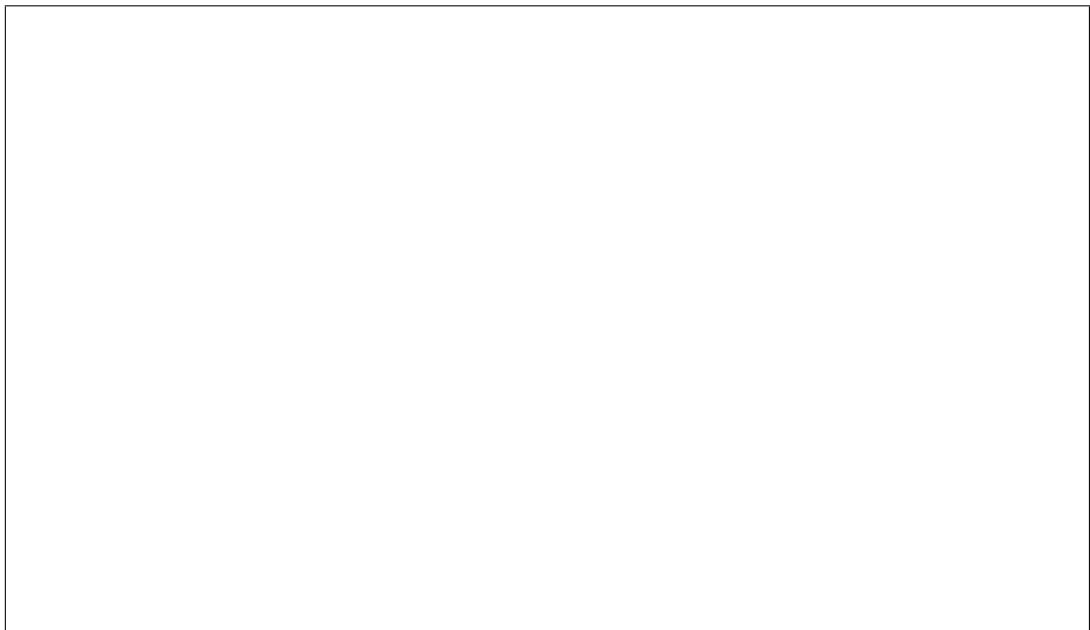
## **Building images for Windows**

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

**Requirements:**

**Preparation:**





(continues on next page)

(continued from previous page)

--

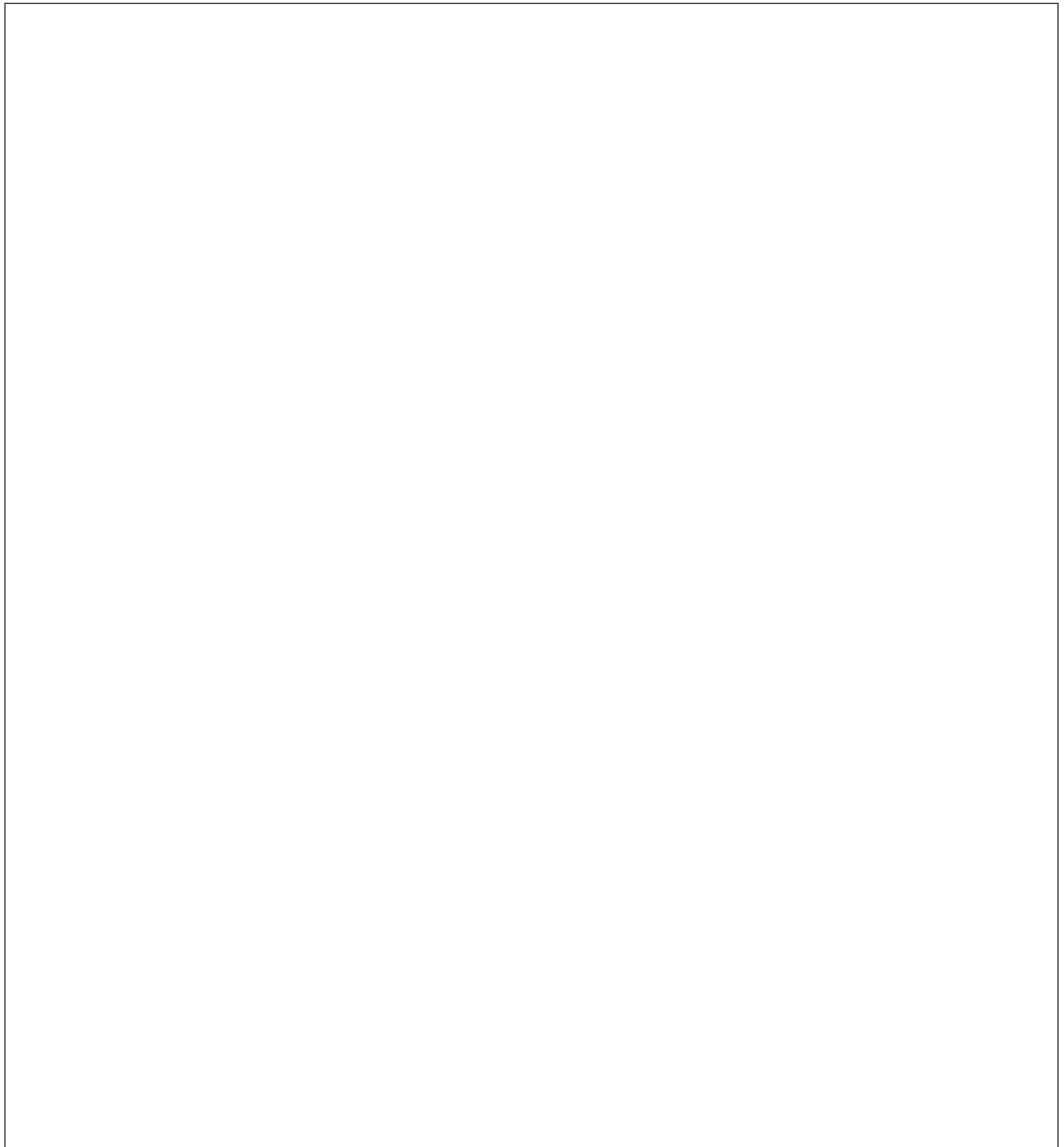
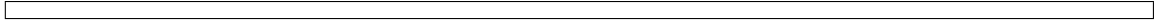
**Implementation:**



(continues on next page)

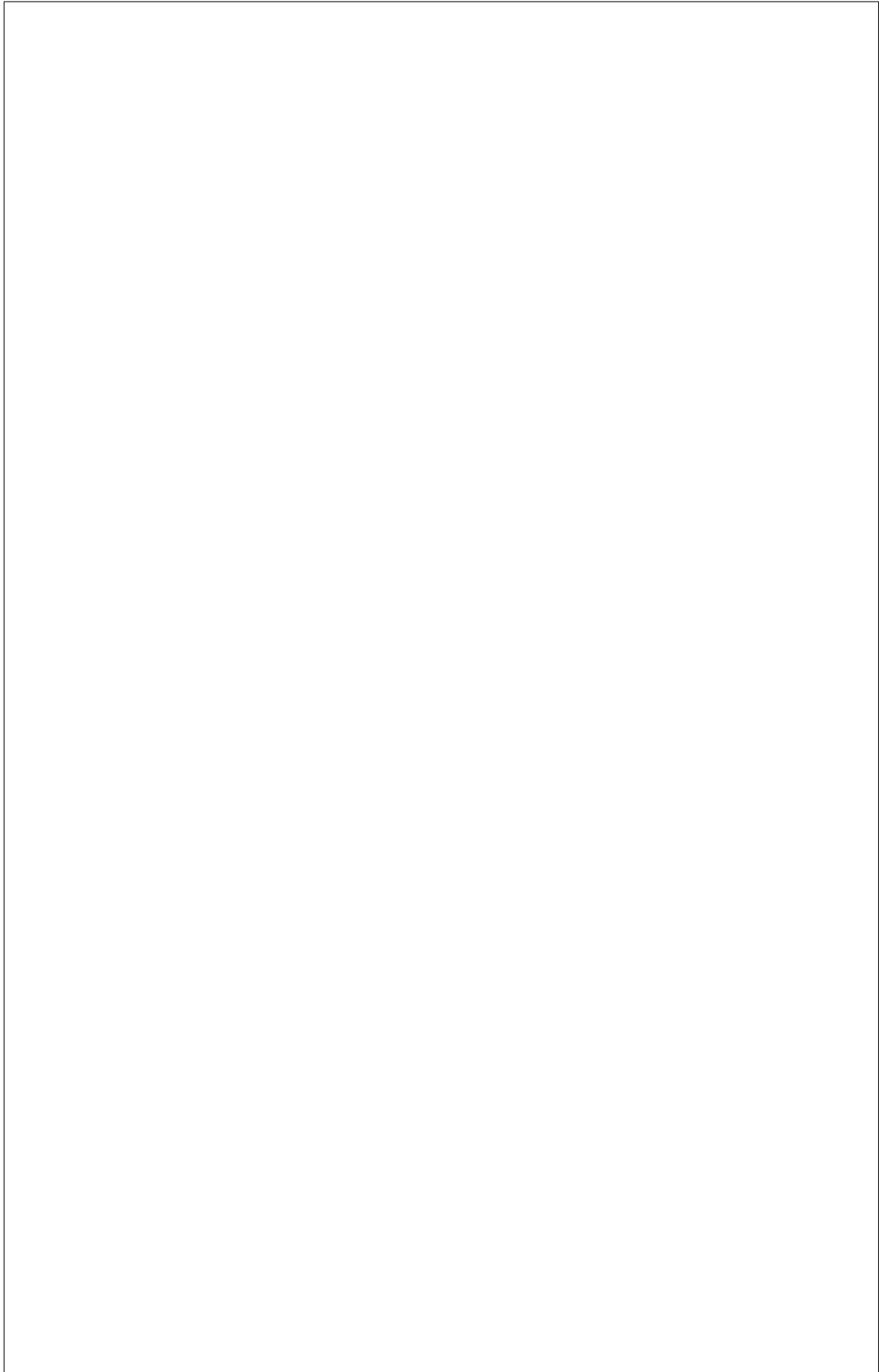


(continued from previous page)



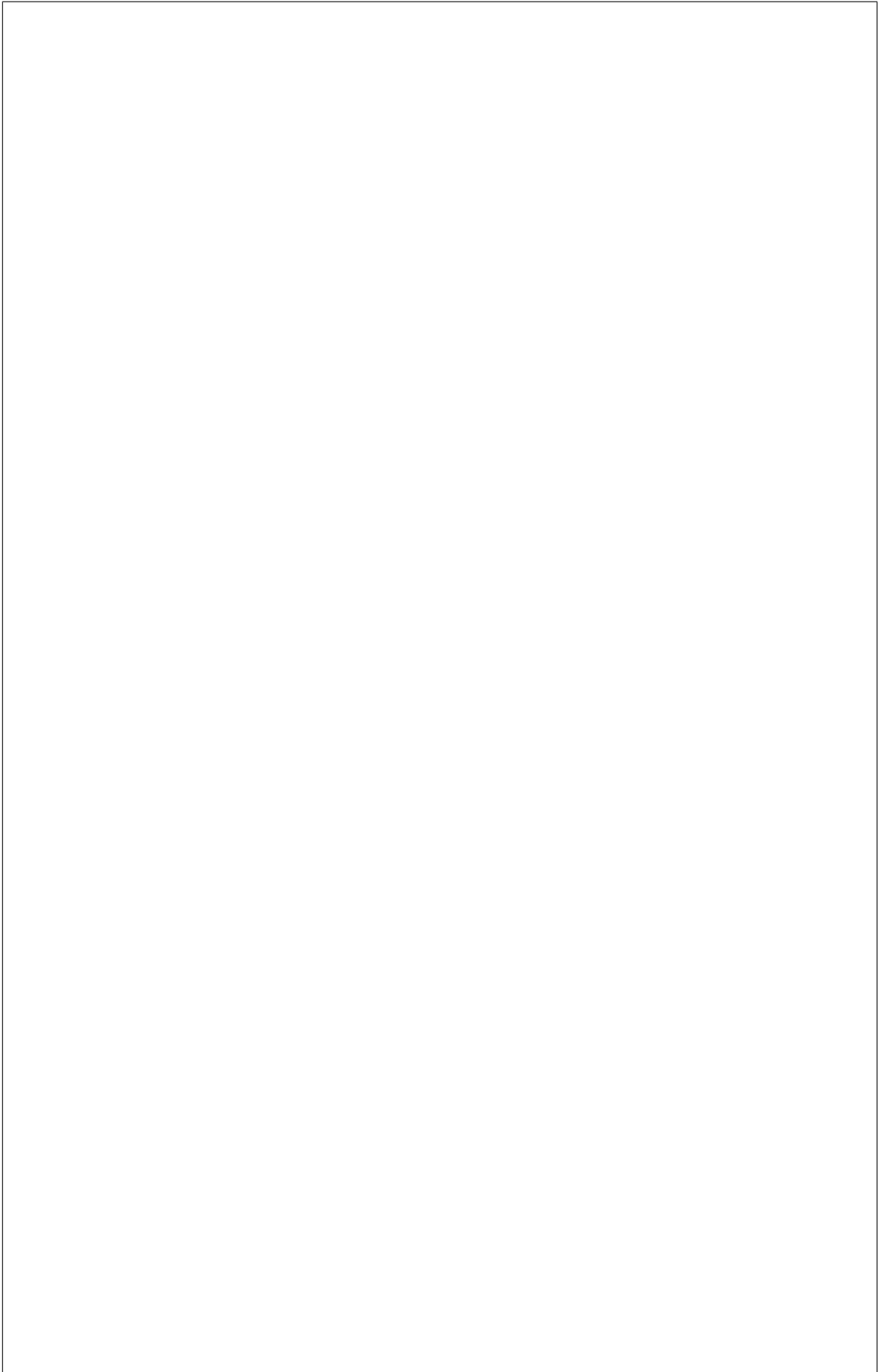
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)





## **Emitting Software Metrics**





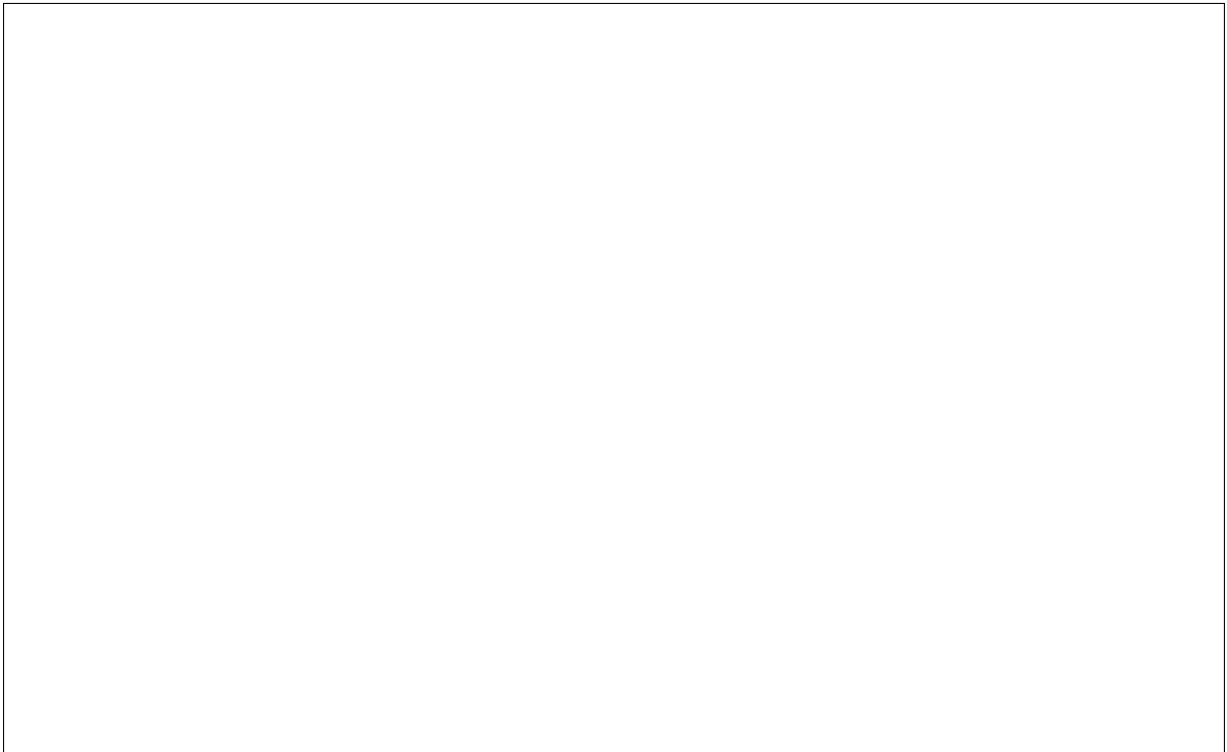


## **Configuring the Bare Metal Service to Enable Metrics**

**Enabling metrics in ironic-api and ironic-conductor**



also supply connection information in the ironiC configuration file:



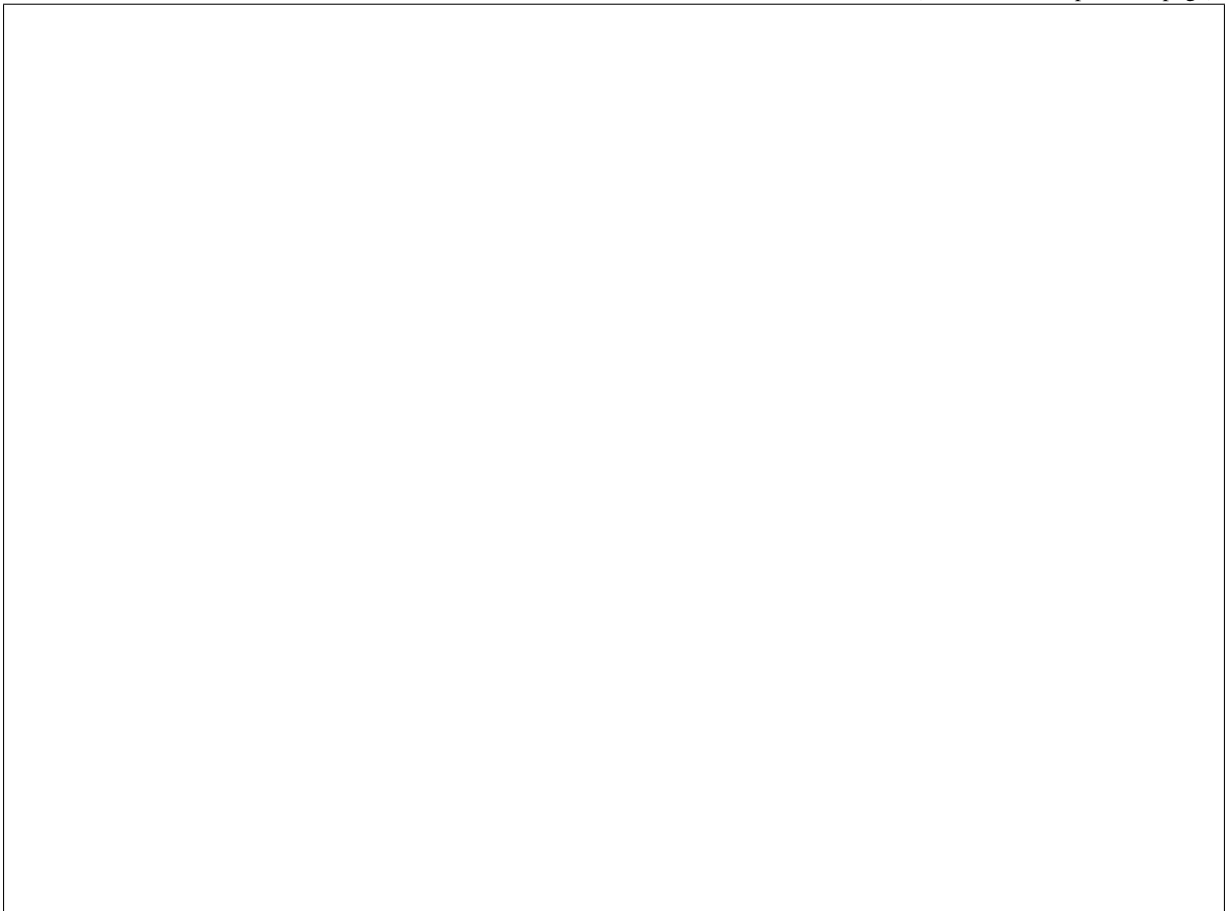
### **Enabling metrics in ironiC-python-agent**

in your ironic configuration file on all ironic-conductor hosts:

ured in the ironiC configuration file as well:

(continues on next page)

(continued from previous page)



## **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**

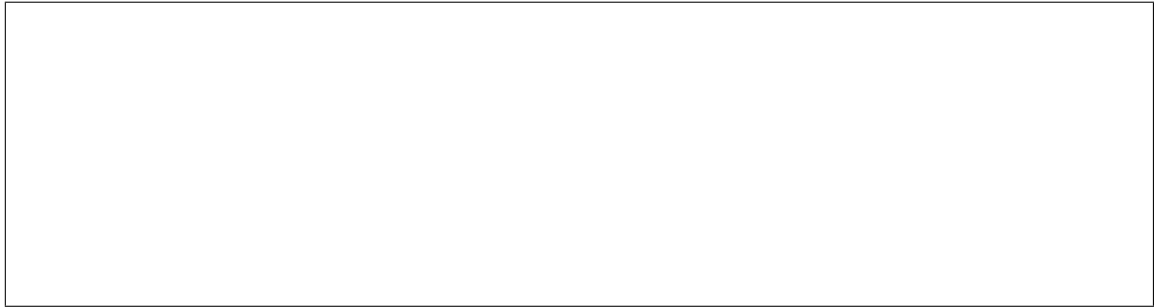


ration options for the Bare Metal service are included in the `etc/ironic/ironic_api_audit_map.conf.sample` file. To understand CADF format specified in `ironic_api_audit_map.conf` file refer to [CADF Format](#).



(continues on next page)

(continued from previous page)



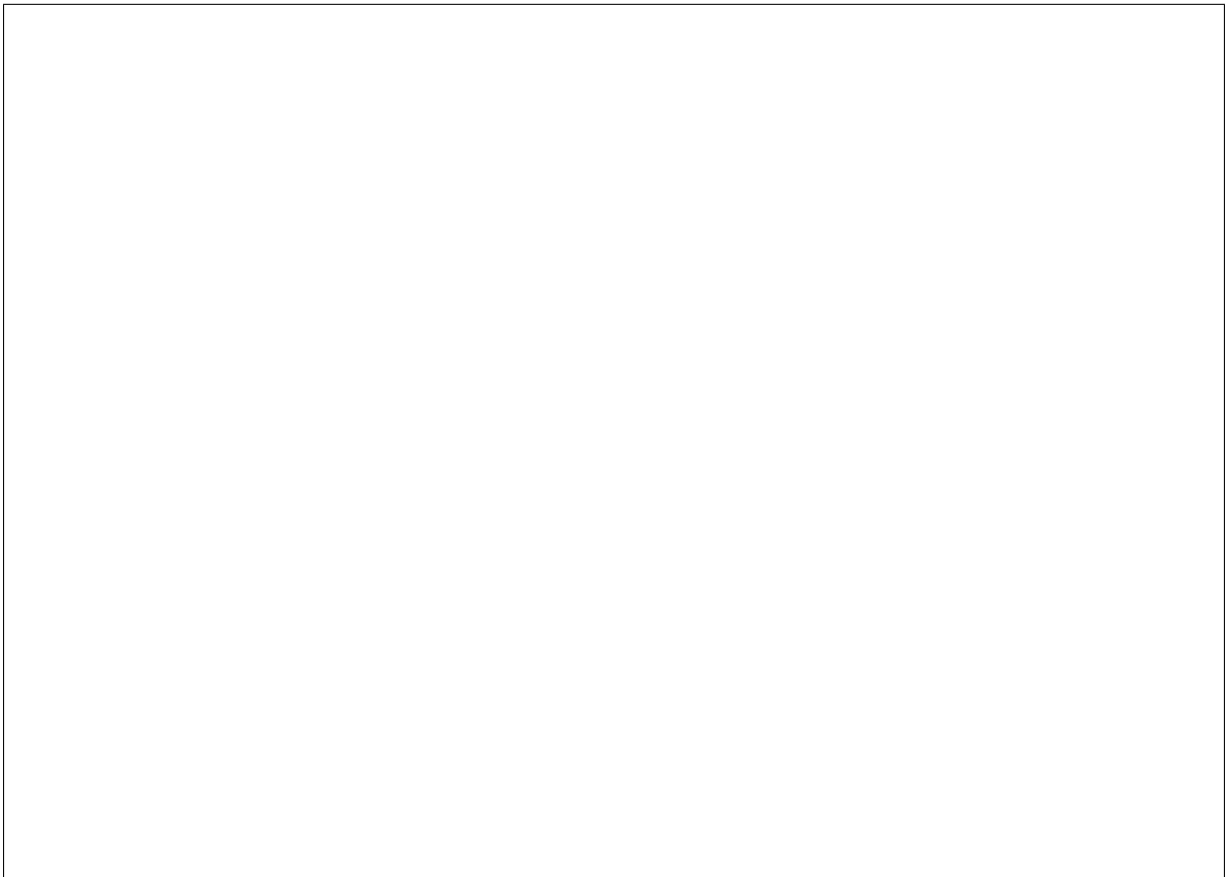
(continues on next page)



(continued from previous page)



### **Sample Audit Event**



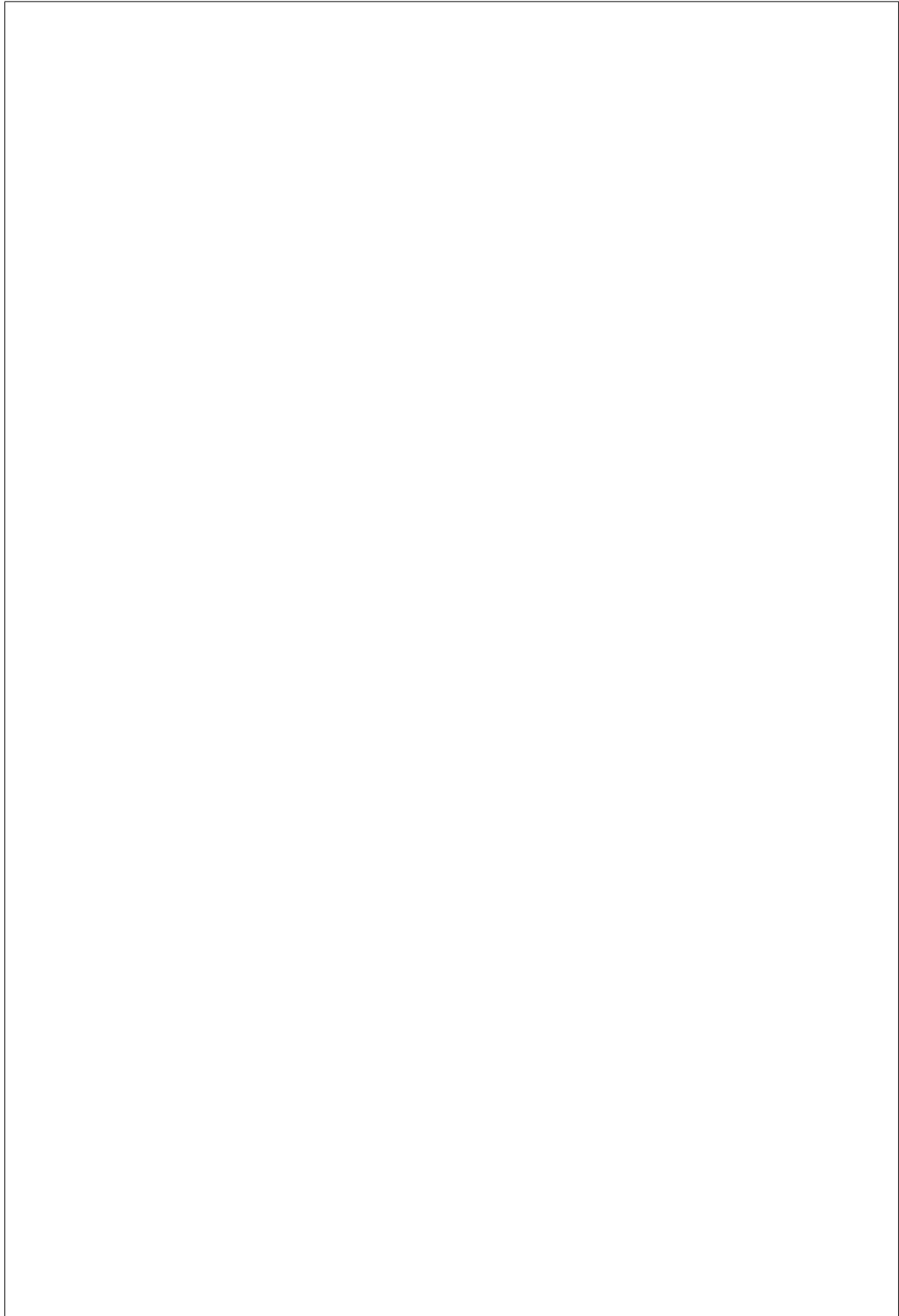
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



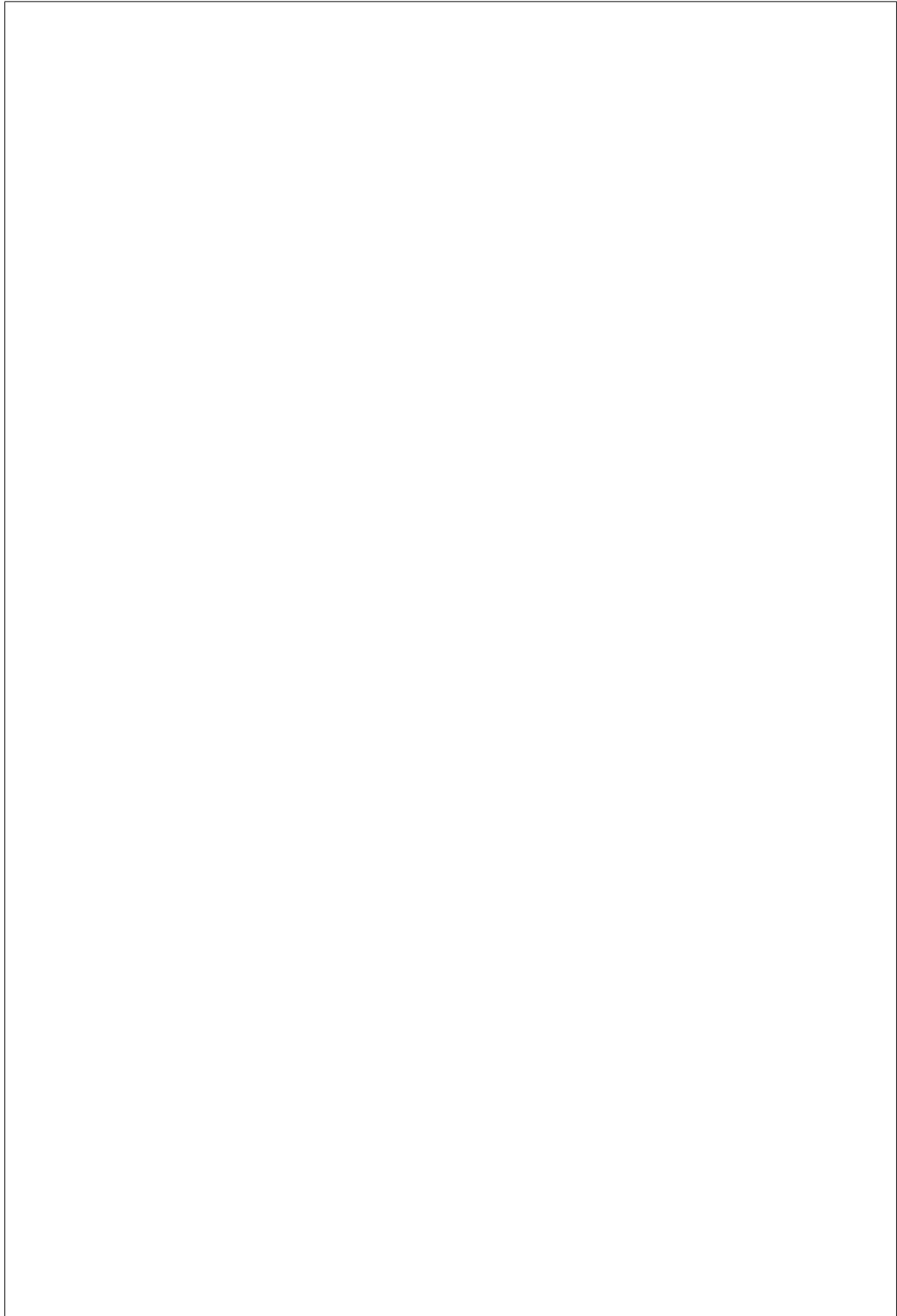
(continues on next page)

(continued from previous page)



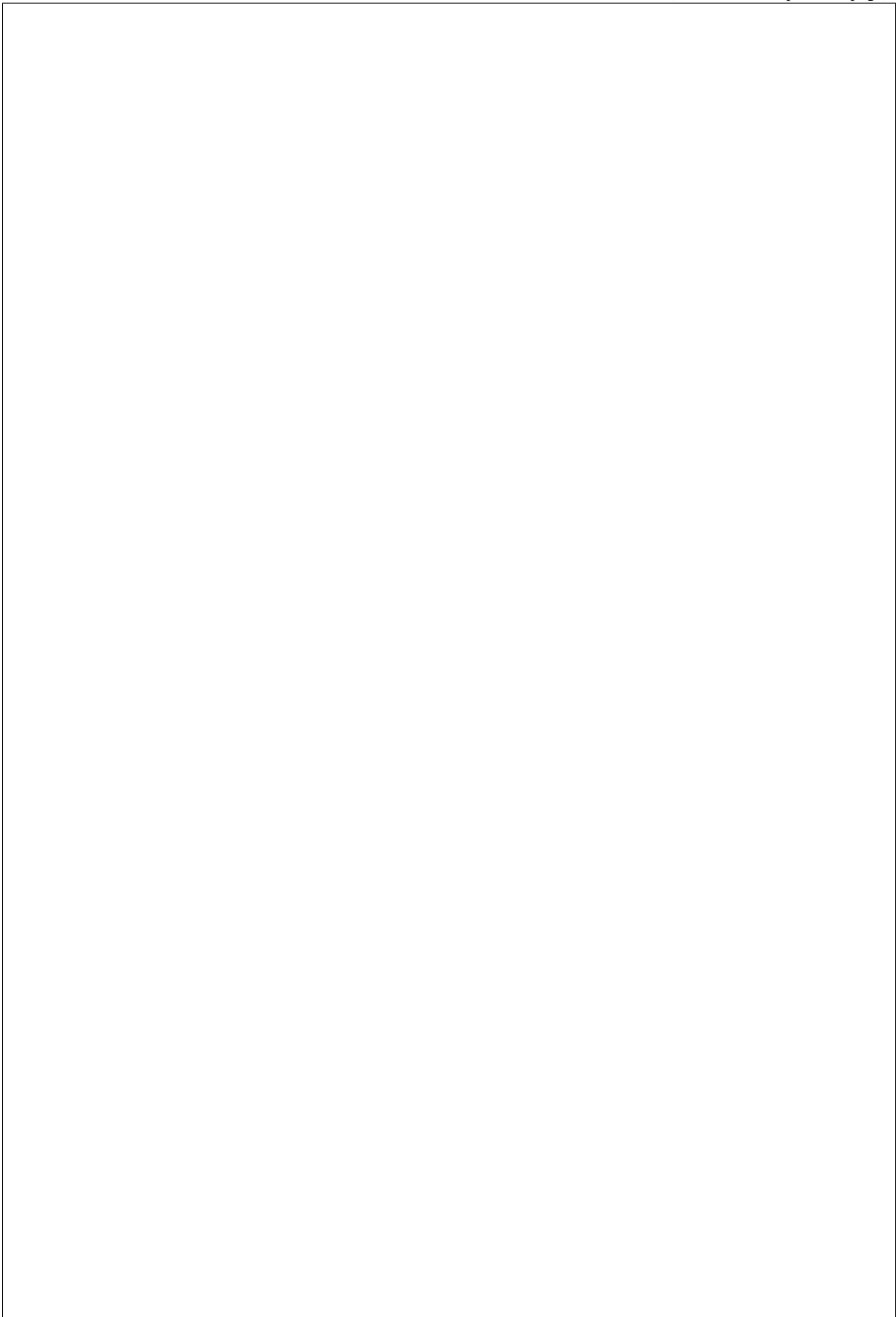
(continues on next page)

(continued from previous page)



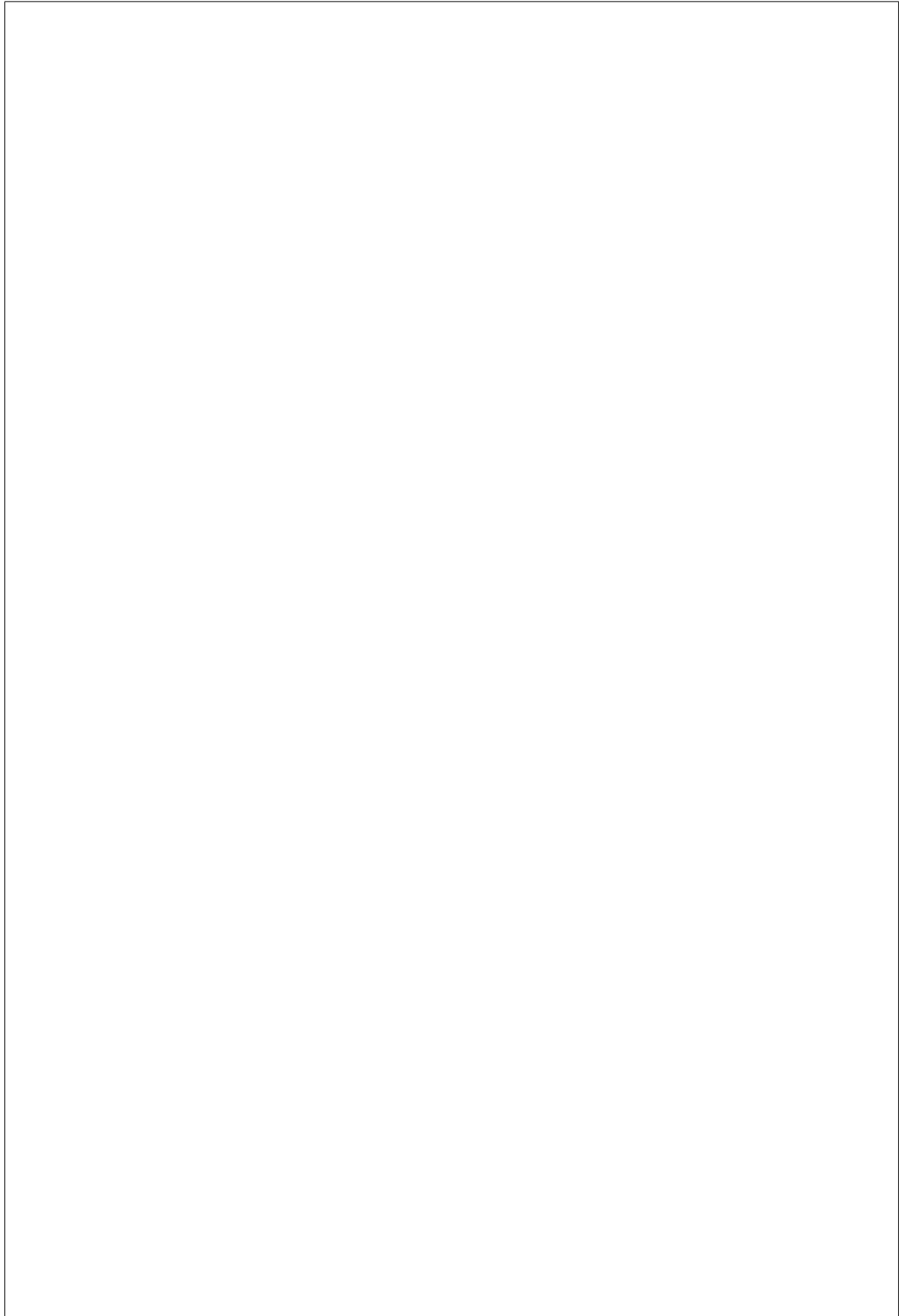
(continues on next page)

(continued from previous page)



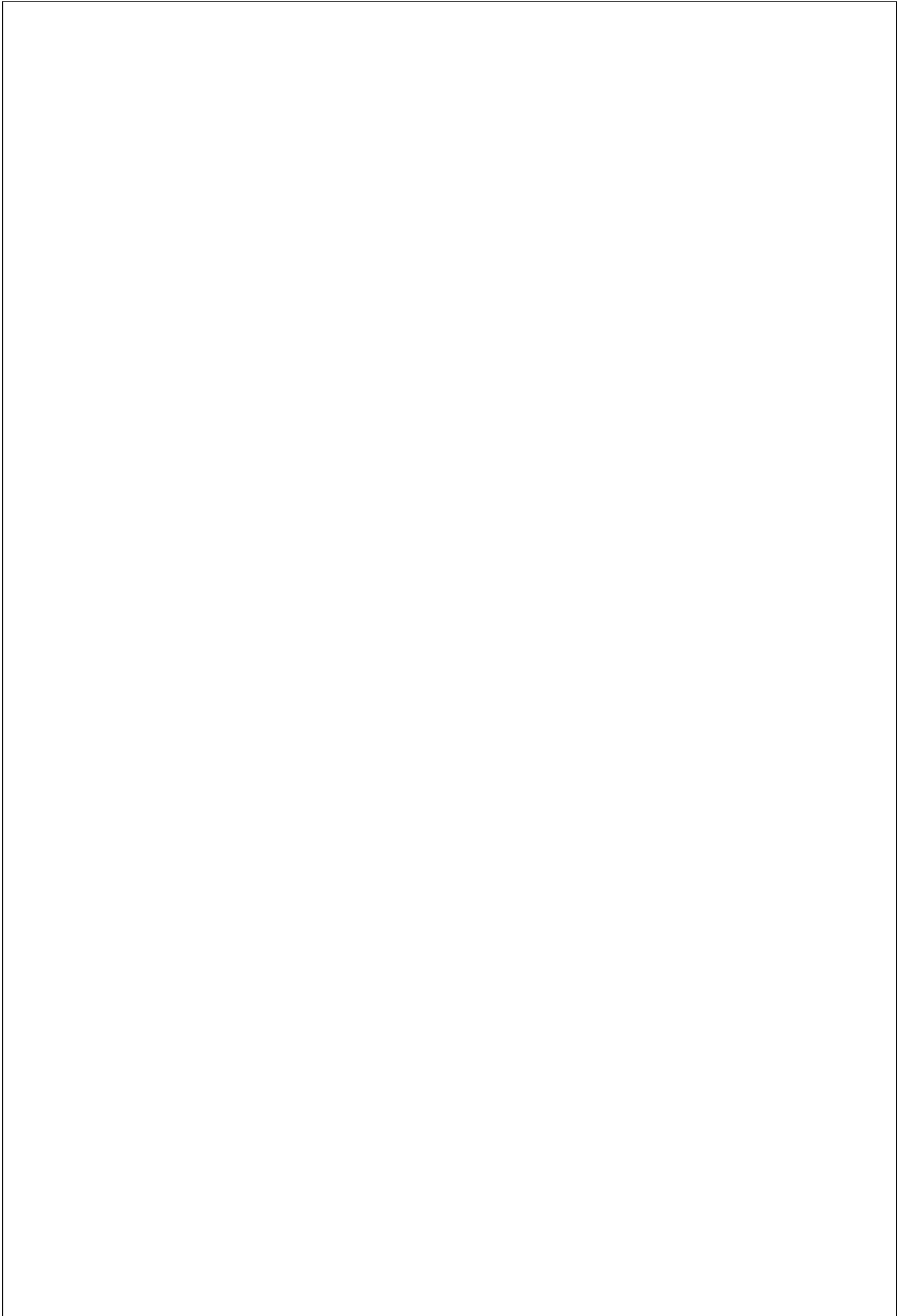
(continues on next page)

(continued from previous page)



(continues on next page)

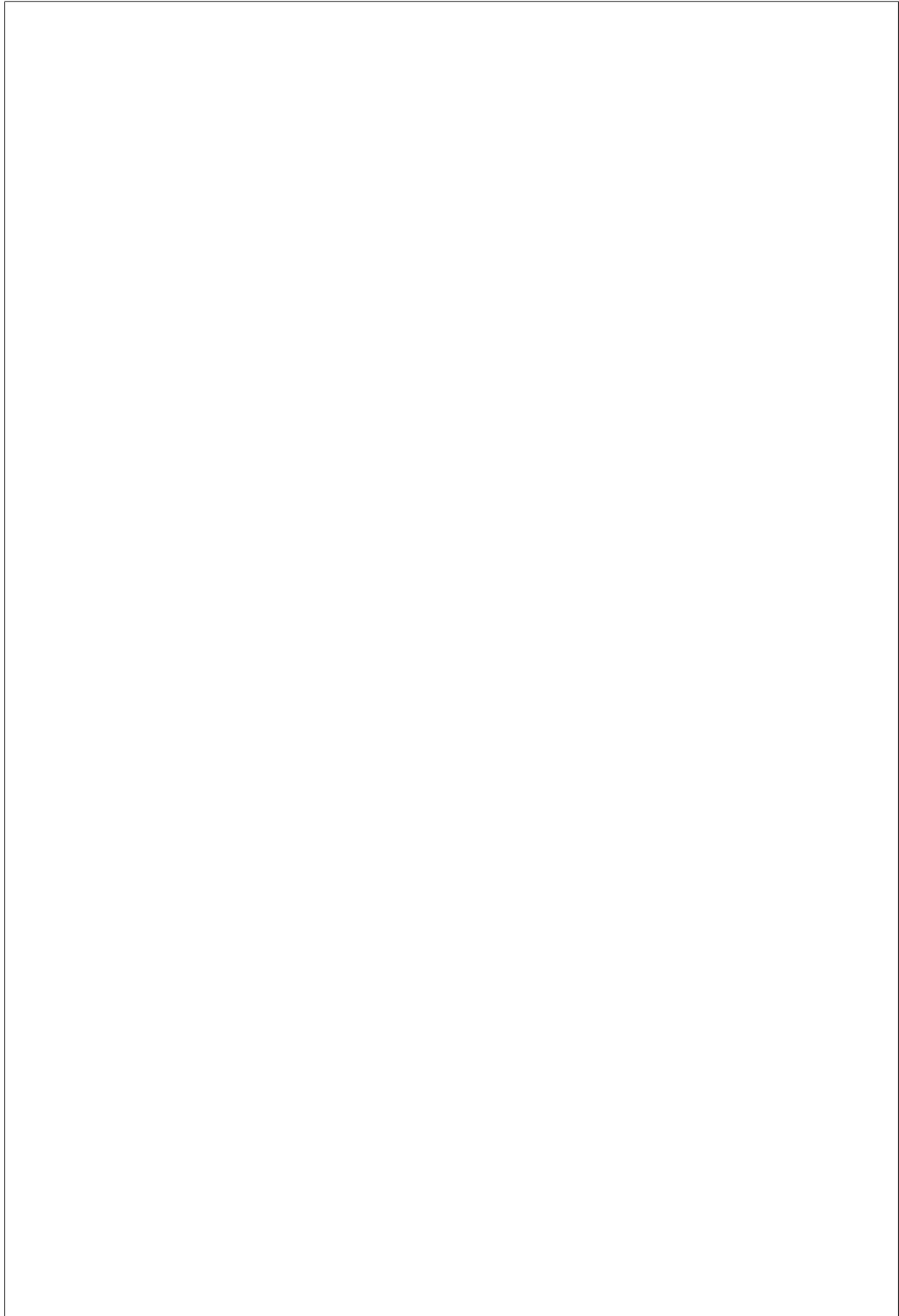
(continued from previous page)



(continues on next page)



(continued from previous page)



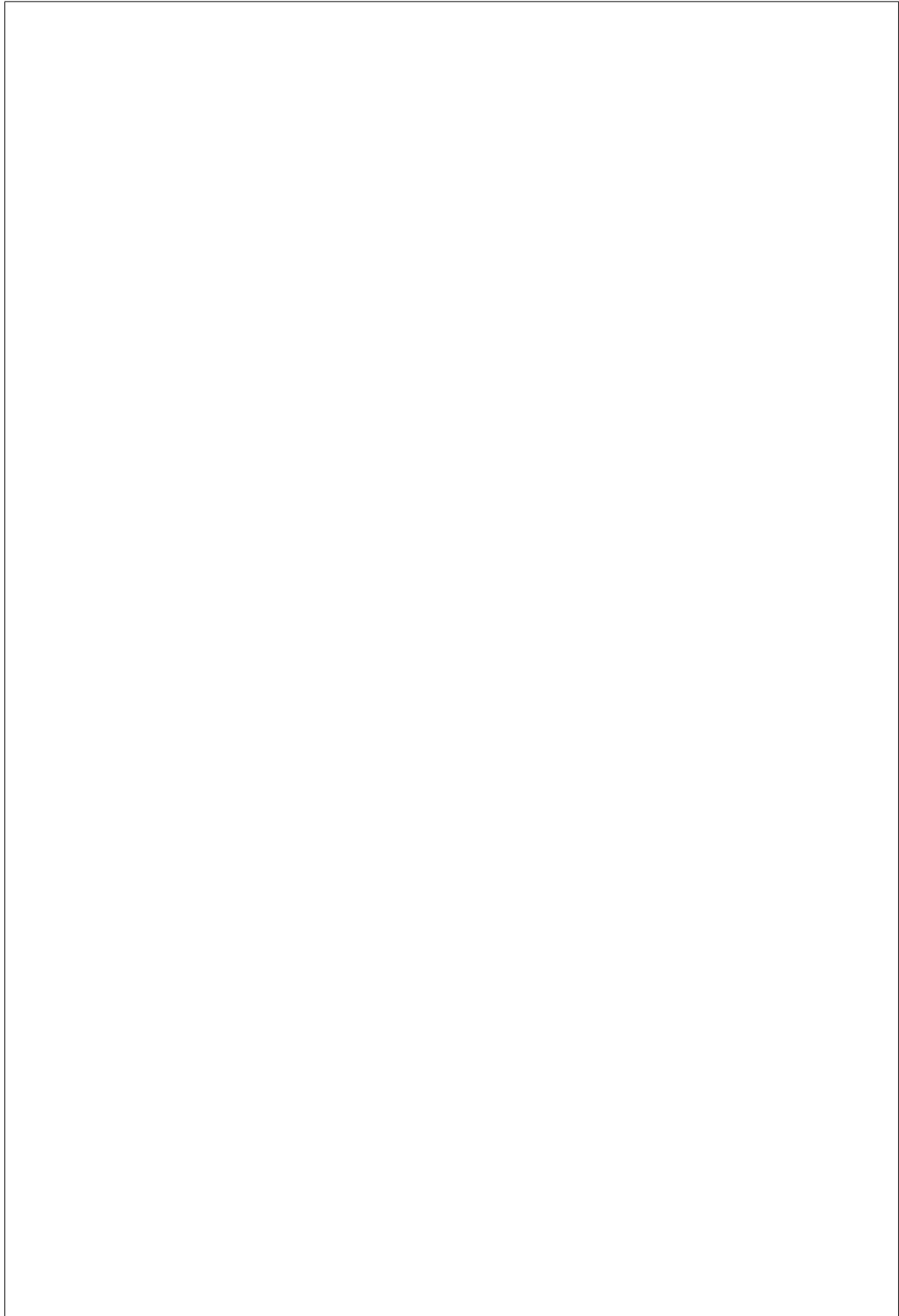
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



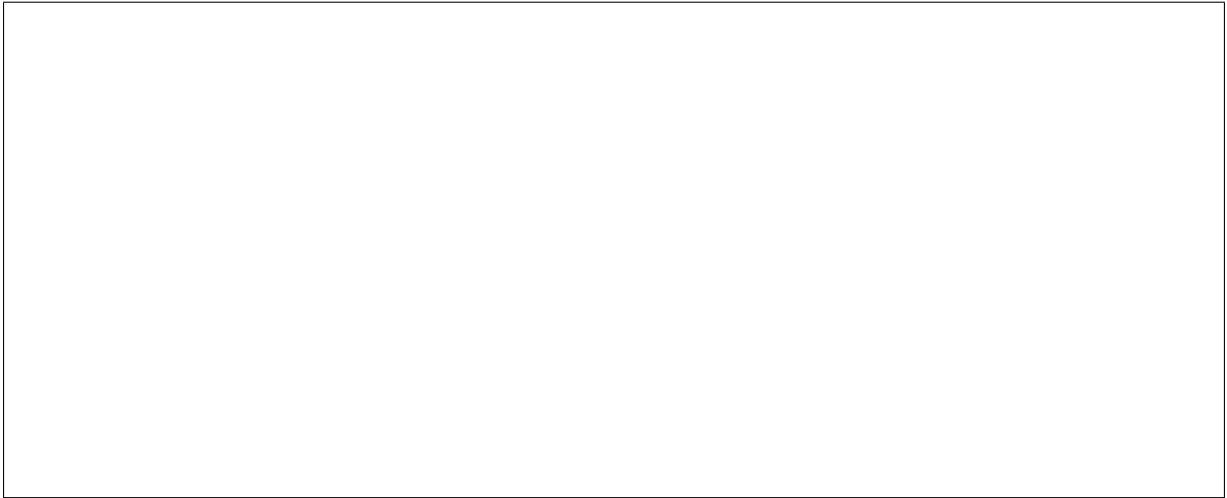
(continues on next page)

(continued from previous page)



(continues on next page)

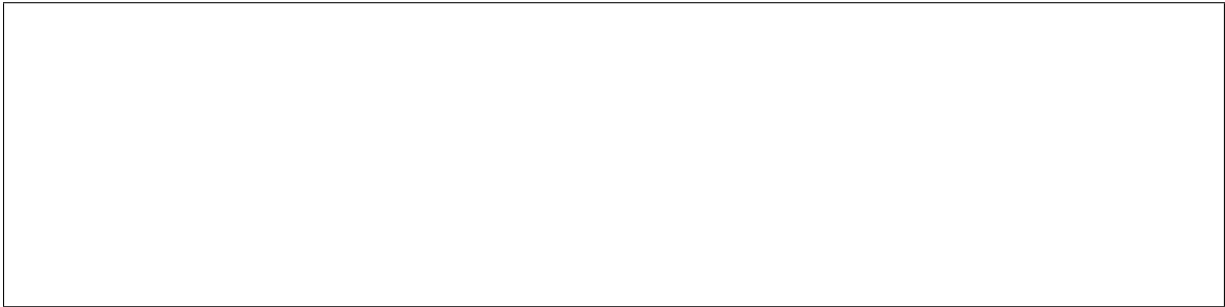
(continued from previous page)



**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**





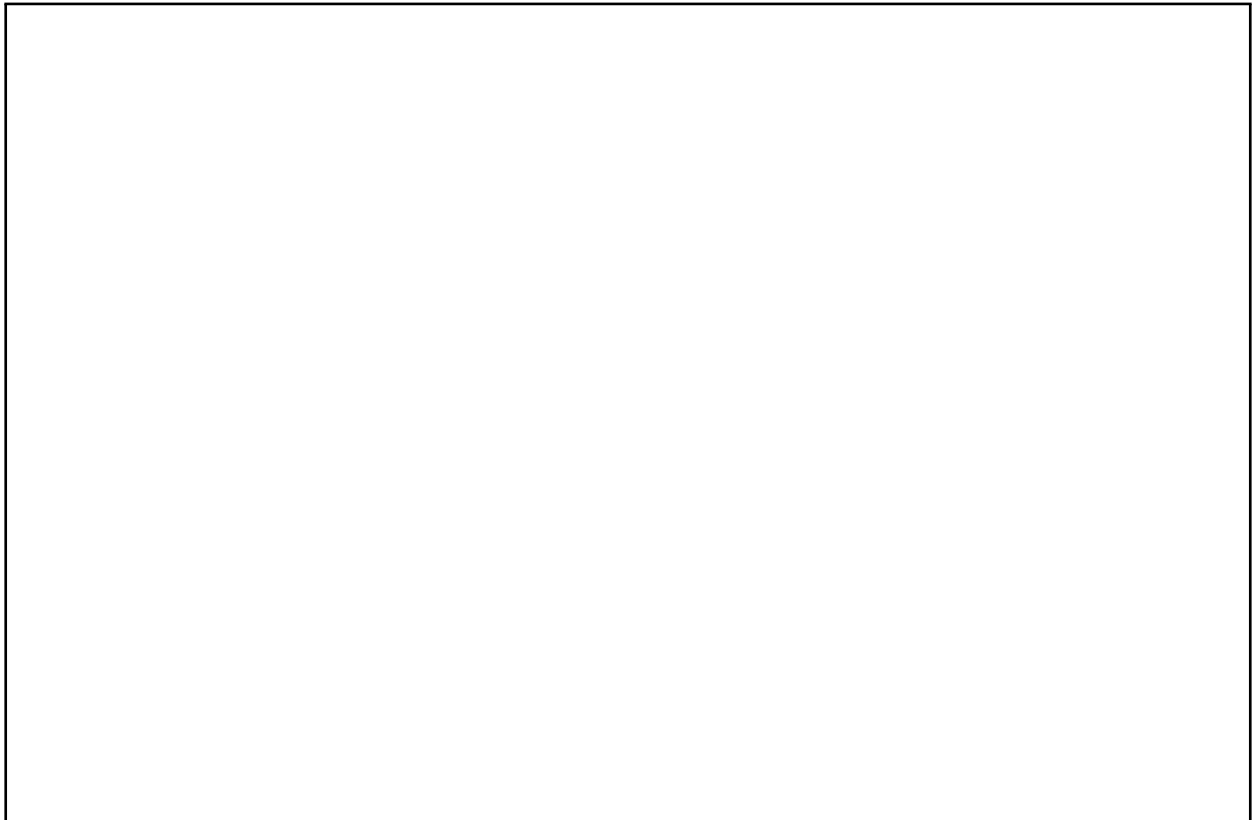
## **Structure of a GMR**

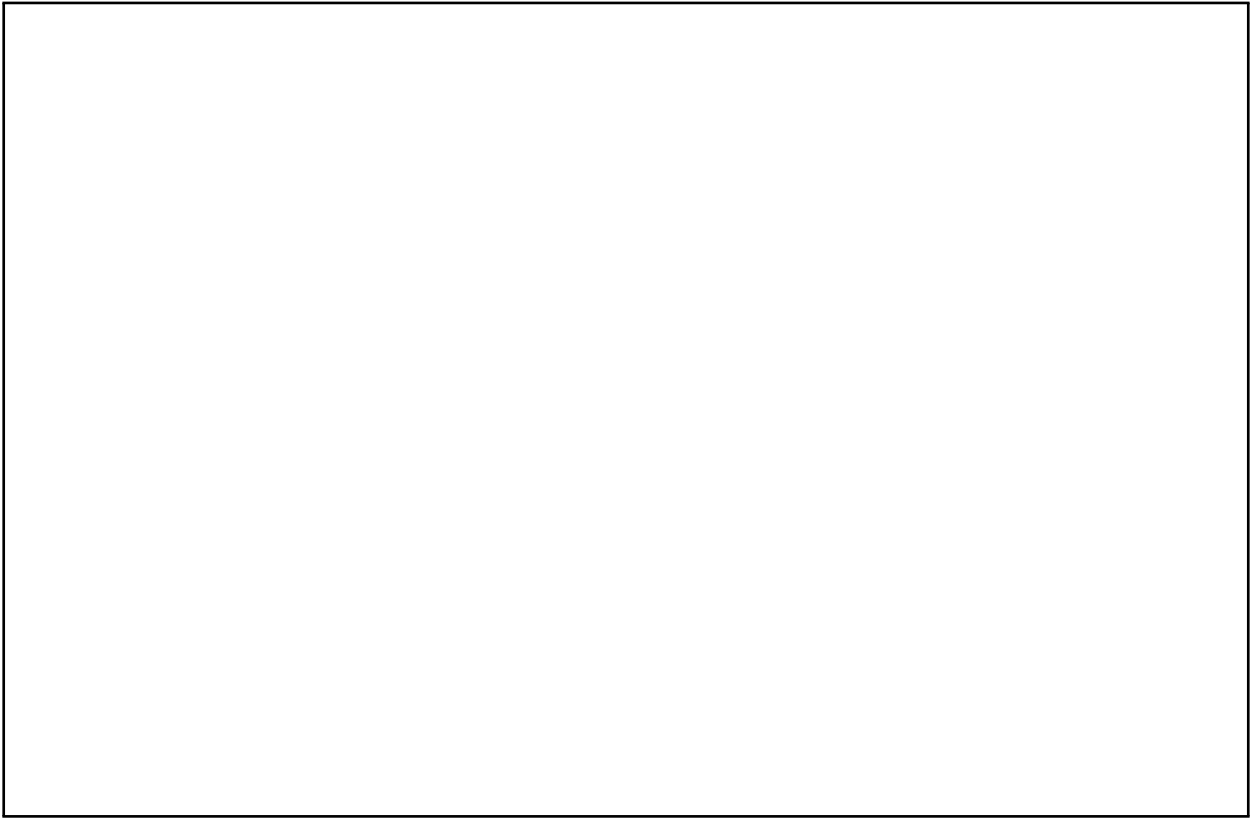


**Agent Token**

**Purpose**

ken can be viewed as a session identifier or authentication token.





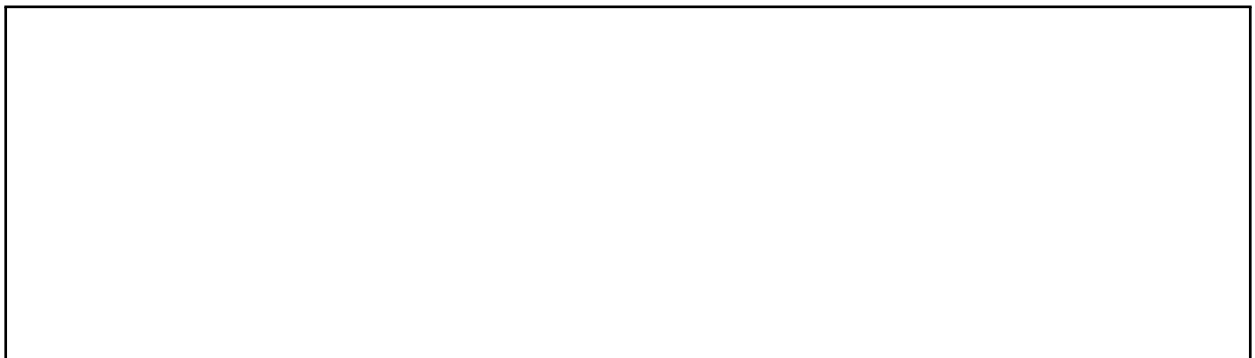
## **How it works**

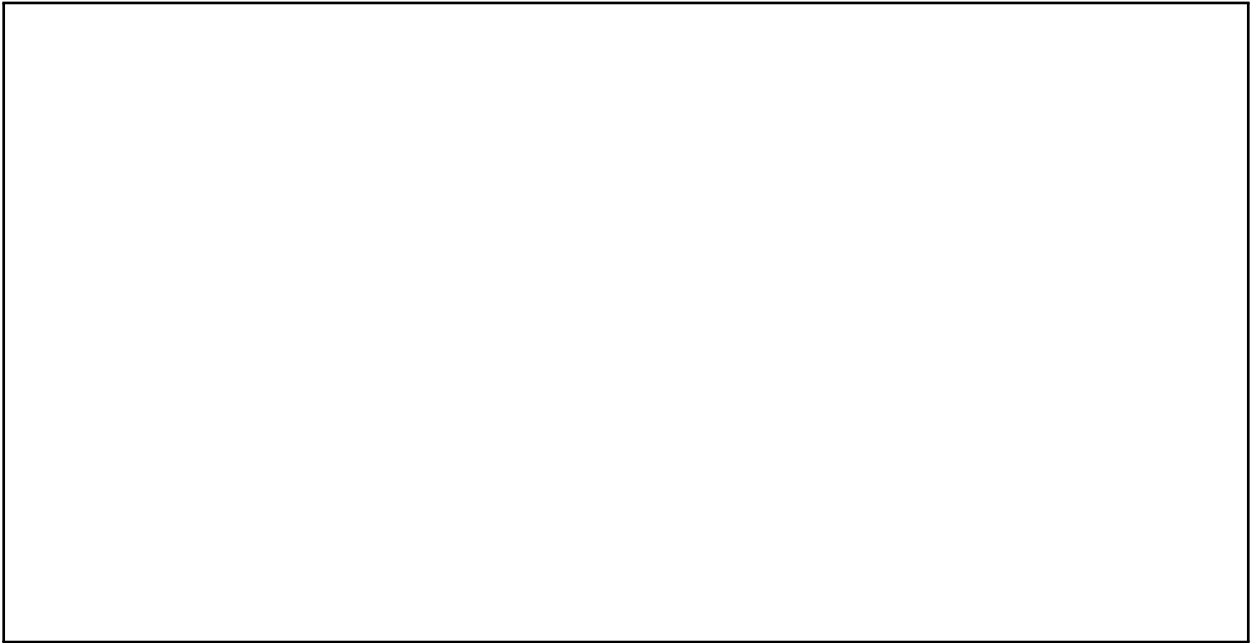


**Note:** In the case of the token being embedded with virtual media, it is read from a configuration file with-in the image. Ideally this should be paired with Swift temporary URLs.



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**

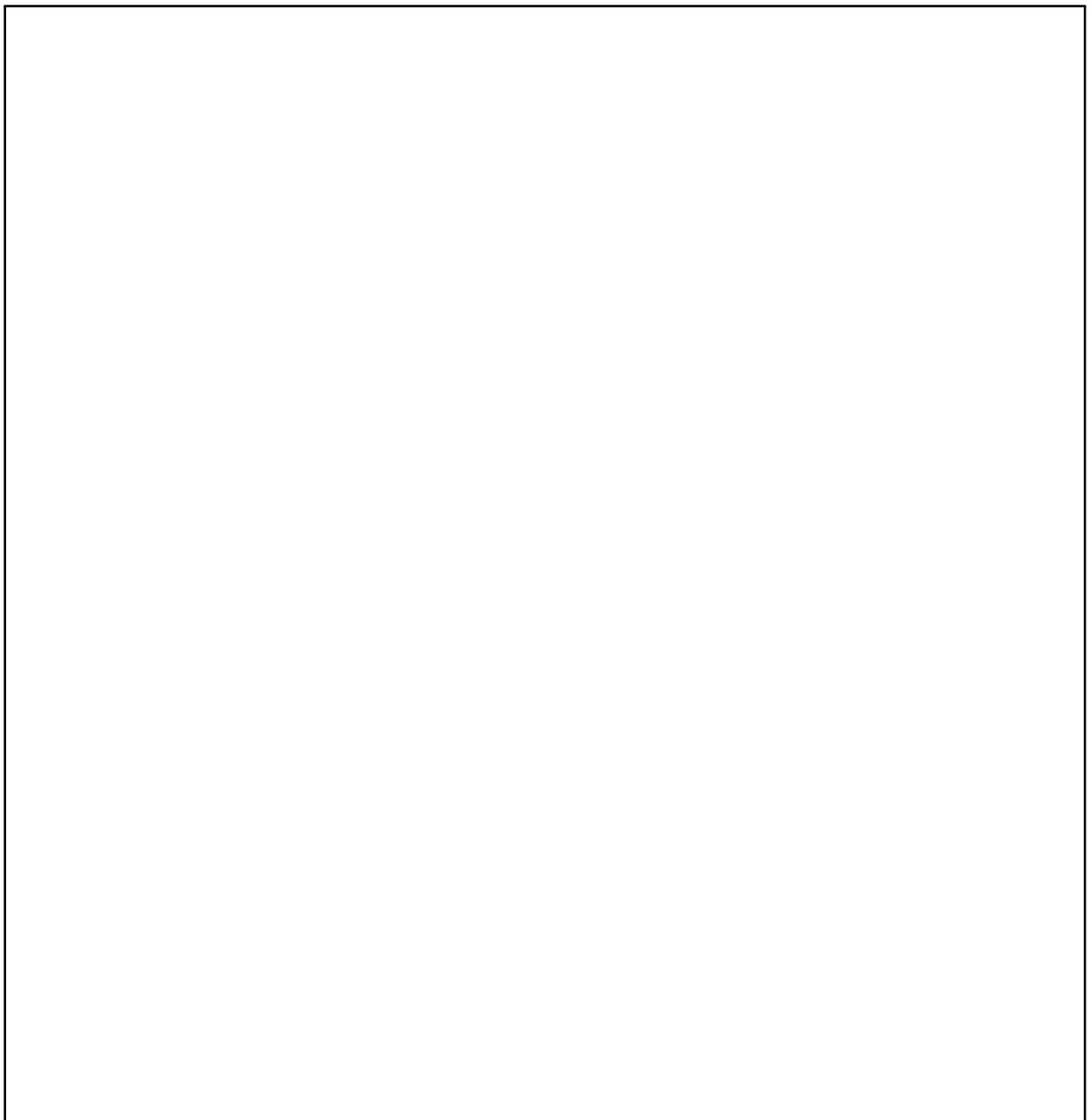
**With PXE/iPXE/etc.**

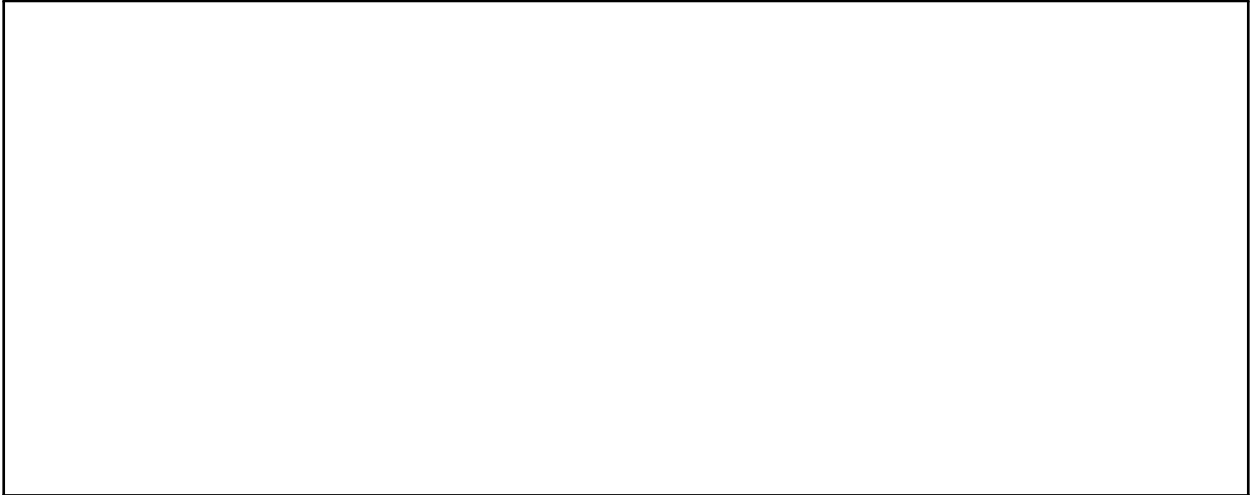
## **Agent Configuration**

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**







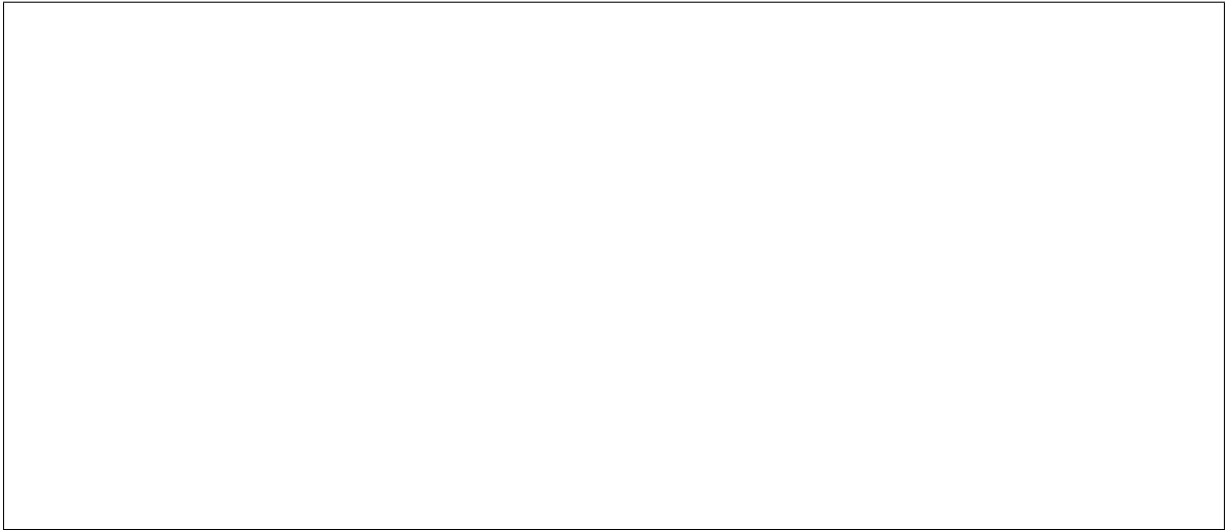


## **Enabling**



(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.



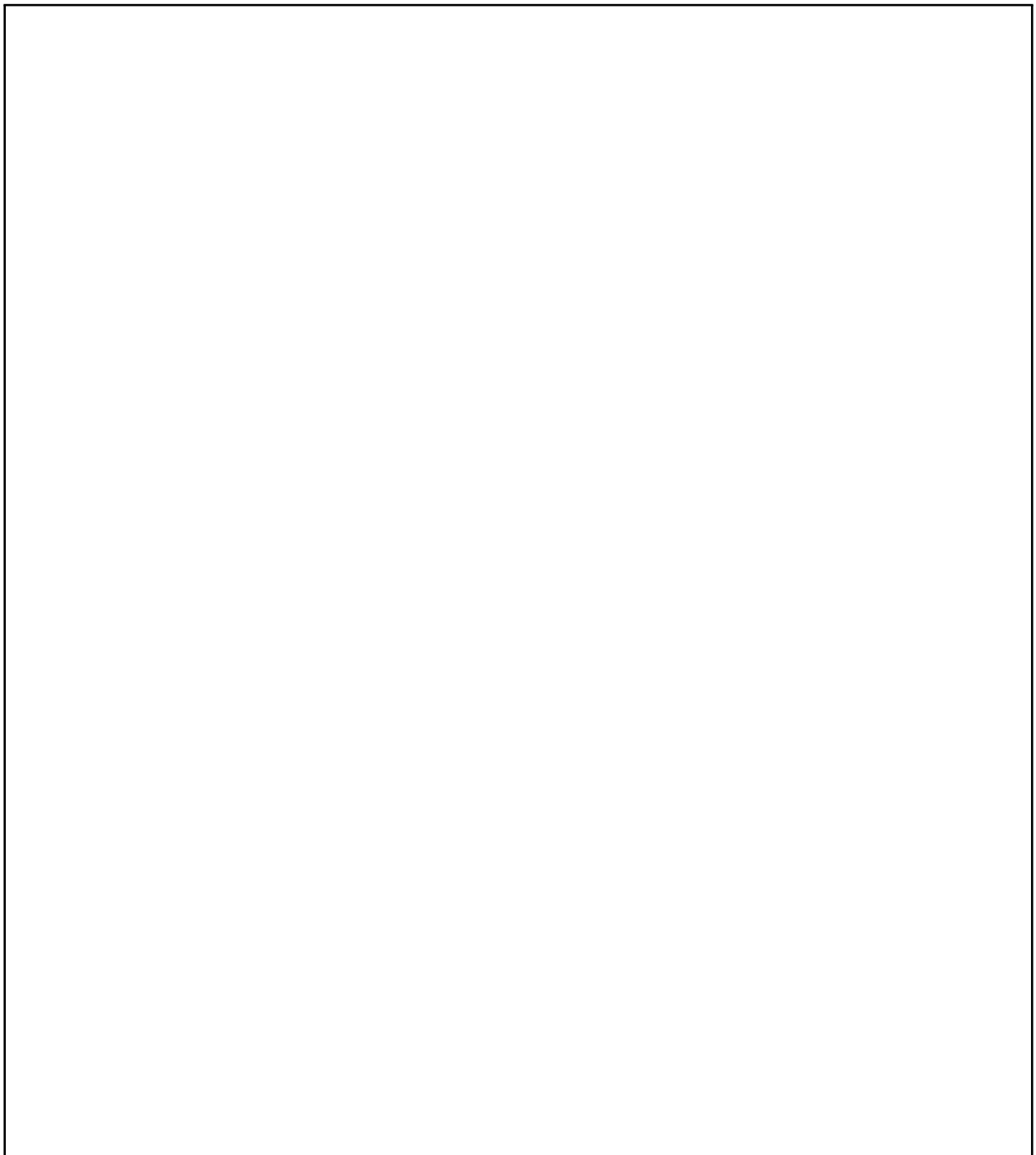
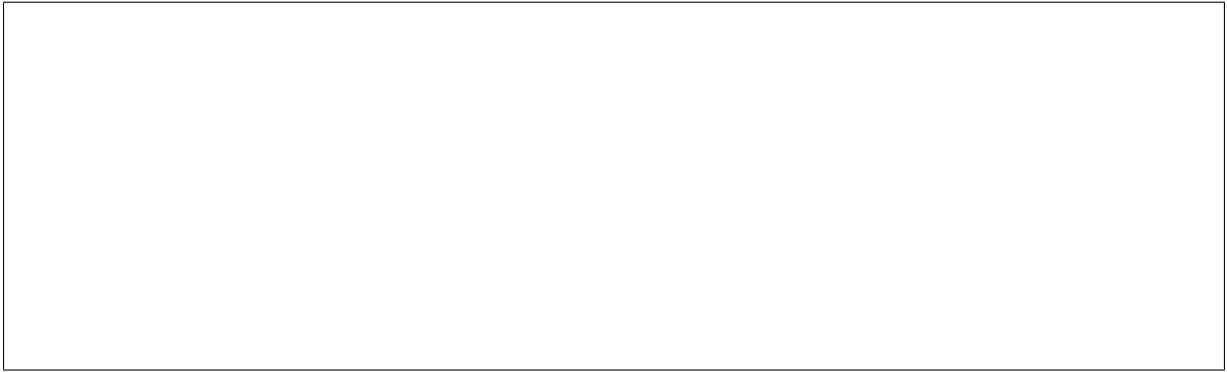


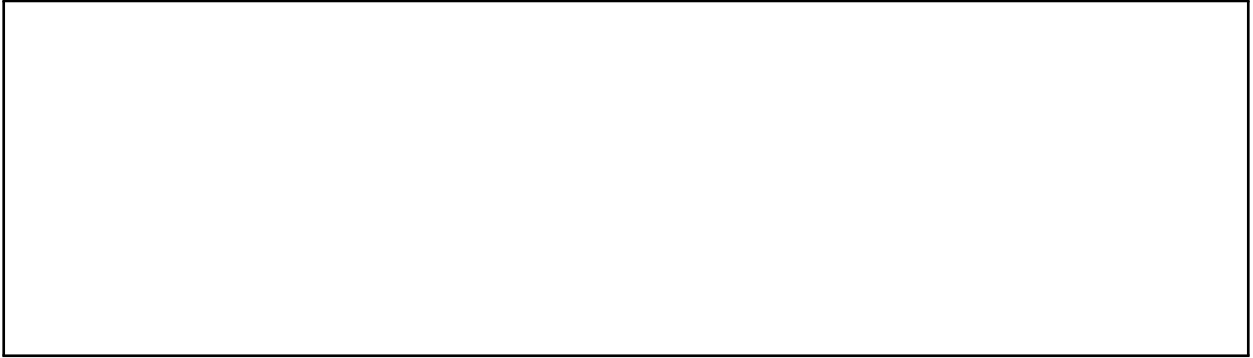
be configured accordingly.



(continues on next page)

(continued from previous page)



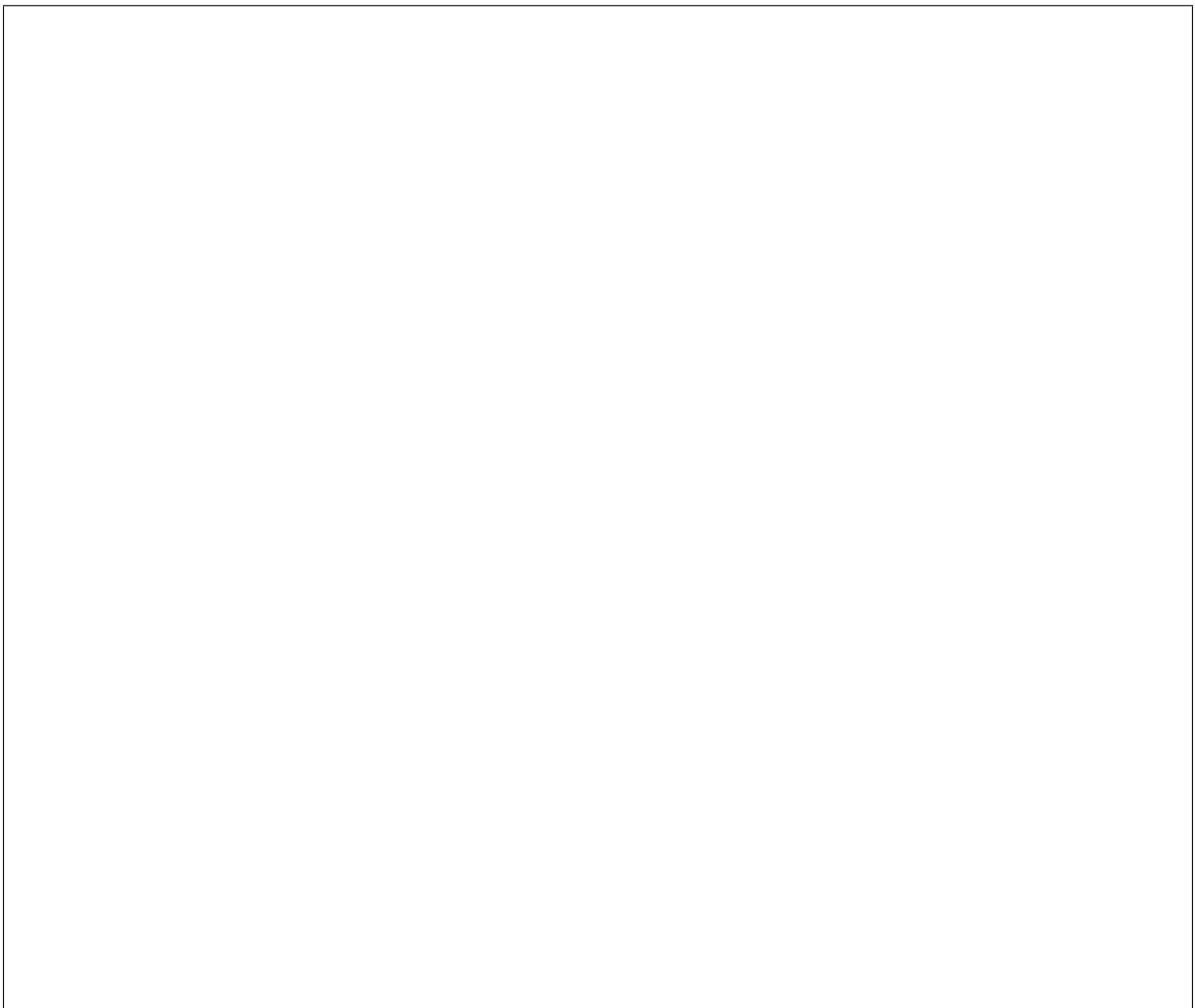
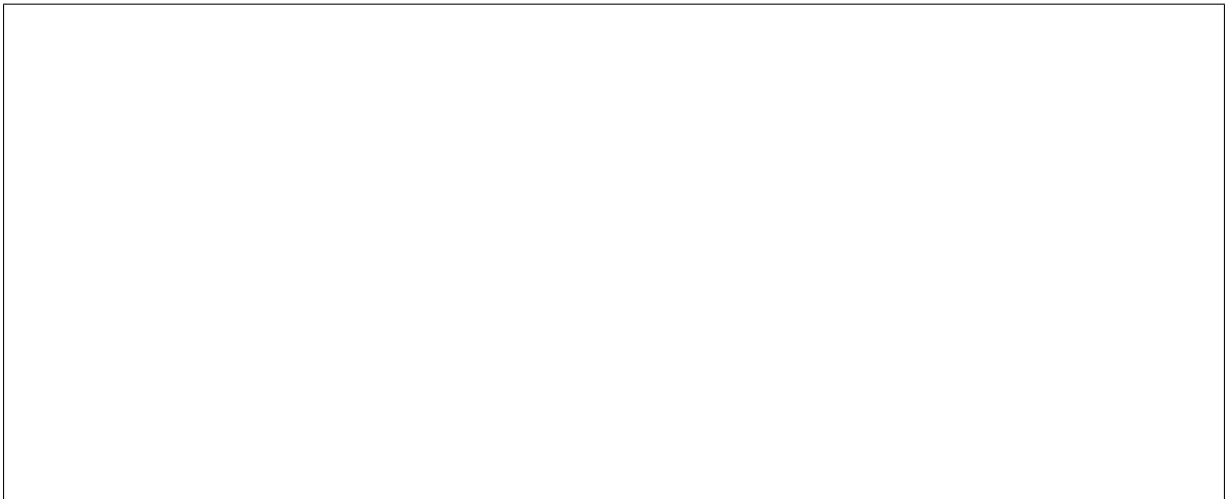


**Note:** If desired, some interfaces can still be configured to use DHCP.

### **Hardware type support**

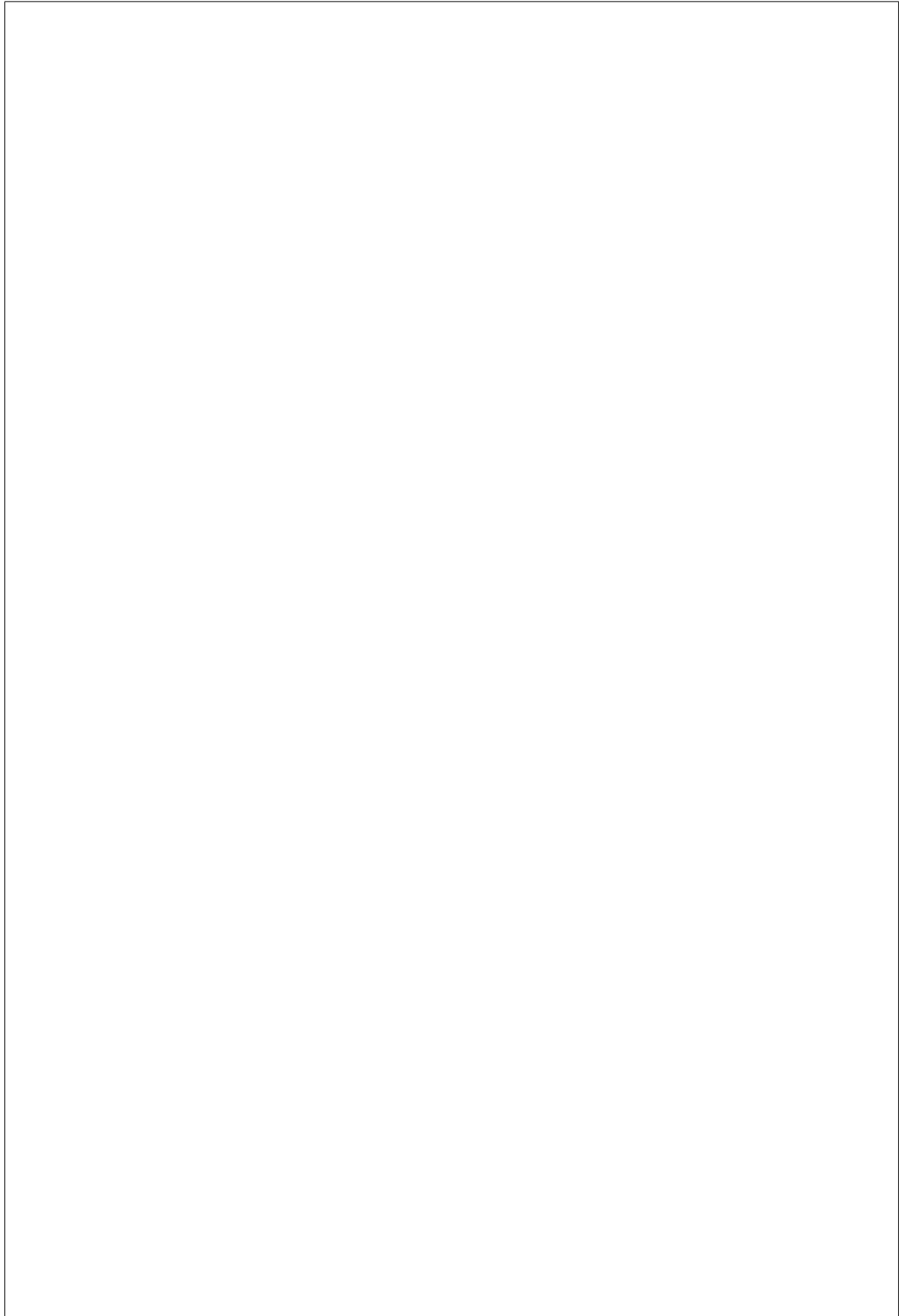
## **Configuring network data**

work service and also works in standalone mode.



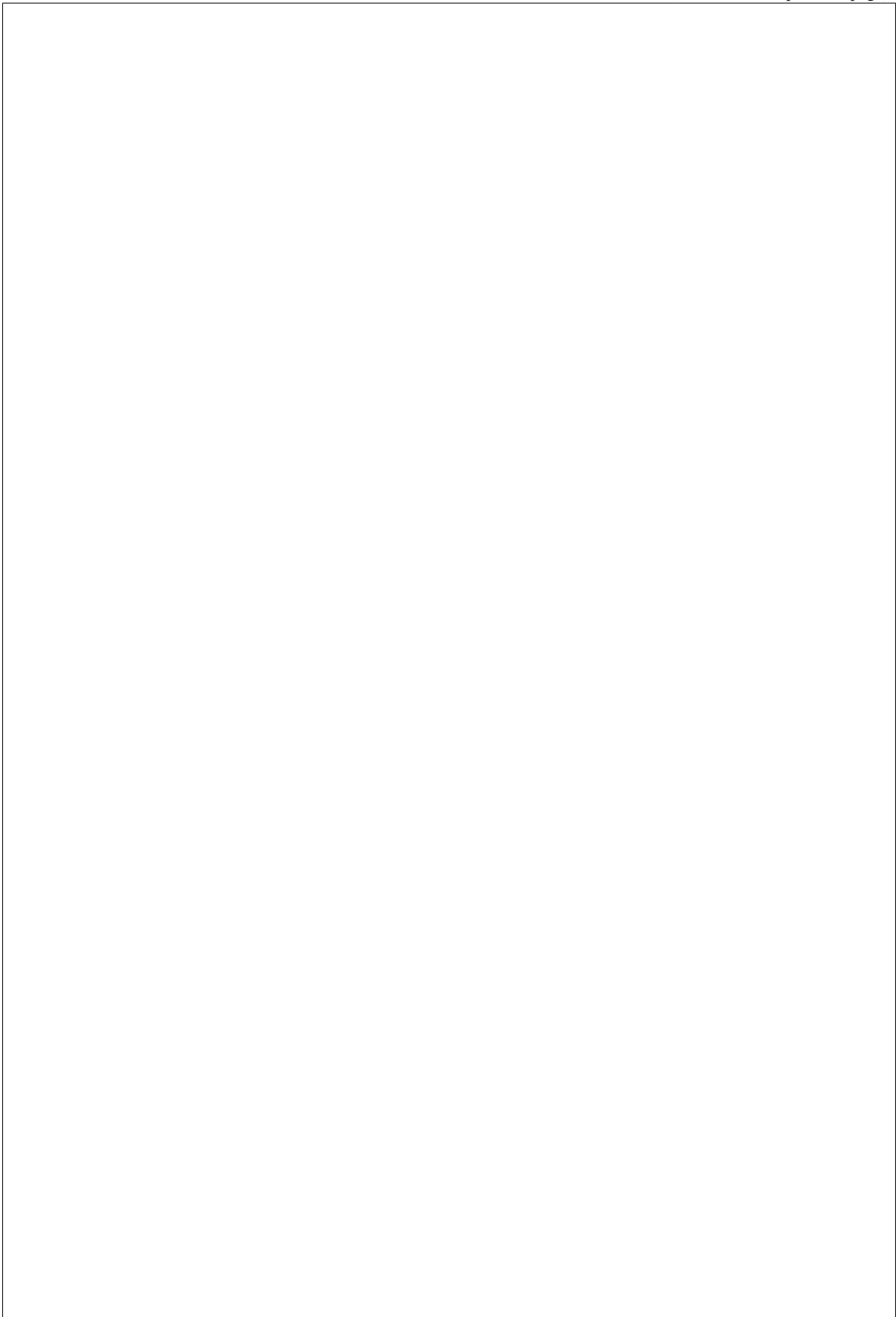
(continues on next page)

(continued from previous page)



(continues on next page)

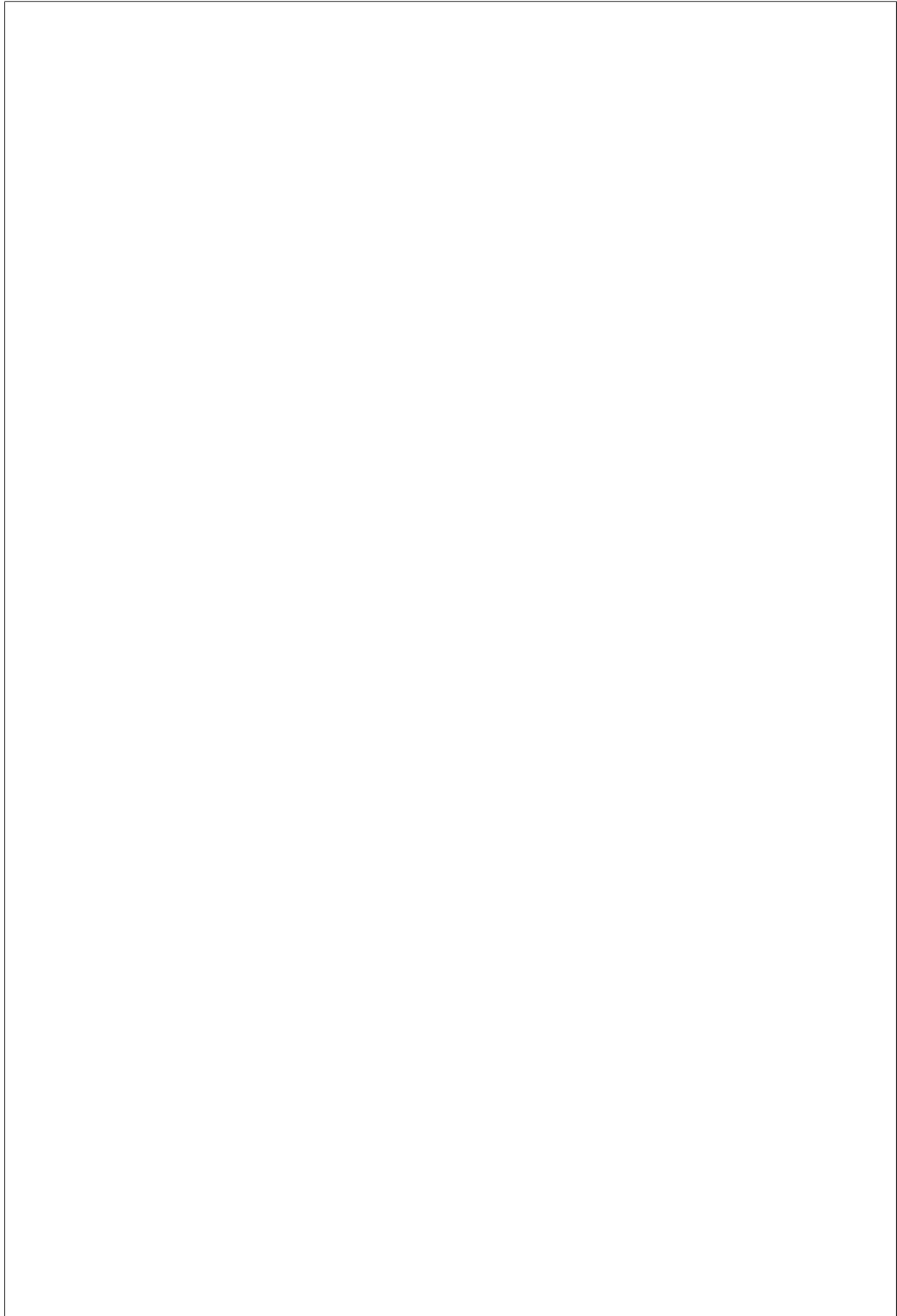
(continued from previous page)



(continues on next page)



(continued from previous page)



(continues on next page)

(continued from previous page)



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

## **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.

### **rpc\_conn\_pool\_size**

**Type** integer

**Default** 30

**Minimum Value** 1

Size of RPC connection pool.

Table 4: Deprecatcd 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: Deprecatcd 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`

#### **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.

**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.

### **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.

## audit

### enabled

**Type** boolean

**Default** False

Enable auditing of API requests (for ironic-api service).

### audit\_map\_file

**Type** string

**Default** /etc/ironic/api\_audit\_map.conf

Path to audit map file for ironic-api service. Used only when API audit is enabled.

### ignore\_req\_list

**Type** string

**Default** ''

Comma separated list of Ironic REST API HTTP methods to be ignored during audit logging. For example: auditing will not be done on any GET or POST requests if this is set to GET,POST. It is used only when API audit is enabled.

## cinder

### action\_retries

**Type** integer

**Default** 3

Number of retries in the case of a failed action (currently only used when detaching volumes).

### action\_retry\_interval

**Type** integer

**Default** 5

Retry interval in seconds in the case of a failed action (only specific actions are retried).

### auth\_url

**Type** unknown type

**Default** <None>

Authentication URL

### auth\_type

**Type** unknown type

**Default** <None>

Authentication type to load

Table 6: Deprecated Variations

Group	Name
cinder	auth_plugin

#### **cafile**

**Type** string

**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

#### **certfile**

**Type** string

**Default** <None>

PEM encoded client certificate cert file

#### **collect\_timing**

**Type** boolean

**Default** False

Collect per-API call timing information.

#### **connect\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for connection errors.

#### **connect\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

#### **default\_domain\_id**

**Type** unknown type

**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **default\_domain\_name**

**Type** unknown type

**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **domain\_id**

**Type** unknown type

**Default** <None>

Domain ID to scope to

#### **domain\_name**

**Type** unknown type

**Default** <None>

Domain name to scope to

#### **endpoint\_override**

**Type** string

**Default** <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

#### **insecure**

**Type** boolean

**Default** False

Verify HTTPS connections.

#### **keyfile**

**Type** string

**Default** <None>

PEM encoded client certificate key file

#### **max\_version**

**Type** string

**Default** <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min\_version. Mutually exclusive with version.

#### **min\_version**

**Type** string

**Default** <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max\_version. Mutually exclusive with version. If min\_version is given with no max\_version it is as if max version is latest.

#### **password**

**Type** unknown type

**Default** <None>

Users password

### **project\_domain\_id**

**Type** unknown type

**Default** <None>

Domain ID containing project

### **project\_domain\_name**

**Type** unknown type

**Default** <None>

Domain name containing project

### **project\_id**

**Type** unknown type

**Default** <None>

Project ID to scope to

Table 7: Deprecated Variations

Group	Name
cinder	tenant-id
cinder	tenant_id

### **project\_name**

**Type** unknown type

**Default** <None>

Project name to scope to

Table 8: Deprecated Variations

Group	Name
cinder	tenant-name
cinder	tenant_name

### **region\_name**

**Type** string

**Default** <None>

The default region\_name for endpoint URL discovery.

### **retries**

**Type** integer

**Default** 3

Client retries in the case of a failed request connection.

### **service\_name**

**Type** string



**Default** <None>

The default service\_name for endpoint URL discovery.

#### **service\_type**

**Type** string

**Default** volumev3

The default service\_type for endpoint URL discovery.

#### **split\_loggers**

**Type** boolean

**Default** False

Log requests to multiple loggers.

#### **status\_code\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

#### **status\_code\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

#### **system\_scope**

**Type** unknown type

**Default** <None>

Scope for system operations

#### **tenant\_id**

**Type** unknown type

**Default** <None>

Tenant ID

#### **tenant\_name**

**Type** unknown type

**Default** <None>

Tenant Name

#### **timeout**

**Type** integer

**Default** <None>

Timeout value for http requests

### **trust\_id**

**Type** unknown type

**Default** <None>

Trust ID

### **user\_domain\_id**

**Type** unknown type

**Default** <None>

Users domain id

### **user\_domain\_name**

**Type** unknown type

**Default** <None>

Users domain name

### **user\_id**

**Type** unknown type

**Default** <None>

User id

### **username**

**Type** unknown type

**Default** <None>

Username

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 Ironiic should try syncing the hardware node power state with the node power state in DB

**sync\_power\_state\_workers**

**Type** integer

**Default** 8

**Minimum Value** 1

The maximum number of worker threads that can be started simultaneously to sync nodes power states from the periodic task.

**periodic\_max\_workers**

**Type** integer

**Default** 8

Maximum number of worker threads that can be started simultaneously by a periodic task. Should be less than RPC thread pool size.

**node\_locked\_retry\_attempts**

**Type** integer

**Default** 3

Number of attempts to grab a node lock.

**node\_locked\_retry\_interval**

**Type** integer

**Default** 1

Seconds to sleep between node lock attempts.

#### **send\_sensor\_data**

**Type** boolean

**Default** False

Enable sending sensor data message via the notification bus

#### **send\_sensor\_data\_interval**

**Type** integer

**Default** 600

**Minimum Value** 1

Seconds between conductor sending sensor data message to ceilometer via the notification bus.

#### **send\_sensor\_data\_workers**

**Type** integer

**Default** 4

**Minimum Value** 1

The maximum number of workers that can be started simultaneously for send data from sensors periodic task.

#### **send\_sensor\_data\_wait\_timeout**

**Type** integer

**Default** 300

The time in seconds to wait for send sensors data periodic task to be finished before allowing periodic call to happen again. Should be less than send\_sensor\_data\_interval value.

#### **send\_sensor\_data\_types**

**Type** list

**Default** [ 'ALL' ]

List of comma separated meter types which need to be sent to Ceilometer. The default value, ALL, is a special value meaning send all the sensor data.

#### **send\_sensor\_data\_for\_undeployed\_nodes**

**Type** boolean

**Default** False

The default for sensor data collection is to only collect data for machines that are deployed, however operators may desire to know if there are failures in hardware that is not presently in use. When set to true, the conductor will collect sensor information from all nodes when sensor data collection is enabled via the send\_sensor\_data setting.

#### **sync\_local\_state\_interval**

**Type** integer

**Default** 180

When conductors join or leave the cluster, existing conductors may need to update any persistent local state as nodes are moved around the cluster. This option controls how often, in seconds, each conductor will check for nodes that it should take over. Set it to 0 (or a negative value) to disable the check entirely.

#### **configdrive\_swift\_container**

**Type** string

**Default** ironic\_configdrive\_container

Name of the Swift container to store config drive data. Used when configdrive\_use\_object\_store is True.

#### **configdrive\_swift\_temp\_url\_duration**

**Type** integer

**Default** <None>

**Minimum Value** 60

The timeout (in seconds) after which a configdrive temporary URL becomes invalid. Defaults to deploy\_callback\_timeout if it is set, otherwise to 1800 seconds. Used when configdrive\_use\_object\_store is True.

#### **inspect\_wait\_timeout**

**Type** integer

**Default** 1800

**Minimum Value** 0

Timeout (seconds) for waiting for node inspection. 0 - unlimited.

#### **automated\_clean**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

Enables or disables automated cleaning. Automated cleaning is a configurable set of steps, such as erasing disk drives, that are performed on the node to ensure it is in a baseline state and ready to be deployed to. This is done after instance deletion as well as during the transition from a manageable to available state. When enabled, the particular steps performed to clean a node depend on which driver that node is managed by; see the individual drivers documentation for details. NOTE: The introduction of the cleaning operation causes instance deletion to take significantly longer. In an environment where all tenants are trusted (eg, because there is only one tenant), this option could be safely disabled.

#### **allow\_provisioning\_in\_maintenance**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

Whether to allow nodes to enter or undergo deploy or cleaning when in maintenance mode. If this option is set to False, and a node enters maintenance during deploy or cleaning, the process will be aborted after the next heartbeat. Automated cleaning or making a node available will also fail. If True (the default), the process will begin and will pause after the node starts heartbeating. Moving it from maintenance will make the process continue.

#### **clean\_callback\_timeout**

**Type** integer

**Default** 1800

**Minimum Value** 0

Timeout (seconds) to wait for a callback from the ramdisk doing the cleaning. If the timeout is reached the node will be put in the clean failed provision state. Set to 0 to disable timeout.

#### **rescue\_callback\_timeout**

**Type** integer

**Default** 1800

**Minimum Value** 0

Timeout (seconds) to wait for a callback from the rescue ramdisk. If the timeout is reached the node will be put in the rescue failed provision state. Set to 0 to disable timeout.

#### **soft\_power\_off\_timeout**

**Type** integer

**Default** 600

**Minimum Value** 1

**Mutable** This option can be changed without restarting.

Timeout (in seconds) of soft reboot and soft power off operation. This value always has to be positive.

#### **power\_state\_change\_timeout**

**Type** integer

**Default** 60

**Minimum Value** 2

**Mutable** This option can be changed without restarting.

Number of seconds to wait for power operations to complete, i.e., so that a baremetal node is in the desired power state. If timed out, the power operation is considered a failure.

#### **power\_failure\_recovery\_interval**

**Type** integer

**Default** 300

**Minimum Value** 0

Interval (in seconds) between checking the power state for nodes previously put into maintenance mode due to power synchronization failure. A node is automatically moved out of maintenance mode once its power state is retrieved successfully. Set to 0 to disable this check.

**conductor\_group**

**Type** string

**Default** ''

Name of the conductor group to join. Can be up to 255 characters and is case insensitive. This conductor will only manage nodes with a matching conductor\_group field set on the node.

**allow\_deleting\_available\_nodes**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

Allow deleting nodes which are in state available. Defaults to True.

**enable\_mdns**

**Type** boolean

**Default** False

Whether to enable publishing the baremetal API endpoint via multicast DNS.

**deploy\_kernel**

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the kernel of the default deploy image.

**deploy\_ramdisk**

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the initramfs of the default deploy image.

**rescue\_kernel**

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the kernel of the default rescue image.

**rescue\_ramdisk**

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Glance ID, [http://](#) or [file://](#) URL of the initramfs of the default rescue image.

**rescue\_password\_hash\_algorithm**



**Type** string

**Default** sha256

**Valid Values** sha256, sha512

**Mutable** This option can be changed without restarting.

Password hash algorithm to be used for the rescue password.

#### **require\_rescue\_password\_hashed**

**Type** boolean

**Default** False

**Mutable** This option can be changed without restarting.

Option to cause the conductor to not fallback to an un-hashed version of the rescue password, permitting rescue with older ironic-python-agent ramdisks.

#### **bootloader**

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Glance ID, <http://> or <file://> URL of the EFI system partition image containing EFI boot loader. This image will be used by ironic when building UEFI-bootable ISO out of kernel and ramdisk. Required for UEFI boot from partition images.

### **console**

#### **terminal**

**Type** string

**Default** shellinaboxd

Path to serial console terminal program. Used only by Shell In A Box console.

#### **terminal\_cert\_dir**

**Type** string

**Default** <None>

Directory containing the terminal SSL cert (PEM) for serial console access. Used only by Shell In A Box console.

#### **terminal\_pid\_dir**

**Type** string

**Default** <None>

Directory for holding terminal pid files. If not specified, the temporary directory will be used.

#### **terminal\_timeout**

**Type** integer

**Default** 600

**Minimum Value 0**

Timeout (in seconds) for the terminal session to be closed on inactivity. Set to 0 to disable timeout. Used only by Socat console.

**subprocess\_checking\_interval**

**Type** integer

**Default** 1

Time interval (in seconds) for checking the status of console subprocess.

**subprocess\_timeout**

**Type** integer

**Default** 10

Time (in seconds) to wait for the console subprocess to start.

**kill\_timeout**

**Type** integer

**Default** 1

Time (in seconds) to wait for the console subprocess to exit before sending SIGKILL signal.

**socat\_address**

**Type** ip address

**Default** \$my\_ip

IP address of Socat service running on the host of ironiC conductor. Used only by Socat console.

**port\_range**

**Type** string

**Default** 10000:20000

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

A range of ports available to be used for the console proxy service running on the host of ironiC conductor, in the form of <start>:<stop>. This option is used by both Shellinabox and Socat console

**cors**

**allowed\_origin**

**Type** list

**Default** <None>

Indicate whether this resource may be shared with the domain received in the requests origin header. Format: <protocol>://<host>[:<port>], no trailing slash. Example: <https://horizon.example.com>

**allow\_credentials**

**Type** boolean

**Default** True

Indicate that the actual request can include user credentials

#### **expose\_headers**

**Type** list

**Default** []

Indicate which headers are safe to expose to the API. Defaults to HTTP Simple Headers.

#### **max\_age**

**Type** integer

**Default** 3600

Maximum cache age of CORS preflight requests.

#### **allow\_methods**

**Type** list

**Default** ['OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE',  
'TRACE', 'PATCH']

Indicate which methods can be used during the actual request.

#### **allow\_headers**

**Type** list

**Default** []

Indicate which header field names may be used during the actual request.

### **database**

#### **sqlite\_synchronous**

**Type** boolean

**Default** True

If True, SQLite uses synchronous mode.

Table 10: Deprecated Variations

Group	Name
DEFAULT	sqlite_synchronous

#### **backend**

**Type** string

**Default** sqlalchemy

The back end to use for the database.

Table 11: Deprecated Variations

Group	Name
DEFAULT	db_backend

### **connection**

**Type** string

**Default** <None>

The SQLAlchemy connection string to use to connect to the database.

Table 12: Deprecated Variations

Group	Name
DEFAULT	sql_connection
DATABASE	sql_connection
sql	connection

### **slave\_connection**

**Type** string

**Default** <None>

The SQLAlchemy connection string to use to connect to the slave database.

### **mysql\_sql\_mode**

**Type** string

**Default** TRADITIONAL

The SQL mode to be used for MySQL sessions. This option, including the default, overrides any server-set SQL mode. To use whatever SQL mode is set by the server configuration, set this to no value. Example: mysql\_sql\_mode=

### **mysql\_enable\_ndb**

**Type** boolean

**Default** False

If True, transparently enables support for handling MySQL Cluster (NDB).

### **connection\_recycle\_time**

**Type** integer

**Default** 3600

Connections which have been present in the connection pool longer than this number of seconds will be replaced with a new one the next time they are checked out from the pool.

Table 13: Deprecated Variations

Group	Name
DATABASE	idle_timeout
database	idle_timeout
DEFAULT	sql_idle_timeout
DATABASE	sql_idle_timeout
sql	idle_timeout

## **max\_pool\_size**

**Type** integer

**Default** 5

Maximum number of SQL connections to keep open in a pool. Setting a value of 0 indicates no limit.

Table 14: Deprecated Variations

Group	Name
DEFAULT	sql_max_pool_size
DATABASE	sql_max_pool_size

## **max\_retries**

**Type** integer

**Default** 10

Maximum number of database connection retries during startup. Set to -1 to specify an infinite retry count.

Table 15: Deprecated Variations

Group	Name
DEFAULT	sql_max_retries
DATABASE	sql_max_retries

## **retry\_interval**

**Type** integer

**Default** 10

Interval between retries of opening a SQL connection.

Table 16: Deprecated Variations

Group	Name
DEFAULT	sql_retry_interval
DATABASE	reconnect_interval

## **max\_overflow**

**Type** integer

**Default** 50

If set, use this value for max\_overflow with SQLAlchemy.

Table 17: Deprecated Variations

Group	Name
DEFAULT	sql_max_overflow
DATABASE	sqlalchemy_max_overflow

## connection\_debug

**Type** integer

**Default** 0

**Minimum Value** 0

**Maximum Value** 100

Verbosity of SQL debugging information: 0=None, 100=Everything.

Table 18: Deprecated Variations

Group	Name
DEFAULT	sql_connection_debug

## connection\_trace

**Type** boolean

**Default** False

Add Python stack traces to SQL as comment strings.

Table 19: Deprecated Variations

Group	Name
DEFAULT	sql_connection_trace

## pool\_timeout

**Type** integer

**Default** <None>

If set, use this value for pool\_timeout with SQLAlchemy.

Table 20: Deprecated Variations

Group	Name
DATABASE	sqlalchemy_pool_timeout

## use\_db\_reconnect

**Type** boolean

**Default** False

Enable the experimental use of database reconnect on connection lost.

#### **db\_retry\_interval**

**Type** integer

**Default** 1

Seconds between retries of a database transaction.

#### **db\_inc\_retry\_interval**

**Type** boolean

**Default** True

If True, increases the interval between retries of a database operation up to db\_max\_retry\_interval.

#### **db\_max\_retry\_interval**

**Type** integer

**Default** 10

If db\_inc\_retry\_interval is set, the maximum seconds between retries of a database operation.

#### **db\_max\_retries**

**Type** integer

**Default** 20

Maximum retries in case of connection error or deadlock error before error is raised. Set to -1 to specify an infinite retry count.

#### **connection\_parameters**

**Type** string

**Default** ''

Optional URL parameters to append onto the connection URL at connect time; specify as param1=value1&param2=value2&

#### **mysql\_engine**

**Type** string

**Default** InnoDB

MySQL engine to use.

### **deploy**

#### **http\_url**

**Type** string

**Default** <None>

ironic-conductor nodes HTTP server URL. Example: <http://192.1.2.3:8080>

#### **http\_root**

**Type** string

**Default** /httpboot

ironic-conductor nodes HTTP root path.

**enable\_ata\_secure\_erase**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

Whether to support the use of ATA Secure Erase during the cleaning process. Defaults to True.

**erase\_devices\_priority**

**Type** integer

**Default** <None>

**Mutable** This option can be changed without restarting.

Priority to run in-band erase devices via the IroniC Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 10 for the GenericHardwareManager). If set to 0, will not run during cleaning.

**erase\_devices\_metadata\_priority**

**Type** integer

**Default** <None>

**Mutable** This option can be changed without restarting.

Priority to run in-band clean step that erases metadata from devices, via the IroniC Python Agent ramdisk. If unset, will use the priority set in the ramdisk (defaults to 99 for the GenericHardwareManager). If set to 0, will not run during cleaning.

**shred\_random\_overwrite\_iterations**

**Type** integer

**Default** 1

**Minimum Value** 0

**Mutable** This option can be changed without restarting.

During shred, overwrite all block devices N times with random data. This is only used if a device could not be ATA Secure Erased. Defaults to 1.

**shred\_final\_overwrite\_with\_zeros**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

Whether to write zeros to a nodes block devices after writing random data. This will write zeros to the device even when `deploy.shred_random_overwrite_iterations` is 0. This option is only used if a device could not be ATA Secure Erased. Defaults to True.

**continue\_if\_disk\_secure\_erase\_fails**

**Type** boolean

**Default** False



**Mutable** This option can be changed without restarting.

Defines what to do if an ATA secure erase operation fails during cleaning in the Ironic Python Agent. If False, the cleaning operation will fail and the node will be put in `clean failed` state. If True, shred will be invoked and cleaning will continue.

#### **disk\_erasure\_concurrency**

**Type** integer

**Default** 1

**Minimum Value** 1

**Mutable** This option can be changed without restarting.

Defines the target pool size used by Ironic Python Agent ramdisk to erase disk devices. The number of threads created to erase disks will not exceed this value or the number of disks to be erased.

#### **power\_off\_after\_deploy\_failure**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

Whether to power off a node after deploy failure. Defaults to True.

#### **default\_boot\_option**

**Type** string

**Default** local

**Valid Values** netboot, local

**Mutable** This option can be changed without restarting.

Default boot option to use when no boot option is requested in nodes `driver_info`. Defaults to local. Prior to the Ussuri release, the default was netboot.

### **Possible values**

**netboot** boot from a network

**local** local boot

#### **default\_boot\_mode**

**Type** string

**Default** bios

**Valid Values** uefi, bios

**Mutable** This option can be changed without restarting.

Default boot mode to use when no boot mode is requested in nodes `driver_info`, capabilities or in the `instance_info` configuration. Currently the default boot mode is bios, but it will be changed to uefi in the future. It is recommended to set an explicit value for this option. This option only has effect when management interface supports boot mode management

## Possible values

**uefi** UEFI boot mode

**bios** Legacy BIOS boot mode

### **configdrive\_use\_object\_store**

**Type** boolean

**Default** False

**Mutable** This option can be changed without restarting.

Whether to upload the config drive to object store. Set this option to True to store config drive in a swift endpoint.

Table 21: Deprecated Variations

Group	Name
conductor	configdrive_use_swift

### **http\_image\_subdir**

**Type** string

**Default** agent\_images

The name of subdirectory under ironic-conductor nodes HTTP root path which is used to place instance images for the direct deploy interface, when local HTTP service is incorporated to provide instance image instead of swift tempurls.

### **fast\_track**

**Type** boolean

**Default** False

**Mutable** This option can be changed without restarting.

Whether to allow deployment agents to perform lookup, heartbeat operations during initial states of a machine lifecycle and by-pass the normal setup procedures for a ramdisk. This feature also enables power operations which are part of deployment processes to be bypassed if the ramdisk has performed a heartbeat operation using the fast\_track\_timeout setting.

### **fast\_track\_timeout**

**Type** integer

**Default** 300

**Minimum Value** 0

**Maximum Value** 300

**Mutable** This option can be changed without restarting.

Seconds for which the last heartbeat event is to be considered valid for the purpose of a fast track sequence. This setting should generally be less than the number of seconds for Power-On Self Test and typical ramdisk start-up. This value should not exceed the [api]ramdisk\_heartbeat\_timeout setting.

### **erase\_skip\_read\_only**

**Type** boolean

**Default** False

**Mutable** This option can be changed without restarting.

If the ironic-python-agent should skip read-only devices when running the `erase_devices` clean step where block devices are zeroed out. This requires ironic-python-agent 6.0.0 or greater. By default a read-only device will cause non-metadata based cleaning operations to fail due to the possible operational security risk of data being retained between deployments of the bare metal node.

## **dhcp**

### **dhcp\_provider**

**Type** string

**Default** neutron

DHCP provider to use. neutron uses Neutron, and none uses a no-op provider.

## **disk\_partitioner**

### **check\_device\_interval**

**Type** integer

**Default** 1

After Ironic has completed creating the partition table, it continues to check for activity on the attached iSCSI device status at this interval prior to copying the image to the node, in seconds

### **check\_device\_max\_retries**

**Type** integer

**Default** 20

The maximum number of times to check that the device is not accessed by another process. If the device is still busy after that, the disk partitioning will be treated as having failed.

## **disk\_utils**

### **efi\_system\_partition\_size**

**Type** integer

**Default** 200

Size of EFI system partition in MiB when configuring UEFI systems for local boot.

### **bios\_boot\_partition\_size**

**Type** integer

**Default** 1

Size of BIOS Boot partition in MiB when configuring GPT partitioned systems for local boot in BIOS.

**dd\_block\_size**

**Type** string

**Default** 1M

Block size to use when writing to the nodes disk.

**partition\_detection\_attempts**

**Type** integer

**Default** 3

**Minimum Value** 1

Maximum attempts to detect a newly created partition.

**partprobe\_attempts**

**Type** integer

**Default** 10

Maximum number of attempts to try to read the partition.

**image\_convert\_memory\_limit**

**Type** integer

**Default** 1024

Memory limit for qemu-img convert in MiB. Implemented via the address space resource limit.

**image\_convert\_attempts**

**Type** integer

**Default** 3

Number of attempts to convert an image.

**drac**

**query\_raid\_config\_job\_status\_interval**

**Type** integer

**Default** 120

**Minimum Value** 1

Interval (in seconds) between periodic RAID job status checks to determine whether the asynchronous RAID configuration was successfully finished or not.

**boot\_device\_job\_status\_timeout**

**Type** integer

**Default** 30

**Minimum Value** 1

Maximum amount of time (in seconds) to wait for the boot device configuration job to transition to the correct state to allow a reboot or power on to complete.

#### **config\_job\_max\_retries**

**Type** integer

**Default** 240

**Minimum Value** 1

Maximum number of retries for the configuration job to complete successfully.

### **glance**

#### **allowed\_direct\_url\_schemes**

**Type** list

**Default** []

A list of URL schemes that can be downloaded directly via the direct\_url. Currently supported schemes: [file].

#### **auth\_url**

**Type** unknown type

**Default** <None>

Authentication URL

#### **auth\_type**

**Type** unknown type

**Default** <None>

Authentication type to load

Table 22: Deprecated Variations

Group	Name
glance	auth_plugin

#### **cafile**

**Type** string

**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

#### **certfile**

**Type** string

**Default** <None>

PEM encoded client certificate cert file

#### **collect\_timing**

**Type** boolean

**Default** False

Collect per-API call timing information.

**connect\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for connection errors.

**connect\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

**default\_domain\_id**

**Type** unknown type

**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

**default\_domain\_name**

**Type** unknown type

**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

**domain\_id**

**Type** unknown type

**Default** <None>

Domain ID to scope to

**domain\_name**

**Type** unknown type

**Default** <None>

Domain name to scope to

**endpoint\_override**

**Type** string

**Default** <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

### **insecure**

**Type** boolean

**Default** False

Verify HTTPS connections.

### **keyfile**

**Type** string

**Default** <None>

PEM encoded client certificate key file

### **max\_version**

**Type** string

**Default** <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with min\_version. Mutually exclusive with version.

### **min\_version**

**Type** string

**Default** <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max\_version. Mutually exclusive with version. If min\_version is given with no max\_version it is as if max version is latest.

### **num\_retries**

**Type** integer

**Default** 0

Number of retries when downloading an image from glance.

### **password**

**Type** unknown type

**Default** <None>

Users password

### **project\_domain\_id**

**Type** unknown type

**Default** <None>

Domain ID containing project

### **project\_domain\_name**

**Type** unknown type

**Default** <None>

Domain name containing project

## **project\_id**

**Type** unknown type

**Default** <None>

Project ID to scope to

Table 23: Deprecatcd Variations

Group	Name
glance	tenant-id
glance	tenant_id

## **project\_name**

**Type** unknown type

**Default** <None>

Project name to scope to

Table 24: Deprecatcd Variations

Group	Name
glance	tenant-name
glance	tenant_name

## **region\_name**

**Type** string

**Default** <None>

The default region\_name for endpoint URL discovery.

## **service\_name**

**Type** string

**Default** <None>

The default service\_name for endpoint URL discovery.

## **service\_type**

**Type** string

**Default** image

The default service\_type for endpoint URL discovery.

## **split\_loggers**

**Type** boolean

**Default** False

Log requests to multiple loggers.

## **status\_code\_retries**

**Type** integer



**Default** <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

#### **status\_code\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

#### **swift\_account**

**Type** string

**Default** <None>

The account that Glance uses to communicate with Swift. The format is AUTH\_uuid. uuid is the UUID for the account configured in the glance-api.conf. For example: AUTH\_a422b2-91f3-2f46-74b7-d7c9e8958f5d30. If not set, the default value is calculated based on the ID of the project used to access Swift (as set in the [swift] section). Swift temporary URL format: endpoint\_url/api\_version/account/container/object\_id

#### **swift\_api\_version**

**Type** string

**Default** v1

The Swift API version to create a temporary URL for. Defaults to v1. Swift temporary URL format: endpoint\_url/api\_version/account/container/object\_id

#### **swift\_container**

**Type** string

**Default** glance

The Swift container Glance is configured to store its images in. Defaults to glance, which is the default in glance-api.conf. Swift temporary URL format: endpoint\_url/api\_version/account/container/object\_id

#### **swift\_endpoint\_url**

**Type** string

**Default** <None>

The endpoint (scheme, hostname, optional port) for the Swift URL of the form endpoint\_url/api\_version/account/container/object\_id. Do not include trailing /. For example, use <https://swift.example.com>. If using RADOS Gateway, endpoint may also contain /swift path; if it does not, it will be appended. Used for temporary URLs, will be fetched from the service catalog, if not provided.

#### **swift\_store\_multiple\_containers\_seed**

**Type** integer

**Default** 0

This should match a config by the same name in the Glance configuration file. When set to 0, a single-tenant store will only use one container to store all images. When set to an integer value between 1 and 32, a single-tenant store will use multiple containers to store images, and this value will determine how many containers are created.

**swift\_temp\_url\_cache\_enabled**

**Type** boolean

**Default** False

Whether to cache generated Swift temporary URLs. Setting it to true is only useful when an image caching proxy is used. Defaults to False.

**swift\_temp\_url\_duration**

**Type** integer

**Default** 1200

The length of time in seconds that the temporary URL will be valid for. Defaults to 20 minutes. If some deploys get a 401 response code when trying to download from the temporary URL, try raising this duration. This value must be greater than or equal to the value for `swift_temp_url_expected_download_start_delay`

**swift\_temp\_url\_expected\_download\_start\_delay**

**Type** integer

**Default** 0

**Minimum Value** 0

This is the delay (in seconds) from the time of the deploy request (when the Swift temporary URL is generated) to when the IPA ramdisk starts up and URL is used for the image download. This value is used to check if the Swift temporary URL duration is large enough to let the image download begin. Also if temporary URL caching is enabled this will determine if a cached entry will still be valid when the download starts. `swift_temp_url_duration` value must be greater than or equal to this options value. Defaults to 0.

**swift\_temp\_url\_key**

**Type** string

**Default** <None>

The secret token given to Swift to allow temporary URL downloads. Required for temporary URLs. For the Swift backend, the key on the service project (as set in the [swift] section) is used by default.

**system\_scope**

**Type** unknown type

**Default** <None>

Scope for system operations

**tenant\_id**

**Type** unknown type

**Default** <None>

Tenant ID

**tenant\_name**

**Type** unknown type

**Default** <None>

Tenant Name

**timeout**

**Type** integer

**Default** <None>

Timeout value for http requests

**trust\_id**

**Type** unknown type

**Default** <None>

Trust ID

**user\_domain\_id**

**Type** unknown type

**Default** <None>

Users domain id

**user\_domain\_name**

**Type** unknown type

**Default** <None>

Users domain name

**user\_id**

**Type** unknown type

**Default** <None>

User id

**username**

**Type** unknown type

**Default** <None>

Username

Table 25: Deprecated Variations

Group	Name
glance	user-name
glance	user_name

**valid\_interfaces**

**Type** list

**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

#### **version**

**Type** string

**Default** <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min\_version and max\_version

### **healthcheck**

#### **path**

**Type** string

**Default** /healthcheck

The path to respond to healthcheck requests on.

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

#### **detailed**

**Type** boolean

**Default** False

Show more detailed information as part of the response. Security note: Enabling this option may expose sensitive details about the service being monitored. Be sure to verify that it will not violate your security policies.

#### **backends**

**Type** list

**Default** []

Additional backends that can perform health checks and report that information back as part of a request.

#### **disable\_by\_file\_path**

**Type** string

**Default** <None>

Check the presence of a file to determine if an application is running on a port. Used by Disable-ByFileHealthcheck 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](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 00644 in Python. This setting must be set to the octal number representation, meaning starting with 00.

## **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.

## irmc

### **remote\_image\_share\_root**

**Type** string

**Default** /remote\_image\_share\_root

Ironic conductor nodes NFS or CIFS root path

### **remote\_image\_server**

**Type** string

**Default** <None>

IP of remote image server

### **remote\_image\_share\_type**

**Type** string

**Default** CIFS

**Valid Values** CIFS, NFS

Share type of virtual media

## Possible values

**CIFS** CIFS (Common Internet File System) protocol

**NFS** NFS (Network File System) protocol

### **remote\_image\_share\_name**

**Type** string

**Default** share

share name of remote\_image\_server

### **remote\_image\_user\_name**

**Type** string

**Default** <None>

User name of remote\_image\_server

### **remote\_image\_user\_password**

**Type** string

**Default** <None>

Password of remote\_image\_user\_name

**remote\_image\_user\_domain**

**Type** string

**Default** ''

Domain name of remote\_image\_user\_name

**port**

**Type** port number

**Default** 443

**Minimum Value** 0

**Maximum Value** 65535

**Valid Values** 443, 80

Port to be used for iRMC operations

**Possible values**

**443** port 443

**80** port 80

**auth\_method**

**Type** string

**Default** basic

**Valid Values** basic, digest

Authentication method to be used for iRMC operations

**Possible values**

**basic** Basic authentication

**digest** Digest authentication

**client\_timeout**

**Type** integer

**Default** 60

Timeout (in seconds) for iRMC operations

**sensor\_method**

**Type** string

**Default** ipmitool

**Valid Values** ipmitool, scci

Sensor data retrieval method.



### Possible values

**ipmitool** IPMItool

**scsi** Fujitsu SCCI (ServerView Common Command Interface)

#### **snmp\_version**

**Type** string

**Default** v2c

**Valid Values** v1, v2c, v3

SNMP protocol version

### Possible values

**v1** SNMPv1

**v2c** SNMPv2c

**v3** SNMPv3

#### **snmp\_port**

**Type** port number

**Default** 161

**Minimum Value** 0

**Maximum Value** 65535

SNMP port

#### **snmp\_community**

**Type** string

**Default** public

SNMP community. Required for versions v1 and v2c

#### **snmp\_security**

**Type** string

**Default** <None>

SNMP security name. Required for version v3

#### **snmp\_polling\_interval**

**Type** integer

**Default** 10

SNMP polling interval in seconds

#### **clean\_priority\_restore\_irmc\_bios\_config**

**Type** integer

**Default** 0

Priority for restore\_irmc\_bios\_config clean step.

#### gpu\_ids

**Type** list

**Default** []

List of vendor IDs and device IDs for GPU device to inspect. List items are in format vendorID/deviceID and separated by commas. GPU inspection will use this value to count the number of GPU device in a node. If this option is not defined, then leave out pci\_gpu\_devices in capabilities property. Sample gpu\_ids value: 0x1000/0x0079,0x2100/0x0080

#### fpga\_ids

**Type** list

**Default** []

List of vendor IDs and device IDs for CPU FPGA to inspect. List items are in format vendorID/deviceID and separated by commas. CPU inspection will use this value to find existence of CPU FPGA in a node. If this option is not defined, then leave out CUSTOM\_CPU\_FPGA in node traits. Sample fpga\_ids value: 0x1000/0x0079,0x2100/0x0080

#### query\_raid\_config\_fgi\_status\_interval

**Type** integer

**Default** 300

**Minimum Value** 1

Interval (in seconds) between periodic RAID status checks to determine whether the asynchronous RAID configuration was successfully finished or not. Foreground Initialization (FGI) will start 5 minutes after creating virtual drives.

### ironic\_lib

#### fatal\_exception\_format\_errors

**Type** boolean

**Default** False

Used if there is a formatting error when generating an exception message (a programming error). If True, raise an exception; if False, use the unformatted message.

Table 30: Deprecated Variations

Group	Name
DEFAULT	fatal_exception_format_errors

#### root\_helper

**Type** string

**Default** sudo ironic-rootwrap /etc/ironic/rootwrap.conf

Command that is prefixed to commands that are run as root. If not specified, no commands are run as root.

## iscsi

### portal\_port

**Type** port number

**Default** 3260

**Minimum Value** 0

**Maximum Value** 65535

**Mutable** This option can be changed without restarting.

The port number on which the iSCSI portal listens for incoming connections.

### conv\_flags

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Flags that need to be sent to the dd command, to control the conversion of the original file when copying to the host. It can contain several options separated by commas.

### verify\_attempts

**Type** integer

**Default** 3

**Minimum Value** 1

**Mutable** This option can be changed without restarting.

Maximum attempts to verify an iSCSI connection is active, sleeping 1 second between attempts. Defaults to 3.

## json\_rpc

### auth\_strategy

**Type** string

**Default** <None>

**Valid Values** noauth, keystone, http\_basic

Authentication strategy used by JSON RPC. Defaults to the global auth\_strategy setting.

### Possible values

**noauth** no authentication

**keystone** use the Identity service for authentication

**http\_basic** HTTP basic authentication

#### **http\_basic\_auth\_user\_file**

**Type** string

**Default** /etc/ironic/htpasswd-json-rpc

Path to Apache format user authentication file used when auth\_strategy=http\_basic

#### **host\_ip**

**Type** host address

**Default** ::

The IP address or hostname on which JSON RPC will listen.

#### **port**

**Type** port number

**Default** 8089

**Minimum Value** 0

**Maximum Value** 65535

The port to use for JSON RPC

#### **use\_ssl**

**Type** boolean

**Default** False

Whether to use TLS for JSON RPC

#### **http\_basic\_username**

**Type** string

**Default** <None>

Name of the user to use for HTTP Basic authentication client requests.

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

**Reason** Use username instead

#### **http\_basic\_password**

**Type** string

**Default** <None>

Password to use for HTTP Basic authentication client requests.

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

**Reason** Use password instead

#### **auth\_url**

**Type** unknown type

**Default** <None>

Authentication URL

#### **auth\_type**

**Type** unknown type

**Default** <None>

Authentication type to load

Table 31: Deprecated Variations

Group	Name
json_rpc	auth_plugin

#### **cafile**

**Type** string

**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

#### **certfile**

**Type** string

**Default** <None>

PEM encoded client certificate cert file

#### **collect\_timing**

**Type** boolean

**Default** False

Collect per-API call timing information.

#### **default\_domain\_id**

**Type** unknown type

**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **default\_domain\_name**

**Type** unknown type

**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

**domain\_id**

**Type** unknown type

**Default** <None>

Domain ID to scope to

**domain\_name**

**Type** unknown type

**Default** <None>

Domain name to scope to

**insecure**

**Type** boolean

**Default** False

Verify HTTPS connections.

**keyfile**

**Type** string

**Default** <None>

PEM encoded client certificate key file

**password**

**Type** unknown type

**Default** <None>

Users password

**project\_domain\_id**

**Type** unknown type

**Default** <None>

Domain ID containing project

**project\_domain\_name**

**Type** unknown type

**Default** <None>

Domain name containing project

**project\_id**

**Type** unknown type

**Default** <None>

Project ID to scope to

Table 32: Deprecated Variations

Group	Name
json_rpc	tenant-id
json_rpc	tenant_id

**project\_name**

**Type** unknown type

**Default** <None>

Project name to scope to

Table 33: Deprecated Variations

Group	Name
json_rpc	tenant-name
json_rpc	tenant_name

**split\_loggers**

**Type** boolean

**Default** False

Log requests to multiple loggers.

**system\_scope**

**Type** unknown type

**Default** <None>

Scope for system operations

**tenant\_id**

**Type** unknown type

**Default** <None>

Tenant ID

**tenant\_name**

**Type** unknown type

**Default** <None>

Tenant Name

**timeout**

**Type** integer

**Default** <None>

Timeout value for http requests

**trust\_id**

**Type** unknown type

**Default** <None>

Trust ID

**user\_domain\_id**

**Type** unknown type

**Default** <None>

Users domain id

**user\_domain\_name**

**Type** unknown type

**Default** <None>

Users domain name

**user\_id**

**Type** unknown type

**Default** <None>

User id

**username**

**Type** unknown type

**Default** <None>

Username

Table 34: Deprecated Variations

Group	Name
json_rpc	user-name
json_rpc	user_name

**keystone\_authtoken**

**www\_authenticate\_uri**

**Type** string

**Default** <None>

Complete public Identity API endpoint. This endpoint should not be an admin endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If youre using a versioned v2 endpoint here, then this should *not* be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint.



Table 35: Deprecated Variations

Group	Name
keystone_authtoken	auth_uri

## auth\_uri

**Type** string

**Default** <None>

Complete public Identity API endpoint. This endpoint should not be an admin endpoint, as it should be accessible by all end users. Unauthenticated clients are redirected to this endpoint to authenticate. Although this endpoint should ideally be unversioned, client support in the wild varies. If you're using a versioned v2 endpoint here, then this should *not* be the same endpoint the service user utilizes for validating tokens, because normal end users may not be able to reach that endpoint. This option is deprecated in favor of `www_authenticate_uri` and will be removed in the S release.

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

**Reason** The `auth_uri` option is deprecated in favor of `www_authenticate_uri` and will be removed in the S release.

## auth\_version

**Type** string

**Default** <None>

API version of the Identity API endpoint.

## interface

**Type** string

**Default** internal

Interface to use for the Identity API endpoint. Valid values are public, internal (default) or admin.

## delay\_auth\_decision

**Type** boolean

**Default** False

Do not handle authorization requests within the middleware, but delegate the authorization decision to downstream WSGI components.

## http\_connect\_timeout

**Type** integer

**Default** <None>

Request timeout value for communicating with Identity API server.

## http\_request\_max\_retries

**Type** integer

**Default** 3

How many times are we trying to reconnect when communicating with Identity API Server.

#### **cache**

**Type** string

**Default** <None>

Request environment key where the Swift cache object is stored. When `auth_token` middleware is deployed with a Swift cache, use this option to have the middleware share a caching backend with swift. Otherwise, use the `memcached_servers` option instead.

#### **certfile**

**Type** string

**Default** <None>

Required if identity server requires client certificate

#### **keyfile**

**Type** string

**Default** <None>

Required if identity server requires client certificate

#### **cafile**

**Type** string

**Default** <None>

A PEM encoded Certificate Authority to use when verifying HTTPs connections. Defaults to system CAs.

#### **insecure**

**Type** boolean

**Default** False

Verify HTTPS connections.

#### **region\_name**

**Type** string

**Default** <None>

The region in which the identity server can be found.

#### **memcached\_servers**

**Type** list

**Default** <None>

Optionally specify a list of memcached server(s) to use for caching. If left undefined, tokens will instead be cached in-process.

Table 36: Deprecated Variations

Group	Name
keystone_authtoken	memcache_servers

### **token\_cache\_time**

**Type** integer

**Default** 300

In order to prevent excessive effort spent validating tokens, the middleware caches previously-seen tokens for a configurable duration (in seconds). Set to -1 to disable caching completely.

### **memcache\_security\_strategy**

**Type** string

**Default** None

**Valid Values** None, MAC, ENCRYPT

(Optional) If defined, indicate whether token data should be authenticated or authenticated and encrypted. If MAC, token data is authenticated (with HMAC) in the cache. If ENCRYPT, token data is encrypted and authenticated in the cache. If the value is not one of these options or empty, auth\_token will raise an exception on initialization.

### **memcache\_secret\_key**

**Type** string

**Default** <None>

(Optional, mandatory if memcache\_security\_strategy is defined) This string is used for key derivation.

### **memcache\_pool\_dead\_retry**

**Type** integer

**Default** 300

(Optional) Number of seconds memcached server is considered dead before it is tried again.

### **memcache\_pool\_maxsize**

**Type** integer

**Default** 10

(Optional) Maximum total number of open connections to every memcached server.

### **memcache\_pool\_socket\_timeout**

**Type** integer

**Default** 3

(Optional) Socket timeout in seconds for communicating with a memcached server.

### **memcache\_pool\_unused\_timeout**

**Type** integer

**Default** 60

(Optional) Number of seconds a connection to memcached is held unused in the pool before it is closed.

**memcache\_pool\_conn\_get\_timeout**

**Type** integer

**Default** 10

(Optional) Number of seconds that an operation will wait to get a memcached client connection from the pool.

**memcache\_use\_advanced\_pool**

**Type** boolean

**Default** False

(Optional) Use the advanced (eventlet safe) memcached client pool. The advanced pool will only work under python 2.x.

**include\_service\_catalog**

**Type** boolean

**Default** True

(Optional) Indicate whether to set the X-Service-Catalog header. If False, middleware will not ask for service catalog on token validation and will not set the X-Service-Catalog header.

**enforce\_token\_bind**

**Type** string

**Default** permissive

Used to control the use and type of token binding. Can be set to: disabled to not check token binding. permissive (default) to validate binding information if the bind type is of a form known to the server and ignore it if not. strict like permissive but if the bind type is unknown the token will be rejected. required any form of token binding is needed to be allowed. Finally the name of a binding method that must be present in tokens.

**service\_token\_roles**

**Type** list

**Default** ['service']

A choice of roles that must be present in a service token. Service tokens are allowed to request that an expired token can be used and so this check should tightly control that only actual services should be sending this token. Roles here are applied as an ANY check so any role in this list must be present. For backwards compatibility reasons this currently only affects the allow\_expired check.

**service\_token\_roles\_required**

**Type** boolean

**Default** False

For backwards compatibility reasons we must let valid service tokens pass that dont pass the service\_token\_roles check as valid. Setting this true will become the default in a future release and should be enabled if possible.

### **service\_type**

**Type** string

**Default** <None>

The name or type of the service as it appears in the service catalog. This is used to validate tokens that have restricted access rules.

### **auth\_type**

**Type** unknown type

**Default** <None>

Authentication type to load

Table 37: Deprecated Variations

Group	Name
keystone_authtoken	auth_plugin

### **auth\_section**

**Type** unknown type

**Default** <None>

Config Section from which to load plugin specific options

## **mdns**

### **registration\_attempts**

**Type** integer

**Default** 5

**Minimum Value** 1

Number of attempts to register a service. Currently has to be larger than 1 because of race conditions in the zeroconf library.

### **lookup\_attempts**

**Type** integer

**Default** 3

**Minimum Value** 1

Number of attempts to lookup a service.

### **params**

**Type** unknown type

**Default** { }

Additional parameters to pass for the registered service.

### **interfaces**

**Type** list

**Default** <None>

List of IP addresses of interfaces to use for mDNS. Defaults to all interfaces on the system.

## **metrics**

### **backend**

**Type** string

**Default** noop

**Valid Values** noop, statsd

Backend to use for the metrics system.

### **prepend\_host**

**Type** boolean

**Default** False

Prepend the hostname to all metric names. The format of metric names is [global\_prefix.][host\_name.]prefix.metric\_name.

### **prepend\_host\_reverse**

**Type** boolean

**Default** True

Split the prepended host value by . and reverse it (to better match the reverse hierarchical form of domain names).

### **global\_prefix**

**Type** string

**Default** <None>

Prefix all metric names with this value. By default, there is no global prefix. The format of metric names is [global\_prefix.][host\_name.]prefix.metric\_name.

### **agent\_backend**

**Type** string

**Default** noop

Backend for the agent ramdisk to use for metrics. Default possible backends are noop and statsd.

### **agent\_prepend\_host**

**Type** boolean

**Default** False

Prepend the hostname to all metric names sent by the agent ramdisk. The format of metric names is [global\_prefix.][uuid.][host\_name.]prefix.metric\_name.

### **agent\_prepend\_uuid**

**Type** boolean

**Default** `False`

Prepend the nodes Ironic uuid to all metric names sent by the agent ramdisk. The format of metric names is `[global_prefix.][uuid.][host_name.]prefix.metric_name`.

#### **agent\_prepend\_host\_reverse**

**Type** `boolean`

**Default** `True`

Split the prepended host value by `.` and reverse it for metrics sent by the agent ramdisk (to better match the reverse hierarchical form of domain names).

#### **agent\_global\_prefix**

**Type** `string`

**Default** `<None>`

Prefix all metric names sent by the agent ramdisk with this value. The format of metric names is `[global_prefix.][uuid.][host_name.]prefix.metric_name`.

### **metrics\_statsd**

#### **statsd\_host**

**Type** `string`

**Default** `localhost`

Host for use with the statsd backend.

#### **statsd\_port**

**Type** `port number`

**Default** `8125`

**Minimum Value** `0`

**Maximum Value** `65535`

Port to use with the statsd backend.

#### **agent\_statsd\_host**

**Type** `string`

**Default** `localhost`

Host for the agent ramdisk to use with the statsd backend. This must be accessible from networks the agent is booted on.

#### **agent\_statsd\_port**

**Type** `port number`

**Default** `8125`

**Minimum Value** `0`

**Maximum Value** `65535`

Port for the agent ramdisk to use with the statsd backend.

## neutron

### add\_all\_ports

**Type** boolean

**Default** False

**Mutable** This option can be changed without restarting.

Option to enable transmission of all ports to neutron when creating ports for provisioning, cleaning, or rescue. This is done without IP addresses assigned to the port, and may be useful in some bonded network configurations.

### auth\_url

**Type** unknown type

**Default** <None>

Authentication URL

### auth\_type

**Type** unknown type

**Default** <None>

Authentication type to load

Table 38: Deprecated Variations

Group	Name
neutron	auth_plugin

### cafile

**Type** string

**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

### certfile

**Type** string

**Default** <None>

PEM encoded client certificate cert file

### cleaning\_network

**Type** string

**Default** <None>

**Mutable** This option can be changed without restarting.

Neutron network UUID or name for the ramdisk to be booted into for cleaning nodes. Required for neutron network interface. It is also required if cleaning nodes when using flat network interface or neutron DHCP provider. If a name is provided, it must be unique among all networks or cleaning will fail.



Table 39: Deprecated Variations

Group	Name
neutron	cleaning_network_uuid

### **cleaning\_network\_security\_groups**

**Type** list

**Default** []

**Mutable** This option can be changed without restarting.

List of Neutron Security Group UUIDs to be applied during cleaning of the nodes. Optional for the neutron network interface and not used for the flat or noop network interfaces. If not specified, default security group is used.

### **collect\_timing**

**Type** boolean

**Default** False

Collect per-API call timing information.

### **connect\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for connection errors.

### **connect\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

### **default\_domain\_id**

**Type** unknown type

**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

### **default\_domain\_name**

**Type** unknown type

**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

### **dhcpcv6\_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: Deprecatcd Variations

Group	Name
neutron	tenant-id
neutron	tenant_id

**project\_name**

**Type** unknown type

**Default** <None>

Project name to scope to

Table 41: Deprecatcd 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: Deprecatcd 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 retrieable HTTP status codes.

**status\_code\_retry\_delay**

**Type** `floating point`

**Default** `<None>`

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

**system\_scope**

**Type** `unknown type`

**Default** `<None>`

Scope for system operations

**tenant\_id**

**Type** `unknown type`

**Default** `<None>`

Tenant ID

**tenant\_name**

**Type** `unknown type`

**Default** `<None>`

Tenant Name

**timeout**

**Type** `integer`

**Default** `<None>`

Timeout value for http requests

**trust\_id**

**Type** `unknown type`

**Default** `<None>`

Trust ID

#### **user\_domain\_id**

**Type** unknown type

**Default** <None>

Users domain id

#### **user\_domain\_name**

**Type** unknown type

**Default** <None>

Users domain name

#### **user\_id**

**Type** unknown type

**Default** <None>

User id

#### **username**

**Type** unknown type

**Default** <None>

Username

Table 43: Deprecated Variations

Group	Name
neutron	user-name
neutron	user_name

#### **valid\_interfaces**

**Type** list

**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

#### **version**

**Type** string

**Default** <None>

Minimum Major API version within a given Major API version for endpoint URL discovery.  
Mutually exclusive with min\_version and max\_version

## **nova**

### **auth\_url**

**Type** unknown type

**Default** <None>

Authentication URL

### **auth\_type**

**Type** unknown type

**Default** <None>

Authentication type to load

Table 44: Deprecated Variations

Group	Name
nova	auth_plugin

### **cafile**

**Type** string

**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

### **certfile**

**Type** string

**Default** <None>

PEM encoded client certificate cert file

### **collect\_timing**

**Type** boolean

**Default** False

Collect per-API call timing information.

### **connect\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for connection errors.

### **connect\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

### **default\_domain\_id**



**Type** unknown type

**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **default\_domain\_name**

**Type** unknown type

**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **domain\_id**

**Type** unknown type

**Default** <None>

Domain ID to scope to

#### **domain\_name**

**Type** unknown type

**Default** <None>

Domain name to scope to

#### **endpoint\_override**

**Type** string

**Default** <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

#### **insecure**

**Type** boolean

**Default** False

Verify HTTPS connections.

#### **keyfile**

**Type** string

**Default** <None>

PEM encoded client certificate key file

#### **max\_version**

**Type** string

**Default** <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with *min\_version*. Mutually exclusive with *version*.

### **min\_version**

**Type** string

**Default** <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with max\_version. Mutually exclusive with version. If min\_version is given with no max\_version it is as if max version is latest.

### **password**

**Type** unknown type

**Default** <None>

Users password

### **project\_domain\_id**

**Type** unknown type

**Default** <None>

Domain ID containing project

### **project\_domain\_name**

**Type** unknown type

**Default** <None>

Domain name containing project

### **project\_id**

**Type** unknown type

**Default** <None>

Project ID to scope to

Table 45: Deprecated Variations

Group	Name
nova	tenant-id
nova	tenant_id

### **project\_name**

**Type** unknown type

**Default** <None>

Project name to scope to

Table 46: Deprecated Variations

Group	Name
nova	tenant-name
nova	tenant_name

### **region\_name**

**Type** string

**Default** <None>

The default region\_name for endpoint URL discovery.

#### **send\_power\_notifications**

**Type** boolean

**Default** True

**Mutable** This option can be changed without restarting.

When set to True, it will enable the support for power state change callbacks to nova. This option should be set to False in deployments that do not have the openstack compute service.

#### **service\_name**

**Type** string

**Default** <None>

The default service\_name for endpoint URL discovery.

#### **service\_type**

**Type** string

**Default** compute

The default service\_type for endpoint URL discovery.

#### **split\_loggers**

**Type** boolean

**Default** False

Log requests to multiple loggers.

#### **status\_code\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

#### **status\_code\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

#### **system\_scope**

**Type** unknown type

**Default** <None>

Scope for system operations

#### **tenant\_id**

**Type** unknown type  
**Default** <None>  
 Tenant ID

**tenant\_name**  
**Type** unknown type  
**Default** <None>  
 Tenant Name

**timeout**  
**Type** integer  
**Default** <None>  
 Timeout value for http requests

**trust\_id**  
**Type** unknown type  
**Default** <None>  
 Trust ID

**user\_domain\_id**  
**Type** unknown type  
**Default** <None>  
 Users domain id

**user\_domain\_name**  
**Type** unknown type  
**Default** <None>  
 Users domain name

**user\_id**  
**Type** unknown type  
**Default** <None>  
 User id

**username**  
**Type** unknown type  
**Default** <None>  
 Username

Table 47: Deprecated Variations

Group	Name
nova	user-name
nova	user_name

## **valid\_interfaces**

**Type** list

**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

## **version**

**Type** string

**Default** <None>

Minimum Major API version within a given Major API version for endpoint URL discovery. Mutually exclusive with min\_version and max\_version

## **oslo\_concurrency**

### **disable\_process\_locking**

**Type** boolean

**Default** False

Enables or disables inter-process locks.

Table 48: Deprecated Variations

Group	Name
DEFAULT	disable_process_locking

### **lock\_path**

**Type** string

**Default** <None>

Directory to use for lock files. For security, the specified directory should only be writable by the user running the processes that need locking. Defaults to environment variable OSLO\_LOCK\_PATH. If external locks are used, a lock path must be set.

Table 49: Deprecated Variations

Group	Name
DEFAULT	lock_path

## **oslo\_messaging\_amqp**

### **container\_name**

**Type** string

**Default** <None>

Name for the AMQP container. must be globally unique. Defaults to a generated UUID

Table 50: Deprecated Variations

Group	Name
amqp1	container_name

### **idle\_timeout**

**Type** integer

**Default** 0

Timeout for inactive connections (in seconds)

Table 51: Deprecated Variations

Group	Name
amqp1	idle_timeout

### **trace**

**Type** boolean

**Default** False

Debug: dump AMQP frames to stdout

Table 52: Deprecated Variations

Group	Name
amqp1	trace

### **ssl**

**Type** boolean

**Default** False

Attempt to connect via SSL. If no other ssl-related parameters are given, it will use the systems CA-bundle to verify the servers certificate.

### **ssl\_ca\_file**

**Type** string

**Default** ''

CA certificate PEM file used to verify the servers certificate

Table 53: Deprecated Variations

Group	Name
amqp1	ssl_ca_file

### **ssl\_cert\_file**

**Type** string

**Default** ''

Self-identifying certificate PEM file for client authentication

Table 54: Deprecated Variations

Group	Name
amqp1	ssl_cert_file

### **ssl\_key\_file**

**Type** string

**Default** ''

Private key PEM file used to sign ssl\_cert\_file certificate (optional)

Table 55: Deprecated Variations

Group	Name
amqp1	ssl_key_file

### **ssl\_key\_password**

**Type** string

**Default** <None>

Password for decrypting ssl\_key\_file (if encrypted)

Table 56: Deprecated Variations

Group	Name
amqp1	ssl_key_password

### **ssl\_verify\_vhost**

**Type** boolean

**Default** False

By default SSL checks that the name in the servers certificate matches the hostname in the transport\_url. In some configurations it may be preferable to use the virtual hostname instead, for example if the server uses the Server Name Indication TLS extension (rfc6066) to provide a certificate per virtual host. Set ssl\_verify\_vhost to True if the servers SSL certificate uses the virtual host name instead of the DNS name.

### **sasl\_mechanisms**

**Type** string

**Default** ''

Space separated list of acceptable SASL mechanisms

Table 57: Deprecated Variations

Group	Name
amqp1	sasl_mechanisms

### **sasl\_config\_dir**

**Type** string

**Default** ''

Path to directory that contains the SASL configuration

Table 58: Deprecated Variations

Group	Name
amqp1	sasl_config_dir

**sasl\_config\_name**

**Type** string

**Default** ''

Name of configuration file (without .conf suffix)

Table 59: Deprecated Variations

Group	Name
amqp1	sasl_config_name

**sasl\_default\_realm**

**Type** string

**Default** ''

SASL realm to use if no realm present in username

**connection\_retry\_interval**

**Type** integer

**Default** 1

**Minimum Value** 1

Seconds to pause before attempting to re-connect.

**connection\_retry\_backoff**

**Type** integer

**Default** 2

**Minimum Value** 0

Increase the connection\_retry\_interval by this many seconds after each unsuccessful failover attempt.

**connection\_retry\_interval\_max**

**Type** integer

**Default** 30

**Minimum Value** 1

Maximum limit for connection\_retry\_interval + connection\_retry\_backoff

**link\_retry\_delay**



**Type** integer

**Default** 10

**Minimum Value** 1

Time to pause between re-connecting an AMQP 1.0 link that failed due to a recoverable error.

#### **default\_reply\_retry**

**Type** integer

**Default** 0

**Minimum Value** -1

The maximum number of attempts to re-send a reply message which failed due to a recoverable error.

#### **default\_reply\_timeout**

**Type** integer

**Default** 30

**Minimum Value** 5

The deadline for an rpc reply message delivery.

#### **default\_send\_timeout**

**Type** integer

**Default** 30

**Minimum Value** 5

The deadline for an rpc cast or call message delivery. Only used when caller does not provide a timeout expiry.

#### **default\_notify\_timeout**

**Type** integer

**Default** 30

**Minimum Value** 5

The deadline for a sent notification message delivery. Only used when caller does not provide a timeout expiry.

#### **default\_sender\_link\_timeout**

**Type** integer

**Default** 600

**Minimum Value** 1

The duration to schedule a purge of idle sender links. Detach link after expiry.

#### **addressing\_mode**

**Type** string

**Default** dynamic

Indicates the addressing mode used by the driver. Permitted values: legacy - use legacy non-routable addressing routable - use routable addresses dynamic - use legacy addresses if the message bus does not support routing otherwise use routable addressing

#### **pseudo\_vhost**

**Type** boolean

**Default** True

Enable virtual host support for those message buses that do not natively support virtual hosting (such as qpidd). When set to true the virtual host name will be added to all message bus addresses, effectively creating a private subnet per virtual host. Set to False if the message bus supports virtual hosting using the hostname field in the AMQP 1.0 Open performative as the name of the virtual host.

#### **server\_request\_prefix**

**Type** string

**Default** exclusive

address prefix used when sending to a specific server

Table 60: Deprecated Variations

Group	Name
amqp1	server_request_prefix

#### **broadcast\_prefix**

**Type** string

**Default** broadcast

address prefix used when broadcasting to all servers

Table 61: Deprecated Variations

Group	Name
amqp1	broadcast_prefix

#### **group\_request\_prefix**

**Type** string

**Default** unicast

address prefix when sending to any server in group

Table 62: Deprecated Variations

Group	Name
amqp1	group_request_prefix

#### **rpc\_address\_prefix**

**Type** string

**Default** openstack.org/om/rpc

Address prefix for all generated RPC addresses

**notify\_address\_prefix**

**Type** string

**Default** openstack.org/om/notify

Address prefix for all generated Notification addresses

**multicast\_address**

**Type** string

**Default** multicast

Appended to the address prefix when sending a fanout message. Used by the message bus to identify fanout messages.

**unicast\_address**

**Type** string

**Default** unicast

Appended to the address prefix when sending to a particular RPC/Notification server. Used by the message bus to identify messages sent to a single destination.

**anycast\_address**

**Type** string

**Default** anycast

Appended to the address prefix when sending to a group of consumers. Used by the message bus to identify messages that should be delivered in a round-robin fashion across consumers.

**default\_notification\_exchange**

**Type** string

**Default** <None>

Exchange name used in notification addresses. Exchange name resolution precedence: Target.exchange if set else default\_notification\_exchange if set else control\_exchange if set else notify

**default\_rpc\_exchange**

**Type** string

**Default** <None>

Exchange name used in RPC addresses. Exchange name resolution precedence: Target.exchange if set else default\_rpc\_exchange if set else control\_exchange if set else rpc

**reply\_link\_credit**

**Type** integer

**Default** 200

**Minimum Value** 1

Window size for incoming RPC Reply messages.

### **rpc\_server\_credit**

**Type** integer

**Default** 100

**Minimum Value** 1

Window size for incoming RPC Request messages

### **notify\_server\_credit**

**Type** integer

**Default** 100

**Minimum Value** 1

Window size for incoming Notification messages

### **pre\_settled**

**Type** multi-valued

**Default** rpc-cast

**Default** rpc-reply

Send messages of this type pre-settled. Pre-settled messages will not receive acknowledgement from the peer. Note well: pre-settled messages may be silently discarded if the delivery fails. Permitted values: rpc-call - send RPC Calls pre-settled rpc-reply- send RPC Replies pre-settled rpc-cast - Send RPC Casts pre-settled notify - Send Notifications pre-settled

## **oslo\_messaging\_kafka**

### **kafka\_max\_fetch\_bytes**

**Type** integer

**Default** 1048576

Max fetch bytes of Kafka consumer

### **kafka\_consumer\_timeout**

**Type** floating point

**Default** 1.0

Default timeout(s) for Kafka consumers

### **pool\_size**

**Type** integer

**Default** 10

Pool Size for Kafka Consumers

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

**Reason** Driver no longer uses connection pool.

### **conn\_pool\_min\_size**

**Type** integer

**Default** 2

The pool size limit for connections expiration policy

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

**Reason** Driver no longer uses connection pool.

### **conn\_pool\_ttl**

**Type** integer

**Default** 1200

The time-to-live in sec of idle connections in the pool

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

**Reason** Driver no longer uses connection pool.

### **consumer\_group**

**Type** string

**Default** oslo\_messaging\_consumer

Group id for Kafka consumer. Consumers in one group will coordinate message consumption

### **producer\_batch\_timeout**

**Type** floating point

**Default** 0.0

Upper bound on the delay for KafkaProducer batching in seconds

### **producer\_batch\_size**

**Type** integer

**Default** 16384

Size of batch for the producer async send

### **compression\_codec**

**Type** string

**Default** none

**Valid Values** none, gzip, snappy, lz4, zstd

The compression codec for all data generated by the producer. If not set, compression will not be used. Note that the allowed values of this depend on the kafka version

**enable\_auto\_commit**

**Type** boolean

**Default** False

Enable asynchronous consumer commits

**max\_poll\_records**

**Type** integer

**Default** 500

The maximum number of records returned in a poll call

**security\_protocol**

**Type** string

**Default** PLAINTEXT

**Valid Values** PLAINTEXT, SASL\_PLAINTEXT, SSL, SASL\_SSL

Protocol used to communicate with brokers

**sasl\_mechanism**

**Type** string

**Default** PLAIN

Mechanism when security protocol is SASL

**ssl\_cafile**

**Type** string

**Default** ''

CA certificate PEM file used to verify the server certificate

**ssl\_client\_cert\_file**

**Type** string

**Default** ''

Client certificate PEM file used for authentication.

**ssl\_client\_key\_file**

**Type** string

**Default** ''

Client key PEM file used for authentication.

**ssl\_client\_key\_password**

**Type** string

**Default** ''

Client key password file used for authentication.

## oslo\_messaging\_notifications

### driver

**Type** multi-valued

**Default** ''

The Drivers(s) to handle sending notifications. Possible values are messaging, messagingv2, routing, log, test, noop

Table 63: Deprecated Variations

Group	Name
DEFAULT	notification_driver

### transport\_url

**Type** string

**Default** <None>

A URL representing the messaging driver to use for notifications. If not set, we fall back to the same configuration used for RPC.

Table 64: Deprecated Variations

Group	Name
DEFAULT	notification_transport_url

### topics

**Type** list

**Default** ['notifications']

AMQP topic used for OpenStack notifications.

Table 65: Deprecated Variations

Group	Name
rpc_notifier2	topics
DEFAULT	notification_topics

### retry

**Type** integer

**Default** -1

The maximum number of attempts to re-send a notification message which failed to be delivered due to a recoverable error. 0 - No retry, -1 - indefinite

## **oslo\_messaging\_rabbit**

### **amqp\_durable\_queues**

**Type** boolean

**Default** False

Use durable queues in AMQP.

### **amqp\_auto\_delete**

**Type** boolean

**Default** False

Auto-delete queues in AMQP.

Table 66: Deprecated Variations

Group	Name
DEFAULT	amqp_auto_delete

### **ssl**

**Type** boolean

**Default** False

Connect over SSL.

Table 67: Deprecated Variations

Group	Name
oslo_messaging_rabbit	rabbit_use_ssl

### **ssl\_version**

**Type** string

**Default** ''

SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1\_1, and TLSv1\_2 may be available on some distributions.

Table 68: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_version

### **ssl\_key\_file**

**Type** string

**Default** ''

SSL key file (valid only if SSL enabled).



Table 69: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_keyfile

### ssl\_cert\_file

**Type** string

**Default** ''

SSL cert file (valid only if SSL enabled).

Table 70: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_certfile

### ssl\_ca\_file

**Type** string

**Default** ''

SSL certification authority file (valid only if SSL enabled).

Table 71: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_ssl_ca_certs

### heartbeat\_in\_pthread

**Type** boolean

**Default** True

Run the health check heartbeat thread through a native python thread by default. If this option is equal to False then the health check heartbeat will inherit the execution model from the parent process. For example if the parent process has monkey patched the stdlib by using eventlet/greenlet then the heartbeat will be run through a green thread.

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

### kombu\_reconnect\_delay

**Type** floating point

**Default** 1.0

How long to wait before reconnecting in response to an AMQP consumer cancel notification.

Table 72: Deprecated Variations

Group	Name
DEFAULT	kombu_reconnect_delay

### **kombu\_compression**

**Type** string

**Default** <None>

EXPERIMENTAL: Possible values are: gzip, bz2. If not set compression will not be used. This option may not be available in future versions.

### **kombu\_missing\_consumer\_retry\_timeout**

**Type** integer

**Default** 60

How long to wait a missing client before abandoning to send it its replies. This value should not be longer than `rpc_response_timeout`.

Table 73: Deprecated Variations

Group	Name
oslo_messaging_rabbit	kombu_reconnect_timeout

### **kombu\_failover\_strategy**

**Type** string

**Default** round-robin

**Valid Values** round-robin, shuffle

Determines how the next RabbitMQ node is chosen in case the one we are currently connected to becomes unavailable. Takes effect only if more than one RabbitMQ node is provided in config.

### **rabbit\_login\_method**

**Type** string

**Default** AMQPLAIN

**Valid Values** PLAIN, AMQPLAIN, RABBIT-CR-DEMO

The RabbitMQ login method.

Table 74: Deprecated Variations

Group	Name
DEFAULT	rabbit_login_method

### **rabbit\_retry\_interval**

**Type** integer

**Default** 1

How frequently to retry connecting with RabbitMQ.

### **rabbit\_retry\_backoff**

**Type** integer

**Default** 2

How long to backoff for between retries when connecting to RabbitMQ.

Table 75: Deprecatcd 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: Deprecatcd 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	policy_file

### **policy\_default\_rule**

**Type** string

**Default** `default`

Default rule. Enforced when a requested rule is not found.

Table 78: Deprecated Variations

Group	Name
DEFAULT	policy_default_rule

### **policy\_dirs**

**Type** multi-valued

**Default** `policy.d`

Directories where policy configuration files are stored. They can be relative to any directory in the search path defined by the `config_dir` option, or absolute paths. The file defined by `policy_file` must exist for these directories to be searched. Missing or empty directories are ignored.

Table 79: Deprecated Variations

Group	Name
DEFAULT	policy_dirs

### **remote\_content\_type**

**Type** string

**Default** `application/x-www-form-urlencoded`

**Valid Values** `application/x-www-form-urlencoded`, `application/json`

Content Type to send and receive data for REST based policy check

**remote\_ssl\_verify\_server\_cert**

**Type** boolean

**Default** False

server identity verification for REST based policy check

**remote\_ssl\_ca\_cert\_file**

**Type** string

**Default** <None>

Absolute path to ca cert file for REST based policy check

**remote\_ssl\_client\_cert\_file**

**Type** string

**Default** <None>

Absolute path to client cert for REST based policy check

**remote\_ssl\_client\_key\_file**

**Type** string

**Default** <None>

Absolute path client key file REST based policy check

## profiler

**enabled**

**Type** boolean

**Default** False

Enable the profiling for all services on this node.

Default value is False (fully disable the profiling feature).

Possible values:

- True: Enables the feature
- False: Disables the feature. The profiling cannot be started via this project operations. If the profiling is triggered by another project, this project part will be empty.

Table 80: Deprecated Variations

Group	Name
profiler	profiler_enabled

**trace\_sqlalchemy**

**Type** boolean

**Default** False

Enable SQL requests profiling in services.

Default value is False (SQL requests wont be traced).

Possible values:

- True: Enables SQL requests profiling. Each SQL query will be part of the trace and can the be analyzed by how much time was spent for that.
- False: Disables SQL requests profiling. The spent time is only shown on a higher level of operations. Single SQL queries cannot be analyzed this way.

### **hmac\_keys**

**Type** string

**Default** SECRET\_KEY

Secret key(s) to use for encrypting context data for performance profiling.

This string value should have the following format: <key1>[,<key2>,<keyn>], where each key is some random string. A user who triggers the profiling via the REST API has to set one of these keys in the headers of the REST API call to include profiling results of this node for this particular project.

Both enabled flag and hmac\_keys config options should be set to enable profiling. Also, to generate correct profiling information across all services at least one key needs to be consistent between OpenStack projects. This ensures it can be used from client side to generate the trace, containing information from all possible resources.

### **connection\_string**

**Type** string

**Default** messaging://

Connection string for a notifier backend.

Default value is `messaging://` which sets the notifier to `oslo_messaging`.

Examples of possible values:

- `messaging://` - use `oslo_messaging` driver for sending spans.
- `redis://127.0.0.1:6379` - use `redis` driver for sending spans.
- `mongodb://127.0.0.1:27017` - use `mongodb` driver for sending spans.
- `elasticsearch://127.0.0.1:9200` - use `elasticsearch` driver for sending spans.
- `jaeger://127.0.0.1:6831` - use `jaeger` tracing as driver for sending spans.

### **es\_doc\_type**

**Type** string

**Default** notification

Document type for notification indexing in elasticsearch.

### **es\_scroll\_time**

**Type** string

**Default** 2m

This parameter is a time value parameter (for example: `es_scroll_time=2m`), indicating for how long the nodes that participate in the search will maintain relevant resources in order to continue and support it.

**es\_scroll\_size**

**Type** integer

**Default** 10000

Elasticsearch splits large requests in batches. This parameter defines maximum size of each batch (for example: `es_scroll_size=10000`).

**socket\_timeout**

**Type** floating point

**Default** 0.1

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

Redis Sentinel uses a service name to identify a master redis service. This parameter defines the name (for example: `sentinel_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 00644 in Python. This setting must be set to the octal number representation, meaning starting with 00.

### **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

## 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: Deprecatcd 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: Deprecatcd Variations

Group	Name
DEFAULT	ssl_key_file

## **version**

**Type** string

**Default** <None>

SSL version to use (valid only if SSL enabled). Valid values are TLSv1 and SSLv23. SSLv2, SSLv3, TLSv1\_1, and TLSv1\_2 may be available on some distributions.

## **ciphers**

**Type** string

**Default** <None>

Sets the list of available ciphers. value should be a string in the OpenSSL cipher list format.

## **swift**

### **auth\_url**

**Type** unknown type

**Default** <None>

Authentication URL

### **auth\_type**

**Type** unknown type

**Default** <None>

Authentication type to load

Table 88: Deprecated Variations

Group	Name
swift	auth_plugin

#### **cafile**

**Type** string

**Default** <None>

PEM encoded Certificate Authority to use when verifying HTTPs connections.

#### **certfile**

**Type** string

**Default** <None>

PEM encoded client certificate cert file

#### **collect\_timing**

**Type** boolean

**Default** False

Collect per-API call timing information.

#### **connect\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for connection errors.

#### **connect\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for connection errors. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

#### **default\_domain\_id**

**Type** unknown type

**Default** <None>

Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **default\_domain\_name**

**Type** unknown type

**Default** <None>

Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

#### **domain\_id**

**Type** unknown type

**Default** <None>

Domain ID to scope to

#### **domain\_name**

**Type** unknown type

**Default** <None>

Domain name to scope to

#### **endpoint\_override**

**Type** string

**Default** <None>

Always use this endpoint URL for requests for this client. NOTE: The unversioned endpoint should be specified here; to request a particular API version, use the *version*, *min-version*, and/or *max-version* options.

#### **insecure**

**Type** boolean

**Default** False

Verify HTTPS connections.

#### **keyfile**

**Type** string

**Default** <None>

PEM encoded client certificate key file

#### **max\_version**

**Type** string

**Default** <None>

The maximum major version of a given API, intended to be used as the upper bound of a range with *min\_version*. Mutually exclusive with *version*.

#### **min\_version**

**Type** string

**Default** <None>

The minimum major version of a given API, intended to be used as the lower bound of a range with *max\_version*. Mutually exclusive with *version*. If *min\_version* is given with no *max\_version* it is as if *max version* is latest.

#### **password**

**Type** unknown type

**Default** <None>

Users password

**project\_domain\_id**

**Type** unknown type

**Default** <None>

Domain ID containing project

**project\_domain\_name**

**Type** unknown type

**Default** <None>

Domain name containing project

**project\_id**

**Type** unknown type

**Default** <None>

Project ID to scope to

Table 89: Deprecated Variations

Group	Name
swift	tenant-id
swift	tenant_id

**project\_name**

**Type** unknown type

**Default** <None>

Project name to scope to

Table 90: Deprecated Variations

Group	Name
swift	tenant-name
swift	tenant_name

**region\_name**

**Type** string

**Default** <None>

The default region\_name for endpoint URL discovery.

**service\_name**

**Type** string

**Default** <None>

The default service\_name for endpoint URL discovery.

**service\_type**



**Type** string

**Default** object-store

The default service\_type for endpoint URL discovery.

#### **split\_loggers**

**Type** boolean

**Default** False

Log requests to multiple loggers.

#### **status\_code\_retries**

**Type** integer

**Default** <None>

The maximum number of retries that should be attempted for retrieable HTTP status codes.

#### **status\_code\_retry\_delay**

**Type** floating point

**Default** <None>

Delay (in seconds) between two retries for retrieable status codes. If not set, exponential retry starting with 0.5 seconds up to a maximum of 60 seconds is used.

#### **swift\_max\_retries**

**Type** integer

**Default** 2

Maximum number of times to retry a Swift request, before failing.

#### **system\_scope**

**Type** unknown type

**Default** <None>

Scope for system operations

#### **tenant\_id**

**Type** unknown type

**Default** <None>

Tenant ID

#### **tenant\_name**

**Type** unknown type

**Default** <None>

Tenant Name

#### **timeout**

**Type** integer

**Default** <None>

Timeout value for http requests

**trust\_id**

**Type** unknown type

**Default** <None>

Trust ID

**user\_domain\_id**

**Type** unknown type

**Default** <None>

Users domain id

**user\_domain\_name**

**Type** unknown type

**Default** <None>

Users domain name

**user\_id**

**Type** unknown type

**Default** <None>

User id

**username**

**Type** unknown type

**Default** <None>

Username

Table 91: Deprecated Variations

Group	Name
swift	user-name
swift	user_name

**valid\_interfaces**

**Type** list

**Default** ['internal', 'public']

List of interfaces, in order of preference, for endpoint URL.

**version**

**Type** string

**Default** <None>

Minimum Major API version within a given Major API version for endpoint URL discovery.  
Mutually exclusive with min\_version and max\_version

## xclarity

### manager\_ip

**Type** string

**Default** <None>

IP address of the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver\_info field to use xclarity\_manager\_ip instead

### username

**Type** string

**Default** <None>

Username for the XClarity Controller. Configuration here is deprecated and will be removed in the Stein release. Please update the driver\_info field to use xclarity\_username instead

### password

**Type** string

**Default** <None>

Password for XClarity Controller username. Configuration here is deprecated and will be removed in the Stein release. Please update the driver\_info field to use xclarity\_password instead

### port

**Type** port number

**Default** 443

**Minimum Value** 0

**Maximum Value** 65535

Port to be used for XClarity Controller connection.

## 6.1.2 Policies

**Warning:** JSON formatted policy files were deprecated in the Wallaby development cycle due to the Victoria deprecation by the `olso.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** `rule:is_admin`

**Operations**

- **POST** /nodes

Create Node records

**baremetal:node:get**

**Default** `rule:is_admin` or `rule:is_observer`

**Operations**

- **GET** /nodes/{node\_ident}

Retrieve a single Node record

**baremetal:node:list**

**Default** `rule:baremetal:node:get`

**Operations**

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

Retrieve multiple Node records, filtered by owner

**baremetal:node:list\_all**

**Default** `rule:baremetal:node:get`

**Operations**

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

Retrieve multiple Node records

**baremetal:node:update**

**Default** `rule:is_admin`

**Operations**

- **PATCH** /nodes/{node\_ident}

Update Node records

**baremetal:node:update\_extra**

**Default** `rule:baremetal:node:update`

**Operations**

- **PATCH** /nodes/{node\_ident}

Update Node extra field

**baremetal:node:update\_instance\_info**

**Default** `rule:baremetal:node:update`

**Operations**

- **PATCH** /nodes/{node\_ident}

Update Node instance\_info field

**baremetal:node:update\_owner\_provisioned**

**Default** rule:is\_admin

**Operations**

- **PATCH** /nodes/{node\_ident}

Update Node owner even when Node is provisioned

**baremetal:node:delete**

**Default** rule:is\_admin

**Operations**

- **DELETE** /nodes/{node\_ident}

Delete Node records

**baremetal:node:validate**

**Default** rule:is\_admin

**Operations**

- **GET** /nodes/{node\_ident}/validate

Request active validation of Nodes

**baremetal:node:set\_maintenance**

**Default** rule:is\_admin

**Operations**

- **PUT** /nodes/{node\_ident}/maintenance

Set maintenance flag, taking a Node out of service

**baremetal:node:clear\_maintenance**

**Default** rule:is\_admin

**Operations**

- **DELETE** /nodes/{node\_ident}/maintenance

Clear maintenance flag, placing the Node into service again

**baremetal:node:get\_boot\_device**

**Default** rule:is\_admin or rule:is\_observer

**Operations**

- **GET** /nodes/{node\_ident}/management/boot\_device
- **GET** /nodes/{node\_ident}/management/boot\_device/supported

Retrieve Node boot device metadata

**baremetal:node:set\_boot\_device**

**Default** `rule:is_admin`

**Operations**

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

Change Node boot device

**baremetal:node:get\_indicator\_state**

**Default** `rule:is_admin` or `rule:is_observer`

**Operations**

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

Retrieve Node indicators and their states

**baremetal:node:set\_indicator\_state**

**Default** `rule:is_admin`

**Operations**

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

Change Node indicator state

**baremetal:node:inject\_nmi**

**Default** `rule:is_admin`

**Operations**

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

Inject NMI for a node

**baremetal:node:get\_states**

**Default** `rule:is_admin` or `rule:is_observer`

**Operations**

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

View Node power and provision state

**baremetal:node:set\_power\_state**

**Default** `rule:is_admin`

**Operations**

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

Change Node power status

**baremetal:node:set\_provision\_state**

**Default** `rule:is_admin`

**Operations**

- **PUT** /nodes/{node\_ident}/states/provision

Change Node provision status

**baremetal:node:set\_raid\_state**

**Default** rule:is\_admin

**Operations**

- **PUT** /nodes/{node\_ident}/states/raid

Change Node RAID status

**baremetal:node:get\_console**

**Default** rule:is\_admin

**Operations**

- **GET** /nodes/{node\_ident}/states/console

Get Node console connection information

**baremetal:node:set\_console\_state**

**Default** rule:is\_admin

**Operations**

- **PUT** /nodes/{node\_ident}/states/console

Change Node console status

**baremetal:node:vif:list**

**Default** rule:is\_admin

**Operations**

- **GET** /nodes/{node\_ident}/vifs

List VIFs attached to node

**baremetal:node:vif:attach**

**Default** rule:is\_admin

**Operations**

- **POST** /nodes/{node\_ident}/vifs

Attach a VIF to a node

**baremetal:node:vif:detach**

**Default** rule:is\_admin

**Operations**

- **DELETE** /nodes/{node\_ident}/vifs/{node\_vif\_ident}

Detach a VIF from a node

**baremetal:node:traits:list**

**Default** rule:is\_admin or rule:is\_observer



### Operations

- **GET** /nodes/{node\_ident}/traits

List node traits

### **baremetal:node:traits:set**

**Default** rule:is\_admin

### Operations

- **PUT** /nodes/{node\_ident}/traits
- **PUT** /nodes/{node\_ident}/traits/{trait}

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

### **baremetal:node:traits:delete**

**Default** rule:is\_admin

### Operations

- **DELETE** /nodes/{node\_ident}/traits
- **DELETE** /nodes/{node\_ident}/traits/{trait}

Remove one or all traits from a node

### **baremetal:node:bios:get**

**Default** rule:is\_admin or rule:is\_observer

### Operations

- **GET** /nodes/{node\_ident}/bios
- **GET** /nodes/{node\_ident}/bios/{setting}

Retrieve Node BIOS information

### **baremetal:node:disable\_cleaning**

**Default** rule:baremetal:node:update

### Operations

- **PATCH** /nodes/{node\_ident}

Disable Node disk cleaning

### **baremetal:port:get**

**Default** rule:is\_admin or rule:is\_observer

### Operations

- **GET** /ports/{port\_id}
- **GET** /nodes/{node\_ident}/ports
- **GET** /nodes/{node\_ident}/ports/detail
- **GET** /portgroups/{portgroup\_ident}/ports
- **GET** /portgroups/{portgroup\_ident}/ports/detail

Retrieve Port records

**baremetal:port:list**

**Default** rule:baremetal:port:get

**Operations**

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

Retrieve multiple Port records, filtered by owner

**baremetal:port:list\_all**

**Default** rule:baremetal:port:get

**Operations**

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

Retrieve multiple Port records

**baremetal:port:create**

**Default** rule:is\_admin

**Operations**

- **POST** /ports

Create Port records

**baremetal:port:delete**

**Default** rule:is\_admin

**Operations**

- **DELETE** /ports/{port\_id}

Delete Port records

**baremetal:port:update**

**Default** rule:is\_admin

**Operations**

- **PATCH** /ports/{port\_id}

Update Port records

**baremetal:portgroup:get**

**Default** rule:is\_admin or rule:is\_observer

**Operations**

- **GET** /portgroups
- **GET** /portgroups/detail
- **GET** /portgroups/{portgroup\_ident}

- **GET** /nodes/{node\_ident}/portgroups
- **GET** /nodes/{node\_ident}/portgroups/detail

Retrieve Portgroup records

#### **baremetal:portgroup:create**

**Default** rule:is\_admin

##### **Operations**

- **POST** /portgroups

Create Portgroup records

#### **baremetal:portgroup:delete**

**Default** rule:is\_admin

##### **Operations**

- **DELETE** /portgroups/{portgroup\_ident}

Delete Portgroup records

#### **baremetal:portgroup:update**

**Default** rule:is\_admin

##### **Operations**

- **PATCH** /portgroups/{portgroup\_ident}

Update Portgroup records

#### **baremetal:chassis:get**

**Default** rule:is\_admin or rule:is\_observer

##### **Operations**

- **GET** /chassis
- **GET** /chassis/detail
- **GET** /chassis/{chassis\_id}

Retrieve Chassis records

#### **baremetal:chassis:create**

**Default** rule:is\_admin

##### **Operations**

- **POST** /chassis

Create Chassis records

#### **baremetal:chassis:delete**

**Default** rule:is\_admin

##### **Operations**

- **DELETE** /chassis/{chassis\_id}

Delete Chassis records

**baremetal:chassis:update**

**Default** `rule:is_admin`

**Operations**

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

Update Chassis records

**baremetal:driver:get**

**Default** `rule:is_admin` or `rule:is_observer`

**Operations**

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

View list of available drivers

**baremetal:driver:get\_properties**

**Default** `rule:is_admin` or `rule:is_observer`

**Operations**

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

View driver-specific properties

**baremetal:driver:get\_raid\_logical\_disk\_properties**

**Default** `rule:is_admin` or `rule:is_observer`

**Operations**

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

View driver-specific RAID metadata

**baremetal:node:vendor\_passthru**

**Default** `rule:is_admin`

**Operations**

- **GET** `nodes/{node_id}/vendor_passthru/methods`
- **GET** `nodes/{node_id}/vendor_passthru?  
method={method_name}`
- **PUT** `nodes/{node_id}/vendor_passthru?  
method={method_name}`
- **POST** `nodes/{node_id}/vendor_passthru?  
method={method_name}`
- **PATCH** `nodes/{node_id}/vendor_passthru?  
method={method_name}`

- **DELETE** nodes/{node\_ident}/vendor\_passthru?method={method\_name}

Access vendor-specific Node functions

#### **baremetal:driver:vendor\_passthru**

**Default** rule:is\_admin

##### **Operations**

- **GET** drivers/{driver\_name}/vendor\_passthru/methods
- **GET** drivers/{driver\_name}/vendor\_passthru?method={method\_name}
- **PUT** drivers/{driver\_name}/vendor\_passthru?method={method\_name}
- **POST** drivers/{driver\_name}/vendor\_passthru?method={method\_name}
- **PATCH** drivers/{driver\_name}/vendor\_passthru?method={method\_name}
- **DELETE** drivers/{driver\_name}/vendor\_passthru?method={method\_name}

Access vendor-specific Driver functions

#### **baremetal:node:ipa\_heartbeat**

**Default** rule:public\_api

##### **Operations**

- **POST** /heartbeat/{node\_ident}

Send heartbeats from IPA ramdisk

#### **baremetal:driver:ipa\_lookup**

**Default** rule:public\_api

##### **Operations**

- **GET** /lookup

Access IPA ramdisk functions

#### **baremetal:volume:get**

**Default** rule:is\_admin or rule:is\_observer

##### **Operations**

- **GET** /volume
- **GET** /volume/connectors
- **GET** /volume/connectors/{volume\_connector\_id}
- **GET** /volume/targets
- **GET** /volume/targets/{volume\_target\_id}

- **GET** /nodes/{node\_ident}/volume
- **GET** /nodes/{node\_ident}/volume/connectors
- **GET** /nodes/{node\_ident}/volume/targets

Retrieve Volume connector and target records

**baremetal:volume:create**

**Default** rule:is\_admin

**Operations**

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

Create Volume connector and target records

**baremetal:volume:delete**

**Default** rule:is\_admin

**Operations**

- **DELETE** /volume/connectors/{volume\_connector\_id}
- **DELETE** /volume/targets/{volume\_target\_id}

Delete Volume connector and target records

**baremetal:volume:update**

**Default** rule:is\_admin

**Operations**

- **PATCH** /volume/connectors/{volume\_connector\_id}
- **PATCH** /volume/targets/{volume\_target\_id}

Update Volume connector and target records

**baremetal:conductor:get**

**Default** rule:is\_admin or rule:is\_observer

**Operations**

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

Retrieve Conductor records

**baremetal:allocation:get**

**Default** rule:is\_admin or rule:is\_observer

**Operations**

- **GET** /allocations/{allocation\_id}
- **GET** /nodes/{node\_ident}/allocation

Retrieve Allocation records

#### **baremetal:allocation:list**

**Default** rule:baremetal:allocation:get

##### **Operations**

- **GET** /allocations

Retrieve multiple Allocation records, filtered by owner

#### **baremetal:allocation:list\_all**

**Default** rule:baremetal:allocation:get

##### **Operations**

- **GET** /allocations

Retrieve multiple Allocation records

#### **baremetal:allocation:create**

**Default** rule:is\_admin

##### **Operations**

- **POST** /allocations

Create Allocation records

#### **baremetal:allocation:create\_restricted**

**Default** rule:baremetal:allocation:create

##### **Operations**

- **POST** /allocations

Create Allocation records that are restricted to an owner

#### **baremetal:allocation:delete**

**Default** rule:is\_admin

##### **Operations**

- **DELETE** /allocations/{allocation\_id}
- **DELETE** /nodes/{node\_ident}/allocation

Delete Allocation records

#### **baremetal:allocation:update**

**Default** rule:is\_admin

##### **Operations**

- **PATCH** /allocations/{allocation\_id}

Change name and extra fields of an allocation

#### **baremetal:events:post**

**Default** rule:is\_admin

##### **Operations**

- **POST** /events

Post events

**baremetal:deploy\_template:get**

**Default** rule:is\_admin or rule:is\_observer

**Operations**

- **GET** /deploy\_templates
- **GET** /deploy\_templates/{deploy\_template\_ident}

Retrieve Deploy Template records

**baremetal:deploy\_template:create**

**Default** rule:is\_admin

**Operations**

- **POST** /deploy\_templates

Create Deploy Template records

**baremetal:deploy\_template:delete**

**Default** rule:is\_admin

**Operations**

- **DELETE** /deploy\_templates/{deploy\_template\_ident}

Delete Deploy Template records

**baremetal:deploy\_template:update**

**Default** rule:is\_admin

**Operations**

- **PATCH** /deploy\_templates/{deploy\_template\_ident}

Update Deploy Template records



## BARE METAL API REFERENCES

Ironics 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**

#### **1.69 (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 ironiic 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 ironiic community, as opposed to the [ironiic project](#) storyboard which represents ironiic 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-ironiic channel on Freenode IRC.

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

### Everything Ironiic

Ironiic is a community of projects centered around the primary project repository ironiic, 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 ironiic and which are developed by the same community.

#### See also:

- [Bifrost Documentation](#)
- [Ironiic Inspector Documentation](#)
- [Ironiic Lib Documentation](#)
- [Ironiic Python Agent \(IPA\) Documentation](#)
- [Ironiic Client Documentation](#)
- [Ironiic Inspector Client Documentation](#)

### Useful Links

**Bug/Task tracker** <https://storyboard.openstack.org/#!/project/943>

**Mailing list (prefix Subject line with [ironiic])** <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-discuss>

**Code Hosting** <https://opendev.org/openstack/ironiic>

**Code Review** <https://review.opendev.org/#/q/status:open+project:openstack/ironiic,n,z>

**Whiteboard** <https://etherpad.openstack.org/p/IronicWhiteBoard>

**Weekly Meeting Agenda** [https://wiki.openstack.org/wiki/Meetings/Ironic#Agenda\\_for\\_next\\_meeting](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://storyboard.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 and Python 3.7. It is strongly encouraged to run the unit tests locally prior to submitting a patch.

---

**Note:** Do not run unit tests on the same environment as devstack due to conflicting configuration with system dependencies.

---

---

**Note:** This document is compatible with Python (3.7), Ubuntu (18.04) and Fedora (31). When referring to different versions of Python and OS distributions, this is explicitly stated.

---

**See also:**

<https://docs.openstack.org/infra/manual/developers.html#development-workflow>

### **Prepare Development System**

#### **System Prerequisites**

The following packages cover the prerequisites for a local development environment on most current distributions. Instructions for getting set up with non-default versions of Python and on older distributions are included below as well.

- Ubuntu/Debian:

```
sudo apt-get install build-essential 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
```

- RHEL7/CentOS7:

```
sudo yum 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
```

If using RHEL and yum reports No package python-pip available and No package git-review available, use the EPEL software repository. Instructions can be found at <https://fedoraproject.org/wiki/EPEL/FAQ#howtouse>.

- Fedora:

```
sudo dnf install 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
```

Additionally, if using Fedora 23, `redhat-rpm-config` package should be installed so that development virtualenv can be built successfully.

- openSUSE/SLE 12:

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

Graphviz is only needed for generating the state machine diagram. To install it on openSUSE or SLE 12, see <https://software.opensuse.org/download.html?project=graphics&package=graphviz-plugins>.

To run the tests locally, it is a requirement that your terminal emulator supports unicode with the `en_US.UTF8` locale. If you use `locale-gen` to manage your locales, make sure you have enabled `en_US.UTF8` in `/etc/locale.gen` and rerun `locale-gen`.

## Python Prerequisites

If your distro has at least `tox` 1.8, use similar command to install `python-tox` package. Otherwise install this on all distros:

```
sudo pip install -U tox
```

You may need to explicitly upgrade `virtualenv` if youve installed the one from your OS distribution and it is too old (`tox` will complain). You can upgrade it individually, if you need to:

```
sudo pip install -U virtualenv
```

## Running Unit Tests Locally

If you havent already, IroniC source code should be pulled directly from git:

```
# from your home or source directory
cd ~
git clone https://opendev.org/openstack/ironic
cd ironic
```

## Running Unit and Style Tests

All unit tests should be run using tox. To run IroniC's 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 `teststr`:

```
# 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 -evenv --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
```

RHEL7/CentOS7:

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

Fedora:

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

openSUSE/SLE 12:

```
sudo zypper install mariadb
sudo systemctl start mysql.service
```

If using MySQL, you need to create the initial database:

```
mysql -u root -pMYSQL_ROOT_PWD -e "create schema ironic"
```

---

**Note:** if you choose not to install `mysql-server`, ironiC will default to using a local sqlite database. The database will then be stored in `ironic/ironic.sqlite`.

---

2. Create a configuration file within the ironiC source directory:

```
# generate a sample config
tox -egenconfig

# copy sample config and modify it as necessary
cp etc/ironic/ironic.conf.sample etc/ironic/ironic.conf.local

# disable auth since we are not running keystone here
sed -i "s/#auth_strategy = keystone/auth_strategy = noauth/" etc/
↳ironic/ironic.conf.local

# use the 'fake-hardware' test hardware type
sed -i "s/#enabled_hardware_types = .*/enabled_hardware_types = fake-
↳hardware/" etc/ironic/ironic.conf.local

# use the 'fake' deploy and boot interfaces
sed -i "s/#enabled_deploy_interfaces = .*/enabled_deploy_interfaces =
↳fake/" etc/ironic/ironic.conf.local
sed -i "s/#enabled_boot_interfaces = .*/enabled_boot_interfaces =
↳fake/" etc/ironic/ironic.conf.local
```

(continues on next page)

(continued from previous page)

```
# enable both fake and ipmitool management and power interfaces
sed -i "s/#enabled_management_interfaces = .*/enabled_management_
↪interfaces = fake,ipmitool/" etc/ironic/ironic.conf.local
sed -i "s/#enabled_power_interfaces = .*/enabled_power_interfaces =
↪fake,ipmitool/" etc/ironic/ironic.conf.local

# change the periodic sync_power_state_interval to a week, to avoid
↪getting NodeLocked exceptions
sed -i "s/#sync_power_state_interval = 60/sync_power_state_interval =
↪604800/" etc/ironic/ironic.conf.local

# if you opted to install mysql-server, switch the DB connection from
↪sqlite to mysql
sed -i "s/#connection = .*/connection = mysql\+pymysql:\+\\\/root:MYSQL_
↪ROOT_PWD@localhost\/ironic/" etc/ironic/ironic.conf.local

# use JSON RPC to avoid installing rabbitmq locally
sed -i "s/#rpc_transport = oslo/rpc_transport = json-rpc/" etc/ironic/
↪ironic.conf.local
```

### Step 3: Start the Services

From within the python virtualenv, run the following command to prepare the database before you start the ironic services:

```
# initialize the database for ironic
ironic-dbsync --config-file etc/ironic/ironic.conf.local create_schema
```

Next, open two new terminals for this section, and run each of the examples here in a separate terminal. In this way, the services will *not* be run as daemons; you can observe their output and stop them with Ctrl-C at any time.

1. Start the API service in debug mode and watch its output:

```
cd ~/ironic
. .tox/venv/bin/activate
ironic-api -d --config-file etc/ironic/ironic.conf.local
```

2. Start the Conductor service in debug mode and watch its output:

```
cd ~/ironic
. .tox/venv/bin/activate
ironic-conductor -d --config-file etc/ironic/ironic.conf.local
```

## Step 4: Interact with the running services

You should now be able to interact with ironic via the python client, which is present in the python virtualenv, and observe both services debug outputs in the other two windows. This is a good way to test new features or play with the functionality without necessarily starting DevStack.

To get started, export the following variables to point the client at the local instance of ironic and disable the authentication:

```
export OS_AUTH_TYPE=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

# 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

# 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
```

(continues on next page)

(continued from previous page)

```
# interfaces, most often power and management:
#IRONIC_ENABLED_MANAGEMENT_INTERFACES=ipmitool,fake
#IRONIC_ENABLED_POWER_INTERFACES=ipmitool,fake
# The 'ipmi' hardware type's default deploy interface is 'iscsi'.
# This would change the default to 'direct':
#IRONIC_DEFAULT_DEPLOY_INTERFACE=direct

# Change this to alter the default driver for nodes created by devstack.
# This driver should be in the enabled list above.
IRONIC_DEPLOY_DRIVER=ipmi

# The parameters below represent the minimum possible values to create
# functional nodes.
IRONIC_VM_SPECS_RAM=2048
IRONIC_VM_SPECS_DISK=10

# Size of the ephemeral partition in GB. Use 0 for no ephemeral partition.
IRONIC_VM_EPHEMERAL_DISK=0

# To build your own IPA ramdisk from source, set this to True
IRONIC_BUILD_DEPLOY_RAMDISK=False

VIRT_DRIVER=ironic

# By default, DevStack creates a 10.0.0.0/24 network for instances.
# If this overlaps with the hosts network, you may adjust with the
# following.
NETWORK_GATEWAY=10.1.0.1
FIXED_RANGE=10.1.0.0/24
FIXED_NETWORK_SIZE=256

# Log all output to files
LOGFILE=$HOME/devstack.log
LOGDIR=$HOME/logs
IRONIC_VM_LOG_DIR=$HOME/ironic-bm-logs

END
```

## Ironic with ironic-tempest-plugin

Using the stack user, clone the ironic-tempest-plugin repository in the same directory you cloned DevStack:

```
git clone https://opendev.org/openstack/ironic-tempest-plugin.git
```

An example local.conf that enables the ironic tempest plugin and Ironic can be found below. The TEMPEST\_PLUGINS variable needs to have the absolute path to the ironic-tempest-plugin folder, otherwise the plugin won't be installed. Ironic will have enabled both `direct` and `iscsi` *deploy interfaces* and uses the `ipmi` hardware type by default:

```
cd devstack
cat >local.conf <<END
[[local|localrc]]
```

(continues on next page)

(continued from previous page)

```
# Credentials
ADMIN_PASSWORD=password
DATABASE_PASSWORD=password
RABBIT_PASSWORD=password
SERVICE_PASSWORD=password
SERVICE_TOKEN=password
SWIFT_HASH=password
SWIFT_TEMPURL_KEY=password

# Enable IroniC plugin
enable_plugin ironic https://opendev.org/openstack/ironic

# Disable nova novnc service, ironic does not support it anyway.
disable_service n-novnc

# Enable Swift for the direct deploy interface.
enable_service s-proxy
enable_service s-object
enable_service s-container
enable_service s-account

# Disable Horizon
disable_service horizon

# Disable Cinder
disable_service cinder c-sch c-api c-vol

# Swift temp URL's are required for the direct deploy interface
SWIFT_ENABLE_TEMPURLS=True

# Create 3 virtual machines to pose as IroniC's baremetal nodes.
IRONIC_VM_COUNT=3
IRONIC_BAREMETAL_BASIC_OPS=True
DEFAULT_INSTANCE_TYPE=baremetal

# Enable additional hardware types, if needed.
#IRONIC_ENABLED_HARDWARE_TYPES=ipmi,fake-hardware
# Don't forget that many hardware types require enabling of additional
# interfaces, most often power and management:
#IRONIC_ENABLED_MANAGEMENT_INTERFACES=ipmitool,fake
#IRONIC_ENABLED_POWER_INTERFACES=ipmitool,fake
# The 'ipmi' hardware type's default deploy interface is 'iscsi'.
# This would change the default to 'direct':
#IRONIC_DEFAULT_DEPLOY_INTERFACE=direct

# Change this to alter the default driver for nodes created by devstack.
# This driver should be in the enabled list above.
IRONIC_DEPLOY_DRIVER=ipmi

# The parameters below represent the minimum possible values to create
# functional nodes.
IRONIC_VM_SPECS_RAM=2048
IRONIC_VM_SPECS_DISK=10

# Size of the ephemeral partition in GB. Use 0 for no ephemeral partition.
IRONIC_VM_EPHEMERAL_DISK=0
```

(continues on next page)



(continued from previous page)

```
# 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
```

(continues on next page)

(continued from previous page)

```
# on Ubuntu
sudo -u libvirt-qemu touch $HOME/ironic-bm-logs/test.log
```

---

**Note:** To check out an in-progress patch for testing, you can add a Git ref to the `enable_plugin` line. For instance:

```
enable_plugin ironic https://opendev.org/openstack/ironic refs/changes/46/
↪295946/15
```

For a patch in review, you can find the ref to use by clicking the Download button in Gerrit. You can also specify a different git repo, or a branch or tag:

```
enable_plugin ironic https://github.com/openstack/ironic stable/kilo
```

For more details, see the [devstack plugin interface documentation](#).

---

Run `stack.sh`:

```
./stack.sh
```

Source credentials, create a key, and spawn an instance as the demo user:

```
. ~/devstack/openrc

# query the image id of the default cirros image
image=$(openstack image show $DEFAULT_IMAGE_NAME -f value -c id)

# create keypair
ssh-keygen
openstack keypair create --public-key ~/.ssh/id_rsa.pub default

# spawn instance
openstack server create --flavor baremetal --image $image --key-name_
↪default testing
```

**Note:** Because devstack create multiple networks, we need to pass an additional parameter `--nic net-id` to the nova boot command when using the admin account, for example:

```
net_id=$(openstack network list | egrep "$PRIVATE_NETWORK_NAME" '[^-]' |_
↪awk '{ print $2 }')

openstack server create --flavor baremetal --nic net-id=$net_id --image
↪$image --key-name default testing
```

You should now see a Nova instance building:

```
openstack server list --long
+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+-----+-----+-----+-----+-----+-----+
```

(continues on next page)

(continued from previous page)

ID	Name	Status	Task State	Power State	Networks
→ Image Name	Image ID	Availability Zone	Host	Properties	
a2c7f812	testing	BUILD	spawning	NOSTATE	
→ cirros-0.3	44d4092a	nova			
-e386-4a					.5-
→ x86_64-	-51ac-47				
22-b393-					disk
→	51-9c50-				
fe1802ab					
→	fd6e2050				
d56e					
→	faa1				

Nova will be interfacing with Ironic conductor to spawn the node. On the Ironic side, you should see an Ironic node associated with this Nova instance. It should be powered on and in a wait call-back provisioning state:

```
baremetal node list
```

UUID	Name	Instance UUID
Power State	Provisioning State	Maintenance
9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f	node-0	None
→ power off	None	False
ec0c6384-cc3a-4edf-b7db-abde1998be96	node-1	None
→ power off	None	False
4099e31c-576c-48f8-b460-75e1b14e497f	node-2	a2c7f812-e386-4a22-b393-
→ fe1802abd56e	power on	wait call-back
		False

At this point, Ironic conductor has called to libvirt (via virtualbmc) to power on a virtual machine, which will PXE + TFTP boot from the conductor node and progress through the Ironic provisioning workflow. One libvirt domain should be active now:

```
sudo virsh list --all
```

Id	Name	State
2	node-2	running
-	node-0	shut off
-	node-1	shut off

This provisioning process may take some time depending on the performance of the host system, but Ironic should eventually show the node as having an active provisioning state:

```
baremetal node list
```

UUID	Name	Instance UUID
Power State	Provisioning State	Maintenance

(continues on next page)

(continued from previous page)

9e592cbe-e492-4e4f-bf8f-4c9e0ad1868f	node-0	None	
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=10.1.0.4
cirros-0.3.5-x86_64	44d4092a	nova			0.4, fd7d:1f3c:4bf1:0:f816:3eff:f39d:6d94
disk	51-9c50-fa				
fd6e2050					
faa1					

The server should now be accessible via SSH:

```
ssh cirros@10.1.0.4
$
```

## Running Tempest tests

After *Deploying Ironic with DevStack* with the ironic-tempest-plugin enabled, one might want to run integration tests against the running cloud. The Tempest project is the project that offers an integration test suite for OpenStack.

First, navigate to Tempest directory:

```
cd /opt/stack/tempest
```

To run all tests from the **Ironic plugin**, execute the following command:

```
tox -e all -- ironic
```

To limit the amount of tests that you would like to run, you can use a regex. For instance, to limit the run to a single test file, the following command can be used:

```
tox -e all -- ironic_tempest_plugin.tests.scenario.test_baremetal_basic_ops
```

## Debugging Tempest tests

It is sometimes useful to step through the test code, line by line, especially when the error output is vague. This can be done by running the tests in debug mode and using a debugger such as [pdb](#).

For example, after editing the `test_baremetal_basic_ops` file and setting up the `pdb` traces you can invoke the `run_tempest.sh` script in the Tempest directory with the following parameters:

```
./run_tempest.sh -N -d ironic_tempest_plugin.tests.scenario.test_baremetal_
↪basic_ops
```

- The `-N` parameter tells the script to run the tests in the local environment (without a virtualenv) so it can find the Ironic tempest plugin.
- The `-d` parameter enables the debug mode, allowing it to be used with `pdb`.

For more information about the supported parameters see:

```
./run_tempest.sh --help
```

---

**Note:** Always be careful when running debuggers in time sensitive code, they may cause timeout errors that weren't there before.

---

## OSProfiler Tracing in Ironic

OSProfiler is an OpenStack cross-project profiling library. It is being used among OpenStack projects to look at performance issues and detect bottlenecks. For details on how OSProfiler works and how to use it in ironic, please refer to [OSProfiler Support Documentation](#).

## Building developer documentation

If you would like to build the documentation locally, eg. to test your documentation changes before uploading them for review, run these commands to build the documentation set:

- On your local machine:

```
# activate your development virtualenv
. .tox/venv/bin/activate

# build the docs
tox -edocs

#Now use your browser to open the top-level index.html located at:

ironic/doc/build/html/index.html
```

- On a remote machine:

```
# Go to the directory that contains the docs
cd ~/ironic/doc/source/

# Build the docs
tox -edocs

# Change directory to the newly built HTML files
cd ~/ironic/doc/build/html/

# Create a server using python on port 8000
python -m SimpleHTTPServer 8000

#Now use your browser to open the top-level index.html located at:

http://your_ip:8000
```

### Developer FAQ (frequently asked questions)

Here are some answers to frequently-asked questions from IRC and elsewhere.

- *How do I*
  - *create a migration script template?*
  - *know if a release note is needed for my change?*
  - *create a new release note?*
  - *update a release note?*
  - *get a decision on something?*
  - *add support for GMRs to new executables and extending the GMR?*

### How do I

#### create a migration script template?

Using the `ironic-dbsync revision` command, e.g:

```
$ cd ironic
$ tox -evenv -- ironic-dbsync revision -m \"create foo table\"
```

It will create an empty alembic migration. For more information see the [alembic documentation](#).

## know if a release note is needed for my change?

[Reno documentation](#) contains a description of what can be added to each section of a release note. If, after reading this, you're still unsure about whether to add a release note for your change or not, keep in mind that it is intended to contain information for deployers, so changes to unit tests or documentation are unlikely to require one.

## create a new release note?

By running `reno` command via `tox`, e.g:

```
$ tox -e venv -- reno new version-foo
venv create: /home/foo/ironic/.tox/venv
venv installdeps: -r/home/foo/ironic/test-requirements.txt
venv develop-inst: /home/foo/ironic
venv runtests: PYTHONHASHSEED='0'
venv runtests: commands[0] | reno new version-foo
Created new notes file in releasenotes/notes/version-foo-
→ecb3875dc1cbf6d9.yaml
    venv: commands succeeded
    congratulations :)

$ git status
On branch test
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    releasenotes/notes/version-foo-ecb3875dc1cbf6d9.yaml
```

Then edit the result file. Note that:

- we prefer to use present tense in release notes. For example, a release note should say `Adds support for feature foo`, not `Added support for feature foo`. (We use `adds` instead of `add` because grammatically, it is ironic `adds` support, not ironic `add` support.)
- any variant of English spelling (American, British, Canadian, Australian) is acceptable. The release note itself should be consistent and not have different spelling variants of the same word.

For more information see the [reno documentation](#).

## update a release note?

If this is a release note that pertains to something that was fixed on master or an intermediary release (during a development cycle, that hasn't been branched yet), you can go ahead and update it by submitting a patch.

If it is the release note of an ironic release that has branched, [it can be updated](#) but we will only allow it in extenuating circumstances. (It can be updated by *only* updating the file in that branch. **DO NOT** update the file in master and cherry-pick it. If you do, [see how the mess was cleaned up](#).)

### **get a decision on something?**

You have an issue and would like a decision to be made. First, make sure that the issue hasn't already been addressed, by looking at documentation, stories, specifications, or asking. Information and links can be found on the [IroniC wiki](#) page.

There are several ways to solicit comments and opinions:

- bringing it up at the [weekly IroniC meeting](#)
- bringing it up on [IRC](#)
- bringing it up on the [mailing list](#) (add [IroniC] to the Subject of the email)

If there are enough core folks at the weekly meeting, after discussing an issue, voting could happen and a decision could be made. The problem with IRC or the weekly meeting is that feedback will only come from the people that are actually present.

To inform (and solicit feedback from) more people about an issue, the preferred process is:

1. bring it up on the mailing list
2. after some period of time has elapsed (and depending on the thread activity), someone should propose a solution via gerrit. (E.g. the person that started the thread if no one else steps up.) The proposal should be made in the git repository that is associated with the issue. (For instance, this decision process was proposed as a documentation patch to the ironiC repository.)
3. In the email thread, don't forget to provide a link to the proposed patch!
4. The discussion then moves to the proposed patch. If this is a big decision, we could declare that some percentage of the cores should vote on it before landing it.

(This process was suggested in an email thread about [process for making decisions](#).)

### **add support for GMRs to new executables and extending the GMR?**

For more information, see the [oslo.reports documentation](#) page.

## **Contributor Vision**

### **Background**

During the Rocky Project Teams Gathering (February/March 2018), The contributors in the room at that time took a few minutes to write out each contributor's vision of where they see ironiC in five years time.

After everyone had a chance to spend a few minutes writing, we went around the room and gave every contributor the chance to read their vision and allow other contributors to ask questions to better understand what each individual contributor wrote. While we were doing that, we also took time to capture the common themes.

This entire exercise did result in some laughs and a common set of words, and truly helped to ensure that the entire team proceeded to use the same words to describe various aspects as the sessions progressed during the week. We also agreed that we should write a shared vision, to have something to reference and remind us of where we want to go as a community.



## **Rocky Vision: For 2022-2023**

### **Common Themes**

Below is an entirely unscientific summary of common themes that arose during the discussion among fourteen contributors.

- Contributors picked a time between 2020, and 2023.
- 4 Contributors foresee ironiC being the leading Open Source baremetal deployment technology
- 2 Contributors foresee ironiC reaching feature parity with Nova.
- 2 Contributors foresee users moving all workloads to the cloud
- 1 Contributor foresees Kubernetes and Container integration being the major focus of Bare Metal as a Service further down the road.
- 2 Contributors foresee greater composable hardware being more common.
- 1 Contributor foresees ironiC growing into or supporting CMDBs.
- 2 Contributors foresee that features are more micro-service oriented.
- 2 Contributors foresee that ironiC supported all of the possible baremetal management needs
- 1 Contributor foresees standalone use being more common.
- 2 Contributors foresee the ironiC developer community growing
- 2 Contributors foresee that auto-discovery will be more common.
- 2 Contributors foresee ironiC being used for devices beyond servers, such as lightbulbs, IOT, etc.

### **Vision Statement**

The year is 2022. We're meeting to plan the Z release of IroniC. We stopped to reflect upon the last few years of IroniC's growth, how we had come such a long way to become the defacto open source baremetal deployment technology. How we had grown our use cases, and support for consumers such as containers, and users who wished to 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 community's CMDB integration to leverage hardware discovery. We've heard of operators deploying racks upon racks of new hardware by just connecting the power and network cables, and from there the operators have discovered time to write the world's greatest operator novel with the time saved in commissioning new racks of hardware.

Time has brought us closer and taught us to be more collaborative across the community, and we look forward to our next release together.

## Comparison to the 2018 OpenStack Technical Vision

In late-2018, the OpenStack Technical composed a [technical vision](#) of what OpenStack clouds should look like. While every component differs, and cloudy interactions change dramatically the closer to physical hardware one gets, there are a few areas where Ironiic could use some improvement.

This list is largely for the purposes of help wanted. It is also important to note that Ironiic as a project has a [vision document](#) for itself.

## The Pillars of Cloud - Self Service

- Ironiics mechanisms and tooling are low level infrastructure mechanisms and as such there has never been a huge emphasis or need on making Ironiic be capable of offering direct multi-tenant interaction. Most users interact with the bare metal managed by Ironiic 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

- Ironiic 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, Ironiic 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 [ironiic-ui](#), currently [ironiic-ui](#) receives only minimal housekeeping. As Ironiic has evolved, [ironiic-ui](#) is stuck on version *1.34* and knows nothing of our evolution since. Ironiic ultimately needs a contributor with sufficient time to pick up [ironiic-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 Ironiic 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 Ironiic 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  
node  
start  
off  
in  
when  
cre-  
ated

using API version 1.11 or newer. When a node is in the `enroll` state, the only thing ironiic knows about it is that it exists, and ironiic 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.

**verifying**  
ironiic  
will  
val-  
i-  
date  
that  
it  
can  
man-  
age  
the  
node

ing the information given in `node.driver_info` and with either the driver/hardware type and interfaces it has been assigned. This involves going out and confirming that the credentials work to access whatever node control mechanism they talk to.

using the driver/interfaces and credentials passed in at node create time, the node will be transitioned to the `manageable` state. From `manageable`, nodes can transition to:

ing the `clean` verb.

ing the `inspect` verb.

**manageable**

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

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

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

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

ing the `adopt` verb.

dates need to be made to it such as changes to fields in `driver_info` and updates to networking information on ironiC ports assigned to the node.

- ava  
(thro  
cle  
cle  
if  
au-  
to-  
mati  
clea  
ing  
is  
en-  
able  
by
- act  
(thro  
ado  
by  
set-  
ting  
the  
node  
pro-  
vi-  
sion  
state  
us-  
  
man  
is  
the  
state  
that  
a  
node  
shou  
be  
mov  
into  
whe  
any  
up-  
  
man  
is  
also  
the

to, from these failure states:

derived node properties to reflect the current state of the hardware. Typically, the node will transition to `manageable` if inspection is synchronous, or `inspect wait` if asynchronous. The node will transition to `inspect failed` if error occurred.

only  
sta-  
ble  
state  
that  
can  
be  
tran-  
si-  
tion

- ado  
fai
- cle  
fai
- ins  
fai

#### inspecting

ins  
will  
uti-  
lize  
node  
in-  
tro-  
spec  
tion  
to  
up-  
date  
hard

#### inspect wa

This  
is  
the  
pro-  
vi-  
sion  
state  
used  
whe  
an  
asyn

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

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

inspect verb.

manage verb

chro  
in-  
  
inspect fail  
This  
is  
the  
state  
a  
node  
will  
mov  
into  
whe  
in-  
spec  
tion  
  
•  
ins  
by  
set-  
ting  
the  
node  
pro-  
vi-  
sion  
state  
us-  
ing  
the  
  
•  
man  
by  
set-  
ting  
the  
node  
pro-  
vi-  
sion  
state  
us-  
ing  
the  
  
cleaning



into a known configuration.

tor is executing the clean step (for out-of-band clean steps) or preparing the environment (building PXE configuration files, configuring the DHCP, etc) to boot the ramdisk for running in-band clean steps.

being scrubbed and reprogrammed. The difference is that in the `clean wait` state the conductor is waiting for the ramdisk to boot or the clean step which is running in-band to finish.

Nod  
in  
the  
cle  
state  
are  
be-  
ing  
scru  
and  
re-  
pro-  
gran

Whe  
a  
node  
is  
in  
the  
cle  
state  
it  
mea  
that  
the  
con-  
duc-

#### clean wait

Just  
like  
the  
cle  
state  
the  
node  
in  
the  
cle  
wai  
state  
are

The  
clea  
ing  
pro-

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

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

ing the `active` verb.

cess  
of  
a  
node  
in  
the  
cle  
wai  
state  
can

#### **available**

Af-  
ter  
node  
have  
been  
suc-  
cess  
fully  
pre-  
con-  
fig-  
ured  
and

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

- man  
by  
set-  
ting  
the  
node

manage verb

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

#### deploying

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

- Set-  
ting  
ap-  
pro-  
pri-  
ate  
BIO  
con-  
fig-  
u-  
ra-  
tions
- Par-  
ti-  
tion-  
ing  
drive  
and  
lay-  
ing  
dow  
file  
sys-  
tems

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

deployed. The difference is that in `wait call-back` the conductor is waiting for the ramdisk to boot or execute parts of the deployment which need to run in-band on the node (for example, installing the bootloader, or writing the image to the disk).

rupted by setting the nodes provision state using the `deleted` verb.

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

#### **wait call-b**

Just  
like  
the  
dep  
state  
the  
node  
in  
wai  
cal  
are  
be-  
ing

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

#### **deploy fai**

This  
is

ment fails, for example a timeout waiting for the ramdisk to PXE boot. From here the node can be transitioned to:

ing either the `active` or `rebuild` verbs.

state using the `deleted` verb.

the  
state  
a  
node  
will  
mov  
into  
whe  
a  
de-  
play

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

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

**active (sta**  
Nod  
in  
act  
have  
a  
worl

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

state using the `deleted` verb.

ing the `rebuild` verb.

load  
run-  
ning  
on  
then  
iron  
may

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

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

- res  
(thro  
res  
by  
set-  
ting  
the  
node  
pro-  
vi-

ing the `rescue` verb.

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

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

state using the `deleted` verb.

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  
mov  
into  
whe  
dele  
ing  
an  
  
•  
ava  
(thro  
del  
and  
cle  
by  
set-  
ting  
the  
node  
pro-  
vi-  
sion

baremetal node with an existing workload on it. Ordinarily when a baremetal node is enrolled and managed by ironic, it must transition through `cleaning` and `deploying` to reach `active` state. However, those baremetal nodes that have an existing workload on them, do not need to be deployed or cleaned again, so this transition allows these nodes to move directly from `manageable` to `active`.

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

**adopting**

This state allows ironic to take over management of a

**rescuing**

Nodes in rescue are being prepared to perform rescue

- Setting appropriate BIO configurations
- Creating



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`).

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

#### rescue wait

Just like the rescue state the node in rescue wait are being

The rescue operation of a node in rescue wait can

#### rescue fail

This is the state a node

operation fails, for example a timeout waiting for the ramdisk to PXE boot. From here the node can be transitioned to:

ing the `rescue` verb.

ing the `unrescue` verb.

will  
mov  
into  
whe  
a  
res  
cue

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

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

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

ing the `deleted` verb.

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

ing the `unrescue` verb.

ing the `deleted` verb.

**rescue (sta**

Node  
in  
res  
have  
a  
res-  
cue  
rame  
run-  
ning  
on  
them  
Iron

- 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-

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

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

ing the `rescue` verb.

**unrescuin**  
 Node  
 in  
 unr  
 are  
 be-  
 ing  
 pre-  
 pare  
 to  
 tran-  
 si-  
 tion  
 to

**unrescue**  
 This  
 is  
 the  
 state  
 a  
 node  
 will  
 mov  
 into  
 whe  
 an  
 un-  
 res-

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

- act  
 (thro  
 unr

ing the `unrescue` verb.

ing the `deleted` verb.

## Developing New Notifications

by external services. Notifications are sent to these services over a message bus by `oslo.messaging` `Notifier` class. For more information about configuring notifications and available notifications, see *Notifications*.

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

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

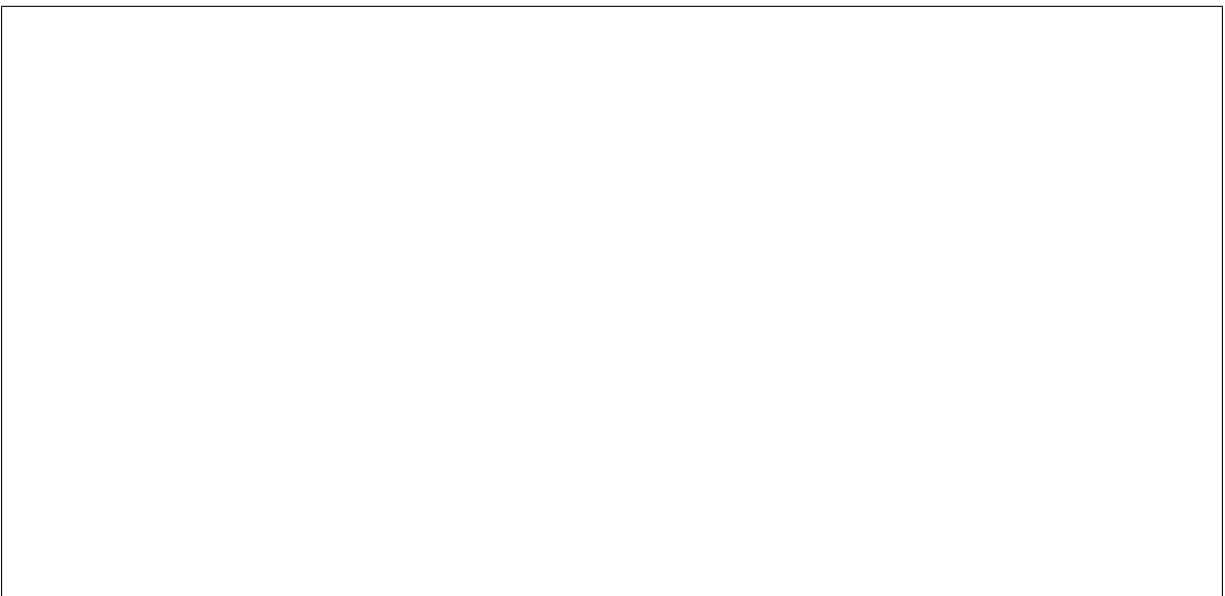
Iron  
no-  
ti-  
fi-  
ca-  
tions  
are  
ever  
in-  
tend  
for  
con-  
sum  
tion

Iron  
also  
has  
a

ing the notification itself, the payload, and the other fields not auto-generated by oslo (level, event\_type and publisher\_id). Below describes how to use these base classes to add a new notification to ironiC.

### **Adding a new notification to ironiC**

sioned notification class should be created by subclassing the NotificationBase class to define the notification itself and the NotificationPayloadBase class to define which fields the new notification will contain inside its payload. You may also define a schema to allow the payload to be automatically populated by the fields of an ironiC object. Heres an example:



(continues on next page)



(continued from previous page)

```

↪StringField(),
↪StringField()

```

(continues on next page)



---

(continued from previous page)

↪ ! )

(continues on next page)

(continued from previous page)

```

→ populate_schema with

→ 'example_obj', 'a_useful_field')

```

(continues on next page)

(continued from previous page)

↪StringField(),

(continues on next page)

↪StringField(nullable=True)

(continued from previous page)



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

optional attribute that subclasses may use to easily populate notifications with data from other objects.

U  
→ U  
→ U  
→ }  
→

Note  
that  
both  
the  
pay-  
load  
and  
no-  
ti-  
fi-  
ca-  
tion  
class  
are

SCH  
de-  
fines  
how  
to  
pop-  
u-  
late  
the  
pay-  
load  
field  
Its  
an

It  
is  
a  
dic-  
tio-  
nary  
where  
every  
key  
valu  
pair  
has



→

→ I

→



U



→



The

<p>

is

the

nan

who

the

14

data:

w11.

be

stor

in

the

pay

The

SCI

nee

to

10  
100

be

ap-

notification can be emitted.

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.

plied  
with  
the  
pop  
call  
be-  
fore  
the

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

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

Then  
to  
cre-  
ate  
a  
pay-  
load  
you  
wou  
do

[illegible]

```
→ not_useful_field='blah')
```

## 9.1. Developers Guide

(continued from previous page)



quired fields (event\_type, publisher\_id, and level, all sender fields needed by oslo that are defined in the ironic notification base classes) and emit it:



(continues on next page)



(continued from previous page)

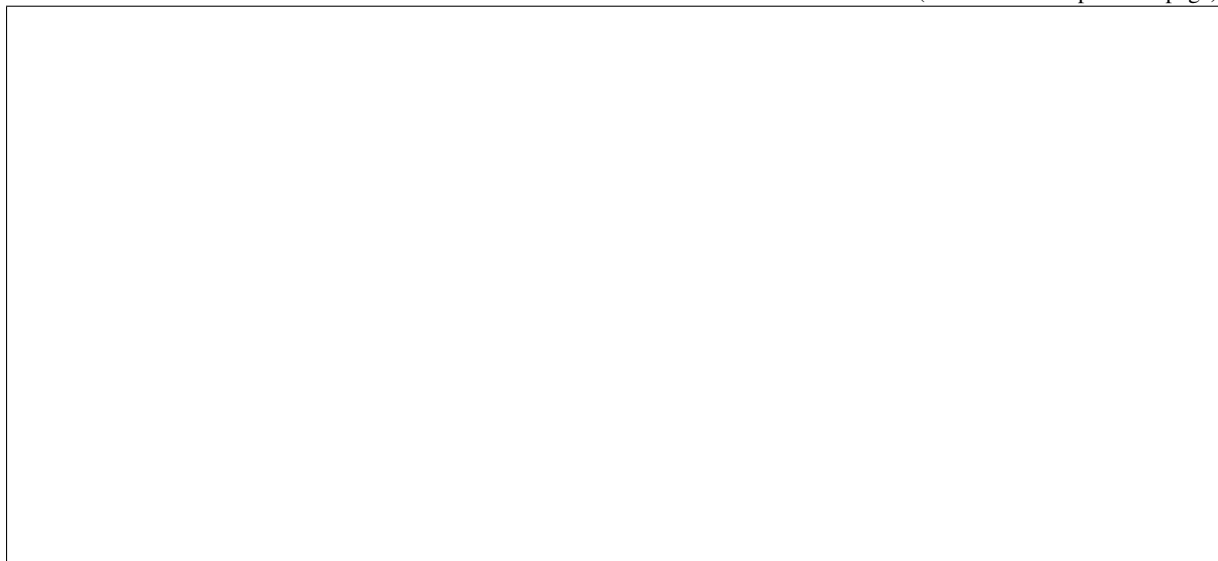
```

→NotificationStatus.START),

```

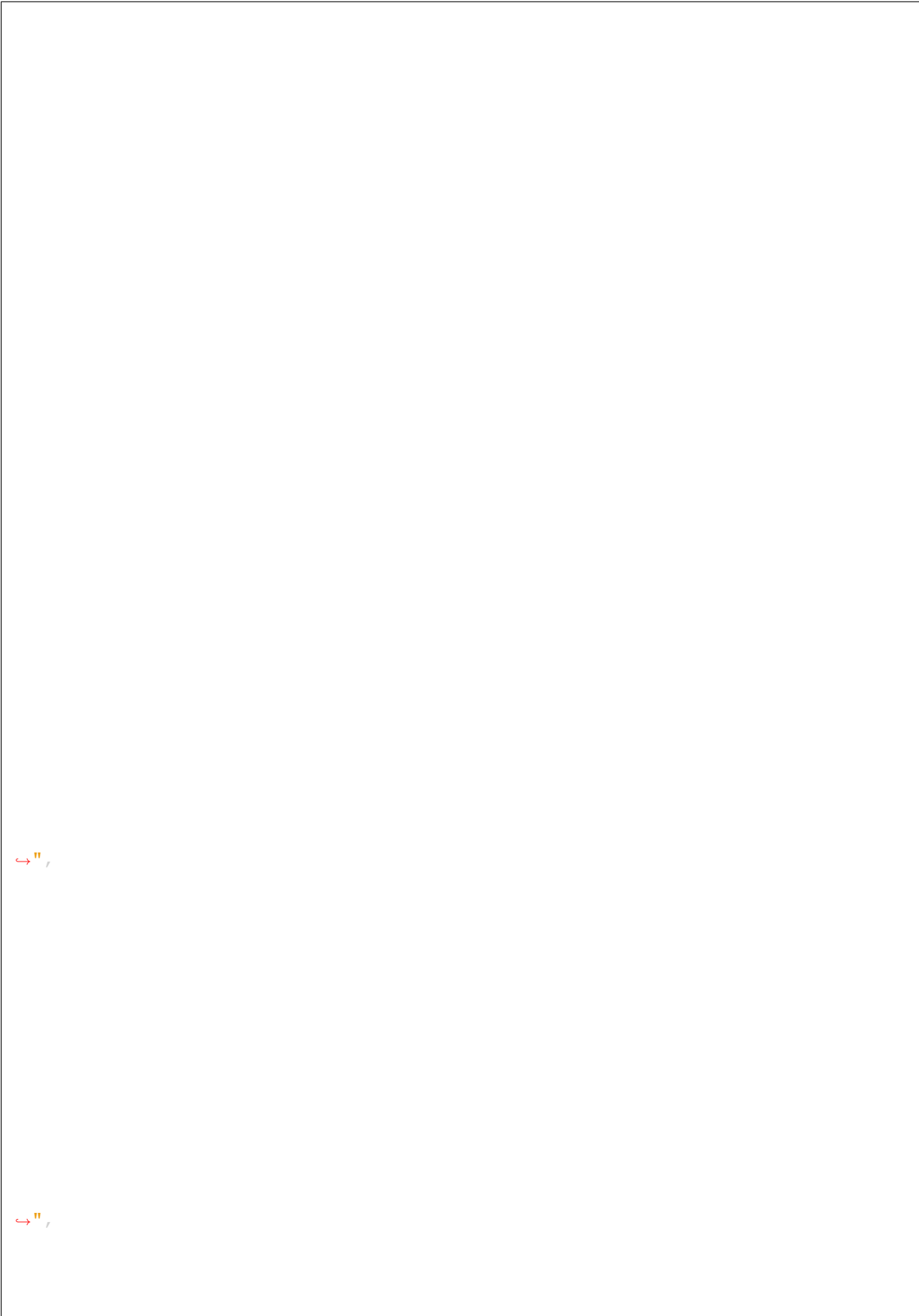
(continues on next page)

(continued from previous page)



being acted on, `action` will be a string describing what action is being performed on that object, and `status` will be one of `start`, `end`, `error`, or `success`. `start` and `end` are used to indicate when actions that are not immediate begin and succeed. `success` is used to indicate when actions that are immediate succeed. `error` is used to indicate when any type of action fails, regardless of whether its immediate or not. As a result of specifying these parameters, `event_type` will be formatted as `baremetal.<object>.<action>.<status>` on the message bus.

tion over the message bus:



(continues on next page)

(continued from previous page)

```

→ "a_useful_field": "important",

→ "an_extra_field": "hello"

```

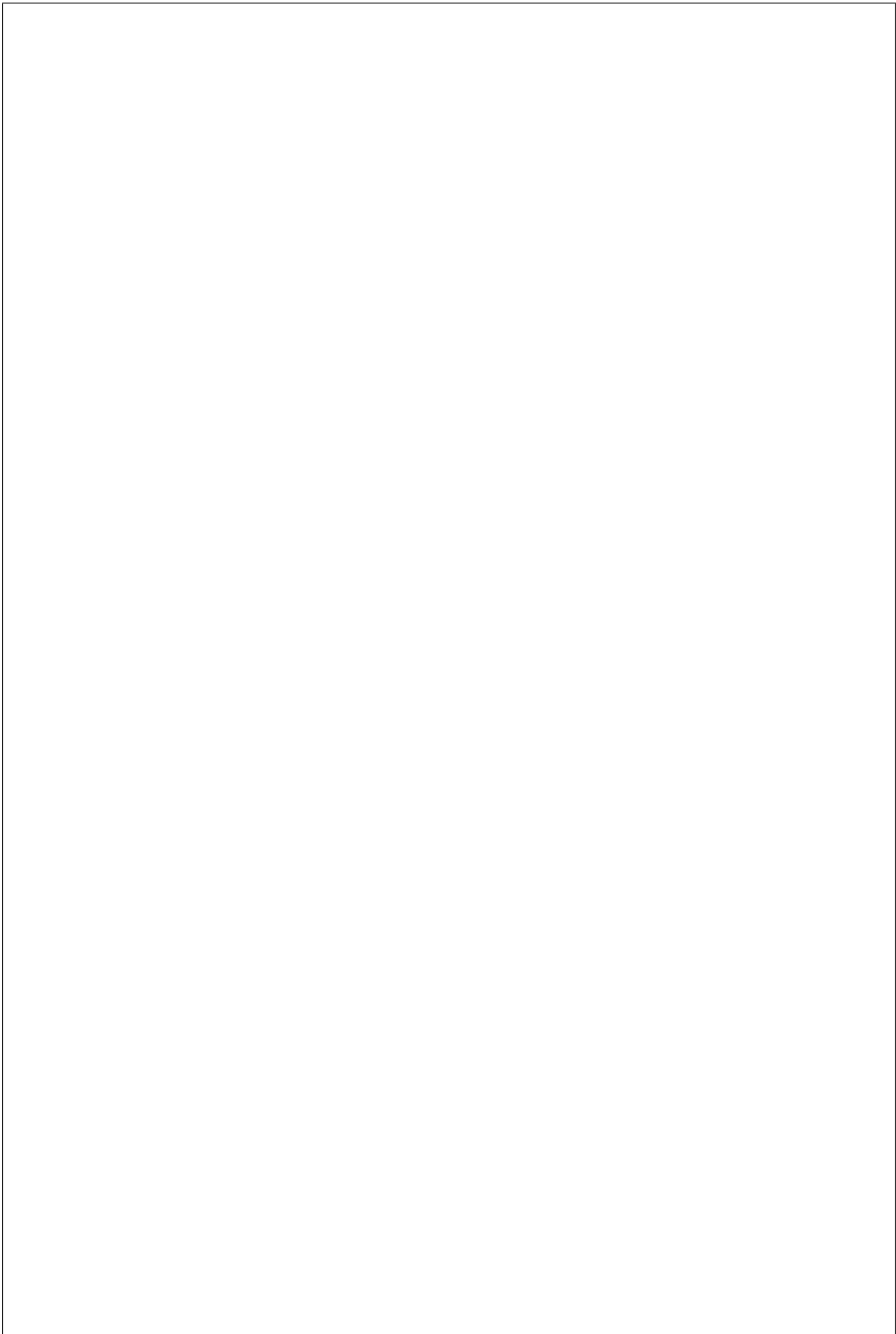
(continues on next page)

(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)

(continued from previous page)



to one trace. This is used to simplify the process of retrieving all trace points (related to one trace) from the collector.

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

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

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

the  
dic-  
tio-  
nary  
that  
con-  
tains  
user  
in-  
for-  
ma-

The  
pro-  
filer  
uses  
ceilometer  
as  
a  
cen-  
tral-  
ized  
col-  
lec-  
tor.

A  
no-  
ti-  
fier  
is  
setu  
to  
send  
no-  
ti-  
fi-  
ca-  
tions  
to

OS-  
Pro-  
filer  
has  
en-  
try  
point  
that



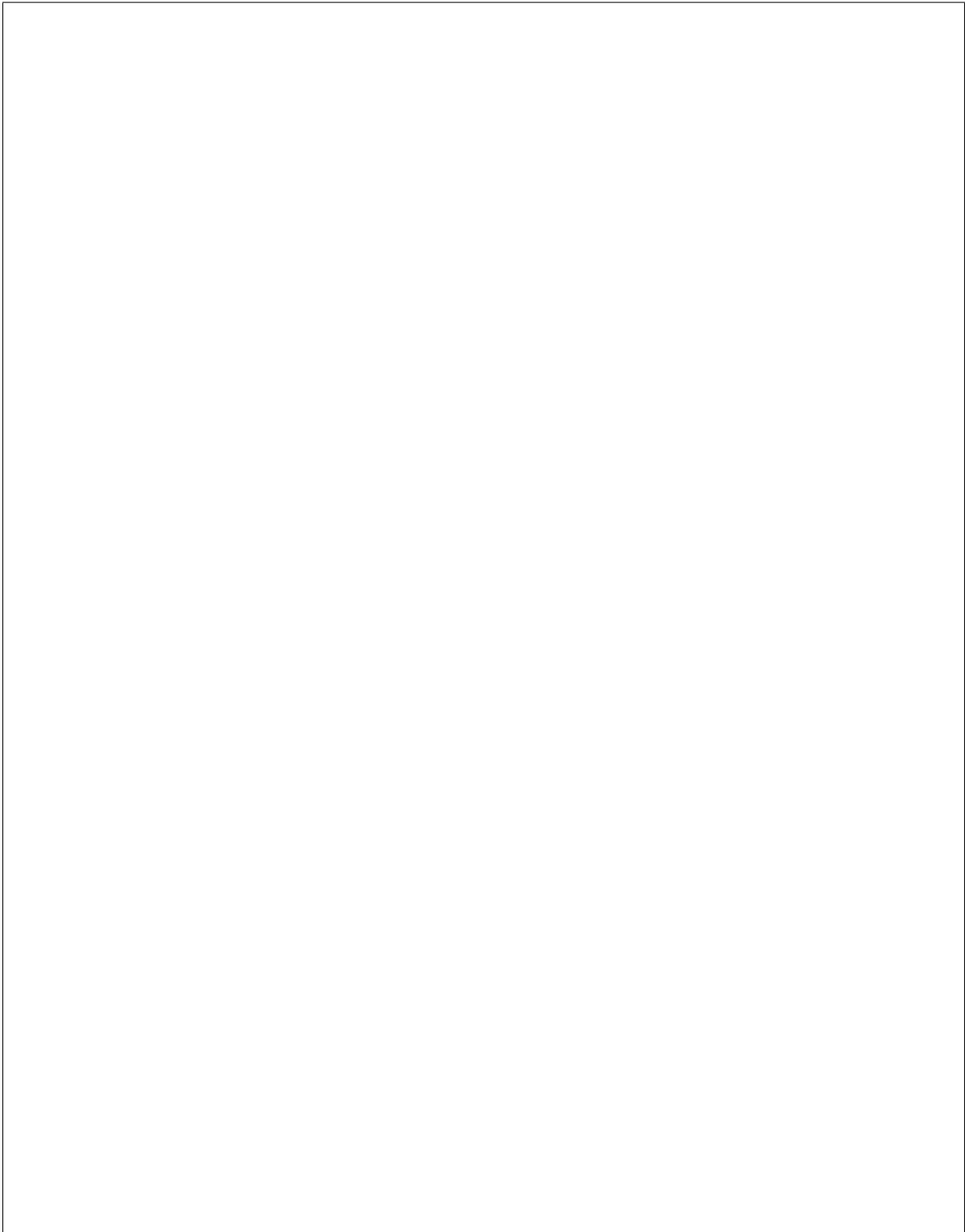
trieve information about traces and present it in HTML/JSON using CLI.

brary.

## How to Use OSProfiler with IroniC in Devstack

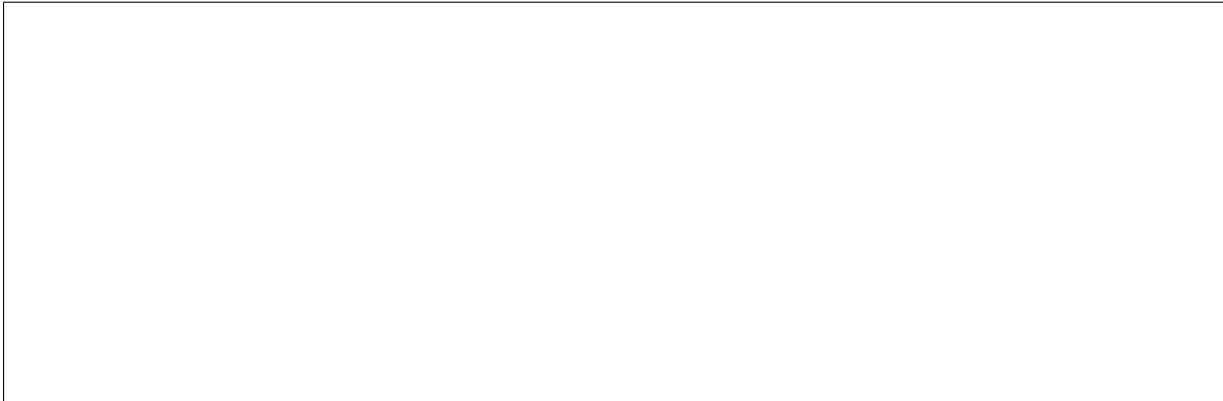
Devstack with OSProfiler and ceilometer. In addition to the setup described at [Deploying IroniC with DevStack](#), the user needs to do the following:

ceilometer:

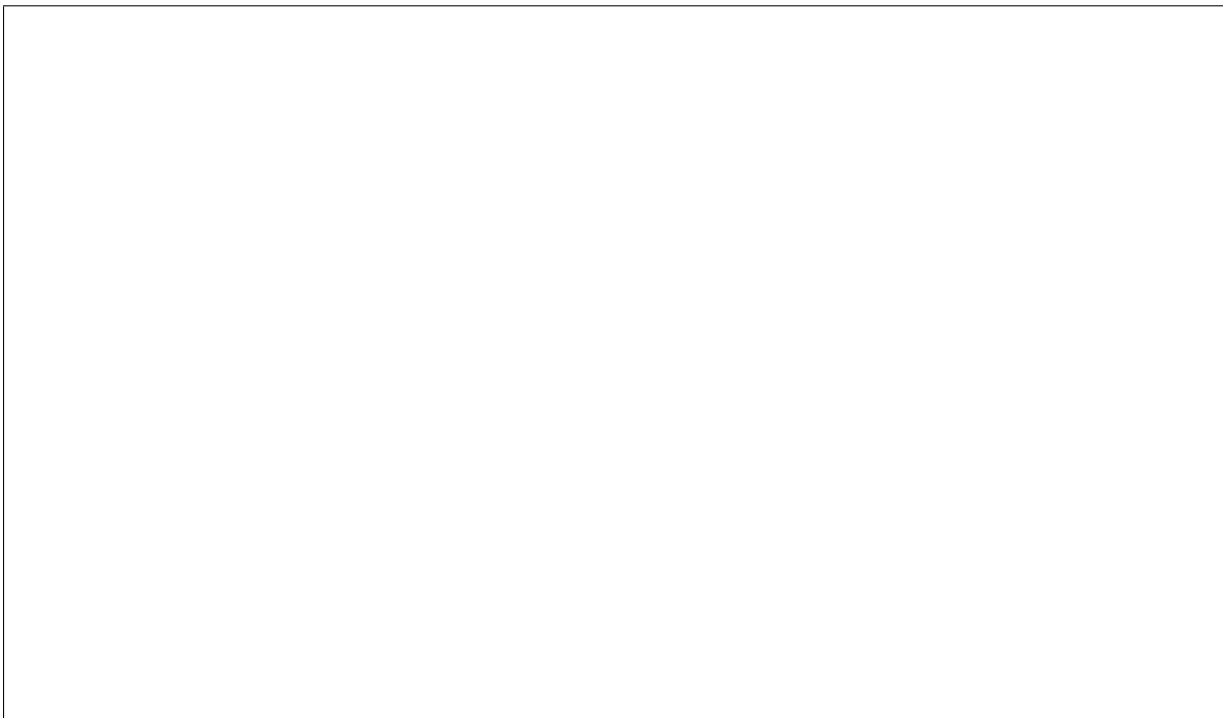


(continues on next page)

(continued from previous page)



set the following profiler options and restart ironiC services:



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

be printed after node list:



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

For  
ex-  
am-  
ple,  
the  
fol-  
low-  
ing  
will  
caus  
a  
<tra  
id>  
to

→ \$  
→ O.  
→ -  
→ -  
→ O  
→ p  
→ S  
→ K  
→ b  
→ n  
→ l

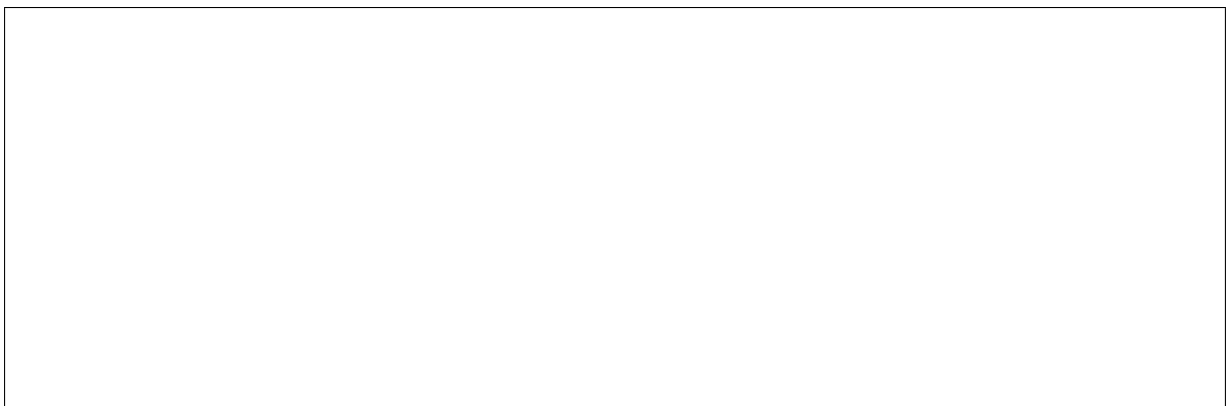
Out-  
put  
of  
the  
abov  
com  
man  
will

in-  
clud  
the  
fol-  
low-  
ing:



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

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



→ \$  
→ o  
→ t  
→ s  
→ -  
→ -  
→ h  
→<  
→ i  
→

The  
trace

tion:

```
→ out trace.html
```

Levels	Duration	Type Project	Service	Host	Details
0	724 ms	total n/a	n/a	n/a	<a href="#">Details</a>
1	9 ms	wsgi keystone	main	ubuntu	<a href="#">Details</a>
1	311 ms	wsgi keystone	main	ubuntu	<a href="#">Details</a>
1	367 ms	wsgi ironic	ironic_api	ubuntu	<a href="#">Details</a>
2	81 ms	wsgi keystone	admin	ubuntu	<a href="#">Details</a>
1778	24 ms	db api ironic	ironic_api	ubuntu	<a href="#">Details</a>
	12 ms	db api ironic	ironic_api	ubuntu	<a href="#">Details</a>
	188 ms	rpc ironic	ironic_conductor	ubuntu	<a href="#">Details</a>
	35 ms	db api ironic	ironic_conductor	ubuntu	<a href="#">Details</a>
	41 ms	db api ironic	ironic_conductor	ubuntu	<a href="#">Details</a>

## Chapter 9. Contributor Guide

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

↪ \$  
 ↪ O  
 ↪ t  
 ↪ s  
 ↪ -  
 ↪ -  
 ↪ h  
 ↪  
 ↪ <  
 ↪ i  
 ↪     
 ↪ -  
 ↪ -

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

Sample  
Trace  
Each  
trace  
has  
em-

Each  
trace  
has  
em-

low:

References

Details
Details
Details
Details
Details
Details
Details
Details
Details
Details
Details
Details
Details
Details

Rolling Upgrades

grade from the Ocata to the Pike release. This describes the design of rolling upgrades, followed by notes for developing new features or modifying an IronicObject.

### **Design**

#### **Rolling upgrades between releases**

`<major>.<minor>.<patch>`. We refer to a named `release` of ironic as the release associated with a development cycle like Pike.



deprecation period must be at least three months and a cycle boundary. This means that there will never be anything that is both deprecated *and* removed between two named releases.

pol-  
icy,  
whic  
says  
that  
the

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-  
ing  
back  
port  
bug  
fixes  
Be-

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

the above bullet point, there may be a bug or a feature introduced on a master branch, that we want to remove before publishing a named release. Deprecation policy allows to do this in a 3 month time frame. If the feature was included and removed in intermediate releases, there should be a release note added, with instructions on how to do a rolling upgrade to master from an affected release or release span. This would typically instruct the operator to upgrade to a particular intermediate release, before upgrading to master.

### **Rolling upgrade process**

leases in this order:

cess  
 will  
 caus  
 the  
 iron  
 ser-  
 vice  
 to  
 be  
 run-  
 ning  
 the  
 Fro  
 and  
 ToV  
 re-

0. Up-grad  
 iron  
 code  
 and  
 run  
 data  
 sche  
 mi-  
 gra-  
 tions  
 via  
 the  
 iro  
 upg  
 com  
 man

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  
ver-  
sion  
so  
that  
the  
ser-  
vice  
can  
now  
use  
the  
lat-

est 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

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.

The  
pol-  
icy  
for  
chan  
to  
the  
DB  
mod  
is  
as  
fol-  
lows

- Add  
new  
item  
to  
the  
DB  
mod  
is  
sup-  
ported

- The  
drop  
ping  
of  
colu  
or  
ta-  
bles  
and  
cor-  
re-  
spor  
ing  
ob-  
jects  
field  
is  
sub-  
ject

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.

may impose table locks and cause downtime. If the change cannot be avoided and the impact is significant (e.g. the table can be frequently accessed and/or store a large dataset), these cases must be mentioned in the release notes.

## API, RPC and object version pinning

a rolling upgrade, the services need to be able to handle different API, RPC and object versions.

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 services to be running old and new releases at the same time during

This versioning is handled via the configuration option. [DE] pin It is

The default value of

versions of API, RPC and IroniCObjects. Its possible values are releases, named (e.g. `ocata`) or sem-versioned (e.g. `7.0`).

IroniCObject versions associated with each release. This mapping is maintained manually.



uration 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

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

and passes it to the `RPCClient` as an initialization parameter. This variable is then used to determine the maximum requested message version that the `RPCClient` can send.

section below has more details about this.

## **Handling IroniCObject versions**

the  
de-  
sired  
(lat-  
est  
or  
pinn  
RPC  
API  
ver-  
sion

Each  
RPC  
call  
can  
cus-  
tomize  
the  
re-  
ques  
ac-  
cord  
ing  
to  
this  
ver  
The  
*IroniC*  
*RPC*  
*ver-*  
*sion*

In-  
ter-  
nally  
ironiC  
ser-  
vice  
deal  
with  
IroniC  
i-  
cOb  
jects  
in

these boundaries, when the IroniCObject enters or leaves the service, do we deal with object versioning:

version

their  
lat-  
est  
ver-  
sion  
Only  
at

- get-  
ting  
ob-  
jects  
from  
the  
data  
con-  
vert  
to  
lat-  
est  
ver-  
sion
- sav-  
ing  
ob-  
jects  
to  
the  
data  
if  
pinn  
save  
in  
pinn  
ver-  
sion  
else  
save  
in  
lat-  
est
- se-  
ri-  
al-  
iz-  
ing

latest version

est version

ob-  
 jects  
 (to  
 send  
 over  
 RPC  
 if  
 pinn  
 send  
 pinn  
 ver-  
 sion  
 else  
 send  
 •  
 de-  
 se-  
 ri-  
 al-  
 iz-  
 ing  
 ob-  
 jects  
 (re-  
 ceiv  
 ing  
 ob-  
 jects  
 from  
 RPC  
 con-  
 vert  
 to  
 lat-  
 The  
 iron  
 api  
 ser-  
 vice  
 also  
 has  
 to  
 han-  
 dle  
 API  
 re-  
 ques  
 base

ture is supported by the API version and object versions. For example, when the ironic-api service is pinned, it can only allow actions that are available to the objects pinned version, and cannot allow actions that are only available for the latest version of that object.

`version`. The value is the version of the object that is saved in the database.

on  
 whe  
 or  
 how  
 a  
 fea

To  
 sup-  
 port  
 this:

- All the data tables (SQL models) of the IroniC objects have a column named

- The method `IroniC.get_return` the target version. If pinned the pinn

turned. Otherwise, the latest version is returned.

sion may be a newer or older version than the existing version of the object. The bulk of the work is done in the helper method `IroniCObject._convert_to_version()`. Subclasses that have new versions redefine this to perform the actual conversions.

ver-  
sion  
is  
re-

- The  
meth  
Iron  
con  
con-  
verts  
the  
ob-  
ject  
into  
the  
tar-  
get  
ver-  
sion  
The  
tar-  
get  
ver-

In  
the  
fol-  
low-  
ing,

- The  
old  
re-  
lease  
is  
Fro  
it  
uses  
ver-  
sion  
1.14  
of  
a  
Nod  
ob-  
ject.

- The new re-release is ToV. It uses ver-sion 1.15 of a Nod ob-ject this has a

deprecated `extra` field and a new `meta` field that replaces `extra`.

- `db_c` and `db_c` are the data rep-re-sen-ta-tions of those node field

## Getting objects from the database (API/conductor < DB)

Both iron api and iron conductor service read val-

to `IroniObjects` via the method `IroniObject._from_db_object()`. This method always returns the `IroniObject` in its latest version, even if it was in an older version in the database. This is done regardless of the service being pinned or not.

retain any changes (in its `_changed_fields` field) resulting from that conversion. This is needed in case the object gets saved later, in the latest version.

ues  
from  
the  
data  
The  
val  
ues  
are  
con-  
verte

Note  
that  
if  
an  
ob-  
ject  
is  
con-  
verte  
to  
a  
later  
ver-  
sion  
that  
Iron  
i-  
cOb  
ject  
will

For  
ex-  
am-  
ple,  
if  
the  
node  
in  
the  
data  
is  
in  
ver-  
sion  
1.14



it doesnt exist)

and  
has  
db\_c  
set:

- a  
From  
ser-  
vice  
will  
get  
a  
Node  
with  
node  
=  
db\_c  
(and  
no  
know  
edge  
of  
node  
since
- a  
ToV  
ser-  
vice  
(pin  
or  
un-  
pinn  
will  
get  
a  
Node  
with
- node  
=  
db\_c
- node  
=  
Non
- node

## **Saving objects to the database (API/conductor > DB)**

lows:

jects are always in their latest version, no conversions are needed.

=  
[met  
ex-  
tra]

The  
ver-  
sion  
used  
for  
sav-  
ing  
Iron  
i-  
cOb  
jects  
to  
the  
data  
is  
de-  
ter-  
mine  
as  
fol-

- For  
an  
un-  
pinn  
ser-  
vice  
the  
ob-  
ject  
is  
save  
in  
its  
lat-  
est  
ver-  
sion  
Sinc  
ob-

always in their latest version, the object needs to be converted to the pinned version before being saved.

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.

• For a pinned service the object is saved in its pinned version. Since objects are

The meth Iro do\_ han- dles this logic re- turn- ing a dic- tio- nary of char field and their

The *Roll up- grac pro- cess*

saved in its latest version, all services are running the newer release (although some may still be pinned) and can handle the latest object versions.

in step 3.1. It is possible for an `IroniCObject` to be saved in a newer version and subsequently get saved in an older version. For example, a `ToVer` unpinned conductor might save a node in version 1.5. A subsequent request may cause a `ToVer` pinned conductor to replace and save the same node in version 1.4!

## Sending objects via RPC (API/conductor -> RPC)

that request are serialized into entities or primitives via `IroniCObjectSerializer.serialize_entity()`. The version used for objects being serialized is as follows:

sion. Since objects are always in their latest version, no conversions are needed.

When  
a  
ser-  
vice  
mak  
an  
RPC  
re-  
ques  
any  
Iron  
i-  
cOb  
jects  
that  
are  
sent  
as  
part  
of

- For  
an  
un-  
pinn  
ser-  
vice  
the  
ob-  
ject  
is  
se-  
ri-  
al-  
ized  
to  
its  
lat-  
est  
ver-

- For  
a  
pinn  
ser-

objects are always in their latest version, the object is converted to the pinned version before being serialized. The converted object includes changes that resulted from the conversion; this is needed so that the service at the other end of the RPC request has the necessary information if that object will be saved to the database.

### **Receiving objects via RPC (API/conductor <- RPC)**

request need to be deserialized (via `oslo.versionedobjects.VersionedObjectSerializer.deserialize_entity()`). For entities that represent `IroniCObjects`, we want the deserialization process (via `IroniCObjectSerializer._process_object()`) to result in `IroniCObjects` that are in their latest version, regardless of the version they were sent in and regardless of whether the receiving service is pinned or not. Again, any objects that are converted will retain the changes that resulted from the conversion, useful if that object is later saved to the database.

version 1.4, where `node.extra` was changed (so `node._changed_fields = [extra]`). This node will be serialized in version 1.4. The receiving `ToVer` pinned `ironic-conductor` deserializes it and converts it to version 1.5. The resulting node will have `node.meta` set (to the changed value from `node.extra` in v1.4), `node.extra = None`, and `node._changed_fields = [meta, extra]`.

### **When developing a new feature or modifying an `IroniCObject`**

that things work during a rolling upgrade.

points to keep in mind when developing code.

### **ironic-api**

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:

ing  
de-  
scrib  
ar-  
eas  
whe  
the  
code  
may  
need  
to  
be  
char  
as  
well  
as  
som

Dur-  
ing  
a  
rolli  
up-  
grad  
the  
new,  
pinn  
iron  
api  
is  
talk-  
ing  
to  
a  
new  
con-  
duc-  
tor

- New  
fea-  
tures  
shou  
not  
be



in the old and new releases. Pinning the API version is in place to handle this.

being handled that cannot or should not be handled, it should be coded so that the response has HTTP status code 406 (Not Acceptable). This is the same response to requests that have an incorrect (old) version specified.

## IroniC RPC versions

the following needs to be considered:

conductor/rpcapi.py, used by ironic-api) and the server (ironic/conductor/manager.py, used by ironic-conductor). It should also be updated in ironic/common/release\_mappings.py.

RPC  
meth  
is  
char  
or  
new  
meth  
ods  
are  
adde

- The  
RPC  
ver-  
sion  
mus  
be  
in-  
cre-  
men  
and  
be  
the  
sam  
valu  
for  
both  
the  
clien  
(ir

- Un-  
til  
there  
is  
a  
ma-  
jor  
ver-  
sion  
bum  
new  
ar-  
gu-  
men  
of

only be added as optional. Existing arguments cannot be removed or changed in incompatible ways with the method in older RPC versions.

structor of `oslo_messaging.RPCClient`). This pinning is in place during a rolling upgrade when the `[DEFAULT]/pin_release_version` configuration option is set.

version. In this case, the corresponding REST API function should return a server error or implement alternative behaviours.

an  
RPC  
meth  
can

- iron  
api  
(clie  
side  
sets  
a  
ver-  
sion  
cap  
(by  
pass  
ing  
the  
ver-  
sion  
cap  
to  
the  
con-
- New  
RPC  
meth  
ods  
are  
not  
avai  
able  
whe  
the  
ser-  
vice  
is  
pinn  
to  
the  
olde  
re-  
lease

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.

working during the rolling upgrade process. The behaviour of ironic-conductor will depend on the input parameters passed from the client-side.

- Methods which character arguments should run the CLI can be used to see if the version of the
- iron conductor (server-side) should tolerate older versions of requests in order to keep
- Old methods can be

ous named release.

## Object versions

sidered:

re-  
mov  
only  
af-  
ter  
they  
are  
no  
long  
used  
by  
a  
pre-  
vi-

When  
sub-  
class  
of  
ironiC  
obj  
bas  
IroniC  
are  
mod  
i-  
fied,  
the  
fol-  
low-  
ing  
need  
to  
be  
con-

- Any  
char  
of  
field  
or  
char  
in  
sig-  
na-  
ture

the object version. The object versions are also maintained in `ironic/common/release_mappings.py`.

be excluded from the version check by adding their class names to the `NEW_MODELS` list in `ironic/cmd/dbsync.py`.

of  
re-  
mot  
meth  
ods  
need  
a  
bum  
of

- New  
ob-  
jects  
must  
be  
adde  
to  
iro  
com  
rel  
py.  
Also  
for  
the  
first  
re-  
lease  
they  
shou

- The  
ar-  
gu-  
men  
of  
re-  
mot  
meth  
ods  
ods  
(met  
ods  
whic  
are  
re-  
mot  
to

tor via RPC) can only be added as optional. They cannot be removed or changed in an incompatible way (to the previous release).

one.

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  
new  
ver-  
sion  
ob-  
jects  
com-  
mu-  
ni-  
cate  
with  
old  
ver-  
sion  
ob-  
jects  
and  
when  
read

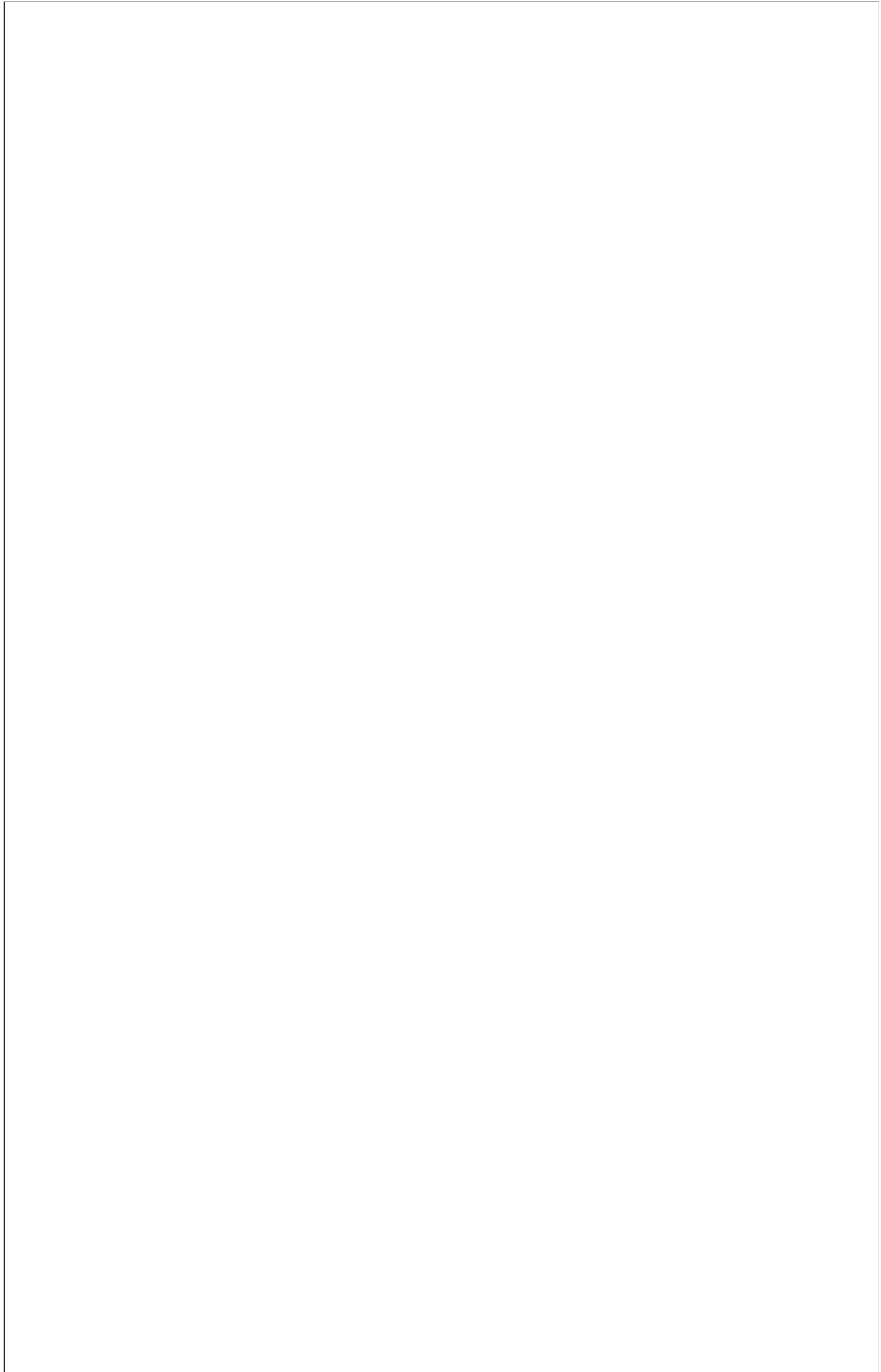
ing or writing to the database, `ironic.objects.base.IronicObject._convert_to_version()` will be called to convert objects to the target version. Objects should implement their own `._convert_to_version()` to remove or alter fields which were added or changed after the target version:

```
↪ remove_unavailable_fields=True):
```

(continues on next page)

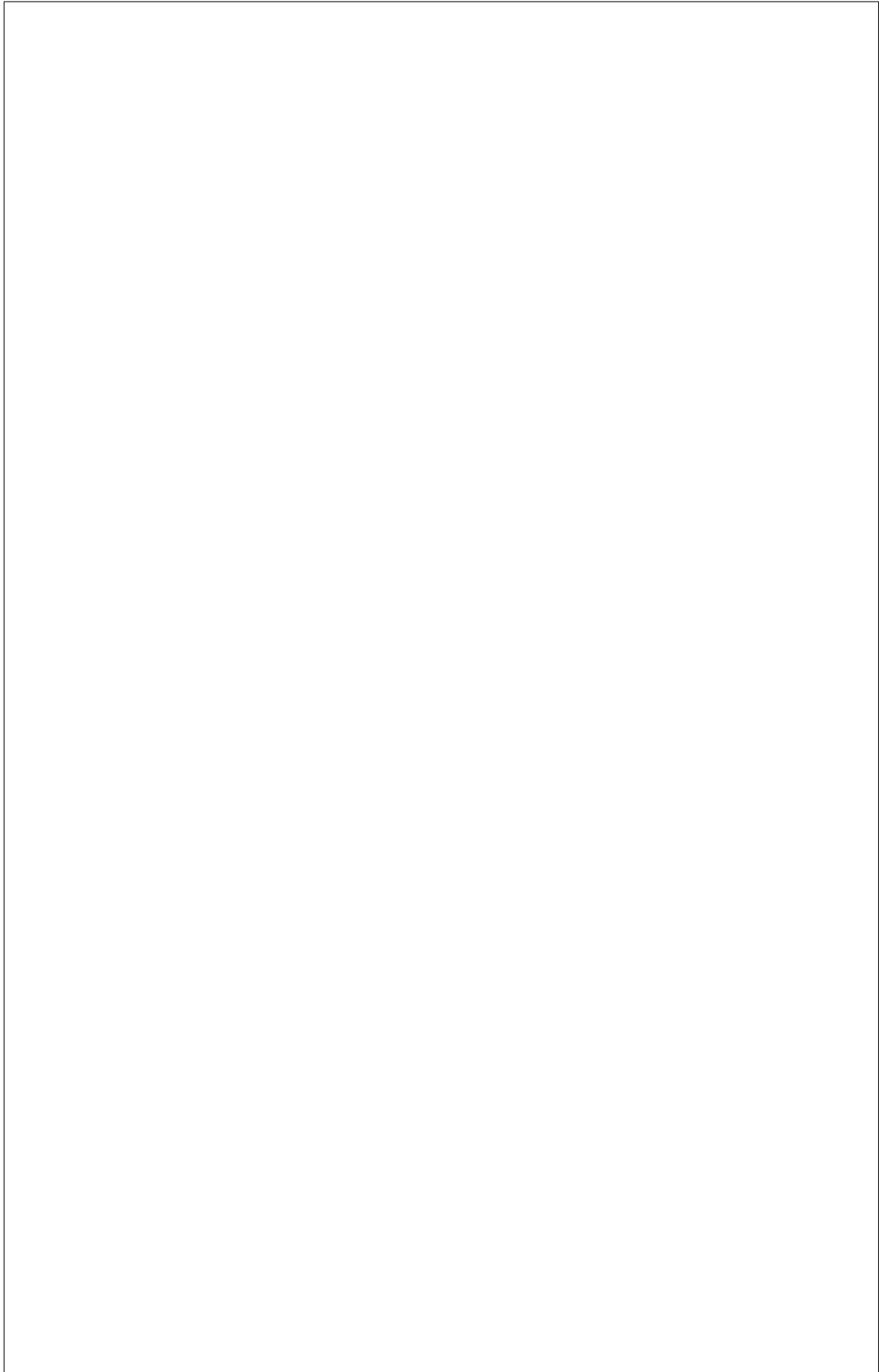


(continued from previous page)



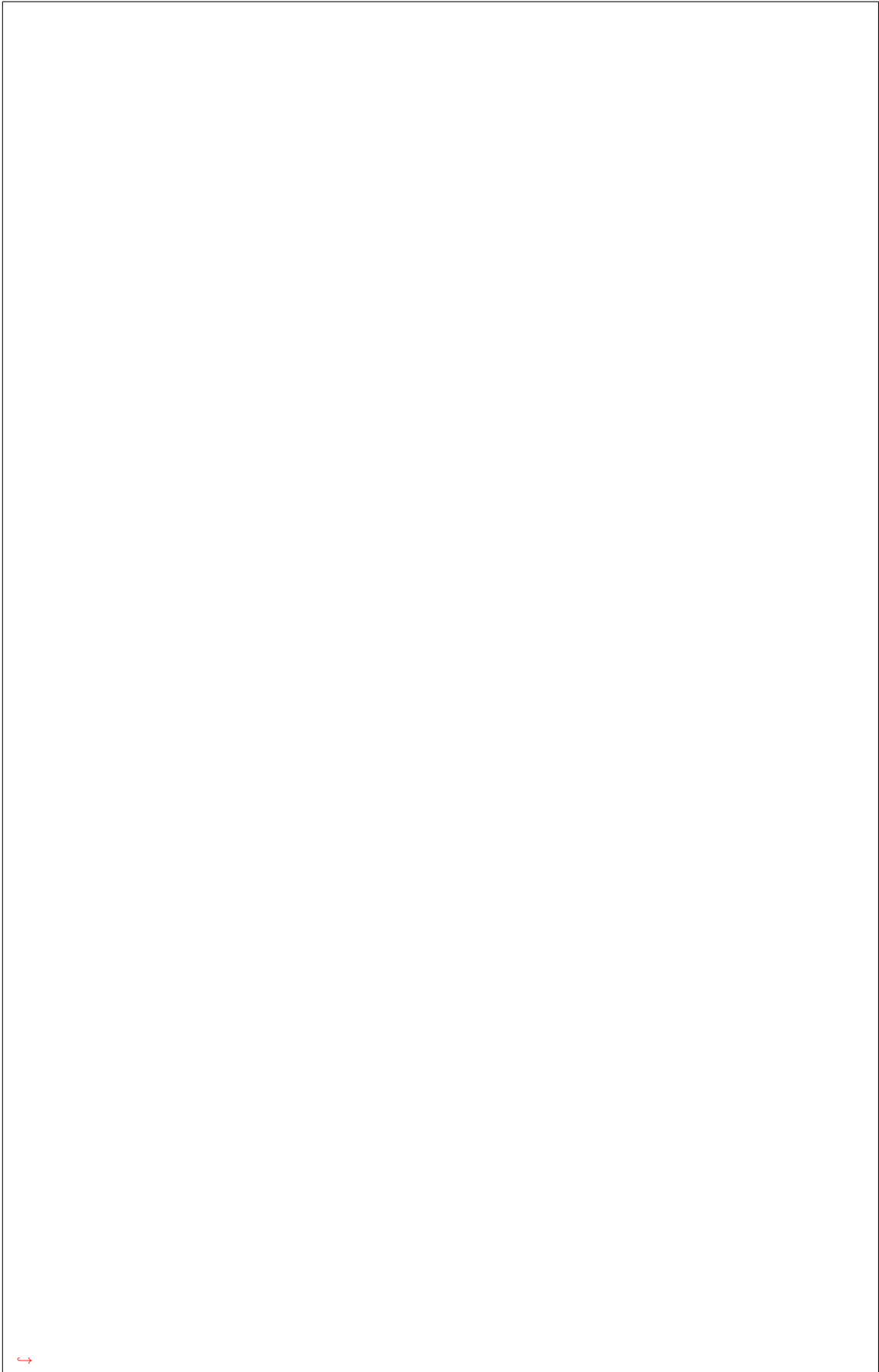
(continues on next page)

(continued from previous page)



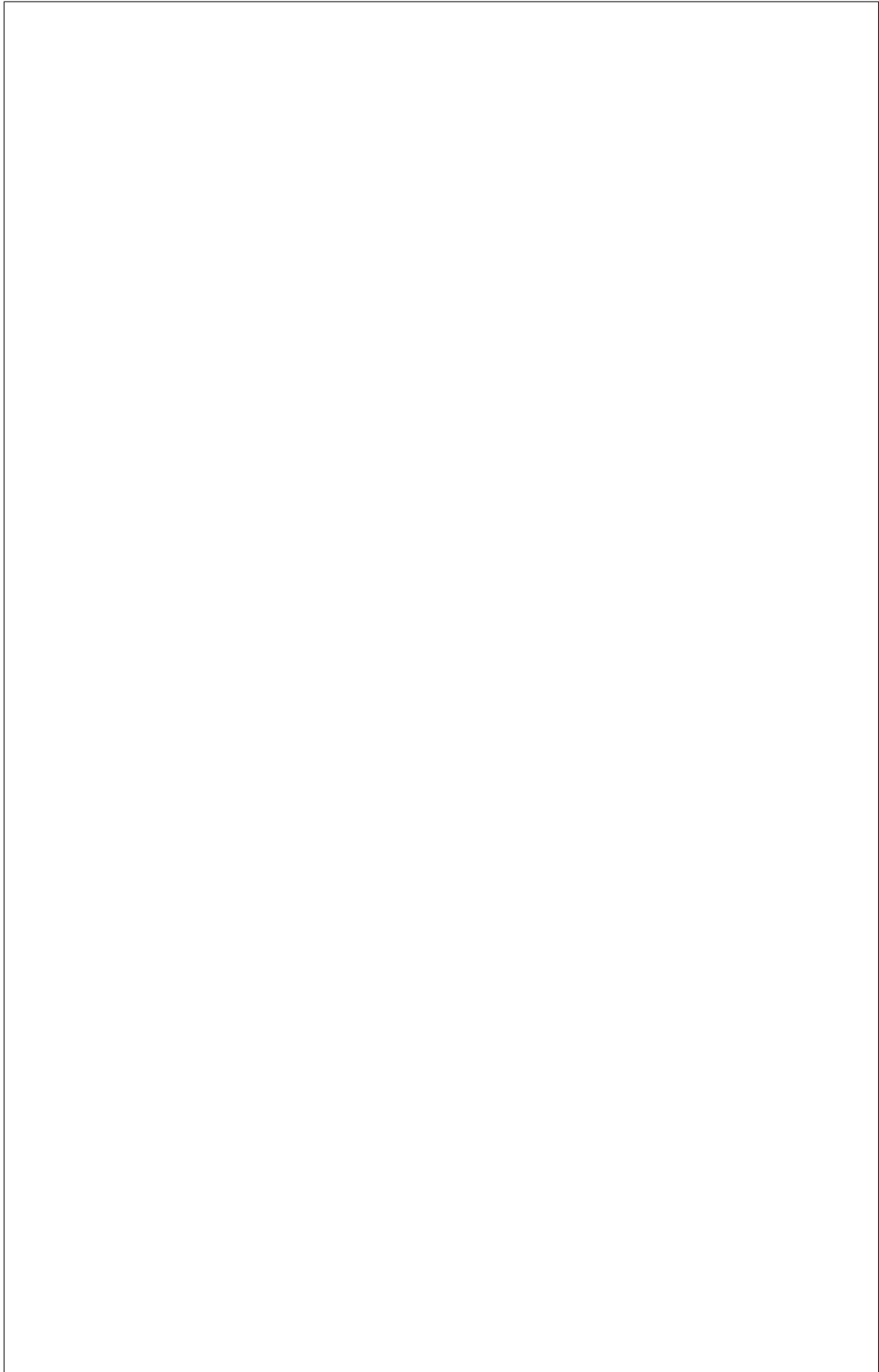
(continues on next page)

(continued from previous page)



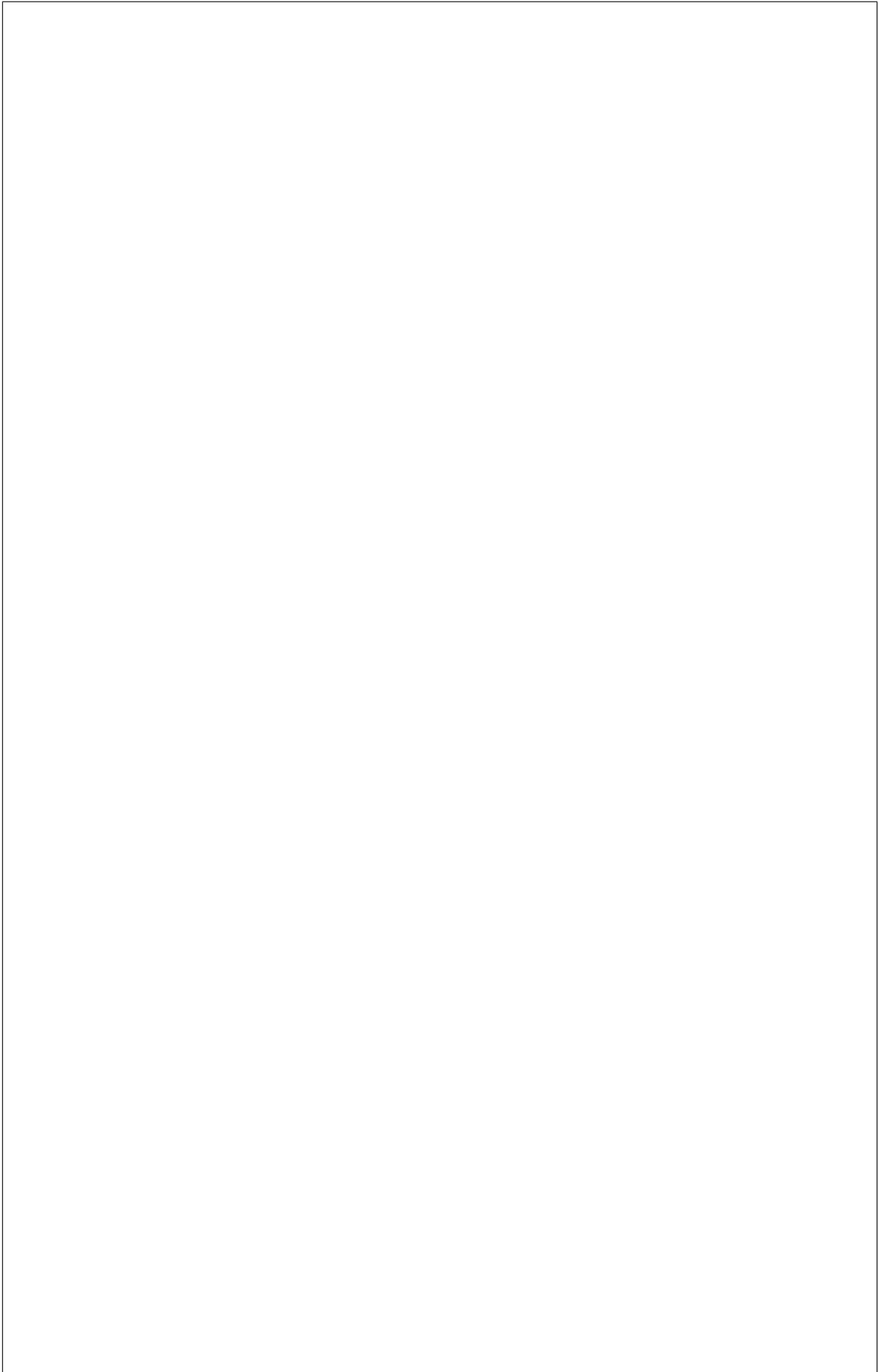
(continues on next page)

(continued from previous page)



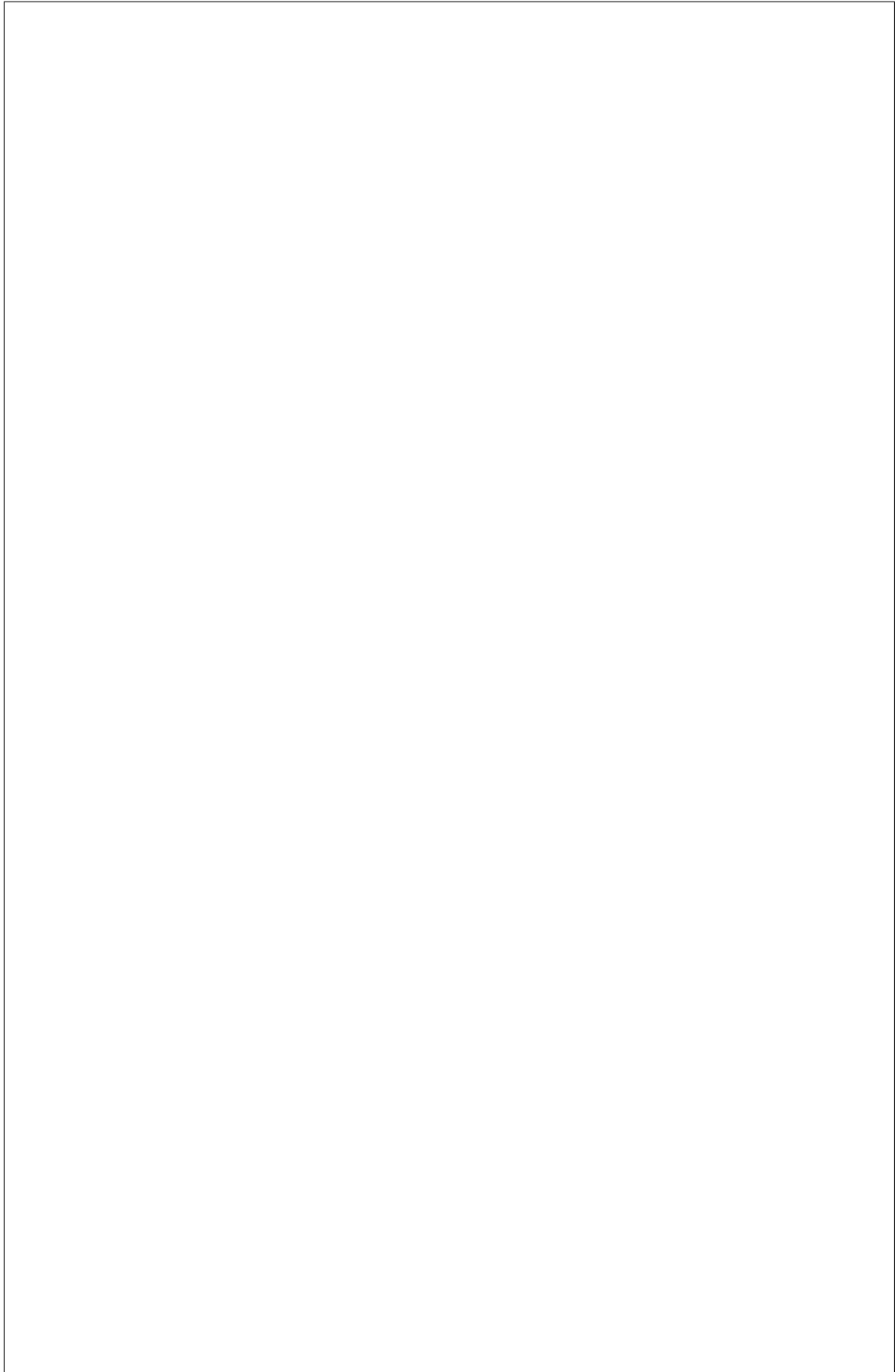
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



This  
 meth  
 mus  
 han-  
 dle:

-

con-  
 vert-  
 ing  
 from  
 an  
 olde  
 ver-  
 sion  
 to  
 a  
 new  
 ver-  
 sion

-

con-  
 vert-  
 ing  
 from  
 a  
 new  
 ver-  
 sion  
 to  
 an  
 olde  
 ver-  
 sion

-

mak  
 ing  
 sure  
 whe  
 con-  
 vert-  
 ing,  
 that  
 you  
 take  
 into  
 con-  
 sid-  
 er-  
 a-  
 tion

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

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.



that any new columns default to NULL. Test the migration out on a non-empty database to make sure that any new constraints dont cause the database to be locked out for normal operations.

[mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html](https://dev.mysql.com/doc/refman/5.7/en/innodb-create-index-overview.html). (You should also check older, widely deployed InnoDB versions for issues.) In the case of PostgreSQL, adding a foreign key may lock a whole table for writes.

ing  
new  
Aler  
bic  
mi-  
gra-  
tions  
whic  
mod  
ify  
ex-  
ist-  
ing  
mod  
els,  
mak  
sure

You  
can  
find  
an  
over  
on  
wha  
DDI  
op-  
er-  
a-  
tions  
may  
caus  
dow  
time  
in  
[https://](https://dev)  
dev.

Mak  
sure  
to  
add  
a  
re-  
lease  
note

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:

if  
there  
are  
any  
down  
relat  
con-  
cern

Back  
fill-  
ing  
de-  
fault  
val-  
ues,  
and  
mi-  
grat-  
ing  
data  
be-  
twee  
colu  
or  
be-  
twee  
ta-  
bles

- con-  
text:  
an  
ad-  
min  
con-  
text

- max  
this  
is  
used  
to  
limi  
the  
quer  
It  
is  
the

to migrate;  $\geq 0$ . If zero, all the objects will be migrated.

method, and

max  
i-  
mun  
num  
ber  
of  
ob-  
jects

It  
re-  
turn  
a  
two-  
tuple

- the  
to-  
tal  
num  
ber  
of  
ob-  
jects  
that  
need  
to  
be  
mi-  
grate  
at  
the  
start  
of  
the

- the  
num  
ber  
of  
mi-  
grate  
ob-  
jects

In  
this  
meth  
the

ver-  
 sion  
 col-  
 umn  
 can  
 be  
 used  
 to  
 se-  
 lect  
 and  
 up-  
 date  
 old  
 ob-  
 jects  
  
 The  
 meth  
 nam  
 shou  
 be  
 adde  
 to  
 the  
 list  
 of  
 ONL  
 in  
 iro  
 cmd  
 dbs  
 py.  
  
 The  
 meth  
 shou  
 be  
 re-  
 mov  
 in  
 the  
 next  
 nam  
 re-  
 lease  
 af-  
 ter  
 this  
 one.  
  
 Af-  
 ter

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.

### **ironic-dbsync upgrade command**

with the (new) release of ironic, before it will make any DB schema changes. If one or more objects are not compatible, the upgrade will not be performed.

(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 Acces Control - Testing**

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.

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`.

tional 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

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.

`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.

These  
tests  
ex-  
e-  
cute  
API  
calls  
thro  
the  
API  
layer  
us-  
ing  
the  
ap-  
pro-  
pri-  
ate  
verb  
and

The  
sec-  
ond  
as-  
pect  
of  
how  
this  
worl  
is  
were  
moc  
ing  
the  
con-  
duc-  
tor  
RPC  
get  
and

At  
the  
sam



would have been a heavier lift. As such, the tests largely look for one of the following error codes.

tion.

time  
wiri  
ev-  
ery-  
thing  
up  
to  
go  
from  
API  
to  
con-  
duc-  
tor  
code  
and  
back

- 200  
-

Got  
the  
item  
from  
the  
API  
-

This  
is  
an  
data  
drive  
in-  
ter-  
ac-

- 201  
-

Cre-  
ated  
-

This  
is  
data

drive  
 in-  
 ter-  
 ac-  
 tion.  
 These  
 are  
 rare.

- 204 -

Ac-  
 cept  
 -

This  
 is  
 a  
 data  
 drive  
 in-  
 ter-  
 ac-  
 tion.  
 These  
 are  
 rare.

- 403 - For  
 ac-  
 cess  
 was  
 de-  
 nied

- 404 - Not  
 iron  
 be-  
 com  
 scop  
 awar  
 thes  
 are  
 gen-  
 er-  
 ally  
 only  
 in  
 the  
 drive  
 API

ior. 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.

to the conductor.

### How to make changes or review these tests?

• **503 - Ser**  
 whe  
 a  
 re-  
 ques  
*has*  
 been  
 suc-  
 cess  
 fully  
 au-  
 then  
 ti-  
 cate  
 and  
 wou  
 have  
 been  
 sent  
 along

The  
 tests  
 cy-  
 cle  
 thro  
 the  
 var-  
 i-  
 ous  
 end-  
 poin  
 and  
 re-  
 peat  
 ing  
 pat-  
 terns  
 are  
 clear

ible. 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.

returned. This is important later with `owner` and `lessee` having slightly different views of the universe.

fields containing infrastructure internal addresses, these values will become hidden and additional tests will examine this.

404, or even 500 errors.

ery-  
thing

- Readers can always read but as we get into sensitive data later on such as

- Third party or external/Admins will find nothing but sadness in emp lists 403,

## **What is/will be tested?**

Access Control related capabilities will come in a series of phases, styles vary a little.

then human reviewed and values populated with expected values.

The  
idea  
is  
to  
in  
esse  
test  
as  
muc  
as  
pos-  
si-  
ble,  
how  
ever  
as  
thes  
tests  
Role  
Base

The  
first  
phas  
is  
"1e  
In  
esse  
thes  
are  
par-  
tially  
pro-  
gra-  
mat-  
i-  
cally  
gen-  
er-  
ated  
and

The  
sec-  
ond  
phas  
is  
re-

execute the `legacy` tests with the updated `oslo.policy` configuration to help enforce scopes. These tests will intentionally begin to fail in phase three.

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.

proach 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.

## Releasing IroniC Projects



that process here.

### Who is responsible for releases?

mov  
be-  
twee  
peo-  
ple,  
we  
doc-  
u-  
men

A  
full  
list  
of  
proj  
that  
iron  
man  
ages  
is  
avai  
able  
in  
the  
gov-  
er-  
nanc  
site.

The  
cur-  
rent  
PTL  
is  
ul-  
ti-  
mate  
re-  
spor  
si-  
ble  
for  
mak  
ing  
sure  
code  
gets  
re-

They may choose to delegate this responsibility to a liaison, which is documented in the [cross-project liaison wiki](#).

liaison must +1 the request for it to be processed.

### **Release process**

mented in the [Project Team Guide](#).

## What do we have to release?

ultimate source of truth for this is [projects.yaml](#) in the governance repository. These deliverables have varying release models, and these are defined in the [deliverables YAML files](#) in the releases repository.

The  
iron  
proj  
has  
a  
num  
ber  
of  
de-  
liv-  
er-  
able  
un-  
der  
its  
gov-  
er-  
nanc  
The  
ul-

In  
gen-  
eral,  
iron  
de-  
liv-  
er-  
able  
fol-  
low  
the  
cycl  
with  
inter  
re-  
leas  
mod

Non-client libraries

The following deliverables are non-client libraries

- ironlib
- metal-smith
- sushy

Client libraries

The following deliverables are client libraries

- python-iron
- python-iron-inspector

## Normal release

- sush  
cli

The  
fol-  
low-  
ing  
de-  
liv-  
er-  
able  
are  
Neu-  
tron  
plu-  
g-  
ins:

- netw  
bare

- netw  
gene  
swit

The  
fol-  
low-  
ing  
de-  
liv-  
er-  
able  
are  
Hori-  
zon  
plu-  
g-  
ins:

- iron  
ui

The  
fol-  
low-  
ing

de-  
 liv-  
 er-  
 able  
 are  
 Tem  
 pest  
 plu-  
 g-  
 ins:

- iron  
temp  
plug  
The  
fol-  
low-  
ing  
de-  
liv-  
er-  
able  
are  
ser-  
vice  
or  
treat  
as  
such

- bifro

- iron

- iron  
insp

- iron  
pron  
expo

- iron  
pyth  
ager

Independent

The  
fol-  
low-  
ing  
de-  
liv-  
er-  
able  
are  
re-  
lease  
in-  
de-  
pen-  
dent

- iron  
pyth  
ager  
buil
- mol  
iron
- sush  
tools
- tenk
- vir-  
tu-  
albn

Not released

The  
fol-  
low-  
ing  
de-  
liv-  
er-  
able  
do  
not  
need

Things to do before releasing

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).



added since the last release, update the REST API version history (`doc/source/contributor/webapi-version-history.rst`) to indicate that they were part of the new release.

is a named release) into `ironic/common/release_mappings.py`:

new  
API  
mi-  
crov-  
sion  
have  
been

- To  
sup-  
port  
rolli-  
up-  
grad  
add  
this  
new  
re-  
lease  
ver-  
sion  
(and  
re-  
lease  
nam  
if  
it

—

in  
REL  
mak  
a  
copy  
of  
the  
mas  
en-  
try,  
and  
re-  
nam  
the  
first  
mas  
en-  
try  
to

the new semver release version.

should be the same as that of the latest semver one (that you just added above).

`switch is made` to use the latest release from stable as the old release). Otherwise, once it is made, CI (the grenade job that tests new-release -> master) will fail.

cluding the related documentation.

### **How to propose a release**

that  
are  
clos  
to  
be  
merg  
or  
re-  
lease  
crit-  
i-  
cal.

This  
usu-  
ally  
in-  
clud  
im-  
por-  
tant  
bug  
fixes  
and/  
fea-  
tures  
that  
wed  
like  
to  
re-  
lease  
in-

The  
step  
that  
lead  
to  
a  
re-  
lease  
pro-  
posa  
are  
main  
man  
ual,

itself is almost a 100% automated process, accomplished by following the next steps:

automation resides.

subproject) grouped by release cycles.

(official) cycles (e.g. ironiC-python-agent-builder).

with this syntax:



(continues on next page)

The  
\_in  
di-  
rec-  
tory  
con-  
tains  
yam  
files  
for  
de-  
liv-  
er-  
able  
that  
are  
not  
bound  
to

- To  
check  
the  
char  
were  
about  
to  
re-  
lease  
we  
can  
use  
the  
tox  
en-  
vi-  
ron-  
men  
lis

(continued from previous page)

stable/train).

tory, to check the changes in the ussuri series for ironic-python-agent type:

(continues on next page)

(continued from previous page)

cess in the form of a tox environment called `new-release`.

- To up-date the de-liv-er-able file for the new re-lease we use a scrip pro-  
  
To get fa-mil-iar with it and see all the op-tions type  
  
Now base

need to decide on whether the next version will be major, minor (feature) or patch (bugfix).

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`.

on  
the  
list  
of  
char  
we  
foun  
in  
the  
prec  
dent  
step  
and  
the  
re-  
lease  
note  
we

Note  
that  
in  
this  
case  
ser  
is  
a  
code  
nam  
(trai  
us-  
suri)  
not  
a  
bran  
That  
is  
also  
valid

The  
--s  
arg  
is  
used  
only  
for  
bran  
ing



projects are not branched this way though.

ing the [new release model](#) for ironiC projects.

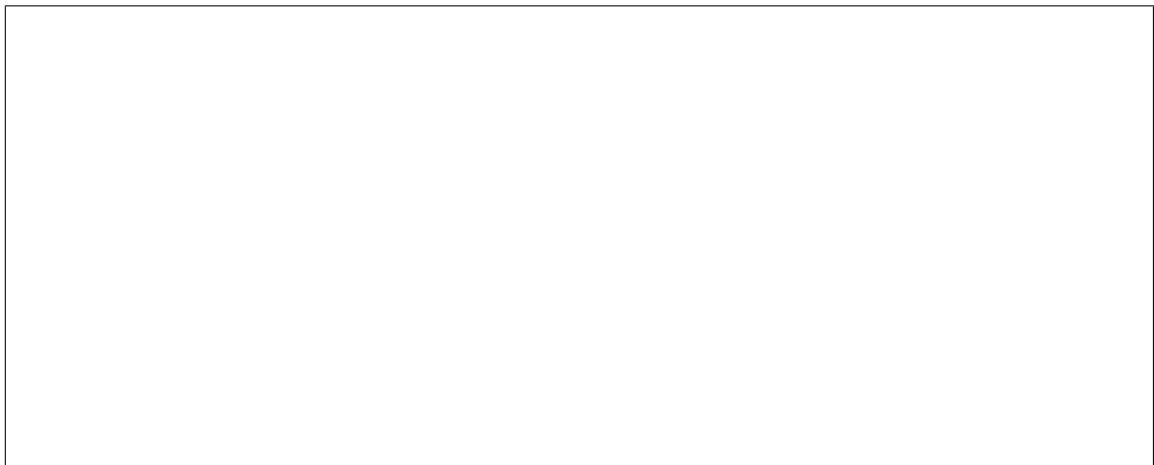
in  
the  
end  
of  
a  
cy-  
cle,  
in-  
de-  
pen-  
dent

The  
--i  
op-  
tion  
is  
used  
to  
cre-  
ate  
an  
in-  
ter-  
me-  
di-  
ate  
bug-  
fix  
bran  
fol-  
low-

To  
pro-  
pose  
the  
re-  
lease  
use  
the  
scrip  
to  
up-  
date  
the  
de-  
liv-  
er-  
able

mit the change, and propose it for review.

rent development branch), considering that the code name of the future stable release is wallaby, use:



file,  
then  
com

For  
ex-  
am-  
ple,  
to  
pro-  
pose  
a  
mi-  
nor  
re-  
lease  
for  
iron  
in  
the  
mas-  
ter  
bran-  
(cur-

Re-  
men-  
ber  
to  
use  
a  
mea-  
ing-  
ful  
topic  
usu-  
ally

erable, the new version and the branch, if applicable.

ironic 1.2.3 for ussuri

us-  
ing  
the  
nam  
of  
the  
de-  
liv-

A  
goo  
com  
mit  
mes  
sage  
ti-  
tle  
shou  
also  
in-  
clud  
the  
sam  
for  
ex-  
am-  
ple  
Re-  
lease

- As  
an  
op-  
tion:  
step  
we  
can  
use  
tox  
-e  
lis  
to  
doub  
chec  
the  
char  
be-  
fore  
sub-

mitting them for review.

some sanity-checks, but since everything is scripted, there shouldnt be any issue.

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.

Also  
tox  
tox  
-e  
val  
val  
(it  
migl  
take  
take  
a  
whil  
to  
run  
run  
base  
on  
the  
num  
ber  
of  
char  
does  
som

All  
the  
scrip  
are  
de-  
sign  
and  
main  
taine  
by  
the  
re-  
lease  
team  
in  
case  
of  
ques  
tions  
or

- Af-  
ter  
the  
char  
is

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**

need to be made.

includes:

a  
num  
ber  
of  
char  
that  
need  
to  
be  
ap-  
prov  
This

•  
In  
the  
new  
sta-  
ble  
bran

-  
a  
char  
to  
poin  
.  
git  
at  
the  
bran

-  
a  
char  
to  
up-  
date  
the  
up-  
per  
con-  
strai  
file  
used  
by  
tox

•  
In  
the  
mas

ter  
bran

—

up-  
dat-  
ing  
the  
re-  
lease  
note  
RST  
to  
in-  
clud  
the  
new  
bran

The  
gen-  
er-  
ated  
RST  
does  
not  
in-  
clud  
the  
ver-  
sion  
rang  
in  
the  
ti-  
tle,  
so  
we  
typ-

ically submit a follow-up patch to do that. An example of this patch is [here](#).

—

up-  
date  
the  
*tem-  
plate*  
in  
.zuu  
or  
zuul  
The

An example of this patch is [here](#).

up-  
date  
is  
nec-  
es-  
sary  
to  
use  
the  
job  
for  
the  
next  
re-  
lease  
*oper*  
*pyth*  
<*ne*.  
*jobs*

We  
need  
to  
sub-  
mit  
patc  
for  
char  
in  
the  
sta-  
ble  
bran  
to:

- up-  
date  
the  
iron  
de-  
vs-  
tack  
plu-  
gin  
to  
poin  
at  
the  
bran  
tar-



example of this patch is [here](#).

versions of any openstack projects (that branch) documents. As of Pike release, the only outlier is [diskimage-builder](#).

unsupported API tempest tests are skipped on stable branches. E.g. [patch 495319](#).

sion. See [example](#) and [pbr documentation](#) for details.

to  
 sub-  
 mit  
 patc  
 for  
 char  
 on  
 mas  
 ter  
 to:  
 •  
 cre-  
 ate  
 an  
 emp  
 com  
 mit  
 with  
 a  
 Sem  
 tag  
 to  
 bum  
 the  
 gen-  
 er-  
 ated  
 mi-  
 nor  
 ver-  
 •  
 to  
 sup-  
 port  
 rolli  
 up-  
 grad  
 sinc  
 the  
 re-  
 lease  
 was  
 a  
 nam  
 re-  
 lease  
 we  
 need  
 to

these changes. Note that we need to wait until *after* the switch in grenade is made to test the latest release (N) with master (e.g. [for stable/queens](#)). Doing these changes sooner after the ironiC release and before the switch when grenade is testing the prior release (N-1) with master, will cause the tests to fail. (You may want to ask/remind infra/qa team, as to when they will do this switch.)

named release. Since we support upgrades between adjacent named releases, the master branch will only support upgrades from the most recent named release to master.

sponding 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**

master branch. [Example for Queens](#).

## **Bifrost**

re-  
mov  
any  
mod  
class  
nam  
from  
iro  
cmd  
db  
NEW

As  
**iron**  
**tem**  
**plug**  
is  
bran  
less,  
we  
need  
to  
sub-  
mit  
a  
patc  
addi  
sta-  
ble  
jobs  
to  
its

Bifr  
need  
to  
be  
up-  
date  
to  
in-  
stall  
de-  
pen-

ample for [Victoria](#). The upper constraints file referenced in `scripts/install-deps.sh` needs to be updated to the new release.

### For all releases

den-  
cies  
us-  
ing  
the  
sta-  
ble  
bran  
Ex-

For  
all  
re-  
lease  
whe  
or  
not  
it  
re-  
sults  
in  
a  
sta-  
ble  
bran

- up-  
date  
the  
spec  
repo  
to  
marl  
any  
spec  
com  
plete  
in  
the  
re-  
lease  
as  
im-  
ple-  
men

• re-mov  
any  
-  
2s  
on  
patc  
that  
were  
bloc  
un-  
til  
af-  
ter  
the  
re-  
leas

Ironic Governance Structure

The  
iron  
proj  
man  
ages  
a  
num  
ber  
of  
repo  
i-  
to-  
ries  
that  
con-  
tribu  
to  
our  
mis-  
sion

The full list of repositories that ironic manages is available in the [governance site](#).

## What belongs in ironiC governance?

library that implements a standard to manage hardware from multiple vendors (such as IPMI or redfish) is okay.

For  
a  
repo  
i-  
tory  
to  
be  
part  
of  
the  
Iron  
proj

- It  
mus  
com  
ply  
with  
the  
TCs  
rules  
for  
a  
new  
proj

- It  
mus  
not  
be  
in-  
tend  
for  
use  
with  
only  
a  
sin-  
gle  
ven-  
dors  
hard  
ware  
A  
li-

itory where only a single company is contributing is okay, with the hope that other companies will contribute after joining the ironiC project.

• It  
mus  
align  
with  
Iron  
ics  
mis-  
sion  
state  
men  
  
Lack  
of  
con-  
trib-  
u-  
tor  
di-  
ver-  
sity  
is  
a  
chic  
egg  
prob  
lem,  
and  
as  
such  
a  
repo  
  
Rep  
i-  
to-  
ries  
that  
are  
no  
long  
main  
taine  
shou  
be  
prun  
from  
gov-  
er-  
nanc



## Proposing a new project to ironiC governance

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.

and directly to the driver. Some guidelines on how to implement this are provided below. Driver authors are strongly encouraged to talk with the developer community about any implementation using this functionality.

### **Pluggable Drivers**

drivers, and operators to use third-party drivers or write their own. A driver is built at runtime from a *hardware type* and *hardware interfaces*. See [Enabling drivers and hardware types](#) for a detailed

To  
fa-  
cil-  
i-  
tate  
that,  
we  
also  
pro-  
vide  
the  
mea-  
for  
API  
calls  
to  
be  
pass  
thro  
iron

Iron  
sup-  
port  
a  
plug  
gabl  
drive  
mod  
This  
al-  
lows  
con-  
trib-  
u-  
tors  
to  
eas-  
ily  
add  
new

explanation of these concepts.

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:

```

#
/
u
b
e
p

imp
p
a
p
pri
[
n
f
p
i
p
i
e
p
"
h
t
"
i
n
p
n
s
"

```

(continues on next page)

(continued from previous page)

the following command against that API end point:

## Writing a hardware type

A  
list  
of  
drive  
en-  
able  
in  
a  
run-  
ning  
Iron  
ser-  
vice  
may  
be  
foun  
by  
is-  
su-  
ing

ba  
→ d  
→ 1

A  
hard  
ware  
type  
is  
a  
Pyth  
class  
in-  
her-  
it-  
ing  
*irc*  
*dri*  
*har*  
*Abs*  
and  
liste  
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`.

implementation is provided by the `GenericHardware` base class.

The  
min-  
i-  
mun  
re-  
quir  
in-  
ter-  
face  
are:

- *`boot`*  
that  
spec  
i-  
fies  
how  
to  
boot  
ram  
and  
in-  
stan  
on  
the  
hard  
ware  
A  
gene  
pxe

- *`de-  
ploy`*  
that  
or-  
ches  
trate  
the  
de-  
ploy  
men  
A  
few  
com  
mon  
im-

are provided by the `GenericHardware` base class.

to indicate that it is a deploy step. Conventionally, the `deploy` method uses a priority of 100.



face

- - pow*
  - im-
  - ple-
  - men
  - pow*
  - ac-
  - tions
  - for
  - the
  - hard
  - ware
  - These
  - com
  - mon
  - im-
  - ple-
  - men
  - ta-
  - tions

may be used, if supported by the hardware:

- - irc*
  - dri*
  - mod*
  - ipm*
  - IPM*

- - irc*
  - dri*
  - mod*
  - reo*
  - pow*
  - Reo*

Oth-  
er-  
wise  
you  
need  
to  
write  
your  
own  
im-  
ple-  
men  
ta-

`base.PowerInterface` and providing missing methods.

not return until the power action is finished or errors out.

tion  
by  
sub-  
class  
ing  
*irc*  
*dri*

---

**Note**  
Pow  
ac-  
tions  
in  
Iron  
are  
bloc  
ing  
-

meth  
ods  
of  
a  
pow  
in-  
ter-  
face  
shou

- *man*  
*age-*  
*men*  
im-  
ple-  
men  
ad-  
di-  
tion  
out-  
of-  
band  
man  
age-  
men  
ac-  
tions  
such



setting a boot device. A few common implementations exist and may be used, if supported by the hardware:

as

- *ironic.drivers.modules.ipm*  
*IPM*

- *ironic.drivers.modules.recman*  
*RecMan*

Some hardware warps type such as snm do not support out-of-banc man age-men They use the

fake implementation in *ironic.drivers.modules.fake.FakeManagement* instead.

Oth-er-wise you need to write your own im-ple-men

*base.ManagementInterface* and providing missing methods.

ported interfaces. These lists are prioritized, with the most preferred implementation first. For example:

(continues on next page)

tion  
by  
sub-  
class  
ing  
*irc*  
*driv*

Combine the interface in a hardware type by populating the lists of sup-

cla

→ M

→ G

↳

↳

↳

↳

↳

↳  
↳  
↳  
↳  
↳ **d**  
↳ s  
↳ m  
↳ i













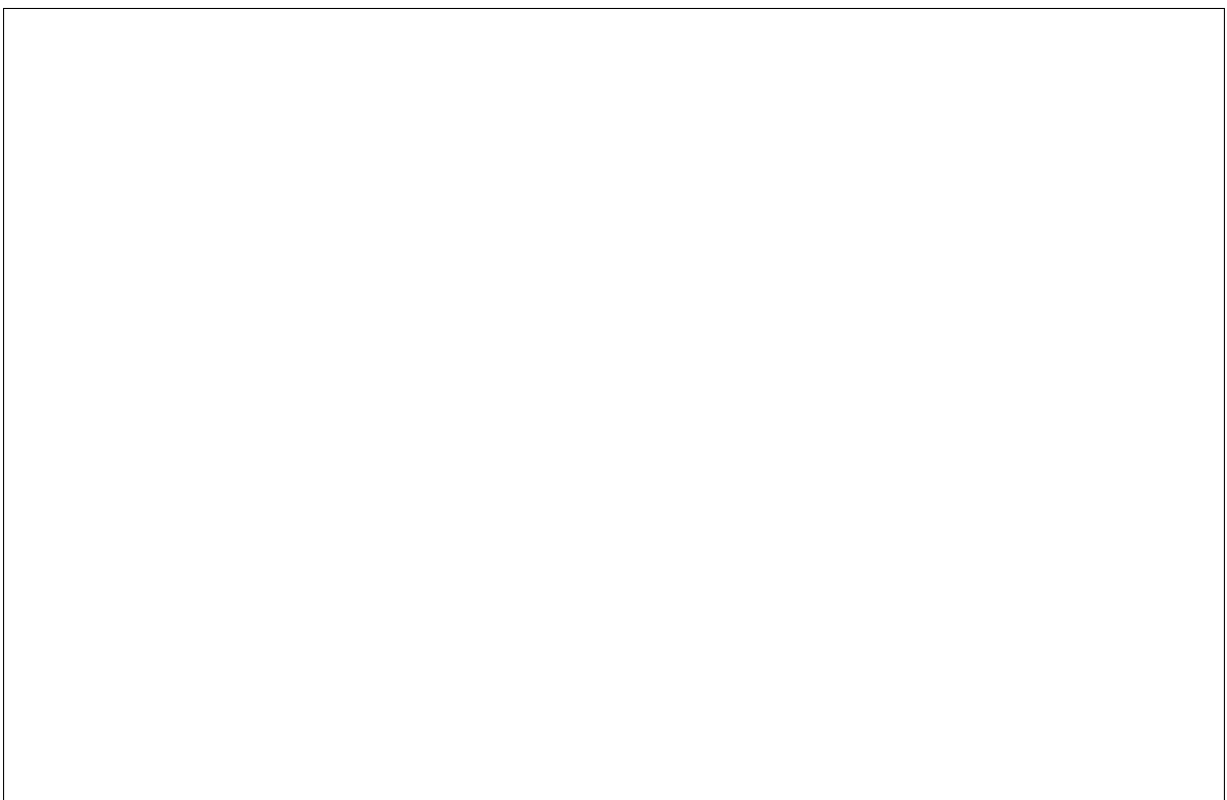
(continues on next page)

(continued from previous page)

---

**Note:** In this example, all interfaces, except for management and power are taken from the `GenericHardware` base class.

try points for them in the `setup.cfg` file:



(continues on next page)

(continued from previous page)



## Supported Drivers

commit) please consult the *drivers page*.

## Node Vendor Passthru

`UUID or Name>/vendor_passthru?method={METHOD}` endpoint. Beyond basic checking, IroniC does not introspect the message body and simply passes it through to the relevant driver.

Driv  
may  
im-  
ple-  
men  
a  
pass  
API  
whic  
is  
ac-  
ces-  
si-  
ble  
via  
the  
/  
v1/  
nod  
<No

A  
meth

- can  
sup-  
port  
one  
or  
more  
HTT  
meth  
ods  
(for  
ex-  
am-  
ple,  
GET  
POS

- is  
asyn  
chro  
or  
syn-  
chro

cate that the request was received, accepted and is being acted upon. No body is returned in the response.

that the request was fulfilled. The response may include a body.

- For  
 asyn  
 chro  
 meth  
 ods,  
 a  
 202  
 (Ac-  
 cept  
 HTT  
 sta-  
 tus  
 code  
 is  
 re-  
 turn  
 to  
 in-  
 di-  
  
 - For  
 syn-  
 chro  
 meth  
 ods,  
 a  
 200  
 (OK  
 HTT  
 sta-  
 tus  
 code  
 is  
 re-  
 turn  
 to  
 in-  
 di-  
 cate  
  
 • can  
 re-  
 quire  
 an  
 ex-  
 clu-  
 sive

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.

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. Ironiic makes no guarantees about backwards compatibility; this is solely up to the discretion of each drivers author.

lock  
on  
the  
node  
This  
only  
oc-  
curs  
if  
the  
meth  
does

This  
end-  
poin  
ex-  
pose  
a  
node  
drive  
di-  
rectl  
and  
as  
such  
it  
is  
ex-  
pres  
not  
part  
of

To  
get  
in-  
for-  
ma-  
tion  
about  
all  
the  
meth  
ods  
avai  
able



a particular node, you can issue an HTTP GET request:



name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

via  
the  
ven-  
dor\_  
end-  
poin  
for

GET  
→/  
→v  
→n  
→  
→<  
→U  
→U  
→  
→n  
→n  
→/  
→v  
→p  
→m

The  
re-  
spor  
JSO  
body  
will  
con-  
tain  
in-  
for-  
ma-  
tion  
for  
each  
meth  
such  
as  
the  
meth  
ods

Driver Vendor Passthru

<driver name>/vendor\_passthru?method={METHOD}.

Driv  
may  
im-  
ple-  
men  
an  
API  
for  
re-  
ques  
not  
re-  
latec  
to  
any  
node  
at  
/  
v1/  
dri

A  
meth

- can support one or more HTTP methods (for example, GET, POST, PUT, PATCH, DELETE, etc.)
- is asynchronous or synchronous

-

cate that the request was received, accepted and is being acted upon. No body is returned in the response.

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.

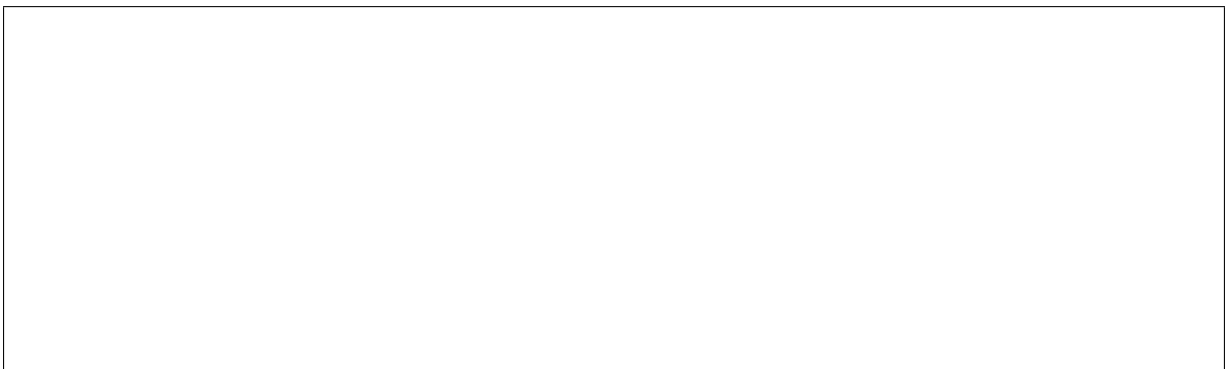
For  
asyn  
chro  
meth  
ods,  
a  
202  
(Ac-  
cept  
HTT  
sta-  
tus  
code  
is  
re-  
turn  
to  
in-  
di-

For  
syn-  
chro  
meth  
ods,  
a  
200  
(OK  
HTT  
sta-  
tus  
code  
is  
re-  
turn  
to  
in-  
di-  
cate

Iron  
mak  
no  
guar  
an-

point. That is left up to each drivers author.

you can issue an HTTP GET request:



tees  
about  
the  
se-  
man  
tics  
of  
the  
mes  
sage  
BOI  
sent  
to  
this  
end-

To  
get  
in-  
for-  
ma-  
tion  
about  
all  
the  
meth  
ods  
avai  
able  
via  
the  
drive  
ven-  
dor\_  
end-  
point

GET  
→/  
→v  
→d  
→  
→<  
→n  
→/  
→v  
→p  
→m

The  
re-

name, a description, the HTTP methods supported, and whether its asynchronous or synchronous.

## Vendor Methods

a driver.

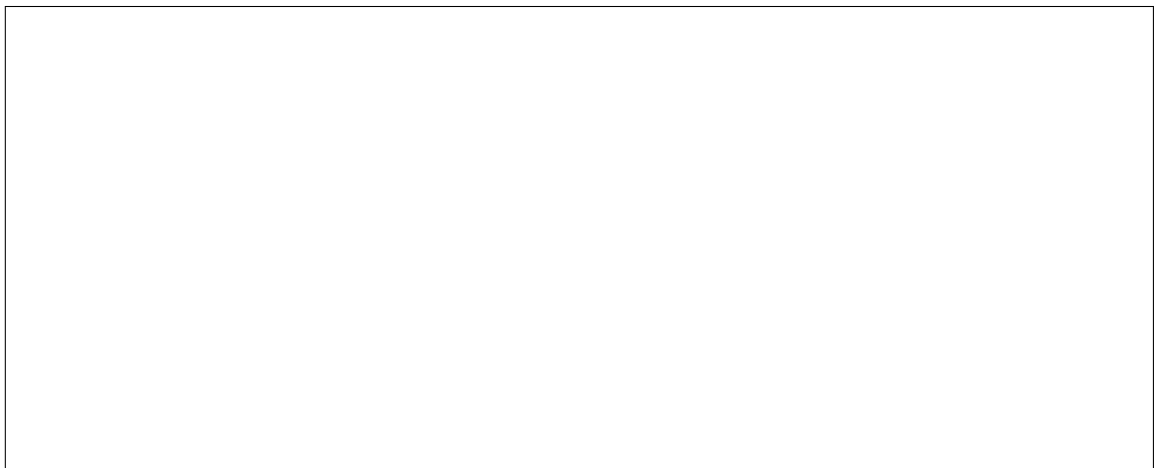
spor  
JSO  
body  
will  
con-  
tain  
in-  
for-  
ma-  
tion  
for  
each  
meth  
such  
as  
the  
meth  
ods

This  
doc-  
u-  
men  
is  
a  
quic  
tu-  
to-  
rial  
on  
writ  
ing  
ven-  
dor  
spe-  
cific  
meth  
ods  
to

The  
first  
thing  
to  
note  
is  
that

node vendor passthru and a node vendor passthru.

which is not specific to a Node. For example, lets say the driver *ipmi* exposed a method called *authentication\_types* that would return what are the authentication types supported. It could be accessed via the Ironi API like:



(continues on next page)

(continued from previous page)

terface implementation for a given hardware type. This limitation will be lifted in the future.

node basis. For example the same driver *ipmi* exposing a method called *send\_raw* that would send raw bytes to the BMC, the method also receives a parameter called *raw\_bytes* which the value would be the bytes to be sent. It could be accessed via the IroniC API like:

```
↪<node UUID>/vendor_passthru/send_raw
```

## Writing Vendor Methods

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.  
The  
first  
thing  
to  
do  
is

```
cla  
↪E  
  
↪  
↪  
↪  
↪  
↪d  
↪g  
↪p
```



(continued from previous page)



of <property>:<description> telling in the description whether that property is required or optional so

the node can be manageable by that driver. For example, a required property for a *ipmi* driver would be *ipmi\_address* which is the IP address or hostname of the node. We are returning an empty dictionary in our example to make it simpler.

passed to the vendor methods. IroniC will not introspect into what is passed to the drivers, its up to the developers writing the vendor method to validate that data.

*tion\_types* which will be exposed on the driver vendor passthru endpoint; And the *send\_raw* method that will be exposed on the node vendor passthru endpoint:



(continues on next page)

The  
val-  
i-  
date  
meth  
meth  
is  
re-  
spon  
si-  
ble  
for  
val-  
i-  
dat-  
ing  
the  
pa-  
ram-  
e-  
ters

Lets  
ex-  
tend  
the  
Ex-  
am-  
pleV  
dor  
class  
to  
sup-  
port  
two  
meth  
ods,  
the  
au-  
then-  
ti-  
ca-

cla  
↪ E

(continues on next page)

(continues on next page)

(continued from previous page)



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.

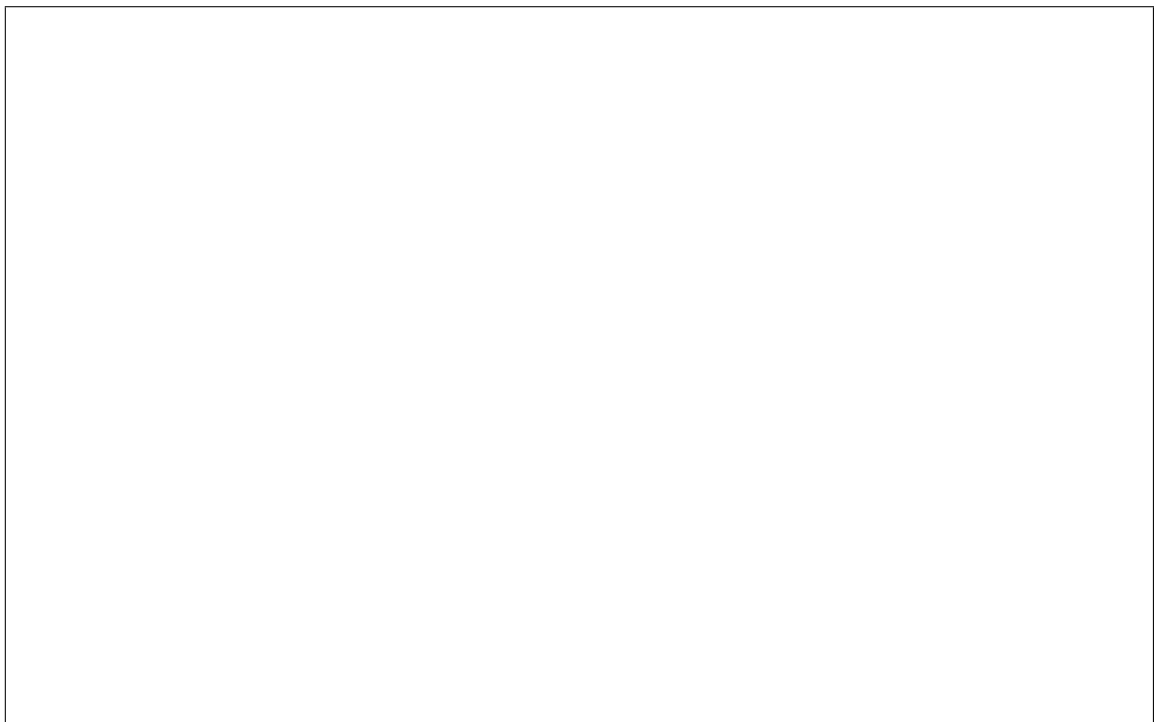
or  
drive  
ven-  
dor  
pass  
meth  
meth  
is  
prett  
much  
the  
same  
the  
only  
dif-  
fer-  
ence

Both  
dec-  
o-  
ra-  
tors  
ac-  
cept  
thes  
pa-  
ram-  
e-  
ters:

- [http](http://...).  
A  
list  
of  
wha  
the  
HTT  
meth  
ods  
sup-  
port  
by  
that  
ven-  
dor  
func  
tion.  
To

what HTTP method that function was invoked with, a *http\_method* parameter will be present in the *kwargs*. Supported HTTP methods are *POST*, *PUT*, *GET* and *PATCH*.

use a different name this parameter is where this name can be set. For example:



posed to do. Defaults to (empty string).

faults to True (Asynchronously).

con-  
tain-  
ing  
a  
nice  
de-  
scrip-  
tion  
about  
what  
that  
meth-  
is  
sup-

- asyn  
A  
bool  
valu  
to  
de-  
ter-  
min  
whe  
this  
meth  
shou  
run  
asyn  
chro  
or  
syn-  
chro  
De-

**Note**  
This  
pa-  
ram-  
e-  
ter  
was  
pre-  
vi-  
ousl  
calle  
asyn



ter:

lock on a node between validate() and the beginning of method execution. For synchronous methods, the lock on the node would also be kept for the duration of method execution. Defaults to True.



The  
node  
ven-  
dor  
pass  
dec-  
o-  
ra-  
tor  
(@p  
also  
ac-  
cept  
the  
fol-  
low-  
ing  
pa-  
ram-  
e-

- re-  
quir  
A  
bool  
valu  
de-  
ter-  
min-  
ing  
whe  
this  
meth  
shou  
re-  
quir  
an  
ex-  
clu-  
sive

**War**  
Plea  
avoi  
hav-  
ing

does talk to a BMC; BMCs are flaky and very easy to break.

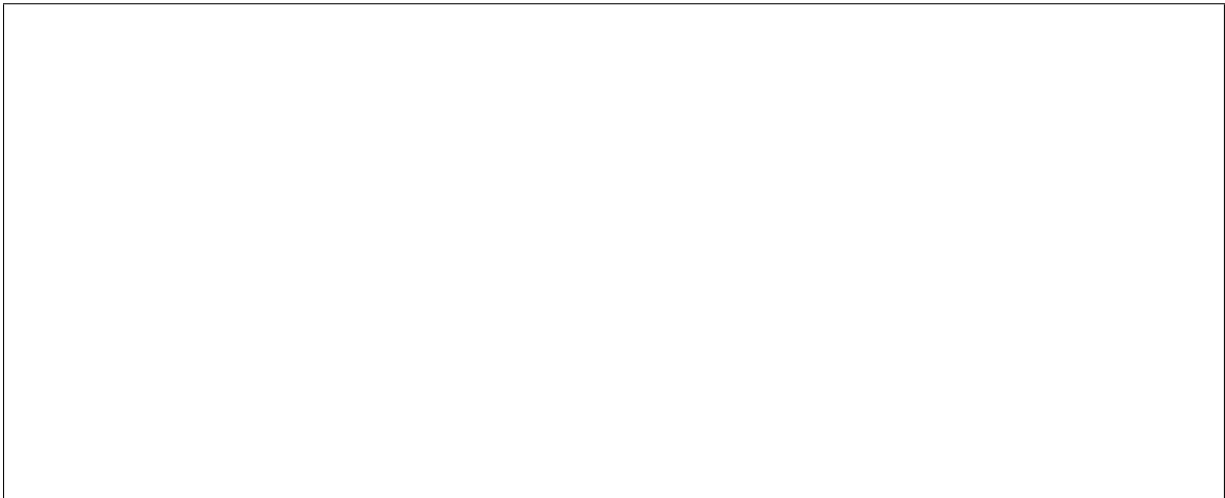
starvation of the thread pool, resulting in a denial of service.

a  
syn-  
chro  
meth  
meth  
for  
slow  
runn  
op-  
er-  
a-  
tions  
**or**  
if  
the  
meth

**War**  
Each  
asyn  
chro  
re-  
ques  
con-  
sum  
a  
worl  
threa  
in  
the  
iro  
pro-  
cess  
This  
can  
lead  
to

Give  
the  
new  
ven-  
dor  
in-  
ter-  
face  
im-

an entry point for it in the `setup.cfg`:



ware types, for example:



(continues on next page)

ple-  
men-  
ta-  
tion  
a  
hum  
frier  
nam  
and  
cre-  
ate

iro  
→h  
→i  
→v  
→=  
└  
→└  
→└  
→└  
→e  
→=  
→i  
→d  
→m  
→e

Fi-  
nally  
add  
it  
to  
the  
list  
of  
sup-  
port  
ven-  
dor  
in-  
ter-  
face  
for  
rel-  
e-  
vant  
hard

cla  
→E  
→G

(continued from previous page)



Backwards Compatibility

However, for your users sakes, we highly recommend that you do so.

same HTTP code is being returned to the user.

cates this.

## Developing BIOS Interface

class inheriting from the BIOSInterface class:

(continues on next page)

(continued from previous page)

that supports BIOS settings should also implement the following three methods:

• Im-  
ple-  
men  
a  
meth  
nam  
cac  
This  
meth  
stor  
BIO  
set-  
ting  
to  
the  
bio  
ta-  
ble  
dur-

ing cleaning operations and updates the `bios_settings` table when `apply_configuration` or `factory_reset` are successfully called.



(continues on next page)



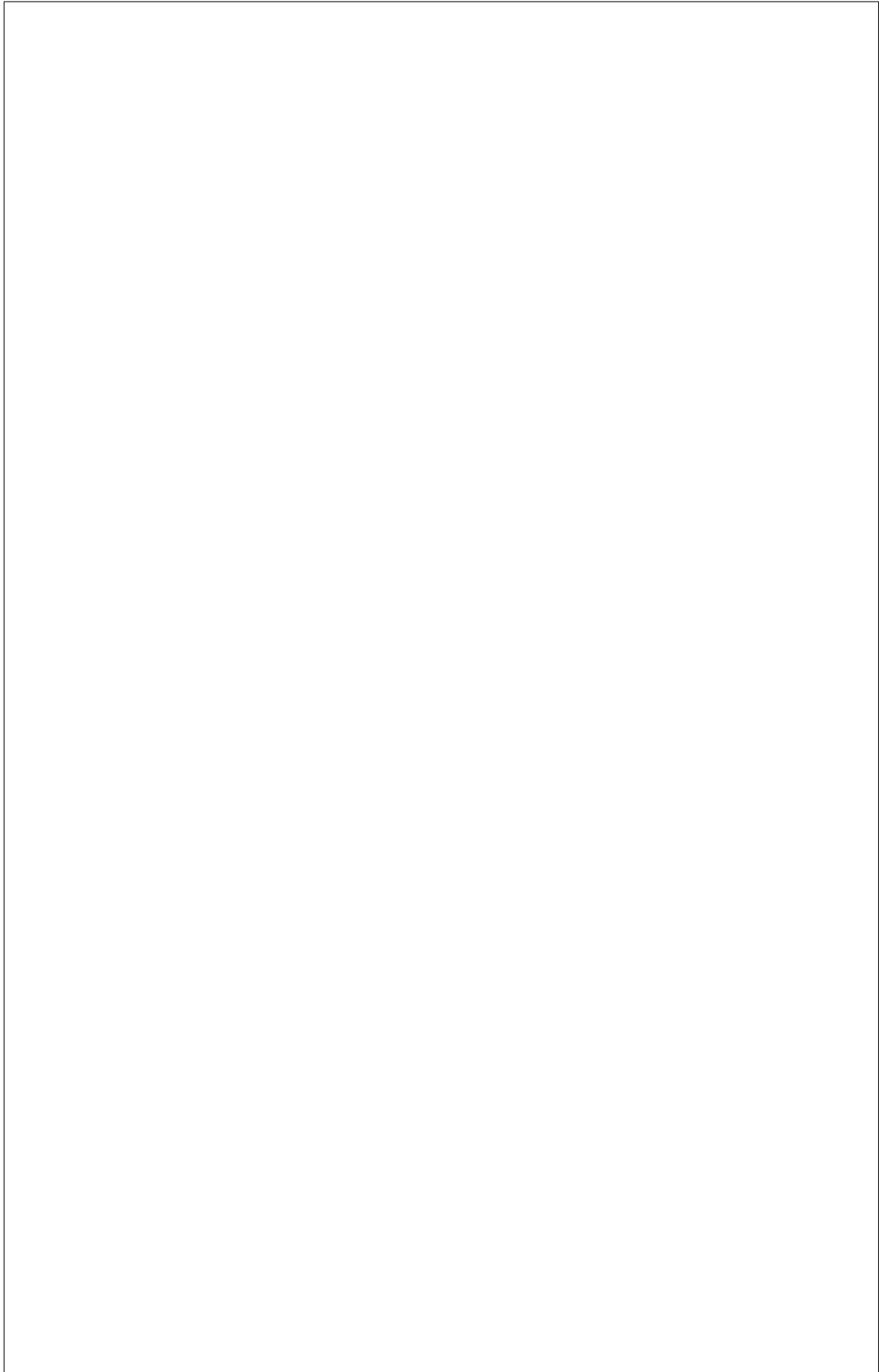
(continued from previous page)

```
↩ _.__name__,
```

```
↩import driver library"))
```

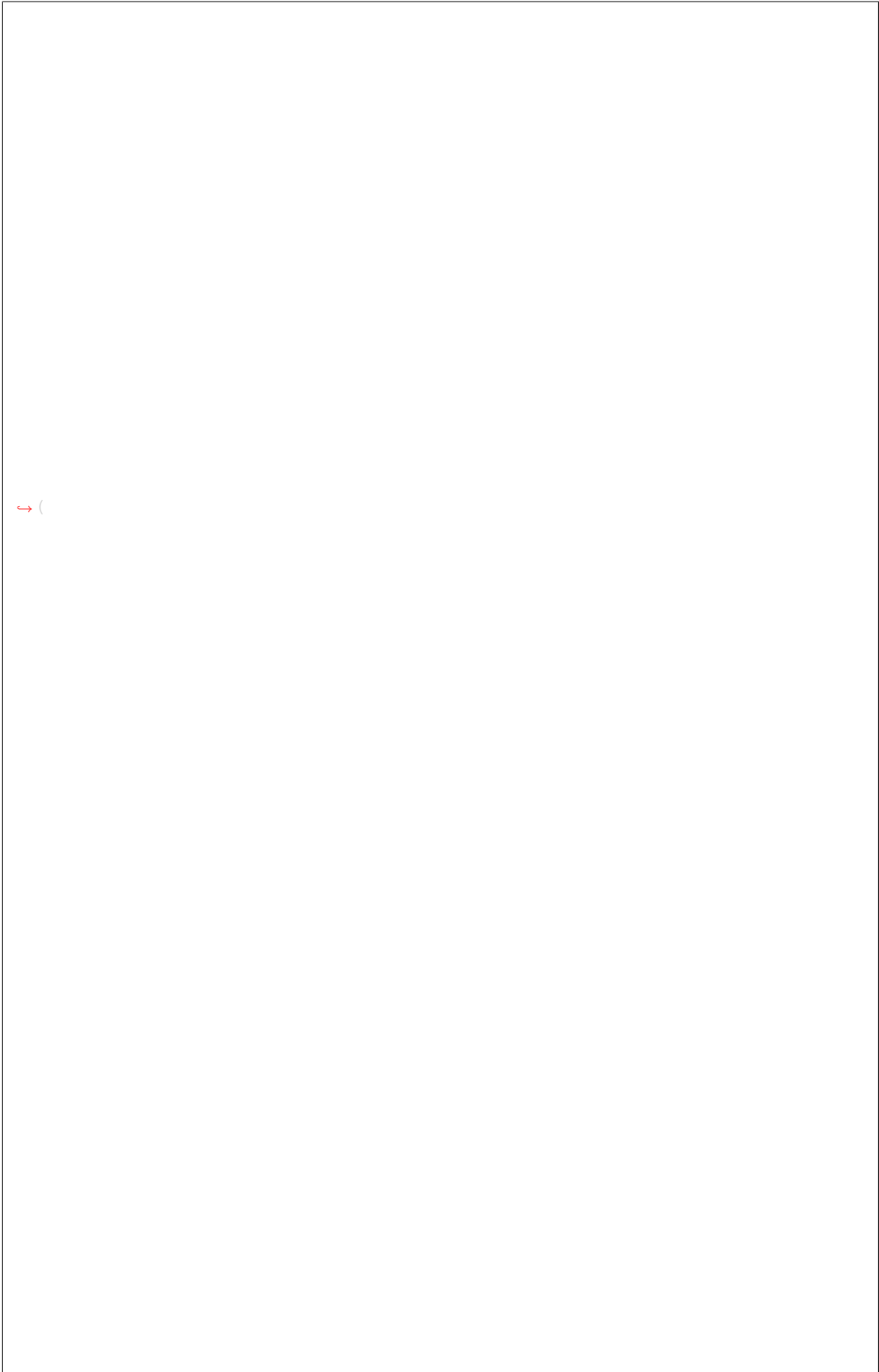
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



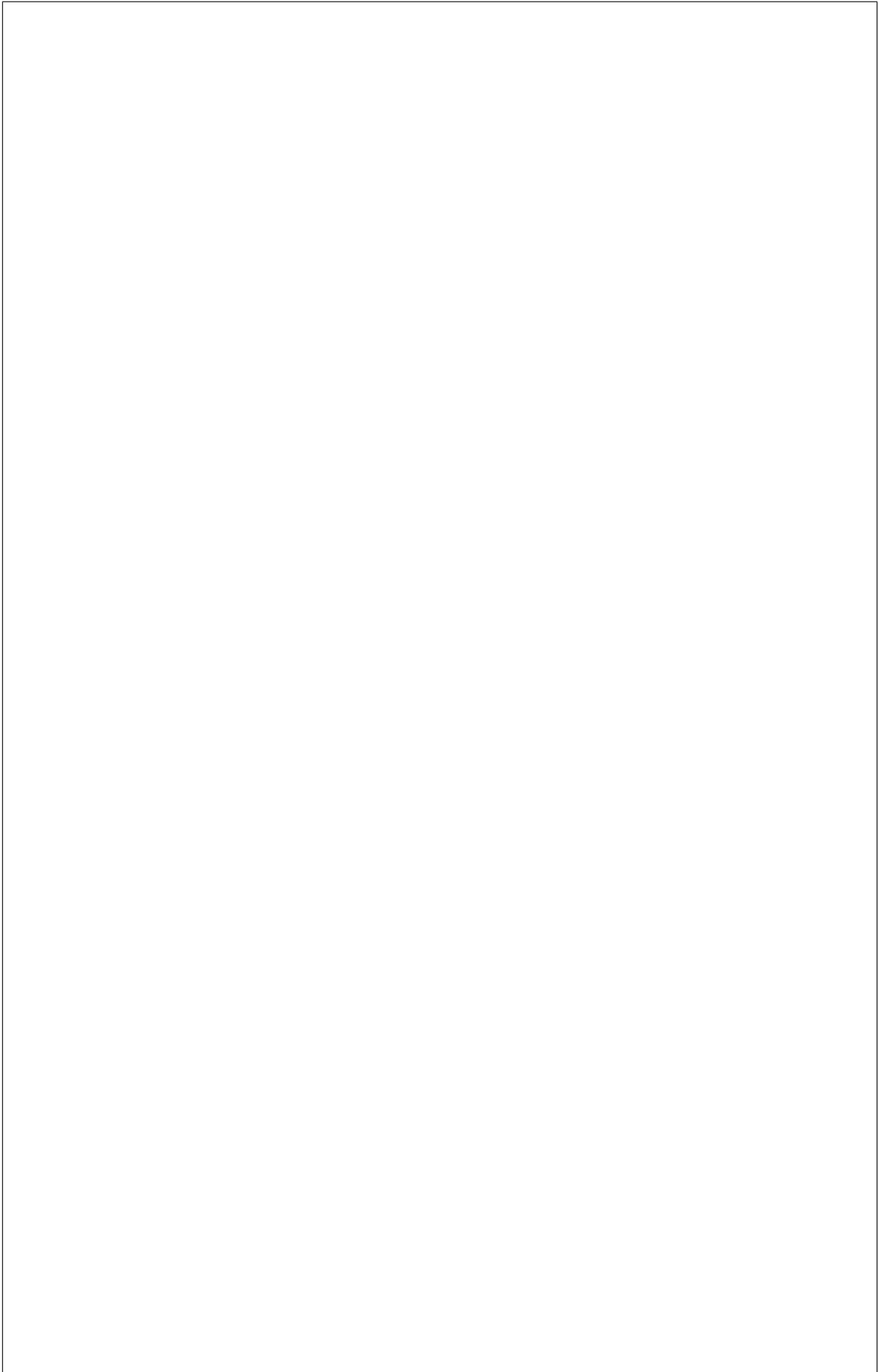
(continues on next page)

(continued from previous page)

```
↩id, create_list)
```

(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

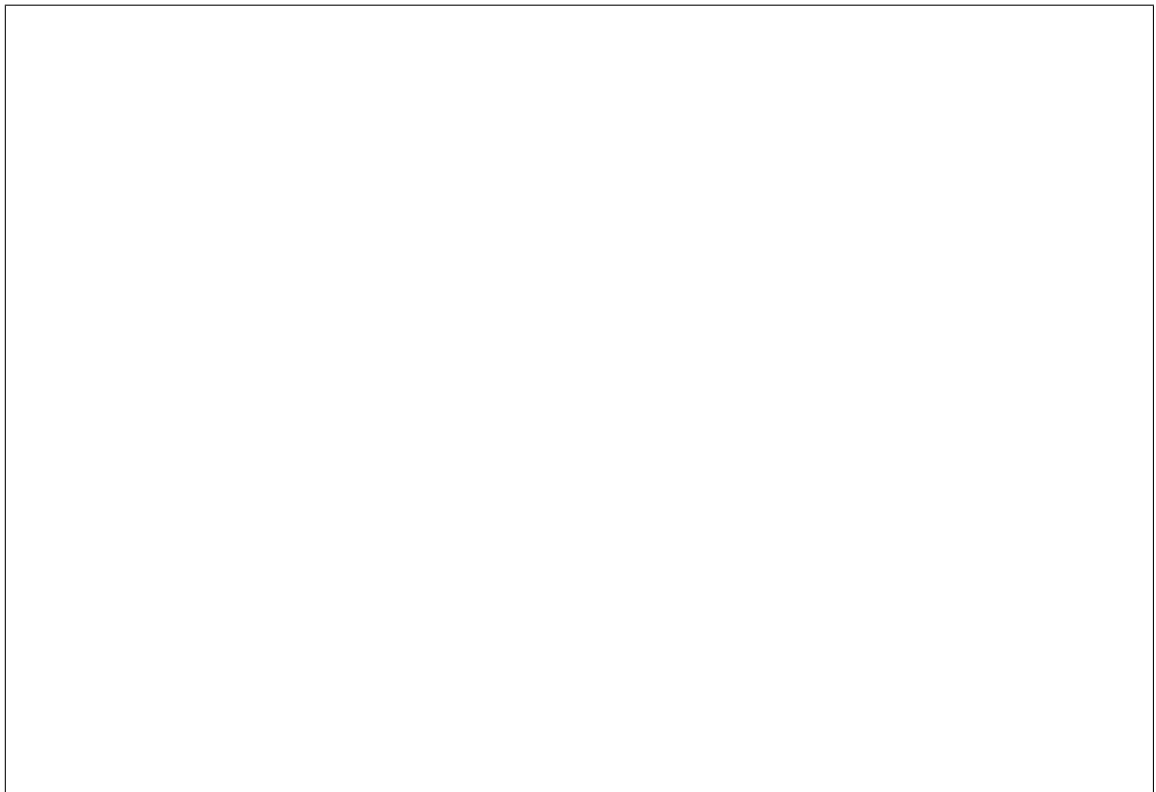
```
↪id, delete_names)
```

---

**Note**  
dri  
cli  
cli  
is  
ven-  
dor  
spe-  
cific  
li-  
brary  
to  
con-  
trol  
and  
man  
age

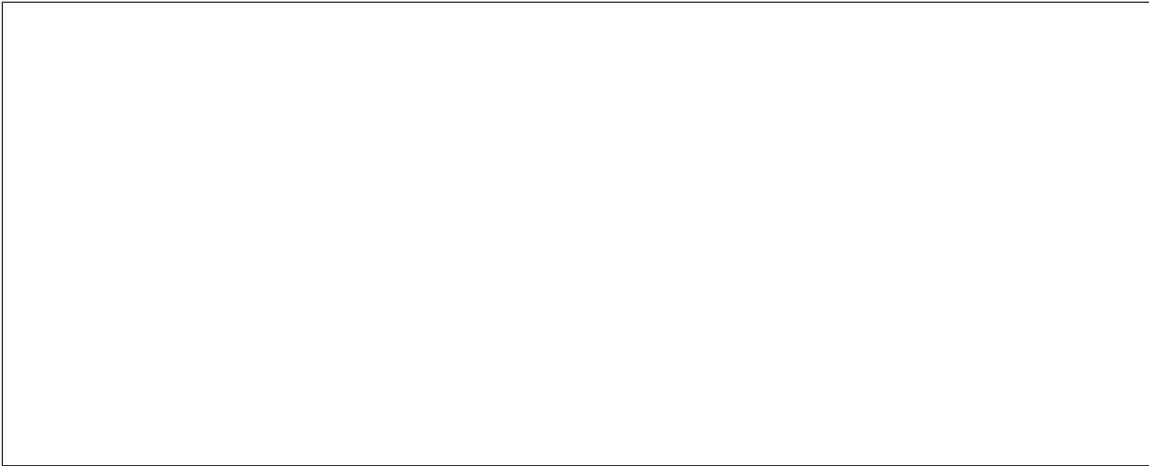
ware, for example: `python-dracclient`, `sushy`.

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.

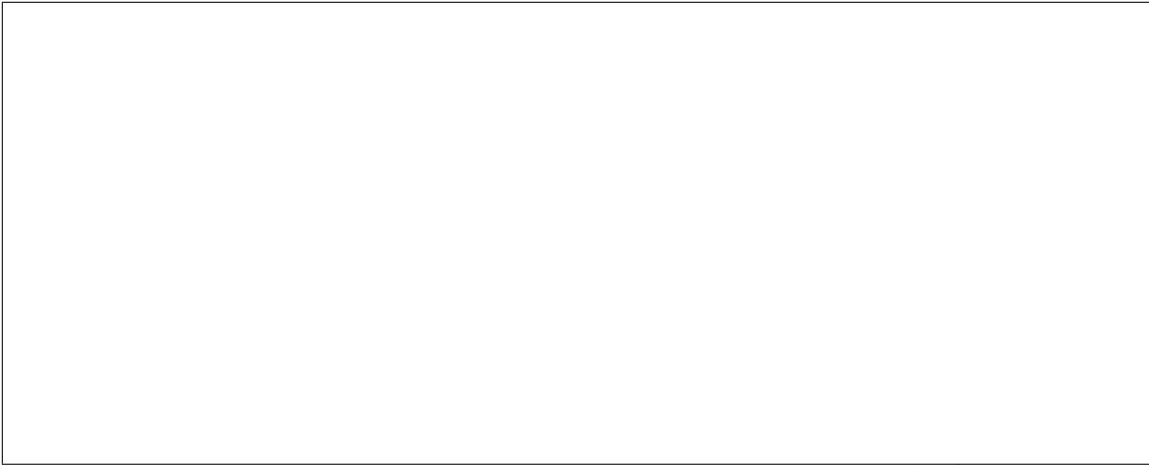


(continues on next page)

(continued from previous page)



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)

• Im-  
ple-  
men  
a  
meth  
nam  
app  
This  
meth  
need  
to  
use  
the  
clea  
dec-  
o-  
ra-  
tor.  
It

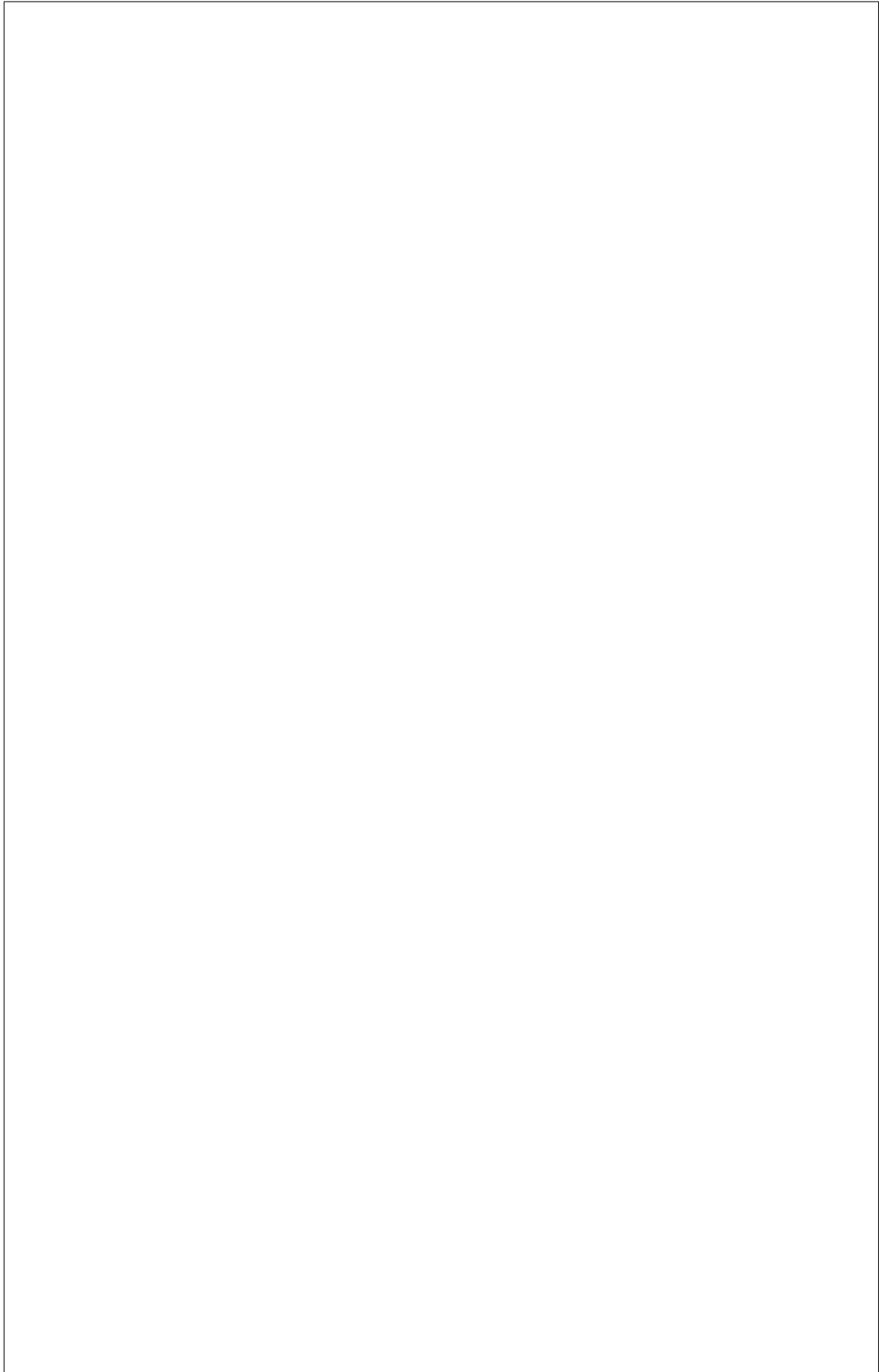


(continued from previous page)

↪BIOS settings to be applied'

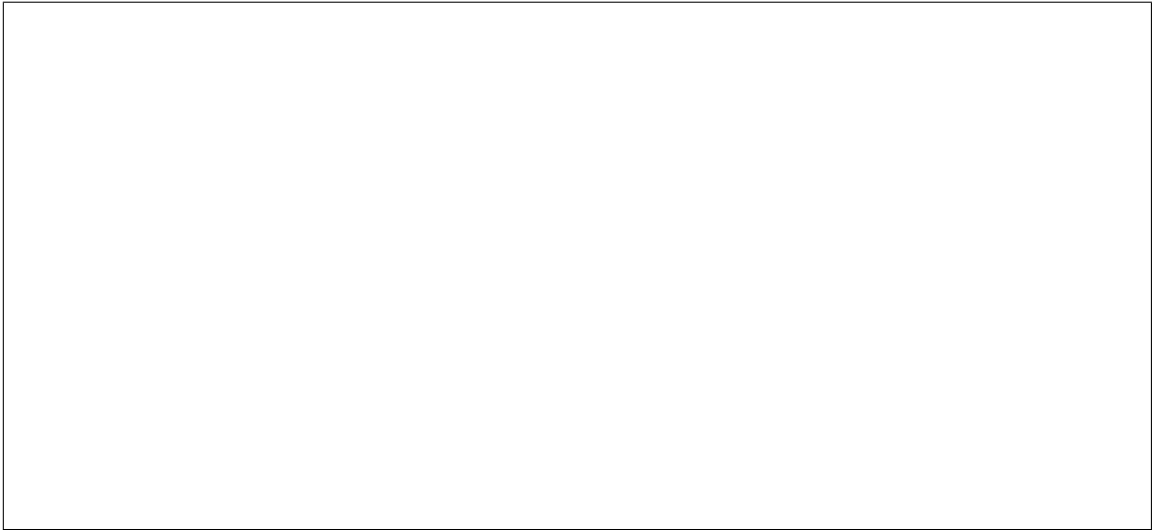
(continues on next page)

(continued from previous page)



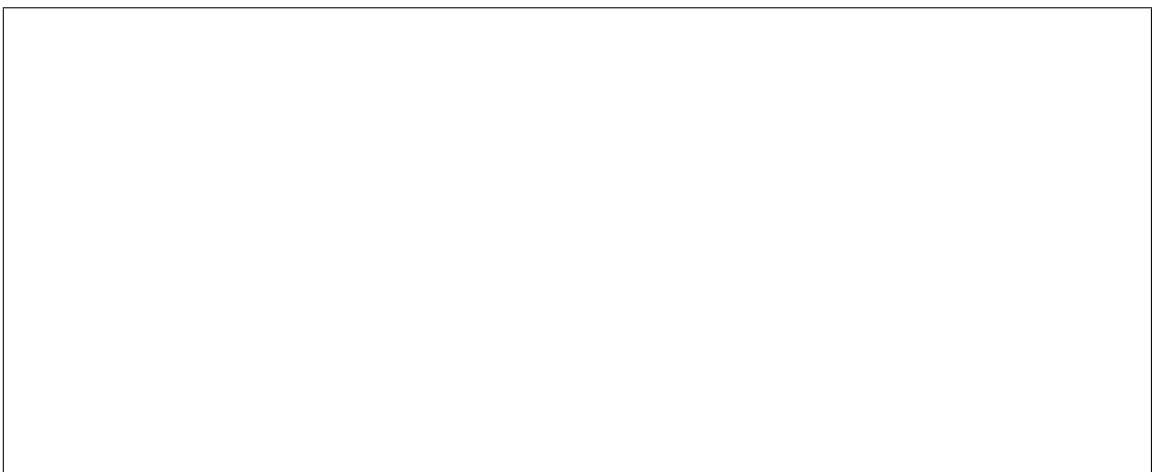
(continues on next page)

(continued from previous page)



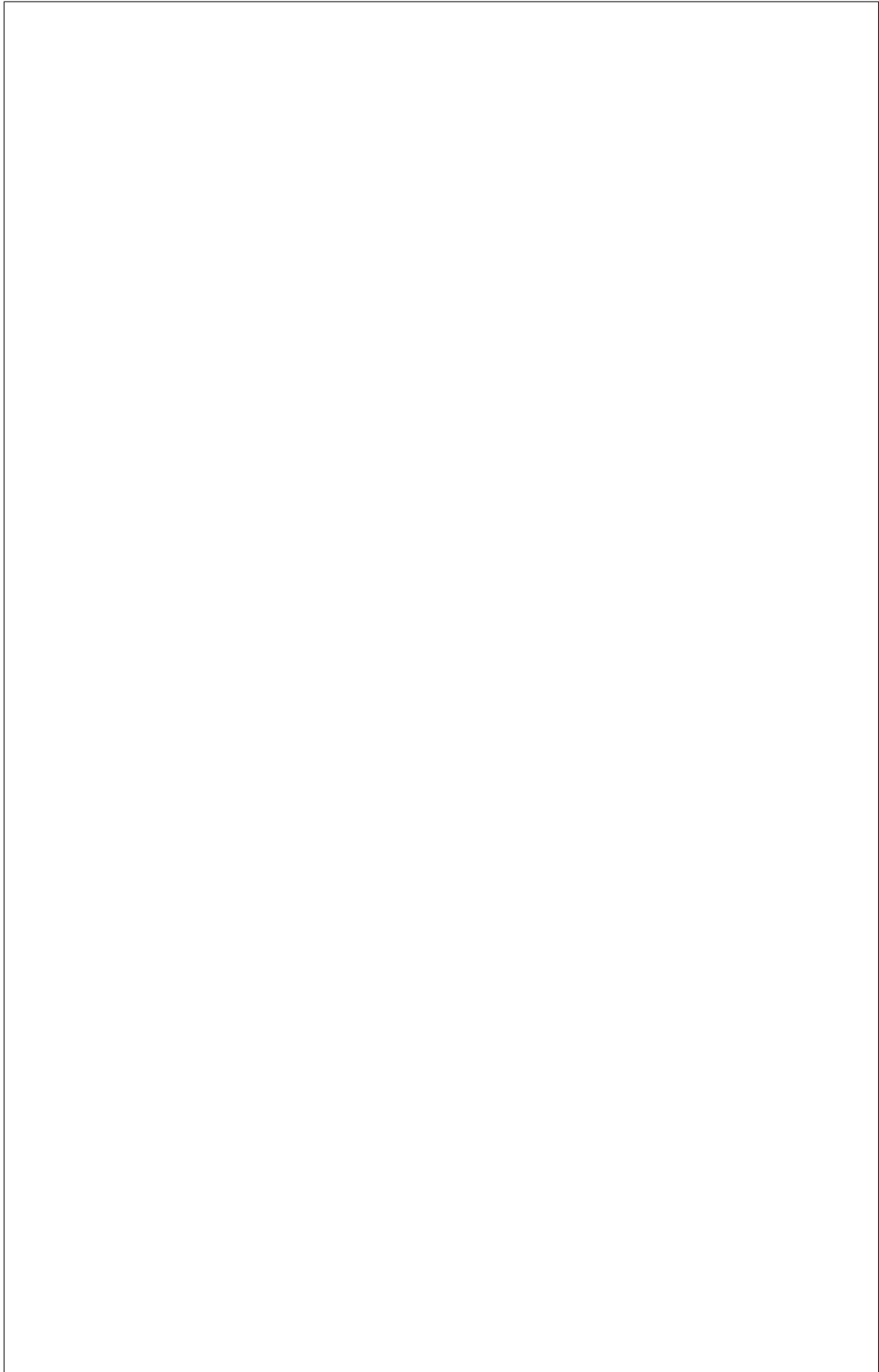
The  
set  
pa-  
ram-  
e-  
ter  
is  
a  
list  
of  
BIO  
set-  
ting  
to  
be  
con-  
fig-  
ured  
for  
ex-

ample:



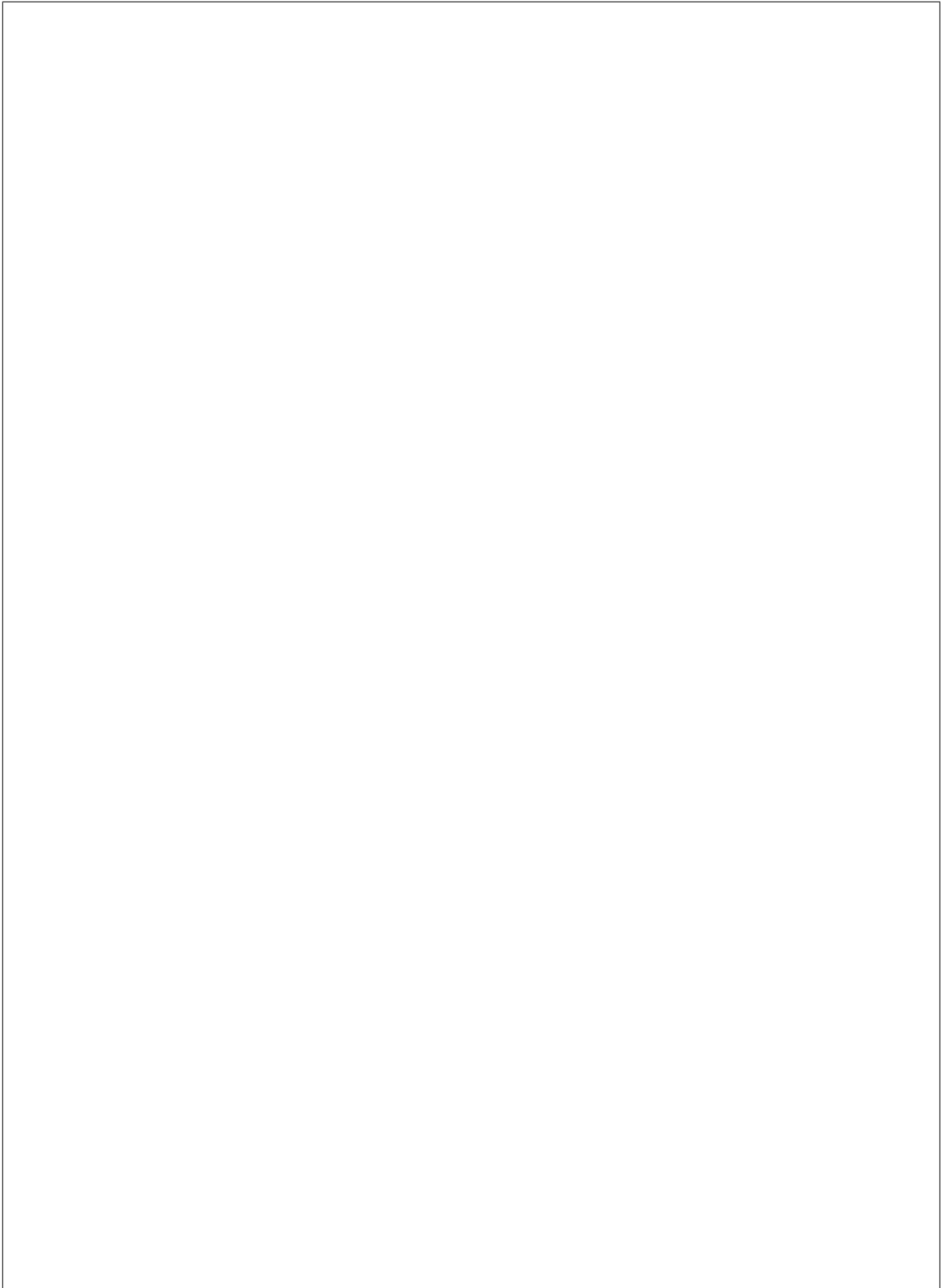
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



## **Third Party Continuous Integration**

**Note:** This document is a work-in-progress. Unfilled sections will be worked in follow-up patchsets. This version is to get a basic outline and index done so that we can then build on it. (krtaylor)

up their continuous integration test systems.

### **CI Architecture Overview**

### **Requirements Cookbook**

### **Sizing**

### **Infrastructure**

This  
doc-  
u-  
men-  
pro-  
vide  
tips  
and  
guid  
lines  
for  
thir  
part  
driv  
de-  
vel-  
op-  
ers  
set-  
ting

This  
sec-  
tion  
de-  
scrib  
wha  
char  
you  
need  
to  
mak  
to  
a

ironic job.

**jenkins changes**

**nodepool changes**

**neutron changes**

**pre-test hook**

**cleanup hook**

**IroniC**

**Hardware Pool Management**

**Problem**

the problem of two jobs trying to use the name target arises. If you have one target machine and a maximum number of one jobs running on your ironic pipeline at a time, then you wont run into this problem. However, one target may not handle the load of ironics daily patch submissions.

## **Solutions**

### **Zuul v3**

### **Molten Iron**

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**

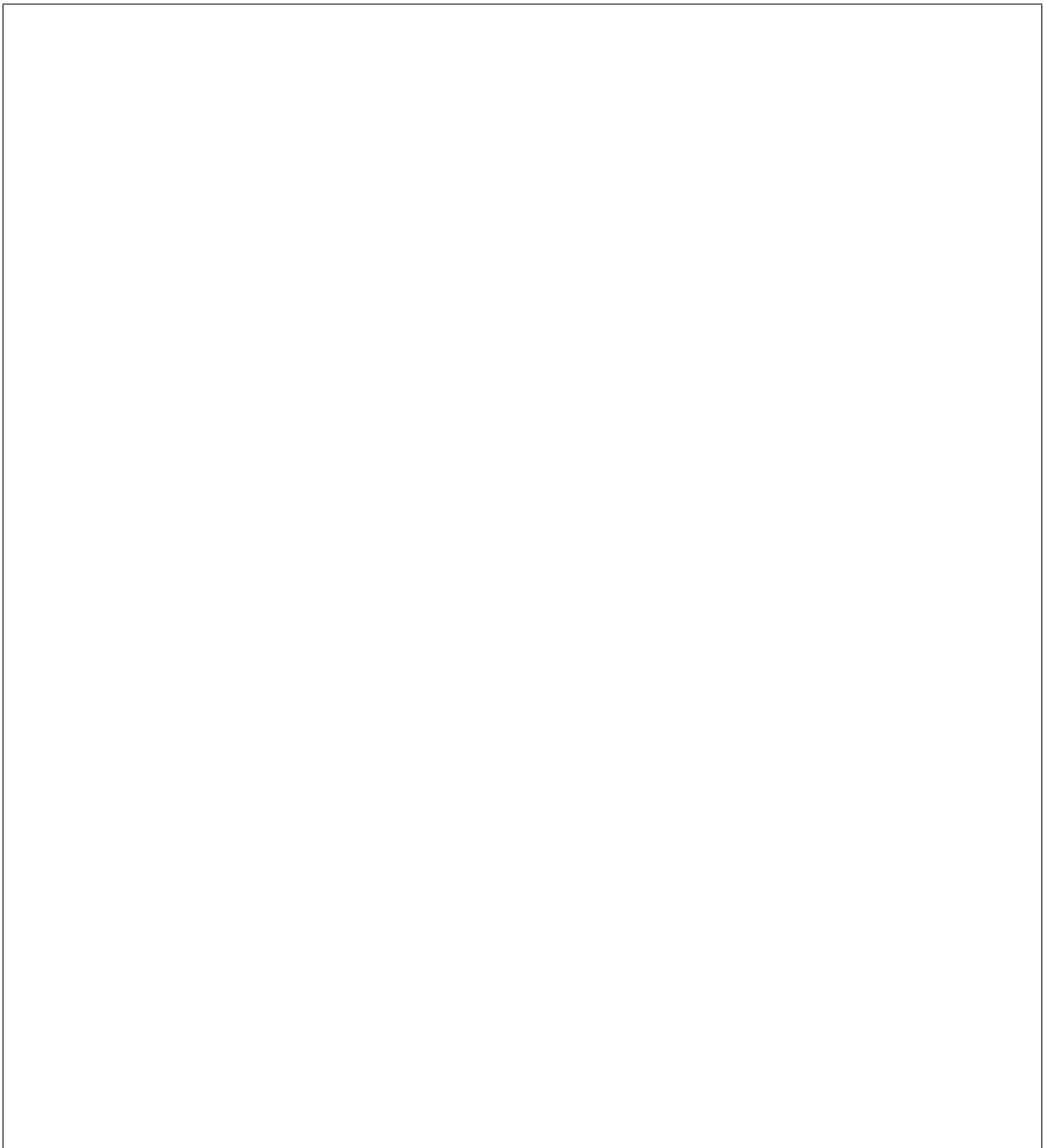
### **Developing a new Deploy Step**

mol  
iron  
is  
a  
tool  
that  
al-  
lows  
you  
to  
re-  
serv  
hard  
ware  
from  
a  
pool  
at  
the  
last

To  
sup-  
port  
cus-  
tomi  
de-  
ploy  
men  
step



class and use the decorator `deploy_step` defined in `ironic/drivers/base.py`. For example, we will implement a `do_nothing` deploy step in the `AgentDeploy` class.



(continues on next page)

(continued from previous page)

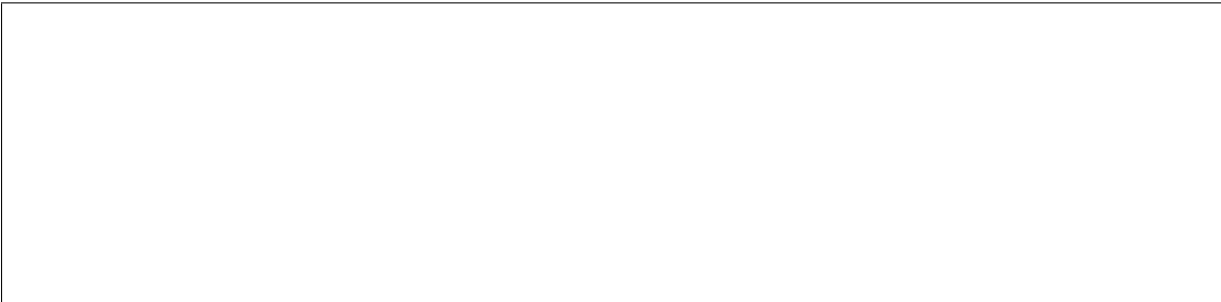
```

↪test argument."

```

(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)

--

--

(continues on next page)

---

(continued from previous page)

(continues on next page)

(continued from previous page)

↪

↪"This is a test argument."

(continues on next page)

---

(continued from previous page)

(continues on next page)

---

(continued from previous page)

1932



(continued from previous page)

**Note:** Similarly, clean steps can be implemented using the `clean_step` decorator.

mented in a custom [IPA hardware manager](#). All in-band deploy steps must have priorities between 41 and 99, see [Agent steps](#) for details.

vices, support has been added to [devstack](#) to mimic an external physical switch. Here we include a recommended configuration for devstack to bring up this environment.

## **IroniC multitenant networking and DevStack**

## Using VMs as baremetal servers

baremetal servers and `ML2 networking-generic-switch` that interacts with OVS.

### DevStack Configuration

tered in `ironic. networking-generic-switch` driver will be installed and configured in Neutron.

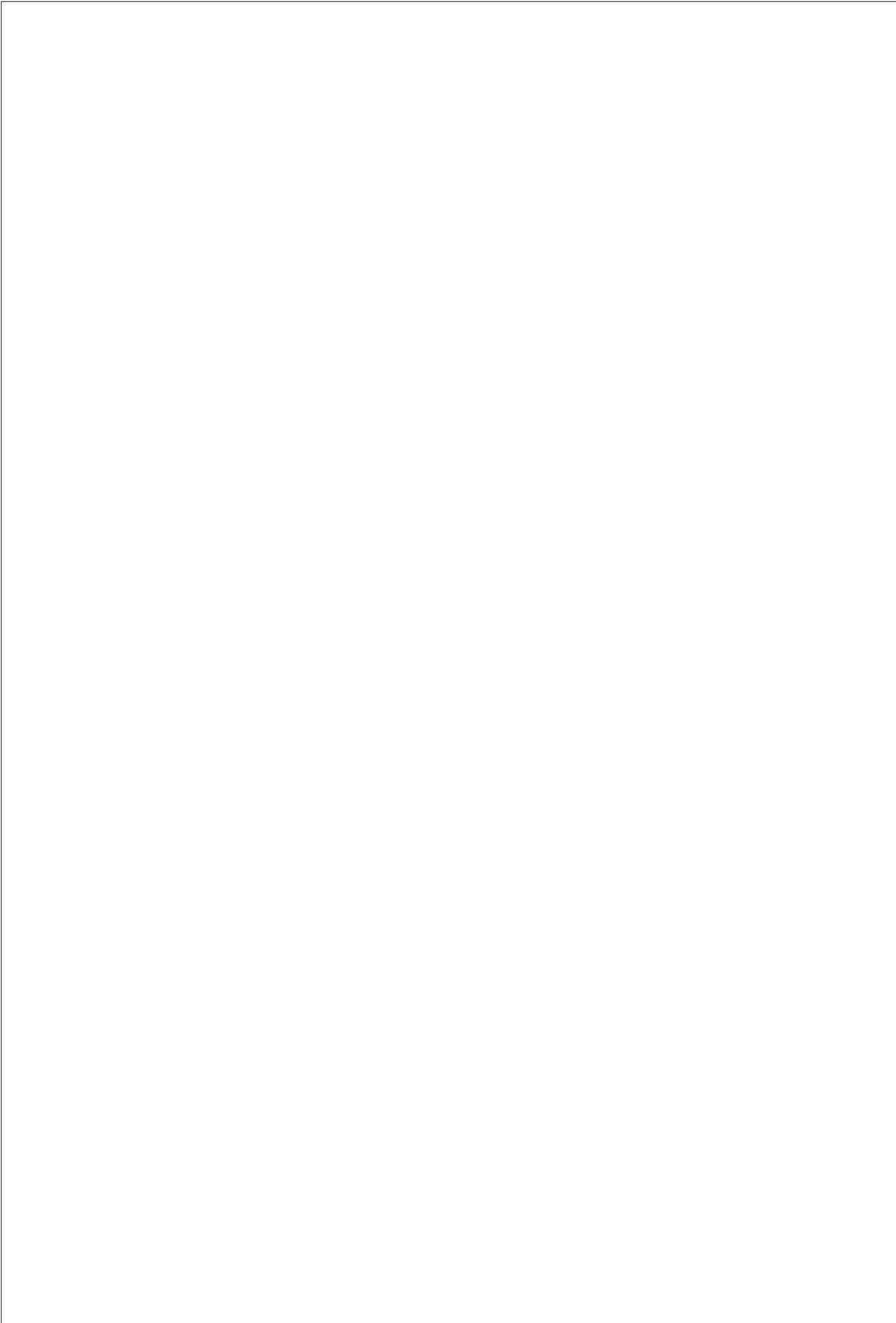
(continues on next page)

This  
sce-  
nari  
show  
how  
to  
setu  
De-  
vs-  
tack  
to  
use  
Iron  
in-  
te-  
gra-  
tion  
with  
VM  
as

The  
fol-  
low-  
ing  
is  
loc  
con  
that  
will  
setu  
De-  
vs-  
tack  
with  
3  
VM  
that  
are  
reg-  
is-

[ [1

(continued from previous page)



(continues on next page)

↪ #  
 ↪ C  
 ↪ i  
 ↪ f  
 ↪ i  
 ↪ d  
 ↪ p  
 ↪  
 ena  
 ↪ p  
 ↪ i  
 ↪ h  
 ↪ /  
 ↪ o  
 ↪ o  
 ↪ o  
 ↪ i  
  
 ↪ #  
 ↪ 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  $\rightarrow$  U  
 $\rightarrow$  L  
 $\rightarrow$  L  
 IRO  $\rightarrow$  E  
 $\rightarrow$  N  
 $\rightarrow$  I  
 $\rightarrow$  n  
 IRO  $\rightarrow$  N  
 $\rightarrow$  I  
 $\rightarrow$  #  
 $\rightarrow$  c  
 OVS.  $\rightarrow$  P  
 $\rightarrow$  B  
 PHY  $\rightarrow$  N  
 IRO  $\rightarrow$  P  
 $\rightarrow$  N  
 $\rightarrow$  N  
 $\rightarrow$  p  
 IRO  $\rightarrow$  P  
 $\rightarrow$  S  
 $\rightarrow$  P  
 $\rightarrow$  0  
 $\rightarrow$  5  
 $\rightarrow$  0  
 $\rightarrow$  2  
 IRO  $\rightarrow$  S  
 $\rightarrow$  S  
 $\rightarrow$  G  
 $\rightarrow$  0  
 $\rightarrow$  5  
 $\rightarrow$  1  
 Q\_  $\rightarrow$  P  
 ENA  $\rightarrow$  T  
 $\rightarrow$  V  
 Q\_  $\rightarrow$  M  
 $\rightarrow$  T  
 $\rightarrow$  N  
 $\rightarrow$  T  
 TEN.  $\rightarrow$  V  
 $\rightarrow$  R

(continues on next page)

---

(continued from previous page)

↪ #  
 ↪ E  
 ↪ S  
 ↪ f  
 ↪ t  
 ↪ d  
 ↪ d  
 ↪ i  
 ↪  
 ena  
 ↪ s  
 ↪ s  
 ↪ p  
 ena  
 ↪ s  
 ↪ s  
 ↪ o  
 ena  
 ↪ s  
 ↪ s  
 ↪ c  
 ena  
 ↪ s  
 ↪ s  
 ↪ a

dis

↪ #  
 ↪ D  
 ↪ C  
 dis  
 ↪ s  
 ↪ C  
 ↪ C  
 ↪ s  
 ↪ C  
 ↪ a  
 ↪ C  
 ↪ v

dis

(continues on next page)

(continued from previous page)



(continues on next page)

→ #  
→ 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  
  
→ #  
→ E  
→ a  
→ h  
→ t  
→ \_  
→ i  
→ n  
→  
  
→ #  
→ E  
→ H  
→ T  
→ f  
→ h





(continued from previous page)

#  
 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  
 IRO  
 V  
 S  
 D

(continues on next page)

---

(continued from previous page)

→ #  
→ S  
→ o  
→ t  
→ e  
→ p  
→ i  
→ G  
→ \_  
→ U  
→ 0  
→ f  
→ n  
→ e  
→ p  
→  
IRO  
→ V  
→ E  
→ D  
→ #  
→ T  
→ b  
→ y  
→ o  
→ I  
→ r  
→ f  
→ s  
→ \_  
→ s  
→ t  
→ t  
→ T  
IRO  
→ B  
→ D  
→ R  
VIR  
→ D  
→ #  
→ B  
→ d  
→ \_  
→ C  
→ a  
→ 1  
→ 0  
→ 0  
→ 0  
→ 2  
→ n  
→ f  
→ i  
→

(continues on next page)

(continued from previous page)

```

#
I
t
o
w
t
h
n
_
y
m
a
w
t
#
f
NET
G
1
0
1
FIX
R
1
0
0
2
FIX
N
S
#
L
a
C
t
f
LOG
$
d
l
LOG
$
l
IRO
V
L
D
$
i
b
l

```

ing booting from Cinder volumes with VMs.

## IroniC Boot-from-Volume with DevStack

supported from the Pike release.

aged by cinder with VMs as baremetal servers.

## **DevStack Configuration**

tered in ironic. A volume connector with IQN is created for each node. These connectors can be used to connect volumes created by cinder. The detailed description for DevStack is at [Deploying IroniC with DevStack](#).



(continues on next page)

This  
sce-  
nari  
shov  
how  
to  
setu  
De-  
vS-  
tack  
to  
en-  
able  
node  
to  
boot  
from  
vol-  
ume  
man

The  
fol-  
low-  
ing  
is  
loc  
con  
that  
will  
setu  
De-  
vS-  
tack  
with  
3  
VM  
that  
are  
reg-  
is-

[ [1

(continues on next page)

```
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
ena      ↪ s
          ↪ q
          ↪
```





(continues on next page)

```
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
DEF  ↪ I
      ↪ T
```

(continued from previous page)

#  
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  
  
#  
E  
M  
I  
f  
  
#  
E  
P  
I  
f  
  
#  
T  
d  
d  
i  
i  
  
'  
'  
,

(continues on next page)

(continues on next 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


(continued from previous page)

IRON  
 $\hookrightarrow$  V  
 $\hookrightarrow$  S  
 $\hookrightarrow$  D

→ #  
→ S  
→ o  
→ t  
→ e  
→ p  
→ i  
→ G  
→  $\sqcup$   
→ U  
→ 0  
→ f  
→ n  
→ e  
→ p  
→

IRON

- V
- E
- D

→ #  
→ T  
→ b  
→ y  
→ o  
→ I  
→ r  
→ f  
→ s  
→   
→ s  
→ t  
→ t  
→ T

IRON

- B
- D
- R

VIR  
↪ D

$\hookrightarrow \#$   
 $\hookrightarrow B$   
 $\hookrightarrow d$   
 $\hookrightarrow \sqcup$   
 $\hookrightarrow D$   
 $\hookrightarrow c$   
 $\hookrightarrow a$   
 $\hookrightarrow 1$   
 $\hookrightarrow 0$   
 $\hookrightarrow 0$   
 $\hookrightarrow 0$   
 $\hookrightarrow 2$

```

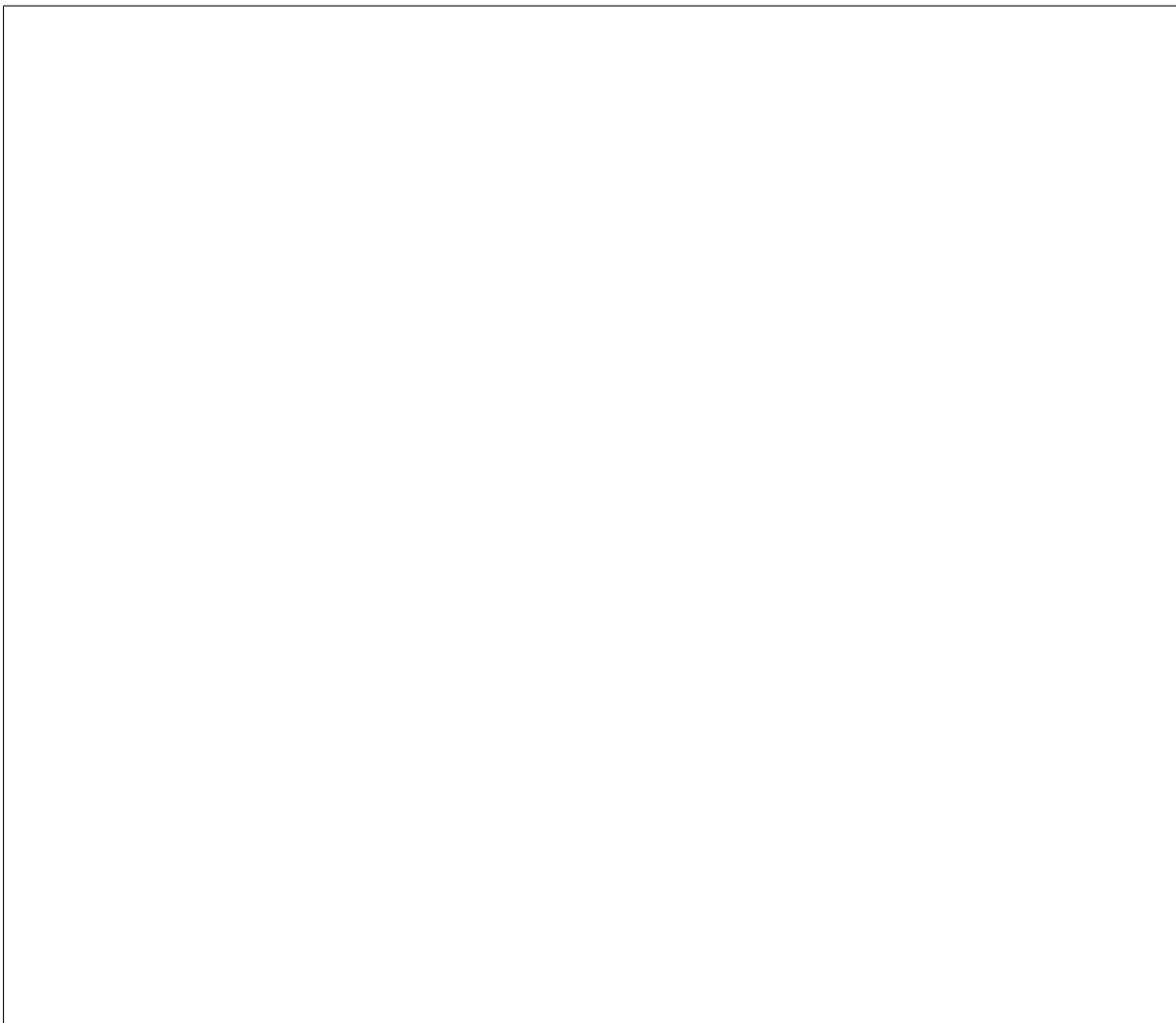
→ #
→ I
→ t
→ o
→ w
→ t
→ h
→ n
→ _
→ y
→ m
→ a
→ w
→ t

→ #
→ f
→
NET → G
→ 1
→ 0
→ 1
FIX → R
→ 1
→ 0
→ 0
→ 2
FIX → N
→ S

→ #
→ L
→ a
→ o
→ t
→ f
LOG → $
→ d
→ l
LOG → $
→ l
IRO → V
→ L
→ D
→ $
→ i
→ b
→ l

```

request an instance with the volume to nova:



(continues on next page)

Af-  
ter  
the  
en-  
vi-  
ron-  
men-  
is  
built  
you  
can  
cre-  
ate  
a  
vol-  
ume  
with  
cin-  
der  
and

.  
→  
→~  
→/  
→d  
→o

→#  
→q  
→t  
→i.  
→i  
→o  
→t  
→d  
→c  
→i.  
ima  
→\$  
→i.  
→s  
→  
→\$  
→I.  
→N.  
→-  
→f  
→v  
→-  
→c  
→i

(continued from previous page)

$\hookrightarrow \text{id}$ )

(continues on next page)

```

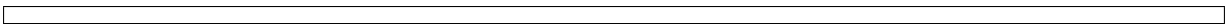
→#
→c
→k
ssh→k
ope→k
→c
→-
→-
→p
→k
→~
→/
→.
→s
→i
→r
→p
→d

→#
→c
→v
vol→$
→v
→c
→-
→-
→i
→$
→-
→-
→s
→l
→m
→v
→-
→f
→v
→-
→c

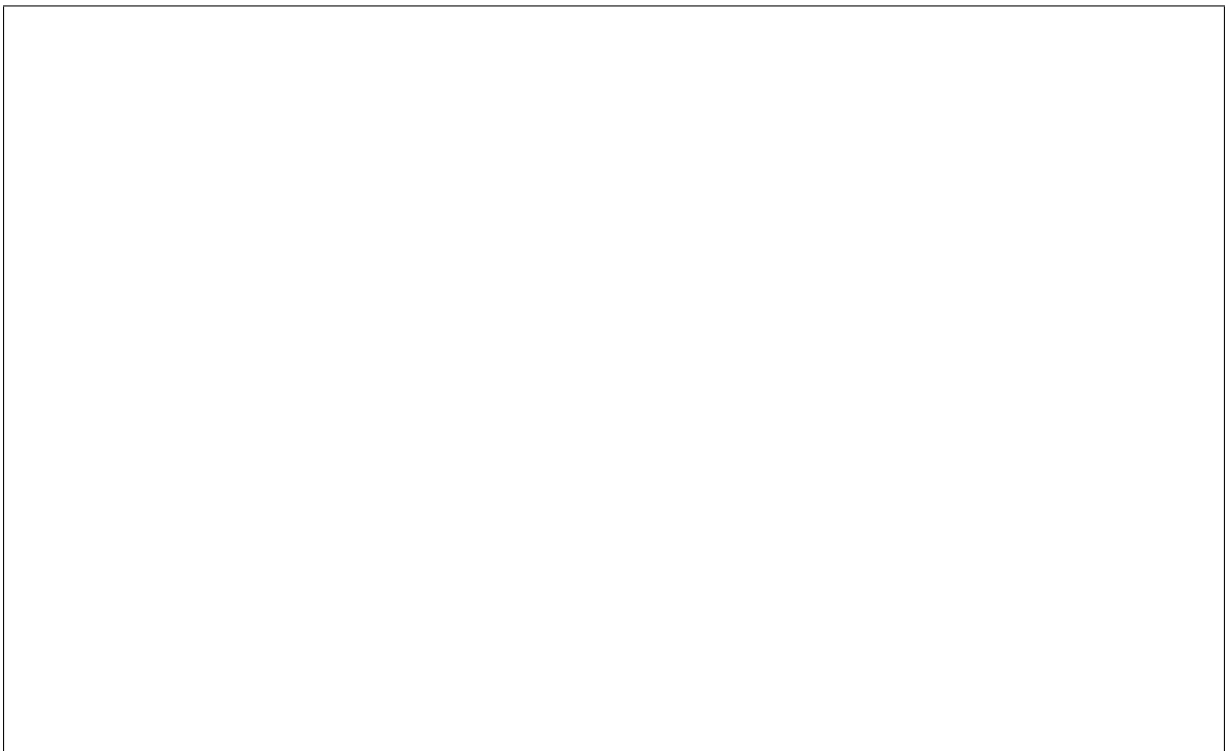
→#
→s
→i
ope→s
→c
→-
→-
→f
→b
→-
→-
→v
→

```

(continued from previous page)



volume with tempest in the environment:



You  
can  
also  
run  
an  
in-  
te-  
gra-  
tion  
test  
that  
an  
in-  
stan-  
is  
boot  
from  
a  
re-  
mote

cd  
→/  
→o  
→s  
→t  
tox  
→-  
→e  
→a  
→p  
→-  
→-  
→  
→i  
→t  
→p  
→t  
→s  
→s  
→t  
→b  
→b  
→f  
→v

Plea  
note  
that  
the  
stor-



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

Parameter

all
  
UUI
  
or
  
log-
  
i-
  
cal
  
nam
  
of
  
an
  
al-
  
lo-
  
ca-
  
tion

get\_all

Re-
  
triev
  
a
  
list
  
of
  
al-
  
lo-
  
ca-
  
tion

Parameters

- **node**  
The name of the node to get only the al-  
location for that node
- **resource**  
The filter by re-  
ques re-  
sour class
- **state**  
The filter by al-  
lo-  
ca-  
tion state
- **margin**  
The pag-  
i-  
na-  
tion mar-  
for large

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.

returned.

- **file**  
Op-  
tion:  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field  
of  
the  
re-  
sour  
to  
be

- **own**  
Fil-  
ter  
by  
own

**get\_one**  
Re-  
triev  
in-  
for-  
ma-  
tion  
about  
the  
give  
al-  
lo-  
ca-  
tion.

**Parame**

- **all**  
UUI  
or  
log-  
i-

returned.

cal  
 nam  
 of  
 an  
 al-  
 lo-  
 ca-  
 tion.

•

**file**  
 Op-  
 tion  
 a  
 list  
 with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

**invalid**

**patch** (*a*

Up-  
 date  
 an  
 ex-  
 ist-  
 ing  
 al-  
 lo-  
 ca-  
 tion.

**Param**

•

**all**  
 UUI  
 or  
 log-

i-  
cal  
nam  
of  
an  
al-  
lo-  
ca-  
tion.

- pat**  
a  
json  
PAT  
doc-  
u-  
men  
to  
ap-  
ply  
to  
this  
al-  
lo-  
ca-  
tion.

**post** (*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-  
 ting

**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.

Buil  
 a  
 dict  
 con-  
 tain-  
 ing  
 a  
 bios  
 set-  
 ting  
 valu

ironic.api.controllers.v1.chassis module

**class** `ironic.api.controllers.v1.chassis.ChassisController`

Base class for the chassis controller.

**delete**

Delete a chassis.

**Parameters**

**chassis\_id**

UUID of a chassis.

**detail**

Retrieve a list of chassis details.

**Parameters**

- marker**

pagination marker for

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.

large  
data  
sets.

- **lim**  
max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.  
This

- **sor**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.

- **sor**  
di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc  
De-  
fault  
asc.

get\_all

Re-  
triev  
a  
list  
of  
chas  
sis.

Paramet

•  
**max**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

•  
**lim**  
max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.  
This

value cannot be larger than the value of max\_limit in the [api] section of the ironic configuration, or only max\_limit resources will be returned.

•  
**sor**  
col-  
umn  
to  
sort

returned.

re-  
 sults  
 by.  
 De-  
 fault  
 id.

•

**sort**  
 di-  
 rec-  
 tion  
 to  
 sort.  
 asc  
 or  
 desc  
 De-  
 fault  
 asc.

•

**fields**  
 Op-  
 tion  
 a  
 list  
 with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

**get\_one**  
 Re-  
 triev  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 give

returned.

chas  
sis.

Paramete

- **cha**  
UUI  
of  
a  
chas  
sis.
- **fi**  
Op-  
tiona  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field  
of  
the  
re-  
sour  
to  
be

invalid

**nodes =**  
Ex-  
pose  
node  
as  
a  
sub-  
elem  
of  
chas  
sis

**patch** (**c**  
Up-  
date

an  
ex-  
ist-  
ing  
chas  
sis.

#### Parame

- **cha**  
UUI  
of  
a  
chas  
sis.
- **pat**  
a  
json  
PAT  
doc-  
u-  
men  
to  
ap-  
ply  
to  
this  
chas  
sis.

**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**

ironic.

Re-  
turn  
a  
link  
to  
the  
next  
sub-  
set  
of  
the  
col-  
lec-  
tion.

ironic.  
Re-  
turn  
the  
col-  
lec-  
tion  
has  
more  
item

ironic.

Buil  
a



col-  
lec-  
tion  
dict  
in-  
clud  
ing  
the  
next  
link  
for  
pag-  
ing  
sup-  
port

Paramet

- **ite**  
 List  
 of  
 un-  
 san-  
 i-  
 tized  
 item  
 to  
 in-  
 clud  
 in  
 the  
 col-  
 lec-  
 tion
- **ite**  
 Nam  
 of  
 dict  
 key  
 for  
 item  
 valu
- **lim**  
 Pag-  
 ing  
 limi
-

**url**  
 Base  
 URI  
 for  
 build  
 ing  
 next  
 link

- **file**  
 Op-  
 tion  
 field  
 to  
 use  
 for  
 san-  
 i-  
 tize  
 func-  
 tion

- **san**  
 Op-  
 tion  
 san-  
 i-  
 tize  
 func-  
 tion  
 run  
 on  
 each  
 item  
 item  
 char  
 will  
 be  
 done  
 in-  
 plac

- **key**  
 Key  
 nam  
 for  
 build  
 ing  
 next  
 URI

•

**kwargs**  
 other  
 ar-  
 gu-  
 men-  
 pass  
 to  
 get

#### Returns

A  
 dict  
 con-  
 tain-  
 ing  
 ite  
 and  
 nex  
 val-  
 ues

### ironic.api.controllers.v1.conductor module

#### class i

Base  
 pec  
 res  
 Res  
 RES  
 con-  
 troll  
 for  
 con-  
 duc-  
 tors.

#### get\_all

Re-  
 triev  
 a  
 list  
 of  
 con-  
 duc-  
 tors.

#### Parame

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.

- **mar**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.
- **lim**  
max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.  
This
- **sort**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.
- **sort**  
di-  
rec-  
tion

returned.

tail.

to  
 sort.  
 asc  
 or  
 desc  
 De-  
 fault  
 asc.  
 •  
**file**  
 Op-  
 tion  
 a  
 list  
 with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be  
 •  
**det**  
 Op-  
 tion  
 bool  
 to  
 in-  
 di-  
 cate  
 whe  
 re-  
 triev  
 a  
 list  
 of  
 con-  
 duc-  
 tors  
 with  
 de-

returned.

get\_one  
 Re-  
 triev  
 in-  
 for-  
 ma  
 tion  
 abou  
 the  
 give  
 con-  
 duc-  
 tor.

Paramete

- **hos**  
 host  
 nam  
 of  
 a  
 con-  
 duc-  
 tor.
- **fie**  
 Op-  
 tion  
 a  
 list  
 with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

invalid

ironic.

ironic.

## ironic.api.controllers.v1.deploy\_template module

**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

- **max**
  
 pag-  
 i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets.

- **lim**
  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to  
 re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult.  
 This

value cannot be larger than the value of max\_limit in the [api] section of the ironic configuration, or only max\_limit resources will be returned.

- **sor**
  
 col-  
 umn



returned.

to  
sort  
re-  
sults  
by.  
De-  
fault  
id.

- **sort**  
di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc  
De-  
fault  
asc.
- **file**  
Op-  
tion:  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field  
of  
the  
re-  
sour  
to  
be
- **det**  
Op-  
tion:  
bool  
to  
in-  
di-

detail.

cate  
 whe  
 re-  
 triev  
 a  
 list  
 of  
 de-  
 ploy  
 tem-  
 plate  
 with  
  
**get\_one**  
 Re-  
 triev  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 give  
 de-  
 ploy  
 tem-  
 plate

Param

- **tem**  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 de-  
 ploy  
 tem-  
 plate
- **fi**  
 Op-  
 tion:  
 a  
 list

returned.

with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

invalid

patch (*t*  
 Up-  
 date  
 an  
 ex-  
 ist-  
 ing  
 de-  
 ploy  
 tem-  
 plate

Paramete

- **tem**  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 de-  
 ploy  
 tem-  
 plate
- **pat**  
 a  
 json  
 PAT

doc-  
 u-  
 men  
 to  
 ap-  
 ply  
 to  
 this  
 de-  
 ploy  
 tem-  
 plate

**post** (*templ*  
 Cre-  
 ate  
 a  
 new  
 de-  
 ploy  
 tem-  
 plate

**Paramete**  
**tem**  
 a  
 de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 re-  
 ques  
 body

ironic.

ironic.

Add  
 link  
 to  
 the  
 de-  
 ploy  
 tem-  
 plate

ironic.

Ar-  
gu-  
men-  
val-  
ida-  
tor  
to  
chec  
tem-  
plate  
for  
du-  
pli-  
cate  
step

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

**file**  
 (list of strings)  
 list of fields to preserve or Non to pre-serv then all

ironic.api.controllers.v1.driver module

**class** i  
 Base  
 peo  
 res  
 Res  
 RES  
 con-  
 troll  
 for  
 drive  
 pass  
 This  
 con-  
 troll  
 al-  
 low  
 ven-  
 dors  
 to  
 ex-  
 pose  
 cross  
 node  
 func  
 tion-  
 al-  
 ity  
 in  
 the

Ironi
 c will merely relay the message from here to the specified driver, no introspection will be made in the message body.

Ironi
 c API

**methods**  
 Re-  
 triev  
 in-  
 for-  
 ma-  
 tion  
 abou  
 ven-  
 dor  
 meth  
 ods  
 of  
 the  
 give  
 driv

**Parame**  
**dri**  
 nam  
 of  
 the  
 driv

**Returns**  
 dic-  
 tio-  
 nary  
 with  
 <ver  
 dor  
 meth  
 nam  
 meta  
 data  
 en-  
 tries

**Raises**  
 Driv  
 Not-  
 Four  
 if  
 the  
 driv  
 nam  
 is  
 in-

valid  
 or  
 the  
 drive  
 can-  
 not  
 be  
 load

**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



cal disks and a textual description for them.

be  
 men  
 tion  
 for  
 log-  
 i-

**Raises**

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 drive  
 does  
 sup-  
 port  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.

**Raises**

No-  
 tAc-  
 cept  
 able  
 if  
 re-  
 ques  
 ver-  
 sion  
 of  
 the  
 API  
 is  
 less  
 than  
 1.12

**Raises**

Driv  
 Not-  
 Four  
 if  
 drive

is  
 not  
 load  
 on  
 any  
 of  
 the  
 con-  
 duc-  
 tors.

**class** i

Base  
 pec  
 res  
 Res  
  
 RES  
 con-  
 troll  
 for  
 Driv

**get\_all**

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

**Parameters**

**drive**  
 name  
 of  
 the  
 drive

**Returns**

dictionary  
 with  
 <propert  
 name  
 description  
 entries

**Raises**

DriveNotFoun  
 (HTTPError, 404)  
 if  
 the  
 drive  
 name  
 is  
 invalid  
 or  
 the  
 drive  
 cannot  
 be  
 loaded

**raid =**

Example  
 RAID  
 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  
host  
nam  
drive  
is  
ac-  
tive  
on.
- **det**  
bool  
whe  
to  
in-  
clud  
de-  
taile  
info  
such  
as

faces fields.

the  
type  
field  
and  
de-  
fault  
in-  
ter-

- **int**  
op-  
tiona  
list  
of  
dicts  
of  
hard  
ware  
in-  
ter-  
face  
info

**Returns**  
dict  
rep-  
re-  
sent  
ing  
the  
drive  
ob-  
ject.

ironic.  
This  
meth  
hide  
field  
that  
were  
adde  
in  
new  
API  
ver-  
sion  
  
Cer-  
tain  
field

when the requests API version matches or exceeds the versions when these fields were introduced.

ironic.

Con-  
vert  
drive  
and  
hard  
ware  
type  
to  
an  
API  
serial-  
ob-  
ject.

Paramet

- **har**  
dict  
map  
ping  
hard  
ware  
type  
nam  
to  
con-  
duc-  
tor  
host

faces fields.

`ironic.api.controllers.v1.event` module

nam

- **det**  
bool  
whe  
to  
in-  
clud  
de-  
taile  
info  
such  
as  
the  
type  
field  
and  
de-  
fault  
in-  
ter-

#### Returns

an  
API  
serial  
drive  
col-  
lec-  
tion  
ob-  
ject.

**class** i

Base  
pec  
res  
Res  
  
RES  
con-  
troll  
for  
Ever

**post** (*ev*

ironi
 c.api.controllers.v1.node module

ironi
 c.  
 Val-  
 ida-  
 tor  
 for  
 even

**class** i  
  
 Base  
 pec  
 res  
 Res

**get** (*nod*  
 Get  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 for  
 a  
 node

**Parame**  
**nod**  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

**Returns**  
  
 a  
 json  
 ob-  
 ject  
 con-  
 tain-  
 ing:



unknown.

**boot\_c**  
the  
boot  
de-  
vice  
one  
of  
*irc*  
*com*  
*boo*  
or  
Non  
if  
it  
is  
un-  
know

**persist**  
Whe  
the  
boot  
de-  
vice  
will  
per-  
sist  
to  
all  
fu-  
ture  
boot  
or  
not,  
Non  
if  
it  
is

**put** (*nod*  
Set  
the  
boot  
de-  
vice  
for  
a  
node  
Set  
the  
boot

de-  
 vice  
 to  
 use  
 on  
 next  
 re-  
 boot  
 of  
 the  
 node

Paramete

- **node**  
 the  
 UI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node
- **boot**  
 the  
 boot  
 de-  
 vice  
 one  
 of  
[irc](#)  
[com](#)  
[boo](#)
- **per**  
 Boo  
 valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist

not. Default: False.

to  
all  
fu-  
ture  
boot  
Fals  
if

#### 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  
 peo  
 res  
 Res

**get\_all**

Get  
 node  
 hard  
 ware  
 com  
 po-  
 nent  
 and  
 their  
 in-  
 di-  
 ca-  
 tors.

**Parame**

**nod**  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

**Returns**

A  
 json  
 ob-  
 ject  
 of  
 hard  
 ware  
 com  
 po-  
 nent  
 (*ir*  
*com*  
*com*  
 as  
 keys  
 with

tor IDs (from *get\_supported\_indicators*) as values.

in-  
di-  
ca-  
  
**get\_one**  
Get  
node  
hard  
ware  
com  
po-  
nent  
in-  
di-  
ca-  
tor  
and  
its  
state

**Parame**

- **node**  
the  
UUID  
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** (*node*  
 Set  
 node  
 hard  
 ware  
 com  
 po-  
 nent  
 in-  
 di-  
 ca-  
 tor  
 to  
 the  
 de-  
 sirec  
 state

Parame

- **node**  
 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-  
di-  
ately

**Parame**  
**nod**  
the  
UUI  
or

log-  
i-  
cal  
nam  
of  
a  
node

Raises

Not-  
Four  
if  
re-  
ques  
ver-  
sion  
of  
the  
API  
does  
sup-  
port  
in-  
ject  
nmi.

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



agement.inject\_nmi.

Nod  
Loc  
if  
the  
node  
is  
lock  
by  
an-  
othe  
con-  
duc-  
tor.

**Raises**

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the  
node  
drive  
does  
sup-  
port  
man  
age-  
men  
or  
man

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-

valid boot device is specified.

fied  
 or  
 an  
 in-

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 miss  
 ing  
 sup-  
 plied  
 info

**class** i  
 Base  
 pec  
 res  
 Res

**get** (*nod*  
 Get  
 con-  
 nec-  
 tion  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 con-  
 sole

**Parame**  
**nod**  
 UI  
 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-  
 nanc  
 mod

Parame

nod  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

put (*nod*

Put  
 the  
 node  
 in  
 main  
 te-  
 nanc  
 mod

Parame

- nod  
 the  
 UUI  
 or  
 log-  
 i-  
 cal\_  
 of  
 a  
 node
- rea  
 Op-  
 tion  
 the  
 rea-  
 son  
 why  
 its  
 in

main  
te-  
nanc

**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  
of  
man  
age-  
men

**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  
 UUI  
 or  
 log-  
 i-  
 cal\_  
 of  
 a  
 node

**power** (*n*  
 Set  
 the  
 pow  
 state  
 of  
 the  
 node

**Parame**  
 •

**node**  
the  
UI  
or  
log-  
i-  
cal  
nam  
of  
a  
node

- **target**  
The  
de-  
sired  
pow  
state  
of  
the  
node

- **timeout**  
time  
out  
(in  
sec-  
onds  
pos-  
i-  
tive  
in-  
te-  
ger  
(>  
0)  
for  
any  
pow  
state  
Non

indicates to use default timeout.

**Raises**  
Clie  
ror  
(HT  
409)  
if  
a  
pow

is in CLEANING state.

op-  
er-  
a-  
tion  
is  
al-  
read  
in  
prog  
  
**Raises**  
 In-  
valid  
State  
(HT  
400)  
if  
the  
re-  
ques  
tar-  
get  
state  
is  
not  
valid  
or  
if  
the  
node  
  
**Raises**  
 No-  
tAc-  
cept  
able  
(HT  
406)  
for  
soft  
re-  
boot  
soft  
pow  
off  
or  
time  
out  
pa-  
ram-  
e-



ter, 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  
back  
grou  
task  
will  
be-

gin which actually applies the state change. This call will return a 202 (Accepted) indicating the request was accepted and is in progress; the client should continue to GET the status of this node to observe the

status of the requested action.

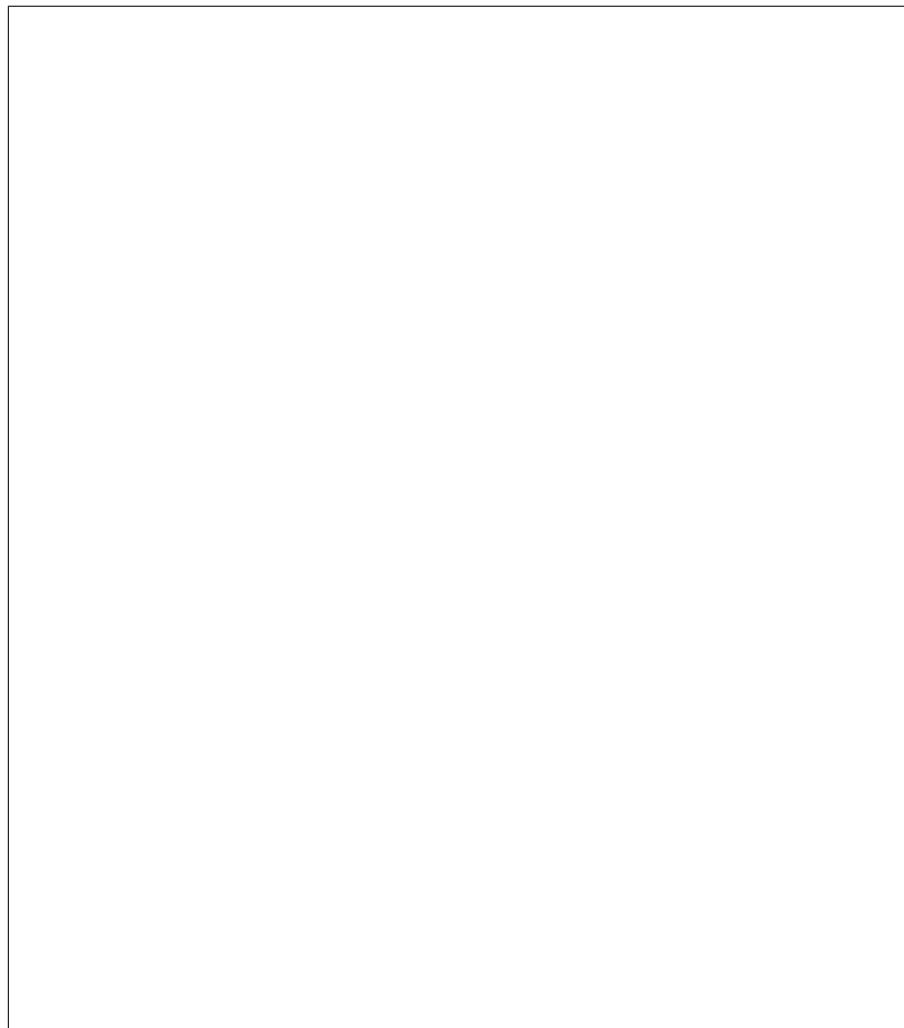
Parameters

- nodeId**  
 Unique identifier for the node, or log-ical name of a node.
- targetState**  
 The desired provision state of the node or verb.
- configdrive**  
 Optional. A gzip and base-encoded configuration drive or a dict to build a configuration.

figdrive from. Only valid when setting provision state to active or rebuild.

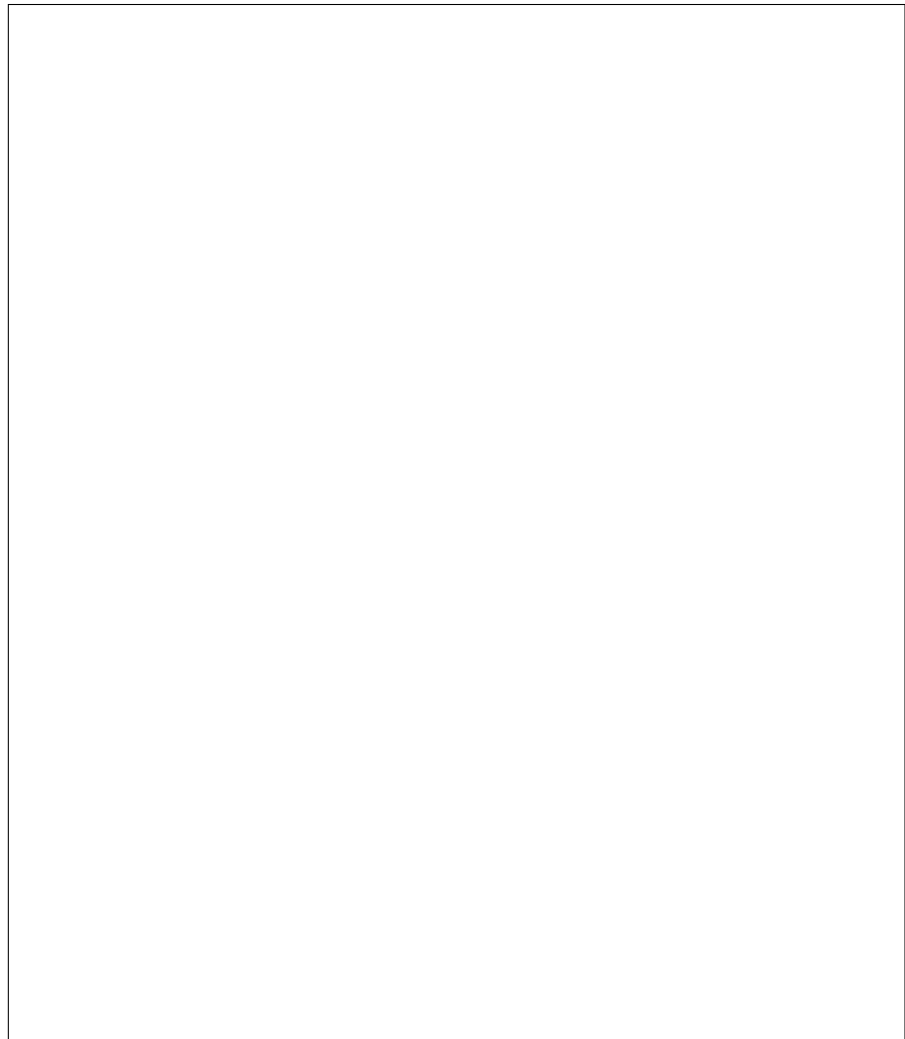
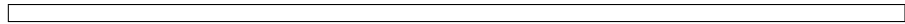
•  
cle  
An  
or-  
dere  
list  
of  
clea  
ing  
step  
that  
will  
be  
per-  
form  
on  
the  
node  
A  
clea

ing step is a dictionary with required keys interface and step, and optional key args. If specified, the value for args is a keyword variable argument dictionary that is passed to the cleaning step method.:



(continues on next page)

(continued from previous page)

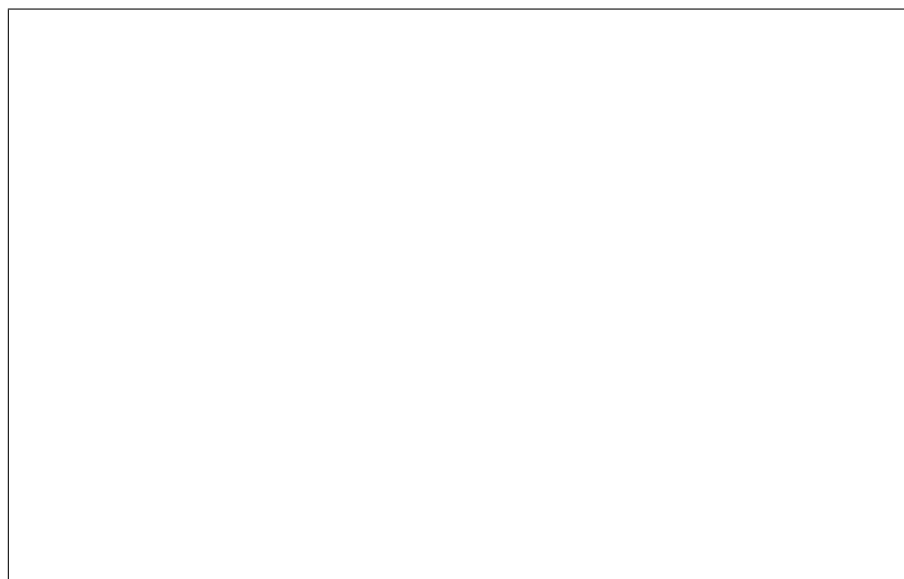


For  
ex-  
am-  
ple  
(this  
isnt  
a  
real  
ex-  
am-  
ple,  
this  
clear  
ing  
step  
does  
ex-  
ist):

This  
is  
re-  
quir  
(and  
only  
valid  
whe  
tar-  
get  
is  
clea

- **dep**  
A  
list  
of  
de-  
ploy  
step  
that  
will  
be  
per-  
form  
on  
the  
node  
A  
de-  
ploy  
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.:



(continues on next page)

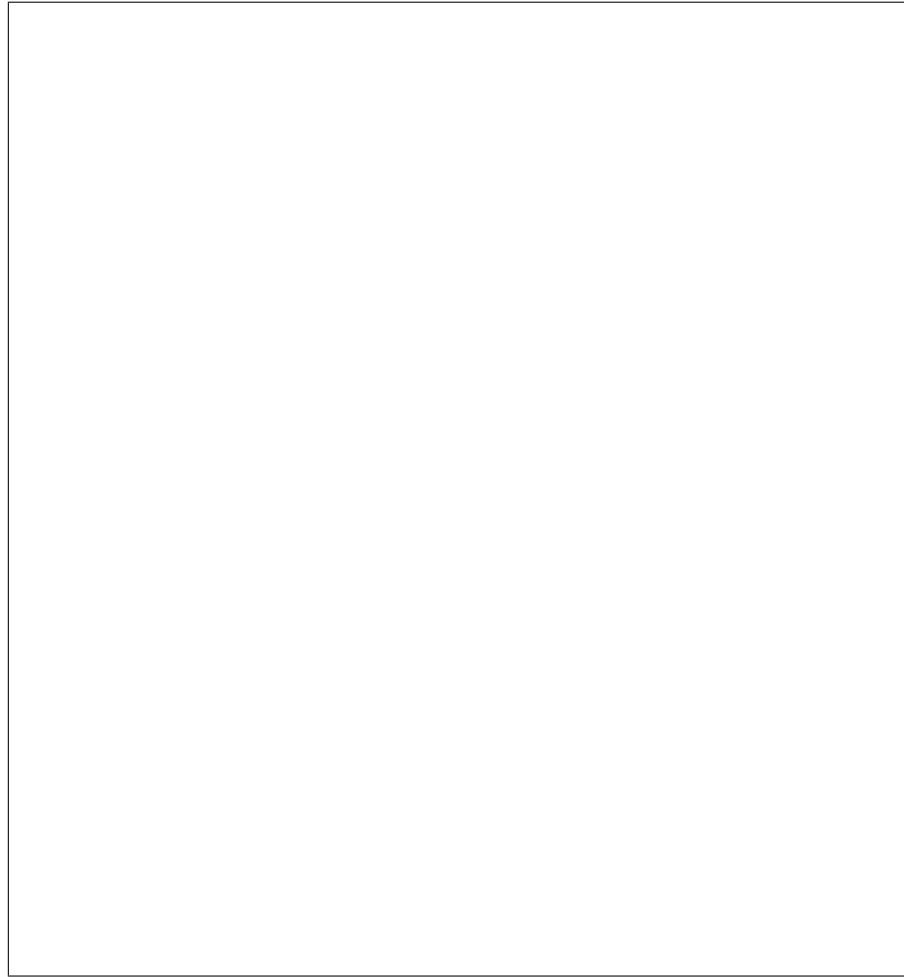
(continued from previous page)

```
↔: <valuen>}
```

For  
ex-  
am-  
ple  
(this  
isnt  
a  
real  
ex-  
am-  
ple,  
this  
de-  
ploy  
step  
does  
ex-  
ist):

(continues on next page)

(continued from previous page)



This  
is  
used  
only  
only  
when  
when  
target  
get  
is  
ac-  
tive  
or  
re-  
build  
and  
is  
op-  
tiona

- **res**  
A  
string  
rep-

vironment. This is required (and only valid), when target is rescue.

re-  
sent  
ing  
the  
pass  
wor  
to  
be  
set  
in-  
side  
the  
res-  
cue  
en-

Raises

Nod  
Loc  
(HT  
409)  
if  
the  
node  
is  
cur-  
rentl  
lock

Raises

Clie  
ror  
(HT  
409)  
if  
the  
node  
is  
al-  
read  
be-  
ing  
pro-  
vi-  
sion

Raises

In-  
valid  
Pa-  
ram-  
e-



power driver interface fails.

current state.

ter-  
Valu  
(HT  
400)  
if  
val  
i-  
da-  
tion  
of  
clea  
de-  
ploy  
or

#### Raises

In-  
valid  
State  
(HT  
400)  
if  
the  
re-  
ques  
tran-  
si-  
tion  
is  
not  
pos-  
si-  
ble  
from  
the

#### Raises

Nod  
Mai  
te-  
nanc  
(HT  
400)  
if  
op-  
er-  
a-  
tion  
can-  
not

node is in maintenance mode.

requested state transition.

be  
per-  
form  
be-  
caus  
the

**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  
al-  
low  
the

**raid** (*no*  
Set  
the  
tar-  
get  
raid  
con-  
fig

of  
 the  
 node

Param

- **nod**  
 the  
 UI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

- **tar**  
 De-  
 sirec  
 tar-  
 get  
 RA  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 of  
 the  
 node  
 It  
 may  
 be  
 an  
 emp

dictionary as well.

Raises

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 node

tion.

drive  
does  
sup-  
port  
RAI  
con-  
fig-  
u-  
ra-

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
val-  
i-  
da-  
tion  
of  
tar-  
get  
raid  
con-  
fig  
fails

Raises

No-  
tAc-  
cept  
able  
if  
re-  
ques  
ver-  
sion  
of  
the  
API  
is  
less  
than  
1.12

class i

Base

peo
 res
 Res

**delete**

Re-
 mov
 one
 or
 all
 trait
 from
 a
 node

**Parame**

**tra**
 Strin
 valu
 trait
 to
 re-
 mov
 from
 a
 node
 or
 Non
 If
 Non
 all
 trait
 are
 re-
 mov

**get\_all**

List
 node
 trait

**put** (*trai*

Add
 a
 trait
 to
 a
 node

**Parame**

•

If not None, adds this trait to the node.

trait. If not None, replaces the nodes traits with this list.

**traits**  
 String  
 value  
 trait  
 to  
 add  
 to  
 a  
 node  
 or  
 None  
 Mu-  
 tu-  
 ally  
 ex-  
 clu-  
 sive  
 with  
 traits

- **traits**  
 List  
 of  
 String  
 traits  
 to  
 set  
 for  
 a  
 node  
 or  
 None  
 Mu-  
 tu-  
 ally  
 ex-  
 clu-  
 sive  
 with

**class** i  
  
 Base  
 pec  
 res  
 Res

**delete**  
 De-

tach
 a
 VIF
 from
 this
 node

Paramete

**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

Paramete

**vif**
 a
 dic-
 tio-
 nary
 of
 in-
 for-
 ma-
 tion
 about
 a
 VIF.
 It
 mus
 have

whose value is a unique identifier for that VIF.

API. Ironic will merely relay the message from here to the appropriate driver, no introspection will be made in the message body.



meth
 ods
 of
 the
 give
 node

Param

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  
 Node

**delete**  
 Delete  
 a  
 node

**Parameter**  
**node**  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

**detail**

Re-  
 triev  
 a  
 list  
 of  
 node  
 with  
 de-  
 tail.

**Parameter**

- cha**  
 Op-  
 tion  
 UUI  
 of  
 a  
 chas  
 sis,  
 to  
 get  
 only  
 node  
 for  
 that

stance.

ated nodes. May be combined with other parameters.

chas  
sis.

- **ins**  
Op-  
tion:  
UUI  
of  
an  
in-  
stan-  
to  
find  
the  
node  
as-  
so-  
ci-  
ated  
with  
that  
in-

- **ass**  
Op-  
tion:  
bool  
whe  
to  
re-  
turn  
a  
list  
of  
as-  
so-  
ci-  
ated  
or  
unas-  
so-  
ci-

- **mai**  
Op-  
tion:  
bool  
valu

or not in maintenance mode (False).

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  
a  
sin-  
gle  
re-  
sult.  
This
- **sort**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.
- **sort**  
di-  
rec-  
tion  
to

sort.  
 asc  
 or  
 desc  
 De-  
 fault  
 asc.

- **dri**  
 Op-  
 tion:  
 strin  
 valu  
 to  
 get  
 only  
 node  
 us-  
 ing  
 that  
 drive

- **res**  
 Op-  
 tion:  
 strin  
 valu  
 to  
 get  
 only  
 node  
 with  
 that  
 re-  
 sour

- **fau**  
 Op-  
 tion:  
 strin  
 valu  
 to  
 get  
 only  
 node  
 with  
 that  
 fault

•

**con**  
Op-  
tiona  
strin  
valu  
to  
get  
only  
node  
with  
that  
con-  
duc-  
tor\_\_

•

**own**  
Op-  
tiona  
strin  
valu  
that  
set  
the  
own  
who  
node  
are  
to  
be  
retru

•

**les**  
Op-  
tiona  
strin  
valu  
that  
set  
the  
lesse  
who  
node  
are  
to  
be  
re-  
turn

•

**pro**  
Op-

to be returned.

tion.  
 strin  
 valu  
 that  
 set  
 the  
 proj  
 -  
  
 lesse  
 or  
 own  
 -  
  
 who  
 node  
 are  
  
 •  
  
**des**  
 Op-  
 tion:  
 strin  
 valu  
 to  
 get  
 only  
 node  
 with  
 de-  
 scrip  
 tion  
 field  
 con-  
 tains  
 matc  
 ing  
 valu  
  
**from\_ch**  
 A  
 flag  
 to  
 in-  
 di-  
 cate  
 if  
 the  
 re-  
 ques  
 to



top-level resource Chassis

this  
con-  
troll  
are  
com  
ing  
from  
the

**get\_all**

Re-  
triev  
a  
list  
of  
node

**Parame**

- **cha**  
Op-  
tiona  
UUI  
of  
a  
chas  
sis,  
to  
get  
only  
node  
for  
that  
chas  
sis.
- **ins**  
Op-  
tiona  
UUI  
of  
an  
in-  
stan

stance.

ated nodes. May be combined with other parameters.

to  
 find  
 the  
 node  
 as-  
 so-  
 ci-  
 ated  
 with  
 that  
 in-  
  
 •  
**ass**  
 Op-  
 tion:  
 bool  
 whe  
 to  
 re-  
 turn  
 a  
 list  
 of  
 as-  
 so-  
 ci-  
 ated  
 or  
 unas  
 so-  
 ci-  
  
 •  
**mai**  
 Op-  
 tion:  
 bool  
 valu  
 that  
 in-  
 di-  
 cate  
 whe  
 to  
 get  
 node  
 in  
 main  
 te-

or not in maintenance mode (False).

nan  
mod  
(Tru

- **ret**  
Op-  
tiona  
bool  
valu  
that  
in-  
di-  
cate  
whe  
to  
get  
re-  
tired  
node

- **pro**  
Op-  
tiona  
strin  
valu  
to  
get  
only  
node  
in  
that  
pro-  
vi-  
sion  
state

- **mar**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

- **lim**

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  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.  
This  
  
•  
**sort**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.  
  
•  
**sort**  
di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc  
De-  
fault  
asc.  
  
•  
**drive**  
Op-  
tion  
strin

valu  
to  
get  
only  
node  
us-  
ing  
that  
drive

- **res**  
Op-  
tiona  
strin  
valu  
to  
get  
only  
node  
with  
that  
re-  
sour

- **con**  
Op-  
tiona  
strin  
valu  
to  
get  
only  
node  
with  
that  
con-  
duc-  
tor\_

- **con**  
Op-  
tiona  
strin  
valu  
to  
get  
only  
node  
man  
aged

by  
 that  
 con-  
 duc-  
 tor.

- **own**  
 Op-  
 tion:  
 strin  
 valu  
 that  
 set  
 the  
 own  
 who  
 node  
 are  
 to  
 be  
 retru

- **les**  
 Op-  
 tion:  
 strin  
 valu  
 that  
 set  
 the  
 lesse  
 who  
 node  
 are  
 to  
 be  
 re-  
 turn

- **pro**  
 Op-  
 tion:  
 strin  
 valu  
 that  
 set  
 the  
 proj  
 -

to be returned.

returned.

lesse  
or  
own  
-

who  
node  
are

- **fi**  
Op-  
tiona  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field  
of  
the  
re-  
sour  
to  
be

- **fa**  
Op-  
tiona  
strin  
valu  
to  
get  
only  
node  
with  
that  
fault

- **des**  
Op-  
tiona  
strin  
valu  
to

get  
only  
node  
with  
de-  
scrip-  
tion  
field  
con-  
tains  
match-  
ing  
valu

**get\_one**  
Re-  
triev-  
in-  
for-  
ma-  
tion  
about  
the  
give  
node

**Parame**

- node**  
UI  
or  
log-  
i-  
cal  
nam  
of  
a  
node
- file**  
Op-  
tion:  
a  
list  
with  
a  
spec  
i-  
fied  
set



returned.

of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

invalid

mainten

Ex-  
 pose  
 pose  
 main  
 te-  
 nanc  
 as  
 a  
 sub-  
 elem  
 of  
 node

managen

Ex-  
 pose  
 pose  
 man  
 age-  
 men  
 as  
 a  
 sub-  
 elem  
 of  
 node

patch (/

Up-  
 date  
 an  
 ex-  
 ist-  
 ing  
 node

Parame

•  
 nod  
 UUI

ing the driver field.

or  
log-  
i-  
cal  
nam  
of  
a  
node

- **res**  
 whe  
to  
re-  
set  
hard  
ware  
in-  
ter-  
face  
to  
their  
de-  
fault  
Only  
valid  
whe  
up-  
dat-

- **pat**  
 a  
json  
PAT  
doc-  
u-  
men  
to  
ap-  
ply  
to  
this  
node

**post** (*no*  
 Cre-  
ate  
a  
new  
node

Parameters

**node**  
 a  
 node  
 with  
 the  
 re-  
 ques  
 body

states

Ex-  
 pose  
 the  
 state  
 con-  
 troll  
 ac-  
 tion  
 as  
 a  
 sub-  
 elem  
 of  
 node

validation

Val-  
 i-  
 date  
 the  
 drive  
 in-  
 ter-  
 face  
 us-  
 ing  
 the  
 node  
 UUI  
 or  
 nam  
  
 Note  
 that  
 the  
 node  
 in-  
 ter-  
 face  
 is  
 dep-  
 re-

cater  
 in  
 favo  
 of  
 the  
 node  
 in-  
 ter-  
 face

Paramete

- **node**  
 UI  
 or  
 nam  
 of  
 a  
 node
- **node**  
 UI  
 of  
 a  
 node

vendor\_

A  
 re-  
 sour  
 used  
 for  
 ven-  
 dors  
 to  
 ex-  
 pose  
 a  
 cus-  
 tom  
 func  
 tion-  
 al-  
 ity  
 in  
 the

API

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  
 The  
 field  
 are  
 only  
 mad  
 avai

able when the requests API version matches or exceeds the versions when these fields were introduced.

ironic.  
 Add  
 link  
 to  
 the  
 in-  
 di-  
 ca-  
 tor.

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

pre-  
serv-  
then  
all

ironic

ironic.  
 Whe  
cre-  
at-  
ing  
an  
ob-  
ject,  
re-  
ject  
field  
that  
ap-  
pear  
in  
new  
ver-  
sion

ironic.

ironic.  
 Cha  
pro-  
vi-  
sion  
state  
nam  
for  
API  
back  
war  
com  
pat-  
i-  
bil-  
ity.

**Paramet**

**obj**  
 The  
dict  
be-  
ing  
re-

method.

turn  
to  
the  
API  
clien  
that  
is  
to  
be  
up-  
date  
by  
this

ironic.  
Val-  
i-  
date  
node  
net-  
worl  
field  
  
This  
meth  
val-  
i-  
date  
net-  
worl  
data  
con-  
fig-  
u-  
ra-  
tion  
agai  
JSO  
sche

Paramet  
**net**  
a  
net-  
worl  
field  
to  
val-  
i-  
date

Raises



In-  
valid  
if  
net-  
worl  
data  
is  
not  
sche  
com

## ironic.api.controllers.v1.notification\_utils module

ironic.

Help  
for  
emit  
ting  
API  
end  
no-  
ti-  
fi-  
ca-  
tions

### Parameters

- **con**  
re-  
ques  
con-  
text.
- **obj**  
re-  
sour  
rpc  
ob-  
ject.
- **act**  
Ac-  
tion  
strin

to  
 go  
 in  
 the  
 Ever  
 Type

- kwa**  
 kwa  
 to  
 use  
 whe  
 cre-  
 at-  
 ing  
 the  
 no-  
 ti-  
 fi-  
 ca-  
 tion  
 pay-  
 load

ironic.

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  
to  
go  
in  
the  
Ever  
Type

• **kwa**  
kwa  
to  
use  
whe  
cre-  
at-  
ing  
the  
no-  
ti-  
fi-  
ca-  
tion  
pay-  
load

ironic.

Con  
text  
man  
ager  
to  
han-  
dle  
any  
er-  
ror  
no-  
ti-

fi-  
 ca-  
 tion

Paramet

- **con**  
 re-  
 ques  
 con-  
 text.
- **obj**  
 re-  
 sour  
 rpc  
 ob-  
 ject.
- **act**  
 Ac-  
 tion  
 strin  
 to  
 go  
 in  
 the  
 Ever  
 Type
- **kwa**  
 kwa  
 to  
 use  
 whe  
 cre-  
 at-  
 ing  
 the  
 no-  
 ti-  
 fi-  
 ca-  
 tion  
 pay-  
 load

## ironic.api.controllers.v1.port module

**class** i

Base  
pec  
res  
Res  
RES  
con-  
troll  
for  
Port

**advance**

**delete**

Dele  
a  
port

**Parame**

**por**  
UUI  
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 interface is deprecated in favor of the node interface.

Parameters

- **node**  
 The node or name of a node to get only port for that node.
- **node**  
 The node or name of a node to get only port for that node.

- **add**  
MA  
ad-  
dres  
of  
a  
port  
to  
get  
the  
port  
whic  
has  
this  
MA  
ad-  
dres
- **por**  
UUI  
or  
nam  
of  
a  
port  
grou  
to  
get  
only  
port  
for  
that  
port  
grou
- **mar**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.
- **lim**  
max  
i-

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.



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  
in-  
ter-  
face

**Parame**

- **nod**  
UUI  
or  
nam  
of  
a  
node  
to  
get  
only  
port  
for  
that  
node
- **nod**  
UUI  
of

a  
 node  
 to  
 get  
 only  
 port  
 for  
 that  
 node

- **add**  
 MA  
 ad-  
 dres  
 of  
 a  
 port  
 to  
 get  
 the  
 port  
 whic  
 has  
 this  
 MA  
 ad-  
 dres

- **mar**  
 pag-  
 i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets.

- **lim**  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to  
 re-

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironiC configuration, or only `max_limit` resources will be returned.

turn  
in  
a  
sin-  
gle  
re-  
sult.  
This

- **sort**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.

- **sort**  
di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc  
De-  
fault  
asc.

- **file**  
Op-  
tion  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field  
of

returned.

the  
 re-  
 sour  
 to  
 be  
 •  
 por  
 UUI  
 or  
 nam  
 of  
 a  
 port  
 grou  
 to  
 get  
 only  
 port  
 for  
 that  
 port  
 grou  
 Raises  
 No-  
 tAc-  
 cept  
 able  
 HTT  
 Not-  
 Four  
 get\_one  
 Re-  
 triev  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 give  
 port  
 Parame  
 •  
 por  
 UUI  
 of  
 a

returned.

port
 

- file**
 Op-
 tion:
 a
 list
 with
 a
 spec
 i-
 fied
 set
 of
 field
 of
 the
 re-
 sour
 to
 be

Raises

No-
 tAc-
 cept
 able
 HTT
 Not-
 Four

invalid

patch (p

Up-
 date
 an
 ex-
 ist-
 ing
 port

Paramete

- port**
 UUI
 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.

**Parameter**  
**field**  
(list  
of  
strings  
list  
of  
field  
to  
pre-  
serv  
or

ironic.api.controllers.v1.portgroup module

Non  
 to  
 pre-  
 serv  
 then  
 all

**class** i  
  
 Base  
 pec  
 res  
 Res  
  
 RES  
 con-  
 troll  
 for  
 port  
 grou

**delete**  
 Dele  
 a  
 port  
 grou

**Paramete**  
**por**  
 UI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 port  
 grou

**detail**  
  
 Re-  
 triev  
 a  
 list  
 of  
 port  
 grou  
 with



de-  
tail.

## Parame

- **node**  
UUI  
or  
nam  
of  
a  
node  
to  
get  
only  
port  
grou  
for  
that  
node
- **add**  
MA  
ad-  
dres  
of  
a  
port  
grou  
to  
get  
the  
port  
grou  
whic  
has  
this  
MA  
ad-  
dres
- **mar**  
pag-  
i-  
na-  
tion  
marl  
for  
large

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.

**get\_all**

Re-  
triev  
a  
list  
of  
port  
grou

**Parame**

- **nod**  
UUU  
or  
nam  
of  
a  
node  
to  
get  
only  
port  
grou  
for  
that  
node

- **add**  
MA  
ad-  
dres  
of  
a  
port  
grou  
to  
get  
the  
port  
grou  
whic  
has  
this  
MA  
ad-  
dres

- **mar**

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironiC configuration, or only `max_limit` resources will be returned.

pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

- **lim**  
max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.  
This

- **sort**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.

- **sort**  
di-  
rec-  
tion  
to  
sort.

returned.

asc  
 or  
 desc  
 De-  
 fault  
 asc.

•
 

**file**  
 Op-  
 tion  
 a  
 list  
 with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

**get\_one**  
 Re-  
 triev  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 give  
 port  
 grou

**Parame**

•
 

**por**  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam

returned.

of  
 a  
 port  
 grou  
 •  
**fie**  
 Op-  
 tion  
 a  
 list  
 with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

invalid

**patch** (*p*  
 Up-  
 date  
 an  
 ex-  
 ist-  
 ing  
 port  
 grou

Paramete

•  
**por**  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 port

group

- **pat**  
a  
json  
PAT  
doc-  
u-  
men  
to  
ap-  
ply  
to  
this  
port  
group

**post** (*po*  
Cre-  
ate  
a  
new  
port  
group

**Parame**  
**por**  
a  
port  
group  
with  
the  
re-  
ques  
body

`ironic.`

Add  
links  
to  
the  
port  
group

`ironic.`

ironic.api.controllers.v1.ramdisk module

**class** `ironic.api.controllers.v1.ramdisk.RamdiskController`

Base class for the ramdisk controller.

The `RamdiskController` class is responsible for handling the ramdisk deployment process.

**post** (*no body*)

POST /v1/ramdisk

This method is used to initiate the ramdisk deployment process.

Parameters

- node**: The node ID of the node to be deployed.
- cal**: The calibration data for the node.



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.

to  
reac  
back  
to  
the  
rame

- **age**  
The  
ver-  
sion  
of  
the  
ager  
that  
is  
hear  
beat  
ing.  
Non  
in-  
di-  
cate  
that  
the  
ager
- **age**  
ran-  
dom  
gen-  
er-  
ated  
val-  
i-  
da-  
tion  
to-  
ken.
- **age**  
TLS  
cer-  
tifi-  
cate  
to  
use  
to

con-  
nect  
to  
the  
ager

Raises

Nod  
Not-  
Four  
if  
node  
with  
pro-  
vide  
UUI  
or  
nam  
was  
not  
foun

Raises

In-  
valid  
uidC  
Nam  
if  
node  
is  
not  
valid  
nam  
or  
UUI

Raises

No-  
Valid  
Hos  
if  
RPC  
topic  
for  
node  
coul  
not  
be  
re-  
triev

Raises

Not-

Four  
if  
re-  
ques  
API  
ver-  
sion  
does  
not  
al-  
low  
this  
end-  
poin

**class** i

Base  
pec  
res  
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  
UI  
  
If  
the

tain transient states (e.g. deploy wait).

re-  
 stric-  
 op-  
 tion  
 is  
 set  
 to  
 True  
 (the  
 de-  
 fault  
 limit  
 the  
 search  
 to  
 node  
 in  
 cer-

Paramete

- **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-

for the lookup.

low  
 this  
 end-  
 poin  
**Raises**  
 Not-  
 Foun  
 if  
 suit-  
 able  
 node  
 was  
 not  
 foun  
 or  
 node  
 pro-  
 vi-  
 sion  
 state  
 is  
 not  
 al-  
 lowe

**Raises**  
 In-  
 com  
 plete  
 if  
 nei-  
 ther  
 node  
 UUI  
 nor  
 any  
 valid  
 MA  
 ad-  
 dres  
 was  
 pro-  
 vide

**property**

ironic.

ironic.

ironic.api.controllers.v1.utils module

**class** `ironic.api.controllers.v1.utils.ObjectHolder`  
 Base object to hold the response from a pass call

**obj**  
 Store the result object from the view

**status\_**  
 Store an optional status\_

`ironic.check_if_agent_token_is_available`  
 Check if agent token is available

`ironic.check_if_agent_verified`  
 Check if agent verified

tion to IroniC on heartbeat.

sion  
is  
al-  
lowe  
to  
be  
pass  
into  
hear  
beat  
  
Ver-  
sion  
1.36  
of  
the  
API  
adde  
the  
abil-  
ity  
for  
ager  
to  
pass  
their  
ver-  
sion  
in-  
for-  
ma-

ironic.  
Che  
if  
back  
fill-  
ing  
al-  
lo-  
ca-  
tions  
is  
al-  
lowe  
  
Ver-  
sion  
1.58  
of  
the  
API

added support for backfilling allocations.  
 ironic. Check if allocation own field is allocated.  
 Version 1.60 of the API added the own field to the allocation object.  
 ironic. Check if updating an existing



ing  
 al-  
 lo-  
 ca-  
 tion  
 is  
 al-  
 lowe  
 or  
 not.

Ver-  
 sion  
 1.57  
 of  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 up-  
 dat-  
 ing  
 an  
 al-  
 lo-  
 ca-  
 tion.

ironic.  
 Che  
 if  
 ac-  
 cess  
 ing  
 al-  
 lo-  
 ca-  
 tion  
 end-  
 poin  
 is  
 al-  
 lowe

Ver-  
 sion  
 1.52  
 of  
 the  
 API  
 ex-

for the node.

pose  
al-  
lo-  
ca-  
tion  
end-  
poin  
and  
al-  
lo-  
ca-  
tion  
field

`ironic.`  
Che  
if  
we  
shou  
sup-  
port  
bios  
in-  
ter-  
face  
and  
end-  
poin  
  
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  
 1.56  
 of  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 buile  
 ing  
 con-  
 fig-  
 drive

ironic.  
 Che  
 if  
 con-  
 fig-  
 drive  
 can  
 con-  
 tain  
 a  
 ven-  
 dor\_  
 key.

Ver-  
 sion  
 1.59  
 of  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 con-  
 fig-  
 drive  
 ven-  
 dor\_

ironic.  
 Che  
 if

de-  
 ploy  
 are  
 avai  
 able  
  
 ironic.  
 Che  
 if  
 ac-  
 cess  
 ing  
 de-  
 ploy  
 tem-  
 plate  
 end-  
 poin  
 is  
 al-  
 lowe  
  
 Ver-  
 sion  
 1.55  
 of  
 the  
 API  
 ex-  
 pose  
 de-  
 ploy  
 tem-  
 plate  
 end-  
 poin  
  
 ironic.  
 Che  
 if  
 pass  
 ing  
 a  
 de-  
 tail=  
 quer  
 strin  
 is  
 al-  
 lowe  
  
 Ver-  
 sion

all the fields.

1.43  
al-  
lows  
a  
user  
to  
pass  
the  
de-  
tail  
quer  
strin  
to  
list  
the  
re-  
sour  
with  
  
ironic.  
Che  
if  
dy-  
nam  
drive  
API  
calls  
are  
al-  
lowe  
  
Ver-  
sion  
1.30  
of  
the  
API  
adde  
sup-  
port  
for  
all  
of  
the  
drive  
com  
po-  
si-  
tion  
re-  
latec

calls in the /v1/drivers API.

the node object.

ironic.  
 Che  
 if  
 dy-  
 nam  
 in-  
 ter-  
 face  
 field  
 are  
 al-  
 lowe  
  
 Ver-  
 sion  
 1.31  
 of  
 the  
 API  
 add  
 sup-  
 port  
 for  
 view  
 ing  
 and  
 set-  
 ting  
 the  
 field  
 in  
 V31  
 on

ironic.  
 Che  
 if  
 ac-  
 cess  
 ing  
 con-  
 duc-  
 tor  
 end-  
 poin  
 is  
 al-  
 lowe  
  
 Ver-

node.

sion  
1.49  
of  
the  
API  
ex-  
pose  
con-  
duc-  
tor  
end-  
poin  
and  
con-  
duc-  
tor  
field  
for  
the

ironic.  
Che  
if  
ac-  
cess  
ing  
ever  
end-  
poin  
is  
al-  
lowe

Ver-  
sion  
1.54  
of  
the  
API  
adde  
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



ing asynchronous hardware inspection.

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  
 wait  
 state  
 to  
 sub-  
 sti-  
 tute  
 in-  
 spec  
 ing  
 state  
 dur-  
  
 ironic.  
 Che  
 if  
 links  
 are  
 dis-  
 play  
 Ver-

ties.

sion  
 1.14  
 of  
 the  
 API  
 al-  
 lows  
 the  
 dis-  
 play  
 of  
 links  
 to  
 node  
 state  
 and  
 drive  
 prop  
 er-  
  
 ironic.  
 Che  
 if  
 net-  
 worl  
 is  
 al-  
 lowe  
 in  
 port  
 link  
  
 ironic.  
  
 ironic.  
 Che  
 if  
 we  
 shou  
 sup-  
 port  
 node  
 re-  
 build  
 with  
 con-  
 fig-  
 drive  
  
 Ver-  
 sion

1.35  
of  
the  
API  
adde  
sup-  
port  
for  
node  
re-  
buil  
with  
con-  
fig-  
drive

ironic.  
Che  
if  
we  
shou  
re-  
turn  
lo-  
cal\_  
and  
pxe\_  
field

Ver-  
sion  
1.19  
of  
the  
API  
adde  
sup-  
port  
for  
thes  
new  
field  
in  
port  
ob-  
ject.

ironic.  
Che  
if  
ac-  
cess  
ing

in-  
 ter-  
 nal\_  
 is  
 al-  
 lowe  
 for  
 the  
 port  
  
 Ver-  
 sion  
 1.18  
 of  
 the  
 API  
 ex-  
 pose  
 in-  
 ter-  
 nal\_  
 read  
 only  
 field  
 for  
 the  
 port  
  
 ironic.  
 Che  
 if  
 port  
 is\_s  
 field  
 is  
 al-  
 lowe  
  
 Ver-  
 sion  
 1.53  
 of  
 the  
 API  
 adde  
 is\_s  
 field  
 to  
 the  
 port  
 ob-  
 ject.

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.

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  
 to  
 the  
 port  
 ob-  
 ject.  
 We

ironic.  
 Che  
 if  
 mod  
 and  
 prop  
 er-  
 ties  
 can  
 be  
 adde  
 to/q  
 from  
 a  
 port

group  
 Ver-  
 sion  
 1.26  
 of  
 the  
 API  
 adde  
 mod  
 and  
 prop  
 er-  
 ties  
 field  
 to  
 port  
 grou  
 ob-  
 ject.  
 ironic.  
 Che  
 if  
 we  
 shou  
 sup-  
 port  
 port  
 grou  
 op-  
 er-  
 a-  
 tions  
 Ver-  
 sion  
 1.23  
 of  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 Port  
 Gro  
 ironic.  
 Che  
 if  
 port  
 grou

can  
 be  
 used  
 as  
 sub-  
 con-  
 troll

Ver-  
 sion  
 1.24  
 of  
 the  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 Port  
 grou  
 as  
 sub-  
 con-  
 troll

ironic.  
 Che  
 if  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 al-  
 lowe  
 for  
 the  
 node

Ver-  
 sion  
 1.12  
 of  
 the  
 the  
 API  
 al-  
 lows  
 RAI  
 con-  
 fig-  
 u-

ra-  
 tion  
 for  
 the  
 node

ironic.  
 Che  
 if  
 hear  
 beat  
 and  
 look  
 end-  
 poin  
 are  
 al-  
 lowe

Ver-  
 sion  
 1.22  
 of  
 the  
 API  
 in-  
 tro-  
 duce  
 then

ironic.  
 Che  
 if  
 chas  
 sis\_  
 can  
 be  
 re-  
 mov  
 from  
 node

Ver-  
 sion  
 1.25  
 of  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 chas



face.

sis\_1  
re-  
mov  
  
ironic.  
Che  
if  
we  
shou  
sup-  
port  
res-  
cue  
and  
un-  
res-  
cue  
op-  
er-  
a-  
tions  
and  
in-  
ter-  
  
Ver-  
sion  
1.38  
of  
the  
API  
adde  
sup-  
port  
for  
res-  
cue  
and  
un-  
res-  
cue.  
  
ironic.  
Che  
if  
pass  
ing  
a  
re-  
set\_  
quer  
strin

is  
al-  
lowe  
  
ironic.  
Che  
if  
Soft  
Pow  
Off  
is  
al-  
lowe  
for  
the  
node  
  
Ver-  
sion  
1.27  
of  
the  
API  
al-  
lows  
Soft  
Pow  
Off,  
in-  
clud  
ing  
Soft  
Re-  
boot  
for  
the  
node  
  
ironic.  
Che  
if  
we  
shou  
sup-  
port  
stor-  
age\_  
node  
and  
drive  
field  
Ver-

sion  
1.33  
of  
the  
API  
add  
sup-  
port  
for  
stor-  
age  
in-  
ter-  
face

`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  
 adde  
 sup-  
 port  
 for  
 VIF  
 to  
 be  
 at-  
 tach  
 to  
 Nod  
 ironic.  
 Che  
 if  
 vol-  
 ume  
 con-  
 nec-  
 tors  
 and  
 tar-  
 gets  
 are  
 al-  
 lowe  
 Ver-  
 sion  
 1.32  
 of  
 the  
 API  
 adde  
 sup-  
 port  
 for  
 vol-  
 ume  
 con-  
 nec-  
 tors  
 and  
 tar-

gets  
 ironic.  
 Apply  
 a  
 JSON  
 patch  
 one  
 op-  
 er-  
 a-  
 tion  
 at  
 a  
 time  
 If  
 the  
 patch  
 fails  
 to  
 ap-  
 ply,  
 this  
 al-  
 lows  
 us  
 to  
 de-  
 ter-  
 mine  
 whic  
 op-  
 er-  
 a-  
 tion

failed, making the error message a little less cryptic.

## Paramet

- **doc**  
 The  
 JSON  
 doc-  
 u-  
 men  
 to  
 patch
- **pat**

The  
 JSO  
 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.

Che  
 if  
 the  
 spec

i-  
 fied  
 pol-  
 icy  
 au-  
 tho-  
 rizes  
 re-  
 ques  
 on  
 al-  
 lo-  
 ca-  
 tion.

**Param**

pol-  
 icy\_  
 Nam  
 of  
 the  
 pol-  
 icy  
 to  
 chec

**Param**

al-  
 lo-  
 ca-  
 tion,  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

**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.

Che  
if  
de-  
ploy  
step  
are  
al-  
lowe

ironic.

Che  
if  
get-  
ting  
de-  
taile  
drive  
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  
al-  
lowe

Ver-  
sion  
1.49  
of  
the  
API  
al-  
lows  
fil-  
ter-  
ing  
node  
by  
con-  
duc-  
tor.

ironic.  
Che  
if  
fil-  
ter-  
ing  
node  
by  
con-  
duc-  
tor\_

is  
al-  
lowe  
  
Ver-  
sion  
1.46  
of  
the  
API  
al-  
lows  
fil-  
ter-  
ing  
node  
by  
con-  
duc-  
tor\_

`ironic.`  
Che  
if  
fil-  
ter-  
ing  
node  
by  
fault  
is  
al-  
lowe

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.`

Check if filtering is allowed by the class/sic/c is allowed. Version 1.30 of the API allows this.

ironic.

ironic. Check if filtering is allowed by the node driver is allowed. Version 1.16 of the API allows filtering nodes by driver ironic.

method checks if the required version is being requested.

Che  
if  
fetc  
ing  
a  
sub-  
set  
of  
the  
re-  
sour  
at-  
tribu  
is  
al-  
lowe  
  
Ver-  
sion  
1.8  
of  
the  
API  
al-  
lows  
fetc  
ing  
a  
sub-  
set  
of  
the  
re-  
sour  
at-  
tribu  
this

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-  
sour  
  
ironic.  
Che  
if  
fetc  
ing  
a  
par-  
tic-  
u-  
lar  
field  
is  
al-  
lowe  
  
This  
meth  
chec  
if  
the  
re-  
quir  
ver-  
sion  
is  
be-  
ing  
re-  
ques  
for  
field  
that  
are  
only  
al-

lowed to be fetched in a particular API version.

lowed to be fetched in a particular API version.

ironic.  
Che  
if  
fetc  
ing  
a  
par-  
tic-  
u-  
lar  
field  
of  
a  
port  
grou  
is  
al-  
lowe  
  
This  
meth  
chec  
if  
the  
re-  
quir  
ver-  
sion  
is  
be-  
ing  
re-  
ques  
for  
field  
that  
are  
only  
al-

ironic.  
  
Che  
for  
re-  
ques  
non-  
exist  
field  
  
Che  
if

the  
 user  
 re-  
 ques  
 non-  
 existi  
 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-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 in-  
 valid  
 field  
 were  
 re-  
 ques

ironic.  
 Che  
 if  
 fil-  
 ter-



ing  
node  
by  
pro-  
vi-  
sion  
state  
is  
al-  
lowe  
  
Ver-  
sion  
1.9  
of  
the  
API  
al-  
lows  
fil-  
ter  
node  
by  
pro-  
vi-  
sion  
state

`ironic.`

Che  
if  
the  
list  
pol-  
icy  
au-  
tho-  
rizes  
this  
re-  
ques  
on  
an  
ob-  
ject.

**Param**  
ob-  
ject  
type  
of  
ob-

ject  
 be-  
 ing  
 chec

**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  
 i-  
 fied  
 poli-  
 cies

au-  
 tho-  
 rize  
 this  
 re-  
 ques  
 on  
 a  
 node

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-  
 ques  
 on  
 a  
 node

**Param**

pol-  
 icy\_  
 Nam

of  
the  
pol-  
icy  
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  
found

**Returns**

RPC  
node  
identi-  
fied  
by  
node

ironic.

Check  
if  
the  
poli-  
cy  
au-  
tho-  
rizes  
this  
re-  
ques  
on  
an  
ob-  
ject.

**Param**

ob-  
ject\_  
type  
of  
ob-  
ject  
be-  
ing  
check

**Param**

poli-  
cy\_  
Name  
of  
the  
poli-  
cy  
to

chec

**Param**

own  
the  
own

**Param**

lesse  
the  
lesse

**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  
re-  
ques

**Policy\_n**

Nam  
of  
the  
pol-  
icy  
to  
chec

**Raises**

HTTP  
 For-  
 bid-  
 den  
 if  
 the  
 pol-  
 icy  
 for-  
 bids  
 ac-  
 cess

ironic.  
 Che  
 if  
 the  
 spec  
 i-  
 fied  
 pol-  
 icy  
 au-  
 tho-  
 rizes  
 this  
 re-  
 ques  
 on  
 a  
 port

Raises

HTTP  
 For-  
 bid-  
 den  
 if  
 the  
 pol-  
 icy  
 for-  
 bids  
 ac-  
 cess

Returns

own  
 that  
 shou  
 be  
 used  
 for



list  
 quer  
 if  
 need

ironi

Che  
 if  
 the  
 spec  
 i-  
 fied  
 pol-  
 icy  
 au-  
 tho-  
 rizes  
 this  
 re-  
 ques  
 on  
 a  
 port

Param

pol-  
 icy\_  
 Nam  
 of  
 the  
 pol-  
 icy  
 to  
 chec

Param

port  
 the  
 UUI  
 of  
 a  
 port

Raises

HT  
 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  
 and  
 as-  
 so-  
 ci-  
 ated  
 node

ironic.  
 Gen  
 er-  
 a-  
 tor  
 of  
 field  
 not  
 al-  
 lowe  
 in  
 the  
 cur-  
 rent  
 re-  
 ques

ironic.  
 Get  
 re-  
 serv  
 nam

that can not be used as an identifier for a resource because the names are either being used as a custom action or is the name of a nested controller inside the given class.

for  
a  
give  
con-  
troll  
  
In-  
spec  
the  
con-  
troll  
class  
and  
re-  
turn  
the  
re-  
serv  
nam  
with  
it.  
Re-  
serv  
nam  
are  
nam

#### Paramet

**cls**  
The  
con-  
troll  
class  
to  
be  
in-  
spec

`ironic.`  
Get  
the  
patc  
val-  
ues  
cor-  
re-  
spor  
ing  
to  
the  
spec

```
↪ 'abc' } ,
```

(continues on next page)

i-  
fied  
path  
  
If  
there  
are  
mul-  
ti-  
ple  
val-  
ues  
spec-  
i-  
fied  
for  
the  
sam-  
path  
for  
ex-  
am-  
ple

(continued from previous page)

--

re-  
 turn  
 all  
 of  
 then  
 in  
 a  
 list  
 (pre  
 serv  
 ing  
 or-  
 der)

Paramet

- **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-  
ques  
at  
the  
sam  
time  
To

use the detail query we need to be on a version of the API greater than 1.43. This function raises an `InvalidParameterValue` exception if either of these conditions are not met.

If  
thes  
chec  
pass  
then  
this  
func  
tion  
will  
re-  
turn  
ei-

fault fields provided.

ther  
the  
field  
pass  
in  
or  
the  
de-

Paramet

- **fi**  
The  
field  
quer  
pass  
into  
the  
API  
re-  
ques
- **det**  
The  
de-  
tail  
quer  
pass  
into  
the  
API  
re-  
ques
- **def**  
The  
de-  
fault  
field  
to  
re-  
turn  
if  
field  
and  
de-  
tail=

Raises

strings or API version.

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 there  
 is  
 an  
 in-  
 valid  
 com  
 bi-  
 na-  
 tion  
 of  
 quer

Returns

field  
 pass  
 in  
 valu  
 or  
 de-  
 fault

ironic.  
 Get  
 the  
 RPC  
 al-  
 lo-  
 ca-  
 tion  
 from  
 the  
 al-  
 lo-  
 ca-  
 tion  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam

Paramet  
 all



the  
 UI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 an  
 al-  
 lo-  
 ca-  
 tion.

Returns

The  
 RPC  
 al-  
 lo-  
 ca-  
 tion.

Raises

In-  
 valid  
 uidC  
 Nam  
 if  
 the  
 nam  
 or  
 uuid  
 pro-  
 vide  
 is  
 not  
 valid

Raises

Al-  
 lo-  
 ca-  
 tion-  
 Not-  
 Foun  
 if  
 the  
 al-  
 lo-  
 ca-  
 tion  
 is  
 not

tion\_ident with .json suffix. Otherwise identical to get\_rpc\_allocation.

found  
 ironic.  
 Get  
 the  
 RPC  
 al-  
 lo-  
 ca-  
 tion  
 from  
 the  
 al-  
 lo-  
 ca-  
 tion  
 UUID  
 or  
 log-  
 i-  
 cal  
 nam  
 If  
 HAS  
 flag  
 is  
 set  
 in  
 the  
 pecca  
 en-  
 vi-  
 ron-  
 men  
 try  
 also  
 look  
 ing  
 for  
 al-  
 lo-  
 ca-  
 Paramet  
 all  
 the  
 UUID  
 or  
 log-  
 i-  
 cal

nam
 of
 an
 al-
 lo-
 ca-
 tion.

Returns

The
 RPC
 al-
 lo-
 ca-
 tion.

Raises

In-
 valid
 uidC
 Nam
 if
 the
 nam
 or
 uuid
 pro-
 vide
 is
 not
 valid

Raises

Al-
 lo-
 ca-
 tion.
 Not-
 Four
 if
 the
 al-
 lo-
 ca-
 tion
 is
 not
 foun

ironic.
 Get
 the
 RPC

de-  
 ploy  
 tem-  
 plate  
 from  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam

Paramet

tem  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 de-  
 ploy  
 tem-  
 plate

Returns

The  
 RPC  
 de-  
 ploy  
 tem-  
 plate

Raises

In-  
 valid  
 uidC  
 Nam  
 if  
 the  
 nam  
 or  
 uuid  
 pro-  
 vide  
 is  
 not  
 valid

**Raises**

De-  
ploy  
plate  
Foun  
if  
the  
de-  
ploy  
tem-  
plate  
is  
not  
foun

`ironic.`

Get  
the  
RPC  
de-  
ploy  
tem-  
plate  
from  
the  
UUID  
or  
log-  
i-  
cal  
nam

If  
HAS  
flag  
is  
set  
in  
the  
peca  
en-  
vi-  
ron-  
men  
try  
also  
look  
ing  
for  
tem-  
plate  
with

.json suffix. Otherwise identical to get\_rpc\_deploy\_template.

Paramet

tem  
the  
UUI  
or  
log-  
i-  
cal  
nam  
of  
a  
de-  
ploy  
tem-  
plate

Returns

The  
RPC  
de-  
ploy  
tem-  
plate

Raises

In-  
valid  
uidC  
Nam  
if  
the  
nam  
or  
uuid  
pro-  
vide  
is  
not  
valid

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

**Parameters**  
 node  
 the  
 UUID  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 node

**Returns**  
 The  
 RPC  
 Node

**Raises**  
 In-  
 valid  
 uuidC  
 Nam  
 if  
 the  
 nam  
 or  
 uuid  
 pro-  
 vide  
 is  
 not  
 valid

**Raises**

suffix. Otherwise identical to `get_rpc_node`.

Node.  
 Not-  
 Four-  
 if  
 the  
 node  
 is  
 not  
 found  
  
`ironic.`  
 Get  
 the  
 RPC  
 node  
 from  
 the  
 node  
 uuid  
 or  
 log-  
 i-  
 cal  
 name  
  
 If  
 HAS  
 flag  
 is  
 set  
 in  
 the  
 peca  
 en-  
 vi-  
 ron-  
 men  
 try  
 also  
 look  
 ing  
 for  
 node  
 with  
 .json  
  
**Paramet**  
**nod**  
 the  
 UUID  
 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  
valid

**Raises**

Nod  
Not-  
Foun  
if  
the  
node  
is  
not  
foun

ironic.  
Get  
the  
RPC  
port  
grou  
from  
the  
port  
grou  
UUID  
or  
log-  
i-

cal  
nam

**Parameters**

**port**  
the  
UUID  
or  
log-  
i-  
cal  
nam  
of  
a  
port  
grou

**Returns**

The  
RPC  
port  
grou

**Raises**

In-  
valid  
uidC  
Nam  
if  
the  
nam  
or  
uuid  
pro-  
vide  
is  
not  
valid

**Raises**

Port  
grou  
Not-  
Foun  
if  
the  
port  
grou  
is  
not  
foun

ironic.  
Get

.json suffix. Otherwise identical to `get_rpc_portgroup`.

the  
 RPC  
 port  
 group  
 from  
 the  
 port  
 group  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
  
 If  
 HAS  
 flag  
 is  
 set  
 in  
 the  
 peca  
 en-  
 vi-  
 ron-  
 men  
 try  
 also  
 look  
 ing  
 for  
 port  
 group  
 with

Paramet

por  
 the  
 UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 of  
 a  
 port  
 group

Returns  
 The

RPC  
 port  
 group  
**Raises**  
 In-  
 valid  
 uidC  
 Nam  
 if  
 the  
 nam  
 or  
 uuid  
 pro-  
 vide  
 is  
 not  
 valid

**Raises**  
 Port  
 grou  
 Not-  
 Foun  
 if  
 the  
 port  
 grou  
 is  
 not  
 foun

ironic.

Han  
 dle  
 a  
 Patc  
 re-  
 ques  
 that  
 mod  
 i-  
 fies  
 .ex-  
 tra[v  
 This  
 han-

field.

Paramet

- **rpc**  
a  
Port  
or  
Port  
grou  
RPC  
ob-  
ject
- **int**  
Dict  
of  
port  
or  
port  
grou  
in-  
ter-  
nal  
info
- **pat**  
the  
JSO  
patc  
in

port groups extra[vif\_port\_id] field.

the  
 API  
 re-  
 ques  
 ironic  
 Han  
 dle  
 a  
 Post  
 re-  
 ques  
 that  
 sets  
 .ex-  
 tra[v  
 This  
 han-  
 dles  
 at-  
 tach  
 of  
 VIF  
 via  
 spec  
 i-  
 fy-  
 ing  
 the  
 VIF  
 port  
 ID  
 in  
 a  
 port  
 or  
 Paramet  
 p\_d  
 a  
 dic-  
 tio-  
 nary  
 with  
 field  
 nam  
 for  
 the  
 port  
 or  
 port

ENROLL.

grou
 **Returns**
 VIF
 or
 Non
 ironic.
 Re-
 turn
 node
 state
 to
 use
 by
 de-
 fault
 whe
 cre-
 at-
 ing
 new
 node
 Pre-
 vi-
 ously
 the
 de-
 fault
 state
 for
 new
 node
 was
 AVA
 ABI
 Star
 ing
 with
 API
 1.11
 it
 is
 ironic.
 Re-
 turn
 whe
 the
 patc
 in-

clud  
 re-  
 mov  
 of  
 the  
 path  
 (or  
 sub-  
 path  
 of).

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

ironic.
   
 Re-
   
 turn
   
 whe
   
 the
   
 patc
   
 in-
   
 clud
   
 op-



er-  
a-  
tion  
on  
path  
(or  
its  
sub-  
path

Paramet

- **pat**  
HTT  
PAT  
re-  
ques  
body
- **pat**  
the  
path  
to  
chec

Returns

True  
if  
path  
or  
sub-  
path  
be-  
ing  
patc  
Fals  
oth-  
er-  
wise

ironic.  
De-  
ter-  
min  
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  
 UUI

**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

#### Paramet

- **obj**  
RPC  
ob-  
ject  
to  
con-  
vert  
to  
a  
dict
- **inc**  
Whe  
to  
in-  
clud  
stan  
dard  
base  
class  
at-  
tribu

resource name

cre-  
 ated  
 •  
**inc**  
 Whe  
 to  
 in-  
 clud  
 stan  
 dard  
 base  
 class  
 at-  
 tribu  
 up-  
 date  
 •  
**inc**  
 Whe  
 to  
 in-  
 clud  
 stan  
 dard  
 base  
 class  
 at-  
 tribu  
 uuid  
 •  
**lin**  
 Whe  
 spec  
 i-  
 fied,  
 gen-  
 er-  
 ate  
 a  
 lin  
 valu  
 with  
 a  
 sel  
 and  
 boo  
 us-  
 ing  
 this

the object `uuid` will be used.

- **lin**  
Re-  
sour  
ar-  
gu-  
men  
to  
be  
adde  
to  
gen-  
er-  
ated  
links  
Whe  
not  
spec  
i-  
fied,

- **fi**  
Key  
nam  
for  
dict  
val-  
ues  
to  
pop-  
u-  
late  
di-  
rectl  
from  
ob-  
ject  
at-  
tribu

**Returns**  
A  
dict  
con-  
tain-  
ing  
val-  
ues  
from  
the

ob-  
 ject  
 ironic.

Up-  
 date  
 rpc  
 ob-  
 ject  
 base  
 on  
 char  
 field  
 in  
 a  
 dict.  
 Only  
 field  
 whic  
 have  
 a  
 cor-  
 re-  
 spor  
 ing  
 sche  
 field  
 are  
 up-  
 date  
 whe  
 char  
 Othe  
 val-  
 ues  
 can

be updated using the id\_map.

Paramet

- **fro**  
 Dict  
 con-  
 tain-  
 ing  
 char

field  
val-  
ues

- **rpc**  
Ob-  
ject  
to  
up-  
date  
char  
field  
on

- **fie**  
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  
val-

no matching field in the schema

ues  
 whe  
 there  
 is

ironic.

Val-  
 i-  
 date  
 that  
 a  
 patc  
 list  
 only  
 mod  
 i-  
 fies  
 al-  
 lowe  
 field

Paramet

- **pat**  
 List  
 of  
 patc  
 dict  
 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.

Val-  
 i-  
 date  
 a  
 patc  
 dict  
 ob-  
 ject  
 agai  
 a  
 val-  
 ida-  
 tor  
 or  
 sche

This  
 func  
 tion  
 has  
 the  
 side  
 effec  
 of  
 dele  
 ing  
 any

allows database-loaded objects to be pruned of their internal values before validation.

Paramet

- pat**

dict  
rep-  
re-  
sen-  
ta-  
tion  
of  
the  
ob-  
ject  
with  
patc  
up-  
date  
ap-  
plie
- sch**

Any  
dict  
key  
not  
in  
the  
sche  
will  
be  
dele  
from  
the  
dict.  
If  
no  
val-  
ida-  
tor

is specified then the resulting `patched_dict` will be validated against the schema

beyond the schema

- **val**  
Op-  
tiona  
val-  
ida-  
tor  
to  
use  
if  
there  
is  
ex-  
tra  
val-  
i-  
da-  
tion  
re-  
quir

**Raises**  
ex-  
cep-  
tion.  
if  
val-  
i-  
da-  
tion  
fails

`ironic.`  
Look  
up  
the  
node  
ref-  
er-  
ence  
in  
the  
ob-  
ject  
and  
pop-  
u-  
late  
a  
dict.

with the node uuid

The node is fetched with the object node attribute and the dict node value is populated with the latest

Parameters

- obj** object to get the node attribute
- to\_dict** to populate with a node value

Raises exception if

the  
 node  
 is  
 not  
 found  
  
 ironic.  
 Re-  
 plac  
 nod  
 dict  
 valu  
 with  
 nod  
  
 nod  
 is  
 found  
 by  
 fetch  
 ing  
 the  
 node  
 by  
 id  
 look

**Paramet**  
**to\_**  
 Dict  
 to  
 set  
 nod  
 valu  
 on

**Returns**  
 The  
 node  
 ob-  
 ject  
 from  
 the  
 look

**Raises**  
 Nod  
 Not-  
 Four  
 with  
 sta-  
 tus\_  
 set  
 to

400  
 BAI  
 whe  
 node  
 is  
 not  
 foun  
 ironic.  
 Re-  
 plac  
 nod  
 dict  
 valu  
 with  
 nod  
 nod  
 is  
 foun  
 by  
 fetch  
 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-  
 Foun  
 with  
 sta-  
 tus\_

set  
 to  
 400  
 BAI  
 whe  
 node  
 is  
 not  
 foun

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  
 (plu  
 the  
 lin  
 field

**Paramet**

- to\_  
 dict  
 to  
 san-  
 i-  
 tize
-

```

file
(list
  of
  strings
  list
  of
  field
  to
  pre-
  serv
  or
  Non
  to
  pre-
  serv
  then
  all
  ironic.

  ironic.

  ironic.

  Call
  a
  ven-
  dor
  pass
  API
  ex-
  ten-
  sion

  Call
  the
  ven-
  dor
  pass
  API
  ex-
  ten-
  sion
  and
  pro-
  cess
  the
  meth
  re-
  spon

```



return code for methods that are asynchronous or synchronous; Attach the return value to the response object if its being served statically.

for drivers vendor passthru this is the drivers name.

to  
set  
the  
righ

## Paramet

- ide**  
The  
re-  
sour  
iden  
ti-  
fi-  
ca-  
tion.  
For  
node  
ven-  
dor  
pass  
this  
is  
the  
node  
UU
- met**  
The  
ven-  
dor  
meth  
nam
- top**  
The  
RPC  
topi
- dat**  
The  
data  
pass  
to  
the  
ven-

dor  
 meth  
 De-  
 fault  
 to  
 Non

- **dri**  
 Boo  
 valu  
 Whe  
 this  
 is  
 a  
 node  
 or  
 drive  
 ven-  
 dor  
 pass  
 De-  
 fault  
 to  
 Fals

**Returns**  
 A  
 WSI  
 re-  
 spor  
 ob-  
 ject  
 to  
 be  
 re-  
 turn  
 by  
 the  
 API

ironic.api.controllers.v1.versions module

ironic.  
 Re-  
 turn  
 the  
 max  
 i-  
 mun  
 sup-

wise, it is the maximum supported API version.

port  
API  
ver-  
sion  
(as  
a  
strin  
  
If  
the  
ser-  
vice  
is  
pinn  
the  
max  
i-  
mun  
API  
ver-  
sion  
is  
the  
pinn  
ver-  
sion  
Oth-  
er-

ironic.  
Re-  
turn  
the  
min-  
i-  
mun  
sup-  
port  
API  
ver-  
sion  
(as  
a  
strin

## ironic.api.controllers.v1.volume module

**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**  
UI  
of  
a  
vol-  
ume  
con-  
nec-  
tor.

**Raises**

Op-  
er-  
a-  
tion)  
Per-  
mit-  
ted  
if  
ac-  
cess  
with  
spec  
i-  
fy-  
ing  
a  
par-  
ent  
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-  
 nec-  
 tor  
 does  
 not  
 ex-  
 ist

**Raises**  
 Vol-  
 ume  
 Con-  
 nec-  
 torN  
 Four  
 if  
 the  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 can-  
 not  
 be  
 foun

**Raises**  
 In-  
 valid  
 State  
 If  
 a  
 node  
 as-  
 so-  
 ci-  
 ated  
 with  
 the  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 is

powered off.

not

**get\_all**

Re-  
triev  
a  
list  
of  
vol-  
ume  
con-  
nec-  
tors.

**Parame**

- **node**  
UUID  
or  
nam  
of  
a  
node  
to  
get  
only  
vol-  
ume  
con-  
nec-  
tors  
for  
that  
node

- **max**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

- **lim**  
max

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironic configuration, or only `max_limit` resources will be returned.

i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to  
 re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult.  
 This  
  
 •  
**sort**  
 col-  
 umn  
 to  
 sort  
 re-  
 sults  
 by.  
 De-  
 fault  
 id.  
  
 •  
**sort**  
 di-  
 rec-  
 tion  
 to  
 sort.  
 asc  
 or  
 desc  
 De-  
 fault  
 asc.  
  
 •  
**file**  
 Op-  
 tion:  
 a  
 list



returned.

is found.

with  
 a  
 spec  
 i-  
 fied  
 set  
 of  
 field  
 of  
 the  
 re-  
 sour  
 to  
 be

- **det**  
 Op-  
 tion:  
 whe  
 to  
 re-  
 triev  
 with  
 de-  
 tail.

Returns

a  
 list  
 of  
 vol-  
 ume  
 con-  
 nec-  
 tors,  
 or  
 an  
 emp  
 list  
 if  
 no  
 vol-  
 ume  
 con-  
 nec-  
 tor

Raises

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  
 in-  
 valid  
 for  
 sort-  
 ing.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 both  
 field  
 and  
 de-  
 tail  
 are  
 spec  
 i-  
 fied.

get\_one

Re-  
 triev  
 in-

for-  
ma-  
tion  
about  
the  
give  
vol-  
ume  
con-  
nec-  
tor.

Paramete

- **con**  
UI  
of  
a  
vol-  
ume  
con-  
nec-  
tor.
- **file**  
Op-  
tiona  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field  
of  
the  
re-  
sour  
to  
be

returned.

Returns

API  
serial  
vol-  
ume

fied UUID.

con-  
 nec-  
 tor  
 ob-  
 ject.  
**Raises**  
 Op-  
 er-  
 a-  
 tion  
 Per-  
 mit-  
 ted  
 if  
 ac-  
 cess  
 with  
 spec  
 i-  
 fy-  
 ing  
 a  
 par-  
 ent  
 node  
**Raises**  
 Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 if  
 no  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 ex-  
 ists  
 with  
 the  
 spec  
 i-  
 invalid  
**patch** (o

Up-  
date  
an  
ex-  
ist-  
ing  
vol-  
ume  
con-  
nec-  
tor.

Paramete

- **con**  
UI  
of  
a  
vol-  
ume  
con-  
nec-  
tor.
- **pat**  
a  
json  
PAT  
doc-  
u-  
men  
to  
ap-  
ply  
to  
this  
vol-  
ume  
con-  
nec-  
tor.

Returns

API  
serial  
vol-  
ume  
con-  
nec-  
tor

ob-  
 ject.

**Raises**

Op-  
 er-  
 a-  
 tionl  
 Per-  
 mit-  
 ted  
 if  
 ac-  
 cess  
 with  
 spec  
 i-  
 fy-  
 ing  
 a  
 par-  
 ent  
 node

**Raises**

Patc  
 ror  
 if  
 a  
 give  
 patc  
 can  
 not  
 be  
 ap-  
 plie

**Raises**

Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 if  
 no  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 ex-  
 ists

fied UUID.

with  
the  
spec  
i-

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
vol-  
ume  
con-  
nec-  
tors  
UUI  
is  
be-  
ing  
char

**Raises**

Nod  
Loc  
if  
node  
is  
lock  
by  
an-  
othe  
con-  
duc-  
tor

**Raises**

Nod  
Not-  
Foun  
if  
the  
node  
as-  
so-  
ci-  
ated  
with

exists with the same values for type and connector\_id fields

the  
 con-  
 nec-  
 tor  
 does  
 not  
 ex-  
 ist  
**Raises**  
 Vol-  
 ume  
 Con-  
 nec-  
 torT  
 pe-  
 An-  
 dI-  
 dAl-  
 read  
 ists  
 if  
 an-  
 othe  
 con-  
 nec-  
 tor  
 al-  
 read  
**Raises**  
 In-  
 valid  
 UID  
 if  
 in-  
 valid  
 node  
 UUI  
 is  
 pass  
 in  
 the  
 patc  
**Raises**  
 In-  
 valid  
 State  
 If  
 a  
 node



powered off.

as-  
 so-  
 ci-  
 ated  
 with  
 the  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 is  
 not

**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-

already exists with the same type and connector\_id

er-  
a-  
tionl  
Per-  
mit-  
ted  
if  
ac-  
cess  
with  
spec  
i-  
fy-  
ing  
a  
par-  
ent  
node  
  
**Raises**  
Vol-  
ume  
Con  
nec-  
torT  
pe-  
An-  
dI-  
dAl-  
read  
ists  
if  
a  
vol-  
ume  
con-  
nec-  
tor  
al-  
  
**Raises**  
Vol-  
ume  
Con  
nec-  
torA  
read  
ists  
if  
a  
vol-

ready exists

ume  
con-  
nec-  
tor  
with  
the  
sam  
UUI  
al-

ironic.

ironic.

## ironic.api.controllers.v1.volume\_target module

**class** i

Base  
pec  
res  
Res  
RES  
con-  
troll  
for  
Vol-  
ume  
gets

**delete**

Dele  
a  
vol-  
ume  
tar-  
get.

**Parame**  
**tar**

UUI  
 of  
 a  
 vol-  
 ume  
 tar-  
 get.

**Raises**

Op-  
 er-  
 a-  
 tionl  
 Per-  
 mit-  
 ted  
 if  
 ac-  
 cess  
 with  
 spec  
 i-  
 fy-  
 ing  
 a  
 par-  
 ent  
 node

**Raises**

Nod  
 Loc  
 if  
 node  
 is  
 lock  
 by  
 an-  
 othe  
 con-  
 duc-  
 tor

**Raises**

Nod  
 Not-  
 Four  
 if  
 the  
 node  
 as-  
 so-  
 ci-

ered off.

ated  
 with  
 the  
 tar-  
 get  
 does  
 not  
 ex-  
 ist  
 Raises  
 Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 the  
 vol-  
 ume  
 tar-  
 get  
 can-  
 not  
 be  
 foun  
 Raises  
 In-  
 valid  
 State  
 If  
 a  
 node  
 as-  
 so-  
 ci-  
 ated  
 with  
 the  
 vol-  
 ume  
 tar-  
 get  
 is  
 not  
 pow  
 get\_all  
 Re-  
 triev

a  
 list  
 of  
 vol-  
 ume  
 tar-  
 gets

Paramete

- **node**  
 UUI  
 or  
 nam  
 of  
 a  
 node  
 to  
 get  
 only  
 vol-  
 ume  
 tar-  
 gets  
 for  
 that  
 node
- **max**  
 pag-  
 i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets.
- **lim**  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to

value cannot be larger than the value of `max_limit` in the `[api]` section of the ironiC configuration, or only `max_limit` resources will be returned.

re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.  
This

- **sort**  
col-  
umn  
to  
sort  
re-  
sults  
by.  
De-  
fault  
id.
- **sort**  
di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc  
De-  
fault  
asc.
- **fields**  
Op-  
tiona  
a  
list  
with  
a  
spec  
i-  
fied  
set  
of  
field

returned.

of  
the  
re-  
sour  
to  
be

- **det**  
Op-  
tion:  
whe  
to  
re-  
triev  
with  
de-  
tail.

**Returns**

a  
list  
of  
vol-  
ume  
tar-  
gets  
or  
an  
emp  
list  
if  
no  
vol-  
ume  
tar-  
get  
is  
foun

**Raises**

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  
in-  
valid  
for  
sort-  
ing.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
both  
field  
and  
de-  
tail  
are  
spec  
i-  
fied.

**get\_one**

Re-  
triev  
in-  
for-  
ma-  
tion  
about  
the  
give  
vol-  
ume  
tar-

returned.

get.

Parameters

- tar

The tarball name of the volume to get.
- file

Optional filename of the file to get from the tarball. If not specified, the file will be returned.

Returns

A tarball object containing the serialized volume data.

Raises

An exception if the operation fails.

ted  
 if  
 ac-  
 cess  
 with  
 spec  
 i-  
 fy-  
 ing  
 a  
 par-  
 ent  
 node

Raises

Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 no  
 vol-  
 ume  
 tar-  
 get  
 with  
 this  
 UUI  
 ex-  
 ists

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  
 tar-  
 get.

**Returns**

API  
 seria  
 vol-  
 ume  
 tar-  
 get  
 ob-  
 ject.

**Raises**

Op-  
 er-  
 a-  
 tion  
 Per-  
 mit-  
 ted  
 if  
 ac-  
 cess  
 with  
 spec  
 i-  
 fy-  
 ing  
 a  
 par-  
 ent

node

**Raises**

Patch  
 error  
 if  
 a  
 give  
 patch  
 can  
 not  
 be  
 ap-  
 plied

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 vol-  
 ume  
 tar-  
 gets  
 UUI  
 is  
 be-  
 ing  
 char

**Raises**

Node  
 Lock  
 if  
 the  
 node  
 is  
 al-  
 read  
 lock

**Raises**

Node  
 Not-  
 Four  
 if  
 the  
 node

ist

as-  
so-  
ci-  
ated  
with  
the  
vol-  
ume  
tar-  
get  
does  
not  
ex-

Raises

Vol-  
ume  
get-  
Not-  
Four  
if  
the  
vol-  
ume  
tar-  
get  
can-  
not  
be  
foun

Raises

Vol-  
ume  
get-  
Boo  
dex-  
Al-  
read  
ists  
if  
a  
vol-  
ume  
tar-  
get  
al-  
read  
ex-  
ists  
with

the same node ID and boot index values

ered off.

#### Raises

In-  
valid  
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-  
ume  
tar-  
get  
is  
not  
pow

#### **post** (*tar*

Cre-  
ate  
a  
new  
vol-  
ume  
tar-  
get.

#### Parame

**tar**  
a  
vol-

ume  
tar-  
get  
with  
the  
re-  
ques  
body

Returns

API  
serial  
vol-  
ume  
tar-  
get  
ob-  
ject.

Raises

Op-  
er-  
a-  
tionl  
Per-  
mit-  
ted  
if  
ac-  
cess  
with  
spec  
i-  
fy-  
ing  
a  
par-  
ent  
node

Raises

Vol-  
ume  
get-  
Boo  
dex-  
Al-  
read  
ists  
if  
a  
vol-  
ume



the same node ID and boot index

tar-  
get  
al-  
read  
ex-  
ists  
with

Raises

Vol-  
ume  
ge-  
tAl-  
read  
ists  
if  
a  
vol-  
ume  
tar-  
get  
with  
the  
sam  
UUI  
ex-  
ists

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

**class** i  
  
Base  
obj  
  
API  
Ver-  
sion  
ob-  
ject.

**max\_str**  
HTT  
re-  
spor  
head

**min\_str**  
HTT  
re-

spon  
 head  
**static**  
 De-  
 ter-  
 mine  
 the  
 API  
 ver-  
 sion  
 re-  
 ques  
 base  
 on  
 the  
 head  
 ers  
 sup-  
 plie

Param

- **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**

HT  
 Hea  
 strin  
 car-  
 ry-  
 ing  
 the  
 re-  
 ques  
 ver-  
 sion

ionic.api.controllers.link module

ionic.

ionic.

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
    cur-
    rent
    de-
    fault
    ver-
    sion

    id:
    The
    ID
    of
    the
    (ma-
    jor)
    ver-
    sion
    also
    acts
    as
    the
    re-
    lease
    num
    ber
```

containing one link that points to the current version of the API

links  
A  
list

sta-  
tus:  
Sta-  
tus  
of  
the  
ver-  
sion  
one  
of  
CUR  
REN  
SUP  
POR  
DEF  
RE-  
CAT

min.  
The  
cur-  
rent,  
max  
i-  
mun  
sup-  
port  
(ma-  
jor.n  
ver-  
sion  
of  
API

ver-  
sion  
Min  
i-  
mun  
sup-  
port  
(ma-  
jor.n  
ver-  
sion  
of  
API



## ironic.api.middleware.json\_ext module

```
class i
    Base
    obj
    Sim
    pli-
    fied
    pro-
    cess
    ing
    of
    .json
    ex-
    ten-
    sion

    Pre-
    vi-
    ously
    Iron
    API
    used
    the
    gues
    fea-
    ture.
    It
    was
    neve
    need
    as
    we
    neve
    al-
    lowe
    non-
```

JSON content types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

## ironic.api.middleware.parsable\_error module

```
Mid
dle-
ware
to
re-
plac
the
plain
```



formatted so the client can parse it.

## Module contents

routes in the API.

mid-  
dle-  
ware  
  
Doe  
not  
per-  
form  
ver-  
i-  
fi-  
ca-  
tion  
of  
au-  
then-  
ti-  
ca-  
tion  
to-  
kens  
for  
pub-  
lic  
  
**class** i  
Base  
obj  
  
Sim  
pli-  
fied  
pro-  
cess  
ing  
of  
.json  
ex-  
ten-  
sion  
  
Pre-  
vi-  
ousl  
Iron  
API  
used  
the  
gues  
fea-  
ture.  
It

JSON content types anyway. Now that it is removed, this middleware strips .json extension for backward compatibility.

## Submodules

**ironic.api.app module**

was  
neve  
need  
as  
we  
neve  
al-  
lowe  
non-

```
class i
Base
obj
Re-
plac
er-
ror
body
with
som
thing
the
clien
can
pars
```

```
class i
Base
osl
cor
COR
Iron
spec
COR
class
Wer
addi
the
Iron
spec
ver-
sion
head
```

that a request bearing those headers might be accepted by the Ironiic REST API.

**ironiic.api.config module**

**ironiic.api.functions module**

ers  
to  
the  
list  
of  
sim-  
ple  
head  
ers  
in  
or-  
der

**simple\_**

**class i**  
Base  
obj  
ironiic.  
  
ironiic.

**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

**mandat**
 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  
 sin-  
 gle  
 ar-  
 gu-  
 men  
 its  
 type  
**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

function and not raise UnknownArgument exceptions

*Fun*  
 from  
 its  
 nam  
**ignore\_**  
 True  
 if  
 ex-  
 tra  
 ar-  
 gu-  
 men  
 shou  
 be  
 ig-  
 nore  
 NOT  
 in-  
 serte  
 in  
 the  
 kwa  
 of  
 the

**name**  
 Func  
 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  
     the  
     ar-  
     gu-  
     men  
     type  
     of  
     an  
     ex-  
     pose  
     func  
     tion.

Paramet

- **ret**  
 Type  
 of  
 the  
 valu  
 re-  
 turn  
 by  
 the  
 func  
 tion
- **arg**  
 Type  
 of  
 the



request body by itself, its type.

Nth  
ar-  
gu-  
men

- **bod**  
If  
the  
func  
tion  
take  
a  
fi-  
nal  
ar-  
gu-  
men  
that  
is  
sup-  
pose  
to  
be  
the
- **sta**  
HTT  
re-  
turn  
sta-  
tus  
code  
of  
the  
func  
tion.
- **ign**  
Al-  
low  
ex-  
tra/u  
ar-  
gu-  
men  
(de-  
fault  
to  
Fals

you are not using WSME on top of another framework.

rator, either a new decorator named `@wsexpose` that takes the same parameters (it will in addition expose the function, hence its name).

Mos  
of  
the  
time  
this  
dec-  
o-  
ra-  
tor  
is  
not  
sup-  
pose  
to  
be  
used  
di-  
rectl  
un-  
less

If  
an  
adap  
is  
used  
it  
will  
pro-  
vide  
ei-  
ther  
a  
spe-  
ciali  
ver-  
sion  
of  
this  
deco  
ro-

ironic.

## ironic.api.hooks module

```
class i
    Base
    peo
    hoo
    Peo

    At-
    tach
    the
    con-
    fig
    ob-
    ject
    to
    the
    re-
    ques
    so
    con-
    troll
    can
    get
    to
    it.
```

```
before
    Ove
    ride
    this
    meth
    to
    cre-
    ate
    a
    hool
    that
    gets
    calle
    af-
    ter
    rout
    ing,
    but
    be-
    fore
```

the request gets passed to your controller.

```
Paramet
sta
```

The  
 Peca  
 sta  
 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  
 re-  
 ques  
**after** (s  
 Ove  
 ride  
 this  
 meth  
 to  
 cre-  
 ate  
 a  
 hool  
 that  
 gets  
 calle  
 af-  
 ter  
 the  
 re-

handled by the controller.

the request gets passed to your controller.

ques  
has  
been

**Parame**  
**sta**  
The  
Peca  
sta  
ob-  
ject  
for  
the  
cur-  
rent  
re-  
ques

**before**  
Ove  
ride  
this  
meth  
to  
cre-  
ate  
a  
hool  
that  
gets  
called  
af-  
ter  
rout  
ing,  
but  
be-  
fore

**Parame**  
**sta**  
The  
Peca  
sta  
ob-  
ject  
for  
the  
cur-  
rent  
re-

the request gets passed to your controller.

ques  
**class** i  
 Base  
 pec  
 hoo  
 Pec  
 At-  
 tach  
 the  
 dbap  
 ob-  
 ject  
 to  
 the  
 re-  
 ques  
 so  
 con-  
 troll  
 can  
 get  
 to  
 it.  
**before**  
 Ove  
 ride  
 this  
 meth  
 to  
 cre-  
 ate  
 a  
 hool  
 that  
 gets  
 calle  
 af-  
 ter  
 rout  
 ing,  
 but  
 be-  
 fore  
**Parame**  
**sta**  
 The  
 Peca

sent to the client. Such behavior is a security concern so this hook is aimed to cut-off traceback from the error message.

sta  
 ob-  
 ject  
 for  
 the  
 cur-  
 rent  
 re-  
 ques  
  
**class** i  
 Bas  
 pec  
 hoo  
 Pec  
  
 Wor  
 rpc.  
 de-  
 se-  
 ri-  
 al-  
 ize\_  
  
 de-  
 se-  
 ri-  
 al-  
 ize\_  
 build  
 rpc  
 ex-  
 cep-  
 tion  
 trace  
 back  
 into  
 er-  
 ror  
 mes  
 sage  
 whic  
 is  
 then  
  
**after** (s  
 Ove  
 ride  
 this  
 meth  
 to

handled by the controller.

cre-  
ate  
a  
hool  
that  
gets  
calle  
af-  
ter  
the  
re-  
ques  
has  
been

Paramete  
**sta**  
The  
Peca  
sta  
ob-  
ject  
for  
the  
cur-  
rent  
re-  
ques

**class i**  
Base  
pec  
hoo  
Pec  
  
At-  
tach  
the  
righ  
pub-  
lic\_u  
to  
the  
re-  
ques  
  
At-  
tach  
the  
righ  
pub-  
lic\_u



API service is behind a proxy or SSL terminator.

the request gets passed to your controller.

to  
the  
re-  
ques  
so  
re-  
sour  
can  
cre-  
ate  
links  
ever  
whe  
the

**before**

Ove  
ride  
this  
meth  
to  
cre-  
ate  
a  
hool  
that  
gets  
call  
af-  
ter  
rout  
ing,  
but  
be-  
fore

**Parame**

**sta**  
The  
Peca  
sta  
ob-  
ject  
for  
the  
cur-  
rent  
re-  
ques

**class i**

the request gets passed to your controller.

Base  
pec  
hoo  
Peo

At-  
tach  
the  
rp-  
capi  
ob-  
ject  
to  
the  
re-  
ques  
so  
con-  
troll  
can  
get  
to  
it.

#### before

Ove  
ride  
this  
meth  
to  
cre-  
ate  
a  
hool  
that  
gets  
calle  
af-  
ter  
rout  
ing,  
but  
be-  
fore

#### Parame

sta  
The  
Peca  
sta  
ob-

ject  
for  
the  
cur-  
rent  
re-  
ques  
  
ironic.

## ironic.api.method module

ironic.  
Dec  
o-  
ra-  
tor  
whic  
plac  
HTT  
re-  
ques  
body  
JSO  
into  
a  
meth  
ar-  
gu-  
men

### Parameter

**body**  
Nam  
of  
ar-  
gu-  
men  
to  
pop-  
u-  
late  
with  
body  
JSO

ironic.

ironic.  
Ex-

tract  
 in-  
 for-  
 ma-  
 tion:  
 that  
 can  
 be  
 sent  
 to  
 the  
 client

ironic.api.wsgi module

WSGI  
 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**  
Chec  
the  
ver-  
sion  
of  
ob-  
jects

Che  
that  
the  
ob-  
ject  
ver-  
sion  
are

this by comparing the objects .version field in the database, with the expected versions of these objects.

com  
 pat-  
 i-  
 ble  
 with  
 this  
 re-  
 lease  
 of  
 iron.  
 It  
 does  
  
 Re-  
 turn  
 Non  
 if  
 com  
 pat-  
 i-  
 ble;  
 a  
 strin  
 de-  
 scrib  
 ing  
 the  
 is-  
 sue  
 oth-  
 er-  
 wise  
  
**create\_**  
  
**online\_**  
  
**revisio**  
  
**stamp (**  
  
**upgrade**  
  
**version**  
  
 ironic.

ironic.

ironic.cmd.status module

**class** i
 Base
 osl
 upg
 Upg
 Up-
 grad
 chec
 for
 the
 iron
 statu
 up-
 grad
 chec
 com
 man
 Up-
 grad
 chec
 shou
 be
 adde
 as
 sep-
 a-
 rate
 meth
 ods
 in
 this
 class
 and
 adde
 to
 \_up-
 grad

tuple.

ironic.

## Module contents

`ironic.common` package

## Subpackages

`ironic.common.glance_service` package

## Submodules

`ironic.common.glance_service.image_service` module

**class** `i`

Base  
obj

**call** (*m*)

Call  
a  
glan  
clien  
meth

If  
we  
get  
a  
con-  
nec-  
tion  
er-  
ror,  
retry  
the  
re-  
ques  
ac-  
cord  
ing  
to  
CON

**Parame**

- **con**  
The



re-ques-  
con-text,  
for ac-cess  
checked

**met**  
The meth-  
re-ques-  
to be calle

**arg**  
A list of po-si-tiona-  
ar-gu-men-  
for the meth-  
calle

**kwa**  
A dict of key-  
wor-  
ar-gu-men-  
for the meth-  
calle

**Raises**  
Glar  
Con

nec-  
tion-  
Failu

**download**  
Call  
out  
to  
Glan  
for  
data  
and  
write  
data

**Parameter**

- image**  
The  
opac  
im-  
age  
iden  
ti-  
fier.
- data**  
(Op-  
tiona  
File  
ob-  
ject  
to  
write  
data  
to.

**show** (*image*)  
Re-  
turn  
a  
dict  
with  
im-  
age  
data  
for  
the  
give  
opac  
im-

age  
 id.

**Parameters**  
**image\_id**  
 The  
 opaque  
 identifier  
 of the  
 image  
 to be  
 fetched.

**Returns**  
 A  
 dictionary  
 containing  
 information  
 about the  
 image  
 metadata  
 and data.

**Raises**  
 ImageNotFoundError  
 if the  
 image  
 is not  
 found.

**Raises**  
 ImageNotAvailableError  
 if the  
 image  
 is not  
 available  
 for  
 the  
 image  
 status  
 is not  
 active.

**swift\_token**  
 Generate  
 a  
 no-auth  
 Swift

the temporary Swift URL using the image id from Glance and the config options: `swift_endpoint_url`, `swift_api_version`, `swift_account` and `swift_container`. The temporary URL will be valid for `swift_temp_url_duration` seconds. This allows IroniC to download a Glance image without passing around an `auth_token`.

a dictionary, with keys like `name` and `checksum`. See <https://docs.openstack.org/glance/latest/user/glanceapi.html> for examples.

tem-  
po-  
rary  
URL  
  
This  
func  
tion  
will  
gen-  
er-  
ate  
(or  
re-  
turn  
the  
cach  
one  
if  
temp  
URL  
cach  
is  
en-  
able

#### **Parame**

**ima**  
The  
re-  
turn  
from  
a  
GET  
re-  
ques  
to  
Glan  
for  
a  
cer-  
tain  
im-  
age.  
Sho  
be

tion.

Returns
 A
 sign
 Swi
 URL
 from
 whic
 an
 im-
 age
 can
 be
 dow
 load
 with
 out
 au-
 then
 ti-
 ca-

Raises
 In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 Swi
 con-
 fig
 op-
 tions
 are
 not
 set
 cor-
 rectl

Raises
 Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 a

re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 not  
 set.

**Raises**

Imag  
 U-  
 nac-  
 cept  
 able  
 if  
 the  
 im-  
 age  
 info  
 from  
 Glar  
 does  
 not  
 have  
 an  
 im-  
 age  
 ID.

**class i**

Base  
 tup

**url**

Alia  
 for  
 field  
 num  
 ber  
 0

**url\_exp**

Alia  
 for  
 field  
 num  
 ber  
 1

ironic.

Cre-  
ates  
a  
glan  
clie  
if  
does  
ex-  
ists  
and  
calls  
the  
func  
tion.

## ironic.common.glance\_service.service\_utils module

ironic.

ironic.

Che  
the  
im-  
age  
sta-  
tus.

This  
chec  
is  
need  
in  
case  
the  
Glar  
im-  
age  
is  
stuc  
in  
queu  
sta-  
tus  
or  
penc  
ing\_

ironic.

Che

on.

im-  
 age  
 avai  
 abil-  
 ity.  
 This  
 chec  
 is  
 need  
 in  
 case  
 Nov  
 and  
 Glar  
 are  
 de-  
 ploy  
 with  
 out  
 au-  
 then  
 ti-  
 ca-  
 tion  
 turn  
 ionic.  
 Pars  
 an  
 im-  
 age  
 id  
 from  
 im-  
 age  
 href  
 Paramet  
 ima  
 href  
 of  
 an  
 im-  
 age  
 Returns  
 im-  
 age  
 id  
 pars  
 from



im-  
age\_

#### Raises

*Inv*  
when  
in-  
put  
im-  
age  
href  
is  
in-  
valid

ironic.

## Module contents

`ironic.common.json_rpc` package

## Submodules

`ironic.common.json_rpc.client` module

A  
sim-  
ple  
JSON  
RPC  
client

This  
client  
is  
com-  
pat-  
i-  
ble  
with  
any  
JSON  
RPC  
2.0  
im-  
ple-  
men-  
ta-  
tion.

ours.

ironic.common.json\_rpc.server module

tors.

in-  
clud  
ing

**class** i  
Base  
obj  
  
JSO  
RPC  
clien  
with  
iron  
ex-  
cep-  
tion  
han-  
dling

**can\_ser**

**prepare**

Im-  
ple-  
men  
ta-  
tion  
of  
JSO  
RPC  
for  
com  
mu-  
ni-  
ca-  
tion  
be-  
twee  
API  
and  
con-  
duc-

This  
mod

[jsonrpc.org/specification](http://jsonrpc.org/specification). Main differences: \* No support for batched requests. \* No support for positional arguments passing. \* No JSON RPC 1.0 fallback.

ule  
im-  
ple-  
men  
a  
sub-  
set  
of  
JSO  
RPC  
2.0  
as  
de-  
fine  
in  
<http://>  
//  
www

**excepti**

Base  
*irc*  
*com*  
*jsc*  
*ser*  
*Jsc*

**code =**

**excepti**

Base  
*irc*  
*com*  
*jsc*  
*ser*  
*Jsc*

**code =**

**excepti**

Base  
iro  
exc  
Iro

**excepti**

Base  
*irc*  
*com*  
*jsc*  
*ser*  
*Jsc*

**code =**

**excepti**

Base  
*irc*  
*com*  
*jsc*  
*ser*  
*Jsc*

**code =**

**class i**

Base  
 osl  
 ser  
 Ser  
  
 Pro-  
 vide  
 abil-  
 ity  
 to  
 laun  
 JSO  
 RPC  
 as  
 a  
 WSO  
 ap-  
 pli-  
 ca-  
 tion.

**reset ( )**

Re-  
 set  
 serv  
 gree  
 pool

size  
 to  
 de-  
 fault

**Returns**  
 Non

**start ()**  
 Start  
 serv  
 ing  
 this  
 ser-  
 vice  
 us-  
 ing  
 load  
 con-  
 fig-  
 u-  
 ra-  
 tion.

**Returns**  
 Non

**stop ()**  
 Stop  
 serv  
 ing  
 this  
 API

**Returns**  
 Non

**wait ()**  
 Wait  
 for  
 the  
 ser-  
 vice  
 to  
 stop  
 serv  
 ing  
 this  
 API

**Returns**  
 Non

Module contents

Submodules

ironic.common.args module

ironic.

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.

Paramet

- **nam**  
 Nam  
 of  
 the  
 ar-

gu-  
men

•

**val**  
A  
valu

**Returns**

The  
valu  
trans  
form  
thro  
ev-  
ery  
sup-  
plied  
val-  
ida-  
tor

**Raises**

The  
er-  
ror  
from  
the  
first  
faile  
val-  
ida-  
tor

ironic.

Val-  
i-  
date  
that  
the  
valu  
is  
a  
strin  
rep-  
re-  
sent  
ing  
a  
bool

**Paramet**

- name**  
 Name of the argument

- value**  
 A string value

**Returns**  
 The bool representation of the value or None if value is None

**Raises**  
 InvalidParameterError if the value cannot be converted to a bool



ironic.  
 Re-  
 turn  
 a  
 val-  
 ida-  
 tor  
 func-  
 tion  
 whic  
 val-  
 i-  
 date  
 dict  
 field  
 Val-  
 ida-  
 tors  
 will  
 re-  
 plac  
 the  
 valu  
 with  
 the  
 val-  
 i-  
 da-  
 tion  
 re-  
 sult.  
 Any  
 dict  
 item  
 whic

has no validator is ignored. When a key is missing in the value then the corresponding validator will not be run.

Param  
 val-  
 ida-  
 tors  
 dict  
 whe  
 the  
 key  
 is  
 a  
 dict  
 key  
 to

a validator function to run on that value

Returns

`ionic.Validate`  
 A validator function that takes the value to validate and returns a boolean representing the validity of the value.

Parameters

- name**  
 Name of the argument.
- validator**  
 A

valu
 rep-
 re-
 sent
 ing
 an
 in-
 te-
 ger

**Returns**

The
 valu
 as
 an
 int,
 or
 Non
 if
 valu
 is
 Non

**Raises**

In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 the
 valu
 does
 not
 rep-
 re-
 sent
 an
 in-
 te-
 ger

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  
valu  
is  
Non

Raises

In-  
valic  
Pa-

ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 valu  
 is  
 not  
 a  
 valid  
 MA  
 ad-  
 dres

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  
 if  
 valu  
 is  
 Non

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 valu  
 is  
 not  
 a  
 valid  
 log-  
 i-  
 cal  
 nam

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-  
i-  
da-  
tion  
fails

`ironic.`

Val-  
i-  
date  
a  
patc  
API  
op-  
er-  
a-

tion  
 ironic.  
 Re-  
 turn  
 a  
 val-  
 ida-  
 tor  
 func-  
 tion  
 which  
 val-  
 i-  
 date  
 the  
 valu-  
 with  
 json  
 sche

**Param**

sche  
 dict  
 rep-  
 re-  
 sent  
 ing  
 json  
 sche  
 to  
 val-  
 i-  
 date  
 with

**Returns**

val-  
 ida-  
 tor  
 func-  
 tion  
 which  
 take  
 nam  
 and  
 valu  
 ar-  
 gu-  
 men

ironic.  
 Val-



i-  
date  
that  
the  
valu  
is  
a  
strin

#### Paramet

- **nam**  
Nam  
of  
the  
ar-  
gu-  
men

- **val**  
A  
strin  
valu

#### Returns

The  
strin  
valu  
or  
Non  
if  
valu  
is  
Non

#### Raises

In-  
valio  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
valu  
is  
not  
a  
strin

ironic.  
 Val-  
 i-  
 date  
 and  
 con-  
 vert  
 com  
 de-  
 lim-  
 ited  
 strin  
 to  
 a  
 list.

Paramet

- **nam**  
 Nam  
 of  
 the  
 ar-  
 gu-  
 men
- **val**  
 A  
 com  
 sep-  
 a-  
 rated  
 strin  
 of  
 val-  
 ues

Returns

A  
 list  
 of  
 uniq  
 val-  
 ues  
 (low  
 case  
 main  
 tain-  
 ing  
 the

is None

sam
 or-
 der,
 or
 Non
 if
 valu

**Raises**

In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 the
 valu
 is
 not
 a
 strin

ironi
 c.
 Re-
 turn
 a
 val-
 ida-
 tor
 func
 tion
 whic
 chec
 the
 valu
 is
 one
 of
 the
 type

**Param**

type
 one
 or
 mor
 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  
 UUI

Paramet

- **nam**  
 Nam  
 of  
 the  
 ar-  
 gu-  
 men-
- **val**  
 A  
 UUI  
 strin  
 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  
UUI

ironic.  
Val-  
i-  
date  
that  
the  
valu  
is  
a  
UUI  
or  
log-  
i-  
cal  
nam

**Paramet**

- **nam**  
Nam  
of  
the  
ar-  
gu-  
men
- **val**  
A

name

UUI  
 or  
 log-  
 i-  
 cal  
 nam  
 strin  
 valu  
**Returns**  
 The  
 valu  
 or  
 Non  
 if  
 valu  
 is  
 Non  
**Raises**  
 In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 valu  
 is  
 not  
 a  
 valic  
 UUI  
 or  
 log-  
 i-  
 cal  
 ironic.  
 Dec  
 o-  
 ra-  
 tor  
 whic  
 val-  
 i-  
 date  
 and  
 trans  
 form

## ironic.common.boot\_devices module

nate device.

find the documentation at: <http://linux.die.net/man/1/ipmitool>

func  
tion  
ar-  
gu-  
men

Map  
ping  
of  
boot  
de-  
vice  
used  
when  
re-  
ques  
ing  
the  
sys-  
tem  
to  
boot  
from  
an  
al-  
ter-

The  
op-  
tions  
pre-  
sent  
were  
base  
on  
the  
IP-  
MI-  
tool  
chas  
sis  
boot  
dev  
com  
man  
You  
can

make sense in the limited context of IroniC right now.

NOT  
This  
mod  
ule  
does  
not  
in-  
clud  
all  
the  
op-  
tions  
from  
ip-  
mi-  
tool  
be-  
caus  
they  
dont

`ironic.`  
    **Boo**  
    into  
    BIO  
    setu

`ironic.`  
    **Boo**  
    from  
    CD/

`ironic.`  
    **Boo**  
    from  
    de-  
    fault  
    Harc  
    drive

`ironic.`  
    **Boo**  
    from  
    a  
    flopp  
    drive

`ironic.`  
    **Boo**  
    from  
    iSCS  
    vol-  
    ume



ironic.  
 Boo  
 from  
 PXE  
 boot

ironic.  
 Boo  
 from  
 de-  
 fault  
 Har  
 drive  
 re-  
 ques  
 Safe  
 Mod

ironic.  
 Boo  
 from  
 Wid  
 Area  
 Net-  
 worl

## ironic.common.boot\_modes module

tive firmware interfaces.

Map  
 ping  
 of  
 boot  
 mod  
 used  
 whe  
 re-  
 ques  
 ing  
 the  
 sys-  
 tem  
 to  
 boot  
 us-  
 ing  
 al-  
 ter-  
 na-

The  
 op-

ically on the `BootSourceOverrideMode` property.

tions.  
pre-  
sent  
were  
base  
on  
the  
Red  
fish  
pro-  
to-  
col  
ca-  
pa-  
bil-  
i-  
ties,  
spec

`ironic.`  
Boo  
over  
lega  
PC  
BIO  
firm  
in-  
ter-  
face

`ironic.`  
Boo  
over  
Uni-  
fied  
Ex-  
ten-  
si-  
ble  
Firm  
In-  
ter-  
face  
(UE  
firm  
in-  
ter-  
face

## ironic.common.cinder module

```

ironic.
    At-
    tach
    vol-
    ume
    to
    a
    node

    Enum-
    mer-
    ate
    thro
    the
    pro-
    vide
    list
    of
    vol-
    ume
    and
    at-
    tach
    the
    vol-
    ume
    to
    the
    node

```

defined in the task utilizing the provided connector information.

```

If
an
at-
tach
men
ap-
pear
to
al-
read
ex-
ist,
we
will
skip
at-
temp
ing
to

```

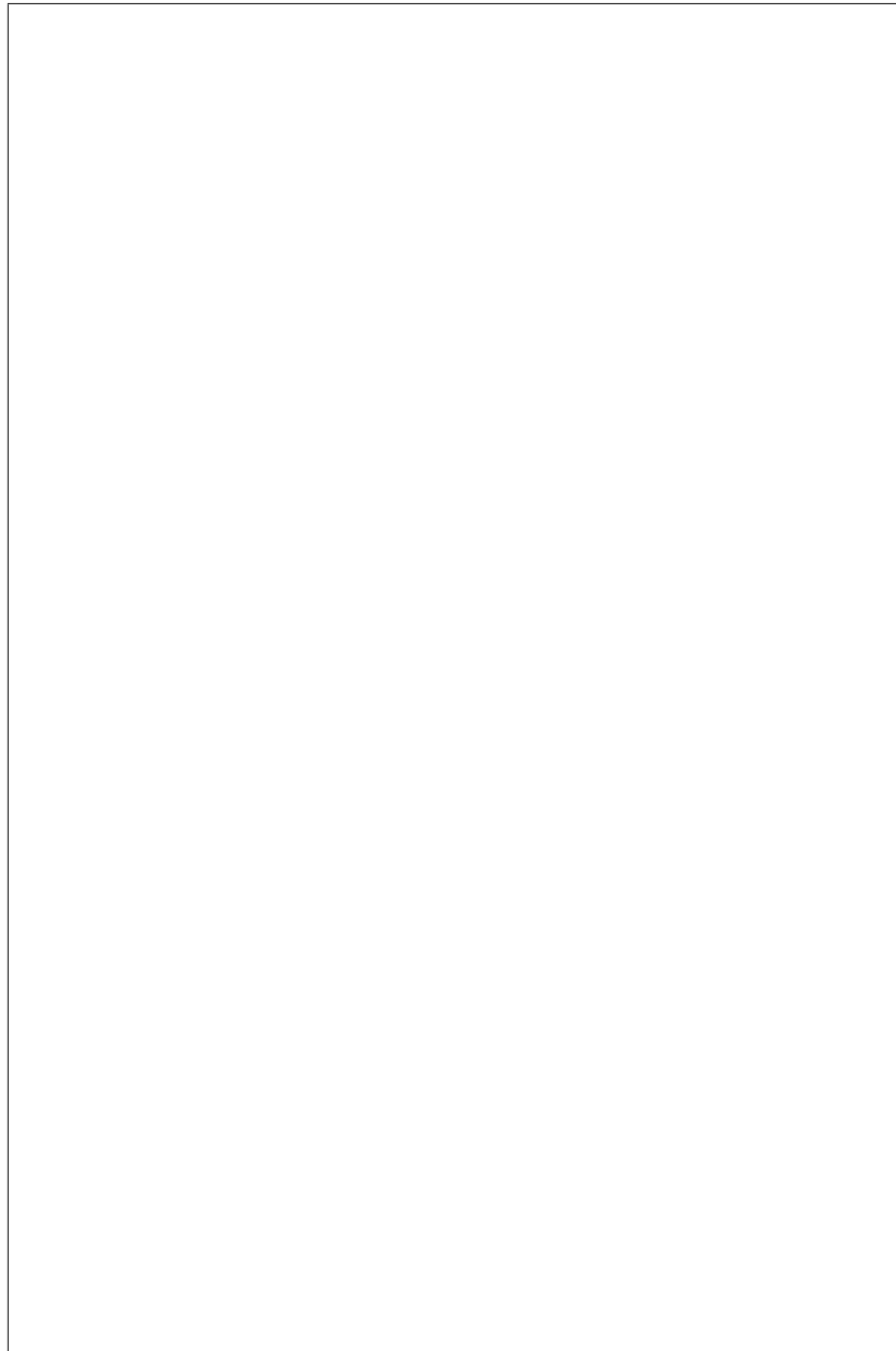
tach the volume. If use of the volume fails, a user may need to remove any lingering pre-existing/unused attachment records since we have no way to validate if the connector profile data differs from what was provided to cinder.

Paramet

- tas
 Task
 ager
 in-
 stan
 rep-
 re-
 sent
 ing
 the
 op-
 er-
 a-
 tion.
- vol
 List
 of
 vol-
 ume
 UUI
 val-
 ues
 rep-
 re-
 sent
 ing
 vol-
 ume
- con
 Dic-
 tio-
 nary
 ob-
 ject
 rep-
 re-
 sent
 ing
 the
 node

suf-  
fi-  
cien  
to  
at-  
tach  
a

volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected by cinder drivers. For FiberChannel, the key wwpns can be used with a list of port addresses. Some drivers support a multipath boolean key, although it is generally False. The host key is generally used for logging by drivers. Example:



(continues on next page)

(continued from previous page)

--

Raises

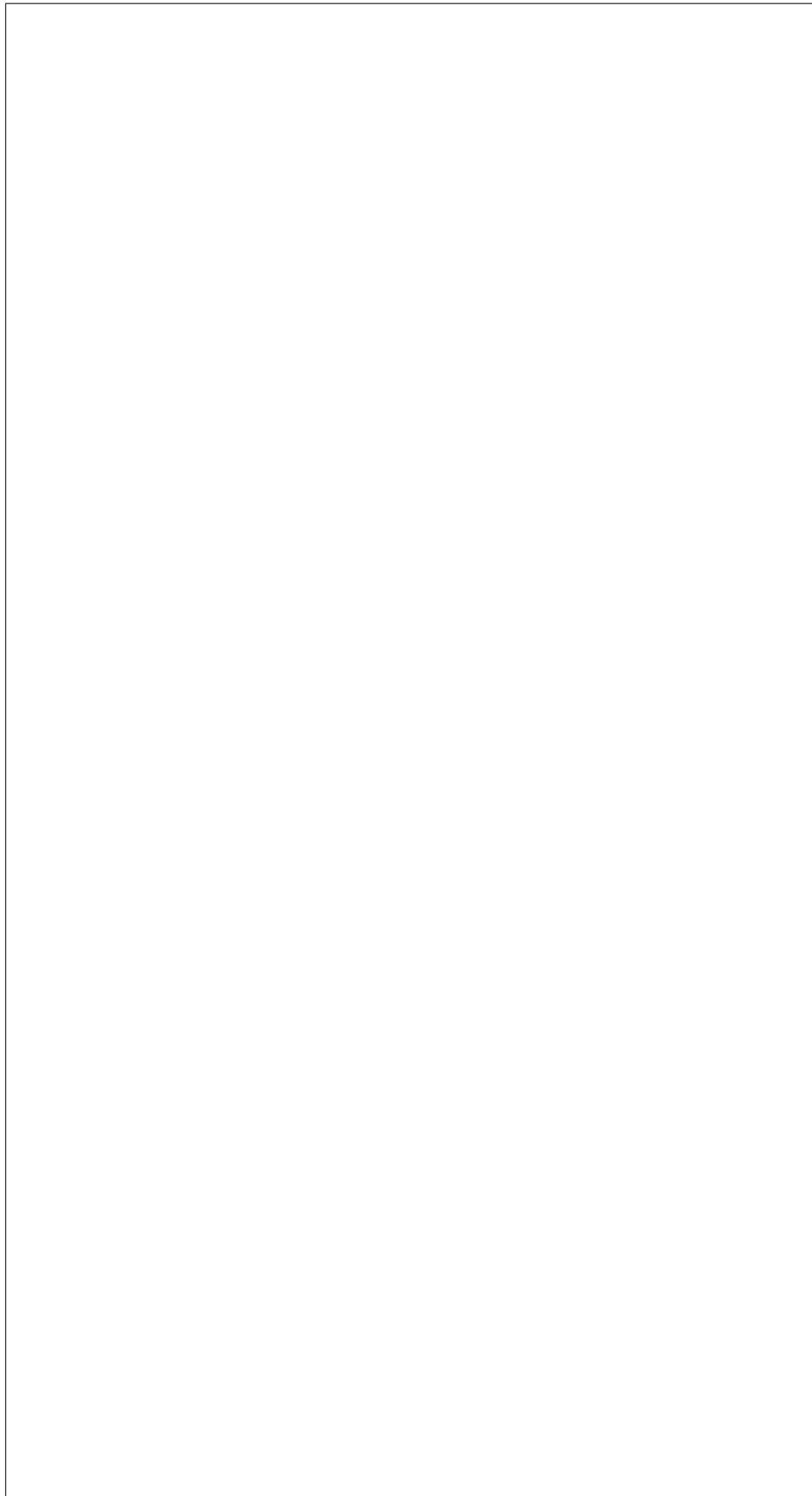
StorageError  
 If storage subsystem raises exception is raised

Returns

List of connection volume information including volume ID that were already read connection to desired

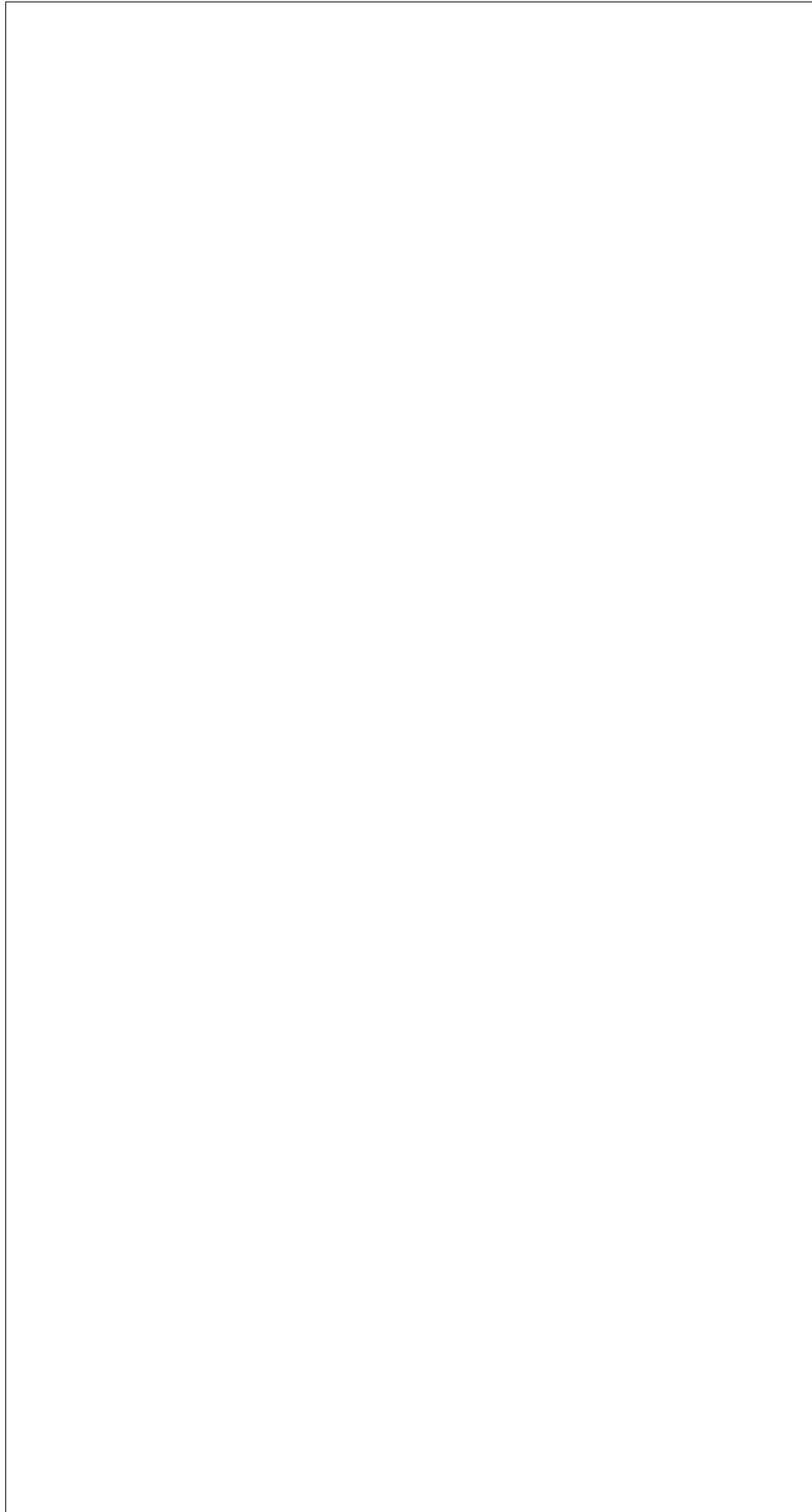
nodes. The returned list can be relatively consistent depending on the end storage driver that the volume is configured for, however the driver\_volume\_type key should not be relied upon as it is a free-form value returned by the driver. The accompanying data key contains the actual target details which 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)

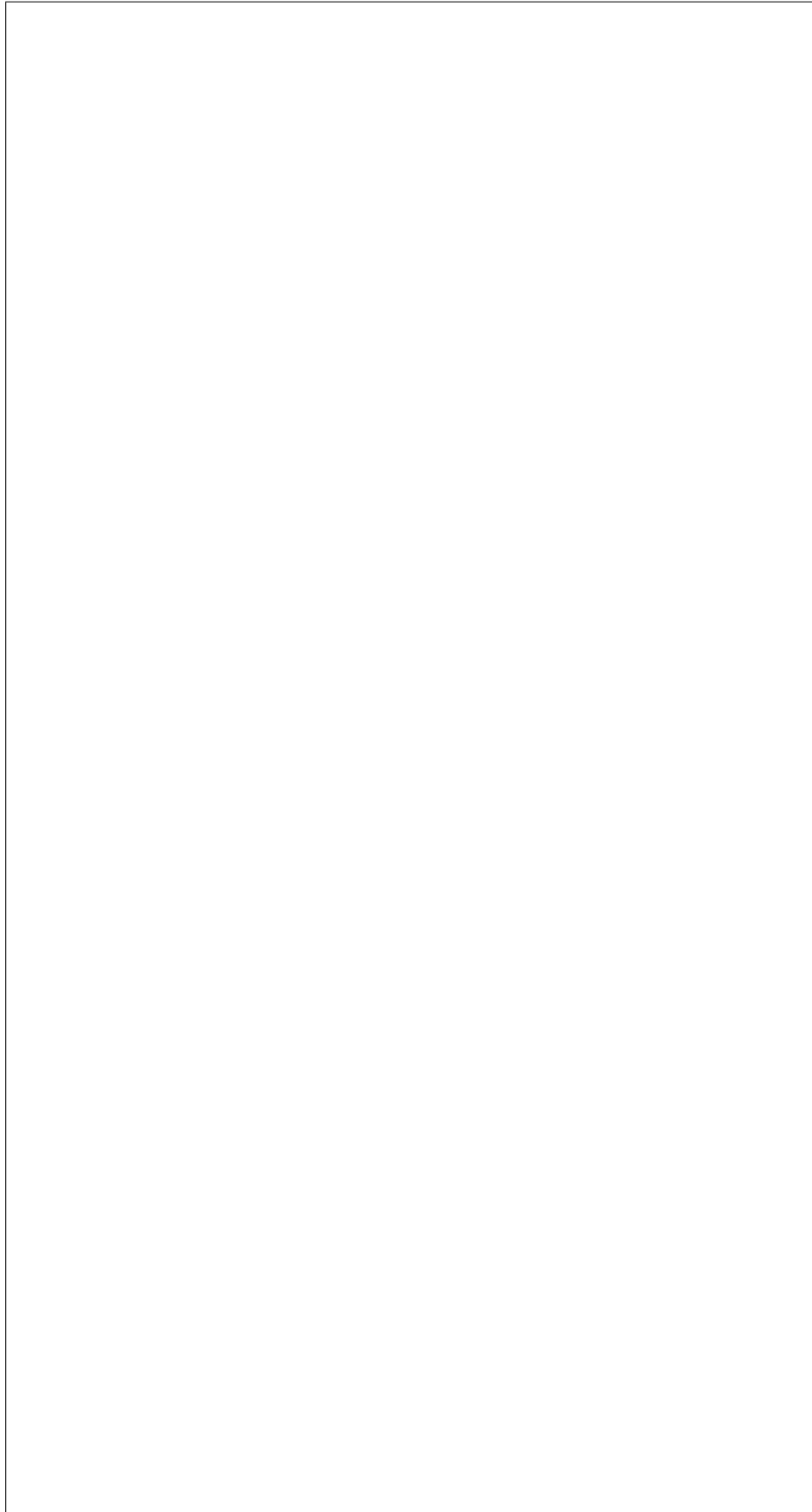
(continued from previous page)



(continues on next page)

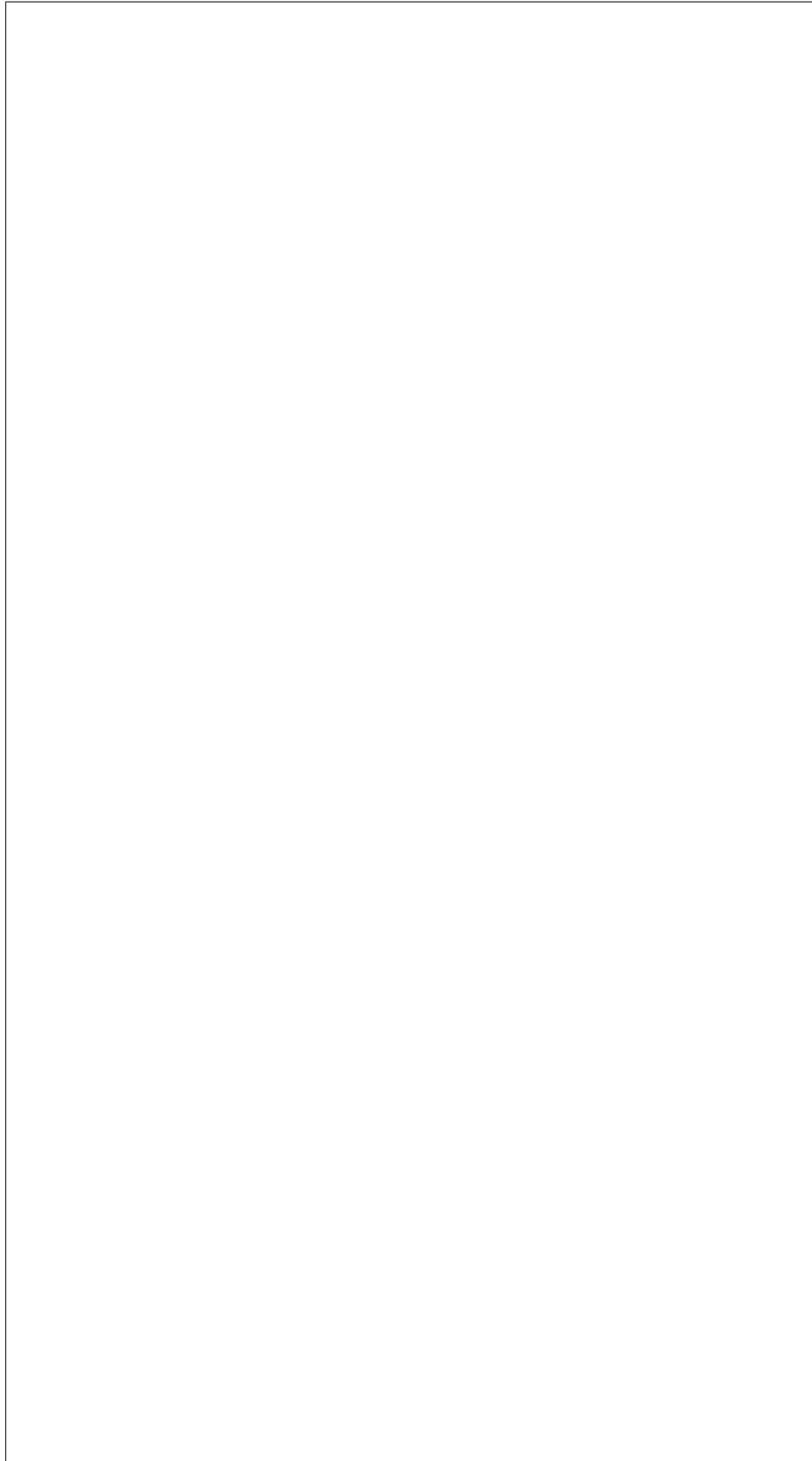


(continued from previous page)



(continues on next page)

(continued from previous page)



ironic.

lizing the connector information that describes the node.

De-  
tach  
a  
list  
of  
vol-  
ume  
from  
a  
pro-  
vide  
con-  
nec-  
tor  
de-  
tail.  
  
Enu-  
mer-  
ates  
thro  
a  
pro-  
vide  
list  
of  
vol-  
ume  
and  
is-  
sues  
de-  
tach  
men  
re-  
ques  
uti-

Paramet

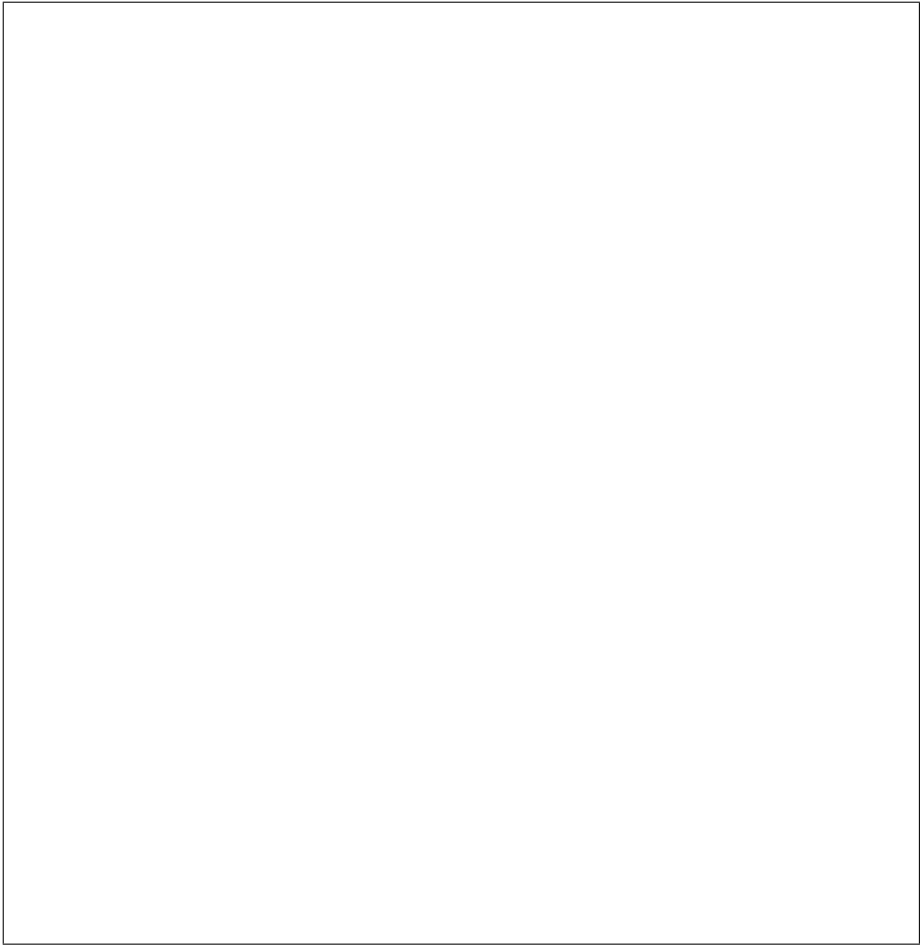
- **task**  
 The  
 Task  
 ager  
 task  
 rep-  
 re-  
 sent  
 ing  
 the

volume. This value can vary based upon the nodes configuration, capability, and ultimately the back-end storage driver. As cinder was designed around iSCSI, the ip and initiator keys are generally expected. For FiberChannel, the key wwpns can be used with a list of port addresses. Some drivers support a multipath boolean key, although it is generally False. The host key is generally used for logging by drivers. Example:



(continues on next page)

(continued from previous page)



instead of exceptions. Default False.

•
   
**all**
  
 Boo
   
 valu
   
 gov-
   
 ern-
   
 ing
   
 if
   
 er-
   
 rors
   
 that
   
 are
   
 re-
   
 turn
   
 are
   
 treat
   
 as
   
 warn
   
 ings
   
 in-
   
**Raises**
  
 Stor

ageE  
 ror  
 ironic.  
 Get  
 a  
 cin-  
 der  
 client  
 con-  
 nec-  
 tion.

**Parameter**  
**con**  
 re-  
 ques  
 con-  
 text,  
 in-  
 stan  
 of  
 iron

**Returns**  
 A  
 cin-  
 der  
 client

ironic.  
 Che  
 if  
 a  
 vol-  
 ume  
 is  
 at-  
 tach  
 to  
 the  
 sup-  
 plied  
 node

**Parameter**  
 •  
**nod**  
 The  
 ob-  
 ject  
 rep-

re-  
 sent  
 ing  
 the  
 node

• **vol**  
 The  
 ob-  
 ject  
 rep-  
 re-  
 sent  
 ing  
 the  
 vol-  
 ume  
 from  
 cin-  
 der.

**Returns**

Boo  
 in-  
 di-  
 cat-  
 ing  
 if  
 the  
 vol-  
 ume  
 is  
 at-  
 tach  
 Re-  
 turn  
 True  
 if  
 cin-  
 der  
 show

the volume as presently attached, otherwise returns False.

ironic.  
 Che  
 if  
 a  
 vol-  
 ume  
 is  
 avai  
 able

for  
 a  
 con-  
 nec-  
 tion.

Paramet

**vol**  
 The  
 ob-  
 ject  
 rep-  
 re-  
 sent  
 ing  
 the  
 vol-  
 ume

Returns

Boo  
 if  
 vol-  
 ume  
 is  
 avai  
 able

ironic.common.components module

Map  
 ping  
 of  
 com  
 mon  
 hard  
 ware  
 com  
 po-  
 nent  
 of  
 a  
 com  
 pute  
 sys-  
 tem.

ironic.  
 Cha  
 sis  
 en-



close  
ing  
one  
or  
more  
hard  
ware  
com  
po-  
nent

ironic.  
Storage  
age  
drive

ironic.  
Net-  
work  
in-  
ter-  
face

ironic.  
Power  
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  
lo-  
cal  
threa-  
ding  
is  
miss-  
ing.  
Set  
it  
with  
re-  
ques-  
con-  
text

and this is useful to log the `request_id` in log messages.

#### **to\_poli**

A  
dic-  
tio-  
nary  
of  
con-  
text  
at-

logged in user on which it applies policy enforcement. This dictionary defines a standard list of attributes that should be available for enforcement across services.

recated values or additional attributes used by that service specific policy.

tribu  
to  
en-  
forc  
pol-  
icy  
with  
  
oslo  
en-  
forc  
men  
re-  
quir  
a  
dic-  
tio-  
nary  
of  
at-  
tribu  
rep-  
re-  
sent  
ing  
the  
cur-  
rent  
  
It  
is  
ex-  
pect  
that  
ser-  
vice  
will  
of-  
ten  
have  
to  
over  
ride  
this  
meth  
with  
ei-  
ther  
dep-

ironic.common.dhcp\_factory module

ironic.  
 Cre-  
 ate  
 an  
 ad-  
 min-  
 is-  
 tra-  
 tor  
 con-  
 text.

**class** i  
 Base  
 obj

**clean\_c**  
 Clea  
 up  
 the  
 DHC  
 BOO  
 op-  
 tion  
 for  
 this  
 node

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan

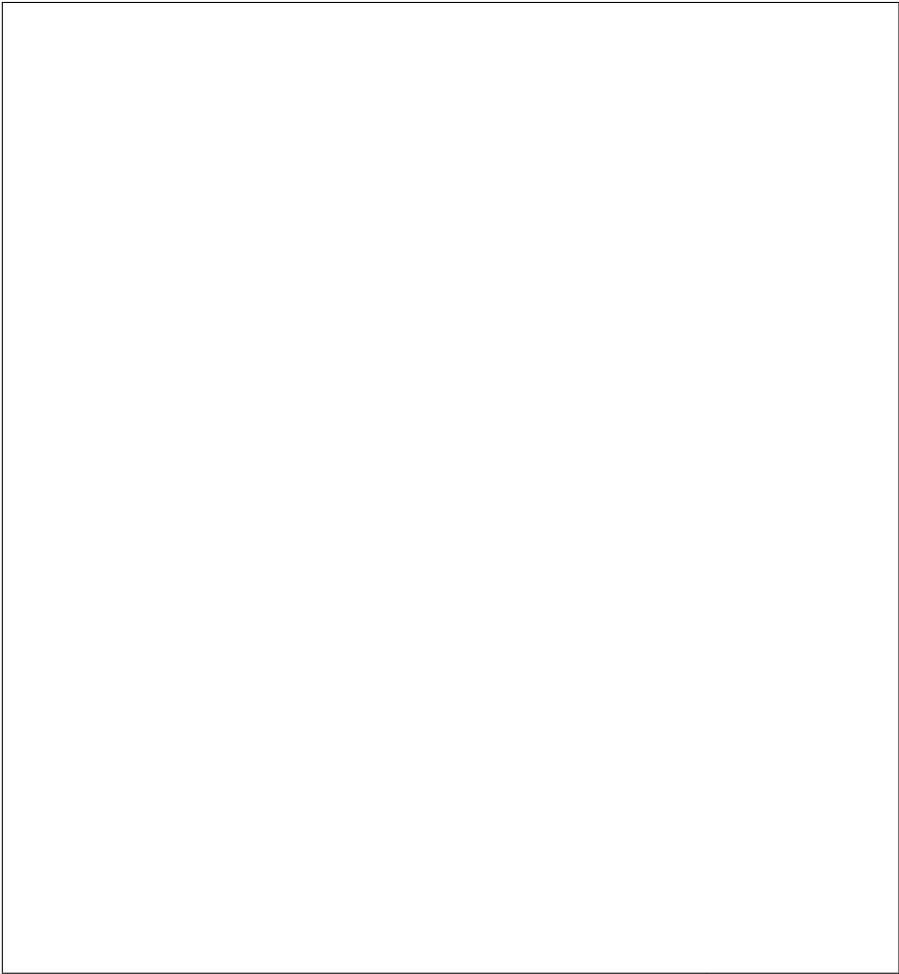
**propert**

**update\_**  
 Seno  
 or  
 up-  
 date  
 the  
 DHC  
 BOO  
 op-  
 tion

for  
 this  
 node

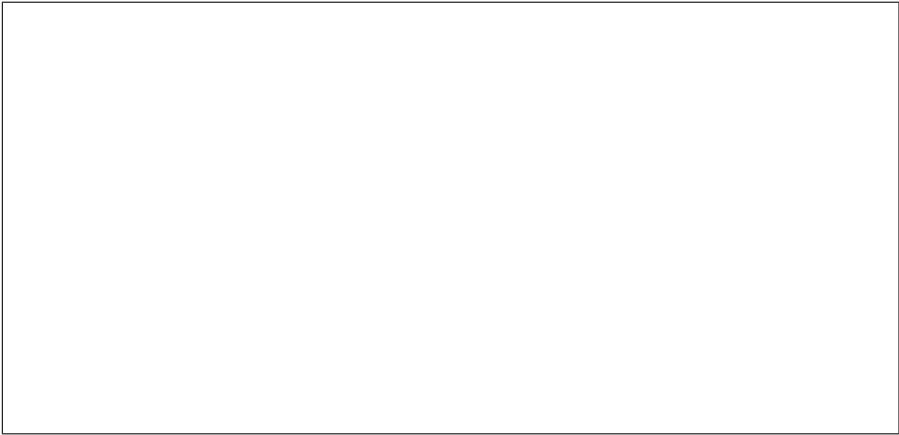
Paramete

- **taskmanager**  
 A Taskmanager instance
- **dicts**  
 this will be a list of dicts e.g.

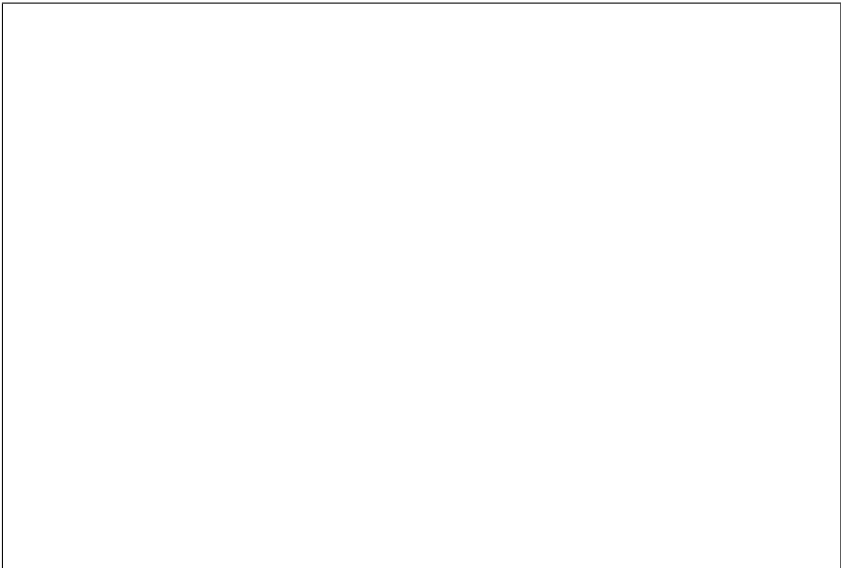


(continues on next page)

(continued from previous page)



of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)

--

If the value is None will get the list of ports from the IroniC port objects

## ironiC.common.driver\_factory module

**class** iBaseObj  
 Base class for all objects  
 Discovered objects are loaded and managed by the driver available  
 This is a subclass to load both main driver and extra inter-

face

**get\_dri**

**items** ( )

It-  
 er-  
 a-  
 tor  
 over  
 pair:  
 (nam  
 in-  
 stan

**propert**

The  
 list  
 of  
 drive  
 nam  
 avai  
 able

**class i**

Base  
*irc*  
*com*  
*dri*  
*Bas*

**class i**

Base  
*irc*  
*com*  
*dri*  
*Bas*

**class i**

Base  
*irc*  
*com*  
*dri*  
*Bas*

ironic.  
 Get  
 all  
 in-  
 ter-  
 face  
 for  
 all



in-  
 ter-  
 face  
 type

Returns

Dic-  
 tio-  
 nary  
 map-  
 ping  
 in-  
 ter-  
 face  
 type  
 to  
 dic-  
 tio-  
 nary  
 map-  
 ping  
 in-  
 ter-  
 face  
 nam

to interface object.

ironic.  
 Buil  
 a  
 com  
 pos-  
 able  
 drive  
 for  
 a  
 give  
 task  
  
 Star  
 with  
 a  
*Bar*  
 ob-  
 ject,  
 and  
 at-  
 tach  
 im-  
 ple-  
 men  
 ta-  
 tions

interfaces to it. They come from separate driver factories and are configurable via the database.

of  
the  
var-  
i-  
ous  
drive

**Parameters**  
 The task containing the node to build a driver for.

**Returns**  
 A driver object for the task

**Raises**  
 DriverNotFound if no node could be found in the iron namespace

**Raises**  
 InterfaceNotFound

unsupported values.

ible with it with the hardware type.

nEn  
try-  
poin  
if  
som  
node  
in-  
ter-  
face  
are  
set  
to  
in-  
valid  
or

Raises  
In-  
com  
pat-  
i-  
bleIn  
ter-  
face  
the  
re-  
ques  
im-  
ple-  
men  
ta-  
tion  
is  
not  
com  
pat-

ironic.

En-  
sure  
that  
node  
in-  
ter-  
face  
(e.g.  
for  
cre-  
ation

they are not provided.

instance is built for a node.

or  
up-  
dat-  
ing)  
are  
valid

Up-  
date  
(but  
does  
save  
to  
the  
data  
hard  
ware  
in-  
ter-  
face  
with  
cal-  
cu-  
latec  
de-  
fault  
if

This  
func-  
tion  
is  
run  
on  
node  
up-  
dat-  
ing  
and  
cre-  
ation  
as  
well  
as  
each  
time  
a  
drive

**Paramet**

- **node**  
node  
ob-  
ject  
to  
check  
and  
po-  
ten-  
tially,  
up-  
date

- **hw\_**  
hard  
ware  
type  
in-  
stan-  
ob-  
ject;  
will  
be  
de-  
tecte  
from  
node  
if  
miss  
ing

**Returns**  
True  
if  
any  
chan-  
were  
mad  
to  
the  
node  
oth-  
er-  
wise  
False

**Raises**  
In-  
ter-  
face  
Four

is not provided in the configuration

nEn  
try-  
poin  
on  
val-  
i-  
da-  
tion  
fail-  
ure

Raises

No-  
Vali  
De-  
fault  
ForI  
ter-  
face  
if  
the  
de-  
fault  
valu  
can-  
not  
be  
cal-  
cu-  
latec  
and

Raises

Driv  
Not-  
Four  
if  
the  
node  
hard  
ware  
type  
is  
not  
foun

ironic.

Cal-  
cu-  
late

abled in the configuration.

and  
 re-  
 turn  
 the  
 de-  
 fault  
 in-  
 ter-  
 face  
 im-  
 ple-  
 men-  
 ta-  
 tion.  
 Find  
 the  
 first  
 im-  
 ple-  
 men-  
 ta-  
 tion  
 that  
 is  
 sup-  
 port  
 by  
 the  
 hard  
 ware  
 type  
 and  
 is  
 en-

Paramet

- **hw\_**  
hard  
ware  
type  
in-  
stan-  
ob-  
ject.
- **int**  
type

sage.

**Returns**  
 an  
 en-



try-  
 poin  
 nam  
 of  
 the  
 cal  
 cu-  
 latec  
 de-  
 fault  
 im-  
 ple-  
 men  
 ta-  
 tion.

Raises

In-  
 ter-  
 face  
 Four  
 nEn  
 try-  
 poin  
 if  
 the  
 en-  
 try  
 poin  
 was  
 not  
 foun

Raises

No-  
 Vali  
 De-  
 fault  
 ForI  
 ter-  
 face  
 if  
 no  
 de-  
 fault  
 in-  
 ter-  
 face  
 can  
 be  
 foun

ironic.

faces for each interface type. This is the set of interfaces that are usable for this hardware type.

Get  
 us-  
 able  
 in-  
 ter-  
 face  
 for  
 a  
 give  
 hard  
 ware  
 type  
  
 For  
 a  
 give  
 hard  
 ware  
 type  
 find  
 the  
 in-  
 ter-  
 sec-  
 tion  
 of  
 en-  
 able  
 and  
 sup-  
 port  
 in-  
 ter-

**Parameter**  
**hardware**  
 The  
 hard  
 ware  
 type  
 ob-  
 ject  
 to  
 search  
  
**Returns**  
 a  
 dict  
 map-  
 ping  
 in-  
 ter-

face names.

face  
type  
to  
a  
list  
of  
en-  
able  
and  
sup-  
port  
in-  
ter-  
  
ironic.  
Get  
a  
hard  
ware  
type  
in-  
stan-  
by  
nam  
  
**Paramet**  
**har**  
the  
nam  
of  
the  
hard  
ware  
type  
to  
find  
  
**Returns**  
An  
in-  
stan-  
of  
iron  
  
**Raises**  
Driv  
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.

Paramet

- **hw\_**  
a  
hard  
ware  
type  
in-  
stan
- **int**  
nam  
of  
the  
in-  
ter-

try point (ironic.hardware.interfaces.<interface type>).

face  
type  
(e.g.  
boot

- **int**  
nam  
of  
the  
in-  
ter-  
face  
im-  
ple-  
men  
ta-  
tion  
from  
an  
ap-  
pro-  
pri-  
ate  
en-

#### Returns

in-  
stan  
of  
the  
re-  
ques  
in-  
ter-  
face  
im-  
ple-  
men  
ta-  
tion.

#### Raises

In-  
ter-  
face  
Four  
nEn  
try-  
poin  
if  
the

plementation is not compatible with it.

en-  
try  
poin  
was  
not  
foun

Raises

In-  
com  
pat-  
i-  
bleIn  
ter-  
face  
if  
hw\_  
is  
a  
hard  
ware  
type  
and  
the  
re-  
ques  
im-

ironic.  
Get  
all  
hard  
ware  
type

Returns

Dic-  
tio-  
nary  
map  
ping  
hard  
ware  
type  
nam  
to  
hard  
ware  
type  
ob-  
ject.

```

    ironic.
        Get
        all
        in-
        ter-
        face
        for
        a
        give
        in-
        ter-
        face
        type

Parameters
    int
    String
    type
    of
    in-
    ter-
    face
    to
    fetch
    for.

Returns
    Dictionary
    mapping
    in-
    ter-
    face
    names
    to
    in-
    ter-
    face
    objects.

```

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
 ird
 com
 exc
 Inv

excepti

Base
 ird
 com
 exc
 Not

excepti

Base
 Run

properti

excepti

Base
 ird
 exc
 Iro

excepti

Base
 ird
 exc
 Iro

excepti

Base
 ird
 exc
 Iro

excepti

Base
 ird
 com
 exc

*Not*

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
 iro  
 exc  
 Iro

**code =**

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
*iro*  
*com*  
 exc  
*Con*

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base
   
*irc*
  
*com*
  
*exc*
  
*Con*

**excepti**

Base
   
*irc*
  
*com*
  
*exc*
  
*Con*

**excepti**

Base
   
*irc*
  
*com*
  
*exc*
  
*Not*

**excepti**

Base
   
 iro
   
 exc
   
 Iro

**excepti**

Base
   
*irc*
  
*com*
  
*exc*
  
*Dri*

**excepti**

Base
   
 iro
   
 exc
   
 Iro

**excepti**

Base
   
*irc*
  
*com*
  
*exc*
  
*Not*

**excepti**

Base
   
*irc*

*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
    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  
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*  
*Con*

**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  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Inv*  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Inv*  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Con*  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Inv*  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Inv*  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Inv*  
**except i**  
 Base  
*irc*  
*com*  
*exc*  
*Inv*  
**except i**  
 Base  
*irc*  
*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  
*Con*

**excepti**

Base  
iro  
exc  
Iro

**excepti**

Base  
*iro*  
*com*

exc
 Tem

code =

excepti

Base
 ird
 com
 exc
 Tem

excepti

Base
 ird
 com
 exc
 Inv

excepti

Base
 ird
 com
 exc
 Inv

excepti

Base
 ird
 com
 exc
 Not

excepti

Base
 ird
 com
 exc
 Con

excepti

Base
 ird
 com
 exc
 Inv

excepti

Base

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  
irc  
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
   
*irc*
  
*com*
  
*exc*
  
*Con*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*
  
*Not*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*
  
*Con*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*
  
*Con*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*
  
*Con*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*
  
*Con*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*
  
*Inv*
  
**excepti**
  
 Base
   
*irc*
  
*com*
  
*exc*

*Not*

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
*iro*  
*com*  
*exc*  
*Inv*

**except i**

Base  
*iro*  
*com*  
*exc*  
*Rea*

**except i**

Base  
*iro*  
*com*  
*exc*  
*Dri*

**except i**

Base  
*iro*  
*com*  
*exc*  
*Dri*

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base

*irc*  
*com*  
*exc*  
*Swi*

**excepti**

Base  
 irc  
 exc  
 Iro

**excepti**

Base  
 irc  
 exc  
 Iro

**code =**

**excepti**

Base  
 irc  
 exc  
 Iro

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Cli*

**propert**

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Cli*

**add\_fie**

Add  
 a  
 field  
 nam  
 to

this method will prepend `name` to the hierarchy of names.

con-  
 cate  
 nate  
 the  
 full  
 nam  
 Add  
 a  
 field  
 nam  
 so  
 that  
 the  
 who  
 hi-  
 er-  
 ar-  
 chy  
 is  
 dis-  
 play  
 Suc-  
 ces-  
 sive  
 calls  
 to

**property**

**except i**

Base  
*irc*  
*com*  
*exc*  
*Inv*

**except i**

Base  
 iro  
 exc  
 Iro

**except i**

Base  
*irc*  
*com*  
*exc*  
*Con*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Con*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Inv*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Con*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Not*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Con*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Con*

**excepti**

Base  
*irc*  
*com*  
*exc*  
*Con*

**except i**

Base  
*irc*  
*com*  
*exc*  
*Not*

**except i**

Base  
 iro  
 exc  
 Iro

## ironic.common.faults module

Fault  
 def-  
 i-  
 ni-  
 tions

`ironic.`  
 Node  
 is  
 mov  
 to  
 main  
 te-  
 nanc  
 due  
 to  
 fail-  
 ure  
 of  
 a  
 clea  
 ing  
 op-  
 er-  
 a-  
 tion.

`ironic.`  
 Node  
 is  
 mov  
 to  
 main  
 te-

abort.

ironic.common.fsm module

nanc  
 due  
 to  
 pow  
 syn-  
 chro  
 niza  
 tion  
 fail-  
 ure.  
 ironic.  
 Nod  
 is  
 mov  
 to  
 main  
 te-  
 nanc  
 due  
 to  
 fail-  
 ure  
 of  
 clea  
 ing  
 up  
 dur-  
 ing  
 res-  
 cue

class i  
 Base  
 aut  
 mac  
 Fin  
 An  
 iron  
 state  
 mac  
 class  
 with  
 som  
 iron  
 spe-  
 cific



ad-  
di-  
tions

#### **add\_sta**

Add  
a  
give  
state  
to  
the  
state  
ma-  
chin

#### **Parame**

- **sta**  
Use  
this  
to  
spec  
ify  
that  
this  
state  
is  
a  
sta-  
ble/p  
state  
A  
state  
mus  
have  
been

previously defined as stable before it can be used as a target

- **tar**  
The  
tar-  
get  
state  
for  
state  
to  
go  
to.

target it must have been previously added and specified as stable

Be-  
 fore  
 a  
 state  
 can  
 be  
 used  
 as  
 a  
  
 Fur-  
 ther  
 ar-  
 gu-  
 men-  
 are  
 in-  
 ter-  
 prete  
 as  
 for  
 par-  
 ent  
 meth  
 add  
  
**add\_tra**  
 Add  
 an  
 al-  
 lowe  
 tran-  
 si-  
 tion  
 from  
 start  
 -  
 >  
 end  
 for  
 the  
 give  
 even  
  
**Parame**  
  
 •  
  
**sta**  
 start  
 ing  
 state

transition already exists.

• **end**  
 end-  
 ing  
 state  
 • **even**  
 even  
 that  
 caus  
 start  
 state  
 to  
 tran-  
 si-  
 tion  
 to  
 end  
 state  
 • **rep**  
 re-  
 plac  
 ex-  
 ist-  
 ing  
 even  
 in-  
 stea  
 of  
 rais-  
 ing  
 a  
 Dup  
 ex-  
 cep-  
 tion  
 whe  
 the

**initial**  
 Ini-  
 tial-  
 ize  
 the  
 FSM

**Parame**

•

use the default target state

**sta**  
 the  
 FSM  
 is  
 ini-  
 tial-  
 ized  
 to  
 start  
 from  
 this  
 state  
 •  
**tar**  
 if  
 spec  
 i-  
 fied,  
 the  
 FSM  
 is  
 ini-  
 tial-  
 ized  
 to  
 this  
 tar-  
 get  
 state  
 Oth-  
 er-  
 wise  
**is\_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

- eve  
 the  
 even  
 to  
 be  
 pro-  
 cess
- tar  
 if  
 spec  
 i-  
 fied,  
 the  
 fi-  
 nal  
 tar-  
 get  
 state  
 for

default target state

ironic.common.hash\_ring module

ironic.common.i18n module

the  
 even  
 Oth-  
 er-  
 wise  
 use  
 the  
  
 propert  
  
  
 class i  
  
 Base  
 obj  
  
 get\_rin  
  
 classme  
  
 propert  
  
  
  
 oslo  
 in-  
 te-  
 gra-  
 tion  
 mod  
 ule.  
  
 See  
<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

**abstract**  
 Dow  
 load  
 im-  
 age  
 to  
 spec  
 i-  
 fied  
 lo-  
 ca-  
 tion.

### Parame

- **ima**  
 Im-  
 age  
 ref-  
 er-  
 ence
- **ima**  
 File  
 ob-  
 ject  
 to  
 writ  
 data  
 to.

**Raises**  
 ex-  
 cep-  
 tion.

erties. updated\_at attribute is a naive UTC datetime object.

**Raises**  
 ex-  
 cep-  
 tion.

**abstract**  
 Get  
 dic-  
 tio-  
 nary  
 of  
 im-  
 age  
 prop  
 er-  
 ties.

**Paramet**  
**ima**  
 Im-  
 age  
 ref-  
 er-  
 ence

**Raises**  
 ex-  
 cep-  
 tion.

**Returns**  
 dic-  
 tio-  
 nary  
 of  
 im-  
 age  
 prop  
 er-  
 ties.  
 It  
 has  
 three  
 of  
 then  
 size,  
 up-  
 date  
 and  
 prop

**abstract**  
 Val-



i-  
date  
im-  
age  
ref-  
er-  
ence

**Parameters**  
**image**  
Im-  
age  
ref-  
er-  
ence

**Raises**  
ex-  
cep-  
tion.

**Returns**  
In-  
for-  
ma-  
tion  
need  
to  
fur-  
ther  
op-  
er-  
ate  
with  
an  
im-  
age.

**class** `IronicImage`  
Base  
class  
for  
Ironic  
images.  
Provides  
re-  
trieval  
of  
disk  
im-  
ages  
avai-

able  
 lo-  
 cally  
 on  
 the  
 con-  
 duc-  
 tor.

downloa

Dow  
 load  
 im-  
 age  
 to  
 spec  
 i-  
 fied  
 lo-  
 ca-  
 tion.

Parame

- **ima**  
 Im-  
 age  
 ref-  
 er-  
 ence
- **ima**  
 File  
 ob-  
 ject  
 to  
 writ  
 data  
 to.

Raises

ex-  
 cep-  
 tion.  
 if  
 sour  
 im-  
 age  
 file  
 does

link.

ex-  
ist.  
  
**Raises**  
 ex-  
cep-  
tion.  
if  
ex-  
cep-  
tions  
were  
raise  
while  
writ  
ing  
to  
file  
or  
cre-  
at-  
ing  
hard

**show** (*im*  
 Get  
dic-  
tio-  
nary  
of  
im-  
age  
prop  
er-  
ties.

**Parame**  
**ima**  
 Im-  
age  
ref-  
er-  
ence

**Raises**  
 ex-  
cep-  
tion.  
if  
im-  
age  
file

erties. updated\_at attribute is a naive UTC datetime object.

spec  
i-  
fied  
does  
ex-  
ist.  
  
**Returns**  
dic-  
tio-  
nary  
of  
im-  
age  
prop  
er-  
ties.  
It  
has  
three  
of  
them  
size,  
up-  
date  
and  
prop  
  
**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  
 HTTP

**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  
equa  
to  
200.

**Raises**

ex-  
cep-  
tion.  
if:  
\*  
IO-  
Er-  
ror  
hap-  
pene  
dur-  
ing  
file  
write

\*

GET  
re-  
ques  
faile

**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  
re-  
turn  
re-  
spor  
code  
not

equal to 200; \* Content-Length header not found in response to HEAD request.

**Returns**  
dic-  
tio-  
nary  
of

erties. updated\_at attribute is a naive UTC datetime object.

im-  
 age  
 prop  
 er-  
 ties.  
 It  
 has  
 three  
 of  
 them  
 size.  
 up-  
 date  
 and  
 prop

**validat**  
 Val-  
 i-  
 date  
 HTT  
 im-  
 age  
 ref-  
 er-  
 ence

**Parame**

- ima**  
 Im-  
 age  
 ref-  
 er-  
 ence
- sec**  
 Spec  
 ify  
 if  
 im-  
 age\_  
 be-  
 ing  
 val-  
 i-  
 date  
 shou  
 not



message.

be  
 show  
 in  
 ex-  
 cep-  
 tion

Raises

ex-  
 cep-  
 tion.  
 if  
 HEA  
 re-  
 ques  
 faile  
 or  
 re-  
 turn  
 re-  
 spon  
 code  
 not  
 equa  
 to  
 200.

Returns

Re-  
 spon  
 to  
 HEA  
 re-  
 ques

ironic.

Get  
 im-  
 age  
 ser-  
 vice  
 in-  
 stan  
 to  
 dow  
 load  
 the  
 im-  
 age.

Paramet

- **ima**  
 Strin  
 con-  
 tain-  
 ing  
 href  
 to  
 get  
 im-  
 age  
 ser-  
 vice  
 for.
- **cli**  
 Glan  
 clien  
 to  
 be  
 used  
 for  
 dow  
 load  
 used  
 only  
 if  
 im-  
 age\_  
 is  
 Glan  
 href
- **con**  
 re-  
 ques  
 con-  
 text,  
 used  
 only  
 if  
 im-  
 age\_  
 is  
 Glan  
 href

Raises  
 ex-

age.

**ironic.common.images module**

**Returns**

Instance of an image service class that is able to download specified image.

Handling of VM disk images

ironic.  
Get

size of the image.

Paramet

- **pat**  
 path  
 to  
 the  
 im-  
 age  
 file.
- **est**  
 Whe  
 to  
 es-  
 ti-  
 mate  
 the  
 size  
 by  
 scal-

ing  
 the  
 origi  
 i-  
 nal  
 size

### Returns

For  
*es-*  
*ti-*  
*mate*  
 re-  
 turn  
 the  
 size  
 of  
 the  
 raw  
 im-  
 age  
 file.  
 For  
*es-*  
*ti-*  
*mate*  
 re-

turn the size of the original image scaled by the configuration value *raw\_image\_growth\_factor*.

ironic.

Cre-  
 ates  
 a  
 boot  
 ISO  
 im-  
 age  
 for  
 a  
 node  
  
 Give  
 the  
 href  
 for  
 ker-  
 nel,  
 ram

this method fetches the kernel and ramdisk, and builds a bootable ISO image that can be used to boot up the baremetal node.

Parameters

- **context**
 The context object.
- **output\_path**
 The absolute path of the output ISO file.
- **kernel\_uri**
 The URI or glance uuid of the kernel to use.
- **ramdisk\_uri**
 The URI

tion image. If not specified, the *esp\_image\_href* option must be present if UEFI-bootable ISO is desired.

or  
glan  
uuid  
of  
the  
rame  
to  
use

- **dep**  
URI  
or  
glan  
UUI  
of  
the  
de-  
ploy  
ISO  
im-  
age  
to  
ex-  
tract  
EFI  
sys-  
tem  
par-

- **esp**  
URI  
or  
glan  
UUI  
of  
FAT  
form  
EFI  
sys-  
tem  
par-  
ti-  
tion  
im-  
age  
con-  
tain-  
ing

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.

ments of the form  $K=V$  or  $K$  (optional).

trieved for to use, instead of building an ISO bootable ramdisk.

- **ker**  
a  
strin  
con-  
tain-  
ing  
whit  
pace  
sep-  
a-  
rate  
val-  
ues  
ker-  
nel  
cmd  
line  
ar-  
gu-
- **bas**  
URI  
or  
glan  
UUI  
of  
a  
to  
be  
used  
as  
an  
over  
ride  
of  
wha  
shou  
be  
re-
- **inj**  
Map  
ping  
of



image.

lo-  
cal  
sour  
file  
path  
to  
their  
lo-  
ca-  
tion  
on  
the  
fi-  
nal  
ISO

**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

rectory, generates the grub configuration file using kernel parameters and then generates a bootable ISO image for UEFI.

**Paramet**

- **out**  
the  
path  
to  
the  
file  
when  
the  
iso  
im-  
age  
need  
to  
be  
cre-

not specified, the *esp\_image* option is required.

ated

- **ker**  
the  
ker-  
nel  
to  
use.

- **ram**  
the  
ram  
to  
use.

- **dep**  
de-  
ploy  
ISO  
im-  
age  
to  
ex-  
tract  
EFI  
sys-  
tem  
par-  
ti-  
tion  
im-  
age  
from  
If

- **esp**  
FAT  
form  
EFI  
sys-  
tem  
par-  
ti-  
tion  
im-  
age  
con-  
tain-

GRUB2) for each hardware architecture to boot. This image will be embedded into the ISO image. If not specified, the *deploy\_iso* option is required.

nation of them like K1=V1,K2,) to be added as the kernel cmdline.

image.

ing  
the  
EFI  
boot  
load  
(e.g.

- **ker**  
a  
list  
of  
strin  
el-  
e-  
men  
be-  
ing  
a  
strin  
like  
K=V  
or  
K  
or  
com  
bi-

- **inj**  
Map  
ping  
of  
lo-  
cal  
sour  
file  
path  
to  
their  
lo-  
ca-  
tion  
on  
the  
fi-  
nal  
ISO

command to generate iso.

Raises  
 Im-  
 age-  
 Cre-  
 ation  
 Fail-  
 if  
 im-  
 age  
 cre-  
 ation  
 fail-  
 while  
 copy  
 ing  
 files  
 or  
 while  
 run-  
 ning  
  
 ironic.

Cre-  
 ates  
 an  
 isoli-  
 im-  
 age  
 on  
 the  
 spec-  
 i-  
 fied  
 file.  
  
 Cop-  
 the  
 pro-  
 vide  
 ker-  
 nel,  
 ram  
 to  
 a  
 di-  
 rec-  
 tory.

ration 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  
 cre-  
 ated
- **ker**  
 the  
 ker-  
 nel  
 to  
 use.
- **ram**  
 the  
 ram  
 to  
 use.
- **ker**  
 a  
 list  
 of  
 strin  
 el-  
 e-

nation of them like  $K1=V1,K2,)$  to be added as the kernel cmdline.

image.

men  
be-  
ing  
a  
strin  
like  
 $K=V$   
or  
K  
or  
com  
bi-

- **inj**  
Map  
ping  
of  
lo-  
cal  
sour  
file  
path  
to  
their  
lo-  
ca-  
tion  
on  
the  
fi-  
nal  
ISO

**Raises**  
Im-  
age-  
Cre-  
ation  
Fail  
if  
im-  
age  
cre-  
ation  
faile  
whil  
copy  
ing  
files

command to generate iso.

ironic.

Creates the fat fs image on the desired file.

This method copies the given files to a root directory (optional) writes the parameters

specified to the parameters file within the root directory (optional), and then creates a vfat image of the root directory.

Parameters

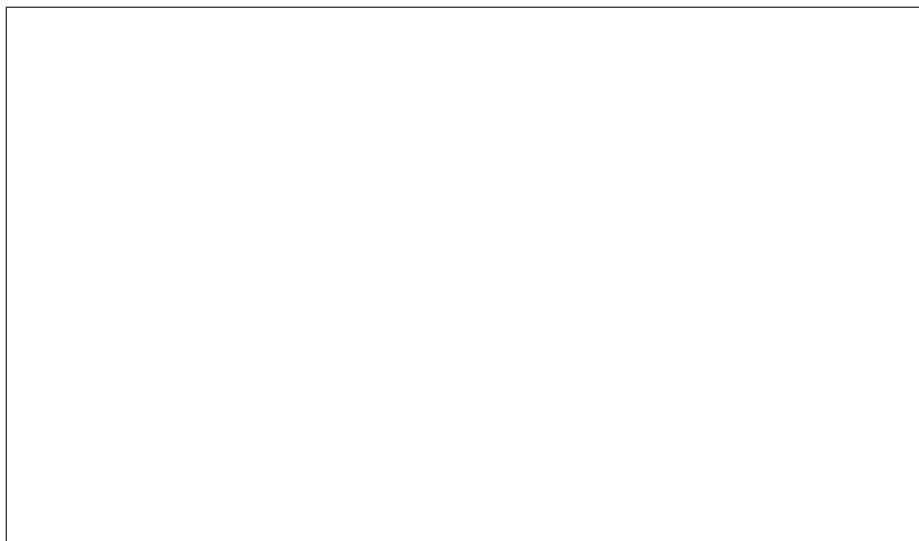
- **out** The path



to  
the  
file  
when  
the  
fat  
fs  
image  
needs  
to  
be  
created

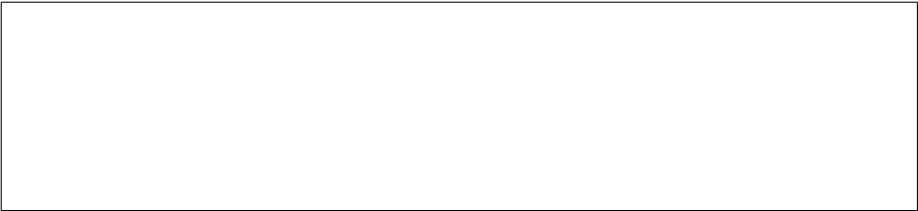
- **fil**  
A  
dict  
containing  
absolute  
path  
of  
file  
to  
be  
copied  
-  
>  
rel-  
a-

tive path within the vfat image. For example:



(continues on next page)

(continued from previous page)



- **par**  
 A dict containing key-value pairs of parameters.
  - **par**  
 The file-name for the parameter-file.
  - **fs\_**  
 size of the vfat files in KiB
- Raises**  
 Im-age-Cre-ation

nipulation activities like creating dirs, mounting, creating filesystem, copying files, etc.

Fail  
if  
im-  
age  
cre-  
ation  
faile  
whil  
do-  
ing  
any  
of  
files  
tem  
ma-

ironic.

ironic.

ironic.

ironic.

ironic.

Re-  
turn  
the  
val-  
ues  
of  
sev-  
eral  
prop  
er-  
ties  
of  
an  
im-  
age

#### Paramet

- **con**  
con-  
text
-

fault value is all, so if not specified all properties will be returned.

data will have a value of None.

Returns  
 a  
 dict  
 of  
 the  
 val-  
 ues  
 of  
 the  
 prop  
 er-  
 ties.  
 A  
 prop  
 erty  
 not  
 on  
 the  
 glan  
 met:

ironic.

ironic.

Re-  
turn  
the  
tmp  
url  
for  
a  
glan  
im-  
age.

#### Paramet

- **con**  
con-  
text
- **ima**  
the  
UUI  
of  
the  
im-  
age  
in  
glan

#### Returns

the  
tmp  
url  
for  
the  
glan  
im-  
age.

ironic.

ironic.

ironic.  
Find  
out  
if  
the  
im-  
age

is  
 a  
 par-  
 ti-  
 tion  
 im-  
 age  
 or  
 a  
 who  
 disk  
 im-  
 age.

**Paramet**

- **ctx**
  
 an  
 ad-  
 min  
 con-  
 text
- **ins**
  
 a  
 node  
 in-  
 stan-  
 info  
 dict

**Returns**

True  
 for  
 who  
 disk  
 im-  
 ages  
 and  
 Fals  
 for  
 par-  
 ti-  
 tion  
 im-  
 ages  
 and  
 Non  
 on  
 no

age\_source or Error.

## ironic.common.indicator\_states module

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

Central  
place  
for  
han-  
dling  
Key  
ston

au-  
tho-  
riza-  
tion  
and  
ser-  
vice  
look

ironic.  
Load  
adap  
from  
op-  
tions  
in  
a  
con-  
fig-  
u-  
ra-  
tion  
file  
sec-  
tion.  
The  
adap  
will  
be  
pass  
di-  
rectl  
to  
key-  
ston  
Ada  
and  
will  
over  
ride  
the  
val-  
ues  
load  
from

config. Consult keystoneauth1 docs for available adapter options.

Paramet  
gro  
nam  
of



the  
con-  
fig  
sec-  
tion  
to  
load  
adap  
op-  
tions  
from

ironic.  
Load  
auth  
plu-  
gin  
from  
op-  
tions  
in  
a  
con-  
fig-  
u-  
ra-  
tion  
file  
sec-  
tion.  
The  
auth  
will  
be  
pass  
di-  
rectl  
to  
key-  
ston  
auth  
plu-  
gin  
and  
will  
over  
ride  
the  
val-  
ues

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.

Paramet

gro  
nam  
of  
the  
con-  
fig  
fig  
sec-  
tion  
to  
load  
auth  
plu-  
gin  
op-  
tions  
from

ironic.  
Get  
an  
end-  
poin  
from  
an  
adap

The  
adap  
will  
be  
pass  
di-  
rectl  
to  
key-  
ston  
Ada  
and  
will  
over  
ride  
the  
val-  
ues  
load  
from

config. Consult keystoneauth1 docs for available adapter options.

Paramet

gro

nam  
of  
the  
con-  
fig  
sec-  
tion  
to  
load  
adap  
op-  
tions  
from

#### **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

will not fail if the user token is expired.

isnt serialized yet.

mid-  
dle-  
ware  
re-  
ques  
with  
valid  
ser-  
vice  
auth

Ide-  
ally  
we  
wou  
use  
the  
plu-  
gin  
pro-  
vide  
by  
auth  
mid-  
dle-  
ware  
how  
ever  
this  
plu-  
gin

ironic.  
Loa  
ses-  
sion  
ob-  
ject  
from  
op-  
tions  
in  
a  
con-  
fig-  
u-  
ra-  
tion  
file  
sec-

loaded from config. Consult keystoneauth1 docs for available options.

tion.  
The  
ses-  
sion  
will  
be  
pass  
di-  
rectl  
to  
key-  
ston  
Ses-  
sion  
and  
will  
over  
ride  
the  
val-  
ues

**Parameter**  
**gro**  
nam  
of  
the  
con-  
fig  
sec-  
tion  
to  
load  
ses-  
sion  
op-  
tions  
from

ironic.  
Wra  
key-  
ston  
func  
tions  
and  
cen-  
tral-  
izes  
ex-  
cep-

ironic.common.network module

tion  
 han-  
 dling

ironic.  
 Get  
 all  
 VIF  
 ids  
 for  
 a  
 node  
 This  
 func  
 tion  
 does  
 not  
 han-  
 dle  
 multi  
 node  
 op-  
 er-  
 a-  
 tions

**Paramet**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan

**Returns**

A  
 dict  
 of  
 Nod  
 neu-  
 tron  
 port  
 whe  
 keys  
 are  
 port  
 &

port  
grou  
and  
the  
val-  
ues  
are  
dict

of UUIDs and their associated VIFs, e.g.



ironic.

Re-  
turn  
the  
set  
of  
phys  
i-  
cal  
net-  
worl  
as-  
so-  
ci-  
ated  
with

physical network, or None.

a  
port  
group

**Parameters**

- task**  
a Task object representing the instance
- port\_id**  
ID of the port group
- exclude**  
A Port object to exclude from the determination of the port group

**Returns**  
The set of physical networks



contain zero or one physical networks.

ical network.

as-  
so-  
ci-  
ated  
with  
the  
port  
grou  
The  
set  
will

#### **Raises**

Port  
grou  
Phys  
net-  
Inco  
sis-  
tent  
if  
the  
port  
grou  
port  
are  
not  
as-  
sign  
the  
sam  
phys

ironic.  
Re-  
turn  
the  
set  
of  
phys  
i-  
cal  
net-  
worl  
for  
a  
node  
  
Re-  
turn  
the

ical network None is excluded from the set.

set  
 of  
 phys  
 i-  
 cal  
 net-  
 worl  
 as-  
 so-  
 ci-  
 ated  
 with  
 a  
 node  
 port.  
 The  
 phys

**Parameter**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan

**Returns**  
 A  
 set  
 of  
 phys  
 i-  
 cal  
 net-  
 worl

ironic.  
 Look  
 a  
 port  
 grou  
 by  
 ID  
 on  
 a  
 task  
 ob-  
 ject.

**Parameter**  
 •

tas
 a
 Task
 ager
 in-
 stan

- **por**
  
 ID
 of
 the
 port
 grou

**Returns**
  
 A
 Port
 grou
 ob-
 ject
 or
 Non

ironic.
 Look
 port
 by
 their
 port
 grou
 ID
 on
 a
 task
 ob-
 ject.

Paramet

- **tas**
  
 a
 Task
 ager
 in-
 stan
- **por**
  
 ID
 of
 the

port  
 grou

Returns

A  
 list  
 of  
 Port  
 ob-  
 jects

ironic.  
 Re-  
 mov  
 all  
 vif  
 at-  
 tach  
 men  
 reco  
 from  
 a  
 node

Paramet

tas  
 a  
 Task  
 ager  
 in-  
 stan

ironic.common.neutron module

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  
 in-  
 spec  
 tion.

Param

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  
 mor  
 re-  
 quir  
 pa-  
 ram-

Raises

Un-

eters

sup-  
port  
ed-  
Driv  
ten-  
sion

ironic.  
Nam  
of  
the  
neu-  
tron  
net-  
worl  
API  
phys  
i-  
cal  
net-  
worl  
pa-  
ram-  
e-  
ter.

ironic.  
  
Cre-  
ate  
neu-  
tron  
port  
to  
boot  
the  
rame  
  
Cre-  
ate  
neu-  
tron  
port  
for  
each  
pxe\_  
port  
on  
task  
to  
boot  
the  
rame

created these neutron ports will not have any assigned IP addresses.

If  
the  
con-  
fig  
op-  
tion  
neu-  
tron  
is  
set,  
neu-  
tron  
port  
for  
non-  
pxe-  
enab  
port  
are  
also

#### Paramet

- **tas**  
a  
Task  
ager  
in-  
stan
- **net**  
UI  
of  
a  
neu-  
tron  
net-  
worl  
whe  
port  
will  
be  
cre-  
ated
- **sec**  
List  
of

Se-  
cu-  
rity  
Grou  
UU  
to  
be  
used  
for  
net-  
worl

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-  
text,  
in-  
stan  
of



ironi

- **aut**  
(boo  
Whe  
True  
use  
auth  
val-  
ues  
from  
conf  
pa-  
ram-  
e-  
ters

**Returns**  
A  
neu-  
tron  
clien

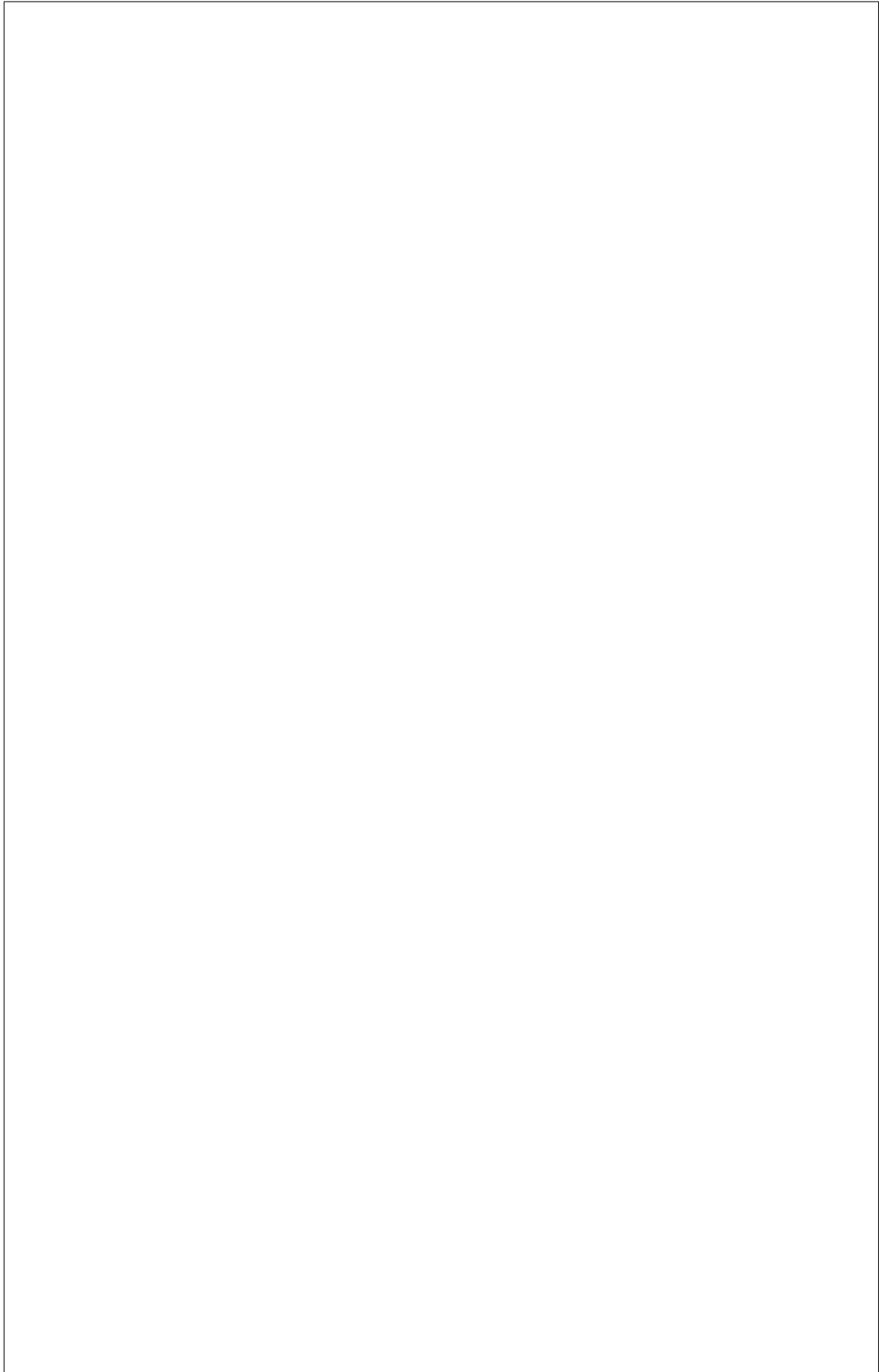
`ironic.`  
Ex-  
tract  
the  
port  
grou  
in-  
for-  
ma-  
tion.

The  
in-  
for-  
ma-  
tion  
is  
re-  
turn  
in  
the  
form  
of:



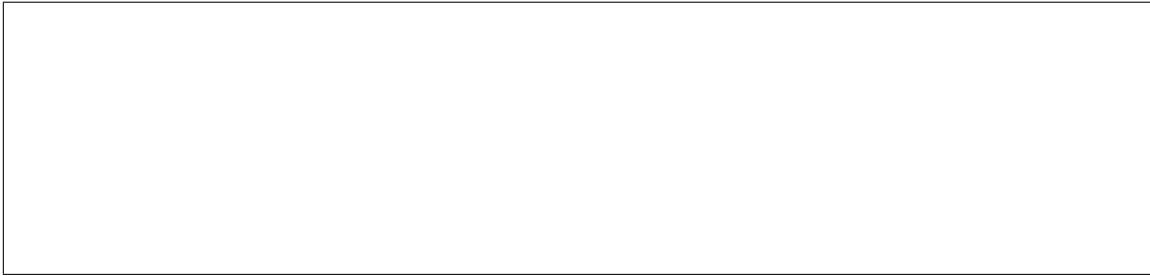
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



Parameters

- **task**  
 a task containing the Node object.
- **port**  
 Iron port group object to extract data for.

Returns

port group information as a dict

ironic.

Gather Neutron

tron  
 port  
 and  
 net-  
 worl  
 con-  
 fig-  
 u-  
 ra-  
 tion  
  
 Que  
 Neu  
 tron  
 for  
 port  
 and  
 net-  
 worl  
 con-  
 fig-  
 u-  
 ra-  
 tion,  
 re-  
 turn  
 wha  
 ever  
 is  
 avai  
 able

Paramet

- **por**  
 iron  
 port  
 ID.
- **vif**  
 Neu  
 tron  
 port  
 ID.
- **cli**  
 Op-  
 tion:  
 a  
 Neu

tron  
clien  
ob-  
ject.

- **con**  
(ir  
com  
com  
Req  
re-  
ques  
con-  
text

#### Raises

Net-  
worl  
Er-  
ror

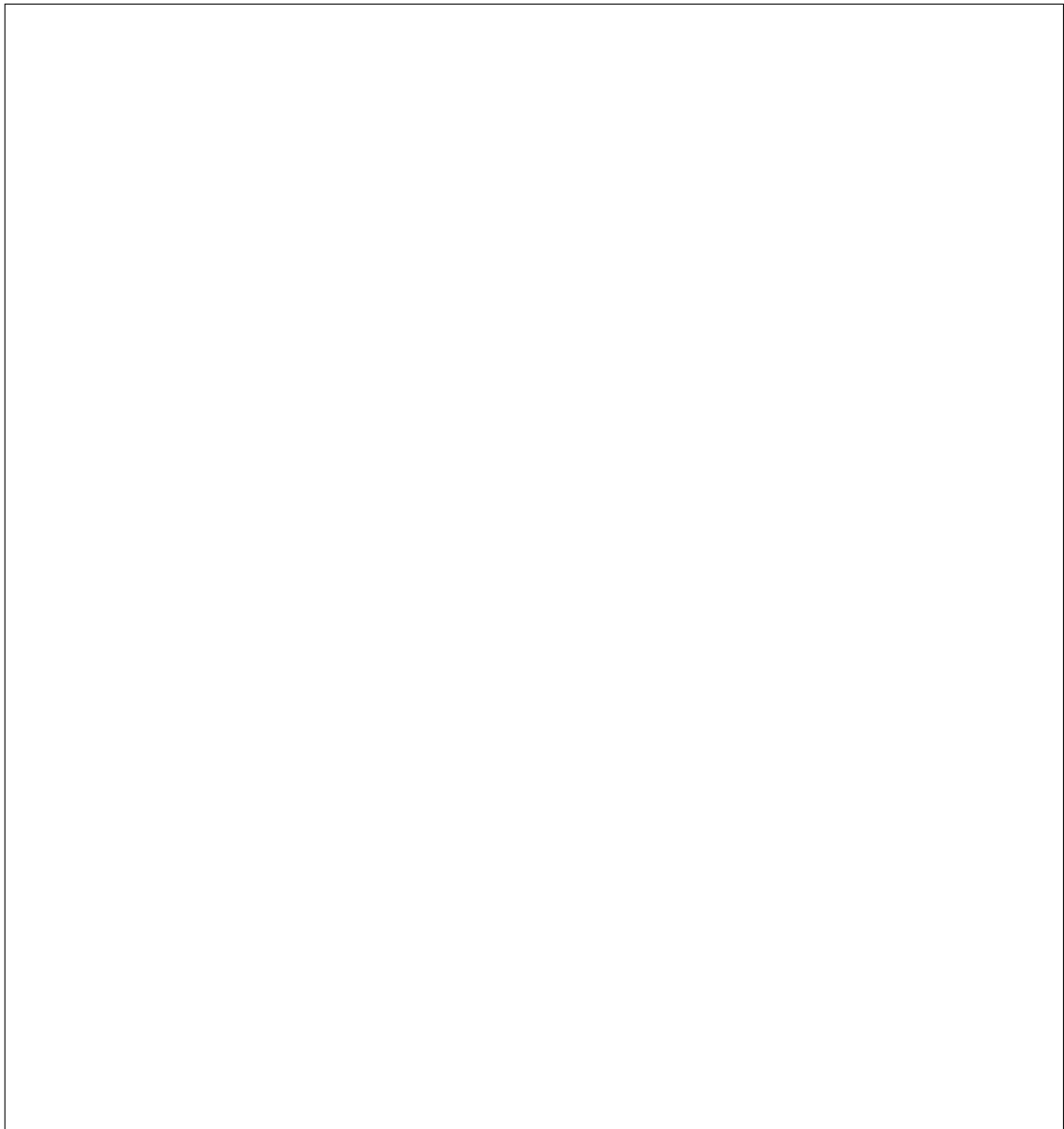
#### Returns

a  
dict  
hold  
ing  
net-  
worl  
con-  
fig-  
u-  
ra-  
tion  
in-  
for-  
ma-  
tion  
as-  
so-  
ci-  
ated

with this ironic or Neutron port.

`ironic.`  
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)



**Parameters**

**task**  
 a  
 task  
 con-  
 tain-  
 ing  
 the  
 Node  
 ob-  
 ject.

**Returns**

port  
 in-  
 for-  
 ma-  
 tion  
 as  
 a  
 dict

ironic.  
 Re-  
 turn  
 the  
 set  
 of

networks associated with the segments in that network.

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**

port  
 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  
 mat  
 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

**Paramet**

- **tas**  
a  
Task  
ager  
in-  
stan

- **net**  
UUI  
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/  
 Pur-  
 pose  
 fully  
 will  
 not  
 raise  
 any  
 ex-  
 cep-  
 tions  
 so  
 er-  
 ror  
 han-  
 dling  
 can  
 con-  
 tinue

Paramet

- **task**  
 a  
 Task  
 ager  
 in-  
 stan
- **network**  
 UUI  
 of  
 a  
 neu-  
 tron  
 net-  
 worl

ironic.

unbound state.

Un-  
bind  
a  
neu-  
tron  
port  
Re-  
mov  
a  
neu-  
tron  
port.  
bind  
ing  
pro-  
file  
and  
host  
ID  
so  
that  
it  
re-  
turn  
to  
an

## Paramet

- **por**  
Neu  
tron  
port  
ID.
- **cli**  
Op-  
tiona  
a  
Neu  
tron  
clien  
ob-  
ject.
- **con**  
(ir  
com

con  
 Req  
 re-  
 ques  
 con-  
 text

- **res**  
 re-  
 set  
 mac  
 ad-  
 dres

**Raises**  
 Net-  
 worl  
 Er-  
 ror

ironic.

Un-  
 date  
 a  
 neu-  
 tron  
 port

Uses  
 neu-  
 tron  
 clien  
 from  
 conf  
 clien  
 to  
 up-  
 date  
 a  
 neu-  
 tron  
 clien  
 an  
 un-  
 boun  
 state

**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-  
tiona  
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  
 UI

**Raises**  
 Miss  
 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-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-

this port.

Valu  
for  
miss  
ing  
or  
du-  
pli-  
cate  
net-  
worl  
  
ironic.  
Che  
that  
port  
con-  
tain  
enou  
in-  
for-  
ma-  
tion  
for  
de-  
ploy  
  
Neu  
tron  
net-  
worl  
in-  
ter-  
face  
re-  
quir  
that  
lo-  
cal\_  
field  
is  
fille  
be-  
fore  
we  
can  
use

Paramet

- nod

Ironi  
node  
ob-  
ject.

- **port**  
Ironi  
port  
ob-  
ject.

**Returns**

True  
if  
port  
info  
is  
valid  
False  
oth-  
er-  
wise

`ironic.`

Wait  
for  
neu-  
tron  
ager  
to  
be-  
com  
tar-  
get  
state

**Paramet**

- **cli**  
A  
Neu  
tron  
clien  
ob-  
ject.

- **hos**  
Age  
host

•
   
 tar
   
 up:
   
 wait
   
 for
   
 up
   
 sta-
   
 tus,
   
 dow
   
 wait
   
 for
   
 dow
   
 sta-
   
 tus

Returns
   
 bool
   
 in-
   
 di-
   
 cate
   
 the
   
 ager
   
 state
   
 matc
   
 para
   
 valu
   
 tar-
   
 get\_

Raises
   
 ex-
   
 cep-
   
 tion.
   
 if
   
 tar-
   
 get\_
   
 is
   
 not
   
 valid

Raises
   
 ex-
   
 cep-
   
 tion.
   
 if
   
 host
   
 sta-
   
 tus
   
 didn
   
 matc
   
 the
   
 re-
   
 quir

tempts.

sta-  
tus  
af-  
ter  
max  
retry  
at-  
  
 ironic.  
Wait  
for  
port  
sta-  
tus  
to  
be  
the  
de-  
sired  
sta-  
tus

Paramet

- **cli**  
 A  
 Neu-  
tron  
 client  
 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  
 matc  
 the  
 re-  
 quir  
 valu  
 pass  
 by  
 para  
 sta-  
 tus.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 port  
 does  
 not  
 ex-  
 ist.

**Raises**

ex-  
 cep-  
 tion.  
 if  
 port  
 sta-  
 tus  
 didn  
 matc  
 the  
 re-  
 quir  
 sta-  
 tus

tempts.

## ironic.common.nova module

af-  
ter  
max  
retry  
at-

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  
state  
char
- **tar**  
Tar-

for testing purposes).

ironic.common.policy module

gete  
 pow  
 state  
 char  
 i.e  
 POV  
 or  
 POV  
 Returns  
 A  
 bool  
 whic  
 in-  
 di-  
 cate  
 if  
 the  
 pow  
 up-  
 date  
 was  
 ex-  
 e-  
 cute  
 suc-  
 cess  
 fully  
 (mai  
 Pol-  
 icy  
 En-  
 gine  
 For  
 Iron  
 ironic.  
 A  
 shor  
 cut  
 for  
 pol-  
 icy.E  
 Che  
 au-  
 tho-



ception if the rule is not defined. Always returns true if CONF.auth\_strategy is not keystone.

or False.

riza-  
tion  
of  
a  
rule  
agai  
the  
tar-  
get  
and  
cre-  
den-  
tials  
and  
raise  
an  
ex-

ironic.  
A  
shor  
cut  
for  
pol-  
icy.F  
  
Che  
au-  
tho-  
riza-  
tion  
of  
a  
rule  
agai  
the  
tar-  
get  
and  
cre-  
den-  
tials  
and  
re-  
turn  
True

ironic.  
Pro-  
vide  
ac-

cess  
 to  
 the  
 sin-  
 gle  
 in-  
 stan-  
 of  
 Pol-  
 icy  
 en-  
 forc

ironic.

ironic.

Syn-  
 chro  
 ini-  
 tial-  
 izes  
 the  
 pol-  
 icy  
 en-  
 forc

Paramet

- **pol**  
 Cus-  
 tom  
 pol-  
 icy  
 file  
 to  
 use,  
 if  
 none  
 is  
 spec  
 i-  
 fied,  
*CON*  
 will  
 be  
 used

- **rul**

first instantiation.

De-  
fault  
dic-  
tio-  
nary  
/  
Rule  
to  
use.  
It  
will  
be  
con-  
sid-  
ered  
just  
in  
the

- **def**  
De-  
fault  
rule  
to  
use,  
CON  
will  
be  
used  
if  
none  
is  
spec  
i-  
fied.

- **use**  
When  
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  
 run-  
 ning  
 on.  
 By

fault host will be set to 0.0.0.0, but specifying host name / address usage is highly recommended.

set in `osprofiler.initializer.init_from_conf`.

## Raises

**Typ**  
in  
case  
of  
in-  
valid  
con-  
nec-  
tion  
strin  
for  
a  
no-  
ti-  
fier  
back  
end,  
whic  
is

ironic.  
Wra  
the  
OS-  
Pro-  
filer  
trace  
dec-  
o-  
ra-  
tor  
Wra  
the  
OS-  
Pro-  
filer  
trace  
dec-  
o-  
ra-  
tor  
so  
that  
it  
will  
not  
try

unless OSProfiler is present and enabled in the config

to  
 patc  
 the  
 class

Paramet

- **nam**  
 The  
 nam  
 of  
 ac-  
 tion.  
 For  
 ex-  
 am-  
 ple,  
 wsg  
 rpc,  
 db,  
 etc..
- **kwa**  
 Any  
 othe  
 key-  
 wor  
 args  
 used  
 by  
 pro-  
 filer.

ironic.common.pxe\_utils module

**class** i  
 Base  
*irc*  
*dri*  
*moo*  
*ima*  
*Ima*

ironic.

ironic.

ironic.

ironic.

Build  
the  
PXE  
con-  
fig  
op-  
tions  
for  
a  
node

This  
meth  
build  
the  
PXE  
boot  
op-  
tions  
for  
a  
node  
give  
all  
the  
re-  
quir  
pa-  
ram-  
e-  
ters.

The  
op-  
tions  
shou  
then  
be  
pass  
to  
pxe\_  
to  
cre-  
ate

the  
 ac-  
 tual  
 con-  
 fig  
 files

Paramet

- **task**  
 A Taskager object
- **pxe**  
 a dict of values to set on the configuration file
- **server**  
 if True build service model pxe configuration for netbooted user image



deployment image kernel and ramdisk info to PXE options.

ments.

and  
 skip  
 addi

• **ipxe**  
 De-  
 fault  
 false  
 bool  
 to  
 in-  
 di-  
 cate  
 if  
 ipxe  
 is  
 in  
 use  
 by  
 the  
 calle

• **ramdisk**  
 the  
 pa-  
 ram-  
 e-  
 ters  
 to  
 be  
 pass  
 to  
 the  
 ram  
 as  
 ker-  
 nel  
 com  
 line  
 ar-  
 gu-

**Returns**  
 A  
 dic-  
 tio-  
 nary  
 of  
 pxe

op-  
 tions  
 to  
 be  
 used  
 in  
 the  
 pxe  
 boot  
 file  
 tem-  
 plate

ironic.

ironic.

Fetc  
 the  
 nec-  
 es-  
 sary  
 ker-  
 nels  
 and  
 rame  
 for  
 the  
 in-  
 stan

ironic.  
 Clea  
 up  
 the  
 TFT  
 en-  
 vi-  
 ron-  
 men  
 for  
 the  
 task  
 node

Paramet  
 tas  
 A

Task  
ager  
in-  
stan

ironic

Clea  
PXE  
en-  
vi-  
ron-  
men  
of  
all  
the  
im-  
ages  
in  
im-  
ages

Clea  
up  
the  
PXE  
en-  
vi-  
ron-  
men  
for  
the  
men  
tion  
im-  
ages  
in  
im-  
ages

Paramet

- **tas**  
a  
Task  
ager  
ob-  
ject

- **ima**  
A

(kernel, ramdisk, etc) and values are a tuple of identifier and absolute path.

dic-  
tio-  
nary  
of  
im-  
ages  
who  
keys  
are  
the  
im-  
age  
nam  
to  
be  
clea  
up

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-  
dres  
links

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.

## Parameters

- **task**  
A Task Manager instance
- **pxe**  
A dictionary with the PXE configuration

cific template will be used.

ironic.

Retrieving the DHCP PXE boot options

Parameters

- task**  
 A Task agent instance
- ipxe**  
 De-

method for DHCP server configuration.

the node. If [pxe]ip\_version is set to 6, then this option has no effect as url\_boot form is required by DHCPv6 standards.

fault  
false  
bool  
that  
sig-  
nals  
if  
iPXE  
for-  
mat-  
ting  
shou  
be  
re-  
turn  
by  
the

- **url**  
De-  
fault  
false  
bool  
to  
in-  
form  
the  
meth  
if  
a  
URI  
shou  
be  
re-  
turn  
to  
boot

- **ip\_**  
The  
IP  
ver-  
sion  
of  
op-  
tions  
to

sion. Default to [pxe]ip\_version. Possible options are integers 4 or 6.

tions to be set.

re-  
 turn  
 as  
 val-  
 ues  
 dif-  
 fer  
 by  
 IP  
 ver-

Returns

Dic-  
 tio-  
 nary  
 to  
 be  
 sent  
 to  
 the  
 net-  
 work  
 ing  
 ser-  
 vice  
 de-  
 scrib-  
 ing  
 the  
 DHCP  
 op-

ironic.

Gen-  
 er-  
 ate  
 the  
 path  
 for  
 TFTP  
 files  
 for  
 de-  
 ploy  
 or  
 res-  
 cue  
 im-  
 ages



rescue) ramdisk.

#### Paramet

- **node**  
a  
node  
ob-  
ject
- **mode**  
La-  
bel  
in-  
di-  
cat-  
ing  
a  
de-  
ploy  
or  
res-  
cue  
op-  
er-  
a-  
tion  
be-  
ing

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy

operation is being carried out.

cue\_kernel, rescue\_ramdisk) and values are the absolute paths of them.

•
   
**ipxe**
  
 A
   
 de-
   
 fault
   
 Fals
   
 bool
   
 valu
   
 to
   
 tell
   
 the
   
 meth
   
 if
   
 the
   
 calle
   
 is
   
 us-
   
 ing
   
 iPXE
   
**Returns**
  
 a
   
 dic-
   
 tio-
   
 nary
   
 who
   
 keys
   
 are
   
 the
   
 nam
   
 of
   
 the
   
 im-
   
 ages
   
 (de-
   
 ploy
   
 de-
   
 ploy
   
 or
   
 res-
   
**Raises**
  
 Miss
   
 ing-
   
 Pa-
   
 ram-
   
 e-
   
 ter-
   
 Valu
   
 if

de-  
 ploy  
 or  
 res-  
 cue\_  
 is  
 miss  
 ing  
 in  
 node  
 drive

ironic.

Gen  
 er-  
 ate  
 the  
 path  
 for  
 TFT  
 files  
 for  
 in-  
 stan  
 re-  
 latec  
 im-  
 ages

This  
 meth  
 gen-  
 er-  
 ates  
 the  
 path  
 for  
 in-  
 stan  
 ker-  
 nel  
 and  
 in-  
 stan  
 ram  
 This  
 meth  
 also  
 up-

dates the node, so caller should already have a non-shared lock on the node.

**Paramet**

• **task**  
 A Task object represents a task in a task manager. It contains information about the task, including its name, its priority, and its context.

• **ipxe**  
 Default IPXE boot loader. If the boot loader is not set, the default is ipxe. The ipxe boot loader is used by the bootloader to load the kernel.

**Returns**  
 a dictionary of information about the task, including the task's name, its priority, and its context. The dictionary is returned as a dictionary of the following keys:

are the absolute paths of them. If its a whole disk image or node is configured for localboot, it returns an empty dictionary.

and  
val-  
ues

ironic.

ironic.

Get  
href  
and  
tftp  
path  
for  
de-  
ploy  
or  
res-  
cue  
ker-  
nel  
and  
ram

## Paramet

- **nod**  
UUI  
of  
the  
node
- **dri**  
Nod  
drive  
dict
- **mod**  
A  
la-  
bel  
to  
in-  
di-  
cate  
whe

ing requested. Supported values are deploy rescue. Defaults to deploy, indicating deploy paths will be returned.

path  
for  
de-  
ploy  
or  
res  
cue  
ram  
are  
be-

- ipx**  
A  
de-  
fault  
Fals  
bool  
valu  
to  
tell  
the  
meth  
if  
the  
calle  
is  
us-  
ing  
iPX

Returns

a  
dic-  
tio-  
nary  
who  
keys  
are  
de-  
ploy  
and  
de-  
ploy  
or  
res-  
cue\_  
and  
res-  
cue\_

whose values are the absolute paths to them.

and

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  
 for  
 the  
 node  
 PXE  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 file.

Paramet

- **node**  
 the  
 UI  
 of  
 the  
 node
- **ipx**  
 A  
 de-  
 fault  
 Fals  
 bool  
 valu  
 to  
 tell  
 the  
 meth  
 if  
 the  
 calle  
 is  
 us-  
 ing  
 iPX

Returns

The  
 path  
 to  
 the  
 node  
 PXE  
 con-  
 fig-



u-  
ra-  
tion  
file.

ironic.  
Re-  
turn  
the  
di-  
rec-  
tory  
whe  
the  
con-  
fig  
files  
and  
im-  
ages  
will  
live.

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
iPXE
tem-
plate
gen-
er-
a-
tion.

ironic.
Re-
turn
true
if
ipxe
is
set.

Parameter
task
A
Task
ager
ob-
ject

Returns
bool
true
if
[pxe
is
con-
fig-
ured
or
if
the
task
drive
in-
stan-
is
the
iPXE
drive

ironic.
Gets

```

information for this driver to deploy images to, or rescue, the node.

#### Parameters

- **node**  
a single Node
- **model**  
Label indicating a deployment

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

or  
res-  
cue  
op-  
er-  
a-  
tion  
be-  
ing

Returns

A  
dict  
with  
the  
drive  
val-  
ues.

Raises

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu

ironic.

Pre-  
pare  
the  
con-  
fig  
file  
for  
PXE  
boot

Paramet

- tas  
a  
task  
from  
Task  
ager

uration file.

- **ima**  
a  
dict  
of  
val  
ues  
of  
in-  
stan  
im-  
age  
meta  
data  
to  
set  
on  
the  
con-  
fig-
- **isc**  
if  
boot  
is  
from  
an  
iSCS  
vol-  
ume  
or  
not.
- **ram**  
if  
the  
boot  
is  
to  
a  
rame  
con-  
fig-  
u-  
ra-  
tion.
- **ipx**  
De-

fault  
false  
bool  
to  
in-  
di-  
cate  
if  
ipxe  
is  
in  
use  
by  
the  
called

Returns

Non

ironic.  
Che  
if  
boot  
pa-  
ram-  
e-  
ters  
are  
valid  
for  
trust  
boot

ironic.common.raid module

ironic.

Fil-  
ter  
the  
tar-  
get  
raid  
con-  
fig  
base  
on  
root  
vol-  
ume

config based on condition whether the root volume will be created or not.

#### Parameters

- **node**  
a node object
- **create\_root\_volume**  
A boolean default value True governing if the root volume is re-

root volumes will be filtered out.

else non-root volumes will be filtered out.

ter skipping root volume and/or non-root volumes.

turn  
 else  
 •  
 cre  
 A  
 bool  
 de-  
 fault  
 valu  
 True  
 gov-  
 ern-  
 ing  
 if  
 the  
 non  
 root  
 vol-  
 ume  
 is  
 re-  
 turn

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 node  
 is  
 miss  
 ing  
 or  
 was  
 foun  
 to  
 be  
 emp  
 af-

Returns

It  
 will  
 re-  
 turn



that is passed.

fil-  
 terec  
 tar-  
 get\_

ironic  
 Get  
 log-  
 i-  
 cal  
 disk  
 prop  
 er-  
 ties  
 from  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 sche

This  
 meth  
 read  
 the  
 log-  
 i-  
 cal  
 prop  
 er-  
 ties  
 and  
 their  
 tex-  
 tual  
 de-  
 scrip  
 tion  
 from  
 the  
 sche

**Paramet**  
**rai**  
 A  
 dic-  
 tio-  
 nary  
 whic  
 is

may be specified for the logical disk.

textual description for them as values.

the  
sche  
to  
be  
used  
for  
get-  
ting  
prop  
er-  
ties  
that

Returns

A  
dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
log-  
i-  
cal  
disk  
prop  
er-  
ties  
as  
keys  
and  
a

ironic.  
Up-  
date  
the  
node  
in-  
for-  
ma-  
tion  
base  
on  
the  
RAI  
con-  
fig.  
This

scheduling purposes (through `properties[capabilities]` and `properties[local_gb]`) and deploying purposes (using `properties[root_device]`).

#### Parameters

- **node**  
a node object
- **raise**  
The dictionary containing the current RAI configuration.

**Raises**  
In-

is malformed.

valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
raid,  
has  
more  
than  
one  
root  
vol-  
ume  
or  
if  
node

ironic.

Val-  
i-  
date  
the  
RAI  
con-  
fig-  
u-  
ra-  
tion  
pass  
us-  
ing  
JSO  
sche  
This  
meth  
val-  
i-  
date  
a  
RAI  
con-  
fig-  
u-  
ra-  
tion  
agai  
a

schema.

RAI  
con-  
fig-  
u-  
ra-  
tion

Paramet

- **rai**
  
 A  
dic-  
tio-  
nary  
con-  
tain-  
ing  
RAI  
con-  
fig-  
u-  
ra-  
tion  
in-  
for-  
ma-  
tion

- **rai**
  
 A  
dic-  
tio-  
nary  
which  
is  
the  
sche  
to  
be  
used  
for  
val-  
i-  
da-  
tion.

Raises  
In-  
valid

tion fails.

ironic.common.release\_mappings module

ironic.

Gets the supported version for all objects

Supported versions are from the RELEASE

Parameter

- **rel** a list of

default).

jects are returned (the default).

re-  
lease  
nam  
if  
emp  
ver-  
sion  
from  
all  
re-  
lease  
are  
re-  
turn  
(the

- **obj**  
a  
list  
of  
nam  
of  
ob-  
jects  
of  
in-  
ter-  
est.  
If  
emp  
ver-  
sion  
of  
all  
ob-

**Returns**  
a  
dic-  
tio-  
nary  
whe  
the  
key  
is  
the  
ob-  
ject  
nam

supported versions.

ironic.common.rpc module

and  
 the  
 valu  
 is  
 a  
 set  
 of

**class** i  
 Base  
 osl  
 ser  
 Ser

**deseria**  
 De-  
 se-  
 ri-  
 al-  
 ize  
 a  
 dic-  
 tio-  
 nary  
 into  
 a  
 re-  
 ques  
 con-  
 text.

**Parame**  
**ctx**  
 Re-  
 ques  
 con-  
 text  
 dic-  
 tio-  
 nary

**Returns**  
 De-  
 se-  
 ri-  
 al-  
 ized  
 form



of  
 en-  
 tity

**deseriali**

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

- 

**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-  
 nary

**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
 man
 ager
 is
 not
 dere
 is-
 tereo
 whe
 it
 is
 shut
 dow

**start ( )**
 Star
 a
 ser-
 vice

**stop ()**  
 Stop  
 a  
 ser-  
 vice

**Parame**  
**gra**  
 in-  
 di-  
 cate  
 whe  
 to  
 wait  
 for  
 all  
 threa  
 to  
 fin-  
 ish  
 or  
 ter-  
 mi-  
 nate  
 then  
 in-

stantly

**ironic.common.service module**

ironic.

ironic.

**ironic.common.states module**

Map  
 ping  
 of  
 bare  
 meta  
 node  
 state  
 Set-  
 ting  
 the  
 node  
 pow

on the power state retrieved from the driver for the node, the state is set to `POWER_ON` or `POWER_OFF`, accordingly. Should this fail, the *power\_state* value is left unchanged, and the node is placed into maintenance mode.

the current state unchanged. The node is NOT placed into maintenance mode in this case.

is  
han-  
dled  
by  
the  
con-  
duc-  
tors  
pow  
syn-  
chro  
niza-  
tion  
threa-  
Base

The  
*pow*  
can  
also  
be  
set  
man-  
u-  
ally  
via  
the  
API  
A  
fail-  
ure  
to  
char-  
the  
state  
leav

ironic.  
Nod  
is  
suc-  
cess-  
fully  
de-  
ploy  
and  
as-  
so-

ci-  
ated  
with  
an  
in-  
stan

`ironic.`

Nod  
faile  
to  
com  
plete  
the  
adop  
tion  
pro-  
cess

This  
state  
is  
the  
re-  
sult-  
ing  
state  
of  
a  
node  
that  
faile  
to  
com  
plete  
adop  
tion.  
po-  
ten-

tially due to invalid or incompatible information being defined for the node.

`ironic.`

Nod  
is  
be-  
ing  
adop

This  
pro-  
vi-  
sion  
state

TIVE state to permit designation of nodes as being managed by IroniC, however deployed previously by external means.

is  
in-  
tend  
for  
use  
to  
mov  
a  
node  
from  
MA  
AGE  
ABL  
ABL  
to  
AC-

`ironic.`  
Nod  
is  
avai  
able  
for  
use  
and  
sche  
ing.

This  
state  
is  
re-  
plac  
ing  
the  
NOS  
TAT  
state  
used  
prior  
to  
Kilo

`ironic.`  
Nod  
faile  
clea  
ing.  
This  
re-  
quir  
op-



er-  
a-  
tor  
in-  
ter-  
ven-  
tion  
to  
re-  
solv

`ironic.`  
Nod  
is  
be-  
ing  
au-  
to-  
mat-  
i-  
cally  
clea  
to  
pre-  
pare  
it  
for  
pro-  
vi-  
sion  
ing.

`ironic.`  
Nod  
is  
wait  
ing  
for  
a  
clea  
step  
to  
be  
fin-  
ishe

This  
will  
be  
the  
node  
*pro-*  
*vi-*

a cleaning step.

sion.  
whil  
the  
node  
is  
wait  
ing  
for  
the  
drive  
to  
fin-  
ish

ironic.  
Nod  
tear  
dow  
was  
suc-  
cess  
ful.  
  
In  
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

target\_provision\_state.

and  
neve  
rep-  
re-  
sent  
in

ironic.  
State  
in  
whic  
node  
dele  
tion  
is  
al-  
lowe

ironic.  
Nod  
is  
ac-  
tivel  
be-  
ing  
torn  
dow

ironic.  
Nod  
was  
suc-  
cess  
fully  
de-  
ploy  
This  
is  
main  
a  
tar-  
get  
pro-  
vi-  
sion  
state  
used  
dur-  
ing  
de-  
ploy

deployed node should go to ACTIVE status.

men  
 A  
 suc-  
 cess  
 fully  
  
 ironic.  
 Nod  
 de-  
 ploy  
 men  
 faile  
  
 ironic.  
 Nod  
 is  
 read  
 to  
 re-  
 ceiv  
 a  
 de-  
 ploy  
 re-  
 ques  
 or  
 is  
 cur-  
 rentl  
 be-  
 ing  
 de-  
 ploy  
  
 A  
 node  
 will  
 have  
 its  
*pro-  
 vi-  
 sion*  
 set  
 to  
 DE-  
 PLC  
 ING  
 brie  
 be-  
 fore  
 it  
 re-

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).

deployment.

ceiv  
its

ironic.  
Nod  
is  
wait  
ing  
to  
be  
de-  
ploy  
  
This  
will  
be  
the  
node  
*pro-  
vi-  
sion*  
whil  
the  
node  
is  
wait  
ing  
for  
the  
drive  
to  
fin-  
ish

ironic.  
Nod  
is  
en-  
rolle  
  
This  
state  
in-  
di-  
cate  
that  
Iron  
is  
awa  
of

a  
node  
but  
is  
not  
man  
ag-  
ing  
it.

ironic.  
An  
er-  
ror  
oc-  
curr  
dur-  
ing  
node  
pro-  
cess  
ing.

The  
*last\_*  
at-  
tribu  
of  
the  
node  
de-  
tails  
shou  
con-  
tain  
an  
er-  
ror  
mes-  
sage

ironic.  
State  
whe  
API  
look  
are  
per-  
mit-  
ted  
with  
fast  
track

node shall transition to MANAGEABLE state. For asynchronous inspection, node shall transition to INSPECTWAIT state.

en-  
able  
  
ironic.  
Nod  
in-  
spec  
tion  
faile  
  
ironic.  
Nod  
is  
un-  
der  
in-  
spec  
tion.  
  
This  
is  
the  
pro-  
vi-  
sion  
state  
used  
whe  
in-  
spec  
tion  
is  
start  
A  
suc-  
cess  
fully  
in-  
spec

ironic.  
Nod  
is  
un-  
der  
in-  
spec  
tion.  
  
This  
is  
the

cessfully inspected node shall transition to MANAGEABLE state.

pro-  
vi-  
sion  
state  
used  
whe  
an  
asyn  
chro  
in-  
spec  
tion  
is  
in  
prog  
A  
suc-

ironic.  
State  
whe  
API  
look  
are  
nor-  
mall  
al-  
lowe  
for  
node

ironic.  
Nod  
is  
in  
a  
man  
age-  
able  
state  
  
This  
state  
in-  
di-  
cate  
that  
Iron  
has  
ver-  
i-  
fied,



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).

target\_\*\_state fields when there is no target.

at  
least  
once  
that  
it  
had  
suf-  
fi-  
cien

ironic.  
No  
state  
in-  
for-  
ma-  
tion.  
This  
state  
is  
used  
with  
pow  
to  
rep-  
re-  
sent  
a  
lack  
of  
know  
edge  
of  
pow  
state  
and  
in

ironic.  
Nod  
is  
pow  
ered  
off.

ironic.  
Nod  
is  
pow

via the REST API.

ered  
 on.  
 ironic.  
 Nod  
 is  
 re-  
 boot  
 ing.  
 ironic.  
 Nod  
 is  
 to  
 be  
 re-  
 built  
 This  
 is  
 not  
 used  
 as  
 a  
 state  
 but  
 rathe  
 as  
 a  
 verb  
 whe  
 char  
 ing  
 the  
 node  
 pro-  
 vi-  
 sion  
 ironic.  
 Nod  
 is  
 in  
 res-  
 cue  
 mod  
 ironic.  
 Nod  
 res-  
 cue  
 faile

rescuing the node.

```

ironic.
    Node
    is
    wait
    ing
    on
    an
    ex-
    ter-
    nal
    call-
    back

    This
    will
    be
    the
    node
    pro-
    vi-
    sion
    while
    the
    node
    is
    wait
    ing
    for
    the
    drive
    to
    fin-
    ish

```

```

ironic.
    Node
    is
    in
    pro-
    cess
    of
    be-
    ing
    res-
    cued

```

```

ironic.
    Node
    is
    in
    the
    pro-

```

cess  
of  
soft  
pow  
off.

ironic.  
Nod  
is  
re-  
boot  
ing  
grac  
fully

ironic.  
State  
that  
will  
not  
tran-  
si-  
tion  
un-  
less  
re-  
ceiv  
ing  
a  
re-  
ques

ironic.  
State  
that  
can-  
not  
be  
re-  
sum  
once  
a  
con-  
duc-  
tor  
dies  
  
If  
a  
node  
gets  
stuc  
with

when executing task), node will be moved to fail state.

one  
of  
thes  
state  
for  
som  
rea-  
son  
(eg.  
con-  
duc-  
tor  
goes  
dow

```
ironic.  
    Nod  
    un-  
    res-  
    cue  
    faile
```

```
ironic.  
    Nod  
    is  
    be-  
    ing  
    re-  
    store  
    from  
    res-  
    cue  
    mod  
    (to  
    ac-  
    tive  
    state
```

```
ironic.  
    State  
    that  
    can  
    be  
    char  
    with  
    out  
    ex-  
    ter-  
    nal  
    re-  
    ques
```

ironic.  
 Tran  
 si-  
 tiona  
 state  
 in  
 whic  
 we  
 al-  
 low  
 up-  
 dat-  
 ing  
 a  
 node

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 i  
 {tar-  
 get  
 strin  
 used  
 by  
 the  
 API  
 in-  
 ter-  
 nal  
 verb  
 This

used to support renaming these actions.

pro-  
vide  
a  
ref-  
er-  
ence  
set  
of  
sup-  
port  
ac-  
tions  
and  
in  
the  
fu-  
ture  
may  
be

`ironic.`  
Nod  
pow  
man  
age-  
men  
cre-  
den-  
tials  
are  
be-  
ing  
ver-  
i-  
fied.

`ironic`  
Use  
to  
log  
whe  
en-  
ter-  
ing  
a  
state

`ironic.`  
Use  
to  
log  
whe

ironic.common.swift module

a  
state  
is  
ex-  
ited.

**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

Paramete

- **con**  
The  
nam  
of  
the



contains  
for  
the  
object.

- **obj**  
The  
name  
of  
the  
object  
in  
Swift

- **file**  
The  
file  
to  
upload  
as  
the  
object  
data

- **obj**  
the  
headers  
for  
the  
object  
to  
pass  
to  
Swift

**Returns**  
The  
Swift  
UUID  
of  
the  
object

**Raises**

Swi  
Op-  
er-  
a-  
tion  
if  
any  
op-  
er-  
a-  
tion  
with  
Swi  
fails

**delete\_**

Dele  
the  
give  
Swi  
ob-  
ject.

**Parame**

- con**  
The  
nam  
of  
the  
con-  
taine  
in  
whic  
Swi  
ob-  
ject  
is  
plac
- obj**  
The  
nam  
of  
the  
ob-  
ject  
in  
Swi

to  
 be  
 dele

**Raises**

Swi  
 b-  
 ject-  
 Not-  
 Four  
 ror,  
 if  
 ob-  
 ject  
 is  
 not  
 foun  
 in  
 Swi

**Raises**

Swi  
 Op-  
 er-  
 a-  
 tionl  
 if  
 op-  
 er-  
 a-  
 tion  
 with  
 Swi  
 fails

**get\_tem**

Re-  
 turn  
 the  
 temp  
 url  
 for  
 the  
 give  
 Swi  
 ob-  
 ject.

**Parame**

•  
**con**  
 The

name  
 of  
 the  
 con-  
 taine  
 in  
 whic  
 Swit  
 ob-  
 ject  
 is  
 plac

- **obj**  
 The  
 nam  
 of  
 the  
 Swit  
 ob-  
 ject.

- **tim**  
 The  
 time  
 out  
 out  
 in  
 sec-  
 onds  
 af-  
 ter  
 whic  
 the  
 gen-  
 er-  
 ated  
 url  
 shou  
 ex-  
 pire.

**Returns**  
 The  
 temp  
 url  
 for  
 the  
 ob-  
 ject.

**Raises**

Swi
 Op-
 er-
 a-
 tion
 if
 any
 op-
 er-
 a-
 tion
 with
 Swi
 fails

**head\_ob**
 Re-
 triev
 the
 in-
 for-
 ma-
 tion
 abou
 the
 give
 Swi
 ob-
 ject.

**Parame**

- con**
 The
 nam
 of
 the
 con-
 taine
 in
 whic
 Swi
 ob-
 ject
 is
 plac
- obj**
 The
 nam

of  
 the  
 ob-  
 ject  
 in  
 Swi

**Returns**

The  
 in-  
 for-  
 ma-  
 tion  
 about  
 the  
 ob-  
 ject  
 as  
 re-  
 turn  
 by  
 Swi  
 clien  
 head  
 call.

**Raises**

Swi  
 Op-  
 er-  
 a-  
 tionl  
 if  
 op-  
 er-  
 a-  
 tion  
 with  
 Swi  
 fails

**update\_**

Up-  
 date  
 the  
 meta  
 data  
 of  
 a  
 give  
 Swi  
 ob-  
 ject.

Param

- **con**  
 The  
 nam  
 of  
 the  
 con-  
 taine  
 in  
 whic  
 Swi  
 ob-  
 ject  
 is  
 plac

- **obj**  
 The  
 nam  
 of  
 the  
 ob-  
 ject  
 in  
 Swi

- **obj**  
 the  
 head  
 ers  
 for  
 the  
 ob-  
 ject  
 to  
 pass  
 to  
 Swi

Raises

Swi  
 Op-  
 er-  
 a-  
 tionl  
 if  
 op-  
 er-

a-  
tion  
with  
Swi  
fails

ironic.

## ironic.common.utils module

Util-  
i-  
ties  
and  
help  
func  
tions

ironic.

Che  
a  
di-  
rec-  
tory  
is  
us-  
able

This  
func  
tion  
can  
be  
used  
by  
drive  
to  
chec  
that  
di-  
rec-  
to-  
ries  
they  
need  
to  
writ  
to

are usable. This should be called from the drivers init function. This function checks that the directory exists and then calls `check_dir_writable` and `check_dir_free_space`. If `directory_to_check` is not



provided the default is to use the temp directory.

#### Parameters

- **dir**  
the  
di-  
rec-  
tory  
to  
check
- **req**  
amo  
of  
space  
to  
check  
for  
in  
MiB

#### Raises

Path  
Not-  
Found  
if  
di-  
rec-  
tory  
can  
not  
be  
found

#### Raises

Di-  
rec-  
to-  
ryNot  
if  
user  
is  
un-  
able  
to  
write  
to  
the  
di-  
rec-

tory

Raises

*Raises*  
 if  
 free  
 spac  
 is  
 <  
 re-  
 quir  
 spac

ironic.

ironic.  
 Con  
 ve-  
 nien  
 wrap  
 per  
 arou  
 os-  
 los  
 ex-  
 e-  
 cute  
 meth

Paramet

- **cmd**  
 Pass  
 to  
 pro-  
 ces-  
 su-  
 tils.6
- **use**  
 True  
 |  
 Fals  
 De-  
 fault  
 to  
 Fals  
 If  
 set  
 to

dard locale added to environment variables.

True  
 ex-  
 e-  
 cute  
 com  
 man  
 with  
 stan

**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

ref-  
er-  
ence

Parameters

- **path**  
path  
to  
file
- **compare**  
ref-  
er-  
ence  
con-  
tent  
to  
check  
agai
- **hash**  
hash  
ing  
algo  
from  
hash  
lib  
to  
use,  
de-  
fault  
is  
sha2

Returns

True  
if  
the  
hash  
of  
ref-  
er-  
ence  
con-  
tent  
is  
the  
same  
as

tent, False otherwise

the  
hash  
of  
files  
con-

ironic.  
Gets  
a  
min  
type  
of  
the  
give  
file.

ironic.  
Re-  
turn  
an  
up-  
date  
ca-  
pa-  
bil-  
ity  
strin

This  
meth  
up-  
date  
the  
orig  
i-  
nal  
(or  
cur-  
rent,  
ca-  
pa-  
bil-  
i-  
ties  
with  
the  
new  
ca-

pabilities. The original capabilities would typically be from a nodes properties[capabilities]. From new\_capabilities, any new capabilities are added, and existing capabilities may have their values updated. This updated capabilities string is returned.

Parameters

- **currentCapabilitiesString**  
 Current capabilities string
- **newCapabilitiesDictionary**  
 the dictionary of new capabilities to be updated

Returns

An updated capabilities string with new

Raises

ValueError, if current is malformed or

if  
 new.  
 is  
 not  
 a  
 dic-  
 tio-  
 nary  
  
 ironic.  
 Old  
 chec  
 for  
 valid  
 log-  
 i-  
 cal  
 node  
 nam  
  
 Re-  
 taine  
 for  
 com  
 pat-  
 i-  
 bil-  
 ity  
 with  
 RES  
 API  
 <  
 1.10

Nominall

- <http://en.wikipedia.org/wiki/Hos>
- <http://tools.ietf.org/html/rfc9>

umented in bug #1468508.

• <http://tools.ietf.org/html/rfc1>  
 In practice, this check has several shortcomings and errors that are more thorough documentation.

**Parameters**  
**hostname**  
 The host name to be validated.

**Returns**  
 True if valid, False if not.

ironic.  
 Check availability



a process by the kernel upon allocation request, and delays the execution until memory has been freed, or until it has timed out.

able  
sys-  
tem  
men  
ory  
and  
hold  
the  
de-  
ploy  
men  
pro-  
cess  
Eval  
u-  
ates  
the  
cur-  
rent  
sys-  
tem  
men  
ory  
avai  
able  
mea  
ing  
can  
be  
al-  
lo-  
cate  
to

This  
meth  
will  
is-  
sue  
a  
sleep  
if  
the  
amo  
of  
avai  
able  
men  
ory

This is configured using the `[DEFAULT]minimum_memory_wait_time` and the `[DEFAULT]minimum_memory_wait_retries`.

is raised upon insufficient memory.

is  
in-  
suf-  
fi-  
cent

**Parameters**  
**raise\_insufficient\_memory**  
De-  
fault  
False  
but  
if  
set  
to  
true  
an  
In-  
suf-  
fi-  
cent  
Memory  
Error  
ex-  
cep-  
tion

**Returns**  
True  
if  
the  
check  
has  
time  
out.  
Otherwise  
er-  
wise  
None  
is  
re-  
turn

**Raises**  
In-  
suf-  
fi-  
cent  
Memory  
Error

if  
the  
raise  
pa-  
ram-  
e-  
ter  
is  
set  
to  
True

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-  
adec  
i-  
mal  
dig-  
its.  
Dat-

apath 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
 Fals
 if
 not.

ironic.
 De-
 ter-
 mine
 if
 a
 log-
 i-
 cal
 nam
 is
 valid
 The
 log-
 i-
 cal
 nam
 may
 only
 con-
 sist
 of
 RFC
 un-
 re-
 serv
 char
 ac-
 ters,
 to

wit:

AL-  
PHA  
/  
DIG  
/  
-  
  
/  
.  
/  
-  
/  
~

ironic.  
Che  
no\_  
va-  
lid-  
ity  
  
Che  
if  
no\_  
valu  
that  
will  
be  
writ  
ten  
to  
en-  
vi-  
ron-  
men  
vari-  
able  
by  
iron  
pyth  
ager

is valid.

**Paramet**  
**no\_**  
the  
valu  
that  
re-  
quir

host names, IP addresses and domain names (with optional :port).

Returns

True if no\_port is valid, False otherwise.

ironic.

Mount a device file on specified i-fied location.

Parameters

- src** the path to the source file for mounting.

ing

- **des**  
the  
path  
when  
it  
need  
to  
be  
mou

- **arg**  
a  
tu-  
ple  
con-  
tain-  
ing  
the  
ar-  
gu-  
men  
to  
be  
pass  
to  
mou  
com  
man

**Raises**  
pro-  
ces-  
su-  
tils.I  
if  
it  
faile  
to  
run  
the  
pro-  
cess

ironic.  
Pars  
the  
in-  
stan  
ca-

ities are defined in the Flavor `extra_spec` and passed to Ironiic by the Nova Ironiic driver.

ity with Juno the Nova Ironiic driver is sending it as a string.

pa-  
bil-  
i-  
ties.

One  
way  
of  
hav-  
ing  
thes  
ca-  
pa-  
bil-  
i-  
ties  
set  
is  
via  
Nov  
whe  
the  
ca-  
pa-  
bil-

NOT  
Al-  
thou  
our  
API  
fully  
sup-  
port  
JSO  
field  
to  
main  
tain  
the  
back  
war  
com  
pat-  
i-  
bil-

**Paramet**  
**nod**  
a  
sin-



tionary or is malformed.

tionary.

gle  
 Nod  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 strin  
 is  
 not  
 a  
 dic-  
**Returns**  
 A  
 dic-  
 tio-  
 nary  
 with  
 the  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 if  
 foun  
 oth-  
 er-  
 wise  
 an  
 emp  
 dic-  
 ironic.  
 Pop  
 a  
 valu  
 from

a  
 dic-  
 tio-  
 nary  
 field  
 of  
 a  
 node

Paramet

- **nod**  
 Nod  
 ob-  
 ject.
- **col**  
 Nam  
 of  
 the  
 field  
 with  
 the  
 dic-  
 tio-  
 nary
- **fie**  
 Nest  
 field  
 nam
- **def**  
 The  
 de-  
 fault  
 valu  
 to  
 re-  
 turn

Returns

The  
 re-  
 mov  
 valu  
 or  
 the  
 de-

fault
 ironi
 c.
 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-
 ters
 to
 use
 whe
 ren-
 der-
 ing
- **is\_**
 whe
 tem-
 plate
 is
 file

or  
 strin  
 with  
 tem-  
 plate  
 it-  
 self

**Returns**

the  
 ren-  
 dere  
 tem-  
 plate  
 as  
 a  
 strin

ironic.

ironic.

Re-  
 mov  
 trail  
 ing  
 char  
 ac-  
 ters  
 from  
 a  
 strin  
 if  
 that  
 does  
 not  
 mak  
 it  
 emp

**Paramet**

- **val**  
 A  
 strin  
 valu  
 that  
 will  
 be  
 strip
-

cha
 Cha
 ac-
 ters
 to
 re-
 mov

Returns
 Strip
 valu

ironic.
 Set
 a
 valu
 in
 a
 dic-
 tio-
 nary
 field
 of
 a
 node

Paramet

- **nod**
 Nod
 ob-
 ject.
- **col**
 Nam
 of
 the
 field
 with
 the
 dic-
 tio-
 nary
- **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-  
 ple  
 con-  
 tain-  
 ing  
 the  
 ar-  
 gu-  
 men  
 to  
 be  
 pass  
 to  
 the  
 umo  
 com  
 man

Raises  
 pro-  
 ces-  
 su-  
 tils.I  
 if  
 it

failed
 to
 run
 the
 process

ironic.

ironic.
 Val-
 i-
 date
 an
 Ope
 Flow
 dat-
 a-
 p-
 ath\_
 and
 re-
 turn
 nor-
 mal-
 ized
 form

Che
 whe
 the
 sup-
 plied
 Ope
 Flow
 dat-
 a-
 p-
 ath\_
 is
 for-
 mall
 cor-
 rect
 and
 nor-
 mal-
 ize

it to all lower case.

Paramet
 dat

Ope  
 Flow  
 dat-  
 a-  
 p-  
 ath\_  
 to  
 be  
 val-  
 i-  
 date  
 and  
 nor-  
 mal-  
 ized

Returns

Nor-  
 mal-  
 ized  
 and  
 val-  
 i-  
 date  
 Ope  
 Flow  
 dat-  
 a-  
 p-  
 ath\_

Raises

In-  
 valid  
 Dat-  
 a-  
 p-  
 athI  
 If  
 an  
 Ope  
 Flow  
 dat-  
 a-  
 p-  
 ath\_  
 is  
 not  
 valid

ironic.  
 Val-  
 i-



lower case.

date  
 a  
 MA  
 ad-  
 dres  
 and  
 re-  
 turn  
 nor-  
 mal-  
 ized  
 form  
  
 Che  
 whe  
 the  
 sup-  
 plied  
 MA  
 ad-  
 dres  
 is  
 for-  
 mall  
 cor-  
 rect  
 and  
 nor-  
 mal-  
 ize  
 it  
 to  
 all

Paramet

add  
 MA  
 ad-  
 dres  
 to  
 be  
 val-  
 i-  
 date  
 and  
 nor-  
 mal-  
 ized

Returns

Nor-  
 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

**Parameters**

- **port**  
 TCP  
 port
- **port**  
 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.`

`ironic.`

Wra  
the  
ad-  
dres  
in  
squa  
brac  
ets  
if  
its  
an  
IPv6  
ad-  
dres

`ironic.`

### **ironic.common.wsgi\_service module**

#### **class i**

Base  
osl  
ser  
Ser  
  
Pro-  
vide  
abil-  
ity  
to

laun  
 iron  
 API  
 from  
 wsg  
 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-  
 tion.

**Returns**  
 Non

**stop ( )**  
 Stop  
 serv  
 ing  
 this  
 API

**Returns**  
 Non

**wait ( )**  
 Wai  
 for  
 the

ser-  
 vice  
 to  
 stop  
 serv  
 ing  
 this  
 API

**Returns**  
 Non

**Module contents**

ironic.conductor package

**Submodules**

ironic.conductor.allocations module

Func  
 tion-  
 al-  
 ity  
 re-  
 latec  
 to  
 al-  
 lo-  
 ca-  
 tions

ironic.

As-  
 sign  
 the  
 pre-  
 vi-  
 ously  
 al-  
 lo-  
 cate  
 node  
 to  
 the  
 node  
 al-  
 lo-

cation\_uuid for a previously allocated node.

Paramet

- **con**  
 an  
 ad-  
 min  
 con-  
 text
- **all**  
 an  
 al-  
 lo-  
 ca-  
 tion  
 ob-  
 ject  
 as-  
 so-  
 ci-  
 ated  
 with  
 the  
 node
-

node
 An
 ID
 of
 the
 node

Raises

Al-
 lo-
 ca-
 tion-
 Fail
 if
 the
 node
 does
 not
 matc
 the
 al-
 lo-
 ca-
 tion

Raises

Nod
 As-
 so-
 ci-
 ated
 if
 the
 node
 is
 al-
 read
 as-
 so-
 ci-
 ated
 with
 an-
 othe
 in-

stance or allocation.

Raises

In-
 stan
 As-
 so-
 ci-

other node as instance\_uuid.

Raises

Node not found if the node with the provided ID cannot be found.

ironic.Process the allocation. This call runs in a separate thread on a



the allocation and reserves one of them.

con-  
duc-  
tor.  
It  
finds  
suit-  
able  
node  
for

This  
call  
does  
not  
raise  
ex-  
cep-  
tions  
since  
its  
de-  
sign  
to  
work  
asyn-  
chro-

Paramet

- **con**  
an  
ad-  
min  
con-  
text
- **all**  
an  
al-  
lo-  
ca-  
tion  
ob-  
ject

ironic.

Ver-  
ify  
that

al-  
 lo-  
 ca-  
 tion  
 can  
 be  
 re-  
 mov  
 for  
 the  
 node

Paramet

- **node**  
 a  
 node  
 ob-  
 ject
- **all**  
 an  
 al-  
 lo-  
 ca-  
 tion  
 ob-  
 ject  
 as-  
 so-  
 ci-  
 ated  
 with  
 the  
 node

ionic.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  
 Load  
 Er-  
 ror  
 if  
 an  
 en-  
 able  
 drive  
 can-  
 not  
 be  
 load

**Raises**

Driv  
 Nam  
 Con  
 flict  
 if  
 a  
 clas-  
 sic  
 drive  
 and  
 a  
 dy-

have the same name.

nam  
drive  
are  
both  
en-  
able  
and

**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  
to  
this  
con-  
duc-  
tor.

Yiel  
tu-  
ples  
(nod  
drive  
con-  
duc-  
tor\_  
)  
whe  
is

means yielding (uuid, driver, conductor\_group), fields=[foo] means yielding (uuid, driver, conductor\_group, foo).

Parameters

- fields**
 list of fields to fetch in addition to uuid, driver and conductor\_group.
- kwargs**
 additional arguments to pass to dbapi when looking for nodes.

Returns

gen-  
er-  
a-  
tor  
yiel  
ing  
tu-  
ples  
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  
for  
node  
in  
a  
tran-  
si-

tory power state and nodes that are presently locked by the hostname of this conductor.

Un-  
der  
nor-  
mal  
op-  
er-  
a-  
tion,

is established for the conductors normal operation.

ironic.conductor.cleaning module



to  
 run  
 next  
 and  
 pass  
 con-  
 trol  
 into  
 do\_1

Paramet

tas  
 a  
 Task  
 ager  
 in-  
 stan  
 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

into the list of clean steps in the nodes driver\_internal\_info[clean\_steps]. Is None if there are no steps to execute.

lock

- steps**  
 The first clean step in the list to execute. This is the index (from 0)

ironic.  
 Internal RPC method to perform cleaning of a node

**Parameters**

- task\_id**  
 a TaskIdentifier instance with an ex-

mated cleaning (default). For more information, see the `clean_steps` parameter of `ConductorManager.do_node_clean()`.

ironic.  
In-  
ter-  
nal  
meth  
to  
abon  
an  
on-  
go-  
ing  
op-  
er-  
a-  
tion.

## Parameter

- **task**  
a  
Task

ironic.conductor.deployments module

ager  
 in-  
 stan  
 with  
 an  
 ex-  
 clu-  
 sive  
 lock  
 •  
 ste  
 The  
 nam  
 of  
 the  
 clea  
 step  
 Func  
 tion-  
 al-  
 ity  
 re-  
 latec  
 to  
 de-  
 ploy  
 ing  
 and  
 un-  
 de-  
 ploy  
 ing.  
 ironic.  
 Con  
 tinue  
 de-  
 ploy  
 men  
 af-  
 ter  
 fin-  
 ish-  
 ing  
 an  
 asyn

first run, deploy steps and templates are also validated.

de-  
 ploy  
 step  
  
 This  
 func  
 tion  
 cal-  
 cu-  
 lates  
 whic  
 step  
 has  
 to  
 run  
 next  
 and  
 pass  
 con-  
 trol  
 into  
 do\_1  
 On  
 the

## 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-

0) into the list of deploy steps in the nodes driver\_internal\_info[deploy\_steps]. Is None if there are no steps to execute.

ploy  
 step  
 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  
 in-  
 dex  
 (from  
 ironic.  
 Pre-  
 pare  
 the  
 en-  
 vi-  
 ron-  
 men

and  
de-  
ploy  
a  
node

ironic.

Star  
de-  
ploy  
men  
or  
re-  
build  
ing  
on  
a  
node

This  
func  
tion  
does  
not  
check  
the  
node  
suit-  
abil-  
ity  
for  
de-  
ploy  
men  
its  
left  
up  
to  
the

caller.

## Paramet

- **tas**  
a  
Task  
ager  
in-  
stan

- **man**  
a  
Con  
duc-  
tor  
Man  
ager  
to  
run  
task  
on.
  - **con**  
a  
con-  
fig-  
drive  
if  
re-  
ques
  - **eve**  
even  
to  
pro-  
cess  
de-  
ploy  
or  
re-  
build
  - **dep**  
Op-  
tiona  
de-  
ploy  
step
- ironic.  
Val-  
i-  
date  
that  
a  
node  
is  
suit-  
able



for  
de-  
ploy  
men

#### Paramet

- **tas**  
a  
Task  
ager  
in-  
stan
- **eve**  
even  
to  
pro-  
cess  
de-  
ploy  
or  
re-  
buil

#### Raises

Nod  
Mai  
te-  
nan  
Nod  
Pro-  
tect  
In-  
vali  
State

**ironic.conductor.manager module**

Con  
duct  
all  
ac-  
tiv-  
ity  
re-  
late  
to  
bare

is responsible for performing all actions on bare metal resources (Chassis, Nodes, and Ports). Commands are received via RPCs. The conductor service also performs periodic tasks, eg. to monitor the status of active deployments.

only once, when the ConductorManager service starts. In this way, a single ConductorManager may use multiple drivers, and manage heterogeneous hardware.

eratively manage all nodes in the deployment. Nodes are locked by each conductor when performing actions which change the state of that node; these locks are represented by the *ironic.conductor.task\_manager.TaskManager* class.

each nodes driver. Rebalancing this ring can trigger various actions by each conductor, such as building or tearing down the TFTP environment for a node, notifying Neutron of a change, etc.

mul-  
ti-  
ple  
*Con*  
are  
run  
on  
dif-  
fer-  
ent  
host  
they  
are  
all  
ac-  
tive  
and  
co-  
op-

A  
*tooz*  
is  
used  
to  
dis-  
tribu-  
node  
acro  
the  
set  
of  
ac-  
tive  
con-  
duc-  
tors  
whic  
sup-  
port

**class** i  
Base  
*irc*  
*con*  
*bas*  
*Bas*

Iron  
 Con  
 duc-  
 tor  
 man  
 ager  
 main  
 class

RPC\_API

add\_node

change\_

continu

RPC  
 meth  
 to  
 con-  
 tinu  
 clea  
 ing  
 a  
 node

This  
 is  
 use-  
 ful  
 for  
 clea  
 ing  
 task  
 that  
 are  
 asyn  
 Whe  
 they  
 com  
 plete  
 they  
 call  
 back  
 via  
 RPC

a new worker and lock are set up, and cleaning continues. This can also be used to resume cleaning on take\_over.

Parame

- **con**  
an  
ad-  
min  
con-  
text.

- **nod**  
the  
id  
or  
uuid  
of  
a  
node

**Raises**  
In-  
valid  
State  
if  
the  
node  
is  
not  
in  
CLE  
WAI  
state

**Raises**  
NoF  
duc-  
tor-  
Wor  
whe  
there  
is  
no  
free  
worl  
to  
start  
asyn  
task

**Raises**  
Nod  
Loc  
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-  
 tinue  
 de-  
 ploy  
 ing  
 a  
 node  
 This  
 is  
 use-  
 ful  
 for  
 de-  
 ploy  
 ing  
 task  
 that  
 are  
 asyn  
 Whe  
 they  
 com  
 plete

RPC, a new worker and lock are set up, and deploying continues. This can also be used to resume deploying on `take_over`.

they  
call  
back  
via

## Parameters

- **con**  
an  
ad-  
min  
con-  
text.
- **nod**  
the  
ID  
or  
UUID  
of  
a  
node

## Raises

In-  
valid  
State  
if  
the  
node  
is  
not  
in  
DE-  
PLC  
WA  
state

## Raises

NoF  
duc-  
tor-  
Wor  
when  
there  
is  
no  
free  
world

to  
 start  
 asyn  
 task

**Raises**  
 Nod  
 Loc  
 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

ac-

tion
- **obj**

The

nam  
of  
the  
ac-  
tion  
meth  
to  
call

• **arg**  
The  
po-  
si-  
tion  
ar-  
gu-  
men  
to  
the  
ac-  
tion  
meth

• **kwa**  
The  
key-  
wor  
ar-  
gu-  
men  
to  
the  
ac-  
tion  
meth

**Returns**

A  
tu-  
ple  
with  
the  
up-  
date  
mad  
to  
the  
ob-  
ject  
and  
the

tion method

ing an object with a version newer than what is in the local registry, is to call this method to request a backport of the object.

re-  
sult  
of  
the  
ac-  
  
**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-  
ri-  
al-  
izer,  
upon  
re-  
ceiv

**Parame**  
  
 •  
  
**con**  
 The  
con-  
text  
with  
whic

to  
 per-  
 form  
 the  
 back  
 port

- **obj**  
 An  
 in-  
 stan-  
 of  
 a  
 Ver-  
 sion  
 dOb  
 ject  
 to  
 be  
 back  
 port

- **obj**  
 A  
 dict  
 of  
 {ob-  
 j-  
 nam  
 ver-  
 sion  
 map  
 ping

**Returns**

The  
 dow  
 grad  
 in-  
 stan-  
 of  
 ob-  
 jinst

**object\_**

Per-  
 form  
 an  
 ac-  
 tion

on  
 a  
 Ver-  
 sion  
 dOb  
 ject  
 class

Paramete

- **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  
 whic  
 may  
 (or  
 may  
 not)  
 be  
 an  
 in-

implementing VersionedObject class.

stand  
 of  
 the  
  
 remove\_  
  
 set\_bo  
  
 set\_cor  
  
 set\_inc  
  
 set\_tar  
  
 target  
  
 update\_  
  
 update\_  
  
 update\_  
  
 update\_  
  
 update\_  
  
 validat  
  
 vendor\_  
  
 vif\_att  
  
 vif\_det  
  
 vif\_lis  
  
 ironic.  
 Syn  
 the  
 pow  
 state  
 for  
 this



recorded.

node  
 in-  
 cre-  
 men  
 ing  
 the  
 cour  
 on  
 fail-  
 ure.  
 Whe  
 the  
 limi  
 of  
 pow  
 is  
 reac  
 the  
 node  
 is  
 put  
 into  
 main  
 te-  
 nanc  
 mod  
 and  
 the  
 er-  
 ror

Paramet

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan
- **cou**  
 num  
 ber  
 of  
 time  
 this  
 node  
 has

is incremented by one

ironic.

ironic.

Hand  
dles  
pow  
state  
sync  
ex-  
ceed  
ing  
the  
max  
re-  
tries

When  
syn-  
chro  
niz-  
ing  
the  
pow  
state  
be-  
twee  
a  
node  
and  
the  
DB  
has  
ex-  
ceed  
the  
max

imum number of retries, change the DB power state to be the actual node power state and place the node in maintenance.

## Parameter

- **task**  
a  
Task  
ager  
in-  
stan  
with

an  
 ex-  
 clu-  
 sive  
 lock

- **act**  
 the  
 ac-  
 tual  
 pow  
 state  
 of  
 the  
 node  
 a  
 pow  
 state  
 from  
 iron

- **exc**  
 the  
 ex-  
 cep-  
 tion  
 ob-  
 ject  
 that  
 caus  
 the  
 sync  
 pow  
 state  
 to  
 fail,  
 if  
 pres

ironic.conductor.notification\_utils module

ironic.

Help  
 for  
 con-  
 duc-

tor  
 send  
 ing  
 a  
 set  
 con-  
 sole  
 state  
 no-  
 ti-  
 fi-  
 ca-  
 tion.

Paramet

- **tas**
  
 a
   
 Task
   
 ager
   
 in-
   
 stan

- **act**
  
 Ac-
   
 tion
   
 strin
   
 to
   
 go
   
 in
   
 the
   
 Ever
   
 Type
   
 Mus
   
 be
   
 ei-
   
 ther
   
 con-
   
 sole,
   
 or
   
 con-
   
 sole,

- **sta**
  
 One
   
 of
   
*iron*
  
 END
   
 or

ER-  
 ROF  
 ironic.

Help  
 for  
 con-  
 duc-  
 tor  
 send  
 ing  
 a  
 set  
 pow  
 state  
 no-  
 ti-  
 fi-  
 ca-  
 tion.

Paramet

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan
- **lev**  
 No-  
 ti-  
 fi-  
 ca-  
 tion  
 leve  
 One  
 of  
*iron*
- **sta**  
 Sta-  
 tus  
 to  
 go

cates that ironic-conductor couldnt retrieve the power state for this node, or that it couldnt set the power state of the node.

used instead of the nodes `target_power_state` attribute since the `baremetal.node.power_set.start` notification is sent early, before `target_power_state` is set on the node.

`ironic.`

Help  
for  
con-  
duc-  
tor  
send  
ing  
a  
node  
pow

ent from the power state on an ironic node (DB), the ironic nodes power state is corrected to be that of the bare metal hardware. A notification is emitted about this after the database is updated to reflect this correction.

state  
cor-  
rect  
no-  
ti-  
fi-  
ca-  
tion.

Whe  
iron  
de-  
tect  
that  
the  
ac-  
tual  
pow  
state  
on  
a  
bare  
meta  
hard  
ware  
is  
dif-  
fer-

Paramet

- **tas**  
a  
Task  
ager  
in-  
stan
- **fro**  
the  
pow  
state  
of  
the  
node  
be-



fore  
 this  
 chan  
 was  
 de-  
 tecte

ironic.

Help  
 for  
 con-  
 duc-  
 tor  
 send  
 ing  
 a  
 set  
 pro-  
 vi-  
 sion  
 state  
 no-  
 ti-  
 fi-  
 ca-  
 tion.

Paramet

- **task**  
 a  
 Task  
 ager  
 in-  
 stan
- **level**  
 One  
 of  
 field
- **state**  
 One

ironic.conductor.rpcapi module

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\_r  
 and  
 do\_r  
  
 1.5  
 -  
  
 Add  
 val-  
 i-  
 date  
  
 1.6  
 -  
  
 char  
 do\_r  
 and  
 do\_r  
  
 acce  
 node  
 id  
 in-  
 stea  
 of  
 node  
 ob-  
 ject.  
  
 1.7  
 -  
  
 Add  
 topic  
 pa-  
 ram-  
 e-  
 ter  
 to  
 RPC

meth  
ods.

1.8  
-

Add  
char

1.9  
-

Add  
de-  
stroy

1.10  
-

Re-  
mov  
get\_

1.11  
-

Add  
get\_  
set\_

1.12  
-

val-  
i-  
date  
do\_v  
re-  
plac  
by  
sin-  
gle

veno  
meth

1.13  
-

Add

up-  
date

1.14  
-

Add  
drive

1.15  
-

Add  
re-  
build  
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\_I

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

get\_

1.31  
-

Add  
Ver-  
sion  
Ob-  
jects  
in-  
di-  
rec-  
tion  
API  
meth  
ods:

obje  
ob-  
ject  
and

obje

1.32

-

Add  
do\_1

1.33

-

Add  
up-  
date  
and  
de-  
stroy  
port  
grou

1.34

-

Add  
hear  
beat

1.35

-

Add  
de-  
stroy  
and  
up-  
date

1.36

-

Add  
cre-  
ate\_

1.37

-

Add  
de-  
stroy  
and  
up-  
date

1.38  
-

Add  
vif\_  
vif\_  
vif\_  
vif\_

1.39  
-

Add  
time  
out  
op-  
tiona  
pa-  
ram-  
e-  
ter  
to  
char

1.40  
-

Add  
in-  
ject\_

1.41  
-

Add  
cre-  
ate\_

1.42  
-

Add  
op-  
tiona  
ager  
to  
hear  
beat

1.43

-

Add  
do\_1  
do\_1  
and  
can\_

1.44

-

Add  
add\_  
and  
re-  
mov

1.45

-

Add  
con-  
tinue

1.46

-

Add  
re-  
set\_  
to  
up-  
date

1.47

-

Add  
sup-  
port  
for  
con-  
duc-  
tor  
grou

1.48

-

Add

al-  
 lo-  
 ca-  
 tion  
 API

1.49  
 -

Add  
 get\_  
 and  
 ager  
 ar-  
 gu-  
 men  
 to  
 hear  
 beat

1.50  
 -

Add  
 set\_  
 get\_  
 and  
  
 get\_

1.51  
 -

Add  
 ager  
 to  
 hear  
 beat

1.52  
 -

Add  
 de-  
 ploy  
 step  
 ar-  
 gu-  
 men  
 to

pro-  
vi-  
sion  
ing

RPC\_API

add\_node

Add  
or  
re-  
plac  
trait  
for  
a  
node

Paramet

- **con**  
re-  
ques  
con-  
text.
- **nod**  
node  
ID  
or  
UUID
- **tra**  
a  
list  
of  
trait  
to  
add  
to  
the  
node
- **rep**  
True  
to  
re-  
plac

all  
of  
the  
node  
trait.

- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
addi  
the  
trait  
wou  
ex-  
ceed  
the  
per-  
node  
trait  
limi

**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



node.

Parameters

- **conductor**  
 the conductor backgroup task to characterize power state of a
- **context**  
 re-ques context.
- **node**  
 node id or uuid
- **new**  
 one of iron power state values
- **time**  
 time out (in seconds) positive integer

indicates to use default timeout.

ger  
(>  
0)  
for  
any  
pow  
state  
Non

- - top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**  
NoF  
duc-  
tor-  
Wor  
whe  
there  
is  
no  
free  
worl  
to  
start  
asyn  
task

**continu**  
Sig-  
nal  
to  
con-  
duc-  
tor  
ser-  
vice  
to  
start  
the  
next  
clea  
ing  
ac-  
tion.

ductor for this RPC.

NOT  
 this  
 is  
 an  
 RPC  
 cast.  
 there  
 will  
 be  
 no  
 re-  
 spon-  
 or  
 ex-  
 cep-  
 tion  
 raise  
 by  
 the  
 con-

Paramete

- **con**  
 re-  
 ques-  
 con-  
 text.
- **nod**  
 node  
 id  
 or  
 uuid
- **top**  
 RPC  
 topic  
 De-  
 fault  
 to  
 self.

continu  
 Sig-  
 nal  
 to  
 con-  
 duc-

ductor for this RPC.

tor  
 ser-  
 vice  
 to  
 start  
 the  
 next  
 de-  
 ploy  
 men  
 ac-  
 tion.  
 NOT  
 this  
 is  
 an  
 RPC  
 cast,  
 there  
 will  
 be  
 no  
 re-  
 spon  
 or  
 ex-  
 cep-  
 tion  
 raise  
 by  
 the  
 con-

Paramete

- **con**  
 re-  
 ques  
 con-  
 text.
- **nod**  
 node  
 id  
 or  
 uuid
- **top**

RPC
 topic
 De-
 fault
 to
 self.

**create\_**
 Cre-
 ate
 an
 al-
 lo-
 ca-
 tion.

**Paramete**

•
 **con**
 re-
 ques
 con-
 text.

•
 **all**
 an
 al-
 lo-
 ca-
 tion
 ob-
 ject.

•
 **top**
 RPC
 topic
 De-
 fault
 to
 self.

**create\_**
 Syn-
 chro
 have
 a
 con-
 duc-
 tor
 val-

i-  
 date  
 and  
 cre-  
 ate  
 a  
 node  
  
 Cre-  
 ate  
 the  
 node  
 in-  
 for-  
 ma-  
 tion  
 in  
 the  
 data  
 and  
 re-  
 turn  
 a  
 node  
 ob-  
 ject.

Parame

- **con**  
re-ques-con-text.
- **nod**  
a  
cre-ated  
(but  
not  
save  
node  
ob-ject.
- **top**  
RPC  
topic  
De-

faces (e.g. `network_interface`).

fault  
 to  
 self.  
**Returns**  
 cre-  
 ated  
 node  
 ob-  
 ject.  
**Raises**  
 In-  
 ter-  
 face  
 Four  
 nEn  
 try-  
 poin  
 if  
 val-  
 i-  
 da-  
 tion  
 fails  
 for  
 any  
 dy-  
 nam  
 in-  
 ter-  
**Raises**  
 No-  
 Vali  
 De-  
 fault  
 ForI  
 ter-  
 face  
 if  
 no  
 de-  
 fault  
 can  
 be  
 cal-  
 cu-  
 lated  
 for  
 som  
 in-

terfaces, and explicit values must be provided.

ductor will lock related node and trigger specific driver actions if they are needed.

**create\_**  
Syn-  
chro-  
have  
a  
con-  
duc-  
tor  
val-  
i-  
date  
and  
cre-  
ate  
a  
port  
  
Cre-  
ate  
the  
port.  
in-  
for-  
ma-  
tion  
in  
the  
data  
and  
re-  
turn  
a  
port  
ob-  
ject.  
The  
con-

#### **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**

perform deallocation.

RPC  
 topic  
 De-  
 fault  
 to  
 self.

Raises

In-  
 valid  
 State  
 if  
 the  
 as-  
 so-  
 ci-  
 ated  
 node  
 is  
 in  
 the  
 wron  
 pro-  
 vi-  
 sion  
 state  
 to

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**

Nod  
As-  
so-  
ci-  
ated  
if  
the  
node  
con-  
tains  
an  
in-  
stan  
as-  
so-  
ci-  
ated  
with  
it.

**Raises**

In-  
valid  
State  
if  
the  
node  
is  
in  
the  
wron  
pro-

vi-  
sion  
state  
to  
per-  
form  
dele  
tion.

**destroy**  
Dele  
a  
port

**Parame**

- **con**  
re-  
ques  
con-  
text.

- **por**  
port  
ob-  
ject

- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**  
Nod  
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  
 does  
 not  
 ex-  
 ist.

**destroy**  
 Dele  
 a  
 port  
 grou

**Parame**

- **con**  
re-quescon-text.
- **por**  
portgrouob-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  
 grou  
 does  
 not  
 ex-  
 ist.

Raises

Port  
 grou  
 Notl  
 if  
 port  
 grou  
 is  
 not  
 emp

destroy

Dele  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor.  
  
 Dele  
 the  
 vol-  
 ume

operation.

con-  
nec-  
tor.  
The  
con-  
duc-  
tor  
will  
lock  
the  
re-  
lated  
node  
dur-  
ing  
this

Paramete

- **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-  
 nec-  
 tor  
 does  
 not  
 ex-  
 ist

**Raises**

Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 if  
 the  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 can-  
 not  
 be  
 foun

**destroy**

Dele  
 a  
 vol-  
 ume  
 tar-



get.

Paramete

- **con**  
re-ques  
con-text
- **tar**  
vol-ume  
tar-get  
ob-ject
- **top**  
RPC  
topic  
De-fault  
to self.

Raises

Node  
Lock  
if  
node  
is  
lock  
by  
an-  
othe  
con-  
duc-  
tor

Raises

Node  
Not-  
Four  
if  
the  
node  
as-  
so-  
ci-

ated  
 with  
 the  
 tar-  
 get  
 does  
 not  
 ex-  
 ist

**Raises**

Vol-  
 ume  
 get-  
 Not-  
 Foun  
 if  
 the  
 vol-  
 ume  
 tar-  
 get  
 can-  
 not  
 be  
 foun

**do\_node**

Sig-  
 nal  
 to  
 con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 per-  
 form  
 man  
 ual  
 clea  
 ing  
 on  
 a  
 node

**Parame**

- con  
re-

ques  
con-  
text.

- **node**  
node  
ID  
or  
UUID

- **clear**  
a  
list  
of  
clear  
step  
dic-  
tio-  
nar-  
ies.

- **topic**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
val-  
i-  
da-  
tion  
of  
pow  
drive  
in-  
ter-  
face  
faile

**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

**do\_node**

Sig-  
 nal  
 to  
 con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 per-  
 form  
 a  
 de-  
 ploy  
 men

**Parame**

- **con**  
 re-  
 ques  
 con-  
 text.
- **nod**  
 node  
 id  
 or  
 uuid
- **reb**  
 True  
 if  
 this  
 is  
 a  
 re-  
 build  
 re-  
 ques
- **con**  
 A  
 gzip

and  
 base  
 en-  
 code  
 con-  
 fig-  
 drive

- top**  
 RPC  
 topic  
 De-  
 fault  
 to  
 self.
- 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-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Raises**

NoF  
 duc-  
 tor-  
 Wor  
 whe  
 there  
 is  
 no  
 free  
 worl  
 to  
 start  
 asyn  
 task

The  
 node  
 mus  
 al-  
 read  
 be  
 con-  
 fig-  
 ured  
 and  
 in  
 the  
 ap-  
 pro-  
 pri-  
 ate  
 un-  
 de-  
 ploy  
 state

before this method is called.

**do\_node**  
 Sig-  
 nal  
 to

vironment.

con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 per-  
 form  
 a  
 res-  
 cue.  
 Paramete  
 •  
 con  
 re-  
 ques  
 con-  
 text.  
 •  
 nod  
 node  
 ID  
 or  
 UUI  
 •  
 res  
 A  
 strin  
 rep-  
 re-  
 sent  
 ing  
 the  
 pass  
 wor  
 to  
 be  
 set  
 in-  
 side  
 the  
 res-  
 cue  
 en-  
 •  
 top  
 RPC



topic  
De-  
fault  
to  
self.

#### **Raises**

In-  
stan-  
cue-  
Fail-  
ure

#### **Raises**

NoF  
duc-  
tor-  
Wor  
when  
there  
is  
no  
free  
world  
to  
start  
asyn  
task

The  
node  
mus  
al-  
read  
be  
con-  
fig-  
ured  
and  
in  
the  
ap-  
pro-  
pri-  
ate  
state  
be-  
fore  
this

method is called.

**do\_node**  
Sig-

nal  
 to  
 con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 tear  
 dow  
 a  
 de-  
 ploy  
 men

Paramet

- **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-  
 valie  
 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-  
e-  
ter  
is  
miss  
ing

**Raises**

NoF  
duc-  
tor-  
Wor  
whe  
there  
is  
no  
free  
worl  
to  
start  
asyn  
task  
  
The  
node  
mus  
al-  
read  
be  
con-  
fig-

fore this method is called.

ured  
 and  
 in  
 the  
 ap-  
 pro-  
 pri-  
 ate  
 de-  
 ploy  
 state  
 be-  
  
**do\_node**  
 Sig-  
 nal  
 to  
 con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 per-  
 form  
 an  
 un-  
 res-  
 cue.  
  
**Parame**  
  
 •  
**con**  
 re-  
 ques  
 con-  
 text.  
  
 •  
**nod**  
 node  
 ID  
 or  
 UUID  
  
 •  
**top**  
 RPC  
 topic  
 De-  
 fault

method is called.

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  
 and  
 in  
 the  
 ap-  
 pro-  
 pri-  
 ate  
 state  
 be-  
 fore  
 this  
**do\_prov**  
 Sig-  
 nal

to  
 con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 per-  
 form  
 the  
 give  
 ac-  
 tion  
 on  
 a  
 node

Param

- **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-  
 form  
  
 This  
 en-  
 cap-  
 su-  
 lates  
 som  
 pro-  
 vi-  
 sion  
 ing  
 ac-  
 tions  
 in  
 a

cuted on a random conductor with the specified driver. If the method mode is async the conductor will start background worker to perform vendor action.

Paramete

- con re- ques con- text.
-



**dri**  
nam  
of  
the  
drive  
on  
whic  
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.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 for  
 pa-  
 ram-  
 e-  
 ter  
 er-  
 rors.

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Raises**

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 drive  
 does  
 have  
 a

if the vendor interface does not support the specified driver\_method.

ven-  
dor  
in-  
ter-  
face  
or

#### **Raises**

Driv  
Not-  
Foun  
if  
the  
sup-  
plic  
driv  
is  
not  
load

#### **Raises**

NoF  
duc-  
tor-  
Wor  
whe  
there  
is  
no  
free  
worl  
to  
start  
asyn  
task

#### **Raises**

In-  
ter-  
face  
Foun  
nEn  
try-  
poin  
if  
the  
de-  
fault  
in-  
ter-  
face  
for

is invalid.

can be found for this drivers vendor interface.

a  
hard  
ware  
type

Raises

No-  
Valid  
De-  
fault  
ForI  
ter-  
face  
if  
no  
de-  
fault  
in-  
ter-  
face  
im-  
ple-  
men-  
ta-  
tion

Returns

A  
dic-  
tio-  
nary  
con-  
tain-  
ing:

return

The  
re-  
spon-  
of  
the  
in-  
voke  
ven-  
dor  
meth

async

Boo  
valu

chronously the response will be always None.

response object (True) or return it in the response body (False).

When  
the  
meth  
was  
in-  
voke  
asyn  
chro  
(Tru  
or  
syn-  
chro  
(Fal  
When  
in-  
voke  
asyn

#### attach

Boo  
valu  
When  
to  
at-  
tach  
the  
re-  
spon  
of  
the  
in-  
voke  
ven-  
dor  
meth  
to  
the  
HTT

#### get\_bo

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  
 node  
 is  
 lock  
 by  
 an-  
 othe  
 con-  
 duc-  
 tor.

Raises

Un-  
 sup-  
 port  
 ed-

Driv
 ten-
 sion
 if
 the
 node
 drive
 does
 sup-
 port
 man
 age-
 men

Raises

In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 whe
 the
 wron
 drive
 info
 is
 spec
 i-
 fied.

Raises

Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 miss
 ing
 sup-
 plied
 info

Returns

a
 dic-
 tio-

unknown.

nary  
 con-  
 tain-  
 ing:  
**boot\_c**  
 the  
 boot  
 de-  
 vice  
 one  
 of  
*irc*  
*com*  
*boo*  
 or  
 Non  
 if  
 it  
 is  
 un-  
 know  
**persist**  
 Whe  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 or  
 not,  
 Non  
 if  
 it  
 is  
**get\_con**  
 Get  
 the  
 con-  
 duc-  
 tor  
 whic  
 the



node  
is  
map  
to.

#### Parameters

**node**  
a  
node  
ob-  
ject.

#### Returns

the  
con-  
duc-  
tor  
host  
nam

#### Raises

No-  
Vali-  
Hos

#### get\_connection

Get  
con-  
nec-  
tion  
in-  
for-  
ma-  
tion  
about  
the  
con-  
sole

#### Parameters

- **con**  
re-  
ques  
con-  
text.
- **nod**  
node  
id  
or  
uuid

- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the  
node  
drive  
does  
sup-  
port  
con-  
sole

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-  
fied.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu

if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**get\_cun**  
 Get  
 RPC  
 topic  
 nam  
 for  
 the  
 cur-  
 rent  
 con-  
 duc-  
 tor.

**get\_dri**  
 Get  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 drive

**Parame**

- **con**  
re-quescon-text.
- **dri**  
namofthe drive
- **top**

RPC  
 topic  
 De-  
 fault  
 to  
 self.

Returns

a  
 dic-  
 tio-  
 nary  
 with  
 <pro  
 erty  
 nam  
 de-  
 scrip  
 tion  
 en-  
 tries

Raises

Driv  
 Not-  
 Foun

get\_dri

Re-  
 triev  
 in-  
 for-  
 ma-  
 tion  
 abou  
 ven-  
 dor  
 meth  
 ods  
 of  
 the  
 give  
 drive

Parame

- con  
 an  
 ad-  
 min  
 con-  
 text.

- **dri**  
nam  
of  
the  
drive
- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
cur-  
rent  
drive  
does  
not  
have  
ven-  
dor  
in-  
ter-  
face

**Raises**

Driv  
Not-  
Four  
if  
the  
sup-  
plie  
drive  
is  
not  
load

**Raises**

In-  
ter-

is invalid.

can be found for this drivers vendor interface.

face  
 Four  
 nEn  
 try-  
 poin  
 if  
 the  
 de-  
 fault  
 in-  
 ter-  
 face  
 for  
 a  
 hard  
 ware  
 type

Raises

No-  
 Vali  
 De-  
 fault  
 ForI  
 ter-  
 face  
 if  
 no  
 de-  
 fault  
 in-  
 ter-  
 face  
 im-  
 ple-  
 men  
 ta-  
 tion

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**

- **com**  
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-  
 tor.

Raises

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 node  
 drive  
 does  
 sup-  
 port  
 man  
 age-  
 men

Raises

In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-



Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-  
fied.

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
miss  
ing  
sup-  
plic  
info

#### **Returns**

In-  
di-  
ca-  
tor  
state  
one  
of  
mod

#### **get\_noo**

Re-  
triev  
in-  
for-  
ma-  
tion  
about  
ven-  
dor  
meth  
ods  
of  
the  
give  
node

Parameters

- **context**  
 an administrative context.
- **nodeId**  
 the id or uuid of a node.
- **topic**  
 RPC topic. Default fault to self.

Returns

dictionary of <meta-data> entries

get\_node

Requests the node from the conductor with an

ager  
 to-  
 ken

Paramete

- **con**
  
 re-
 ques
 con-
 text.
- **nod**
  
 node
 ID
 or
 UUI
- **top**
  
 RPC
 topic
 De-
 fault
 to
 self.

Raises

Nod
 Loc
 if
 node
 is
 lock
 by
 an-
 othe
 con-
 duc-
 tor.

Returns

A
 Nod
 ob-
 ject
 with
 ager
 to-
 ken.

get\_rai

in the input RAID configuration.

Get the log-ical disk properties for RAID configuration. Gets the information about log-ical disk properties which can be specified

Parameters

- **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  
RAI  
con-  
fig-  
u-  
ra-  
tion.

#### **Raises**

In-  
ter-  
face  
Four  
nEn  
try-  
poin  
if  
the  
de-  
fault  
in-  
ter-  
face  
for  
a  
hard  
ware  
type

is invalid.

**Raises**

No-  
Valid  
De-  
fault  
ForI  
ter-  
face  
if  
no  
de-  
fault  
in-  
ter-  
face  
im-  
ple-  
men-  
ta-  
tion

can be found for this drivers RAID interface.

**Returns**

A  
dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
prop  
er-  
ties  
that  
can  
be  
men-  
tioned  
for  
log-  
i-

cal disks and a textual description for them.

**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**

NodeLockException  
 if node is locked by another conductor.

**Raises**

UnsupportedDriverExtensionException  
 if the node driver does not support management.

**Raises**

InvalidParameterException  
 if the wrong driver info is specified.



fied.

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
miss  
ing  
sup-  
plie  
info

#### **Returns**

A  
list  
with  
the  
sup-  
port  
boot  
de-  
vice  
de-  
fine  
in  
*irc*  
*com*  
*boo*

#### **get\_sup**

Get  
node  
hard  
ware  
com  
po-  
nent  
and  
their  
in-  
di-  
ca-  
tors.

#### **Parame**

- con

re-  
ques  
con-  
text.

- **node**  
node  
id  
or  
uuid

- **components**  
The  
hard  
ware  
com  
po-  
nent  
one  
of  
*irc*  
*com*  
*com*

- **topics**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

Node  
Lock  
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
 man
 age-
 men

Raises

In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 whe
 the
 wron
 drive
 info
 is
 spec
 i-
 fied.

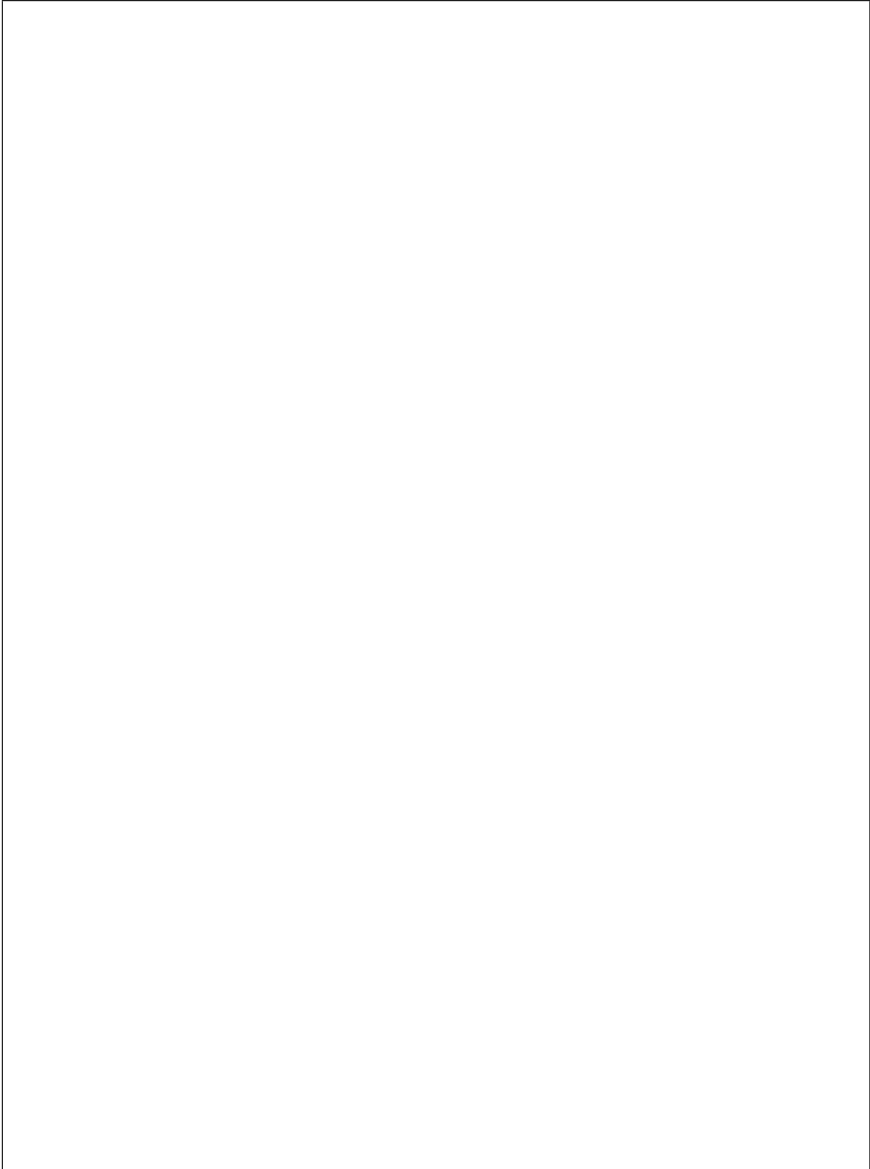
Raises

Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 miss
 ing
 sup-
 plied
 info

Returns

A
 dic-
 tio-

IDs as values.



(continues on next page)

nary  
 of  
 hard  
 ware  
 com  
 po-  
 nent  
 (*ir*  
*com*  
*com*  
 as  
 keys  
 with  
 in-  
 di-  
 ca-  
 tor

(continued from previous page)

--

**get\_top**  
Get  
the  
RPC  
topic  
for  
the  
con-  
duc-  
tor  
ser-  
vice  
the  
node  
is  
map  
to.

**Paramete**  
**nod**  
a  
node  
ob-  
ject.

**Returns**  
an  
RPC  
topic  
strin

**Raises**  
No-  
Vali  
Hos

**get\_top**  
Get  
RPC  
topic  
nam  
for  
a  
con-  
duc-  
tor  
sup-  
port  
ing  
the

driver. A conductor is selected at random from the set of qualified conductors.

give  
 drive  
 The  
 topic  
 is  
 used  
 to  
 route  
 route  
 mes  
 sage  
 to  
 the  
 con-  
 duc-  
 tor  
 sup-  
 port  
 ing  
 the  
 spec  
 i-  
 fied  
 Param  
 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

Paramete

- **con**  
re-ques  
con-text.
- **nod**  
node  
ID  
or  
UUID
- **cal**  
URI  
to  
reac  
back  
to  
the  
rame
- **top**  
RPC  
topic  
De-fault  
to  
self.
- **age**  
ran-dom  
gen-er-ated  
val-i-da-tion  
to-ken.
- **age**  
the

ver-  
 sion  
 of  
 the  
 ager  
 that  
 is  
 hear  
 beat  
 ing

- **age**  
 TLS  
 cer-  
 tifi-  
 cate  
 for  
 the  
 ager

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 ager  
 to-  
 ken  
 is  
 re-  
 ceiv

**inject\_**

In-  
 ject  
 NM  
 for  
 a  
 node  
  
 In-  
 ject  
 NM  
 (Nor  
 Mas



all drivers support this.

able  
In-  
ter-  
rupt  
for  
a  
node  
im-  
me-  
di-  
ately  
Be  
awa  
that  
not

#### 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-

agement.inject\_nmi.

valid boot device is specified.

duc-  
tor.  
  
**Raises**  
 Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the  
node  
drive  
does  
sup-  
port  
man  
age-  
men  
or  
man

**Raises**  
 In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-  
fied  
or  
an  
in-

**Raises**  
 Mis-  
ing-  
Pa-  
ram-  
e-

ter-  
 Valu  
 if  
 miss  
 ing  
 sup-  
 plied  
 info

**inspect**  
 Sig-  
 nals  
 the  
 con-  
 duc-  
 tor  
 ser-  
 vice  
 to  
 per-  
 form  
 hard  
 ware  
 in-  
 tro-  
 spec  
 tion.

**Parame**

- **con**  
 re-  
 ques  
 con-  
 text.
- **nod**  
 node  
 id  
 or  
 uuid
- **top**  
 RPC  
 topic  
 De-  
 fault  
 to  
 self.

Raises

Node  
 Local  
 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

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 node  
 drive  
 does  
 sup-  
 port

in-  
spec  
tion.

**Raises**

In-  
valid  
State  
if  
in-  
spec  
is  
not  
a  
valid  
ac-  
tion  
to  
do  
in  
the  
cur-  
rent  
state

**object\_**

Per-  
form  
an  
ac-  
tion  
on  
a  
Ver-  
sion  
dOb  
ject  
in-  
stan-  
  
We  
wan  
any  
con-  
duc-  
tor  
to  
han-  
dle  
this,  
so  
it  
is

topic argument for this method.

Paramete

- **con**  
 The  
 con-  
 text  
 with  
 whic  
 to  
 per-  
 form  
 the  
 ac-  
 tion
- **obj**  
 The  
 ob-  
 ject  
 in-  
 stan  
 on  
 whic  
 to  
 per-  
 form  
 the  
 ac-  
 tion
- **obj**  
 The  
 nam  
 of  
 the  
 ac-  
 tion  
 meth  
 to  
 call

grade

- arg**  
 The  
 po-  
 si-  
 tion:  
 ar-  
 gu-  
 men  
 to  
 the  
 ac-  
 tion  
 meth
- kwa**  
 The  
 key-  
 wor  
 ar-  
 gu-  
 men  
 to  
 the  
 ac-  
 tion  
 meth

**Raises**

Notl  
 ple-  
 men  
 ed-  
 Er-  
 ror  
 whe  
 an  
 op-  
 er-  
 a-  
 tor  
 mak  
 an  
 er-  
 ror  
 dur-  
 ing  
 up-

**Returns**

A

tion method

tu-  
 ple  
 with  
 the  
 up-  
 date  
 mad  
 to  
 the  
 ob-  
 ject  
 and  
 the  
 re-  
 sult  
 of  
 the  
 ac-  
 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-  
 ri-  
 al-  
 izer,  
 upon  
 re-



ing an object with a version newer than what is in the local registry, is to call this method to request a backport of the object.

topic argument for this method.

#### Parameters

- **context**  
The context with which to perform the backport.
- **obj**  
An instance of a VersionedObject.

grade

ject  
 to  
 be  
 back  
 port

•

obj  
 A  
 dict  
 of  
 {ob-  
 j-  
 nam  
 ver-  
 sion  
 map  
 ping

Raises

Notl  
 ple-  
 men  
 ed-  
 Er-  
 ror  
 whe  
 an  
 op-  
 er-  
 a-  
 tor  
 mak  
 an  
 er-  
 ror  
 dur-  
 ing  
 up-

Returns

The  
 dow  
 grad  
 in-  
 stan  
 of  
 ob-  
 jinst

object\_

topic argument for this method.

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-  
ten-  
tiona  
that  
there  
is  
no

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

grade

key-  
 wor  
 ar-  
 gu-  
 men  
 to  
 the  
 ac-  
 tion  
 meth

Raises

Notl  
 ple-  
 men  
 ed-  
 Er-  
 ror  
 whe  
 an  
 op-  
 er-  
 a-  
 tor  
 mak  
 an  
 er-  
 ror  
 dur-  
 ing  
 up-

Returns

The  
 re-  
 sult  
 of  
 the  
 ac-  
 tion  
 meth  
 whic  
 may  
 (or  
 may  
 not)  
 be  
 an  
 in-  
 stan  
 of

implementing VersionedObject class.

removed from the node.

the

**remove\_**

Re-  
 mov  
 som  
 or  
 all  
 trait  
 from  
 a  
 node

**Parame**

- con**  
 re-  
 ques  
 con-  
 text.
- nod**  
 node  
 ID  
 or  
 UUI
- tra**  
 a  
 list  
 of  
 trait  
 to  
 re-  
 mov  
 from  
 the  
 node  
 or  
 Non  
 If  
 Non  
 all  
 trait  
 will  
 be

top  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

Nod  
Locl  
if  
node  
is  
lock  
by  
an-  
othe  
con-  
duc-  
tor.

**Raises**

Nod  
Not-  
Four  
if  
the  
node  
does  
not  
ex-  
ist.

**Raises**

Nod  
Trai  
Not-  
Four  
if  
one  
of  
the  
trait  
is  
not  
foun

**set\_boot**

Set  
the  
boot  
de-

support this.

Parameters

- **con**  
re-ques-  
con-  
text.
- **nod**  
node  
id  
or  
uuid
- **dev**  
the  
boot  
de-  
vice  
one  
of  
*ironic*  
*com*



*boo*

- **per**  
Whe  
to  
set  
next  
boot  
or  
mak  
the  
char  
per-  
ma-  
nent  
De-  
fault  
Fals

- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**  
Nod  
Loc  
if  
node  
is  
lock  
by  
an-  
othe  
con-  
duc-  
tor.

**Raises**  
Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the

valid boot device is specified.

node  
drive  
does  
sup-  
port  
man  
age-  
men

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-  
fied  
or  
an  
in-

Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
miss  
ing  
sup-  
plie  
info

set\_con

En-  
able  
the  
con-  
sole

Parameters

- **con**  
re-ques  
con-text.
- **nod**  
node  
id  
or  
uuid
- **top**  
RPC  
topic  
De-fault  
to  
self.
- **ena**  
Boo  
valu  
whe  
the  
con-sole  
is  
en-able  
or  
dis-able

Raises

Un-sup-ported  
ed-Drive  
ten-sion  
if  
the  
node  
drive  
does

sup-  
port  
con-  
sole

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-  
fied.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

**Raises**

NoF  
duc-  
tor-  
Wor  
whe  
ther  
is  
no  
free

world  
 to  
 start  
 asyn  
 task

**set\_ino**

Set  
 node  
 hard  
 ware  
 com  
 po-  
 nent  
 in-  
 di-  
 ca-  
 tor  
 to  
 the  
 de-  
 sirec  
 state

**Paramete**

- con**  
 re-  
 ques  
 con-  
 text.
- nod**  
 node  
 id  
 or  
 uuid
- com**  
 The  
 hard  
 ware  
 com  
 po-  
 nent  
 one  
 of  
*irc*  
*com*  
*com*

- **ind**  
In-  
di-  
ca-  
tor  
IDs,  
as  
re-  
port  
by  
*get\_*
- **sta**  
In-  
di-  
ca-  
tor  
state  
one  
of  
mod
- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

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  
man  
age-  
men

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
drive  
info  
is  
spec  
i-  
fied  
or  
an  
in-

valid boot device is specified.

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
miss  
ing  
sup-  
plie  
info

#### **set\_target**

Stor  
the

tar-  
 get  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 node  
  
 Stor  
 the  
 tar-  
 get  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 node

Paramet

- **con**  
re-ques  
con-text.
- **nod**  
node  
id  
or  
uuid
- **tar**  
Dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
tar-  
get  
RAI



an empty dictionary as well.

tion.

con-  
fig-  
u-  
ra-  
tion.  
It  
may  
be

- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**  
Un-  
sup-  
port  
ed-  
Drive  
ten-  
sion  
if  
the  
node  
drive  
does  
sup-  
port  
RAI  
con-  
fig-  
u-  
ra-

**Raises**  
In-  
valid  
Pa-  
ram  
e-  
ter-  
Valu  
if  
val-  
i-  
da-

tion  
 of  
 tar-  
 get  
 raid  
 con-  
 fig  
 fails

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

**Raises**

Nod  
 Loc  
 if  
 node  
 is  
 lock  
 by  
 an-  
 othe  
 con-  
 duc-  
 tor.

**update\_**

Syn-  
 chro  
 have  
 a  
 con-  
 duc-  
 tor  
 up-  
 date

ductor 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.

the  
node  
in-  
for-  
ma-  
tion.  
Up-  
date  
the  
node  
in-  
for-  
ma-  
tion  
in  
the  
data  
and  
re-  
turn  
a  
node  
ob-  
ject.  
The  
con-

Note  
that  
pow  
shou  
not  
be  
pass  
via  
this  
meth  
Use  
char  
for  
ini-  
ti-  
at-  
ing  
drive  
ac-  
tions

Param

- **con**  
re-ques  
con-text.

- **nod**  
a char  
(but not  
save node  
ob-ject.

- **top**  
RPC topic  
De-fault  
to self.

- **res**  
when  
to re-set  
hard ware  
in-ter-face  
to their  
de-fault

**Returns**  
up-date  
node ob-ject,  
in-cluding

terfaces, and explicit values must be provided.

all  
 field  
**Raises**  
 No-  
 Vali-  
 De-  
 fault  
 ForI  
 ter-  
 face  
 if  
 no  
 de-  
 fault  
 can  
 be  
 cal-  
 cu-  
 latec  
 for  
 som  
 in-

**update\_**  
 Syn-  
 chro  
 have  
 a  
 con-  
 duc-  
 tor  
 up-  
 date  
 the  
 port.  
 in-  
 for-  
 ma-  
 tion.  
 Up-  
 date  
 the  
 port.  
 in-  
 for-  
 ma-  
 tion  
 in  
 the  
 data

ductor will lock related node and trigger specific driver actions if they are needed.

Paramete

- **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-  
 for-  
 ma-  
 tion.  
 Up-  
 date  
 the  
 port  
 grou-  
 in-  
 for-  
 ma-  
 tion  
 in  
 the  
 data  
 and  
 re-  
 turn  
 a  
 port  
 grou-  
 ob-  
 ject.

The conductor will lock related node and trigger specific driver actions if they are needed.

Param

- **con**
  
 re-
 ques
 con-
 text.
- **por**
  
 a
 char

(but not save port group ob- ject.

- **top** RPC topic De- fault to self.

**Returns**

up- date port group ob- ject, in- clud ing all field

**update\_**

Up- date the vol- ume con- nec- tors in- for- ma- tion.

Up- date the vol- ume con- nec- tors in-



ume connector object. The conductor will lock the related node during this operation.

Parameters

- **connector**: A connector object. The conductor will lock the related node during this operation.
- **context**: A context object. The conductor will lock the related node during this operation.
- **topology**: A topology object. The conductor will lock the related node during this operation.

Raises

- **In-ValidParameterError**: Invalid parameter.

Valu  
 if  
 the  
 vol-  
 ume  
 con-  
 nec-  
 tors  
 UUI  
 is  
 be-  
 ing  
 char

**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  
 con-  
 nec-  
 tor  
 does  
 not  
 ex-  
 ist

**Raises**

Vol-  
 ume  
 Con

nec-  
 torN  
 Four  
 if  
 the  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 can-  
 not  
 be  
 foun

Raises

Vol-  
 ume  
 Con  
 nec-  
 torT  
 pe-  
 An-  
 dI-  
 dAl-  
 read  
 ists  
 if  
 an-  
 othe  
 con-  
 nec-  
 tor  
 al-  
 read

exists with the same values for type and connector\_id fields

Returns

up-  
 date  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 ob-  
 ject,  
 in-  
 clud  
 ing  
 all  
 field

target object. The conductor will lock the related node during this operation.

update\_
 Up-
 date
 the
 vol-
 ume
 tar-
 gets
 in-
 for-
 ma-
 tion.
 Up-
 date
 the
 vol-
 ume
 tar-
 gets
 in-
 for-
 ma-
 tion
 in
 the
 data
 and
 re-
 turn
 a
 vol-
 ume

Paramet
 •
 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-  
Valu  
if  
the  
vol-  
ume  
tar-  
gets  
UUI  
is  
be-  
ing  
char

#### **Raises**

Nod  
Loc  
if  
the  
node  
is  
al-  
read  
lock

#### **Raises**

Nod  
Not-  
Four  
if  
the  
node

ist

as-  
so-  
ci-  
ated  
with  
the  
vol-  
ume  
tar-  
get  
does  
not  
ex-

Raises

Vol-  
ume  
get-  
Not-  
Four  
if  
the  
vol-  
ume  
tar-  
get  
can-  
not  
be  
foun

Raises

Vol-  
ume  
get-  
Boo  
dex-  
Al-  
read  
ists  
if  
a  
vol-  
ume  
tar-  
get  
al-  
read  
ex-  
ists  
with

the same node ID and boot index values

Returns

up-  
date  
vol-  
ume  
tar-  
get  
ob-  
ject,  
in-  
clud  
ing  
all  
field

validate

Val-  
i-  
date  
the  
*core*  
and  
*stan*  
*dard*  
*ized*  
in-  
ter-  
face  
for  
drive

Parameters

- **con**  
re-  
ques  
con-  
text.
- **nod**  
node  
id  
or  
uuid
- **top**  
RPC  
topic

De-  
 fault  
 to  
 self.

**Returns**

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 re-  
 sults  
 of  
 each  
 in-  
 ter-  
 face  
 val-  
 i-  
 da-  
 tion.

**vendor\_**

Re-  
 ceiv  
 re-  
 ques  
 for  
 venco  
 spec  
 ac-  
 tions

Syn-  
 chro  
 val-  
 i-  
 date  
 drive  
 spe-  
 cific  
 info  
 or  
 get  
 drive  
 sta-  
 tus,  
 and  
 if



vokes the vendor method. If the method mode is async the conductor will start background worker to perform vendor action.

## Parameters

- **conduct**  
re-ques con-text.
- **node**  
node id or uuid
- **driver**  
name of meth for drive
- **http**  
the HTTP meth used for the re-ques
- **info**  
info for node drive
- **topic**  
RPC topic De-

fault  
 to  
 self.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 sup-  
 plied  
 info  
 is  
 not  
 valid

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Raises**

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 cur-  
 rent  
 drive  
 does

not  
 have  
 ven-  
 dor  
 in-  
 ter-  
 face

**Raises**

NoF  
 duc-  
 tor-  
 Wor  
 whe  
 there  
 is  
 no  
 free  
 worl  
 to  
 start  
 asyn  
 task

**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

chronously the response will be always None.

response object (True) or return it in the response body (False).

the  
in-  
voke  
ven-  
dor  
meth

**asyn**  
Boo  
valu  
Whe  
the  
meth  
was  
in-  
voke  
asyn  
chro  
(Tru  
or  
syn-  
chro  
(Fal  
Whe  
in-  
voke  
asyn

**attach**  
Boo  
valu  
Whe  
to  
at-  
tach  
the  
re-  
spor  
of  
the  
in-  
voke  
ven-  
dor  
meth  
to  
the  
HTT

**vif\_att**  
At-

tach  
VIF  
to  
a  
node

#### Parame

- **con**  
re-  
ques-  
con-  
text.
- **nod**  
node  
ID  
or  
UUID
- **vif**  
a  
dic-  
tio-  
nary  
rep-  
re-  
sent-  
ing  
VIF  
ob-  
ject.  
It  
mus-  
have  
an  
id  
key,  
who

value is a unique identifier for that VIF.

- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**

Node  
 Local  
 if  
 node  
 has  
 an  
 ex-  
 clu-  
 sive  
 lock  
 held  
 on  
 it

**Raises**

Net-  
 work  
 Er-  
 ror,  
 if  
 an  
 er-  
 ror  
 oc-  
 curs  
 dur-  
 ing  
 at-  
 tach-  
 ing  
 the  
 VIF.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter  
 Valu  
 if  
 a  
 pa-  
 ram-  
 e-  
 ter  
 that  
 re-  
 quir  
 for

tach is wrong/missing.

VIF  
at-

**vif\_det**  
De-  
tach  
VIF  
from  
a  
node

#### Parame

- **con**  
re-  
ques  
con-  
text.
- **nod**  
node  
ID  
or  
UUI
- **vif**  
an  
ID  
of  
a  
VIF.
- **top**  
RPC  
topic  
De-  
fault  
to  
self.

**Raises**  
Nod  
Lock  
if  
node  
has  
an  
ex-  
clu-

sive  
 lock  
 held  
 on  
 it

Raises

Net-  
 worl  
 Er-  
 ror,  
 if  
 an  
 er-  
 ror  
 oc-  
 curs  
 dur-  
 ing  
 de-  
 tach  
 ing  
 the  
 VIF

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 pa-  
 ram-  
 e-  
 ter  
 that  
 re-  
 quir  
 for  
 VIF  
 de-

tach is wrong/missing.

vif\_list

List  
 at-  
 tach  
 VIF  
 for



a  
 node

Paramete

- **con**  
re-ques  
con-text.
- **nod**  
node  
ID  
or  
UUID
- **top**  
RPC  
topic  
De-fault  
to  
self.

Returns

List  
 of  
 VIF  
 dic-  
 tio-  
 nar-  
 ies,  
 each  
 dic-  
 tio-  
 nary  
 will  
 have  
 an  
 id  
 en-  
 try  
 with  
 the

ID of the VIF.

Raises

Net-  
 worl

is wrong/missing.

ironic.conductor.steps module

Er-  
 ror,  
 if  
 an  
 er-  
 ror  
 oc-  
 curs  
 dur-  
 ing  
 list-  
 ing  
 the  
 VIF  
 Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 pa-  
 ram-  
 e-  
 ter  
 that  
 re-  
 quir  
 for  
 VIF  
 list

ironic.  
 Find  
 an  
 iden-  
 ti-  
 cal  
 step  
 in  
 the  
 list  
 of  
 step

ironic.  
Com  
pare  
step  
ig-  
nor-  
ing  
their  
pri-  
or-  
ity.

ironic.  
Set  
up  
the  
node  
with  
clea  
step  
in-  
for-  
ma-  
tion  
for  
clea  
ing.

For  
au-  
to-  
mate  
clea  
ing,  
get  
the  
clea  
step  
from  
the  
drive  
For  
man  
ual  
clea  
ing,  
the  
user

clean steps are known but need to be validated against the drivers clean steps.

**Raises**  
In-  
valid

Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 there  
 is  
 a  
 prob  
 lem  
 with  
 the  
 user  
 clea  
 step

Raises

Nod  
 Clea  
 ing-  
 Fail-  
 ure  
 if  
 there  
 was  
 a  
 prob  
 lem  
 get-  
 ting  
 the  
 clea  
 step

ironic.

Set  
 up  
 the  
 node  
 with  
 de-  
 ploy  
 men  
 step  
 in-  
 for-  
 ma-  
 tion  
 for

de-  
 ploy  
 ing.  
  
 Get  
 the  
 de-  
 ploy  
 step  
 from  
 the  
 drive

Paramet

res  
 Whe  
 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  
 de-  
 ploy  
 men  
 step

ironic.

Val-

internal info.

i-  
date  
the  
user  
de-  
ploy  
step  
and  
the  
de-  
ploy  
tem-  
plate  
for  
a  
node

**Paramet**

- tas**  
A  
Task  
ager  
ob-  
ject
- dep**  
De-  
ploy  
step  
to  
val-  
i-  
date  
Op-  
tiona  
If  
not  
pro-  
vide  
then  
will  
chec  
node  
drive
- ski**  
whe

tion.

that are unsupported by the nodes driver interfaces or user deploy steps are unsupported by the nodes driver interfaces

skip  
miss  
ing  
step  
that  
are  
not  
yet  
avai  
able  
at  
the  
time  
of  
val-  
i-  
da-

#### Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
in-  
stan  
has  
trait  
that  
map  
to  
de-  
ploy  
step

#### Raises

In-  
stan  
ploy  
Fail-  
ure  
if  
there  
was  
a

driver.

### **ironic.conductor.task\_manager module**

prob  
lem  
get-  
ting  
the  
de-  
ploy  
step  
from  
the

A  
con-  
text  
man-  
ager  
to  
per-  
form  
a  
se-  
ries  
of  
task  
on  
a  
set  
of  
re-  
sour

*Tas*  
is  
a  
con-  
text  
man-  
ager  
cre-  
ated  
on-  
dem  
to  
al-  
low  
syn-  
chro  
nize



a node and its resources.

tion that the TaskManager instance exists. You may create a TaskManager instance without locking by passing `shared=True` when creating it, but certain operations on the resources held by such an instance of TaskManager will not be possible. Requiring this exclusive lock guards against parallel operations interfering with each other.

dating the driver interfaces.

ac-  
cess  
to

The  
*Tas*  
will.  
by  
de-  
fault  
ac-  
quir  
an  
ex-  
clu-  
sive  
lock  
on  
a  
node  
for  
the  
du-  
ra-

A  
shar  
lock  
is  
use-  
ful  
whe  
per-  
form  
ing  
non-  
inter  
op-  
er-  
a-  
tions  
such  
as  
val-  
i-

An

instances, that are typically deployed on different hosts.

invocation requires an exclusive lock.

ex-  
clu-  
sive  
lock  
is  
store  
in  
the  
data  
to  
co-  
or-  
di-  
nate  
be-  
twee  
*irc*  
*con*  
*man*

*Tas*  
meth  
ods,  
as  
well  
as  
drive  
meth  
ods,  
may  
be  
dec-  
o-  
rated  
to  
de-  
ter-  
mine  
when  
their

The  
Task  
ager  
in-  
stan  
ex-  
pose  
cer-  
tain

you may access:

node  
re-  
sour  
and  
prop  
er-  
ties  
as  
at-  
tribu  
that

#### **task.cont**

The  
con-  
text  
pass  
to  
Task  
ager

#### **task.shar**

Fals  
if  
Nod  
is  
lock  
True  
if  
it  
is  
not  
lock  
(The  
shar  
kwa  
arg  
of  
Task  
ager

#### **task.node**

The  
Nod  
ob-  
ject

#### **task.port**

Port  
be-  
long  
ing

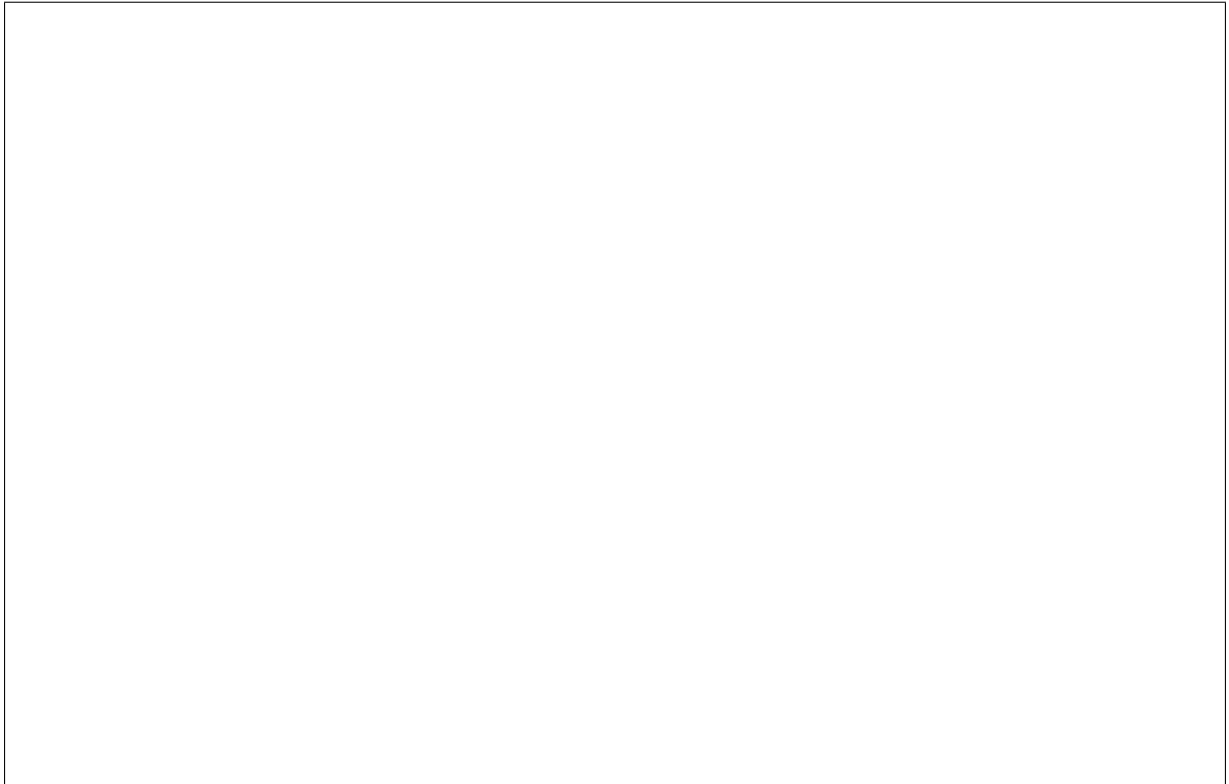
to  
 the  
 Nod

**task.port**  
 Port  
 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-  
 sign  
 to  
 the  
 Nod

**task.driv**  
 The  
 Driv  
 for  
 the  
 Nod  
 or  
 the  
 Driv  
 base  
 on  
 the  
 driv  
 kwa  
 of



provides an interface to handle this for you, making sure to release resources when the thread finishes

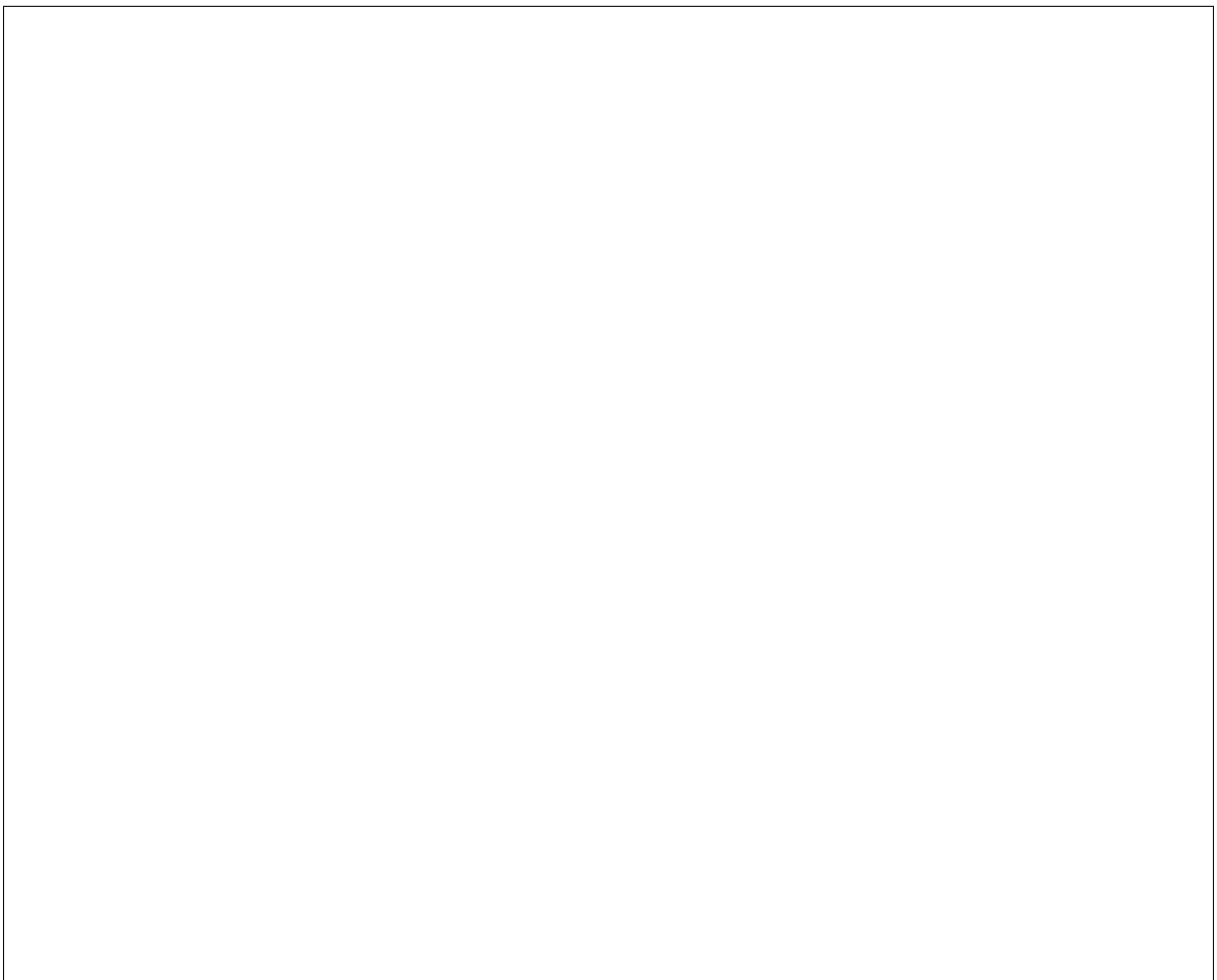
(successfully or if an exception occurs). Common use of this is within the Manager like so:

```

    ↪ utils.node_power_action, task, new_state)

```

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  
part  
of  
the  
spav-  
han-  
dling

**def**

→ `o`

→ `e`

→ `l`

→ `l`

→ `l`

→ `i`

→ `i`

→ `i`

→ `E`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

→ `l`

**with**

→ `t`

→ `m`

→ `a`

→ `l`

→ `n`

→ `i`

→ `l`

→ `p`

→ `'`

→ `w`

→ `'`

→ `a`

(continued from previous page)

```
→ utils.node_power_action, task, new_state)
```



**class** i

Base  
obj

Con  
text  
man  
ager  
for  
task

This  
class  
wrap  
the  
lock  
ing,  
drive  
load  
ing,  
and  
ac-  
qui-  
si-  
tion  
of  
re-  
latec  
re-  
sour  
(eg,

Node and Ports) when beginning a unit of work.

**downgra**

Dow  
grad  
the  
lock  
to  
a  
shar  
one.

**load\_di**

**propert**

process

Pro-  
cess  
the  
give  
even  
for  
the  
task.  
cur-  
rent  
state

Paramete

- **eve**  
the  
nam  
of  
the  
even  
to  
pro-  
cess
- **cal**  
op-  
tiona  
call-  
back  
to  
in-  
voke  
upon  
even  
tran-  
si-  
tion
- **cal**  
op-  
tiona  
args  
to  
pass  
to  
the  
call-

are no workers available (`err_handler` should accept arguments `node`, `prev_prov_state`, and `prev_target_state`)

back  
meth

- **cal**  
op-  
tiona  
kwa  
to  
pass  
to  
the  
call-  
back  
meth

- **err**  
op-  
tiona  
er-  
ror  
han-  
dler  
to  
in-  
voke  
if  
the  
call-  
back  
fails  
eg.  
be-  
caus  
there

- **tar**  
if  
spec  
i-  
fied,  
the  
tar-  
get  
pro-  
vi-  
sion  
state  
for

the target state from the fsm

the  
node  
Oth-  
er-  
wise  
use

#### **Raises**

In-  
valid  
State  
if  
the  
even  
is  
not  
al-  
lowe  
by  
the  
as-  
so-  
ci-  
ated  
state  
ma-  
chin

#### **release**

Un-  
lock  
a  
node  
and  
re-  
lease  
re-  
sour  
  
If  
an  
ex-  
clu-  
sive  
lock  
is  
held  
un-  
lock  
the  
node  
Re-

that this instance of TaskManager should no longer be accessed.

ground thread to do a task.

set  
at-  
tribu  
to  
mak  
it  
clear

#### set\_spa

Cre-  
ate  
a  
hool  
to  
han-  
dle  
ex-  
cep-  
tions  
whe  
spav  
ing  
a  
task  
  
Cre-  
ate  
a  
hool  
that  
gets  
called  
upon  
an  
ex-  
cep-  
tion  
be-  
ing  
raise  
from  
spav  
ing  
a  
back

#### Parame

•  
\_on

ject that was raised.

a  
 calla  
 ob-  
 ject,  
 its  
 first  
 pa-  
 ram-  
 e-  
 ter  
 shou  
 ac-  
 cept  
 the  
 Ex-  
 cep-  
 tion  
 ob-

• **arg**  
 ad-  
 di-  
 tion  
 args  
 pass  
 to  
 the  
 calla  
 ob-  
 ject.

• **kwa**  
 ad-  
 di-  
 tion  
 kwa  
 pass  
 to  
 the  
 calla  
 ob-  
 ject.

**spawn\_a**  
 Call  
 this  
 to  
 spav  
 a  
 thre

to  
 com  
 plete  
 the  
 task  
  
 The  
 spec  
 i-  
 fied  
 meth  
 will  
 be  
 calle  
 whe  
 the  
 Task  
 ager  
 in-  
 stan  
 ex-  
 its.

Paramete

- **\_sp**
  
 a
   
 meth
   
 that
   
 re-
   
 turn
   
 a
   
 Gree
   
 Thre
   
 ob-
   
 ject
- **arg**
  
 args
   
 pass
   
 to
   
 the
   
 meth
- **kwa**
  
 ad-
   
 di-
   
 tiona
   
 kwa
   
 pass

purpose when provided with one.

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-  
 clu-  
 sive  
 only  
 char  
 the  
 lock  
**Parame**  
 •  
**pur**  
 op-  
 tion-  
 ally  
 char  
 the  
 pur-  
 pose  
 of  
 the



lock

- **ret**  
whe  
to  
retry  
lock  
ing  
if  
it  
fails  
the  
class  
leve  
valu  
is  
used  
by  
de-  
fault

#### **Raises**

Nod  
Loc  
if  
an  
ex-  
clu-  
sive  
lock  
re-  
main  
on  
the  
node  
af-  
ter  
node

`ironic.`  
Shor  
cut  
for  
ac-  
quir  
ing  
a  
lock  
on  
a  
Nod

Parameters

context.

Returns

An instance of *TaskManager*.

ironic.

Decorator that requires an exclusive lock.

Decorator that requires a *TaskManager* as the first parameter.

Decorator that requires a *TaskManager* as the first parameter.

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  
con-  
duc-  
tor  
take  
over

**Parameter**  
**task**  
a  
Task  
ager  
in-  
stan

ironic.  
Add  
a  
se-  
cret  
to-  
ken  
to  
drive  
for  
IPA  
ver-  
i-  
fi-  
ca-  
tion.

**Parameter**  
  
•  
**node**  
Node  
ob-  
ject

generated in order to facilitate virtual media booting where the token is embedded into the configuration.

by [deploy]fast\_track\_timeout, then agent is presumed alive.

Paramet

- **node**  
A node object.
- **time**  
Heartbeat time out, default to *fast*

ironic.  
 Build a configuration drive from provide metadata network and user.  
 If uuid or name are not provided in the metadata they default to the node

accordingly.

Parameters

- **node**  
 an  
 Iron  
 node  
 ob-  
 ject.
- **config**  
 A  
 con-  
 fig-  
 drive  
 as  
 a  
 dict  
 with  
 keys  
 met  
 net  
 use  
 and  
 ven  
 (all  
 op-  
 tion:

Returns

A  
 gzip  
 and  
 base  
 en-  
 code  
 con-  
 fig-  
 drive  
 as  
 a  
 strin

ironic.

Put a failed node in CLE
 FAIL and main
 te- nanc
 (if need

Paramet

- **task**  
 a Task  
 ager  
 in- stan
- **log**  
 Mes  
 sage  
 to be  
 logg
- **err**  
 Mes  
 sage  
 for the  
 user  
 Op- tion  
 if not  
 pro- vide  
*logn*  
 is

used

- **trace**  
 When  
 to  
 log  
 a  
 trace  
 back  
 De-  
 fault  
 to  
 Fals
- **tear**  
 When  
 to  
 clear  
 up  
 the  
 PXE  
 and  
 DHCP  
 files  
 af-  
 ter  
 clear  
 ing.  
 De-  
 fault  
 to  
 True
- **set**  
 When  
 to  
 set  
 node  
 to  
 failed  
 state  
 De-  
 fault  
 to  
 True
- **set**  
 When  
 to



only if a clean step is being executed on a node.

set  
main  
te-  
nanc  
mod  
If  
Non  
main  
te-  
nanc  
mod  
will  
be  
set  
if  
and

`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-

stand-  
 ionic.  
 Clear  
 res-  
 cue  
 task  
 af-  
 ter  
 time  
 out.

Paramet  
 tas  
 a  
 Task  
 ager  
 in-  
 stan

ionic.

Put  
 a  
 faile  
 node  
 in  
 DE-  
 PLC  
 FAIL

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  
fail-  
ure.

ironic.

Che  
if  
the  
op-  
er-  
a-  
tion  
can  
be  
a  
strea  
line  
de-  
ploy

ations if we already have a ramdisk heartbeating through external means.

**Parameters**  
**task**  
 Task  
 agent  
 object

**Returns**  
 True  
 if  
 [de-  
 ploy  
 is  
 set  
 to  
 True  
 no  
 iSCSI  
 boot  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 pres  
 and

no `last_error` is present for the node indicating that there was a recent failure.

`ironic.`  
 Get  
 any  
 at-  
 tach  
 vif  
 ID  
 for  
 the  
 port

#### Parameters

**port**  
 The  
 port  
 ob-  
 ject  
 upon  
 which  
 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

port.

State  
 ex-  
 cep-  
 tion  
 upon  
 find-  
 ing  
 a  
 port  
 with  
 a  
 tran-  
 sien-  
 state  
 vif  
 on  
 the

ironic.

ironic.

ironic.  
 Has  
 a  
 sup-  
 plied  
 pass  
 wor

Paramet

val

Valu  
 to  
 be  
 hash

ironic.  
 De-  
 ter-  
 mine  
 if  
 the  
 to-  
 ken  
 was  
 gen-  
 er-  
 ated  
 for

tion.

a virtual floppy or as part of the virtual media image which is attached to the BMC.

token prior to rebooting the token. This is important as tokens provided through out of band means

out  
of  
band  
con-  
fig-  
u-  
ra-  
  
Iron  
sup-  
port  
the  
abil-  
ity  
to  
pro-  
vide  
con-  
fig-  
u-  
ra-  
tion  
data  
to  
the  
ager  
thro  
the  
  
This  
meth  
help  
us  
iden  
tify  
WH  
we  
did  
so  
as  
we  
dont  
need  
to  
re-  
mov  
reco  
of  
the

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.

False in all other cases.

**Parameters**  
**node**  
 Node object

**Returns**  
 True if the token was pre-generated as indicated by the node's drive field

ironic.  
 Determine if an agent token is present upon a node.

**Parameters**  
**node**  
 Node object

**Returns**



True
 if
 an
 ager
 valu
 is
 pres
 in
 a
 node
 drive
 field

ironic.
 Val-
 i-
 date
 if
 a
 sup-
 plied
 to-
 ken
 is
 valid
 for
 the
 node

**Parameter**
**node**
Node
ob-
ject

**Token**
A
to-
ken
valu
to
val-
i-
date
agai
the
drive
field
ager

**Returns**
True
if

perform a fast track sequence meaning that we already have a ramdisk running through another means like discovery. If not valid, False is returned.

by `[deploy]fast_track_timeout` and the power state for the machine is `POWER_ON`, then fast track is permitted.

Parameters

task

Task

object

object

object

Returns

True

if

the

last

heard

beat

that

was

reco

was

with

the

[de

ploy

set

ting

ironic.

Gen

er

ate

a

ran

dom

salt

with  
 the  
 in-  
 di-  
 ca-  
 tor  
 tag  
 for  
 pass  
 wor  
 type

Returns

a  
 valid  
 salt  
 for  
 use  
 with  
 cryp

ironic.  
 Do  
 cach  
 of  
 bios  
 set-  
 tings  
 if  
 sup-  
 port  
 by  
 drive

ironic.  
 Cack  
 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  
*Non*  
is  
re-  
turn

#### Paramet

**tas**  
a  
Task  
ager  
in-  
stan

#### Raises

Driv  
Op-  
er-  
a-  
tionl  
or  
its  
deriv  
tive  
in  
case  
of  
drive  
run-  
time  
er-  
ror.

#### Raises

Un-

face or `get_boot_mode()` method is not supported.

Returns

Boolean value representing the boot mode. One of `ironic.common.boot` or `Non` if boot mode cannot be discovered. `ironic.Chassis.power.state` or `reset` for a node. Perform the

re-  
 ques-  
 pow  
 ac-  
 tion  
 if  
 the  
 tran-  
 si-  
 tion  
 is  
 re-  
 quir

Paramet

- **task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **new**  
 Any  
 pow  
 state  
 from  
 iron
- **time**  
 time  
 out  
 (in  
 sec-  
 onds  
 pos-  
 i-  
 tive  
 in-  
 te-

indicates to use default timeout.

info is specified.

ger  
(>  
0)  
for  
any  
pow  
state  
Non

Raises  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
state  
is  
spec  
i-  
fied  
or  
the  
wron  
drive

Raises  
Stor  
ageE  
ror  
whe  
a  
fail-  
ure  
oc-  
curs  
up-  
dat-  
ing  
the  
node  
stor-  
age  
in-  
ter-  
face



upon setting power on.

action.

Raises

othe  
 ex-  
 cep-  
 tions  
 by  
 the  
 node  
 pow  
 drive  
 if  
 som  
 thing  
 wron  
 oc-  
 curre  
 dur-  
 ing  
 the  
 pow

ironic.  
 Set  
 the  
 boot  
 de-  
 vice  
 for  
 a  
 node  
  
 If  
 the  
 node  
 that  
 the  
 boot  
 de-  
 vice  
 char  
 is  
 be-  
 ing  
 re-  
 ques  
 for  
 is  
 in  
 ADC  
 ING

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.

state

#### **Parameter**

- **task**  
a Task  
ager  
in-  
stan
- **dev**  
Boo  
de-  
vice  
Val-  
ues  
are  
veno  
spec
- **per**  
Whe  
to  
set  
next  
boot  
or  
mak  
the  
char  
per-  
ma-  
nent  
De-  
fault  
Fals

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the

face fails.

interface.

val-  
i-  
da-  
tion  
of  
the  
Man  
age-  
men  
ter-

ironic.  
Set  
the  
boot  
mod  
for  
a  
node  
  
Sets  
the  
boot  
mod  
for  
a  
node  
if  
the  
node  
drive  
in-  
ter-  
face  
con-  
tains  
a  
man  
age-  
men

If  
the  
node  
that  
the  
boot  
mod  
char  
is  
be-

boot mode will not be set as that change could potentially result in the future running state of an adopted node being modified erroneously.

Parameters

- **task**  
 a Task agent instance
- **mode**  
 Boot mode values are one of *ironic-com* *boot*

Raises

InvalidParameterException if the validation of the

face fails.

Man  
age-  
men  
ter-

**Raises**

Driv  
Op-  
er-  
a-  
tionl  
or  
its  
deriv  
tive  
in  
case  
of  
drive  
run-  
time  
er-  
ror.

**Raises**

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
cur-  
rent  
drive  
does  
not  
have  
ven-  
dor  
in-  
ter-  
face

or method is unsupported.

ironic.

Wai  
for  
node  
to

be  
 in  
 new  
 pow  
 state

Paramet

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan
- **new**  
 the  
 de-  
 sired  
 new  
 pow  
 state  
 one  
 of  
 the  
 pow  
 state  
 in  
*irc*  
*com*  
*sta*
- **tim**  
 num  
 ber  
 of  
 sec-  
 onds  
 to  
 wait  
 be-  
 fore  
 giv-  
 ing  
 up.  
 If  
 not  
 spec  
 i-

the conductor.power\_state\_change\_timeout config value.

fied,  
uses

#### Raises

Power-  
State-  
Failure  
if  
time-  
out

ironic.

ironic.

ironic.

Noti-  
tify  
the  
con-  
duc-  
tor  
to  
re-  
sum  
an  
op-  
er-  
a-  
tion.

#### Parameters

- **task**  
the  
task
- **operation**  
the  
op-  
er-  
a-  
tion.  
a  
string

ironic.

Pow  
 ers  
 on  
 node  
 if  
 it  
 is  
 pow  
 ered  
 off  
 and  
 has  
 a  
 Sma  
 NIC  
 port

**Parameter**  
**task**  
 A  
 Task  
 ager  
 ob-  
 ject

**Returns**  
 the  
 pre-  
 vi-  
 ous  
 pow  
 state  
 or  
 Non  
 if  
 no  
 char  
 were  
 mad

**Raises**  
 ex-  
 cep-  
 tion.  
 if  
 ager  
 sta-  
 tus  
 didn  
 matc  
 the  
 re-  
 quir



tempts.

the power state of a node.

sta-  
tus  
af-  
ter  
max  
retry  
at-

ironic.  
Set  
the  
node  
power  
state  
if  
er-  
ror  
oc-  
curs  
  
This  
hool  
gets  
called  
upon  
an  
ex-  
cep-  
tion  
be-  
ing  
raise  
when  
spaw  
ing  
the  
work  
threa  
to  
chan

#### Parameter

- e  
the  
ex-  
cep-  
tion  
ob-

ject  
 that  
 was  
 raise

- **node**  
 an  
 Iron  
 node  
 ob-  
 ject.

- **power**  
 the  
 pow  
 state  
 to  
 set  
 on  
 the  
 node

ironic.  
 Han  
 dle  
 the  
 pow  
 state  
 for  
 a  
 node  
 re-  
 con-  
 fig-  
 u-  
 ra-  
 tion.

Pow  
 ers  
 the  
 node  
 on  
 if  
 and  
 only  
 if  
 it  
 has  
 a  
 Sma

reconfiguration, then restores the power state.

NIC  
port  
Yiel  
for  
the  
ac-  
tual

#### Paramet

**tas**  
A  
Task  
ager  
ob-  
ject.

ironic.

Set  
the  
node  
pro-  
vi-  
sion  
ing  
state  
if  
er-  
ror  
oc-  
curs

This  
hool  
gets  
called  
upon  
an  
ex-  
cep-  
tion  
be-  
ing  
raise  
whe  
spaw  
ing  
the  
worl  
to  
do

provisioning to a node like deployment, tear down, or cleaning.

Parameters

- **exception**  
 the exception object that was raised
- **node**  
 an Ionic node object.
- **provision**  
 the provision state to be set on the node
- **target**  
 the target provision state to be set on the

then caller needs to explicitly indicate it.

node  
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-  
fault  
If  
node  
shou  
not  
be  
save

## Paramet

- **nod**  
an  
Iron  
node

ob-  
ject.

- **sav**  
 Boo  
 True  
 (de-  
 fault  
 to  
 save  
 the  
 node  
 Fals  
 oth-  
 er-  
 wise

ironic.

Clea  
 res-  
 cue  
 task  
 af-  
 ter  
 time  
 out  
 or  
 fail-  
 ure.

Paramet

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan
- **msg**  
 a  
 mes-  
 sage  
 to  
 set  
 into  
 node  
 last\_  
 field

state. By default node would be transitioned to a failed state.

`ironic.`

Cha  
the  
node  
pow  
state  
if  
pow  
is  
not  
Non

## Paramet

- **tas**  
A  
Task  
ager  
ob-  
ject
  - **pow**  
pow  
state
- `ironic.`  
Che  
if

node  
 clear  
 ing  
 need  
 to  
 be  
 skip  
 for  
 an  
 spe-  
 cific  
 node

Paramet

nod  
 the  
 node  
 to  
 con-  
 sider

ironic.  
 Han  
 dle  
 spaw  
 ing  
 er-  
 ror  
 for  
 node  
 clear  
 ing.

ironic.  
 Han  
 dle  
 spaw  
 ing  
 er-  
 ror  
 for  
 node  
 de-  
 ploy  
 ing.

ironic.  
 Han  
 dle  
 spaw  
 ing  
 er-  
 ror



for  
 node  
 res-  
 cue.

ironic  
 Stor  
 cer-  
 tifi-  
 cate  
 re-  
 ceiv  
 from  
 the  
 ager  
 and  
 re-  
 turn  
 its  
 path

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  
 clea  
 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

**Parameter**

**node**  
 an  
 Iron  
 node  
 ob-  
 ject.

**Raises**

In-  
 valid  
 Pa-  
 ram.  
 e-  
 ter-  
 Valu  
 if

tain traits that are not set on the node.

the  
in-  
stan-  
trait  
are  
badl  
for-  
mat-  
ted,  
or  
con-  
  
ironic.  
Val-  
i-  
date  
the  
con-  
sis-  
tenc  
of  
phys  
i-  
cal  
net-  
worl  
of  
port  
in  
a  
port  
grou  
Val-  
i-  
date  
the  
con-  
sis-  
tenc  
of  
a  
port  
phys  
i-  
cal  
net-  
worl  
with  
othe  
port

same portgroup. All ports in a portgroup should have the same value (which may be None) for their `physical_network` field.

ing validation criteria:

raise `PortgroupPhysnetInconsistent`. This shouldnt ever happen.

in the portgroup, we raise `exception.Conflict`.

ment mapping algorithm should operate in a legacy (physical network unaware) mode for this port or portgroup. This allows existing ironiC nodes to continue to function after an upgrade to a release including physical network support.

has  
a  
phys  
i-  
cal  
net-  
worl  
that  
is  
in-  
con-  
sis-  
tent  
with  
othe  
port

If  
a  
port  
phys  
i-  
cal  
net-  
worl  
is  
Non  
this  
in-  
di-  
cate  
that  
iron  
ics  
VIF  
at-  
tach

## Paramet

- **tas**  
a  
Task  
ager  
in-  
stan

ent physical network.

- port  
 a  
 port  
 ob-  
 ject  
 to  
 be  
 val-  
 i-  
 date

**Raises**  
 Con-  
 flict  
 if  
 the  
 port  
 is  
 a  
 men-  
 ber  
 of  
 a  
 port-  
 grou-  
 whic-  
 is  
 on  
 a  
 dif-  
 fer-

**Raises**  
 Port-  
 grou-  
 Phys-  
 net-  
 Inco-  
 sis-  
 tent  
 if  
 the  
 port-  
 port-  
 grou-  
 has  
 port-  
 whic-  
 are  
 not

signed the same physical network.

all  
as-  
  
ironic.  
Che  
if  
the  
time  
is  
with  
the  
pre-  
vi-  
ous  
time  
out  
sec-  
onds  
from  
now

Paramet

- - val**  
a  
strin  
rep-  
re-  
sent  
ing  
date  
and  
time  
or  
Non
  - tim**  
time  
out  
in  
sec-  
onds

ironic.  
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.ansible module

ironic.

### ironic.conf.api module

ironic.

### ironic.conf.audit module

ironic.

### ironic.conf.auth module

ironic.

Add

auth

op-

tions

to

sam

ple

con-

fig

As

thes

are

dy-

nam

i-

cally

auth\_plugins when generating sample config.

istered at runtime depending on auth plugin used.

reg-  
is-  
terece  
at  
run-  
time  
this  
adds  
op-  
tions  
for  
mos  
used

ironic.  
Reg  
is-  
ter  
sess

and  
auth  
relat  
op-  
tions

Reg  
is-  
ters  
only  
ba-  
sic  
auth  
op-  
tions  
shar  
by  
all  
auth  
plu-  
g-  
ins.  
The  
rest  
are  
reg-

**ironic.conf.cinder module**

ironic.

ironic.

**ironic.conf.conductor module**

ironic.

**ironic.conf.console module**

ironic.

**ironic.conf.database module**

ironic.

**ironic.conf.default module**

ironic.

**ironic.conf.deploy module**

ironic.

**ironic.conf.dhcp module**

ironic.

**ironic.conf.drac module**

ironic.

**ironic.conf.glance module**

ironic.

ironic.

**ironic.conf.healthcheck module**

ironic.

**ironic.conf.ibmc module**

ironic.

**ironic.conf.ilo module**

ironic.

**ironic.conf.inspector module**

ironic.

ironic.

**ironic.conf.ipmi module**

ironic.

**ironic.conf.irmc module**

ironic.

**ironic.conf.iscsi module**

ironic.

**ironic.conf.json\_rpc module**

ironic.

ironic.

**ironic.conf.metrics module**

ironic.

**ironic.conf.metrics\_statsd module**

ironic.

**ironic.conf.neutron module**

ironic.

ironic.

**ironic.conf.nova module**

ironic.

ironic.

## ironic.conf.opts module

ple. The first element is the name of the group, the second element is the options.

ironic.  
Re-  
turn  
a  
list  
of  
oslo  
op-  
tions  
avai-  
able  
in  
Iron  
code  
  
The  
re-  
turn  
list  
in-  
clud-  
all  
oslo  
op-  
tions  
Each  
el-  
e-  
men-  
of  
the  
list  
is  
a  
tu-

The  
func-  
tion  
is  
dis-  
cov-  
er-  
able  
via  
the  
iron-  
en-  
try

options.

**Returns**

a list of (group) options tuples.

ironic.

**ironic.conf.pxe module**

ironic.

**ironic.conf.redfish module**

ironic.

**ironic.conf.service\_catalog module**

ironic.

ironic.

**ironic.conf.snmp module**

ironic.

**ironic.conf.swift module**

ironic.

ironic.

**ironic.conf.xclarity module**

ironic.

**Module contents**

**ironic.db package**

**Subpackages**

**ironic.db.sqlalchemy package**

**Submodules**



## ironic.db.sqlalchemy.api module

SQL  
stor-  
age  
back  
end.

**class** `ironic.db.sqlalchemy.api`  
Base  
*ironic.db.sqlalchemy.api*  
*Connection*  
SqlA  
con-  
nec-  
tion.

**add\_node**  
Add  
tag  
to  
the  
node  
If  
the  
node  
and  
tag  
pair  
al-  
read  
ex-  
ists,  
this  
shou  
still  
suc-  
ceed

### Parameters

- **node**  
The  
id  
of  
a

node

- **tag**  
A tag string

**Returns**  
the Node Tag object.

**Raises**  
NodeNotFoundError if the node is not found

**add\_node**  
Add trait to the node  
  
If the node and trait pair already exists, this should still succeed

**Parameters**

- **node**  
The

id  
 of  
 a  
 node

- **traits**  
 A  
 trait  
 string

- **version**  
 the  
 ver-  
 sion  
 of  
 the  
 ob-  
 ject.

**Returns**

the  
 Node  
 Trait  
 ob-  
 ject.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 addi  
 the  
 trait  
 wou  
 ex-  
 ceed  
 the  
 per-  
 node  
 trait  
 limi

**Raises**

Node  
 Not-  
 Four  
 if

the  
 node  
 is  
 not  
 found

**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-  
 vert  
 then  
 to  
 UUID  
 It  
 fails

early if any identities cannot possible be used as names or UUIDs.

**Parame**

**ide**  
 List  
 of  
 iden  
 ti-  
 ties.

**Returns**

A

map
 ping
 from
 re-
 ques
 iden
 ti-
 ties
 to
 node
 UUI

Raises

Nod
 Not-
 Foun
 if
 som
 iden
 ti-
 ties
 were
 not
 foun
 or
 can-
 not
 be
 valid
 nam
 or
 UUI

check\_v

Che
 the
 who
 data
 for
 in-
 com
 pat-
 i-
 ble
 ob-
 jects
 This
 scan
 all
 the
 ta-
 bles

not specified in `ironic.common.release_mappings.RELEASE_MAPPING`. This includes objects that have null version values.

in  
 search  
 of  
 ob-  
 jects  
 that  
 are  
 not  
 sup-  
 port  
 i.e.,  
 thos  
 that  
 are

**Parameters**  
**ignore**  
 List  
 of  
 mod  
 nam  
 to  
 skip

**Returns**  
 A  
 Boo  
 True  
 if  
 all  
 the  
 ob-  
 jects  
 have  
 sup-  
 port  
 ver-  
 sion  
 Fals  
 oth-  
 er-  
 wise

**clear\_r**

**clear\_r**

**create\_**  
 Cre-  
 ate

a  
 new  
 al-  
 lo-  
 ca-  
 tion.

**Parameters**  
**value**  
 Dict  
 of  
 val-  
 ues  
 to  
 cre-  
 ate  
 an  
 al-  
 lo-  
 ca-  
 tion  
 with

**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

Paramet

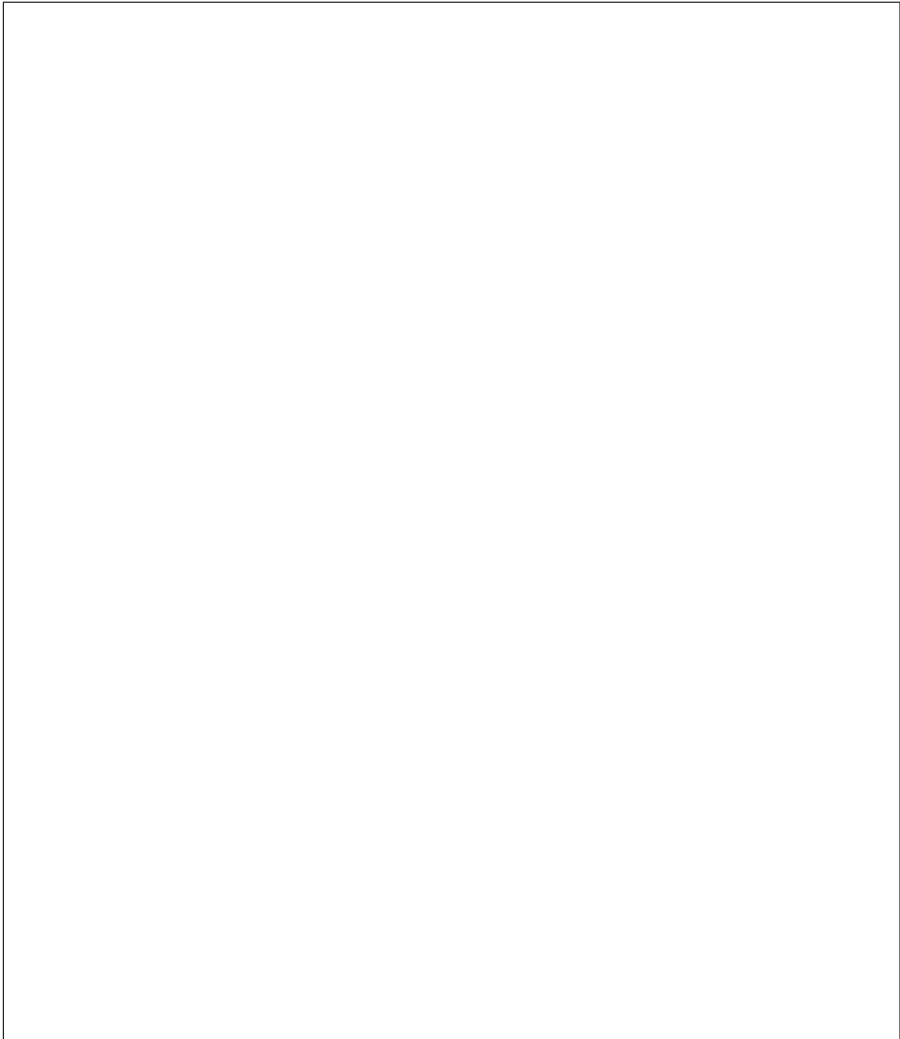
- **node**  
 The  
 node  
 id.
- **set**  
 A  
 list  
 of  
 BIO  
 Set-  
 ting  
 to  
 be  
 cre-  
 ated



(continues on next page)



(continued from previous page)



- **version**  
the  
ver-  
sion  
of  
the  
ob-  
ject.

**Returns**  
A  
list  
of  
BIO  
Set-  
ting  
ob-  
ject.

**Raises**  
Nod

Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 found

**Raises**

BIO  
 Set-  
 tin-  
 gAl-  
 read  
 ists  
 if  
 any  
 of  
 the  
 set-  
 ting  
 reco  
 al-  
 read  
 ex-  
 ists.

**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  
 ex-  
 am-  
 ple:



**Raises**  
 De-  
 ploy  
 plat-  
 eDu  
 pli-  
 cate  
 Nam  
 if  
 a  
 de-  
 ploy  
 tem-  
 plate  
 with

the  
 same  
 name  
 ex-  
 ists.

**Raises**

De-  
 ploy  
 plate  
 read  
 ists  
 if  
 a  
 de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 same  
 UI  
 ex-  
 ists.

**Returns**

A  
 de-  
 ploy  
 tem-  
 plate

**create\_**

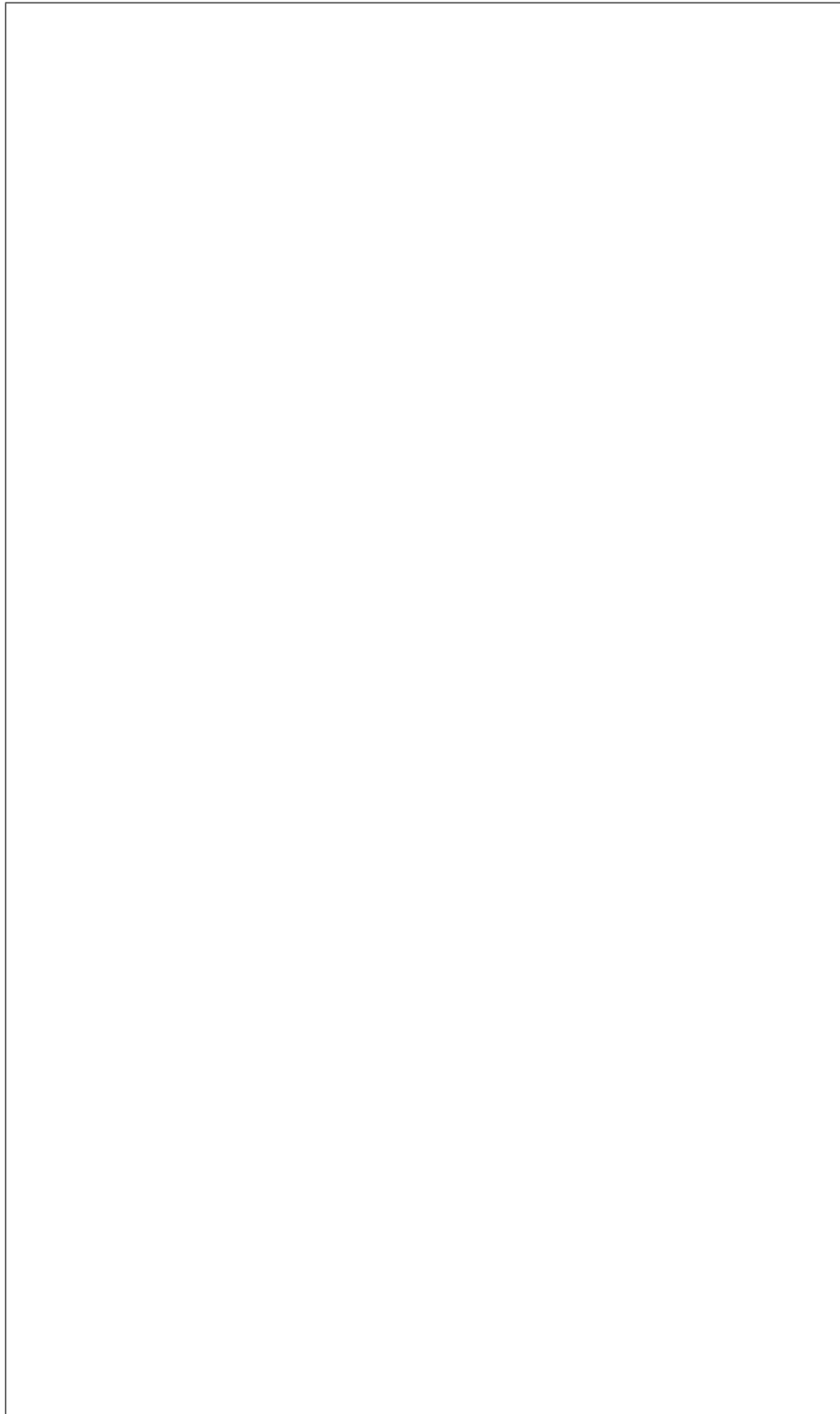
Cre-  
 ate  
 a  
 new  
 node

**Parame**

**val**  
 A  
 dict  
 con-  
 tain-  
 ing  
 sev-  
 eral  
 item  
 used  
 to  
 iden-  
 tify

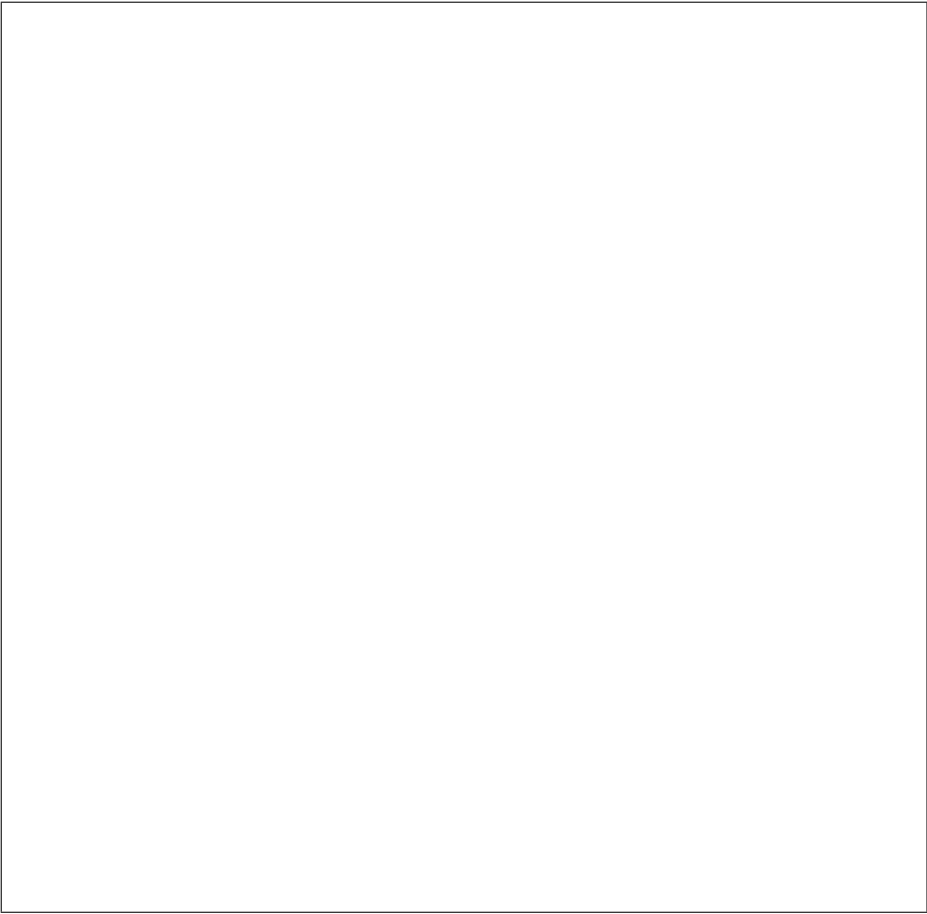
and  
track  
the  
node  
and  
sev-

eral dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
val-  
ues  
con-  
tains  
tags  
or  
trait

**Returns**

A  
node

**create\_**

Cre-  
ate  
a

	new	
	port	
	<b>Parameters</b>	
	<b>values</b>	
	Dict	
	of	
	val-	
	ues.	
	<b>create_</b>	
	Cre-	
	ate	
	a	
	new	
	port	
	grou	
	<b>Parameters</b>	
	<b>values</b>	
	Dict	
	of	
	val-	
	ues	
	with	
	the	
	fol-	
	low-	
	ing	
	keys	
	id	
	uuid	
	nam	
	node	
	ad-	
	dres	
	ex-	
	tra	
created_at		
updated_at		
	<b>Returns</b>	
	A	
	port	
	grou	
	<b>Raises</b>	
	Port	
	grou	
	pli-	
	cate	
	Nam	
	<b>Raises</b>	
	Port	

grou  
 MA  
 read  
 ists

**Raises**

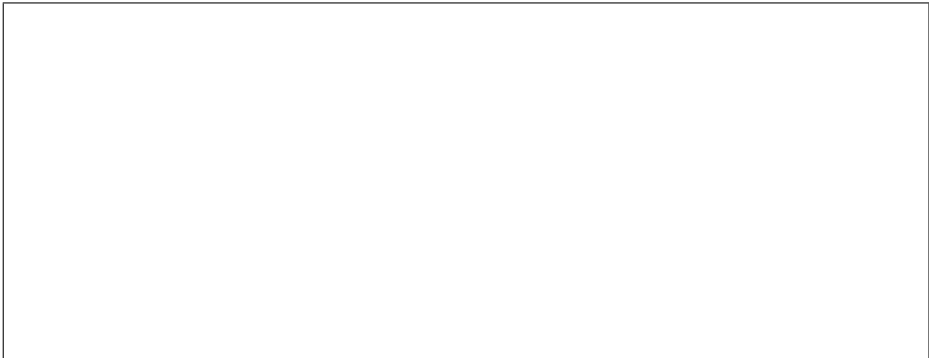
Port  
 grou  
 read  
 ists

**create\_**

Cre-  
 ate  
 a  
 new  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**Parame**

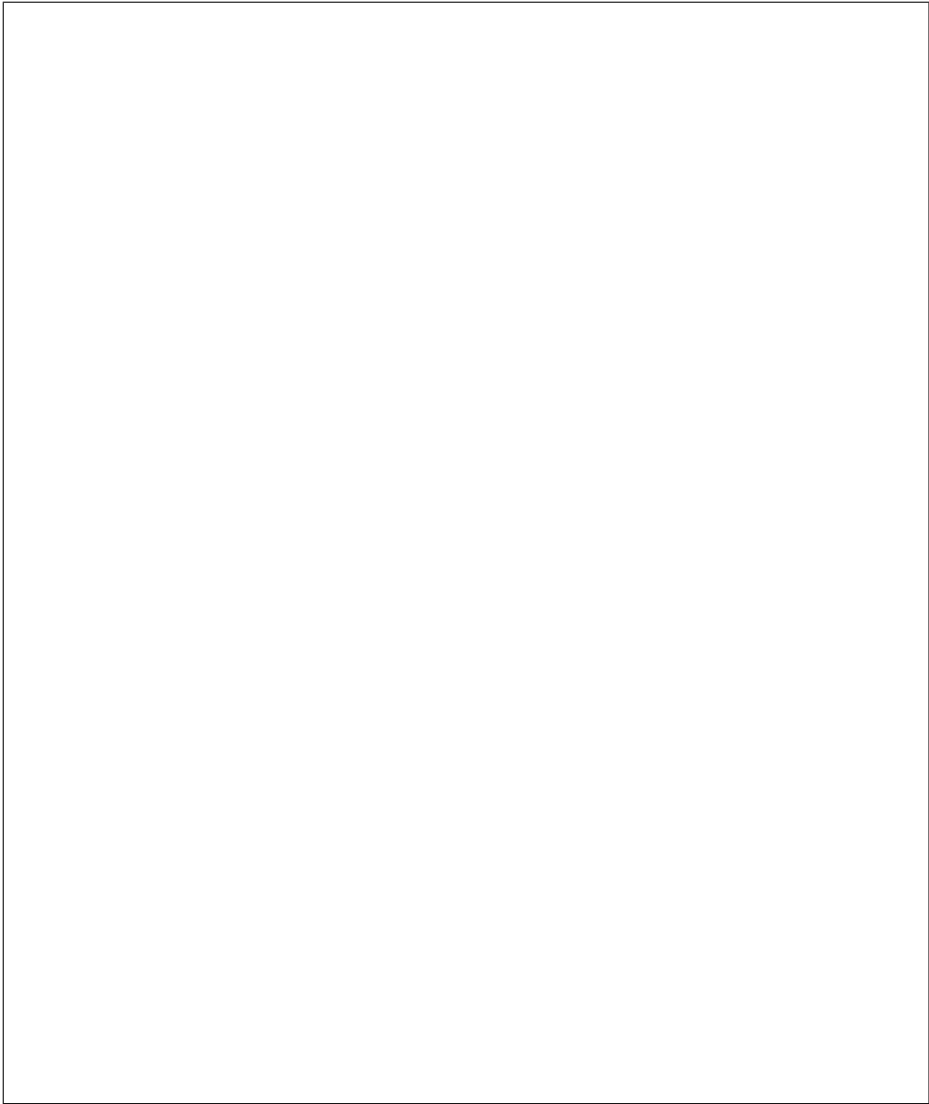
**con**  
 Dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 con-  
 nec-  
 tor.  
 Ex-  
 am-  
 ple:



(continues on next page)



(continued from previous page)



**Returns**

A  
vol-  
ume  
con-  
nec-  
tor.

**Raises**

Vol-  
ume  
Con  
nec-  
torT  
pe-  
An-  
dI-  
dAl-  
read

ists with a matching type and connector\_id.

ready exists.

ists  
 If  
 a  
 con-  
 nec-  
 tor  
 al-  
 read  
 ex-

Raises

Vol-  
 ume  
 Con  
 nec-  
 torA  
 read  
 ists  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 sam  
 UUI  
 al-

create\_

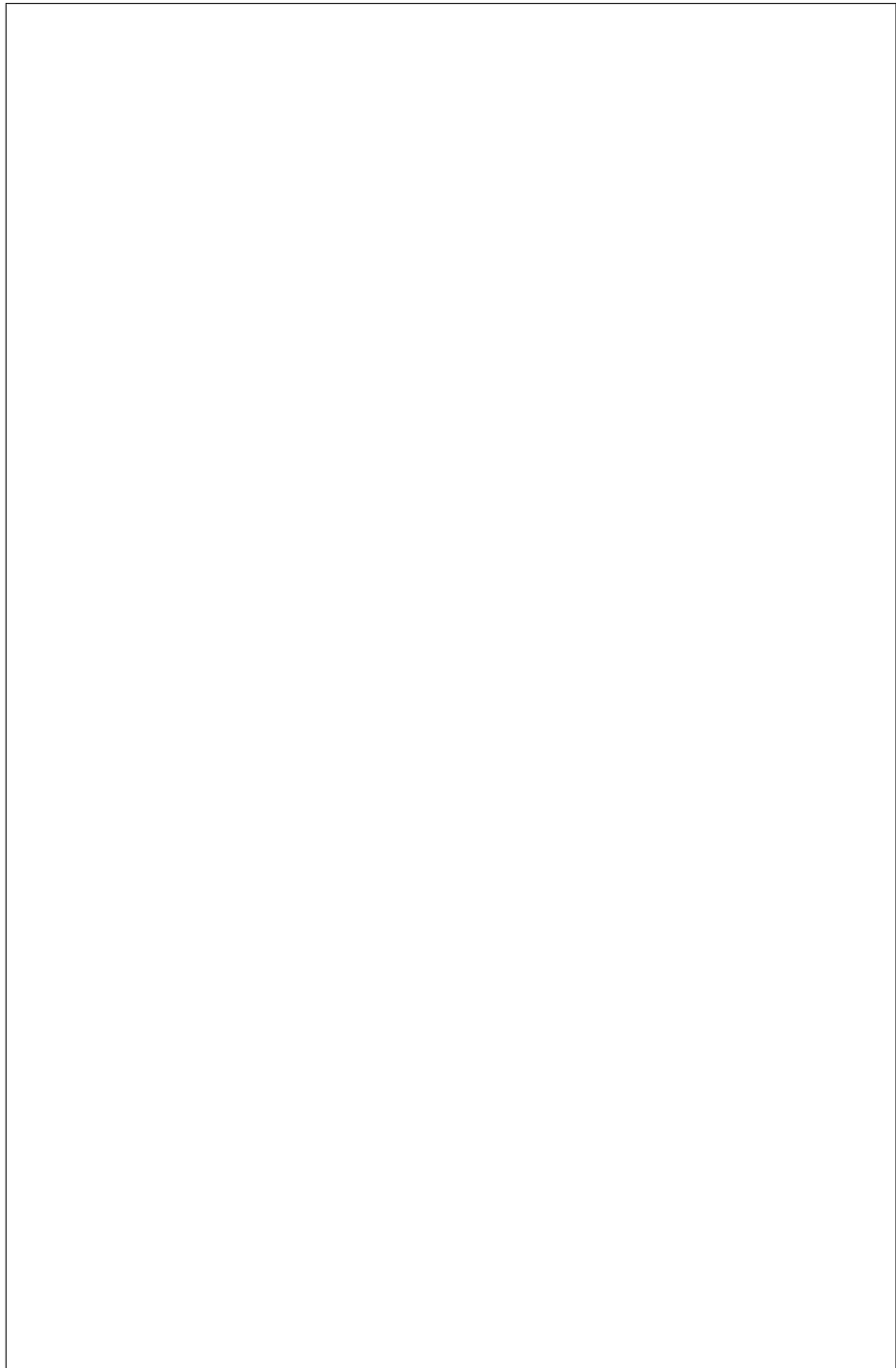
Cre-  
 ate  
 a  
 new  
 vol-  
 ume  
 tar-  
 get.

Parame

tar  
 Dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 in-

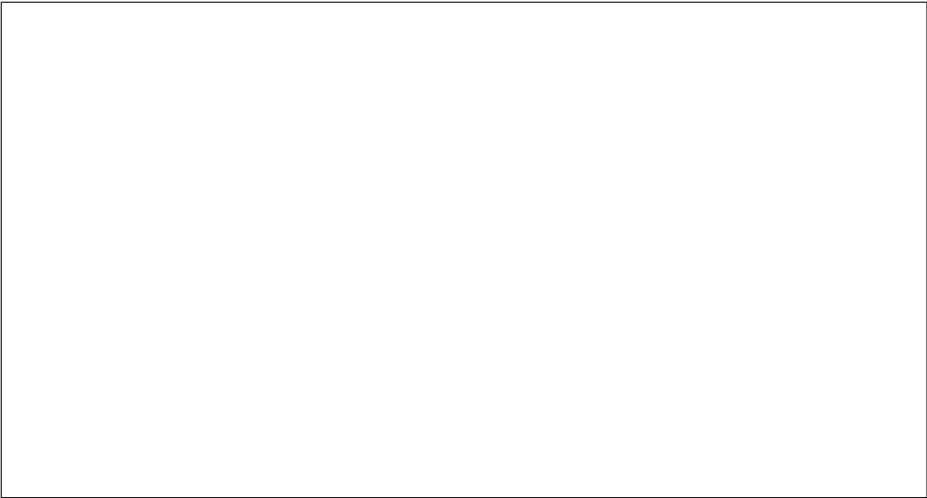
for-  
ma-  
tion  
about  
the  
vol-  
ume  
tar-  
get.  
Ex-

ample:



(continues on next page)

(continued from previous page)



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-  
get  
al-  
read  
ex-  
ists  
with

**Raises**

Vol-  
ume  
ge-  
tAl-  
read  
ists  
if

a  
 vol-  
 ume  
 tar-  
 get  
 with  
 the  
 sam  
 UUI  
 ex-  
 ists.

delete\_

Dele  
 a  
 list  
 of  
 BIO  
 set-  
 tings

Parame

- **nod**  
 The  
 node  
 id.
- **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-
 Foun
 if
 any
 of
 BIO
 set-
 ting
 nam
 is
 not
 foun

**delete\_**
 Dele
 spec
 i-
 fied
 tag
 from
 the
 node

**Parame**

- nod**
 The
 id
 of
 a
 node
- tag**
 A
 tag
 strin

**Raises**

Nod
 Not-
 Foun
 if
 the
 node
 is
 not
 foun

**Raises**

Nod  
Tag-  
Not-  
Foun  
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-  
Foun  
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.

**Parameter**  
**all**  
 Al-  
 lo-  
 ca-  
 tion  
 ID  
 or  
 UUI

**Raises**  
 Al-  
 lo-  
 ca-  
 tion.  
 Not-  
 Foun

**destroy**  
 De-  
 stroy  
 a  
 chas  
 sis.

**Parameter**  
**cha**  
 The  
 id  
 or  
 the  
 uuid  
 of  
 a  
 chas  
 sis.



**destroy**

De-  
stroy  
a  
de-  
ploy  
men  
tem-  
plate

**Parame**

**tem**  
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-  
ist.

**destroy**

De-  
stroy  
a  
node  
and  
its  
as-  
so-  
ci-  
ated  
re-  
sour

nectors, and volume targets.

De-  
 stroy  
 a  
 node  
 in-  
 clud  
 ing  
 any  
 as-  
 so-  
 ci-  
 ated  
 port  
 port  
 grou  
 tags  
 trait  
 vol-  
 ume  
 con-

Paramete

**node**  
 The  
 ID  
 or  
 UUID  
 of  
 a  
 node

**destroy**

De-  
 stroy  
 an  
 port

Paramete

**port**  
 The  
 id  
 or  
 MAC  
 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  
 UUI  
 or  
 in-  
 te-  
 ger  
 ID  
 of  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**Raises**

Vol-  
 ume

does not exist.

Con  
nec-  
torN  
Four  
If  
a  
vol-  
ume  
con-  
nec-  
tor  
with  
the  
spec  
i-  
fied  
iden

**destroy**  
De-  
stroy  
a  
vol-  
ume  
tar-  
get.

**Parameter**  
**id**  
The  
UUID  
or  
in-  
te-  
ger  
ID  
of  
a  
vol-  
ume  
tar-  
get.

**Raises**  
Vol-  
ume  
get-  
Not-  
Four  
if  
a  
vol-

exist.

ume  
tar-  
get  
with  
the  
spec  
i-  
fied  
iden  
does  
not

get\_act  
Re-  
triev  
hard  
ware  
type  
for  
the  
reg-  
is-  
tere  
and  
ac-  
tive  
con-  
duc-  
tors.

Parame  
use  
Whe  
to  
fac-  
tor  
con-  
duc-  
tor\_  
into  
the  
keys

Returns

A  
dict  
whic  
map  
hard  
ware

ple:



type  
 nam  
 to  
 the  
 set  
 of  
 host  
 whic  
 sup-  
 port  
 then  
 For  
 ex-  
 am-

**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-  
Foun

**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-  
Four

**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

Param

- **fil**  
 Fil-  
 ters  
 to  
 ap-  
 ply.  
 De-  
 fault  
 to  
 Non

**node\_u**  
 uuid  
 of  
 node

**state**  
 al-  
 lo-  
 ca-  
 tion  
 state

**resour**  
 re-  
 ques  
 re-  
 sour  
 class

- **lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 al-  
 lo-  
 ca-  
 tion  
 to  
 re-

turn

- **max**  
The last item of the previous page we return the next result set.

- **sort**  
Attribute by which results should be sorted

- **sort**  
Direction in which results should be sorted (ascending or descending)

**Returns**  
 A list of

al-  
lo-  
ca-  
tions

#### get\_bio

Re-  
triev  
BIO  
set-  
ting  
valu

#### Parame

- **node**  
The  
node  
id.
- **name**  
Strin  
con-  
tain-  
ing  
nam  
of  
BIO  
set-  
ting  
to  
be  
re-  
triev

#### Returns

The  
BIO  
Set-  
ting  
ob-  
ject.

#### Raises

Nod  
Not-  
Four  
if  
the  
node  
is

not  
 found

**Raises**

BIO  
 Set-  
 ting-  
 Not-  
 Foun  
 if  
 the  
 BIO  
 set-  
 ting  
 is  
 not  
 foun

**get\_bic**

Re-  
 triev  
 BIO  
 set-  
 tings  
 of  
 a  
 give  
 node

**Parame**

**nod**  
 The  
 node  
 id.

**Returns**

A  
 list  
 of  
 BIO  
 Set-  
 ting  
 ob-  
 jects

**Raises**

Nod  
 Not-  
 Foun  
 if  
 the  
 node  
 is  
 not

found

**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**

Return  
a  
list  
of  
chas  
sis.

Paramete

- **limit**  
Max  
i-  
mun  
num  
ber  
of  
chas  
sis  
to  
re-  
turn
- **marker**  
the  
last  
item  
of  
the  
pre-  
vi-  
ous  
page  
we  
re-  
turn  
the  
next  
re-  
sult  
set.
- **sortBy**  
At-  
tribu  
by  
whic  
re-  
sults

should
 be
 sorted

- **sort**  
 direction  
 in  
 which  
 results  
 should  
 be  
 sorted  
 (ascending  
 or descending)

**get\_connections**  
 Retrieve  
 a  
 connection  
 descriptors  
 service  
 records  
 from  
 the  
 database

**Parameters**

- **hostname**  
 The  
 host  
 name  
 of  
 the  
 connection  
 descriptor  
 service
- **only**  
 Specify  
 if  
 the

online field is ignored if this value is set to None.

meet the specified online expectation.

fil-  
 ter  
 valu  
 on  
 the  
*on-*  
*line*  
 field  
 whe  
 quer  
 ing  
 con-  
 duc-  
 tors.  
 The

Returns

A  
 con-  
 duc-  
 tor.

Raises

Con  
 duc-  
 torN  
 Four  
 if  
 the  
 con-  
 duc-  
 tor  
 with  
 give  
 host  
 nam  
 does  
 not  
 ex-  
 ist  
 or  
 does

get\_con

Re-  
 turn  
 a  
 list  
 of  
 con-  
 duc-



tors.

Param
 eters

- **limit**  
 Maximum number of conductors to return
- **max**  
 the last item of the previous page we return the next result set.
- **sort**  
 Attribute by which results should be sorted
- **sort**

di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**get\_deploy**  
 Re-  
 triev  
 a  
 de-  
 ploy  
 men  
 tem-  
 plate  
 by  
 ID.

**Parameter**  
**template\_id**  
 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-

ist.

**Returns**

A  
 de-  
 ploy  
 tem-  
 plate

**get\_dep**

Re-  
 triev  
 a  
 de-  
 ploy  
 men  
 tem-  
 plate  
 by  
 nam

**Paramete**

**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-  
 plate  
 by  
 UUI

**Paramete**  
**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-  
 ist.

**Returns**  
 A  
 de-  
 ploy  
 tem-  
 plate

get\_dep

Re-  
triev  
a  
list  
of  
de-  
ploy  
men  
tem-  
plate

Parame

- **lim**  
Max  
i-  
mun  
num  
ber  
of  
de-  
ploy  
tem-  
plate  
to  
re-  
turn
- **mar**  
The  
last  
item  
of  
the  
pre-  
vi-  
ous  
page  
we  
re-  
turn  
the  
next  
re-  
sult  
set.
-

sor  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- sor  
 Di-  
 rec-  
 tion  
 in  
 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  
 a  
 list  
 of  
 nam

**Parameters**

**name**  
 List  
 of  
 name  
 to  
 fil-  
 ter  
 by.

**Returns**

A  
 list  
 of  
 de-  
 ploy  
 tem-  
 plate

**get\_node**

Re-  
 turn  
 a  
 node

**Parameters**

**node**  
 The  
 id  
 of  
 a  
 node

**Returns**

A  
 node

**get\_node**

Re-  
 turn  
 a  
 node

**Parameters**

**instance**  
 The  
 in-  
 stan-  
 uuid  
 to  
 sear-  
 for.

**Returns**

A

node

**Raises**

In-  
stan-  
ceNo-  
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

**Parameter**

**node**  
The  
log-  
i-  
cal  
nam  
of  
a  
node

**Returns**

A  
node

**get\_node**

Find  
a  
node  
by



any  
 matc  
 ing  
 port  
 ad-  
 dres

**Parame**

**add**  
 list  
 of  
 port  
 ad-  
 dres  
 (e.g.  
 MA

**Returns**

Nod  
 ob-  
 ject.

**Raises**

Nod  
 Not-  
 Foun  
 if  
 none  
 or  
 sev-  
 eral  
 node  
 are  
 foun

**get\_noo**

Re-  
 turn  
 a  
 node

**Parame**

**nod**  
 The  
 uuid  
 of  
 a  
 node

**Returns**

A  
 node

**get\_noo**

Re-  
 turn  
 a  
 list  
 of  
 node

Paramete

- **fil**
  
 Fil-  
 ters  
 to  
 ap-  
 ply.  
 De-  
 fault  
 to  
 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 provision field before this interval in seconds

- **limit\_max\_items**
 Maximum number of nodes to return

- **mark\_previous\_page**
 If the last item of the previous page we return the next result set.

- **sort\_tribe**
 Attribute

by  
 which  
 re-  
 sults  
 shou  
 be  
 sorte

- - sort**  
 di-  
 rec-  
 tion  
 in  
 which  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc  
 desc

**get\_node**  
 Get  
 node  
 tags  
 base  
 on  
 its  
 id.

**Parameters**  
**node\_id**  
 The  
 id  
 of  
 a  
 node

**Returns**  
 A  
 list  
 of  
 Node  
 Tag  
 ob-  
 jects

**Raises**  
 Node  
 Not-  
 Found  
 if

the  
node  
is  
not  
found

**get\_node**  
Get  
node  
traits  
base  
on  
its  
id.

**Parameter**  
**node\_id**  
The  
id  
of  
a  
node

**Returns**  
A  
list  
of  
Node  
Trait  
ob-  
jects

**Raises**  
Node  
Not-  
Found  
if  
the  
node  
is  
not  
found

**get\_node**  
Get  
spe-  
cific  
colu  
for  
mat  
ing  
node

ters.

Re-  
 turn  
 a  
 list  
 of  
 the  
 spec  
 i-  
 fied  
 colu  
 for  
 all  
 node  
 that  
 matc  
 the  
 spec  
 i-  
 fied  
 fil-

Parame

- **col**  
 List  
 of  
 col-  
 umn  
 nam  
 to  
 re-  
 turn  
 De-  
 fault  
 to  
 id  
 col-  
 umn  
 whe  
 colu  
 ==  
 Non
- **fil**  
 Fil-  
 ters  
 to  
 ap-  
 ply.

De-  
 fault  
 to  
 Non

**associ**  
 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  
 be-  
 fore  
 this  
 in-  
 ter-  
 val  
 in  
 sec-  
 onds

- **limit**  
 Max  
 i-  
 mum  
 num  
 ber  
 of  
 node  
 to  
 re-  
 turn

- **mark**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults



shou
 be
 sorte

- **sort**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc  
 desc

**Returns**

A  
 list  
 of  
 tu-  
 ples  
 of  
 the  
 spec  
 i-  
 fied  
 colu

**get\_not**

Re-  
 turn  
 ob-  
 jects  
 with  
 ver-  
 sion  
 that  
 are  
 not  
 the  
 spec  
 i-  
 fied  
 ver-  
 sion

This  
 re-  
 turn

null versions (there shouldnt be any) are also returned.

ob-  
jects  
with  
ver-  
sion  
that  
are  
not  
the  
spec  
i-  
fied  
ver-  
sion  
Ob-  
jects  
with

Paramete

- **mod**  
the  
nam  
of  
the  
mod  
(clas  
of  
de-  
sired  
ob-  
jects
- **ver**  
list  
of  
ver-  
sion  
of  
ob-  
jects  
not  
to  
be  
re-  
turn

Returns  
list

of  
 the  
 DB  
 ob-  
 jects

**Raises**

Iron  
 icEx  
 cep-  
 tion  
 if  
 there  
 is  
 no  
 class  
 as-  
 so-  
 ci-  
 ated  
 with  
 the  
 nam

**get\_off**

Get  
 a  
 list  
 con-  
 duc-  
 tors  
 that  
 are  
 of-  
 fline  
 (dea

**Parame**

**fie**  
 A  
 field  
 to  
 re-  
 turn  
 host  
 nam  
 by  
 de-  
 fault

**Returns**

A  
 list

of  
 re-  
 ques  
 field  
 of  
 of-  
 fline  
 con-  
 duc-  
 tors.

**get\_onl**  
 Get  
 a  
 list  
 con-  
 duc-  
 tor  
 host  
 nam  
 that  
 are  
 on-  
 line  
 and  
 ac-  
 tive.

**Returns**  
 A  
 list  
 of  
 con-  
 duc-  
 tor  
 host  
 nam

**get\_por**  
 Re-  
 turn  
 a  
 net-  
 worl  
 port  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parame**  
**add**

The  
 MAC  
 ad-  
 dres  
 of  
 a  
 port

Returns

A  
 port

get\_port

Re-  
 turn  
 a  
 net-  
 worl  
 port  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

Paramete

port  
 The  
 id  
 of  
 a  
 port

Returns

A  
 port

get\_port

Re-  
 turn  
 a  
 net-  
 worl  
 port  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

Paramete

port  
 The  
 nam

of  
 a  
 port

**Returns**  
 A  
 port

**get\_port**  
 Re-  
 turn  
 a  
 net-  
 work  
 port  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parameter**  
**port**  
 The  
 uuid  
 of  
 a  
 port

**Returns**  
 A  
 port

**get\_ports**  
 Re-  
 turn  
 a  
 list  
 of  
 port

**Parameter**

- **limit**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 port  
 to

re-  
turn

- **max**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**get\_pos**  
 Re-  
 turn

a  
 net-  
 worl  
 port  
 grou  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parame**

**add**  
 The  
 MA  
 ad-  
 dres  
 of  
 a  
 port  
 grou

**Returns**

A  
 port  
 grou

**Raises**

Port  
 grou  
 Not-  
 Foun

**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\_por**

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-  
 Four

**get\_por**

Re-  
 turn  
 a  
 net-

world  
 port  
 grou  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parame**

**por**  
 The  
 uuid  
 of  
 a  
 port  
 grou

**Returns**

A  
 port  
 grou

**Raises**

Port  
 grou  
 Not-  
 Four

**get\_por**

Re-  
 turn  
 a  
 list  
 of  
 port  
 grou

**Parame**

•  
**lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 port  
 grou  
 to  
 re-  
 turn

- max**  
 The  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- sort**  
 Di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc  
 desc

**Returns**  
 A  
 list  
 of  
 port  
 grou

get\_por

List  
 all  
 the  
 port  
 grou  
 for  
 a  
 give  
 node

Paramet

- **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  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 Di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**Returns**

A  
 list  
 of  
 port  
 grou

**get\_port**

List  
 all  
 the  
 port  
 for  
 a  
 give  
 node

**Parame**

-

**node**  
 The  
 in-  
 te-  
 ger  
 node  
 ID.

- **limit**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 port  
 to  
 re-  
 turn

- **mark**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
di-  
rec-  
tion  
in  
whic  
re-  
sults  
shou  
be  
sorte  
(asc.  
desc

**Returns**  
A  
list  
of  
port

**get\_por**

List  
all  
the  
port  
for  
a  
give  
port  
grou

**Parame**

- **por**  
The  
in-  
te-  
ger  
port  
grou  
ID.
- **lim**  
Max  
i-  
mun  
num  
ber

of  
 port  
 to  
 re-  
 turn

- **max**  
 The  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 Di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc



**Returns**

A list of ports.

**get\_vol**

Return a volume connector representation.

**Parameter**

**db\_id**  
 The integer data ID of a volume connector.

**Returns**

A volume connector with the specified ID.

**Raises**

Vol-

is not found.

ume  
 Con  
 nec-  
 torN  
 Four  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 spec  
 i-  
 fied  
 ID

**get\_vol**  
 Re-  
 turn  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parame**  
**con**  
 The  
 UUI  
 of  
 a  
 con-  
 nec-  
 tor.

**Returns**  
 A  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with

is not found.

the  
 spec  
 i-  
 fied  
 UUI

**Raises**  
 Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 spec  
 i-  
 fied  
 UUI

**get\_vol**

Re-  
 turn  
 a  
 list  
 of  
 vol-  
 ume  
 con-  
 nec-  
 tors.

**Parame**

•  
**lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 vol-

ume  
 con-  
 nec-  
 tors  
 to  
 re-  
 turn

- **mark**  
 The  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 Di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc

desc

**Returns**

A list of volume connectors.

**Raises**

InvalidParameterError: If sort\_does\_not\_exist.

**get\_volumes**

List all the volumes connectors for a given node.

**Parameters**

- **node\_id**: The integer node ID.

- **limit**  
 Maximum number of volume connectors to return
- **marker**  
 The last item of the previous page we return the next result set.
- **sort**  
 Attribute by which results should be sorted
- **sort\_order**  
 Direction of sorting

tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**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**

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-  
 ume  
 tar-  
 get.

Returns

A  
 vol-  
 ume  
 tar-  
 get.

Raises

Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 no  
 vol-  
 ume  
 tar-  
 get  
 with  
 this  
 ID  
 ex-  
 ists.

get\_vol

Re-  
 turn  
 a  
 vol-  
 ume  
 tar-  
 get  
 rep-  
 re-  
 sen-  
 ta-  
 tion.



**Parameters**

**uuifile**  
 The  
 UUI  
 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  
 UUI  
 ex-  
 ists.

**get\_volumes**

Re-  
 turn  
 a  
 list  
 of  
 vol-  
 ume  
 tar-  
 gets

**Parameters**

- limit**  
 Max

i-  
 mun  
 num  
 ber  
 of  
 vol-  
 ume  
 tar-  
 gets  
 to  
 re-  
 turn

- **mar**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **son**  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **son**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-

sults  
 shou  
 be  
 sorte  
 (asc  
 desc

**Returns**

A  
 list  
 of  
 vol-  
 ume  
 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.

- **limit**  
Max  
i-  
mun  
num  
ber  
of  
vol-  
ume  
tar-  
gets  
to  
re-  
turn

- **max**  
the  
last  
item  
of  
the  
pre-  
vi-  
ous  
page  
we  
re-  
turn  
the  
next  
re-  
sult  
set.

- **sort**  
At-  
tribu  
by  
whic  
re-  
sults  
shou  
be  
sorte

- **sort**

di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**Returns**

A  
 list  
 of  
 vol-  
 ume  
 tar-  
 gets

**Raises**

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.

Parameters

- **volume**  
The volume of the volume.
- **limit**  
Maximum number of volume targets to return.
- **marker**  
The last item of the previous page we return the next result set.
- **sortBy**  
Attribute 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-
 terec
 hard

ware  
in-  
ter-  
face  
for  
a  
con-  
duc-  
tor.

**Parameters**

**conductors**  
Data  
ID  
of  
con-  
duc-  
tor.

**Returns**

List  
of  
Con-  
ob-  
jects

**list\_hardware**

List  
reg-  
is-  
tered  
hard-  
ware  
in-  
ter-  
face  
for  
give  
hard-  
ware  
type

This  
is  
re-  
stric-  
to  
only  
ac-  
tive  
con-  
duc-  
tors.



filter by. :returns: list of `ConductorHardwareInterfaces` objects.

migrate

Try to migrate away from the iscsi de- ploy in- ter- face

Parameters

- **con** the ad- min con- text
- **max** The max i- mun num ber of ob- jects to mi- grate Mus be

all the objects will be migrated.

the beginning of this call) and 2. the number of migrated objects.

>=  
 0.  
 If  
 zero

Returns

A  
 2-  
 tuple  
 1.  
 the  
 to-  
 tal  
 num  
 ber  
 of  
 ob-  
 jects  
 that  
 need  
 to  
 be  
 mi-  
 grate  
 (at

node\_tag

Che  
 if  
 the  
 spec  
 i-  
 fied  
 tag  
 ex-  
 ist  
 on  
 the  
 node

Parameters

- 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

**node\_traits**  
Check  
if  
the  
spec-  
i-  
fied  
trait  
ex-  
ists  
on  
the  
node

**Parameters**

- **node**  
The  
id  
of  
a  
node
-

**traits**  
 A  
 trait  
 string

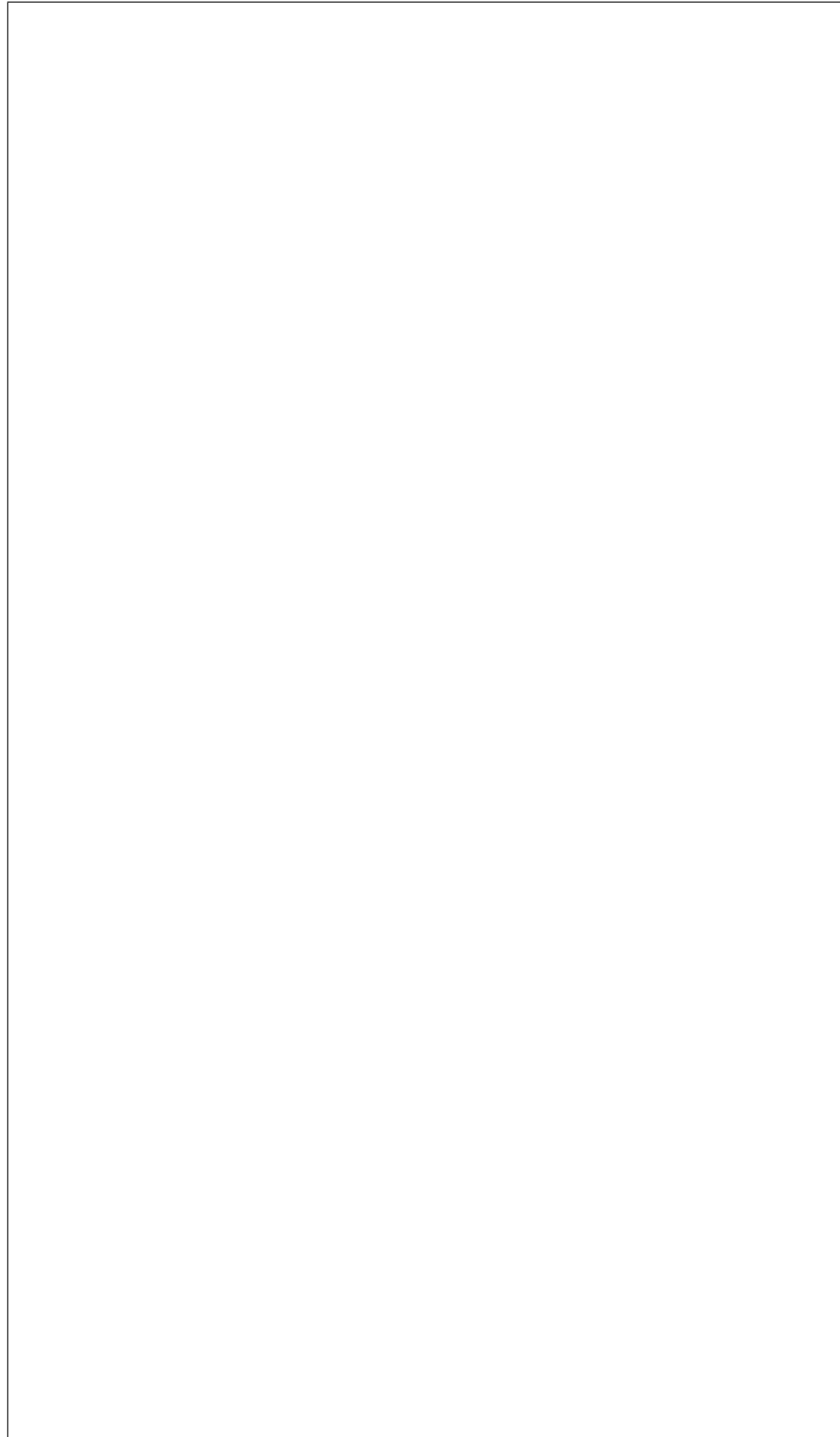
**Returns**  
 True  
 if  
 the  
 trait  
 exists  
 otherwise  
 False

**Raises**  
 NodeNotFound  
 if  
 the  
 node  
 is  
 not  
 found

**register\_cluster**  
 Register  
 a  
 cluster  
 as  
 an  
 active  
 conductor  
 with  
 the  
 cluster.

**Parameters**  
 •  
**value**  
 A  
 dict  
 of  
 values  
 which

mus  
con-  
tain  
the  
fol-  
low-  
ing:



(continues on next page)

(continued from previous page)

line 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  
 whe  
 a  
 con-  
 flict-  
 ing  
 on-

**Returns**  
 A  
 con-  
 duc-  
 tor.

**Raises**  
 Con  
 duc-  
 torA  
 read  
 is-  
 terec

**register**  
 Reg  
 is-  
 ters  
 hard  
 ware  
 in-  
 ter-  
 face  
 for  
 a  
 con-  
 duc-

tor.

**Parameters**

- conduct**  
 Data ID of conductor to register interface for.
- hardware**  
 Name of hardware type for the interface.
- interface**  
 Type of interface e.g. deployment or boot.
- interfaces**  
 List of interface names to

in the combination of all parameters is already registered.

reg-  
 is-  
 ter.  
 •  
**def**  
 Strin  
 the  
 de-  
 fault  
 in-  
 ter-  
 face  
 for  
 this  
 hard  
 ware  
 type  
 and  
 in-  
 ter-  
 face  
 type  
**Raises**  
 Con  
 duc-  
 torH  
 ware  
 ter-  
 face  
 sAl-  
 read  
 is-  
 terec  
 if  
 at  
 least  
 one  
 of  
 the  
 in-  
 ter-  
 face  
**release**  
 Re-  
 lease  
 the  
 rese  
 va-  
 tion



on  
a  
node

#### Parameters

- **tag**  
A  
string  
unique  
iden-  
ti-  
fy-  
ing  
the  
rese-  
va-  
tion  
hold
- **node**  
A  
node  
id  
or  
uuid

#### Raises

Nod  
Not-  
Foun  
if  
the  
node  
is  
not  
foun

#### Raises

Nod  
Loc  
if  
the  
node  
is  
re-  
serv  
by  
an-  
othe  
host

performed, mark it reserved by this host.

**Raises**  
 NodeError  
 Not-FoundError  
 LocalizedError  
 if the node was found to not have a reservation at all.

**reserved**  
 Re-serv a node  
 To prevent other Manager agents vice from manipulating the give Node while a Task is

**Parameters**  
 • tag  
 A

string  
unique  
identi-  
fiers  
fy-  
ing  
the  
rese-  
va-  
tion  
hold

- **node**  
A  
node  
id  
or  
uuid

#### Returns

A  
Node  
ob-  
ject.

#### Raises

Node  
Not-  
Found  
if  
the  
node  
is  
not  
found

#### Raises

Node  
Locked  
if  
the  
node  
is  
al-  
ready  
re-  
serv

#### set\_node

Re-  
place  
all  
of

the  
 node  
 tags  
 with  
 spec  
 i-  
 fied  
 list  
 of  
 tags  
 This  
 ig-  
 nore  
 du-  
 pli-  
 cate  
 tags  
 in  
 the  
 spec  
 i-  
 fied  
 list.

Paramete

- **node**  
 The  
 id  
 of  
 a  
 node
- **tag**  
 List  
 of  
 tags

Returns

A  
 list  
 of  
 Node  
 Tag  
 ob-  
 jects

Raises

Node  
 Not-

Four
 if
 the
 node
 is
 not
 found

**set\_nod**

Re-
 plac
 all
 of
 the
 node
 traits
 with
 spec
 i-
 fied
 list
 of
 traits

This
 ig-
 nore
 du-
 pli-
 cate
 traits
 in
 the
 spec
 i-
 fied
 list.

**Parame**

- **nod**  
The  
id  
of  
a  
node
- **tra**  
List  
of  
traits

limit.

• **version**  
 the  
 ver-  
 sion  
 of  
 the  
 ob-  
 ject.

**Returns**  
 A  
 list  
 of  
 Node  
 Trait  
 ob-  
 jects

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 set-  
 ting  
 the  
 trait  
 wou  
 ex-  
 ceed  
 the  
 per-  
 node  
 trait

**Raises**  
 Node  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 foun

**take\_ove**

thus guarding against races.

Do  
 a  
 take  
 over  
 for  
 an  
 al-  
 lo-  
 ca-  
 tion.  
 The  
 al-  
 lo-  
 ca-  
 tion  
 is  
 only  
 up-  
 date  
 if  
 the  
 old  
 con-  
 duc-  
 tor  
 mate  
 the  
 pro-  
 vide  
 valu

Parame

- **all**  
 Al-  
 lo-  
 ca-  
 tion  
 ID
- **old**  
 The  
 con-  
 duc-  
 tor  
 ID  
 we  
 ex-

cation.

pect  
 to  
 be  
 the  
 cur-  
 rent  
 con  
 of  
 the  
 al-  
 lo-

•

new  
 The  
 con-  
 duc-  
 tor  
 ID  
 of  
 the  
 new  
 con

Returns

True  
 if  
 the  
 take  
 over  
 was  
 suc-  
 cess  
 ful,  
 Fals  
 oth-  
 er-  
 wise

Raises

Al-  
 lo-  
 ca-  
 tion.  
 Not-  
 Four

touch\_c

Mar  
 a  
 con-  
 duc-



tor  
 as  
 ac-  
 tive  
 by  
 up-  
 dat-  
 ing  
 its  
 up-  
 date  
 prop  
 erty.

**Parameters**

**host**  
 The  
 host  
 name  
 of  
 this  
 con-  
 duc-  
 tor  
 ser-  
 vice

**Raises**

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  
 its  
 pro-  
 vi-  
 sion  
 prop  
 erty.

**Parame**  
**nod**  
 The  
 id  
 of  
 a  
 node

**Raises**  
 Nod  
 Not-  
 Four

**unregis**  
 Re-  
 mov  
 this  
 con-  
 duc-  
 tor  
 from  
 the  
 ser-  
 vice  
 reg-  
 istry  
 im-  
 me-  
 di-  
 ately

**Parame**  
**hos**  
 The  
 host  
 nam  
 of  
 this  
 con-  
 duc-

tor  
 ser-  
 vice

**Raises**  
 Con  
 duc-  
 torN  
 Foun

**unregis**  
 Un-  
 reg-  
 is-  
 ters  
 all  
 hard  
 ware  
 in-  
 ter-  
 face  
 for  
 a  
 con-  
 duc-  
 tor.

**Parame**  
**con**  
 Data  
 ID  
 of  
 con-  
 duc-  
 tor  
 to  
 un-  
 reg-  
 is-  
 ter  
 for.

**unset\_r**  
 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  
 foun  
**update\_**  
 Up-  
 date  
 prop  
 er-  
 ties  
 of  
 an  
 al-  
 lo-

ca-  
tion.

Parameters

- **all**  
Al-  
lo-  
ca-  
tion  
ID
- **val**  
Dict  
of  
val-  
ues  
to  
up-  
date
- **upd**  
If  
True  
and  
node  
is  
up-  
date  
up-  
date  
the  
node  
with  
in-  
stan-  
and  
trait  
from  
the

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

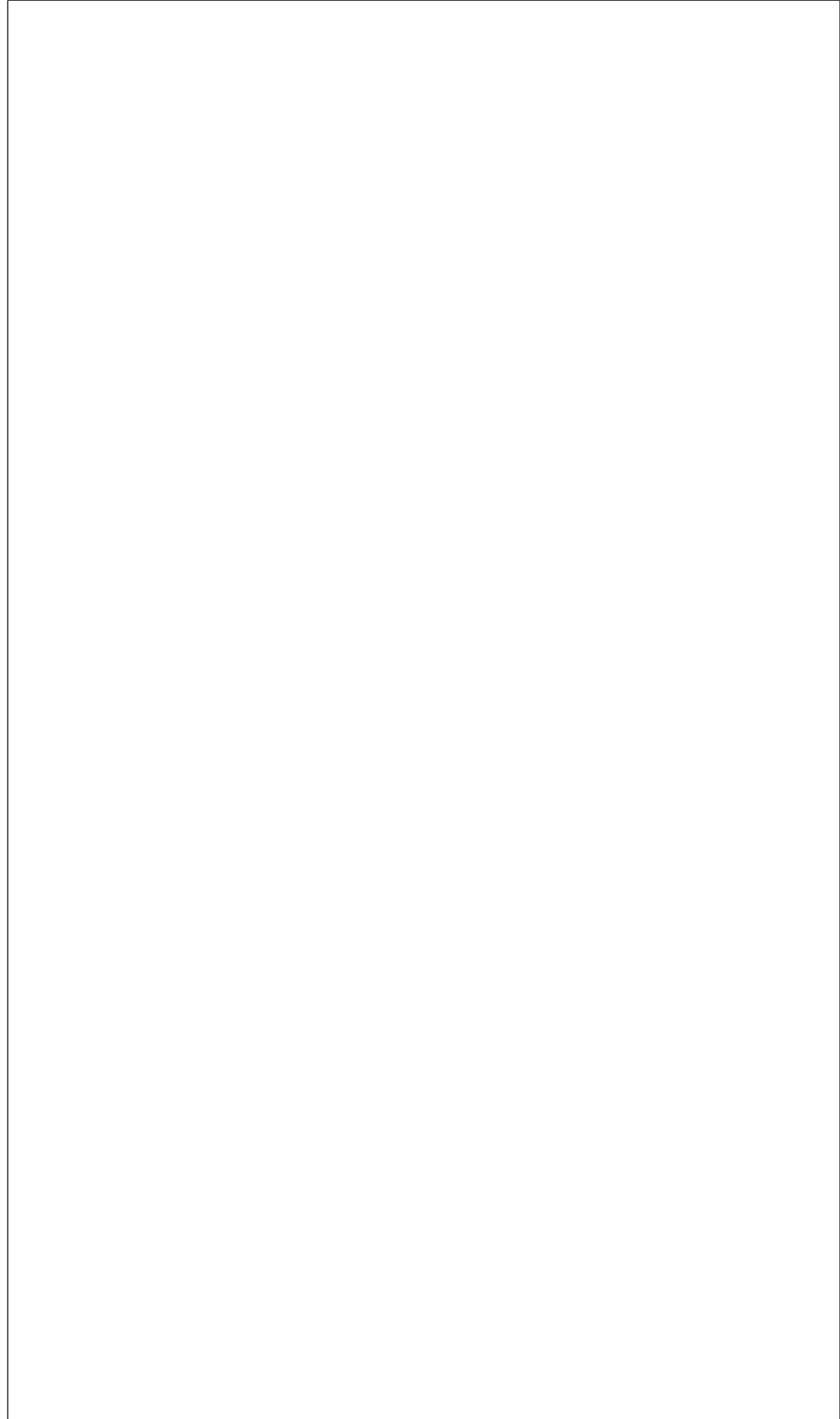
**Raises**  
Nod  
As-  
so-  
ci-  
ated

**update\_**  
Up-  
date  
a  
list  
of  
BIO  
Set-  
ting  
reco

**Parame**

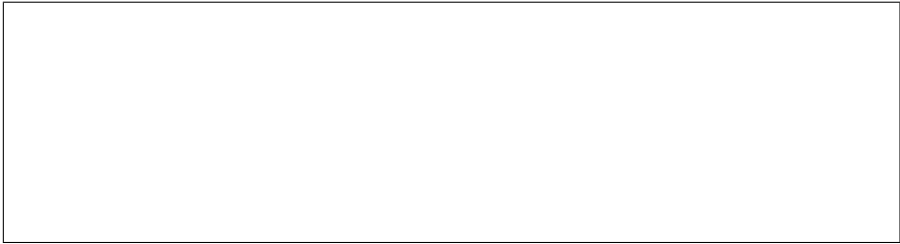
- **nod**  
The  
node  
id.
- **set**  
A  
list

of  
BIO  
Set-  
ting  
to  
be  
up-  
date



(continues on next page)

(continued from previous page)



- **version**  
the  
ver-  
sion  
of  
the  
ob-  
ject.

**Returns**  
A  
list  
of  
BIO  
Set-  
ting  
ob-  
jects

**Raises**  
Node  
Not-  
Found  
if  
the  
node  
is  
not  
found

**Raises**  
BIO  
Set-  
ting  
Not-  
Found  
if  
any  
of  
the  
set-  
tings  
is



not  
 found.  
**update\_**  
 Up-  
 date  
 prop-  
 er-  
 ties  
 of  
 an  
 chas-  
 sis.

**Parame**

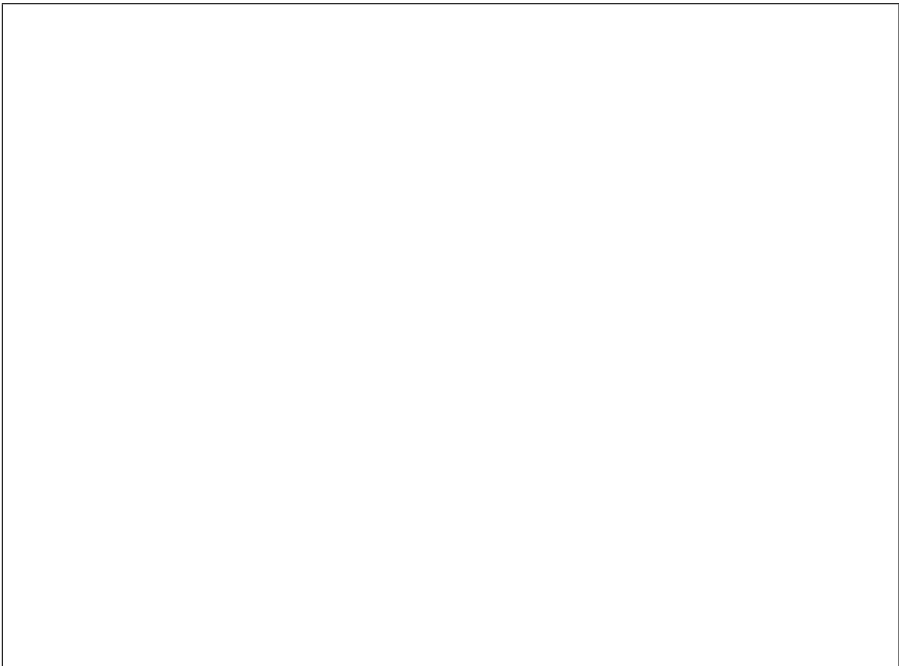
- cha**  
 The  
 id  
 or  
 the  
 uuid  
 of  
 a  
 chas-  
 sis.
- val**  
 Dict  
 of  
 val-  
 ues  
 to  
 up-  
 date

**Returns**  
 A  
 chas-  
 sis.

**update\_**  
 Up-  
 date  
 a  
 de-  
 ploy  
 men-  
 tem-  
 plate

**Parame**

- **templateId**  
ID of the deployment template to update
- **validation**  
A dictionary describing the deployment template. For example:



(continues on next page)

(continued from previous page)

--

**Raises**

De-  
 ploy  
 plat-  
 eDu  
 pli-  
 cate  
 Nam  
 if  
 a  
 de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 sam  
 nam  
 ex-  
 ists.

**Raises**

De-  
 ploy  
 plate  
 Four  
 if  
 the  
 de-  
 ploy  
 tem-  
 plate  
 does  
 not  
 ex-  
 ist.

**Returns**

A  
 de-  
 ploy  
 tem-  
 plate

**update\_**

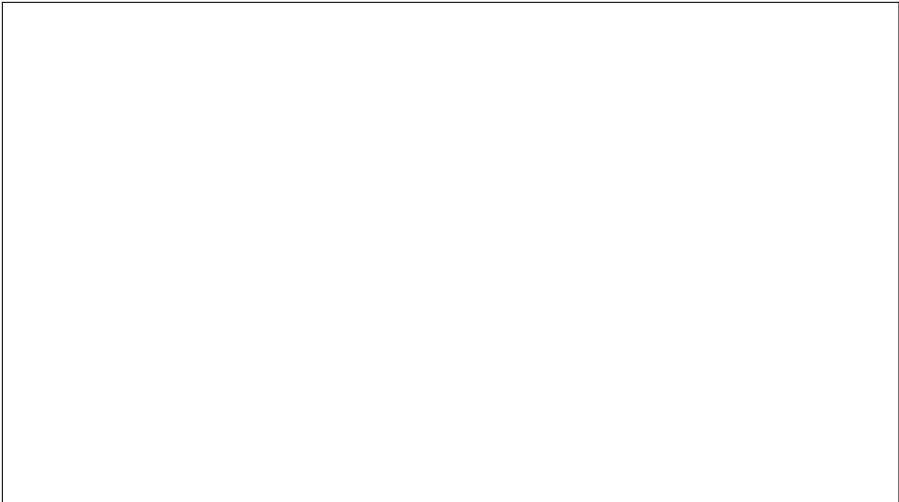
Up-  
 date  
 prop  
 er-  
 ties

of  
 a  
 node

Paramete

- **nod**
  
 The
   
 id
   
 or
   
 uuid
   
 of
   
 a
   
 node
- **val**
  
 Dict
   
 of
   
 val-
   
 ues
   
 to
   
 up-
   
 date
   
 May
   
 be
   
 a
   
 par-
   
 tial
   
 list,
   
 eg.
   
 whe
   
 set-
   
 ting
   
 the

properties for a driver. For example:



(continues on next page)

(continued from previous page)



**Returns**

A  
node

**Raises**

Node  
As-  
so-  
ci-  
ated

**Raises**

Node  
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-

address extra created\_at updated\_at

date  
 May  
 con-  
 tain  
 the  
 fol-  
 low-  
 ing  
 keys  
 uuid  
 nam  
 node

**Returns**  
 A  
 port  
 grou

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu

**Raises**  
 Port  
 grou  
 Not-  
 Foun

**Raises**  
 Port  
 grou  
 pli-  
 cate  
 Nam

**Raises**  
 Port  
 grou  
 MA  
 read  
 ists

**update\_**  
 Up-  
 date  
 ob-  
 jects  
 to

dates them to that version.

their  
 lat-  
 est  
 know  
 ver-  
 sion  
 This  
 scan  
 all  
 the  
 ta-  
 bles  
 and  
 for  
 ob-  
 jects  
 that  
 are  
 not  
 in  
 their  
 lat-  
 est  
 ver-  
 sion  
 up-

Parame

- **con**  
 the  
 ad-  
 min  
 con-  
 text
- **max**  
 The  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 ob-  
 jects  
 to  
 mi-



all the objects will be migrated.

the beginning of this call) and 2. the number of migrated objects.

grate  
Mus  
be  
>=  
0.  
If  
zero  
  
**Returns**  
A  
2-  
tuple  
1.  
the  
to-  
tal  
num  
ber  
of  
ob-  
jects  
that  
need  
to  
be  
mi-  
grate  
(at  
  
**update\_**  
Up-  
date  
prop  
er-  
ties  
of  
a  
vol-  
ume  
con-  
nec-  
tor.  
  
**Parame**  
  
•  
**ide**  
The  
UI  
or

in-  
 te-  
 ger  
 ID  
 of  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor.

- **con**  
 Dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 in-  
 for-  
 ma-  
 tion  
 about  
 con-  
 nec-  
 tor  
 to  
 up-  
 date

**Returns**  
 A  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**Raises**  
 Vol-  
 ume  
 Con-  
 nec-  
 torT  
 pe-  
 An-  
 dI-  
 dAl-  
 read  
 ists

exists with a matching type and connector\_id field.

does not exist.

If  
 an-  
 othe  
 con-  
 nec-  
 tor  
 al-  
 read

**Raises**  
 Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 spec  
 i-  
 fied  
 iden

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 Whe  
 a  
 UUI  
 is  
 in-  
 clud  
 in  
 con-  
 nec-  
 tor\_

**update\_**  
 Up-

date  
 in-  
 for-  
 ma-  
 tion  
 for  
 a  
 vol-  
 ume  
 tar-  
 get.

Paramete

- **ide**  
 The  
 UI  
 or  
 in-  
 te-  
 ger  
 ID  
 of  
 a  
 vol-  
 ume  
 tar-  
 get.
- **tar**  
 Dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 in-  
 for-  
 ma-  
 tion  
 about  
 vol-  
 ume  
 tar-  
 get  
 to  
 up-

date.

the same boot index and node ID.

## Returns

A  
vol-  
ume  
tar-  
get.

## Raises

Invalid Parameter Value if a UUID is included in target\_

## Raises

Volume  
get-  
Book-  
Al-  
read  
ists  
if  
a  
vol-  
ume  
tar-  
get  
al-  
read  
ex-  
ists  
with

## Raises

Vol-  
ume  
get-  
Not-  
Four

if  
 no  
 vol-  
 ume  
 tar-  
 get  
 with  
 this  
 iden-  
 ex-  
 ists.

ironic.

ironic.

ironic.  
 Add  
 an  
 iden-  
 tity  
 fil-  
 ter  
 to  
 a  
 quer-

Fil-  
 ters  
 re-  
 sults  
 by  
 ID,  
 if  
 sup-  
 plied  
 valu-  
 is  
 a  
 valid  
 in-  
 te-  
 ger.  
 Oth-  
 er-  
 wise  
 at-

tempts 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-  
plie  
valu  
is

attempts to filter results by identity.

a  
 valid  
 MA  
 ad-  
 dres  
 Oth-  
 er-  
 wise

Paramet

- **que**  
 Ini-  
 tial  
 quer  
 to  
 add  
 fil-  
 ter  
 to.
- **val**  
 Valu  
 for  
 fil-  
 ter-  
 ing  
 re-  
 sults  
 by.

Returns

Moc  
 i-  
 fied  
 quer

ironic.

ironic.

ironic.

ironic.

ironic.  
 Add  
 a



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.

The  
back  
end  
is  
this  
mod  
ule  
it-  
self.

ironic.

Que  
help  
for  
sim-  
pler  
ses-  
sion  
us-  
age.

**Paramet**

**ses**  
if  
pres  
the  
ses-  
sion  
to  
use

ironic.db.sqlalchemy.migration module

ironic.

Cre-  
ate  
data  
sche  
from  
mod  
els

de-  
scrip-  
tion.  
  
Can  
be  
used  
for  
ini-  
tial  
in-  
stal-  
la-  
tion  
in-  
stea  
of  
up-  
grad

ironic.  
Use  
for  
dow  
grad  
ing  
data

Paramet  
ver  
(st.  
De-  
sirec  
data  
ver-  
sion

ironic.  
  
Cre-  
ates  
tem-  
plate  
for  
mi-  
gra-  
tion.

Paramet  
  
•  
  
mes  
(st.

Text  
 that  
 will  
 be  
 used  
 for  
 mi-  
 gra-  
 tion  
 ti-  
 tle

- **aut**  
 (*bo*  
 If  
 True  
 -

gen-  
 er-  
 ates  
 diff  
 base  
 on  
 cur-  
 rent  
 data  
 state

ironic.  
 Stan  
 data  
 with  
 pro-  
 vide  
 re-  
 vi-  
 sion

Don  
 run  
 any  
 mi-  
 gra-  
 tion

**Paramet**  
**rev**  
 (*st*  
 Sho  
 matc  
 one

cent revision

from  
 repo  
 i-  
 tory  
 or  
 head  
 -  
  
 to  
 stan  
 data  
 with  
 mos  
 re-  
  
 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  
men

**candida**

**conduct**

**createc**

**extra**

**id**

**last\_er**

**name**

node\_id

owner

resource

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** `ironic.common.db.sqlalchemy`  
 Base  
 sql  
 ext  
 dec  
 api  
 Bas  
 In-  
 ter-  
 nal  
 ta-  
 ble  
 used  
 to  
 track  
 wha  
 is  
 load  
 on  
 each  
 con-  
 duc-

tor.

**conduct**

**created**

**default**

**hardware**

**id**

**interfa**

**interfa**

**updated**

**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

updated

version

class i  
   Base  
   osl  
   sql  
   mod  
   Tim  
   osl  
   sql  
   mod  
   Mod

as\_dict

metadat

version

class i  
   Base  
   sql  
   ext  
   dec  
   api  
   Bas  
  
   Rep  
   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

instance

instance

last\_err

lessee

maintenance

maintenance

management

name

network

network

owner

power\_info

power\_state

properties

protected

protected

provisioning

provisioning

raid\_controller

raid\_info

rescue\_

reserva

resourc

retireco

retireco

storage

target\_

target\_

target\_

updateco

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\_sma

local\_l

name

node\_id

physical

portgro

pxe\_ena

updatec

**uuid**

**version**

**class** i  
 Base  
 sql  
 ext  
 dec  
 api  
 Bas  
  
 Rep  
 re-  
 sent  
 a  
 grou  
 of  
 net-  
 worl  
 port  
 of  
 a  
 bare  
 meta  
 node

**address**

**created**

**extra**

**id**

**internal**

**mode**

**name**

**node\_id**

**property**

standa

updated

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

created

extra

id

node\_id

type

updated

**uuid**

**version**

**class** i

Base  
 sql  
 ext  
 dec  
 api  
 Bas

Rep  
 re-  
 sent  
 a  
 vol-  
 ume  
 tar-  
 get  
 of  
 a  
 bare  
 meta  
 node

**boot\_in**

**created**

**extra**

**id**

**node\_id**

**property**

**updated**

**uuid**

**version**

**volume\_**

Return the model class with the specified name.

**Returns** the class with the specified identifier name.

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  
 node  
 and  
 tag  
 pair  
 al-  
 read  
 ex-  
 ists,  
 this  
 shou  
 still  
 suc-

ceed

**Parameters**

- **node**  
The id of a node
- **tag**  
A tag string

**Returns**

the Node Tag object.

**Raises**

NodeNotFoundError if the node is not found

**abstract**

Add trait to the node  
  
If the node and trait pair already exists, this

shou  
 still  
 suc-  
 ceed

Parame

- **node**  
 The  
 id  
 of  
 a  
 node
- **traits**  
 A  
 trait  
 strin
- **version**  
 the  
 ver-  
 sion  
 of  
 the  
 ob-  
 ject.

Returns

the  
 Nod  
 Trai  
 ob-  
 ject.

Raises

In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 addi  
 the  
 trait  
 wou  
 ex-  
 ceed



the  
per-  
node  
trait  
limi

#### **Raises**

Nod  
Not-  
Four  
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  
then  
to  
UUI  
It  
fails

early if any identities cannot possible be used as names or UUIDs.

**Parameters**

**identity**

List of identities.

**Returns**

A mapping from request identities to node UUIDs.

**Raises**

NodeNotFound if some identities were not found or cannot be validated.
 NameError or UUIDError

**abstract**

Check the who data for incoming patterns.

not specified in *ironic.common.release\_mappings.RELEASE\_MAPPING*.

**Parameters**  
**ignore**  
 List  
 of  
 mod  
 nam  
 to  
 skip

**Returns**  
 A  
 Boo  
 True  
 if  
 all  
 the  
 ob-  
 jects  
 have  
 sup-  
 port  
 ver-  
 sion  
 Fals  
 oth-  
 er-  
 wise

**abstract**  
 Create  
 a  
 new  
 al-  
 lo-  
 ca-  
 tion.

**Parameter**  
**value**  
 Dictionary  
 of  
 val-  
 ues  
 to  
 cre-  
 ate  
 an  
 al-  
 lo-  
 ca-  
 tion  
 with

**Returns**  
 An  
 al-  
 lo-  
 ca-  
 tion

**Raises**  
 Al-  
 lo-  
 ca-  
 tion  
 pli-  
 cate  
 Name

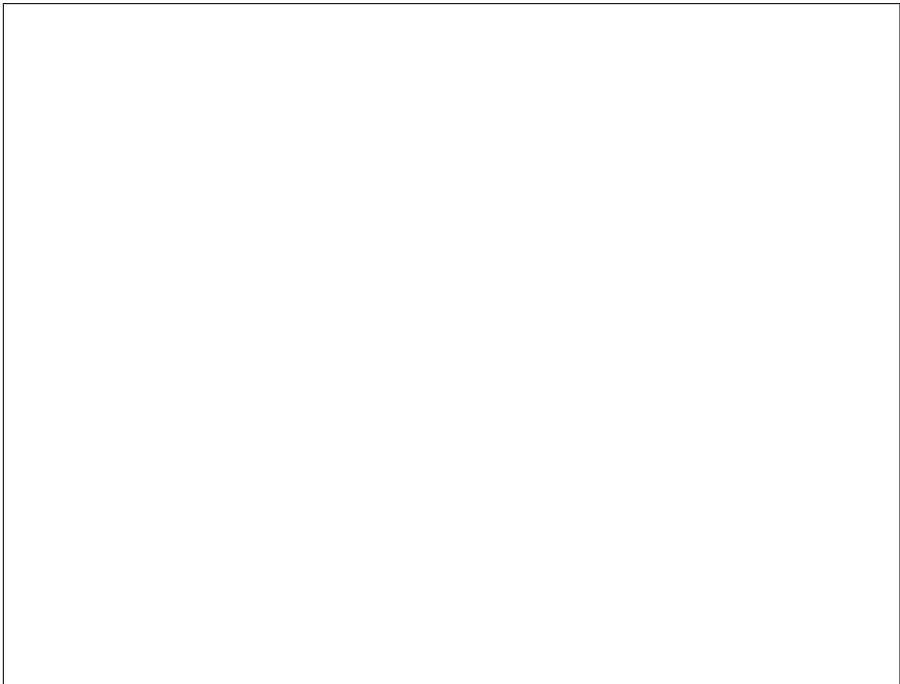
**Raises**  
 Al-  
 lo-  
 ca-  
 tion  
 Al-  
 read  
 ists

**abstract**  
 Cre-  
 ate

a  
 list  
 of  
 BIO  
 Set-  
 ting  
 reco  
 for  
 a  
 give  
 node

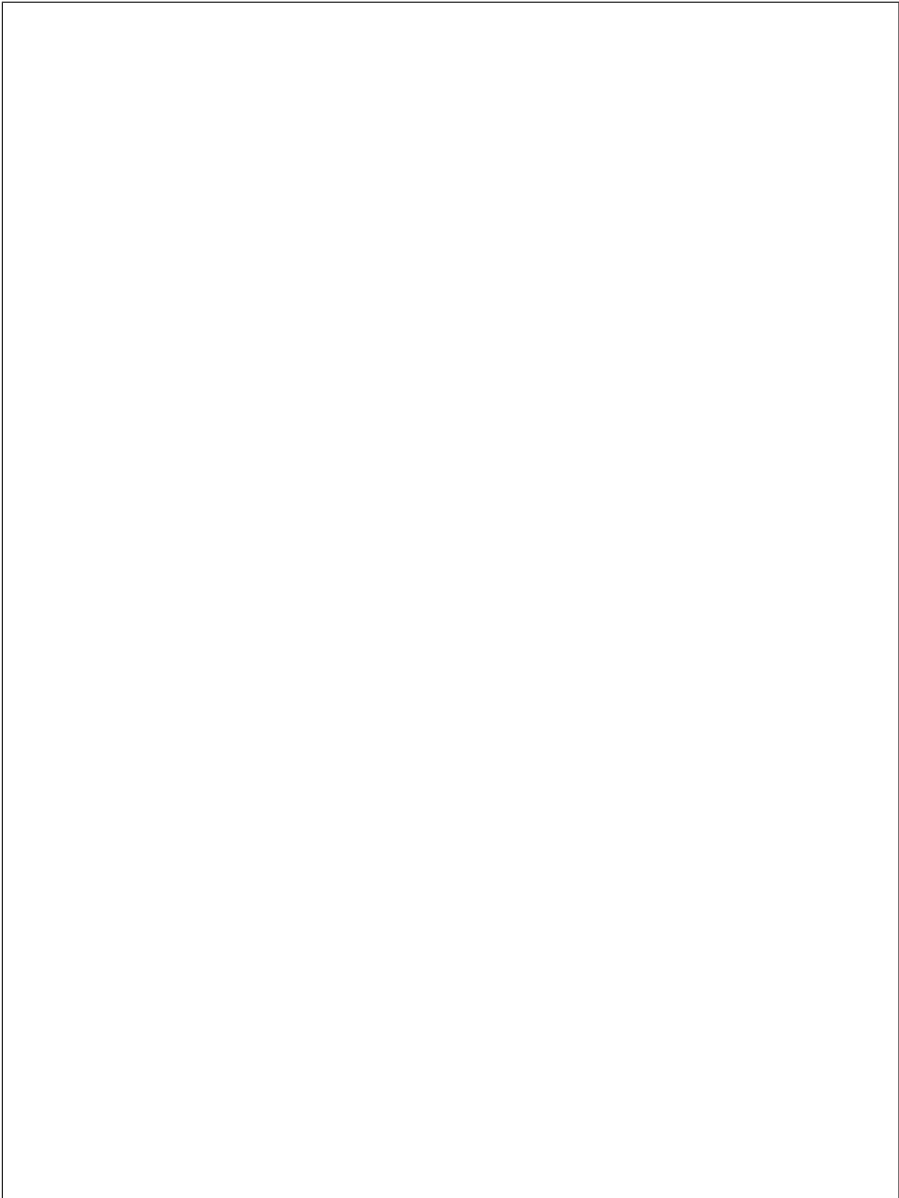
Paramet

- **node**  
 The  
 node  
 id.
- **set**  
 A  
 list  
 of  
 BIO  
 Set-  
 tings  
 to  
 be  
 cre-  
 ated



(continues on next page)

(continued from previous page)



•

**version**  
 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  
Set-  
tin-  
gAl-  
read  
ists  
if  
any  
of  
the  
set-  
ting  
reco  
al-  
read  
ex-  
ists.

**abstract**

Cre-  
ate  
a  
new  
chas  
sis.

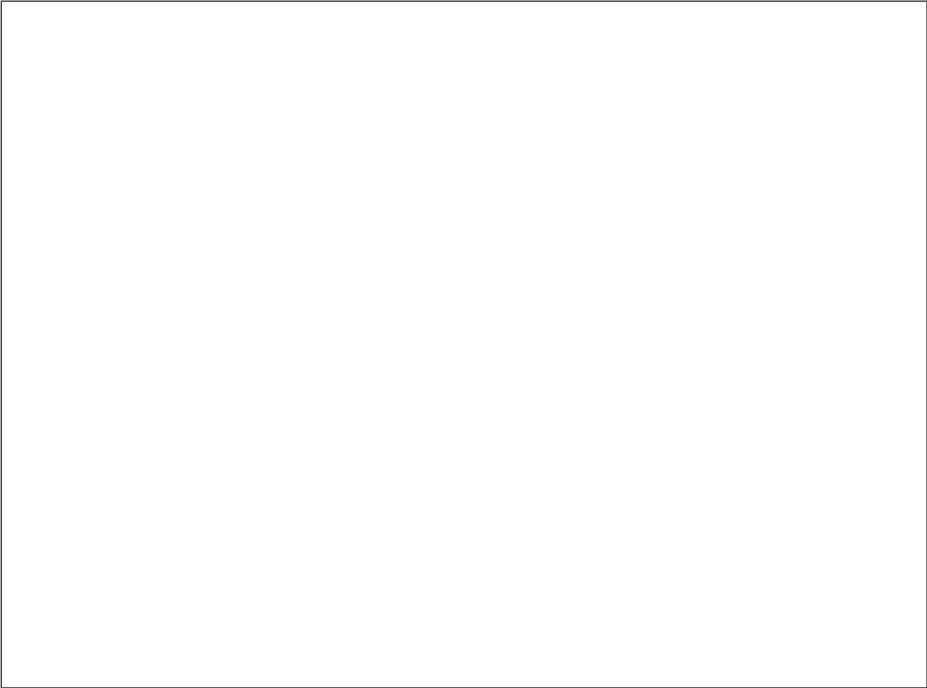
**Parameter**

**val**  
Dict  
of  
val-  
ues.

**abstract**

Cre-  
ate  
a  
de-  
ploy

men  
 tem-  
 plate  
**Parameter**  
**value**  
 A  
 dict  
 de-  
 scrib  
 ing  
 the  
 de-  
 ploy  
 men  
 tem-  
 plate  
 For  
 ex-  
 am-  
 ple:



**Raises**  
 De-  
 ploy  
 plat-  
 eDu  
 pli-  
 cate  
 Nam  
 if  
 a



de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 same  
 nam  
 ex-  
 ists.

**Raises**

De-  
 ploy  
 plate  
 read  
 ists  
 if  
 a  
 de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 same  
 UUI  
 ex-  
 ists.

**Returns**

A  
 de-  
 ploy  
 tem-  
 plate

**abstract**

Cre-  
 ate  
 a  
 new  
 node

**Parameter**

**val**  
 A  
 dict  
 con-  
 tain-  
 ing  
 sev-  
 eral

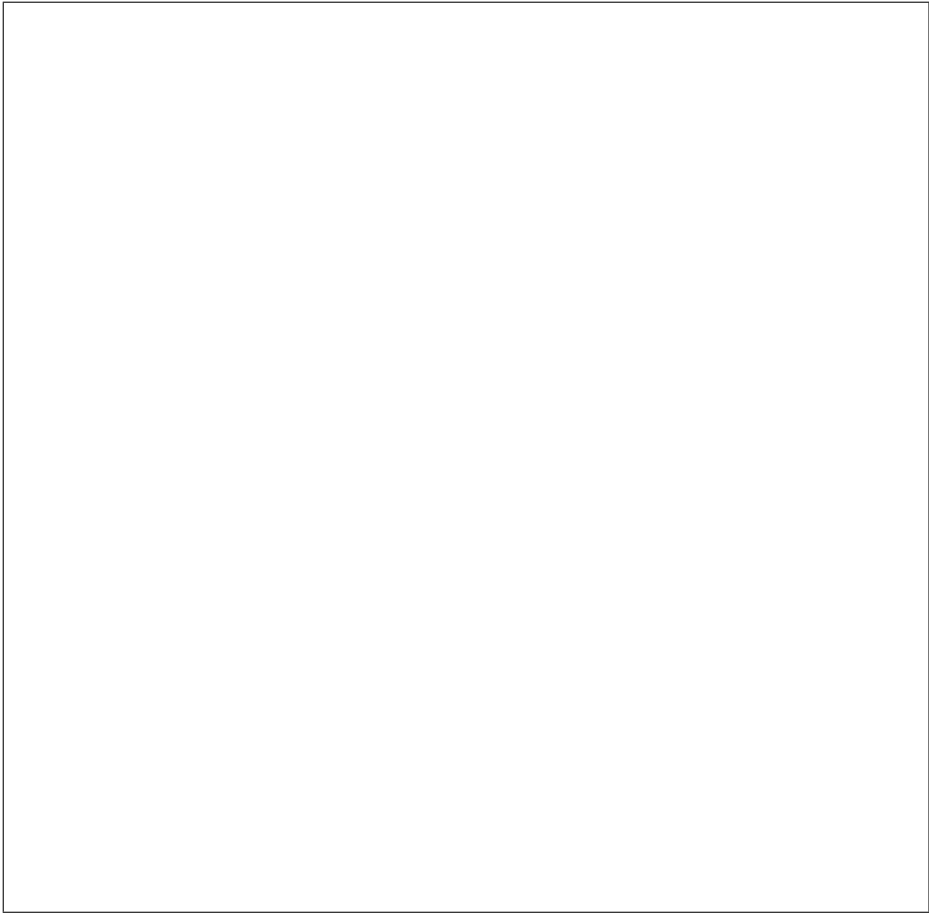
item  
used  
to  
iden  
tify  
and  
track  
the  
node  
and  
sev-

eral dicts which are passed into the Drivers when managing this node. For example:



(continues on next page)

(continued from previous page)



**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 val-  
 ues  
 con-  
 tains  
 tags  
 or  
 trait

**Returns**

A  
 node

**abstract**

Cre-  
 ate  
 a

created\_at updated\_at

new  
 port

**Parameters**  
**value**  
 Dict  
 of  
 val-  
 ues.

**abstract**  
 Cre-  
 ate  
 a  
 new  
 port  
 grou

**Parameters**  
**value**  
 Dict  
 of  
 val-  
 ues  
 with  
 the  
 fol-  
 low-  
 ing  
 keys  
 id  
 uuid  
 nam  
 node  
 ad-  
 dres  
 ex-  
 tra

**Returns**  
 A  
 port  
 grou

**Raises**  
 Port  
 grou  
 pli-  
 cate  
 Nam

**Raises**  
 Port

grou
 MA
 read
 ists

**Raises**

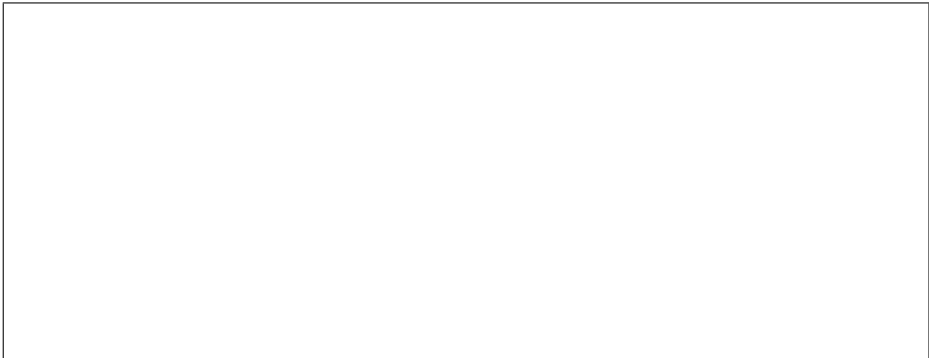
Port
 grou
 read
 ists

**abstract**

Cre-
 ate
 a
 new
 vol-
 ume
 con-
 nec-
 tor.

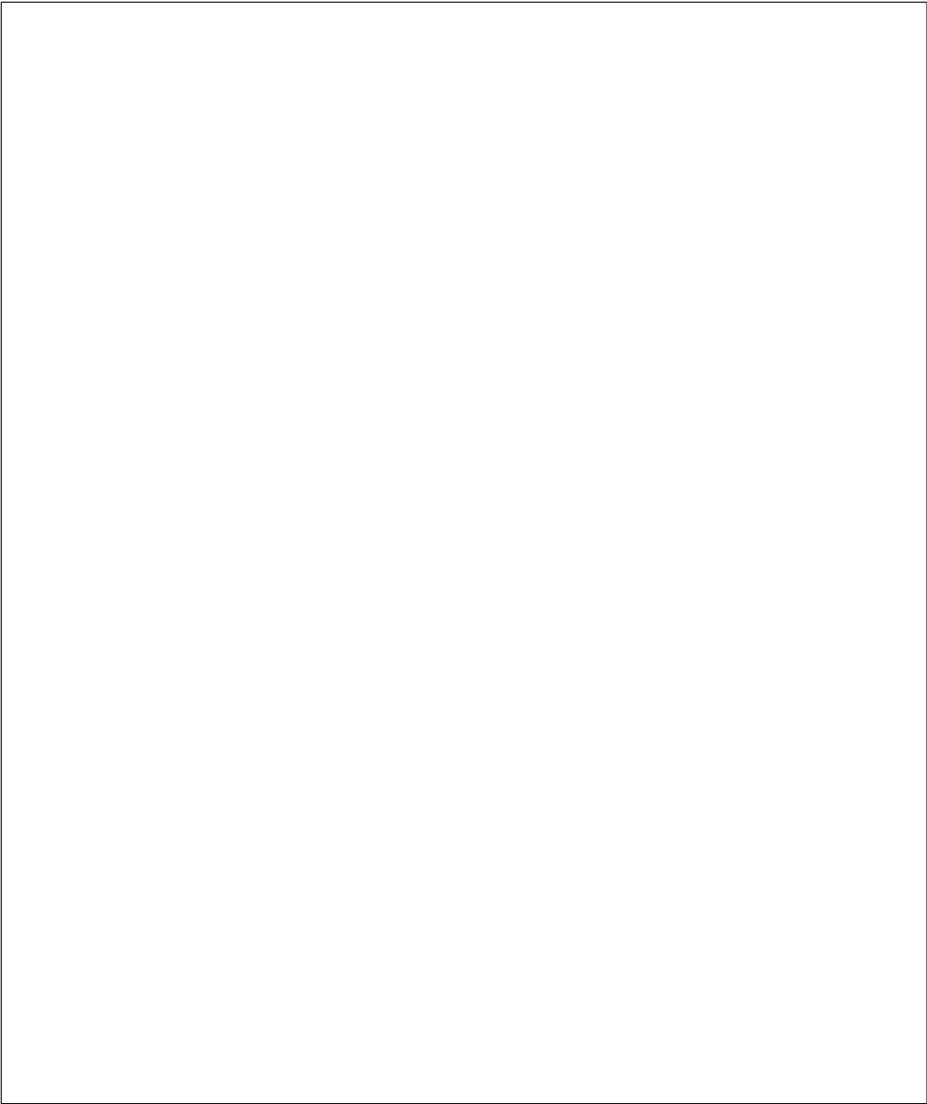
**Parame**

**con**
 Dic-
 tio-
 nary
 con-
 tain-
 ing
 in-
 for-
 ma-
 tion
 abou
 the
 con-
 nec-
 tor.
 Ex-
 am-
 ple:



(continues on next page)

(continued from previous page)



**Returns**

A  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**Raises**

Vol-  
 ume  
 Con  
 nec-  
 torT  
 pe-  
 An-  
 dI-  
 dAl-  
 read

ists with a matching type and connector\_id.

ready exists.

ists  
 If  
 a  
 con-  
 nec-  
 tor  
 al-  
 read  
 ex-

Raises

Vol-  
 ume  
 Con  
 nec-  
 torA  
 read  
 ists  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 sam  
 UUI  
 al-

abstract

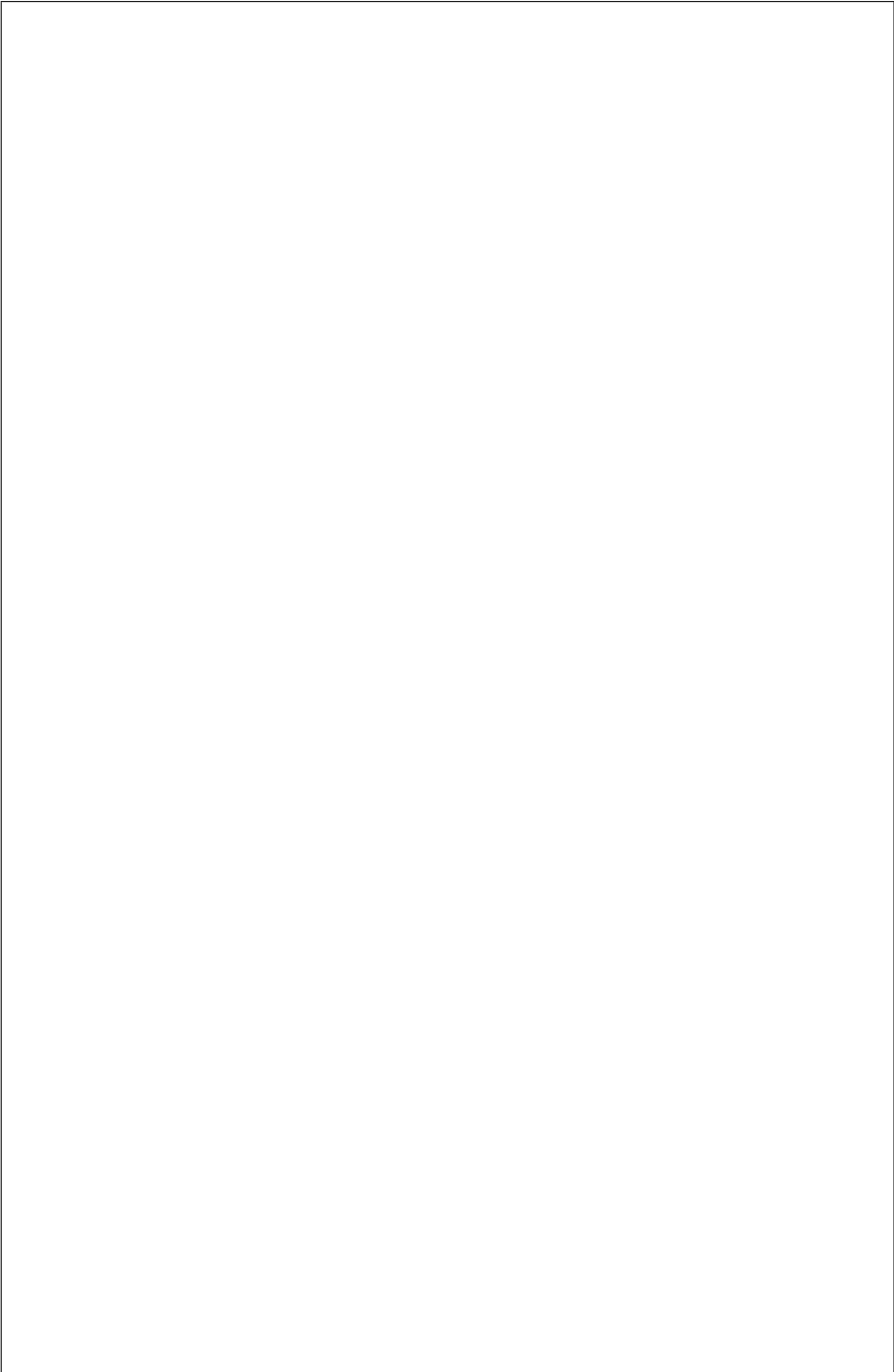
Cre-  
 ate  
 a  
 new  
 vol-  
 ume  
 tar-  
 get.

Paramete

tar  
 Dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 in-

for-  
 ma-  
 tion  
 about  
 the  
 vol-  
 ume  
 tar-  
 get.  
 Ex-

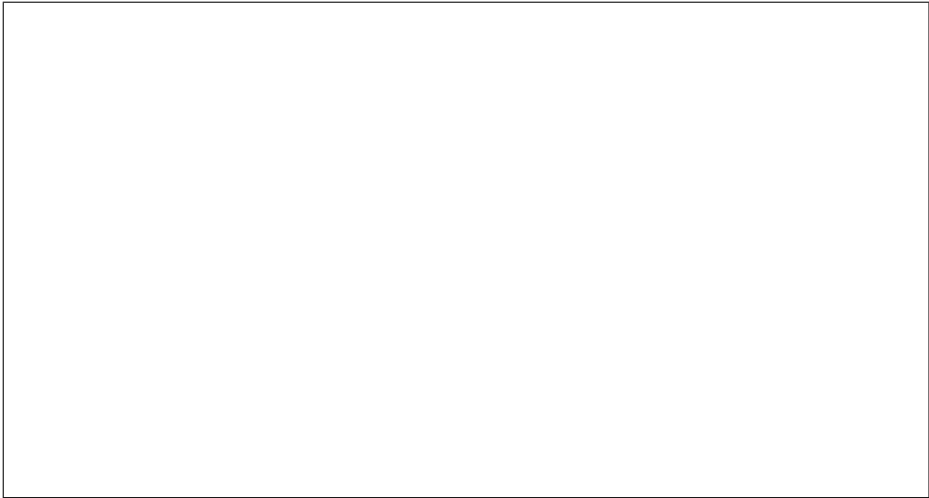
ample:



(continues on next page)



(continued from previous page)



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-  
get  
al-  
read  
ex-  
ists  
with

**Raises**

Vol-  
ume  
ge-  
tAl-  
read  
ists  
if

a  
 vol-  
 ume  
 tar-  
 get  
 with  
 the  
 sam  
 UUI  
 ex-  
 ists.

**abstract**

Dele  
 a  
 list  
 of  
 BIO  
 set-  
 tings

**Parame**

- **node**  
The  
node  
id.
- **name**  
List  
of  
BIO  
set-  
ting  
nam  
to  
be  
dele

**Raises**

Nod  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 foun

**Raises**

BIO
 Set-
 ting-
 Not-
 Foun
 if
 any
 of
 BIO
 set-
 ting
 nam
 is
 not
 foun

**abstract**

Dele
 spec
 i-
 fied
 tag
 from
 the
 node

**Parameters**

- **node**  
 The  
 id  
 of  
 a  
 node
- **tag**  
 A  
 tag  
 strin

**Raises**

Nod
 Not-
 Foun
 if
 the
 node
 is
 not
 foun

**Raises**

Node  
 Tag-  
 Not-  
 Four  
 if  
 the  
 tag  
 is  
 not  
 found

**abstract**

Delete  
 spec  
 i-  
 fied  
 trait  
 from  
 the  
 node

**Parameter**

- **node**  
 The  
 id  
 of  
 a  
 node
- **trait**  
 A  
 trait  
 string

**Raises**

Node  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 found

**Raises**

Node  
 Trait  
 Not-

Four
 if
 the
 trait
 is
 not
 found

**abstract**
 De-
 stroy
 an
 al-
 lo-
 ca-
 tion.

**Parameter**
**all**
 Al-
 lo-
 ca-
 tion
 ID

**Raises**
 Al-
 lo-
 ca-
 tion-
 Not-
 Four

**abstract**
 De-
 stroy
 a
 chas-
 sis.

**Parameter**
**cha**
 The
 id
 or
 the
 uuid
 of
 a
 chas-
 sis.

**abstract**
 De-

stroy  
 a  
 de-  
 ploy  
 men  
 tem-  
 plate

**Parameters**

**template\_id**  
 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-  
 ist.

**abstract**

De-  
 stroy  
 a  
 node  
 and  
 its  
 as-  
 so-  
 ci-  
 ated  
 re-  
 sour  
 De-

nectors, and volume targets.

stroy  
 a  
 node  
 in-  
 clud  
 ing  
 any  
 as-  
 so-  
 ci-  
 ated  
 port  
 port  
 grou  
 tags  
 trait  
 vol-  
 ume  
 con-

**Parame**  
**nod**  
 The  
 ID  
 or  
 UUI  
 of  
 a  
 node

**abstract**  
 De-  
 stroy  
 an  
 port

**Parame**  
**por**  
 The  
 id  
 or  
 MA  
 of  
 a  
 port

**abstract**  
 De-  
 stroy  
 a  
 port  
 grou

**Parameters**

**port**  
 The  
 UUI  
 or  
 MA  
 of  
 a  
 port  
 grou

**Raises**

Port  
 grou  
 Notl

**Raises**

Port  
 grou  
 Not-  
 Foun

**abstract**

De-  
 stroy  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**Parameters**

**ide**  
 The  
 UUI  
 or  
 in-  
 te-  
 ger  
 ID  
 of  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**Raises**

Vol-  
 ume  
 Con  
 nec-



does not exist.

torN  
 Four  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 spec  
 i-  
 fied  
 iden  
  
**abstrac**  
 De-  
 stroy  
 a  
 vol-  
 ume  
 tar-  
 get.  
  
**Parame**  
**ide**  
 The  
 UUI  
 or  
 in-  
 te-  
 ger  
 ID  
 of  
 a  
 vol-  
 ume  
 tar-  
 get.  
  
**Raises**  
 Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 a  
 vol-  
 ume  
 tar-

exist.

get  
with  
the  
spec  
i-  
fied  
iden  
does  
not

abstract

Re-  
triev  
hard  
ware  
type  
for  
the  
reg-  
is-  
tere  
and  
ac-  
tive  
con-  
duc-  
tors.

Parameters

use  
When  
to  
fac-  
tor  
con-  
duc-  
tor\_  
into  
the  
keys

Returns

A  
dict  
whic  
map  
hard  
ware  
type  
nam

to  
 the  
 set  
 of  
 host  
 which  
 sup-  
 port  
 them  
 For  
 ex-  
 am-

ple:



**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

**abstract**

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.

**Parameter**

**all**  
 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-  
tions

Paramete

- **filters**  
 Fil-  
ters  
to  
ap-  
ply.  
De-  
fault  
to  
Non

**node\_id**  
 uuid  
 of  
 node

**state**  
 al-  
lo-  
ca-  
tion  
 state

**resource**  
 re-  
ques-  
re-  
sour-  
class

- **limit**  
 Max  
i-  
mun-  
num-  
ber  
of  
al-  
lo-  
ca-  
tions  
to  
re-  
turn

•

**max**

The last item of the previous page we return the next result set.

•

**sort**

Attributed by which results should be sorted

•

**sort**

Direction in which results should be sorted (ascending or descending)

**Returns**

A list of allocated

tions

**abstract**

Re-  
triev  
BIO  
set-  
ting  
valu

**Parame**

- nod**  
The  
node  
id.

- nam**  
Strin  
con-  
tain-  
ing  
nam  
of  
BIO  
set-  
ting  
to  
be  
re-  
triev

**Returns**

The  
BIO  
Set-  
ting  
ob-  
ject.

**Raises**

Nod  
Not-  
Foun  
if  
the  
node  
is  
not  
foun

**Raises**



BIO
 Set-
 ting-
 Not-
 Foun
 if
 the
 BIO
 set-
 ting
 is
 not
 foun

**abstract**

Re-
 triev
 BIO
 set-
 ting-
 of
 a
 give
 node

**Parameters**

**node**
 The
 node
 id.

**Returns**

A
 list
 of
 BIO
 Set-
 ting
 ob-
 jects

**Raises**

Node
 Not-
 Foun
 if
 the
 node
 is
 not
 foun

**abstract**

Re-

turn  
 a  
 chas  
 sis  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parame**  
**cha**  
 The  
 id  
 of  
 a  
 chas  
 sis.

**Returns**  
 A  
 chas  
 sis.

**abstract**  
 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.

Param

- **lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 chas  
 sis  
 to  
 re-  
 turn
- **mar**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.
- **sor**  
 At-  
 tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be

sorte

- **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

Paramete

- **host**  
The  
host  
nam  
of  
the  
con-  
duc-  
tor  
ser-  
vice
- **only**  
Spec  
ify  
the  
fil-  
ter

online field is ignored if this value is set to None.

meet the specified online expectation.

valu  
on  
the  
*on-*  
*line*  
field  
whe  
quer  
ing  
con-  
duc-  
tors.  
The

**Returns**  
A  
con-  
duc-  
tor.

**Raises**  
Con  
duc-  
torN  
Four  
if  
the  
con-  
duc-  
tor  
with  
give  
host  
nam  
does  
not  
ex-  
ist  
or  
does

**abstract**  
  
Re-  
turn  
a  
list  
of  
con-  
duc-  
tors.

Parameters

- **limit**  
 Maximum number of conductors to return
- **marker**  
 the last item of the previous page we return the next result set.
- **sortBy**  
 Attribute by which results should be sorted
- **sortOrder**  
 direction

rec-  
tion  
in  
whic  
re-  
sults  
shou  
be  
sorte  
(asc  
desc

**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-  
ist.

**Returns**

A  
de-  
ploy  
tem-  
plate

**abstract**

Re-  
triev  
a  
de-  
ploy  
men  
tem-  
plate  
by  
nam

**Parameter**

**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-  
plate  
by  
UUI

**Parameters**

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-  
ist.

**Returns**

A  
de-  
ploy  
tem-  
plate

**abstract**

Re-  
 triev  
 a  
 list  
 of  
 de-  
 ploy  
 men  
 tem-  
 plate

Paramete

- **limit**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 de-  
 ploy  
 tem-  
 plate  
 to  
 re-  
 turn
- **max**  
 The  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.
- **sort**  
 At-

tribu  
by  
whic  
re-  
sults  
shou  
be  
sorte

- **sort**  
Di-  
rec-  
tion  
in  
whic  
re-  
sults  
shou  
be  
sorte  
(asc.  
desc

**Returns**

A  
list  
of  
de-  
ploy  
tem-  
plate

**abstract**

Re-  
turn  
a  
list  
of  
de-  
ploy  
men  
tem-  
plate  
with  
one  
of  
a  
list  
of  
nam

**Parame**

**name**  
 List  
 of  
 name  
 to  
 fil-  
 ter  
 by.

**Returns**  
 A  
 list  
 of  
 de-  
 ploy  
 tem-  
 plate

**abstract**  
 Re-  
 turn  
 a  
 node

**Parameter**  
**node**  
 The  
 id  
 of  
 a  
 node

**Returns**  
 A  
 node

**abstract**  
 Re-  
 turn  
 a  
 node

**Parameter**  
**instance**  
 The  
 in-  
 stan-  
 uuid  
 to  
 sear  
 for.

**Returns**  
 A  
 node

**Raises**

In-  
stan-  
ceNo-  
Foun-  
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  
nam  
of  
a  
node

**Returns**

A  
node

**abstract**

Find  
a  
node  
by  
any  
mat

ing  
 port  
 ad-  
 dres

**Parameters**  
**add**  
 list  
 of  
 port  
 ad-  
 dres  
 (e.g.  
 MA

**Returns**  
 Nod  
 ob-  
 ject.

**Raises**  
 Nod  
 Not-  
 Four  
 if  
 none  
 or  
 sev-  
 eral  
 node  
 are  
 foun

**abstract**  
 Re-  
 turn  
 a  
 node

**Parameters**  
**nod**  
 The  
 uuid  
 of  
 a  
 node

**Returns**  
 A  
 node

**abstract**  
 Re-  
 turn

a  
list  
of  
node

Paramete

- **filters**  
 Fil-  
ters  
to  
ap-  
ply.  
De-  
fault  
to  
Non

**associat**  
 True  
|  
 Fals

**reserv**  
 True  
|  
 Fals

**mainte**  
 True  
|  
 Fals

**chassis**  
 uuid  
of  
chas  
sis

**driver**  
 driv  
nam

**provis**  
 pro-  
vi-  
sion  
state  
of  
node

**provis**  
 node  
with

pro-  
 vi-  
 sion  
 field  
 be-  
 fore  
 this  
 in-  
 ter-  
 val  
 in  
 sec-  
 onds

- **limit**  
 Max-  
 i-  
 mum  
 num-  
 ber  
 of  
 node  
 to  
 re-  
 turn
- **mark**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.
- **sort**  
 At-  
 tribu-  
 by  
 whic



re-  
 sults  
 shou  
 be  
 sorte

•  
**son**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc  
 desc

**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-  
 Four  
 if  
 the  
 node

is  
not  
foun

**abstract**  
Get  
node  
trait  
base  
on  
its  
id.

**Parameter**  
**node**  
The  
id  
of  
a  
node

**Returns**  
A  
list  
of  
Node  
Trait  
ob-  
jects

**Raises**  
Node  
Not-  
Foun  
if  
the  
node  
is  
not  
foun

**abstract**  
Get  
spe-  
cific  
colu  
for  
mat  
ing  
node  
Re-  
turn

ters.

Param
 eters

- **columns**  
 List of column names to return. Default to id column when columns == None.
- **filters**  
 Filters to apply. Default

to  
 Non  
**associ**  
 True  
 |  
 Fals  
**reserv**  
 True  
 |  
 Fals  
**reserv**  
 [com  
 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

be-  
fore  
this  
in-  
ter-  
val  
in  
sec-  
onds

- **lim**  
Max  
i-  
mun  
num  
ber  
of  
node  
to  
re-  
turn

- **mar**  
the  
last  
item  
of  
the  
pre-  
vi-  
ous  
page  
we  
re-  
turn  
the  
next  
re-  
sult  
set.

- **son**  
At-  
tribu  
by  
whic  
re-  
sults  
shou  
be

sorte

- **sort**  
di-  
rec-  
tion  
in  
whic  
re-  
sults  
shou  
be  
sorte  
(asc.  
desc

**Returns**

A  
list  
of  
tu-  
ples  
of  
the  
spec  
i-  
fied  
colu

**abstract**

Re-  
turn  
ob-  
jects  
with  
ver-  
sion  
that  
are  
not  
the  
spec  
i-  
fied  
ver-  
sion

**Parame**

- **mod**  
the

nam
 of
 the
 mod
 (clas
 of
 de-
 sirec
 ob-
 jects

- **ver**
 list
 of
 ver-
 sion
 of
 ob-
 jects
 not
 to
 be
 re-
 turn

**Returns**
 list
 of
 the
 DB
 ob-
 jects

**Raises**
 Iron
 icEx
 cep-
 tion
 if
 there
 is
 no
 class
 as-
 so-
 ci-
 ated
 with
 the
 nam

**abstract**

Get  
 a  
 list  
 con-  
 duc-  
 tors  
 that  
 are  
 of-  
 fine  
 (dea

**Parameters**  
**field**  
 A  
 field  
 to  
 re-  
 turn  
 host  
 nam  
 by  
 de-  
 fault

**Returns**  
 A  
 list  
 of  
 re-  
 ques  
 field  
 of  
 of-  
 fine  
 con-  
 duc-  
 tors.

**Abstract**  
 Get  
 a  
 list  
 con-  
 duc-  
 tor  
 host  
 nam  
 that  
 are  
 on-  
 line  
 and



ac-  
tive.

**Returns**

A  
list  
of  
con-  
duc-  
tor  
host  
nam

**abstract**

Re-  
turn  
a  
net-  
worl  
port  
rep-  
re-  
sen-  
ta-  
tion.

**Parame**

**add**  
The  
MA  
ad-  
dres  
of  
a  
port

**Returns**

A  
port

**abstract**

Re-  
turn  
a  
net-  
worl  
port  
rep-  
re-  
sen-  
ta-  
tion.

**Parame**

**por**  
 The  
 id  
 of  
 a  
 port

**Returns**  
 A  
 port

**abstract**  
 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-  
 worl  
 port  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

**Parame**  
**por**  
 The  
 uuid  
 of

a  
 port

Returns

A  
 port

abstract

Re-  
 turn  
 a  
 list  
 of  
 port

Paramete

- **lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 port  
 to  
 re-  
 turn
- **max**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sor**

At-tribu-  
by  
whic  
re-sults  
shou  
be  
sorte

- **sort**  
di-  
rec-  
tion  
in  
whic  
re-  
sults  
shou  
be  
sorte  
(asc.  
desc

**abstract**  
Re-  
turn  
a  
net-  
worl  
port  
grou  
rep-  
re-  
sen-  
ta-  
tion.

**Parameter**  
**add**  
The  
MA  
ad-  
dres  
of  
a  
port  
grou

**Returns**  
A  
port

grou

**Raises**

Port  
 grou  
 Not-  
 Four

**abstract**

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

**abstract**

Re-  
 turn  
 a  
 net-  
 worl  
 port  
 grou  
 rep-  
 re-  
 sen-  
 ta-

tion.

**Parameters**

**name**

The log-ical name of a port group.

**Returns**

A port group.

**Raises**

PortGroupException: Not found.

**abstract**

Return a network port group representation.

**Parameters**

**port**

The uuid of a port group.

**Returns**

A port group.

**Raises**

Port

grou
 Not-
 Four

abstract

Re-
 turn
 a
 list
 of
 port
 grou

Paramet

- **lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 port  
 grou  
 to  
 re-  
 turn

- **max**  
 The  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **son**  
 At-

tribu  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **son**
  
 Di-
   
 rec-
   
 tion
   
 in
   
 whic
   
 re-
   
 sults
   
 shou
   
 be
   
 sorte
   
 (asc
   
 desc

Returns

A
   
 list
   
 of
   
 port
   
 grou

abstract

List
   
 all
   
 the
   
 port
   
 grou
   
 for
   
 a
   
 give
   
 node

Paramete

- **nod**
  
 The
   
 in-
   
 te-
   
 ger
   
 node
   
 ID.



- **limit**  
Maximum number of ports grouped to return
- **max**  
The last item of the previous page we return the next result set.
- **sort**  
Attribute by which results should be sorted
- **sort**  
Direction in which

re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**Returns**  
 A  
 list  
 of  
 port  
 grou

**abstract**  
  
 List  
 all  
 the  
 port  
 for  
 a  
 give  
 node

**Parame**

- nod**  
 The  
 in-  
 te-  
 ger  
 node  
 ID.
- lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 port  
 to  
 re-  
 turn
- mar**  
 the

last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

•
   
**sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

•
   
**sort**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**Returns**  
 A  
 list  
 of  
 port

**abstrac**

List  
all  
the  
port.  
for  
a  
give  
port.  
grou

Paramete

- **por**  
The  
in-  
te-  
ger  
port.  
grou  
ID.
- **lim**  
Max  
i-  
mun  
num  
ber  
of  
port.  
to  
re-  
turn
- **mar**  
The  
last  
item  
of  
the  
pre-  
vi-  
ous  
page  
we  
re-  
turn  
the  
next  
re-

sult  
 set.

- **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**
  
 Re-
 turn
 a
 vol-
 ume
 con-
 nec-
 tor
 rep-
 re-
 sen-
 ta-
 tion.

**Parameter**
  
**db\_**
  
 The

is not found.

in-  
te-  
ger  
data  
ID  
of  
a  
vol-  
ume  
con-  
nec-  
tor.

**Returns**  
A  
vol-  
ume  
con-  
nec-  
tor  
with  
the  
spec  
i-  
fied  
ID.

**Raises**  
Vol-  
ume  
Con  
nec-  
torN  
Four  
If  
a  
vol-  
ume  
con-  
nec-  
tor  
with  
the  
spec  
i-  
fied  
ID

**abstract**  
Re-  
turn  
a

vol-  
 ume  
 con-  
 nec-  
 tor  
 rep-  
 re-  
 sen-  
 ta-  
 tion.

Param

con  
 The  
 UUI  
 of  
 a  
 con-  
 nec-  
 tor.

Returns

A  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 spec  
 i-  
 fied  
 UUI

Raises

Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 If  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the  
 spec  
 i-

is not found.

fied  
 UUI

abstract

Re-  
 turn  
 a  
 list  
 of  
 vol-  
 ume  
 con-  
 nec-  
 tors.

Paramete

- **lim**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 vol-  
 ume  
 con-  
 nec-  
 tors  
 to  
 re-  
 turn

- **mar**  
 The  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next



re-  
sult  
set.

- **sort**  
At-  
tribu-  
by  
whic  
re-  
sults  
shou  
be  
sorte

- **sort**  
Di-  
rec-  
tion  
in  
whic  
re-  
sults  
shou  
be  
sorte  
(asc  
desc

**Returns**

A  
list  
of  
vol-  
ume  
con-  
nec-  
tors.

**Raises**

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

**Parameter**

- **nodeId**  
The  
in-  
te-  
ger  
node  
ID.
- **limit**  
Max  
i-  
mun  
num  
ber  
of  
vol-  
ume  
con-  
nec-  
tors  
to  
re-  
turn
- **marker**  
The  
last  
item

of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 Di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc  
 desc

**Returns**  
 A  
 list  
 of  
 vol-  
 ume  
 con-  
 nec-  
 tors.

**Raises**

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.

**Parameter**

**db\_**  
 The  
 data  
 pri-  
 mary  
 key  
 (in-  
 te-  
 ger)  
 ID  
 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  
ID  
ex-  
ists.

**abstract**

Re-  
turn  
a  
vol-  
ume  
tar-  
get  
rep-  
re-  
sen-  
ta-  
tion.

**Parame**

**uui**  
The  
UUI  
of  
a  
vol-  
ume  
tar-  
get.

**Returns**

A  
vol-  
ume  
tar-  
get.

**Raises**

Vol-  
ume

get-  
 Not-  
 Four  
 if  
 no  
 vol-  
 ume  
 tar-  
 get  
 with  
 this  
 UUI  
 ex-  
 ists.

**abstract**

Re-  
 turn  
 a  
 list  
 of  
 vol-  
 ume  
 tar-  
 gets

**Parameters**

- **limit**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 vol-  
 ume  
 tar-  
 gets  
 to  
 re-  
 turn
- **max**  
 the  
 last  
 item  
 of  
 the

pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**Returns**  
 A  
 list  
 of  
 vol-  
 ume  
 tar-  
 gets

**Raises**  
 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

**Parameters**

- node\_id**  
 The  
 in-  
 te-  
 ger  
 node  
 ID.
- limit**  
 Max  
 i-  
 mun  
 num  
 ber  
 of  
 vol-  
 ume  
 tar-  
 gets  
 to  
 re-  
 turn



- **mar**  
 the  
 last  
 item  
 of  
 the  
 pre-  
 vi-  
 ous  
 page  
 we  
 re-  
 turn  
 the  
 next  
 re-  
 sult  
 set.

- **sort**  
 At-  
 tribu-  
 by  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte

- **sort**  
 di-  
 rec-  
 tion  
 in  
 whic  
 re-  
 sults  
 shou  
 be  
 sorte  
 (asc.  
 desc

**Returns**  
 A  
 list  
 of  
 vol-  
 ume

tar-gets

Raises

In-valid Pa-ram-e-ter-Valu-if sort\_does not ex-ist.

abstract

List all the vol-ume tar-gets for a give vol-ume id.

Parameters

- **vol** The UUI of the vol-ume
- **lim** Maximum i-mun-num

ber  
of  
vol-  
ume  
tar-  
gets  
to  
re-  
turn

- **mar**  
the  
last  
item  
of  
the  
pre-  
vi-  
ous  
page  
we  
re-  
turn  
the  
next  
re-  
sult  
set.

- **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  
vol-  
ume  
tar-  
gets

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
sort  
does  
not  
ex-  
ist.

**abstract**

List  
all  
reg-  
is-  
tered  
hard  
ware  
in-  
ter-  
face  
for  
a  
con-  
duc-  
tor.

**Parameter**

**con**  
Data  
ID  
of  
con-  
duc-  
tor.

#### Returns

List  
of  
Con  
ob-  
jects

#### abstract

List  
reg-  
is-  
tere  
hard  
ware  
in-  
ter-  
face  
for  
give  
hard  
ware  
type

This  
is  
re-  
stric  
to  
only  
ac-  
tive  
con-  
duc-  
tors.  
:par  
hard  
ware  
list  
of  
hard  
ware  
type  
to

filter by. :returns: list of ConductorHardwareInterfaces objects.

#### abstract

Trie  
to  
mi-  
grate  
away  
from  
the

all the objects will be migrated.

iscsi  
 de-  
 ploy  
 in-  
 ter-  
 face  
**Parameters**  

- con**  
 the  
 ad-  
 min  
 con-  
 text
- max**  
 The  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 ob-  
 jects  
 to  
 mi-  
 grate  
 Mus  
 be  
 >=  
 0.  
 If  
 zero

**Returns**  
 A  
 2-  
 tuple  
 1.  
 the  
 to-  
 tal  
 num  
 ber  
 of  
 ob-  
 jects

the beginning of this call) and 2. the number of migrated objects.

that  
 need  
 to  
 be  
 mi-  
 grate  
 (at

**abstract**

Check  
 if  
 the  
 spec  
 i-  
 fied  
 tag  
 ex-  
 ist  
 on  
 the  
 node

**Parameters**

- node**  
 The  
 id  
 of  
 a  
 node
- tag**  
 A  
 tag  
 string

**Returns**

True  
 if  
 the  
 tag  
 ex-  
 ists  
 oth-  
 er-  
 wise  
 False

**Raises**

Node

Not-  
 Four  
 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
- **trait**  
A  
trait  
string

**Returns**

True  
 if  
 the  
 trait  
 ex-  
 ists  
 oth-  
 er-  
 wise  
 False

**Raises**

Node



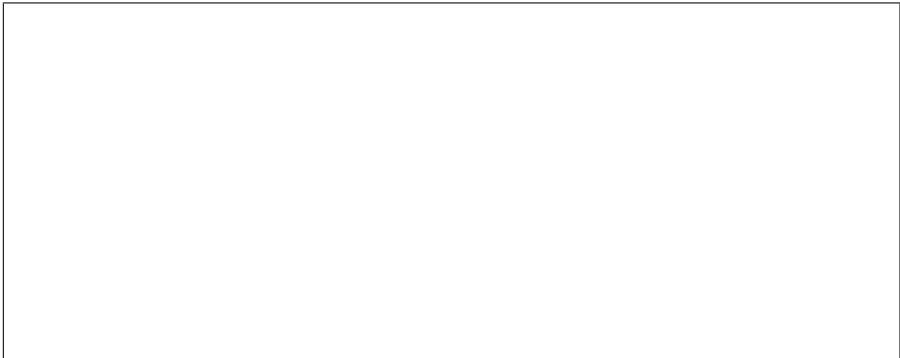
Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 found

**abstract**

Reg  
 is-  
 ter  
 an  
 ac-  
 tive  
 con-  
 duc-  
 tor  
 with  
 the  
 clus  
 ter.

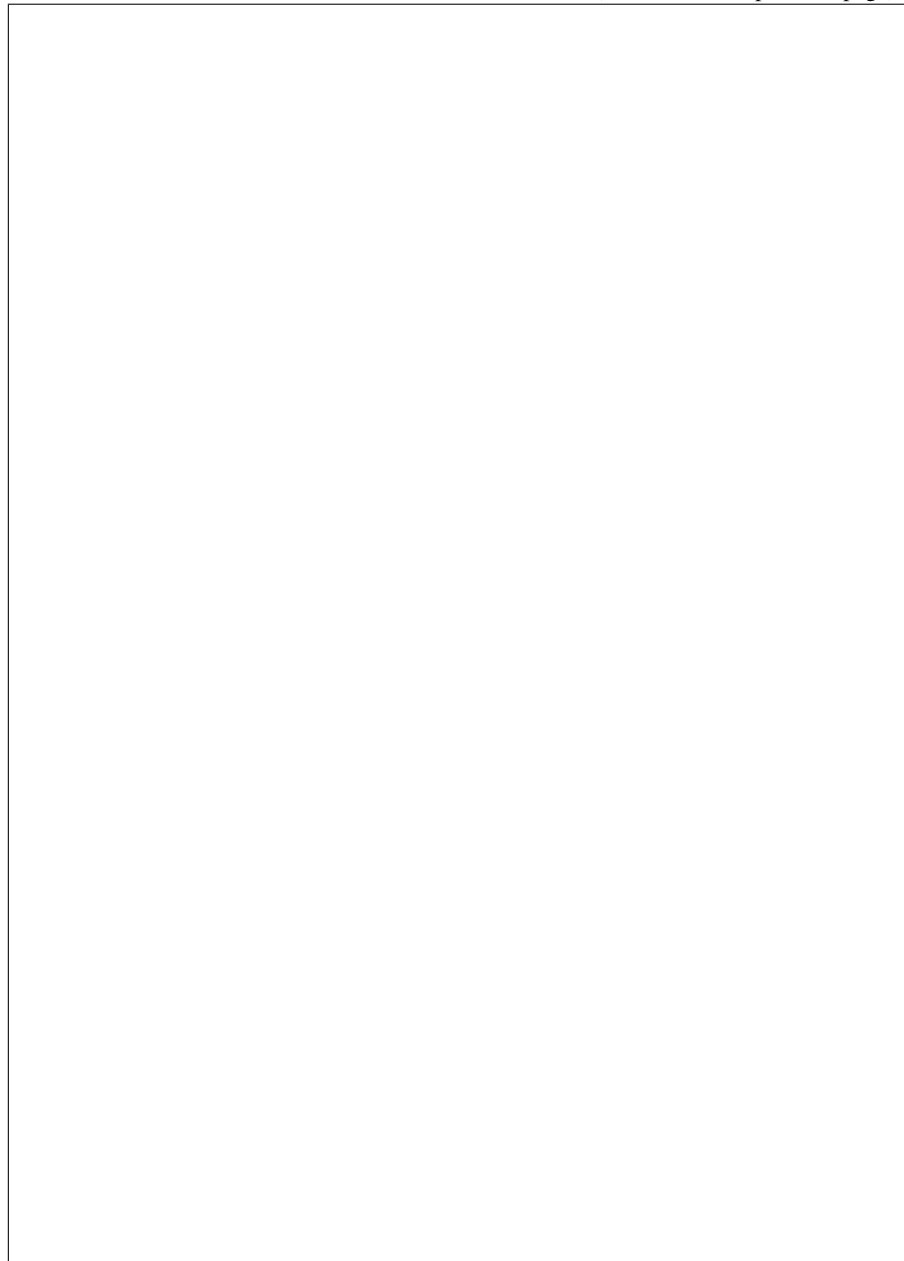
**Paramete**

- val**  
 A  
 dict  
 of  
 val-  
 ues  
 whic  
 mus  
 con-  
 tain  
 the  
 fol-  
 low-  
 ing:



(continues on next page)

(continued from previous page)



•  
upd  
Whe  
false  
reg-  
is-  
tra-  
tion  
will  
raise  
an  
ex-  
cep-  
tion  
whe

line record is found. When true, will overwrite the existing record. Default: False.

a  
con-  
flict-  
ing  
on-

Returns

A  
con-  
duc-  
tor.

Raises

Con  
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-  
 is-  
 ter.  
 •  
**def**  
 Strin  
 the  
 de-  
 fault  
 in-  
 ter-  
 face  
 for  
 this

hard  
ware  
type  
and  
in-  
ter-  
face  
type

#### Raises

Con  
duc-  
torH  
ware  
ter-  
face  
sAl-  
read  
is-  
tere  
if  
at  
least  
one  
of  
the  
in-  
ter-  
face

in the combination of all parameters is already registered.

#### abstract

Re-  
lease  
the  
rese  
va-  
tion  
on  
a  
node

#### Parame

- 

**tag**  
A  
strin  
uniq  
iden  
ti-  
fy-

ing  
 the  
 rese  
 va-  
 tion  
 hold

- **node**  
 A  
 node  
 id  
 or  
 uuid

**Raises**

Nod  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 found

**Raises**

Nod  
 Loc  
 if  
 the  
 node  
 is  
 re-  
 serv  
 by  
 an-  
 othe  
 host

**Raises**

Nod  
 Not-  
 Loc  
 if  
 the  
 node  
 was  
 found  
 to  
 not  
 have  
 a

rese  
va-  
tion  
at  
all.

**abstract**

Re-  
serv  
a  
node  
  
To  
pre-  
vent  
othe  
Man  
ager  
vice  
from  
ma-  
nip-  
u-  
lat-  
ing  
the  
give  
Nod  
whil  
a  
Task  
is

performed, mark it reserved by this host.

**Parame**

- tag**  
A  
strin  
uniq  
iden  
ti-  
fy-  
ing  
the  
rese  
va-  
tion  
hold
- nod**

A  
 node  
 id  
 or  
 uuid

**Returns**

A  
 Node  
 ob-  
 ject.

**Raises**

Node  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 foun

**Raises**

Node  
 Lock  
 if  
 the  
 node  
 is  
 al-  
 read  
 re-  
 serv

**abstract**

Re-  
 plac  
 all  
 of  
 the  
 node  
 tags  
 with  
 spec  
 i-  
 fied  
 list  
 of  
 tags  
 This  
 ig-  
 nore



du-  
pli-  
cate  
tags  
in  
the  
spec  
i-  
fied  
list.

Param

- **node**  
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

abstrac

Re-  
plac  
all  
of  
the

node  
 trait  
 with  
 spec  
 i-  
 fied  
 list  
 of  
 trait.  
 This  
 ig-  
 nore  
 du-  
 pli-  
 cate  
 trait.  
 in  
 the  
 spec  
 i-  
 fied  
 list.

Paramete

- **node**  
 The  
 id  
 of  
 a  
 node
- **traits**  
 List  
 of  
 trait
- **version**  
 the  
 ver-  
 sion  
 of  
 the  
 ob-  
 ject.

Returns

A  
 list

limit.

of  
Nod  
Trai  
ob-  
jects

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
set-  
ting  
the  
trait  
wou  
ex-  
ceed  
the  
per-  
node  
trait

#### **Raises**

Nod  
Not-  
Foun  
if  
the  
node  
is  
not  
foun

#### **abstract**

Do  
a  
take  
over  
for  
an  
al-  
lo-  
ca-  
tion.  
The

thus guarding against races.

cation.

Parame

- **all**  
 Al-  
 lo-  
 ca-  
 tion  
 ID
- **old**  
 The  
 con-  
 duc-  
 tor  
 ID  
 we  
 ex-  
 pect  
 to  
 be  
 the  
 cur-  
 rent  
 con  
 of  
 the  
 al-  
 lo-

•

**new**  
The  
con-  
duc-  
tor  
ID  
of  
the  
new  
con

**Returns**

True  
if  
the  
take  
over  
was  
suc-  
cess  
ful,  
Fals  
oth-  
er-  
wise

**Raises**

Al-  
lo-  
ca-  
tion-  
Not-  
Foun

**abstract**

Mar  
a  
con-  
duc-  
tor  
as  
ac-  
tive  
by  
up-  
dat-  
ing  
its  
up-  
date  
prop  
erty.

**Parameters**

**host**

The host name of this contributor service vice

**Raises**

ContributorNotFoundError
 Four

**abstract**

Mark the node provision ing as run ning

Mark the node provision ing as run ning by up dat ing its pro vision prop erty.

**Parameters**

**nod**  
The  
id  
of  
a  
node

**Raises**  
Nod  
Not-  
Four

**abstract**  
Re-  
mov  
this  
con-  
duc-  
tor  
from  
the  
ser-  
vice  
reg-  
istry  
im-  
me-  
di-  
ately

**Parame**  
**hos**  
The  
host  
nam  
of  
this  
con-  
duc-  
tor  
ser-  
vice

**Raises**  
Con  
duc-  
torN  
Four

**abstract**  
Un-  
reg-  
is-  
ters

all  
 hard  
 ware  
 in-  
 ter-  
 face  
 for  
 a  
 con-  
 duc-  
 tor.

Paramete

con  
 Data  
 ID  
 of  
 con-  
 duc-  
 tor  
 to  
 un-  
 reg-  
 is-  
 ter  
 for.

abstract

Re-  
 mov  
 all  
 tags  
 of  
 the  
 node

Paramete

nod  
 The  
 id  
 of  
 a  
 node

Raises

Nod  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not



found

**abstract**

Re-  
mov  
all  
trait  
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.

**Parameter**

- **all**  
Al-  
lo-  
ca-  
tion  
ID
-

allocation

val  
 Dict  
 of  
 val-  
 ues  
 to  
 up-  
 date  
 •  
 upd  
 If  
 True  
 and  
 node  
 is  
 up-  
 date  
 up-  
 date  
 the  
 node  
 with  
 in-  
 stan  
 and  
 trait  
 from  
 the

Returns  
 An  
 al-  
 lo-  
 ca-  
 tion.

Raises  
 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

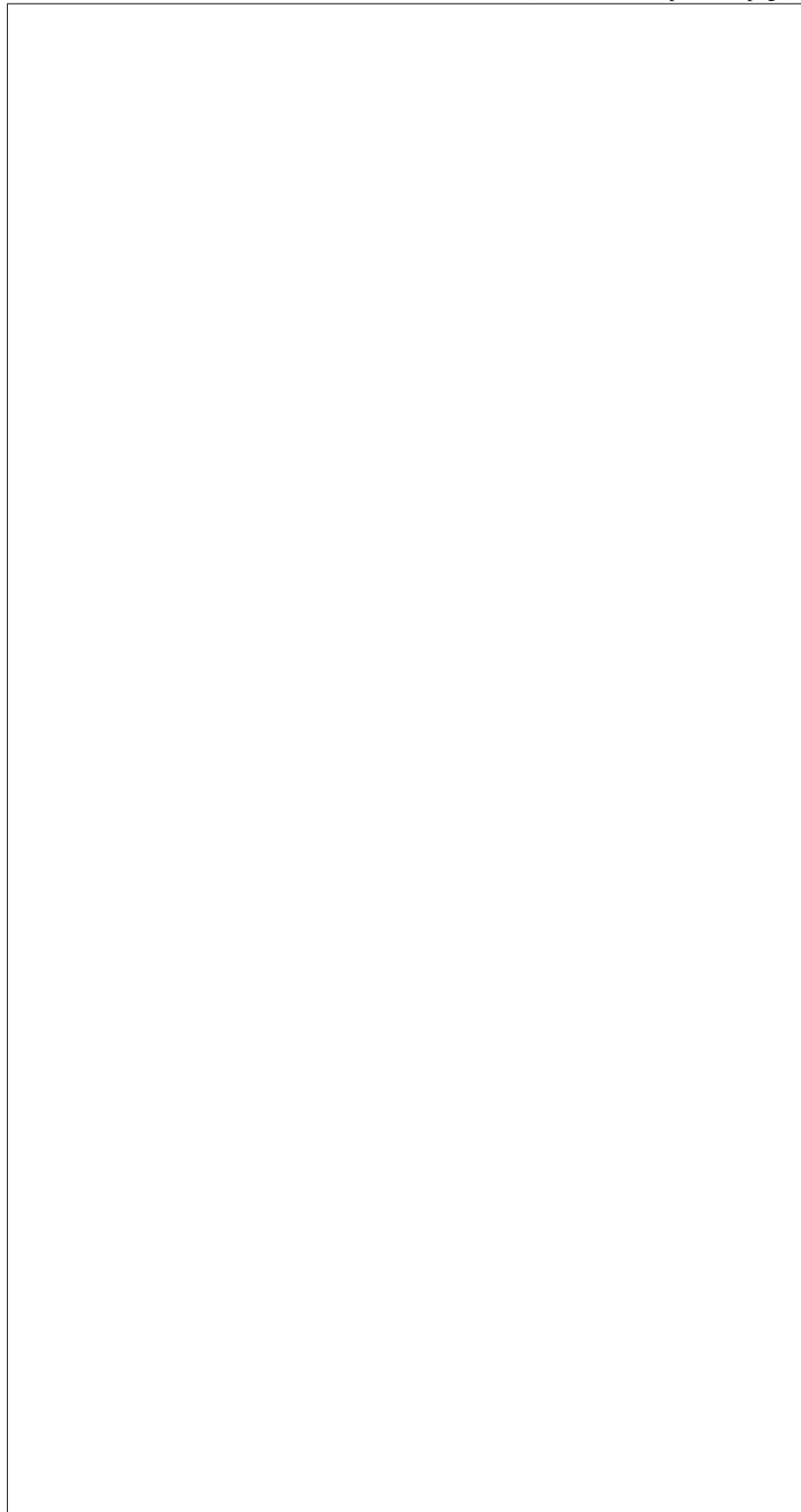
**Parame**

- nod**
  
 The
 node
 id.
- set**
  
 A
 list
 of
 BIO
 Set-
 tings
 to
 be
 up-
 date



(continues on next page)

(continued from previous page)



the  
 ver-  
 sion  
 of  
 the  
 ob-  
 ject.

**Returns**

A  
 list  
 of  
 BIO-  
 Set-  
 ting  
 ob-  
 jects

**Raises**

Nod-  
 Not-  
 Four-  
 if  
 the  
 node  
 is  
 not  
 foun

**Raises**

BIO-  
 Set-  
 ting-  
 Not-  
 Four-  
 if  
 any  
 of  
 the  
 set-  
 ting-  
 is  
 not  
 foun

**abstract**

Up-  
 date  
 prop-  
 er-  
 ties  
 of  
 an

chas  
sis.

Paramete

- **cha**  
 The  
 id  
 or  
 the  
 uuid  
 of  
 a  
 chas  
 sis.
- **val**  
 Dict  
 of  
 val-  
 ues  
 to  
 up-  
 date

Returns

A  
 chas  
 sis.

abstra

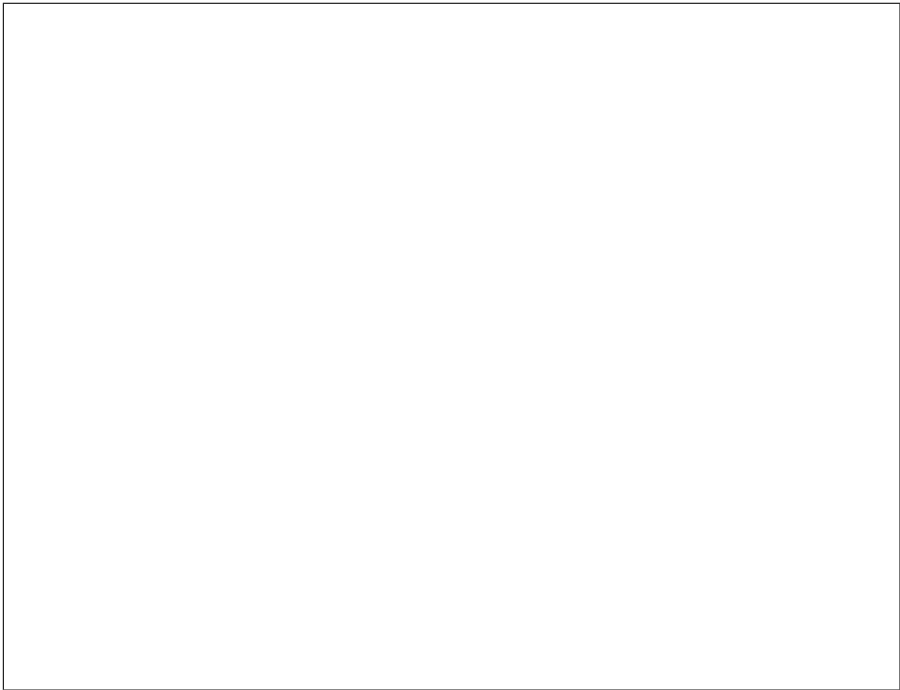
Up-  
 date  
 a  
 de-  
 ploy  
 men  
 tem-  
 plate

Paramete

- **tem**  
 ID  
 of  
 the  
 de-  
 ploy  
 men  
 tem-

plate  
 to  
 up-  
 date

- **val**  
 A  
 dict  
 de-  
 scrib  
 ing  
 the  
 de-  
 ploy  
 men  
 tem-  
 plate  
 For  
 ex-  
 am-  
 ple:



**Raises**  
 De-  
 ploy  
 plat-  
 eDu  
 pli-  
 cate  
 Nam  
 if

a  
 de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 same  
 name  
 ex-  
 ists.

**Raises**

De-  
 ploy  
 plate  
 Four  
 if  
 the  
 de-  
 ploy  
 tem-  
 plate  
 does  
 not  
 ex-  
 ist.

**Returns**

A  
 de-  
 ploy  
 tem-  
 plate

**abstract**

Up-  
 date  
 prop  
 er-  
 ties  
 of  
 a  
 node

**Parameter**

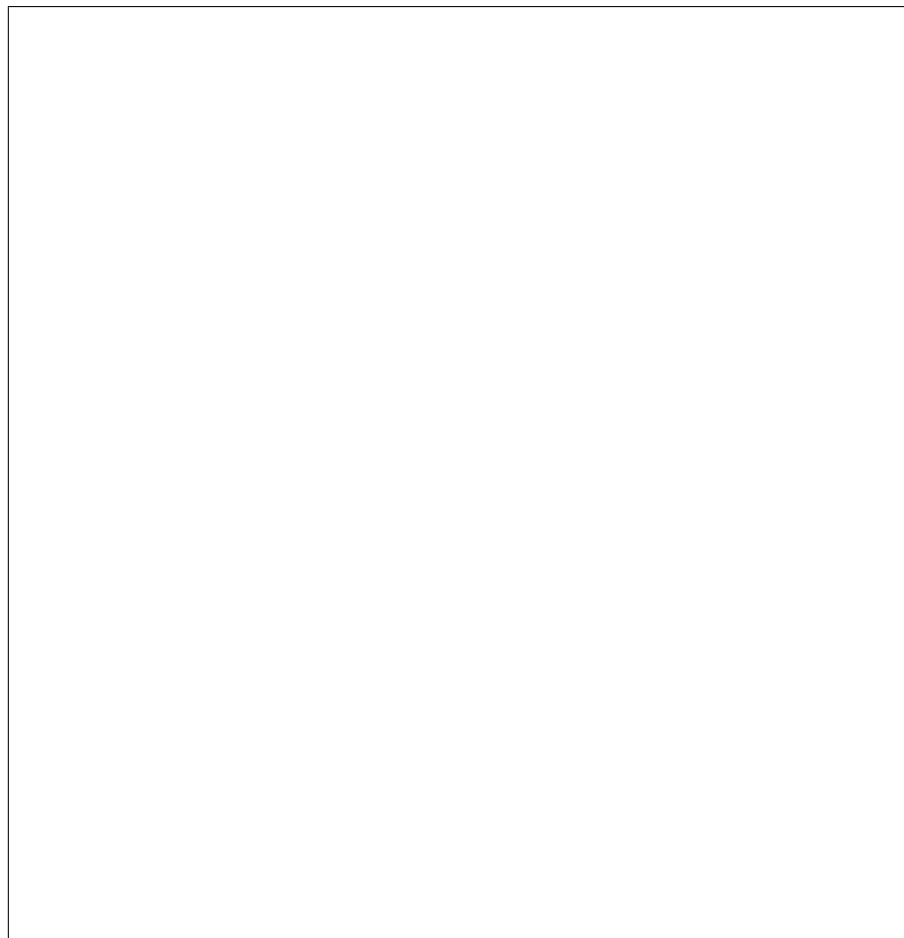
- **node**  
 The  
 id  
 or  
 uuid



of  
a  
node

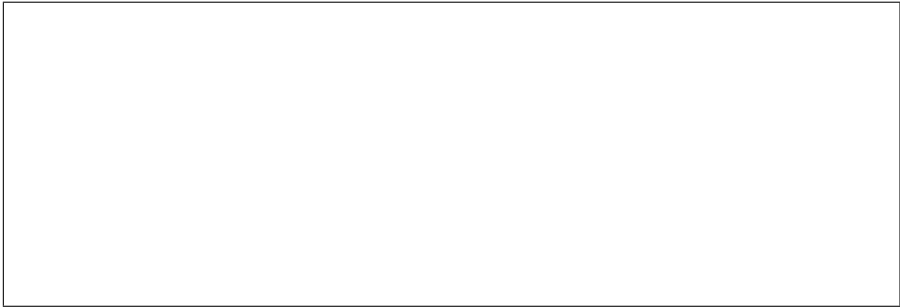
- **val**  
Dict  
of  
val-  
ues  
to  
up-  
date  
May  
be  
a  
par-  
tial  
list,  
eg.  
when  
set-  
ting  
the

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

### Paramete

- 

**por**  
The  
UUI  
or  
MA  
of  
a  
port  
group

- 

**val**  
Dict  
of  
val-  
ues  
to  
up-  
date  
May  
con-  
tain  
the  
fol-  
low-  
ing  
keys  
uuid  
nam  
node

address extra created\_at updated\_at

Returns

A  
port  
group

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu

Raises

Port  
group  
Not-  
Foun

Raises

Port  
group  
pli-  
cate  
Nam

Raises

Port  
group  
MA  
read  
ists

abstract

Up-  
date  
ob-  
jects  
to  
their  
lat-  
est  
know  
ver-  
sion  
  
This  
scan  
all  
the  
ta-  
bles

dates them to that version.

all the objects will be migrated.

and  
 for  
 ob-  
 jects  
 that  
 are  
 not  
 in  
 their  
 lat-  
 est  
 ver-  
 sion  
 up-

Parame

- **con**  
 the  
 ad-  
 min  
 con-  
 text
- **max**  
 The  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 ob-  
 jects  
 to  
 mi-  
 grate  
 Mus  
 be  
 >=  
 0.  
 If  
 zero

Returns

A  
 2-  
 tuple

the beginning of this call) and 2. the number of migrated objects.

1.  
the  
to-  
tal  
num  
ber  
of  
ob-  
jects  
that  
need  
to  
be  
mi-  
grate  
(at

**abstract**  
Up-  
date  
prop  
er-  
ties  
of  
a  
vol-  
ume  
con-  
nec-  
tor.

**Paramete**

- **ide**  
The  
UI  
or  
in-  
te-  
ger  
ID  
of  
a  
vol-  
ume  
con-  
nec-  
tor.
-

exists with a matching type and connector\_id field.

con-  
 Dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 in-  
 for-  
 ma-  
 tion  
 about  
 con-  
 nec-  
 tor  
 to  
 up-  
 date  
**Returns**  
 A  
 vol-  
 ume  
 con-  
 nec-  
 tor.  
**Raises**  
 Vol-  
 ume  
 Con-  
 nec-  
 torT  
 pe-  
 An-  
 dI-  
 dAl-  
 read  
 ists  
 If  
 an-  
 othe  
 con-  
 nec-  
 tor  
 al-  
 read  
**Raises**  
 Vol-  
 ume

does not exist.

Con  
nec-  
torN  
Four  
If  
a  
vol-  
ume  
con-  
nec-  
tor  
with  
the  
spec  
i-  
fied  
iden

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
Whe  
a  
UUI  
is  
in-  
clud  
in  
con-  
nec-  
tor\_

**abstract**

Up-  
date  
in-  
for-  
ma-  
tion  
for  
a  
vol-  
ume  
tar-  
get.

**Parame**



date.

- **ide**  
The  
UUI  
or  
in-  
te-  
ger  
ID  
of  
a  
vol-  
ume  
tar-  
get.
- **tar**  
Dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
in-  
for-  
ma-  
tion  
about  
vol-  
ume  
tar-  
get  
to  
up-

**Returns**  
A  
vol-  
ume  
tar-  
get.

**Raises**  
In-  
vali-  
Pa-  
ram-  
e-

the same boot index and node ID.

ter-  
 Valu  
 if  
 a  
 UUI  
 is  
 in-  
 clud  
 in  
 tar-  
 get\_  
**Raises**  
 Vol-  
 ume  
 get-  
 Boo  
 dex-  
 Al-  
 read  
 ists  
 if  
 a  
 vol-  
 ume  
 tar-  
 get  
 al-  
 read  
 ex-  
 ists  
 with  
**Raises**  
 Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 no  
 vol-  
 ume  
 tar-  
 get  
 with  
 this  
 iden  
 ex-  
 ists.  
 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  
port  
in  
*task*

**Parame**  
**tas**  
A  
Task  
ager  
in-  
stan

**Raises**

Fail  
To-  
Cle-  
anD  
HCF

**get\_ip**  
Get  
IP  
ad-  
dres  
for  
all  
port  
in  
*task*

**Paramet**  
**tas**  
A  
Task  
ager  
in-  
stan

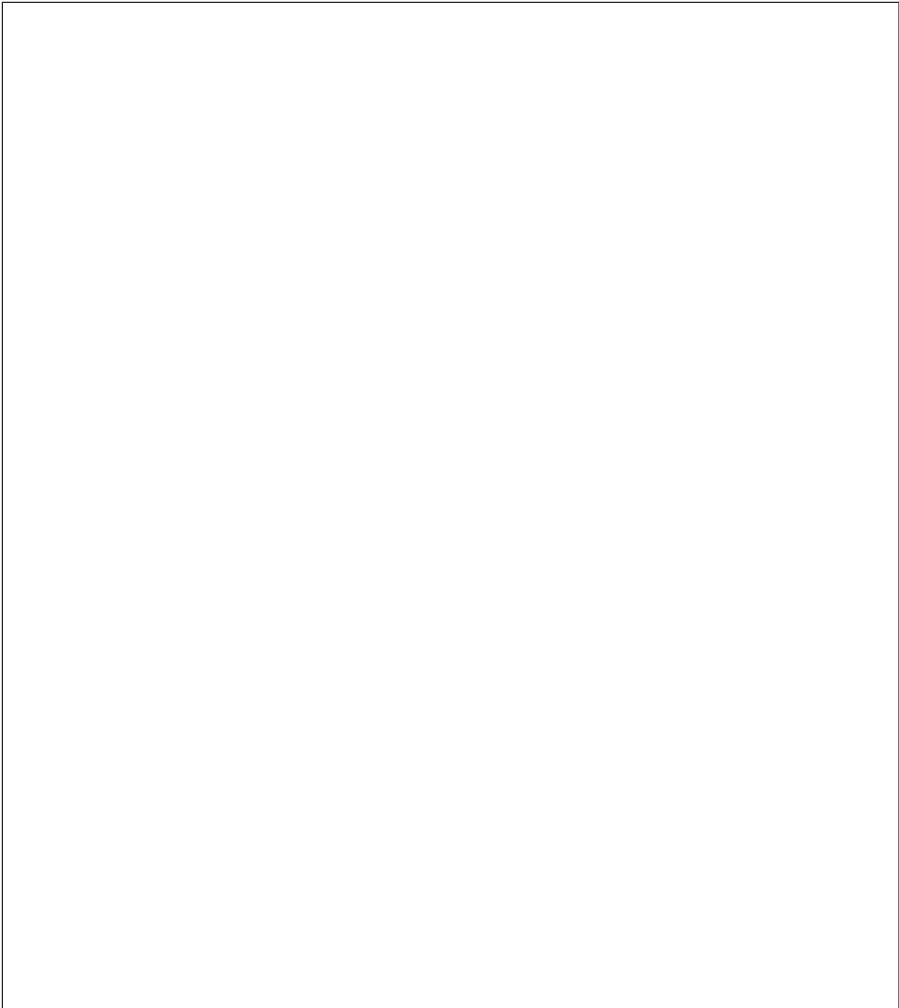
**Returns**  
List  
of  
IP  
ad-  
dres  
as-  
so-  
ci-  
ated  
with  
task  
port  
and  
port  
grou

**abstrac**  
Send  
or  
up-  
date  
the  
DHCP  
BOC  
op-  
tions  
for

this  
 node

Paramete

- **task**  
 A  
 Task  
 ager  
 in-  
 stan
- **options**  
 this  
 will  
 be  
 a  
 list  
 of  
 dict  
 e.g.



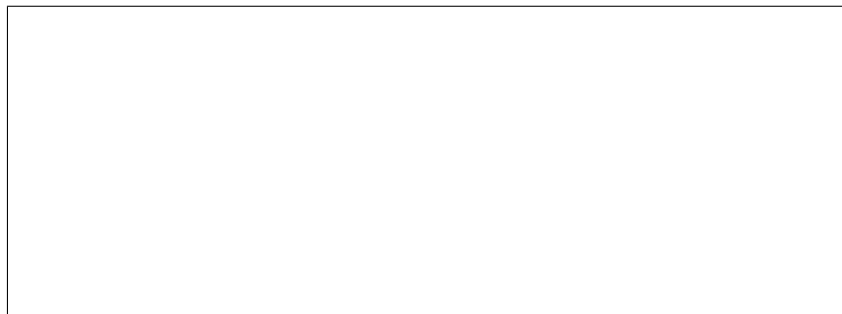
(continues on next page)

(continued from previous page)



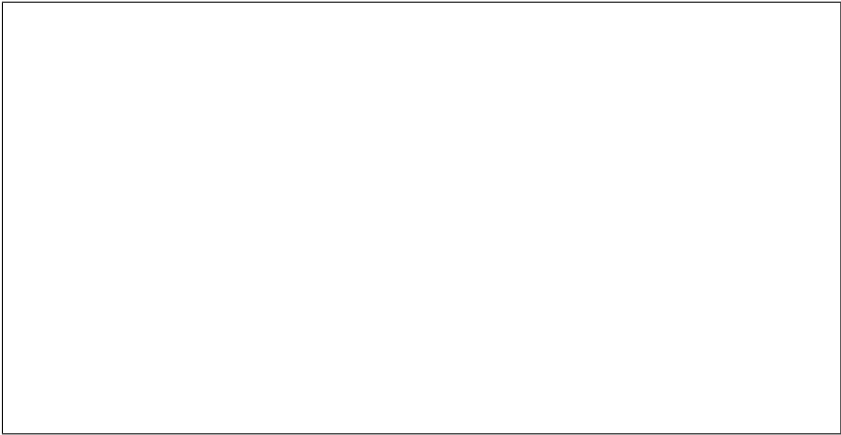
• **vi**  
A  
dict  
with  
keys  
port  
and  
port  
grou  
and  
dicts  
as  
val-  
ues.  
Each  
dict  
has  
key/  
pairs

of the form <ironic UUID>:<neutron port UUID>. e.g.



(continues on next page)

(continued from previous page)



If  
 the  
 valu  
 is  
 Non  
 will  
 get  
 the  
 list  
 of  
 port  
 from  
 the  
 Iron  
 port  
 ob-  
 jects

**Raises**

Fail  
 ToU  
 dat-  
 eD-  
 HCF  
 tOn-  
 Port

**abstract**

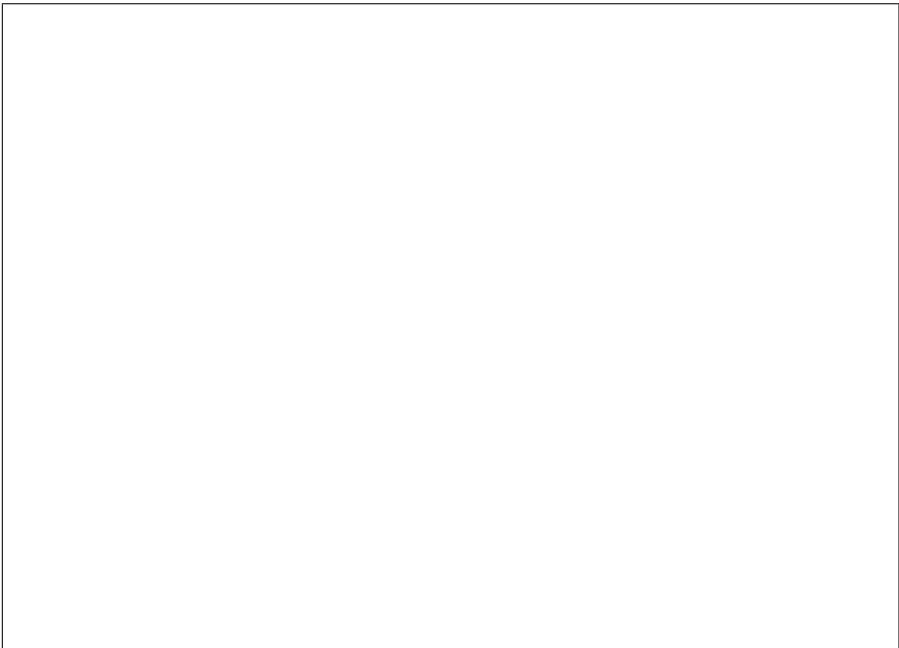
Up-  
 date  
 one  
 or  
 more  
 DHC  
 op-  
 tions  
 on  
 the



spec  
i-  
fied  
port

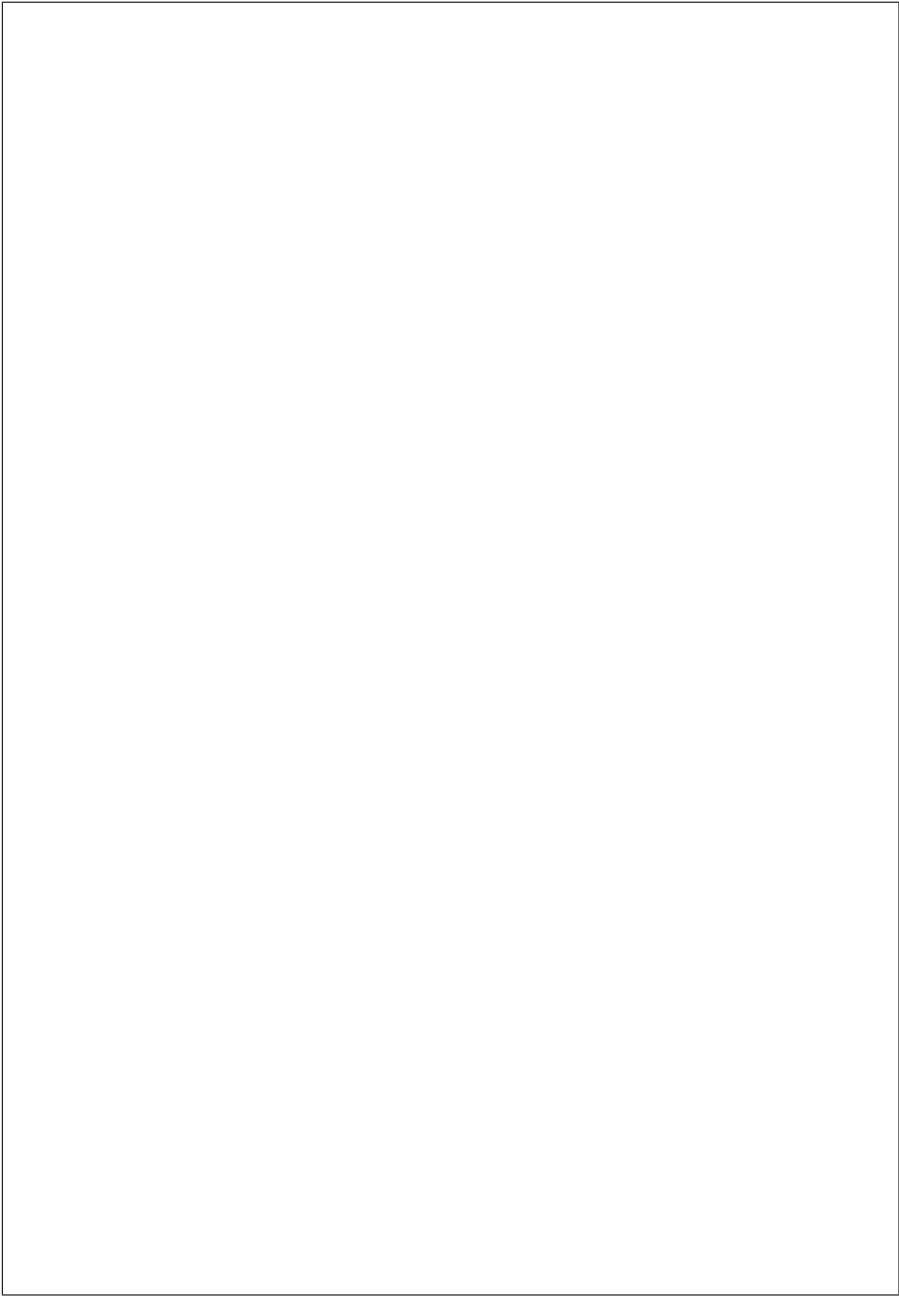
Param

- por**  
des-  
ig-  
nate  
whic  
port  
thes  
at-  
tribu  
will  
be  
ap-  
plied  
to.
- dhc**  
this  
will  
be  
a  
list  
of  
dicts  
e.g.



(continues on next page)

(continued from previous page)



• **token**  
 An  
 op-  
 tion  
 au-  
 then-  
 ti-  
 ca-  
 tion  
 to-  
 ken.  
 Dep  
 re-

cate  
use  
con-  
text

- **con**  
(ir  
com  
con  
Req  
re-  
ques  
con-  
text

**Raises**  
Fail  
ToU  
dat-  
eD-  
HCF  
tOn-  
Port

## ironic.dhcp.neutron module

**class** i  
Base  
*irc*  
*dhc*  
*bas*  
*Bas*  
API  
for  
com  
mu-  
ni-  
cat-  
ing  
to  
neu-  
tron  
2.x  
API

**get\_ip\_**  
Get  
IP  
ad-

dres  
 for  
 all  
 port  
 in  
*task*

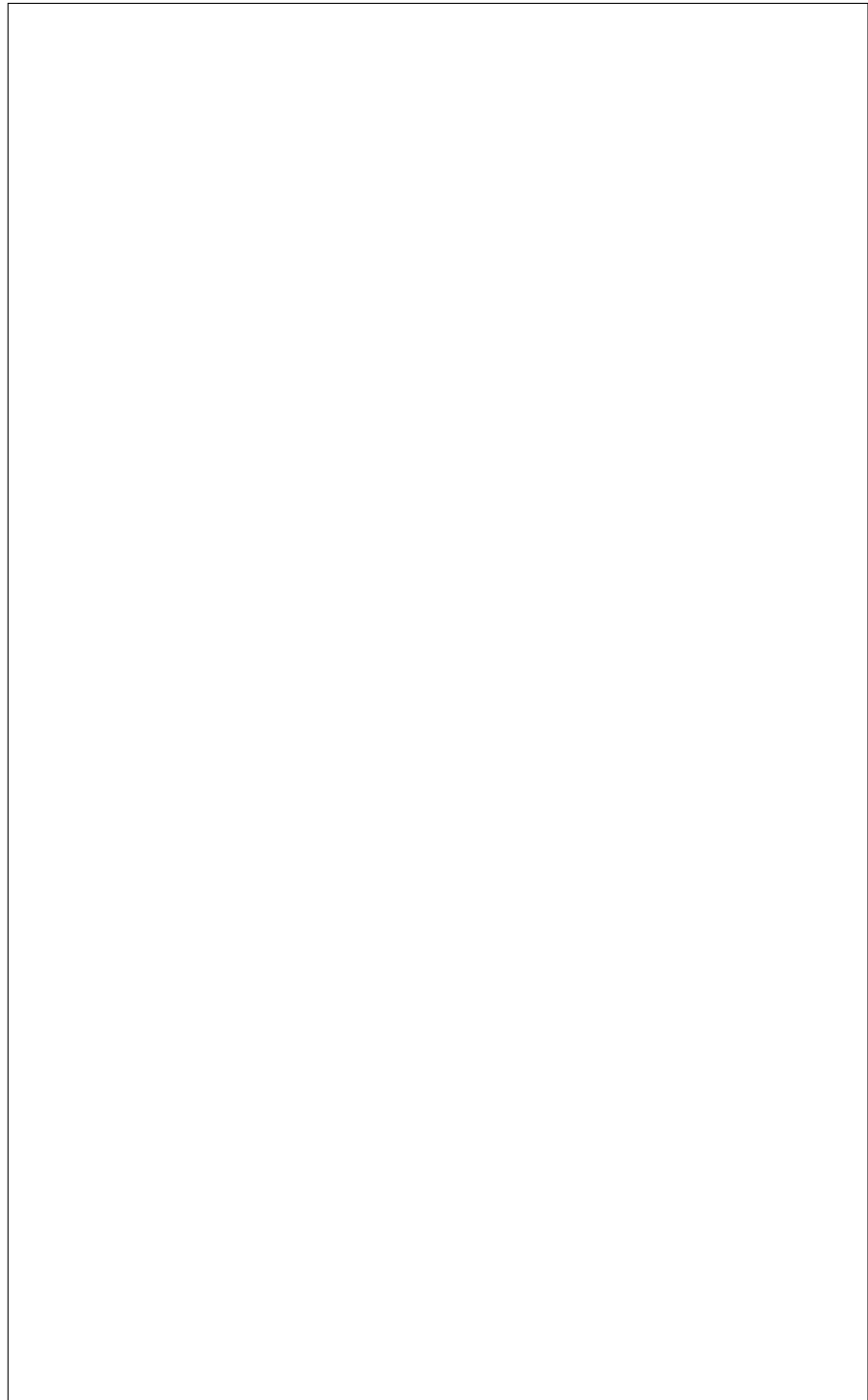
**Parameters**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan

**Returns**  
 List  
 of  
 IP  
 ad-  
 dres  
 as-  
 so-  
 ci-  
 ated  
 with  
 task  
 port

**update\_**  
 Send  
 or  
 up-  
 date  
 the  
 DHCP  
 BOO  
 op-  
 tions  
 for  
 this  
 node

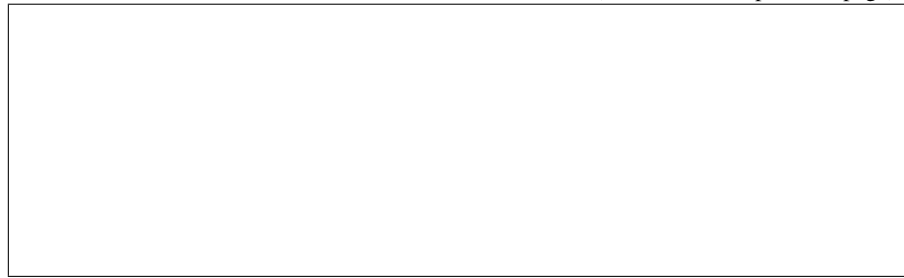
**Parameters**  
 •  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

- **opt**  
this  
will  
be  
a  
list  
of  
dicts  
e.g.



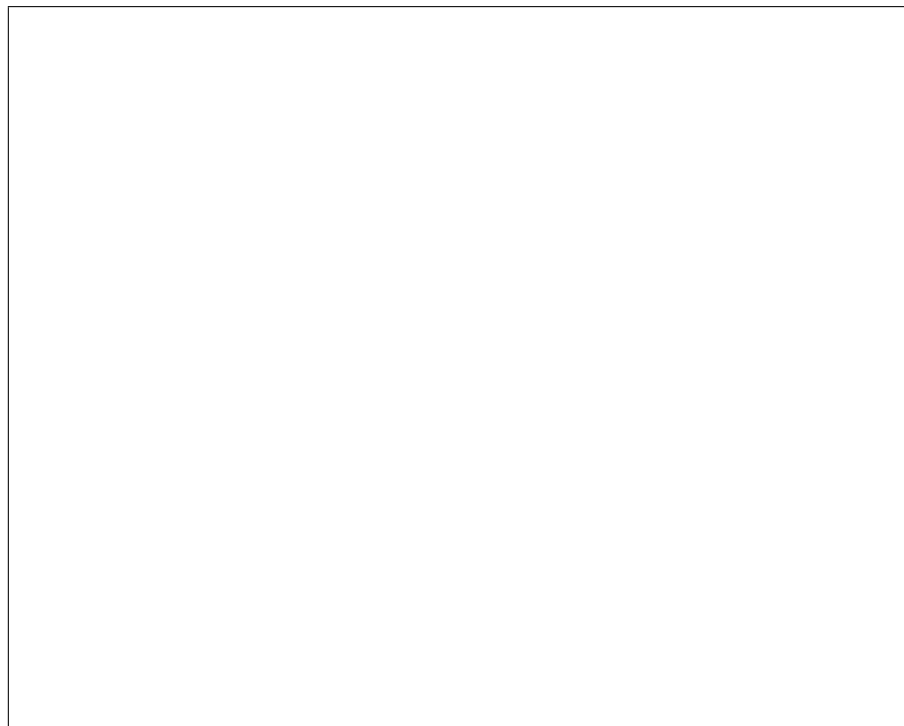
(continues on next page)

(continued from previous page)



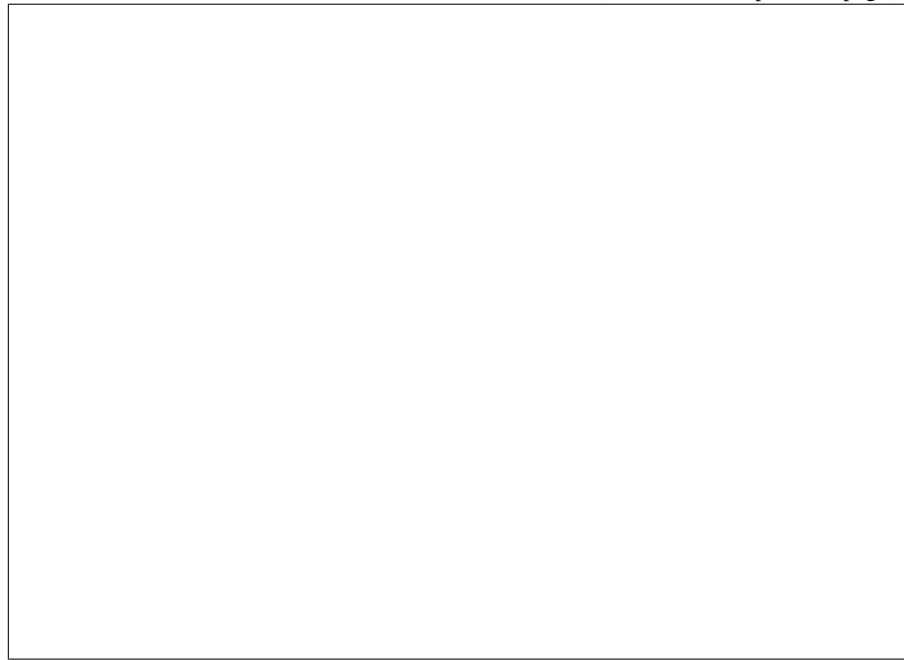
- **vi f**  
a  
dict  
of  
Neu  
tron  
port  
dicts  
to  
up-  
date  
DHCP  
op-  
tions  
on.  
The  
port  
dict  
key

should be Ironic port UUIDs, and the values should be Neutron port UUIDs, e.g.



(continues on next page)

(continued from previous page)



**update\_**

Up-  
date  
a  
port  
at-  
tribu

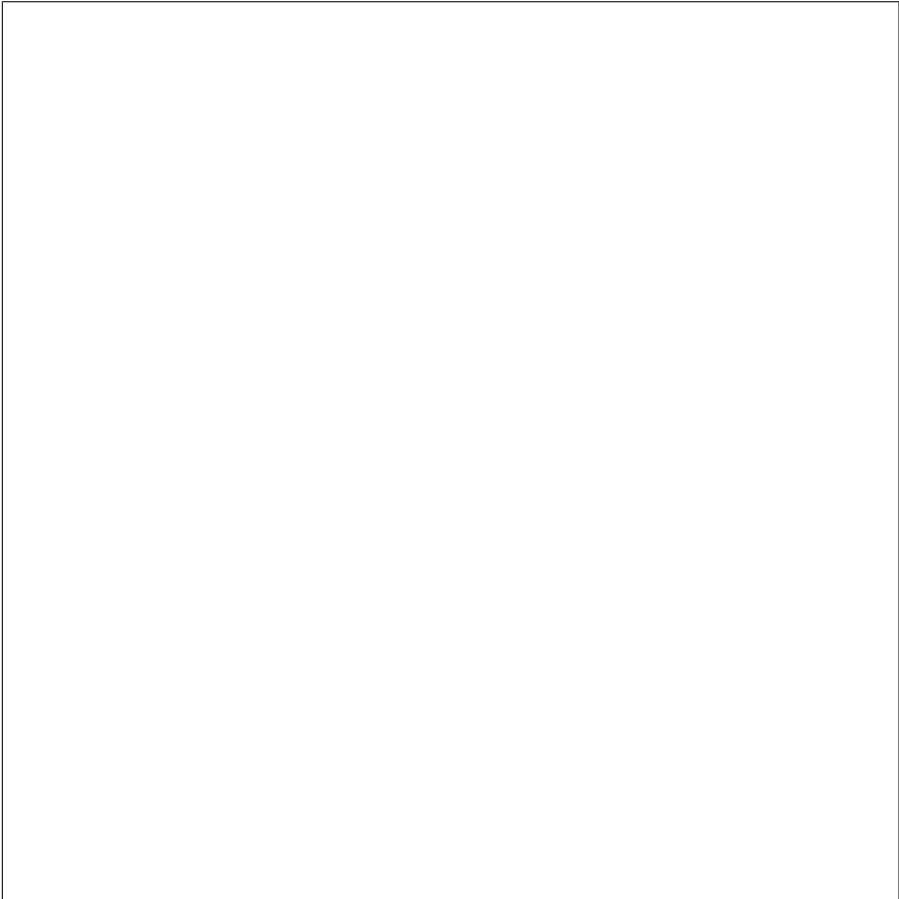
Up-  
date  
one  
or  
more  
DHCP  
op-  
tions  
on  
the  
spec  
i-  
fied  
port  
For  
the  
rel-  
e-  
vant  
API

spec, see <https://docs.openstack.org/api-ref/network/v2/index.html#update-port>

**Parame**

• **por**  
 des-  
 ig-  
 nate  
 whic  
 port  
 thes  
 at-  
 tribu  
 will  
 be  
 ap-  
 plied  
 to.

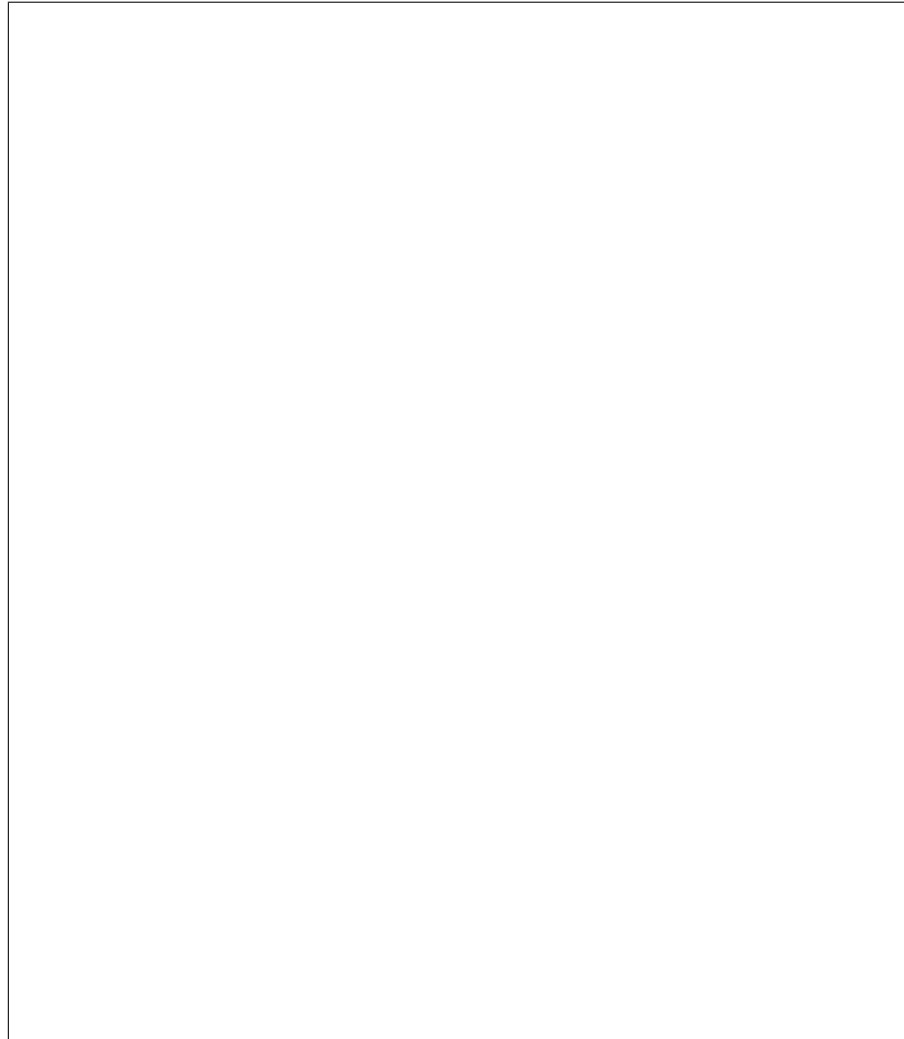
• **dhc**  
 this  
 will  
 be  
 a  
 list  
 of  
 dict  
 e.g.



(continues on next page)



(continued from previous page)



- **tok**  
op-  
tion:  
auth  
to-  
ken.  
Dep  
re-  
cate  
use  
con-  
text.
- **con**  
(ir  
com  
con  
Req  
re-

ironic.dhcp.none module

ques

con-

text

Raises

Fail

ToU

dat-

eD-

HCF

tOn-

Port

class i

Base

*irc*

*dhc*

*bas*

*Bas*

No-

op

DHC

API

get\_ip\_

Get

IP

ad-

dres

for

all

port

in

*task*

Parame

**tas**

A

Task

ager

in-

stan

Returns

List

of

IP

ad-

dres

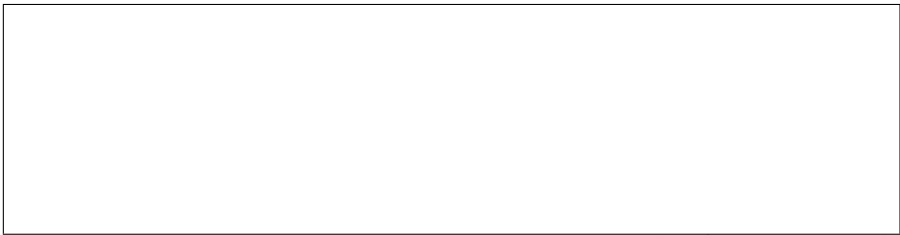
as-  
 so-  
 ci-  
 ated  
 with  
 task  
 port  
 and  
 port  
 grou

update\_

Send  
 or  
 up-  
 date  
 the  
 DHCP  
 BOOTP  
 op-  
 tions  
 for  
 this  
 node

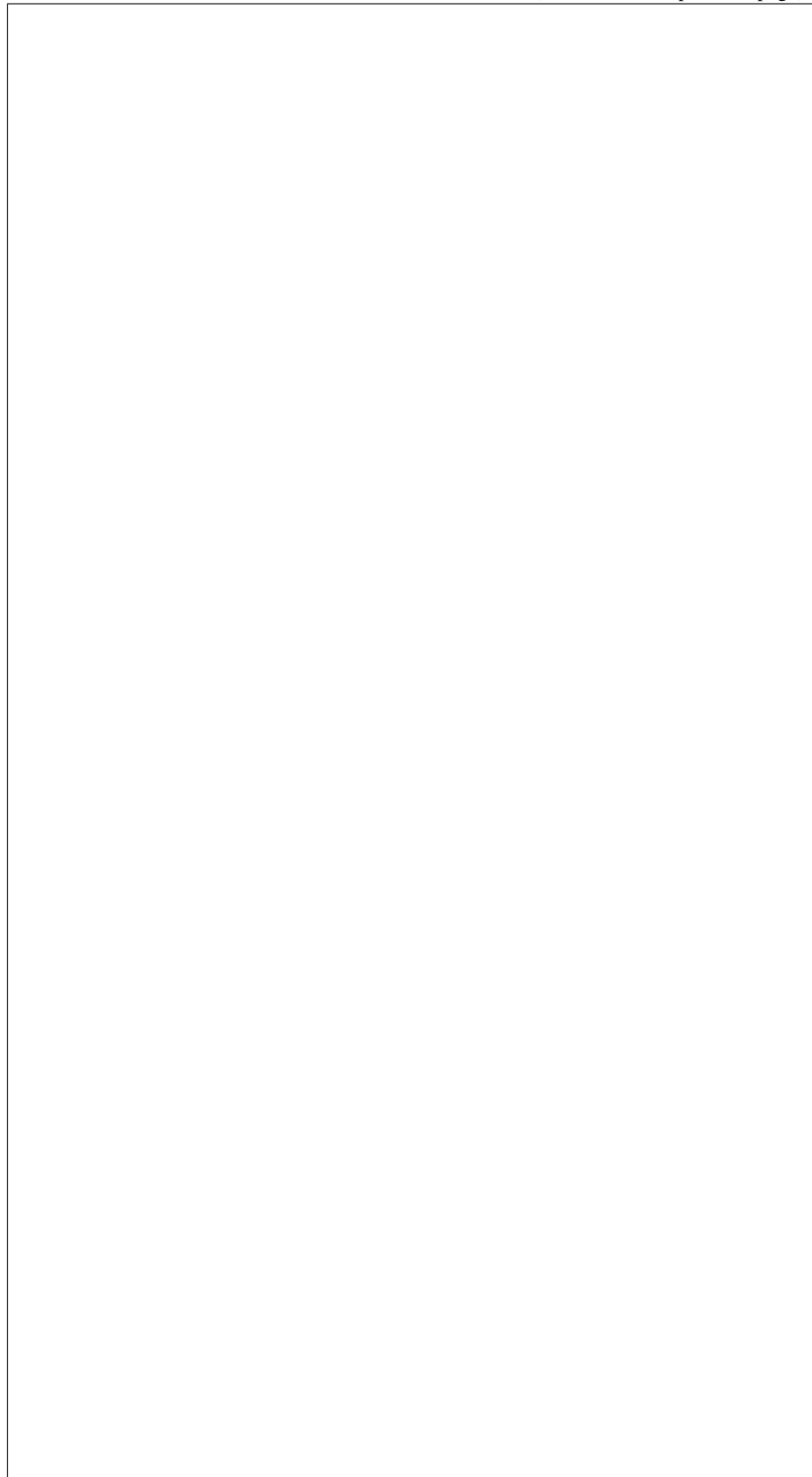
Paramet

- **task**  
 A  
 Task  
 agent  
 in-  
 stan
- **opt**  
 this  
 will  
 be  
 a  
 list  
 of  
 dict  
 e.g.

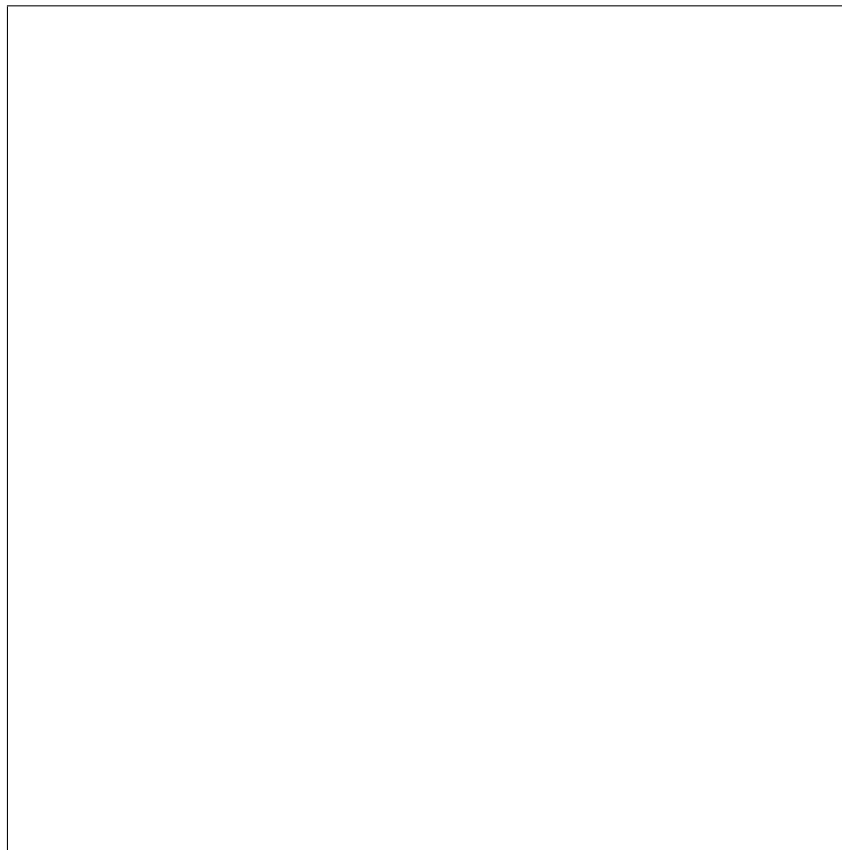


(continues on next page)

(continued from previous page)



of the form <ironic UUID>:<neutron port UUID>. e.g.



with  
keys  
port  
and  
port.  
grou  
and  
dicts  
as  
val-  
ues.  
Each  
dict  
has  
key/  
pairs

If  
the  
valu  
is  
Non  
will  
get  
the  
list  
of

port.  
 from  
 the  
 Iron  
 port.  
 ob-  
 jects

**Raises**

Fail  
 ToU  
 dat-  
 eD-  
 HCF  
 tOn-  
 Port

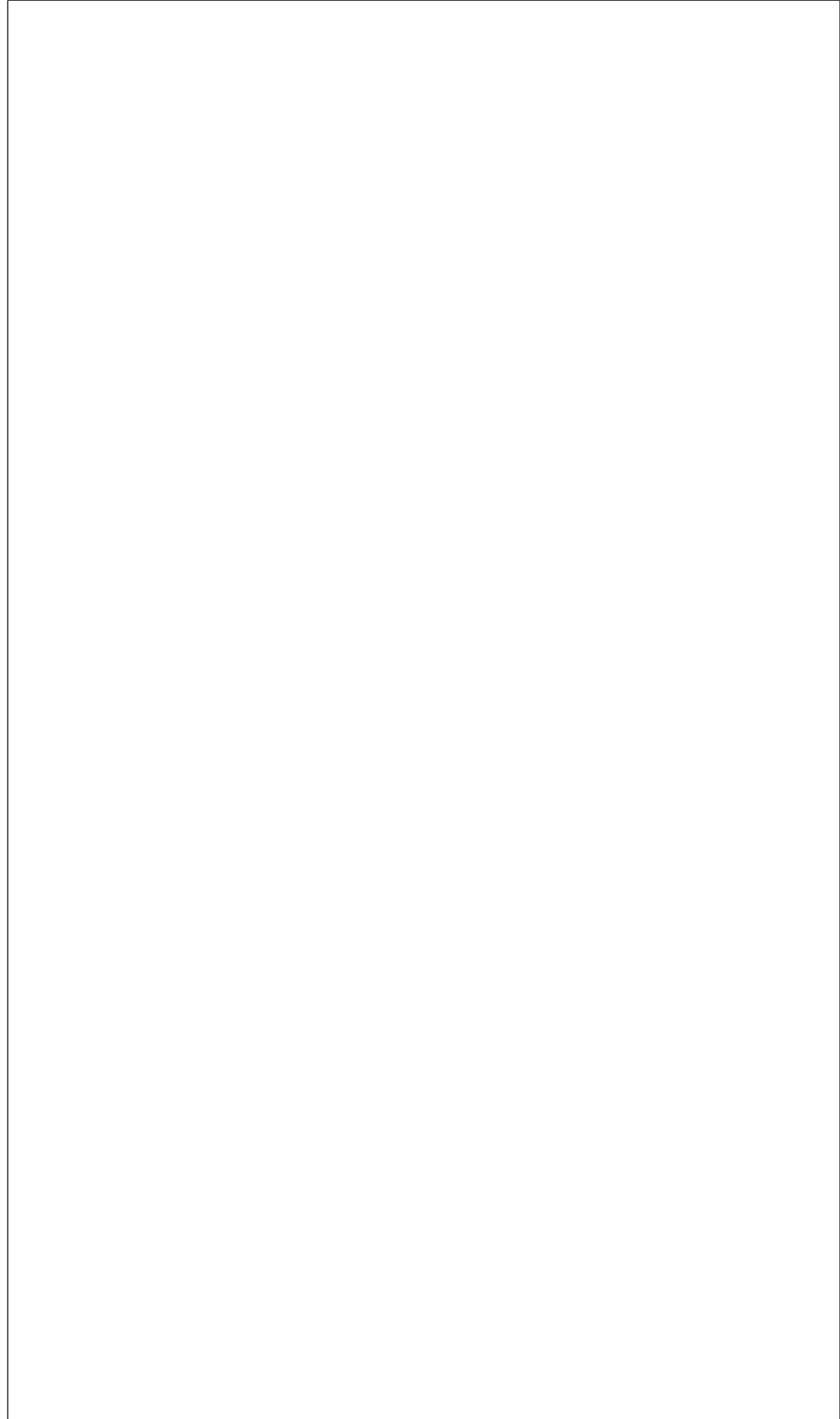
**update\_**

Up-  
 date  
 one  
 or  
 more  
 DHC  
 op-  
 tions  
 on  
 the  
 spec  
 i-  
 fied  
 port

**Parame**

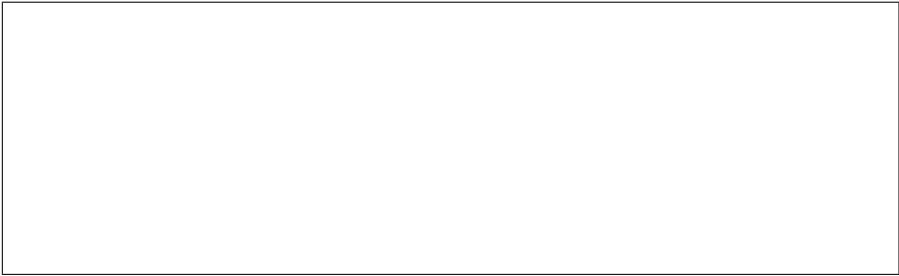
- **por**  
 des-  
 ig-  
 nate  
 whic  
 port  
 thes  
 at-  
 tribu  
 will  
 be  
 ap-  
 plied  
 to.
- **dho**

this  
will  
be  
a  
list  
of  
dicts  
e.g.



(continues on next page)

(continued from previous page)



- **token**  
An optional authentication token. Deprecates use of context

- **context**  
(optional) context object. Request requires context

**Raises**

- Failure
- ToUser
- data
- error
- HTTPError
- Timeout
- 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*  
*bas*  
Dep

In-  
ter-  
face  
for  
depl  
relat

ac-  
tions

**clean\_u**  
Clea  
up  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
for  
this  
node

**deploy**  
Per-  
form  
a  
de-  
ploy  
men  
to  
a  
node

**execute**  
Ex-  
e-  
cute  
a  
clear  
step

**Parame**

- tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

- 

**ste**  
a  
clea  
step  
dic-  
tio-  
nary  
to  
ex-  
e-  
cute

**Returns**  
Non

**get\_cle**  
Get  
the  
list  
of  
clea  
step  
from  
the  
file.

**Parame**  
**tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

**Returns**  
A  
list  
of  
clea  
step  
dic-  
tio-  
nar-  
ies

**get\_pro**  
Re-  
turn

terface. 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.

the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face  
 has\_dec  
 Whe  
 the  
 drive  
 sup-  
 port  
 de-  
 com  
 pose  
 de-  
 ploy  
 step  
 Pre-  
 vi-  
 ously  
 (sinc  
 Roc  
 drive  
 used  
 a  
 sin-  
 gle  
 de-  
 ploy  
 de-  
 ploy  
 step  
 on  
 the  
 de-  
 ploy  
 in-  
 prepare  
 Pre-  
 pare  
 the  
 de-  
 ploy  
 men

en-  
vi-  
ron-  
men-  
for  
this  
node

**prepare**  
Boo-  
into  
the  
ram-  
to  
pre-  
pare  
for  
clea-  
ing.

**Parame**  
**tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

**Raises**  
*No*  
if  
the  
pre-  
vi-  
ous  
clea-  
ing  
port-  
can-  
not  
be  
re-  
mov-  
or  
if  
new  
clea-  
ing

ports cannot be created

Returns

Non-  
or  
state  
for  
asyn-  
pre-  
pare

process

Start  
the  
next  
clear  
step  
if  
the  
pre-  
vi-  
ous  
one  
is  
com-  
plete

Parameters

- **task**  
a  
Task  
ager  
in-  
stan-
- **step**  
clear  
or  
de-  
ploy

take\_over

Take  
over  
man-  
age-  
men-  
of  
this  
task

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

tftpboot environment for the given node. When a conductor goes offline, another conductor must change

node  
from  
a  
dead  
con-  
duc-  
tor.  
  
If  
con-  
duc-  
tors  
host  
main-  
tain  
a  
stati-  
re-  
la-  
tion-  
ship  
to  
node  
this  
meth-  
shou-  
be  
im-

**For exam**

Neu-  
tron  
mus-  
for-  
ware  
DHCP  
BOO  
re-  
ques-  
to  
a  
con-  
duc-  
tor  
which  
has  
pre-  
pare  
the

this setting in Neutron as part of remapping that nodes control to itself. This is performed within the *takeover* method.

**Parameters**  
**task\_manager**  
 A TaskManager object containing the node to act on.

**tear\_down**  
 Tear down a previous deployment on the task node

**tear\_down**  
 A deployment step to tear down the agent  
 Shut down the machine and re-



mov
 it
 from
 the
 pro-
 vi-
 sion
 ing
 net-
 worl

**Parame**  
**tas**  
 a  
 Task  
 ager  
 ob-  
 ject  
 con-  
 tain-  
 ing  
 the  
 node

**tear\_d**  
 Clea  
 up  
 the  
 PXE  
 and  
 DHCP  
 files  
 af-  
 ter  
 clea  
 ing.

**Parame**  
**tas**  
 a  
 Task  
 ager  
 ob-  
 ject  
 con-  
 tain-  
 ing  
 the  
 node

**Raises**  
*No*  
 if

the  
 clear  
 ing  
 port  
 can-  
 not  
 be  
 re-  
 mov

**validat**  
 Val-  
 i-  
 date  
 the  
 drive  
 spec  
 Nod  
 de-  
 ploy  
 men  
 info

**write\_i**

**excepti**  
  
 Base  
 iro  
 exc  
 Iro

Module contents

ironic.drivers.modules.drac package

Submodules

ironic.drivers.modules.drac.bios module

DRA  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 spe-  
 cific  
 meth

ods

**class i**

Base

*irc*

*dri*

*mod*

*reco*

*bic*

*Reco*

iDR

Red

fish

in-

ter-

face

for

BIO

setti

relat

ac-

tions

Pres

this

class

en-

tirel

de-

fers

to

its

base

class

a

gene

veno

inde

Red

fish

in-

ter-

face

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

**class i**

Base

*irc*

*dri*

*base*  
*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**

- tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on
- set**  
 List  
 of  
 BIO

or None if it is completed.

set-  
 tings  
 to  
 ap-  
 ply  
**Raises**  
 DRA  
 C-  
 Op-  
 er-  
 a-  
 tionl  
 upon  
 an  
 er-  
 ror  
 from  
 pyth  
 drac  
**Returns**  
 state  
 (clea  
 ing)  
 or  
 state  
 (de-  
 ploy  
 men  
 if  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
**cache\_k**  
 Stor  
 or  
 up-  
 date  
 the  
 cur-  
 rent  
 BIO  
 set-

ting  
 for  
 the  
 node  
  
 Get  
 the  
 cur-  
 rent  
 BIO  
 set-  
 ting  
 and  
 store  
 them  
 in  
 the  
 bios  
 data  
 ta-  
 ble.

**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**factory**

Re-  
set  
the  
BIO  
set  
tings  
of  
the  
node  
to  
the  
fac-  
tory  
de-  
fault

This  
uses  
the  
Life  
cy-  
cle  
Con-  
troll  
con-  
fig-  
u-  
ra-  
tion  
to  
per-  
form  
BIO  
con-  
fig-  
u-

ration reset. Leveraging the python-dracclient methods already available.

**Parame**

**tas**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to

it is completed.

act  
on

**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  
prog  
asyn  
chro  
or  
Non  
if

**get\_pro**

Re-  
turn  
the  
prop  
er-  
ties  
of  
the  
BIO  
In-  
ter-  
face



Returns

dictionary of properties and values. The dictionary is a mapping from the name of the property to its value. The dictionary is a mapping from the name of the property to its value. The dictionary is a mapping from the name of the property to its value.

validate

Validate the data against the schema. The data is a dictionary of properties and values. The schema is a dictionary of properties and values. The data is a dictionary of properties and values. The schema is a dictionary of properties and values.

Parameter

The parameter is a dictionary of properties and values. The parameter is a dictionary of properties and values. The parameter is a dictionary of properties and values. The parameter is a dictionary of properties and values.

Raises

InvalidParameterError

the node or on invalid inputs

ram-  
e-  
ter-  
Valu  
if  
som  
man  
tory  
in-  
for-  
ma-  
tion  
is  
miss  
ing  
on

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  
on.

**Raises**  
Dra-  
c-  
Op-  
er-  
a-  
tion)

on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac  
  
 ironic.  
 Com  
 mits  
 penc  
 ing  
 char  
 adde  
 by  
 set\_

Paramet

- **task**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **reboot**  
 in-  
 di-  
 cate  
 whe  
 a  
 re-  
 boot  
 job  
 shou  
 be  
 au-  
 to-  
 mat-  
 i-

the config job.

Raises

Dra-  
c-  
Op-  
er-  
a-  
tionl  
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:

```
↪ 'Value',
```

```
↪ 'New Value', # could also be None
```

(continues on next page)

(continued from previous page)

```
↩ ['Value', 'New Value', 'None']},
```

```
↩value': 'Information',
```

(continues on next page)

(continued from previous page)

```
↪only': False,
```

```
↪length': 0,
```

(continues on next page)

```
↪length': 255,
```

(continued from previous page)

```
↪'regex': '^ [0-9A-Za-z]{0,255}$',
```

```
↪'current_value': 0,
```

(continues on next page)



(continued from previous page)

```
↪ 'read_only': True,
```

```
↪ 'lower_bound': 0,
```

(continues on next page)

```
↪ 'upper_bound': 65535}}}
```

(continued from previous page)

--

**Parameters**

**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

always be either an enumerated attribute, a string attribute, or an integer attribute. All attributes have the following parameters:

#### Parameters

integer or a string.

ex-  
pose  
via  
this  
API  
will

- **name**  
is  
the  
name  
of  
the  
BIO  
at-  
tribu
- **current**  
is  
the  
cur-  
rent  
valu  
of  
the  
at-  
tribu  
It  
will  
al-  
way  
be  
ei-  
ther  
an  
in-

- **pen**  
is  
the  
new  
valu  
that  
we

pending value.

will result in an error. The read-only flag can change depending on other attributes. A future version of this call may expose the dependencies that indicate when that may happen.

wan  
the  
at-  
tribu  
to  
have  
Non  
mea  
that  
there  
is  
no

•  
**rea**  
in-  
di-  
cate  
whe  
this  
at-  
tribu  
can  
be  
char  
Try-  
ing  
to  
char  
a  
read  
only  
valu

Enu  
mer-  
able  
at-  
tribu  
also  
have  
the  
fol-  
low-  
ing  
pa-  
ram-  
e-  
ters:

to.

Parameter

**pos**  
 is  
 an  
 ar-  
 ray  
 of  
 val-  
 ues  
 it  
 is  
 per-  
 mis-  
 si-  
 ble  
 to  
 set  
 the  
 at-  
 tribu

Strin  
 at-  
 tribu  
 also  
 have  
 the  
 fol-  
 low-  
 ing  
 pa-  
 ram-  
 e-  
 ters:

Parameter

- min**  
 is  
 the  
 min-  
 i-  
 mum  
 leng  
 of  
 the  
 strin
- max**

It may be None if the string is read only or if the string does not have to match any particular regular expression.

**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  
each  
of  
the  
val-  
ues  
pass  
in.

#### Paramet

- **tas**  
a  
Task  
ager

in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

- **kwa**  
 a  
 dic-  
 tio-  
 nary  
 of  
 { At-  
 tribu  
 Nam  
 New  
 Valu

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**Returns**

A  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 is\_c  
 key  
 with  
 a  
 bool



`commit_config()` needs to be called to make the changes, and the `is_reboot_required` key which has a value of `true` or `false`. This key is used to indicate to the `commit_config()` call if a reboot should be performed.

## ironic.drivers.modules.drac.boot module

**class** `i`

Base  
*irc*  
*dri*  
*mod*  
*reco*  
*boot*  
*Reco*

iDR  
Red  
fish  
in-  
ter-  
face  
for  
vir-  
tual  
me-  
dia  
boot  
relat  
ac-  
tions

Vir-  
tual  
Me-  
dia  
al-  
lows  
boot  
ing  
the  
sys-  
tem  
from  
vir-  
tual

image that BMC inserts into the drive.

tion) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

interface, which looks like this:

CD/  
drive  
con-  
tain-  
ing  
user

The  
CD/  
im-  
ages  
must  
be  
in  
ISO  
for-  
mat  
and  
(de-  
pend-  
ing  
on  
BMC  
im-  
ple-  
men-  
ta-

The  
base  
line  
boot  
work-  
flow  
is  
most  
base  
on  
the  
stan-  
dard  
Red  
fish  
vir-  
tual  
me-  
dia  
boot

1.

EFI boot loader) images

Swift temporary URL

Pull  
ker-  
nel,  
ram  
and  
ESP  
if  
UEFI  
boot  
is  
re-  
ques  
(FAT  
par-  
ti-  
tion  
im-  
age  
with

2.  
Cre-  
ate  
boot  
ISO  
out  
of  
im-  
ages  
(#1)  
push  
it  
to  
Glar  
and  
pass  
to  
the  
BM  
as

3.  
Op-  
tion-  
ally  
cre-  
ate  
flopp  
im-  
age

push it to Glance and pass to the BMC as Swift temporary URL

*cue\_kernel/rescue\_ramdisk* properties from *[instance\_info]* or *[driver\_info]*.

with  
de-  
sired  
sys-  
tem  
con-  
fig-  
u-  
ra-  
tion  
data

4.  
In-  
sert  
CD/  
and  
(op-  
tion-  
ally)  
flopp  
im-  
ages  
and  
set  
prop  
boot  
mod  
  
For  
buil  
ing  
de-  
ploy  
or  
res-  
cue  
ISO  
red-  
fish  
boot  
in-  
ter-  
face  
uses  
*de-  
ploy*  
or  
*res-*

For

in the Glance image metadata found in *[instance\_info]image\_source* node property.

to boot from a virtual media device - this is done via OEM action call implemented in Dell sushy OEM extension package.

build  
ing  
boot  
(use  
ISO  
red-  
fish  
boot  
in-  
ter-  
face  
seek  
ker-  
nel\_  
and  
ram  
prop  
er-  
ties  
  
iDR  
vir-  
tual  
me-  
dia  
boot  
in-  
ter-  
face  
only  
dif-  
fers  
by  
the  
way  
how  
it  
sets  
the  
node

ironic.drivers.modules.drac.common module

Com
 mon
 func
 tion-
 al-
 i-
 ties
 shar
 be-
 twee
 dif-
 fer-
 ent
 DRA
 mod
 ules

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-

node or on invalid input.

combination of both.

ram-  
e-  
ter-  
Valu  
if  
man  
tory  
in-  
for-  
ma-  
tion  
is  
miss  
ing  
on  
the

ironic.  
Pars  
a  
node  
drive  
val-  
ues.  
Pars  
the  
drive  
of  
the  
node  
read  
de-  
fault  
val-  
ues  
and  
re-  
turn  
a  
dict  
con-  
tain-  
ing  
the

**Paramet**  
**nod**  
an  
iron  
node

ob-  
ject.

**Returns**

a  
dict  
con-  
tain-  
ing  
in-  
for-  
ma-  
tion  
from  
drive  
and  
de-  
fault  
val-  
ues.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
man  
tory  
in-  
for-  
ma-  
tion  
is  
miss  
ing  
on

the node or on invalid inputs.



## ironic.drivers.modules.drac.inspect module

DRAC  
in-  
spec  
tion  
in-  
ter-  
face

**class** i

Base  
*ironic*  
*drivers*  
*modules*  
*drac*  
*inspect*  
*DracInspect*

Class  
alias  
of  
class  
DracInspect  
Main  
In-  
spec

This  
class  
pro-  
vide  
on-  
go-  
ing  
sup-  
port  
of  
the  
dep-  
re-  
cate  
idrac  
in-  
spec  
in-  
ter-  
face

implementation entrypoint.

All

That makes them available to both the deprecated idrac and new idrac-wsman endpoints. Such changes should not be made to this class.

bug  
 fixes  
 and  
 new  
 fea-  
 tures  
 shou  
 be  
 im-  
 ple-  
 men  
 in  
 its  
 base  
 class  
 Drac  
 Man  
 In-  
 spec

```

class i

Base
    irc
    dri
    mod
    rec
    ins
    Rec

iDR
Red
fish
in-
ter-
face
for
insp
relat
ac-
tions

Pres
this
class
en-
tirel
de-
fers
to
its

```

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

base  
class  
a  
gene  
vend  
inde  
Red  
fish  
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.  
 In-  
 spec-  
 hard-  
 ware  
 to  
 ob-  
 tain  
 the  
 es-  
 sen-  
 tial  
 &  
 ad-  
 di-  
 tion-  
 hard-  
 ware  
 prop-  
 er-  
 ties.

**Parameters**

**task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**

Har-  
 ware  
 spec-  
 tion-  
 Fail-  
 ure,  
 if  
 un-  
 able  
 to  
 get  
 es-

sen-  
 tial  
 hard  
 ware  
 prop  
 er-  
 ties.

Returns

state

validate

Val-  
 i-  
 date  
 the  
 drive  
 spec  
 info  
 sup-  
 plied

This  
 meth  
 val-  
 i-  
 date  
 whe  
 the  
 drive  
 prop  
 erty  
 of  
 the  
 sup-  
 plied  
 node  
 con-  
 tains  
 the  
 re-  
 quir

information for this driver to manage the node.

Parameters

task  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-

on the node.

**ironic.drivers.modules.drac.job module**

ing  
 the  
 node  
 to  
 act  
 on.  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 drive  
 at-  
 tribu  
 is  
 miss  
 ing  
 or  
 in-  
 valid  
 DRA  
 Life  
 cy-  
 cle  
 job  
 spe-  
 cific  
 meth  
 ods  
 ironic.  
 Get  
 the  
 de-  
 tails  
 of  
 a  
 Life  
 cy-

cle  
 job  
 of  
 the  
 node

**Parameters**

- **node**  
an ironic node object.
- **job\_id**  
ID of the Life cycle job.

**Returns**

a Job object from drac client

**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  
 clier

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

ironic.

Val-  
 i-  
 date  
 the  
 job  
 queu  
 on  
 the



node: an ironic node object. :param name\_prefix: A name prefix for jobs to validate. :raises: DracOperationError on an error from python-dracclient.

ironic.

Wait  
for  
job  
to  
com  
plete  
  
It  
will  
wait  
for  
the  
job  
to  
com  
plete  
for  
20  
min-  
utes  
and  
raise  
time  
out  
if  
job

complete within given interval of time. :param node: an ironi 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.

## **ironi.drivers.modules.drac.management module**

neve

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  
on-  
go-  
ing  
sup-  
port  
of  
the  
dep-  
re-  
cate  
idrac  
man

face implementation entrypoin

ment. That makes them available to both the deprecated idrac and new idrac-wsman entrypoin

age-  
men  
in-  
ter-

All  
bug  
fixes  
and  
new  
fea-  
tures  
shou  
be  
im-  
ple-  
men  
in  
its  
base  
class  
Drac  
Man  
Man  
age-

**class** i

Base  
*irc*  
*dri*  
*mod*  
*rec*  
*man*  
*Rec*  
  
iDR  
Red  
fish  
in-  
ter-  
face  
for  
man  
relat  
ac-  
tion  
  
Pres  
this

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

**class** *ironic*  
 Base class for the Ironic driver.  
 This class is the base class for all Ironic drivers.  
 It defines the basic methods and attributes that all drivers must implement.

**clear\_queue**  
 Clear the queue of jobs.  
 This method is used to clear the queue of jobs that are currently being processed.

**Parameter tas**  
 A Task object representing the task to be executed.  
 This parameter is used to pass the task object to the driver's execute method.  
 The task object contains information about the job, including the node ID, the action to be performed, and the timeout.  
 The driver's execute method should return the result of the job, which can be a string or a dictionary.

**Returns**

Non  
if  
it  
is  
com  
plete

#### **Raises**

Dra-  
c-  
Op-  
er-  
a-  
tionl  
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

#### **Parame**

**tas**  
a  
Task  
ager  
in-  
stan

con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**Returns**

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing:

**boot\_c**

the  
 boot  
 de-  
 vice  
 one  
 of  
*irc*  
*com*  
*boo*  
 or  
 Non  
 if  
 it  
 is  
 un-  
 know

**persist**

unknown.

when  
the  
boot  
de-  
vice  
will  
per-  
sist  
to  
all  
fu-  
ture  
boot  
or  
not,  
Non  
if  
it  
is

#### **get\_prop**

Re-  
turn  
the  
prop  
er-  
ties  
of  
the  
in-  
ter-  
face

#### **get\_sensors**

Get  
sen-  
sors  
data

#### **Parameters**

**task**  
a  
Task  
ager  
in-  
stan

#### **Raises**

Fail-  
To-  
Get-  
Sen-

can be processed by Ceilometer.

sor-  
Data  
whe  
get-  
ting  
the  
sen-  
sor  
data  
fails

Raises

Failu  
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  
sen-  
sor  
type  
whic

get\_sup

Get  
a  
list  
of  
the



sup-  
port  
boot  
de-  
vice

**Parameters**  
**task**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**Returns**  
A  
list  
with  
the  
sup-  
port  
boot  
de-  
vice  
de-  
fine  
in  
[ironic](#)  
[command](#)  
[book](#)

**known\_c**  
Re-  
set  
the  
iDR  
Clea  
the  
job  
queu

**Parameters**  
**task**  
a  
Task

ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**

Non  
 if  
 it  
 is  
 com  
 plete

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 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  
 on.

**Returns**

Non  
 if  
 it  
 is  
 com  
 plete

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**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**

not. Default: False.

Raises  
 In-  
 valid

Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valid  
boot  
de-  
vice  
is  
spec  
i-  
fied.

**validat**

Val-  
i-  
date  
the  
drive  
spec  
info  
sup-  
plied

This  
meth  
val-  
i-  
date  
when  
the  
drive  
prop  
erty  
of  
the  
sup-  
plied  
node  
con-  
tains  
the  
re-  
quir

information for this driver to manage the node.

**Parame**  
**tas**  
a

on the node.

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-  
tribu  
is  
miss  
ing  
or  
in-  
valid

`ironic.`

Set  
the  
boot  
de-  
vice  
for  
a  
node  
  
Set  
the  
boot  
de-  
vice  
to

use  
 on  
 next  
 boot  
 of  
 the  
 node

Paramet

- **node**  
 an  
 ironi  
 node  
 ob-  
 ject.
- **dev**  
 the  
 boot  
 de-  
 vice  
 one  
 of  
*irc*  
*com*  
*boo*
- **per**  
 Boo  
 valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 Fals  
 if

not. Default: False.

Raises

ironic.drivers.modules.drac.power module

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 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-



plementation entrypoint.

makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

**class** i

Base  
*irc*  
*dri*  
*mod*  
*rec*  
*pow*  
*Rea*  
iDR  
Red  
fish  
in-  
ter-  
face  
for  
pow  
relat  
ac-

Future resolution of Dell EMC- specific incompatibilities and introduction of vendor value added should be implemented by this class.

**class** `ironic`

Base class for the `ironic` driver.

Interface for power related actions.

**get\_power**

Return the power state of the node.

**Parameters**

tas  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 the  
 pow  
 state  
 one  
 of  
*irc*  
*com*  
*sta*

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 DRA  
 cre-  
 den-  
 tials  
 are  
 miss  
 ing.

**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

**reboot**  
 Per-  
 form  
 a  
 re-  
 boot  
 of  
 the  
 task  
 node

**Parame**

- tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- tim**  
 time  
 out  
 (in

sec-  
onds  
Un-  
sup-  
port  
by  
this  
in-  
ter-  
face

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
re-  
quir  
DRA  
cre-  
den-  
tials  
are  
miss  
ing.

**Raises**

Dra-  
c-  
Op-  
er-  
a-  
tionl  
on  
an  
er-  
ror  
from  
pyth  
drac

**set\_pow**

Set  
the  
pow  
state  
of  
the  
node

Parameters

- **taskManager**  
 a TaskManager instance containing the node to act on.
- **power**  
 a power state from *irc.com.sta*
- **timeout**  
 time out (in seconds) Un-supported by this interface

**Raises**  
 InvalidParameterException  
 ValueError

if  
 re-  
 quir  
 DRA  
 cre-  
 den-  
 tials  
 are  
 miss  
 ing.

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tion  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**validat**

Val-  
 i-  
 date  
 the  
 drive  
 spec  
 Nod  
 pow  
 info  
 This  
 meth  
 val-  
 i-  
 date  
 whe  
 the  
 drive  
 prop  
 erty  
 of  
 the  
 sup-  
 pliec  
 node  
 con-

information for this driver to manage the power state of the node.

on the node.

tains  
 the  
 re-  
 quir  
  
**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**Raises**  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 drive  
 at-  
 tribu  
 is  
 miss  
 ing  
 or  
 in-  
 valio



## ironic.drivers.modules.drac.raid module

DRA  
RAI  
spe-  
cific  
meth  
ods

**class** i  
Base  
*iro*  
*dri*  
*mod*  
*dra*  
*rai*  
*Dra*

Clas  
alias  
of  
class  
Drac  
Man  
RAI

This  
class  
pro-  
vide  
on-  
go-  
ing  
sup-  
port  
of  
the  
dep-  
re-  
cate  
idrac  
RAI  
in-  
ter-  
face  
im-

plementation entrypoint.

All  
bug  
fixes  
and

makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

**class** `IronicIdrac`  
 Base class for idrac and idrac-wsman.  
*irc*  
*dri*  
*bas*  
*RAI*

**apply\_config**  
 Applies configuration to the node.  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give  
 node

**Parameters**  

- task\_manager**  
 A TaskManager object to use for task execution.

ified in `raid_config`. Default value is `True`.

- **rai**  
The  
RAI  
con-  
fig-  
u-  
ra-  
tion  
to  
ap-  
ply.
- **cre**  
Set-  
ting  
this  
to  
Fals  
in-  
di-  
cate  
not  
to  
cre-  
ate  
root  
vol-  
ume  
that  
is  
spec
- **cre**  
Set-  
ting  
this  
to  
Fals  
in-  
di-  
cate  
not  
to  
cre-  
ate  
non-  
root  
vol-

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

ume  
(all  
ex-

- **del**  
Set-  
ting  
this  
to  
`True`  
in-  
di-  
cate  
to  
dele  
RAI  
con-  
fig-  
u-  
ra-  
tion  
prior  
to

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
RAI  
con-  
fig-  
u-  
ra-  
tion  
is  
in-  
valid

**Returns**  
state  
if  
RAI  
con-  
fig-

plete.

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-

erwise, no root volume is created. Default is True.

stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

- **cre**  
If  
True  
a  
root  
vol-  
ume  
is  
cre-  
ated  
dur-  
ing  
RAI  
con-  
fig-  
u-  
ra-  
tion.  
Oth-

- **cre**  
If  
True  
non-  
root  
vol-  
ume  
are  
cre-  
ated  
If  
Fals  
no  
non-  
root  
vol-  
ume  
are  
cre-

ated. Default is True.

creating the new configuration. Default is False.

it is completed.

- **del**  
Set-  
ting  
this  
to  
True  
in-  
di-  
cate  
to  
dele  
RAI  
con-  
fig-  
u-  
ra-  
tion  
prior  
to

#### Returns

state  
(clea  
ing)  
or  
state  
(de-  
ploy  
men  
if  
cre-  
ation  
is  
in  
prog  
asyn  
chro  
or  
Non  
if

#### Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-

Value  
 if  
 node  
 is  
 miss  
 ing  
 or  
 emp

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 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



it is completed.

ing)  
 or  
 state  
 (de-  
 ploy  
 men  
 if  
 dele  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if

Raises

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 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-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**get\_proc**

Re-  
 turn  
 the  
 prop-  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

ironic.  
 Dele  
 all  
 penc  
 ing

char  
 on  
 a  
 RAI  
 con-  
 troll

Paramet

- **nod**  
 an  
 iron  
 node  
 ob-  
 ject.
- **rai**  
 id  
 of  
 the  
 RAI  
 con-  
 troll

Raises

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

ironic.

Con  
 vert  
 disk  
 RAI  
 sta-  
 tus  
 This

this by only converting the disks that are not already in the correct state.

Paramet

- **nod**  
 an  
 iron  
 node  
 ob-  
 ject.
- **mod**  
 the  
 mod  
 to  
 char  
 the  
 disk  
 ei-  
 ther  
 to  
 RAI  
 or  
 JBO
- **con**  
 Dic-  
 tio-  
 nary  
 of

requested mode.

troller 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.

#### Returns

a dictionary containing:

- 

conversion results a dictionary that maps controller ids to the conversion results for that controller. The conversion results are a dict that contains:

#### Raises

`DRACError` C-Operation not supported on an error

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  
 key  
 with  
 a  
 bool  
 valu  
 in-  
 di-

cating whether a config job must be created for the values to be applied. - The is\_reboot\_required

key with a `RebootRequired` enumerated value indicating whether the server must be rebooted to clear foreign configuration.

#### Raises

Dra-  
c-  
Op-  
er-  
a-  
tion  
on  
an  
er-  
ror  
from  
pyth  
drac

`ironic.`

Ap-  
ply  
all  
penc  
ing  
char  
on  
a  
RAI  
con-  
troll

#### Paramet

- **nod**  
an  
iron  
node  
ob-  
ject.
- **rai**  
id  
of  
the  
RAI  
con-  
troll
-

the config job. (optional, defaults to False)

reb  
in-  
di-  
cate  
whe  
a  
re-  
boot  
job  
shou  
be  
au-  
to-  
mat-  
i-  
cally  
cre-  
ated  
with

- **rea**  
in-  
di-  
cate  
RAI  
con-  
troll  
sup-  
port  
re-  
al-  
time  
(op-  
tiona  
de-  
fault  
to  
Fals

**Returns**  
id  
of  
the  
cre-  
ated  
job

**Raises**  
Dra-  
c-  
Op-



er-  
a-  
tionl  
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  
ac-

configuration once the changes are applied by calling the `commit_config` method.

#### Parameters

- **node**  
an ironiC node object.
- **raid**  
id of the RAID controller.
- **physical**  
ids of the physical disk.
- **raid\_level**  
RAID level of the virtual disk.
- **size**  
size of the virtual disk.
- **disk\_name**  
name of the disk.

of  
 the  
 vir-  
 tual  
 disk  
 (op-  
 tion:

- **spa**  
 Nun  
 ber  
 of  
 span  
 in  
 vir-  
 tual  
 disk  
 (op-  
 tion:

- **spa**  
 Nun  
 ber  
 of  
 disk  
 per  
 span  
 (op-  
 tion:

**Returns**

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 com  
 mit\_  
 key  
 with  
 a  
 bool  
 valu  
 in-  
 di-  
 cat-  
 ing

whether a config job must be created for the values to be applied.

Raises

Dra-  
c-  
Op-  
er-  
a-  
tionl  
on  
an  
er-  
ror  
from  
pyth  
drac

ironic.

Dele  
a  
sin-  
gle  
vir-  
tual  
disk  
on  
a  
RAI  
con-  
troll

The  
dele  
vir-  
tual  
disk  
will  
be  
in  
pend  
ing  
state  
The  
DRA  
card  
will  
do  
the  
ac-  
tual  
con-

figuration once the changes are applied by calling the `commit_config` method.

Paramet

- **node**  
an  
iron  
node  
ob-  
ject.
- **virtual**  
id  
of  
the  
vir-  
tual  
disk

#### Returns

a  
dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
com  
mit\_  
key  
with  
a  
bool  
valu  
in-  
di-  
cat-  
ing

whether a config job must be created for the values to be applied.

#### Raises

Dra-  
c-  
Op-  
er-  
a-  
tion  
on  
an  
er-  
ror  
from

pyth
drac

ironic.
List
the
phys
i-
cal
disk
of
the
node

**Paramet**
**nod**
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

**Parameters**  
**node**  
 an  
 iron  
 node  
 ob-  
 ject.

**Returns**  
 a  
 list  
 of  
 RAI  
 Con  
 troll  
 ob-  
 jects  
 from  
 drac  
 client

**Raises**  
 Dra-  
 c-  
 Op-  
 er-  
 a-  
 tion  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

ironic.  
 List  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 set-

tributes are RAIDEnumerableAttribute, RAIDStringAttribute and RAIDIntegerAttribute objects.

tings  
**Parameter**  
**node**  
 an  
 iron  
 node  
 ob-  
 ject.  
**Returns**  
 a  
 dic-  
 tio-  
 nary  
 with  
 the  
 RAID  
 set-  
 tings  
 us-  
 ing  
 In-  
 stan-  
 ceID  
 as  
 the  
 key.  
 The  
 at-

**Raises**  
 DRA  
 C-  
 Op-  
 er-  
 a-  
 tion  
 on  
 er-  
 ror  
 re-  
 port  
 back  
 by  
 the  
 DRA  
 in-  
 ter-  
 face

ironic.



List  
 the  
 vir-  
 tual  
 disk  
 of  
 the  
 node

**Parameter**  
**node**  
 an  
 iron  
 node  
 ob-  
 ject.

**Returns**  
 a  
 list  
 of  
 Vir-  
 tuall  
 isk  
 ob-  
 jects  
 from  
 drac  
 clien

**Raises**  
 Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

ironic.

Sets  
 the  
 RAI  
 con-  
 fig-

ues to be applied, a config job must be created.

Paramet

- **node**  
 an  
 iron  
 node  
 ob-  
 ject.
- **con**  
 the  
 ID  
 of  
 the  
 RAI  
 con-  
 troll
- **set**  
 a  
 dic-  
 tio-  
 nary  
 con-

name of attribute and the value being the proposed value.

ing whether a config job must be created for the values to be applied. - The `is_reboot_required` key with a `RebootRequired` enumerated value indicating whether the server must be rebooted for the values to be applied. Possible values are `true` and `false`.

tain-  
ing  
the  
pro-  
pose  
val  
ues,  
with  
each  
key  
be-  
ing  
the

#### Returns

a  
dic-  
tio-  
nary  
con-  
tain-  
ing:  
-

The  
is\_c  
key  
with  
a  
bool  
valu  
in-  
di-  
cat-

#### Raises

DRA  
C-  
Op-  
er-  
a-  
tion  
on  
er-  
ror  
re-  
port  
back

by  
 the  
 DRA  
 in-  
 ter-  
 face

ironic.drivers.modules.drac.vendor\_passthru module

DRA  
 venco  
 pass  
 in-  
 ter-  
 face

class i

Base  
*irc*  
*dri*  
*mod*  
*rec*  
*ven*  
*Reo*

iDR  
 Red  
 fish  
 in-  
 ter-  
 face  
 for  
 ven-  
 dor\_

Use  
 the  
 Red  
 fish  
 im-  
 ple-  
 men  
 ta-  
 tion  
 for  
 ven-  
 dor  
 pass

class i

face implementation entrypoin

Base  
*irc*  
*dri*  
*moo*  
*dra*  
*ven*  
*Dra*  
 Clas  
 alias  
 of  
 class  
 Dra  
 Man  
 Ven-  
 dor-  
 Pass  
 This  
 class  
 pro-  
 vide  
 on-  
 go-  
 ing  
 sup-  
 port  
 of  
 the  
 dep-  
 re-  
 cate  
 idra  
 ven-  
 dor  
 pass  
 in-  
 ter-  
 All  
 bug  
 fixes  
 and  
 new  
 fea-  
 tures  
 shou  
 be  
 im-  
 ple-  
 men  
 in

Passthru. That makes them available to both the deprecated idrac and new idrac-wsman entrypoints. Such changes should not be made to this class.

its  
base  
class  
Dra  
Man  
Ven  
dor-  
  
**class** i  
  
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  
aban  
don  
a  
BIO  
con-  
fig-  
u-

through `set_bios_config()`.

ra-  
tion  
pre-  
vi-  
ousl  
sub-  
mit-  
ted

## 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

## 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.  
 sub-  
 mit-  
 ted  
 thro  
 set\_

Parame

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **reb**  
 in-  
 di-  
 cate  
 whe



the config job.

a  
 re-  
 boot  
 job  
 shou  
 be  
 au-  
 to-  
 mat-  
 i-  
 cally  
 cre-  
 ated  
 with

- **kwargs**  
 not  
 used

**Raises**

Dra-  
 c-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 pyth  
 drac

**Returns**

A  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 job  
 key  
 with  
 the  
 id  
 of  
 the  
 new

fig job, and the `reboot_required` key indicating whether the node needs to be rebooted to start the config job.

created  
 ated  
 con-  
  
 get\_bic  
 Get  
 the  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion.  
 This  
 meth  
 is  
 used  
 to  
 re-  
 triev  
 the  
 BIO  
 set-  
 tings  
 from  
 a  
 node

Paramete

- **task**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **kwargs**  
 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\_pro**

Re-  
turn  
the  
prop  
er-  
ties  
of  
the  
in-  
ter-  
face

**list\_un**

List  
un-  
fin-  
ishe  
con-  
fig  
jobs  
of  
the  
node

**Parame**

each dict representing a Job object.

• **task**  
 a Task object  
 containing information  
 about the task  
 to be performed.  
 The task object is  
 used to create a  
 task object.

• **kwargs**  
 not used.

**Returns**  
 a dictionary  
 containing  
 information  
 about the  
 unfilled  
 key; this  
 key points  
 to a list  
 of  
 dict  
 with

**Raises**  
 Dra-  
 c-  
 Op-  
 er-  
 a-  
 tion  
 on

an  
er-  
ror  
from  
pyth  
drac

#### set\_bic

Cha  
BIO  
set-  
tings

This  
meth  
is  
used  
to  
chan  
the  
BIO  
set-  
tings  
on  
a  
node

#### Parame

- **task**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **kwargs**  
a  
dic-  
tio-  
nary  
of  
{ At-

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.

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  
 with  
 a  
 Boo  
 valu  
 in-  
 di-  
 cat-  
 ing  
 whe  
  
**validat**  
 Val-  
 i-  
 date  
 the  
 drive  
 spec  
 info  
 sup-

information for this driver to manage the power state of the node.

**Parameters**

- **task**  
a TaskManager instance containing the node to act on.
- **kwargs**  
not used

**Raises**

InvalidParameterException

on the node.

Module contents

ironic.drivers.modules.ibmc package

Submodules

ironic.drivers.modules.ibmc.management module

ter-  
 Valu  
 if  
 re-  
 quir  
 drive  
 at-  
 tribu  
 is  
 miss  
 ing  
 or  
 in-  
 valid

iBM  
 Man  
 age-  
 men  
 In-  
 ter-  
 face

class i

Base  
 iro  
 dri  
 bas  
 Man

get\_boo

Get  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 for



a  
 node

**Parameters**  
**task**  
 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**  
 IBM  
 C-  
 Con  
 tion-  
 Error  
 whe  
 it  
 fails  
 to

con-  
nect  
to  
iBM

#### **Raises**

IBM  
CER  
ror  
whe  
iBM  
re-  
spor  
an  
er-  
ror  
in-  
for-  
ma-  
tion

#### **Returns**

a  
dic-  
tio-  
nary  
con-  
tain-  
ing:

#### **boot\_c**

the  
boot  
de-  
vice  
one  
of  
*irc*  
*com*  
*boo*  
or  
Non  
if  
it  
is  
un-  
know

#### **persist**

Boo  
valu  
or

disabled.

Non  
True  
if  
the  
boot  
de-  
vice  
per-  
sists  
Fals  
oth-  
er-  
wise  
Non  
if  
its

#### **get\_boot**

Get  
the  
cur-  
rent  
boot  
mod  
for  
a  
node  
  
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  
 mal-  
 form  
 pa-  
 ram-  
 e-  
 ter(s)

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 miss  
 ing  
 pa-  
 ram-  
 e-  
 ter(s)

**Raises**  
 IBM  
 C-  
 Con  
 tion-  
 Erro  
 whe  
 it  
 fails  
 to  
 con-  
 nect  
 to  
 iBM

**Raises**  
 IBM  
 CEr  
 ror  
 whe  
 iBM  
 re-  
 spon  
 an  
 er-  
 ror

in-  
for-  
ma-  
tion

**Returns**

The  
boot  
mod  
one  
of  
iro  
com  
boo  
or  
Non  
if  
it  
is  
un-  
know

**get\_prop**

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  
 Not  
 im-  
 ple-  
 men  
 for  
 this  
 drive

**Raises**

Notl  
 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-  
 valio  
 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**

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

**Returns**

A
 list
 with
 the
 sup-

cant be determined, empty list is returned.

port  
 boot  
 de-  
 vice  
 de-  
 finec  
 in  
*irc*  
*com*  
*boo*  
**get\_sup**  
 Get  
 a  
 list  
 of  
 the  
 sup-  
 port  
 boot  
 mod  
**Parame**  
**tas**  
 A  
 task  
 from  
 Task  
 ager  
**Returns**  
 A  
 list  
 with  
 the  
 sup-  
 port  
 boot  
 mod  
 de-  
 finec  
 in  
*irc*  
*com*  
*boo*  
 If  
 boot  
 mod  
 sup-  
 port  
**inject\_**



In-  
 ject  
 NM  
 Non  
 Mas  
 able  
 In-  
 ter-  
 rupt  
 In-  
 ject  
 NM  
 (Nor  
 Mas  
 able  
 In-  
 ter-  
 rupt  
 for  
 a  
 node  
 im-  
 me-  
 di-  
 ately

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 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**

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

**set\_boot**

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  
fu-  
ture  
boot  
Fals  
if

not. Default: False.

Raises

InvalidParameterException: Raised when the parameter is invalid.

Raises

MissingParameterException: Raised when a required parameter is missing.

Raises

IBMCloudException: Raised when an error occurs while connecting to the IBM Cloud service.

Raises

IBMCloudException: Raised when an error occurs while connecting to the IBM Cloud service.

when  
iBM  
re-  
spon-  
an  
er-  
ror  
in-  
for-  
ma-  
tion

#### set\_boot

Set  
the  
boot  
mod  
for  
a  
node

Set  
the  
boot  
mod  
to  
use  
on  
next  
re-  
boot  
of  
the  
node

#### Parameters

- **task**  
A  
task  
from  
Task  
ager
- **mod**  
The  
boot  
mod  
one  
of  
*irc*

com  
 boo

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

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

**validat**

Val-  
 i-  
 date  
 the  
 drive  
 in-  
 for-  
 ma-  
 tion  
 need  
 by  
 the  
 iBM  
 drive

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**

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)

**ironic.drivers.modules.ibm.mc.mappings module**

iBM  
and  
Iron  
con-  
stan  
map  
ping

**ironic.drivers.modules.ibm.mc.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

**Paramet**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**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**

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

**get\_proc**

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  
 act  
 on.  
 Not  
 used  
 by  
 this  
 driv

at the moment.

**Returns**

A list with the supported power state defined in *ironic-compute-sta*

**reboot**

Perform a hard re-boot of the task node

**Parameters**

- task**

A Task ager instance containing the node to act on.
- time**

Time to wait for

the  
node  
to  
be-  
com  
pow  
ered  
on.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
on  
mal-  
form  
pa-  
ram-  
e-  
ter(s)

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

**Raises**

IBM  
C-  
Con  
tion-  
Erro  
whe  
it

fails  
 to  
 con-  
 nect  
 to  
 iBM

**Raises**

IBM  
 CEr  
 ror  
 whe  
 iBM  
 re-  
 spon  
 an  
 er-  
 ror  
 in-  
 for-  
 ma-  
 tion

**set\_pow**

Set  
 the  
 pow  
 state  
 of  
 the  
 task  
 node

**Parame**

- **tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **pow**

Any  
pow  
state  
from  
*irc*  
*com*  
*sta*

- **tim**  
Time  
to  
wait  
for  
the  
node  
to  
reac  
the  
re-  
ques  
state

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
on  
mal-  
form  
pa-  
ram-  
e-  
ter(s)

**Raises**  
Mis-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-

e-  
 ter  
 is  
 miss  
 ing.

**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

**validat**

Val-  
 i-  
 date  
 the  
 drive  
 in-  
 for-  
 ma-  
 tion  
 need  
 by  
 the  
 iBM  
 drive



Param

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

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)

## ironic.drivers.modules.ibm.mc.raid module

iBM  
RAID  
con-  
fig-  
u-  
ra-  
tion  
spe-  
cific  
meth  
ods

**class** `i`  
Base  
*irc*  
*dri*  
*bas*  
*RAI*  
Im-  
ple-  
men  
ta-  
tion  
of  
RAID  
In-  
ter-  
face  
for  
iBM

**RAID\_AE**

**apply\_c**  
Ap-  
plies  
RAID  
con-  
fig-  
u-  
ra-  
tion  
on  
the  
give  
node

Parameters

- task\_manager\_instance\_name**  
 A Task Manager instance name.
- raid\_config\_apply**  
 The RAID configuration to apply.
- create\_root\_volume**  
 Set this to False to indicate not to create root volume that is specified in raid\_config. Default value is True.
- create\_root\_volume**  
 Set this to False to indicate not to create root volume that is specified in raid\_config. Default value is True.

ified in raid\_config. Default value is True.

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

in-  
di-  
cate  
not  
to  
cre-  
ate  
non-  
root  
vol-  
ume  
(all  
ex-

- **del**  
Set-  
ting  
this  
to  
`True`  
in-  
di-  
cate  
to  
dele  
RAI  
con-  
fig-  
u-  
ra-  
tion  
prior  
to

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
RAI  
con-  
fig-  
u-  
ra-

tion  
 is  
 in-  
 valid

Returns

state  
 if  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if  
 it  
 is  
 com

plete.

create\_

Cre-  
 ate  
 a  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.  
 This  
 meth  
 cre-  
 ates  
 a  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give

otherwise, no root volume is created. Default is True.

node

**Parameters:**

- task\_agent\_instantiate**  
 a TaskAgent instance
- create\_root\_volume**  
 If True, a root volume is created during RAI configuration. Otherwise,
- create\_non\_root\_volumes**  
 If True, non-root volumes are created. If False, no non-root volumes

ated. Default is True.

creating the new configuration. Default is False.

volume and/or non-root volumes.

are  
cre-

- **del**  
Set-  
ting  
this  
to  
True  
in-  
di-  
cate  
to  
dele  
RAI  
con-  
fig-  
u-  
ra-  
tion  
prior  
to

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
node  
is  
miss  
ing  
or  
emp  
af-  
ter  
skip  
ping  
root

**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  
 on.

**Returns**  
 state  
 if  
 clear  
 ing  
 op-  
 er-  
 a-  
 tion  
 in  
 prog  
 asyn  
 chro  
 or  
 state  
 if  
 de-  
 ploy  
 op-



ation in progress synchronously or None if it is completed.

er-

#### **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  
 clien  
 ex-  
 cep-  
 tion.

Dec  
 o-  
 rated  
 func  
 tions  
 mus  
 take  
 a  
 Tas  
 as  
 the  
 first  
 pa-  
 ram-  
 e-  
 ter.

ironic.  
 Pars  
 the  
 in-  
 for-  
 ma-  
 tion  
 re-  
 quir  
 for  
 Iron  
 to

con-  
nect  
to  
iBM

**Parameters**  
**node**  
an  
Iron  
node  
ob-  
ject

**Returns**  
dic-  
tio-  
nary  
of  
pa-  
ram-  
e-  
ters

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter  
Value  
on  
mal-  
form  
pa-  
ram-  
e-  
ter(s)

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter  
Value  
on  
miss  
ing  
pa-  
ram-  
e-

ironic.drivers.modules.ibm.vend

	<div> <div>             ter(s </div> <div>             ironic. </div> </div>
	<div> <div>             iBM </div> <div>             Ven- </div> <div>             dor </div> <div>             In- </div> <div>             ter- </div> <div>             face </div> </div>
	<div> <div> <b>class</b> i </div> <div>             Base </div> <div> <i>irc</i> </div> <div> <i>dri</i> </div> <div> <i>bas</i> </div> <div> <i>Ven</i> </div> </div>
	<div> <div> <b>boot_up</b> </div> <div>             List </div> <div>             boot </div> <div>             type </div> <div>             or- </div> <div>             der </div> <div>             of </div> <div>             the </div> <div>             node </div> </div>
	<div> <div> <b>Param</b> </div> <div> </div> </div>
	<div> <div> <ul style="list-style-type: none"> <li> <div> <div> <b>tas</b> </div> <div>                     A </div> <div>                     Task </div> <div>                     ager </div> <div>                     in- </div> <div>                     stan </div> <div>                     con- </div> <div>                     tain- </div> <div>                     ing </div> <div>                     the </div> <div>                     node </div> <div>                     to </div> <div>                     act </div> <div>                     on. </div> </div> </li></ul></div> </div>

**kwa**  
Not  
used

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
kwa  
does  
not  
con-  
tain  
meth

**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-  
 ma-  
 tion

Returns

A  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 node  
 boot  
 up  
 se-  
 quer  
 in  
 as-  
 cend  
 ing  
 or-  
 der.

get\_proc

Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

Returns

dic-  
 tio-  
 nary  
 of  
 <proc  
 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-  
 Error  
 whe  
 it  
 fails  
 to  
 con-  
 nect  
 to  
 iBM

**Raises**  
 IBM  
 CER

summary of node.

ror  
 whe  
 iBM  
 re-  
 spor  
 an  
 er-  
 ror  
 in-  
 for-  
 ma-  
 tion

**Returns**  
 A  
 list  
 of  
 dic-  
 tio-  
 nar-  
 ies,  
 ev-  
 ery  
 dic-  
 tio-  
 nary  
 rep-  
 re-  
 sent  
 a  
 RAI  
 con-  
 troll

**validat**  
 Val-  
 i-  
 date  
 venc  
 spec  
 ac-  
 tions  
  
 If  
 in-  
 valid  
 raise  
 an  
 ex-  
 cep-  
 tion.  
 oth-



er-  
wise  
re-  
turn  
Non

Param

- **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  
sup-  
port  
in-  
ter-

faces.

Module contents

ironic.drivers.modules.ilo package

Submodules

ironic.drivers.modules.ilo.bios module

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
kwa  
does  
not  
con-  
tain  
meth

Raises

Miss  
ing-  
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

Paramet

- **task**  
 a  
 Task  
 ager  
 in-  
 stan
- **settings**  
 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  
 clea  
 step

**Raises**

In-  
stan-  
dard  
ploy-  
Fail-  
ure,  
on  
fail-  
ure  
to  
ex-  
e-  
cute  
of  
de-  
ploy  
step

**cache\_k**

Stor-  
the  
BIO-  
set-  
ting  
in  
the  
data

**Parame**

**tas**  
a  
Task  
ager  
in-  
stan-

**Raises**

Nod-  
Clea-  
ing-  
Fail-  
ure,  
on  
fail-  
ure  
to  
ex-  
e-  
cute  
of  
clea-  
step

Raises

In-  
stan-  
dard  
ploy-  
Fail-  
ure,  
on  
fail-  
ure  
to  
ex-  
e-  
cute  
of  
de-  
ploy  
step

factory

Re-  
set  
the  
BIO  
set-  
ting  
to  
fac-  
tory  
con-  
fig-  
u-  
ra-  
tion.

Parame

tas  
a  
Task  
ager  
in-  
stan

Raises

Nod  
Clea  
ing-  
Fail-  
ure,  
on  
fail-  
ure  
to  
ex-

e-  
cute  
of  
clea  
step

**Raises**

In-  
stan  
ploy  
Fail-  
ure,  
on  
fail-  
ure  
to  
ex-  
e-  
cute  
of  
de-  
ploy  
step

**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**

Che

dentials information.

that  
 drive  
 con-  
 tains  
 re-  
 quir  
 ILO  
 cre-  
 den-  
 tials  
 Val-  
 i-  
 date  
 whe  
 the  
 drive  
 prop  
 erty  
 of  
 the  
 sup-  
 plies  
 task  
 node  
 con-  
 tains  
 the  
 re-  
 quir  
 cre-

**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 In-  
 valid  
 Pa-  
 ram  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 iLO

pa-  
 ram-  
 e-  
 ters  
 are  
 not  
 valid

**Raises**

Miss-  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

ironic.drivers.modules.ilo.boot module

Boo  
 In-  
 ter-  
 face  
 for  
 iLO  
 drive  
 and  
 its  
 sup-  
 port  
 ing  
 meth  
 ods.

**class** i  
 Base  
*irc*  
*dri*  
*mod*



*pxe*  
*PXE*

#### **clean\_up**

Clea  
up  
the  
boot  
of  
in-  
stan  
  
This  
meth  
clea  
up  
the  
PXE  
en-  
vi-  
ron-  
men  
that  
was  
setu  
for  
boot  
ing  
the  
in-  
stan  
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.

#### **Param**

**tas**  
a  
task  
from  
Task  
ager

#### **Returns**

Non

#### **Raises**

Ilo-  
Op-  
er-  
a-  
tionl  
if

tion from the nodes instance\_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config. In case of boot from volume, it updates the iSCSI info onto iLO and sets the node to boot from UefiTarget boot device.

som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

**prepare**  
Pre-  
pare  
the  
boot  
of  
in-  
stan

This  
meth  
pre-  
pare  
the  
boot  
of  
the  
in-  
stan  
af-  
ter  
read  
ing  
rel-  
e-  
vant  
in-  
for-  
ma-

**Parame**  
**tas**  
a  
task  
from  
Task  
ager

**Returns**  
Non

**Raises**

Ilo-  
Op-  
er-  
a-  
tion  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

#### 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  
af-  
ter  
read  
ing  
rel-  
e-

vant information from the nodes `driver_info` and `instance_info`.

#### Parame

•

or instance\_info.

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  
miss  
ing  
in  
node  
drive

Raises  
In-  
valid  
Pa-  
ram-  
e-  
ter-

the node.

Valu
 if
 som
 in-
 for-
 ma
 tion
 pro-
 vide
 is
 in-
 valid

**Raises**
 Iron
 icEx
 cep-
 tion.
 if
 som
 pow
 or
 set
 boot
 boot
 de-
 vice
 op-
 er-
 a-
 tion
 faile
 on

**Raises**
 Ilo-
 Op-
 er-
 a-
 tionl
 if
 som
 op-
 er-
 a-
 tion
 on
 iLO
 faile

**class** i

Base
   
[irc](#)
  
[dri](#)
  
[bas](#)
  
[Boo](#)

**capabil**

**clean\_u**

Clea
   
 up
   
 the
   
 boot
   
 of
   
 in-
   
 stan

This
   
 meth
   
 clea
   
 up
   
 the
   
 en-
   
 vi-
   
 ron-
   
 men
   
 that
   
 was
   
 setu
   
 for
   
 boot
   
 ing
   
 the
   
 in-
   
 stan

**Parame**

**tas**
  
 A
   
 task
   
 from
   
 Task
   
 ager

**Returns**

Non

**clean\_u**

Clea
   
 up
   
 the
   
 boot

of  
 iron  
 ram  
 This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that  
 was  
 setu  
 for  
 boot  
 ing  
 the  
 de-  
 ploy  
 ram

**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

This  
meth  
pre-  
pare  
the  
boot  
of  
the  
in-  
stan  
af-  
ter  
read  
ing  
rel-  
e-  
vant  
in-  
for-  
ma-

tion from the nodes instance\_info. It does the following depending on boot\_option for deploy:

- If  
the  
boot  
re-  
ques  
for  
this  
de-  
ploy  
is  
lo-  
cal  
or



image, then it sets the node to boot from disk.

sets the URL as the boot ISO to boot the instance image.

im-  
age  
is  
a  
who  
disk

- Oth-  
er-  
wise  
it  
finds  
the  
boot  
ISO  
sets  
the  
node  
boot  
op-  
tion  
to  
UE-  
FI-  
HTT  
and

**Parame**  
**tas**  
a  
task  
from  
Task  
ager

**Returns**  
Non

**Raises**  
Ilo-  
Op-  
er-  
a-  
tion  
if  
som  
op-  
er-  
a-  
tion

on  
iLO  
failed  
**Raises**  
In-  
stan-  
ploy  
Fail-  
ure,  
if  
its  
try  
to  
boot  
iSCS  
vol-  
ume  
in  
BIO  
boot  
mod

**prepare**  
Pre-  
pare  
the  
boot  
of  
de-  
ploy  
ram  
us-  
ing  
UEFI  
HTT  
boot  
This  
meth  
pre-  
pare  
the  
boot  
of  
the  
de-  
ploy  
or  
res-  
cue  
ram

vant information from the nodes `driver_info` and `instance_info`.

## Parameters

- **task**  
a task from TaskManager
- **parameters**  
the parameters to be passed to the ram

## Returns

None

## Raises

MissingParameterError  
ValueError  
if some information is missing in

or instance\_info.

the node.

node  
 drive  
 Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 pro-  
 vide  
 is  
 in-  
 valid  
 Raises  
 Iron  
 icEx  
 cep-  
 tion,  
 if  
 som  
 pow  
 or  
 set  
 boot  
 boot  
 de-  
 vice  
 op-  
 er-  
 a-  
 tion  
 faile  
 on  
 Raises  
 Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 if

some  
 op-  
 er-  
 a-  
 tion  
 on  
 iLO  
 failed

**validation**

Val-  
 i-  
 date  
 the  
 de-  
 ploy  
 men  
 in-  
 for-  
 ma-  
 tion  
 for  
 the  
 task  
 node

This  
 meth  
 val-  
 i-  
 date  
 whe  
 the  
 drive  
 and/  
 in-  
 stan  
 prop  
 er-  
 ties  
 of  
 the  
 task  
 node  
 con-  
 tains

the required information for this interface to function.

**Parameters**

**task**  
 A  
 Task  
 agent

in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**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)

**validat**

Val-  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quir

prop  
er-  
ties  
for  
in-  
spec  
tion.

**Parameters**  
**task**  
 A  
Task  
ager  
in-  
stan-  
with  
the  
node  
be-  
ing  
check

**Raises**  
 Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
node  
is  
miss  
ing  
one  
or  
more  
re-  
quir  
pa-  
ram-

**Raises**  
 Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion

eters

**validation**  
 Validation  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quire  
 prop  
 er-  
 ties  
 for  
 res-  
 cue.

**Parameter**  
**task**  
 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  
 re-  
 quire  
 pa-  
 ram-

eters



**class i**

Base  
*irc*  
*dri*  
*bas*  
*Boc*

**capabil**

**clean\_u**

Clea  
up  
the  
boot  
of  
in-  
stan  
  
This  
meth  
clea  
up  
the  
en-  
vi-  
ron-  
men  
that  
was  
setu  
for  
boot  
ing  
the  
in-  
stan  
It  
ejec

virtual media. In case of UEFI iSCSI booting, it cleans up iSCSI target information from the node.

**Parame**

**tas**  
a  
task  
from  
Task  
ager

**Returns**

Non

**Raises**

Ilo-  
Op-  
er-  
a-  
tionl  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

**clean\_up**

Clea  
up  
the  
boot  
of  
iron  
rame  
  
This  
meth  
clea  
up  
vir-  
tual  
me-  
dia  
de-  
vice  
setu  
for  
the  
de-  
ploy  
or  
res-  
cue  
rame

**Parame**

**tas**  
a  
task  
from  
Task  
ager

**Returns**

Non

**Raises**

Ilo-  
Op-  
er-  
a-  
tionl  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

**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-

tion from the nodes instance\_info. It does the following depending on boot\_option for deploy:

get info and node to boot from UefiTarget boot device.

image is a whole disk image, then it sets the node to boot from disk.

ISO to the bare metal and then sets the node to boot from CDROM.

and  
the  
boot  
re-  
ques  
for  
this  
de-  
ploy  
is  
lo-  
cal  
or

- Oth-  
er-  
wise  
it  
finds  
the  
boot  
ISO  
to  
boot  
the  
in-  
stan-  
im-  
age,  
at-  
tach  
the  
boot

**Parame**  
**tas**  
a  
task  
from  
Task  
ager

**Returns**  
Non

**Raises**  
Ilo-  
Op-  
er-  
a-

tion  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

**Raises**

In-  
stan  
ploy  
Fail-  
ure,  
if  
its  
try  
to  
boot  
iSCS  
vol-  
ume  
in  
BIO  
boot  
mod

**prepare**

Pre-  
pare  
the  
boot  
of  
de-  
ploy  
rame  
us-  
ing  
vir-  
tual  
me-  
dia.  
  
This  
meth  
pre-  
pare  
the  
boot

vant information from the nodes `driver_info` and `instance_info`.

## Parameters

- **task**  
a task from TaskManager
- **parameters**  
the parameters to be passed to the ram

## Returns

None

## Raises

MissingParameterError  
ValueError  
if some

or instance\_info.

in-  
for-  
ma-  
tion  
is  
miss  
ing  
in  
node  
drive

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
in-  
for-  
ma-  
tion  
pro-  
vide  
is  
in-  
valid

Raises

Iron  
icEx  
cep-  
tion,  
if  
som  
pow  
or  
set  
boot  
boot  
de-  
vice  
op-  
er-  
a-  
tion  
faile  
on



the node.

**Raises**

Ilo-  
Op-  
er-  
a-  
tionl  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

**validat**

Val-  
i-  
date  
the  
de-  
ploy  
men  
in-  
for-  
ma-  
tion  
for  
the  
task  
node

**Parame**

**tas**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**Raises**

In-

age or kernel and ramdisk not provided in instance\_info for non-Glance image.

valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
in-  
for-  
ma-  
tion  
is  
in-  
valid

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
ker-  
nel\_  
and  
ram  
are  
miss  
ing  
in  
the  
Glar  
im-

#### **validat**

Val-  
i-  
date  
that  
the  
node  
has  
re-  
quir  
prop  
er-  
ties  
for

in-  
spec-  
tion.

**Parameters**

**task**  
A  
Task  
ager  
in-  
stan-  
with  
the  
node  
be-  
ing  
check

**Raises**

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu-  
if  
node  
is  
miss-  
ing  
one  
or  
more  
re-  
quir-  
pa-  
ram-

eters

**Raises**

Un-  
sup-  
port-  
ed-  
Drive  
ten-  
sion

**validation**

Val-  
i-  
date

eters

that  
 the  
 node  
 has  
 re-  
 quir  
 prop  
 er-  
 ties  
 for  
 res-  
 cue.

Param

**task**  
 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  
 mor  
 re-  
 quir  
 pa-  
 ram-

**class** i  
 Base  
*irc*  
*dri*

moc
 ipx
 iPX

**clean\_u**

Clea
 up
 the
 boot
 of
 in-
 stan
 This
 meth
 clea
 up
 the
 PXE
 en-
 vi-
 ron-
 men
 that
 was
 setu
 for
 boot
 ing
 the
 in-
 stan
 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

**Raises**

Ilo-
 Op-
 er-
 a-
 tionl

tion from the nodes instance\_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config. In case of boot from volume, it updates the iSCSI info onto iLO and sets the node to boot from UefiTarget boot device.

if  
 som  
 op-  
 er-  
 a-  
 tion  
 on  
 iLO  
 faile  
  
**prepare**  
 Pre-  
 pare  
 the  
 boot  
 of  
 in-  
 stan  
  
 This  
 meth  
 pre-  
 pare  
 the  
 boot  
 of  
 the  
 in-  
 stan  
 af-  
 ter  
 read  
 ing  
 rel-  
 e-  
 vant  
 in-  
 for-  
 ma-  
  
**Param**  
**tas**  
 a  
 task  
 from  
 Task  
 ager  
  
**Returns**  
 Non

vant information from the nodes driver\_info and instance\_info.

Raises

Ilo-  
Op-  
er-  
a-  
tion  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO  
faile

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  
af-  
ter  
read  
ing  
rel-  
e-

Parame

or instance\_info.

- **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  
 miss  
 ing  
 in  
 node  
 drive

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-



the node.

ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 pro-  
 vide  
 is  
 in-  
 valid

Raises

Iron  
 icEx  
 cep-  
 tion.  
 if  
 som  
 pow  
 or  
 set  
 boot  
 boot  
 de-  
 vice  
 op-  
 er-  
 a-  
 tion  
 faile  
 on

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tion  
 if  
 som  
 op-  
 er-  
 a-  
 tion  
 on  
 iLO  
 faile

ironic.

Dis-  
 able  
 se-  
 cure  
 boot  
 on  
 node  
 does  
 not  
 thro  
 if  
 its  
 not  
 sup-  
 port

Paramet

tas  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 som  
 op-  
 er-  
 a-  
 tion  
 on  
 iLO  
 faile

ironic.

Gets  
 the  
 drive

information for this driver to deploy images to the node.

#### Parameters

- **node**  
a  
sin-  
gle  
Node
- **model**  
La-  
bel  
in-  
di-  
cat-  
ing  
a  
de-  
ploy  
or  
res-

carried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation is being carried out.

ing.

Returns

A dict with the drive values.

Raises

MissingParameterError: Value if any of the require parameters are missing.

ironic.

Common preparatory steps for all iLO drives. This



## **ironic.drivers.modules.ilo.common module**

faile

Com  
mon  
func  
tion-  
al-  
i-  
ties  
shar  
be-  
twee  
dif-  
fer-  
ent  
iLO  
mod  
ules

ironic.  
Nod  
is  
in  
Fin-  
ishe  
Post  
post  
state

ironic.  
Nod  
is  
in  
In-  
Post  
Dis-  
cov-  
eryC  
post  
state

ironic.  
Nod  
is  
in  
In-  
Post  
post  
state

ironic.  
Nod  
is  
in  
Null  
post  
state

ironic.  
Nod  
is  
in  
Pow  
post  
state

ironic.  
Nod  
is  
in  
Re-  
set  
post  
state

ironic.  
Nod  
is  
in  
Un-  
know  
post  
state

ironic.  
Nod  
sup-  
port  
both  
lega  
BIO  
and  
UEFI  
boot  
mod

ironic.  
Nod  
sup-  
port  
only  
lega  
BIO  
boot

mod
 ironic.
 Node
 support
 port
 only
 UEFI
 boot
 mod
 ironic.
 Attach
 the
 give
 url
 as
 virtual
 media
 on
 the
 node

Paramet

- **node**
 an
 ironic
 node
 ob-
 ject.
- **dev**
 the
 vir-
 tual
 me-
 dia
 de-
 vice
 to
 at-
 tach
- **url**
 the
 http



url  
to  
at-  
tach  
as  
the  
vir-  
tual  
me-  
dia  
de-  
vice

**Raises**  
Ilo-  
Op-  
er-  
a-  
tionl  
if  
in-  
sert  
vir-  
tual  
me-  
dia  
faile

`ironic.`  
Clea  
a  
node  
af-  
ter  
a  
vir-  
tual  
me-  
dia  
boot  
This  
meth  
clea  
up  
a  
node  
af-  
ter  
a  
vir-  
tual  
me-

if it exists in `CONF.ilo.swift_ilo_container` or web server. It also ejects both virtual media cdrom and virtual media floppy.

Paramet

tas

a

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

ironic.

Up-

load

the

give

im-

age

to

swif

This

meth

copi

the

give

im-

age

to

swif

Paramet

•

sou

The  
 ab-  
 so-  
 lute  
 path  
 of  
 the  
 im-  
 age  
 file  
 which  
 need  
 to  
 be  
 copi  
 to  
 swif

- **des**  
 The  
 nam  
 of  
 the  
 ob-  
 ject  
 that  
 will  
 con-  
 tain  
 the  
 copi  
 im-  
 age.

**Raises**

Swi  
 Op-  
 er-  
 a-  
 tionl  
 if  
 any  
 op-  
 er-  
 a-  
 tion  
 with  
 Swi  
 fails

**Returns**

temp

url  
 from  
 swift  
 af-  
 ter  
 the  
 sour  
 im-  
 age  
 is  
 up-  
 load  
 ironic.

Cop  
 the  
 give  
 im-  
 age  
 to  
 the  
 http  
 web  
 serv  
 This  
 meth  
 copi  
 the  
 give  
 im-  
 age  
 to  
 the  
 http.  
 lo-  
 ca-  
 tion.  
 It  
 en-  
 able  
 read  
 writ  
 ac-  
 cess

to the image else the deploy fails as the image file at the web\_server url is inaccessible.

Paramet  
 •

server root.

**source**  
The  
ab-  
so-  
lute  
path  
of  
the  
im-  
age  
file  
which  
needs  
to  
be  
copied  
to  
the  
web

- **destination**  
The  
name  
of  
the  
file  
that  
will  
con-  
tain  
the  
copied  
im-  
age.

**Raises**  
ImageU-  
pload-  
Failure  
ex-  
cep-  
tion  
if  
copy-  
ing  
the  
source  
file  
to

the  
 web  
 serv  
 fails

**Returns**

im-  
 age  
 url  
 af-  
 ter  
 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  
 an  
 iron  
 node  
 ob-  
 ject.

ironic.  
 Ejec  
 vir-

tual  
 me-  
 dia  
 de-  
 vice  
 This  
 meth  
 ejec  
 vir-  
 tual  
 me-  
 dia  
 flopp  
 and  
 cdro

Paramet

tas  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

Returns

Non

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 som  
 er-  
 ror  
 was  
 en-  
 cour  
 terec  
 whil  
 try-  
 ing

tual media floppy or cdrom.

to  
 ejec  
 vir-  
  
 ironic  
 Get  
 the  
 cur-  
 rent  
 boot  
 mod  
 for  
 a  
 node  
  
**Paramet**  
**nod**  
 an  
 iron  
 node  
 ob-  
 ject.  
  
**Raises**  
 Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 faile  
 to  
 fetch  
 boot  
 mod  
  
**Raises**  
 Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 Sup-  
 port  
 if  
 node  
 does  
 not  
 sup-  
 port  
 get-  
 ting



tions on the iLO.

peno  
 ing  
 boot  
 mod

ironi
 Gets  
 an  
 Ilo-  
 Clie  
 ob-  
 ject  
 from  
 pro-  
 liant  
 tils  
 li-  
 brary  
 Give  
 an  
 iron  
 node  
 ob-  
 ject,  
 this  
 meth  
 give  
 back  
 a  
 Ilo-  
 Clie  
 ob-  
 ject  
 to  
 do  
 op-  
 er-  
 a-

Paramet
 nod  
 an  
 iron  
 node  
 ob-  
 ject.

Returns
 an  
 Ilo-  
 Clie

the node

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-  
 ma-  
 tion  
 is  
 miss  
 ing  
 on  
 ironic.  
 Re-  
 triev  
 cur-  
 rent  
 en-  
 able  
 state  
 of  
 UEF  
 se-  
 cure  
 boot  
 on

the  
 node  
 Re-  
 turn  
 the  
 cur-  
 rent  
 en-  
 able  
 state  
 of  
 UEF  
 se-  
 cure  
 boot  
 on  
 the  
 node

Paramet
 tas  
 a  
 task  
 from  
 Task  
 ager

Raises  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 iLO  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

Raises  
 Ilo-  
 Op-  
 er-

a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 Sup-  
 port  
 if  
 UEF  
 se-  
 cure  
 boot  
 is  
 not  
 sup-  
 port

Returns

Boo  
 valu  
 in-  
 di-  
 cat-  
 ing  
 cur-  
 rent  
 state  
 of  
 UEF  
 se-  
 cure  
 boot  
 on  
 the  
 node

ironic.  
 Get  
 the  
 cur-

rent  
 state  
 of  
 sys-  
 tem  
 POS

Paramet

nod  
 an  
 iron  
 node  
 ob-  
 ject.

Returns

POS  
 state  
 of  
 the  
 serv  
 The  
 val-  
 ida  
 state  
 are:-

null,  
 Un-  
 know  
 Re-  
 set,  
 Pow  
 In-  
 Post

InPostDiscoveryComplete and FinishedPost.

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tion  
 on  
 an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary

server.

information for this driver.

Raises  
 Ilo-  
 Op-  
 er-  
 a-  
 tion  
 Sup-  
 port  
 if  
 re-  
 triev  
 ing  
 post  
 state  
 is  
 not  
 sup-  
 port  
 on  
 the  
  
 ironic.  
 Gets  
 the  
 drive  
 spe-  
 cific  
 Nod  
 info  
  
 This  
 meth  
 val-  
 i-  
 date  
 whe  
 the  
 drive  
 prop  
 erty  
 of  
 the  
 sup-  
 plied  
 node  
 con-  
 tains  
 the  
 re-  
 quir

values).

**Parameters**  
**node**  
 an  
 ironi  
 Node  
 ob-  
 ject.

**Returns**  
 a  
 dict  
 con-  
 tain-  
 ing  
 in-  
 for-  
 ma-  
 tion  
 from  
 drive  
 (or  
 whe  
 ap-  
 pli-  
 ca-  
 ble,  
 con-  
 fig

**Raises**  
 In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 any  
 pa-  
 ram-  
 e-  
 ters  
 are  
 in-  
 cor-  
 rect

**Raises**  
 Miss  
 ing-  
 Pa-

the node

in `CONF.ilo.swift_ilo_container`

ram-  
e-  
ter-  
Valu  
if  
som  
man  
tory  
in-  
for-  
ma-  
tion  
is  
miss  
ing  
on

ironic.

Re-  
mov  
the  
give  
im-  
age  
from  
swif

This  
meth  
re-  
mov  
the  
give  
im-  
age  
nam  
from  
swif  
It  
dele  
the  
im-  
age  
if  
it  
ex-  
ists

Paramet



- **obj**  
The  
nam  
of  
the  
ob-  
ject  
whic  
need  
to  
be  
re-  
mov  
from  
swif

- **ass**  
strin  
to  
de-  
pict  
the  
com  
po-  
nent  
this  
ob-  
ject  
is  
as-  
so-  
ci-  
ated  
to.

`ironic.`  
Re-  
mov  
the  
give  
im-  
age  
from  
the  
con-  
fig-  
ured  
web  
serv

This  
 meth  
 re-  
 mov  
 the  
 give  
 im-  
 age  
 from  
 the  
 http  
 lo-  
 ca-  
 tion.  
 if  
 the  
 im-  
 age  
 ex-  
 ists.

Paramet

obj  
 The  
 nam  
 of  
 the  
 im-  
 age  
 file  
 whic  
 need  
 to  
 be  
 re-  
 mov  
 from  
 the  
 web  
 serv  
 root

ironic.  
 Re-  
 mov  
 (dele  
 the  
 file  
 or  
 list  
 of  
 files

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  
sin-  
gle  
file  
is  
pass  
this

**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**

•

**node**  
 an  
 iron  
 node  
 ob-  
 ject.

- **boot**  
 Nex  
 boot  
 mod

**Raises**

Ilo-  
 Op-  
 er-  
 a-  
 tion  
 if  
 set-  
 ting  
 boot  
 mod  
 fail

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

- **task**  
 a  
 task  
 from  
 Task  
 ager
- **flag**  
 Boo  
 valu  
 True  
 if  
 the  
 se-  
 cure  
 boot  
 to  
 be  
 en-  
 able  
 in  
 next  
 boot

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 Sup-

port  
 if  
 UEF  
 se-  
 cure  
 boot  
 is  
 not  
 sup-  
 port  
 ironic.

Sets  
 up  
 sys-  
 tem  
 to  
 boot  
 from  
 UE-  
 FI-  
 HTI  
 boot  
 de-  
 vice

Sets  
 the  
 one-  
 time  
 boot  
 de-  
 vice  
 to  
 UE-  
 FI-  
 HTI  
 base  
 on  
 the  
 ar-  
 gu-  
 men-  
 sup-  
 plied

Paramet

- tas
   
 a

Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

- **iso**  
ISO  
URI  
to  
be  
set  
to  
boot  
from

- **per**  
In-  
di-  
cate  
whe  
the  
sys-  
tem  
shou  
be  
set  
to  
boot  
from  
the  
give  
de-  
vice  
one-

time or each time.

**Raises**  
Ilo-  
Op-  
er-  
a-  
tionl  
on

server.

an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary  
 Raises  
 Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 Sup-  
 port  
 if  
 re-  
 triev  
 ing  
 post  
 state  
 is  
 not  
 sup-  
 port  
 on  
 the  
 ironic.  
 At-  
 tach  
 vir-  
 tual  
 me-  
 dia  
 and  
 sets  
 it  
 as  
 boot  
 de-  
 vice  
 This  
 meth  
 at-  
 tach  
 the  
 give



ramdisk in virtual media floppy.

boot  
ISO  
as  
vir-  
tual  
me-  
dia,  
pre-  
pare  
the  
ar-  
gu-  
men-  
for

## Paramet

- **tas**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **iso**  
a  
boot  
ISO  
im-  
age  
[href](#)  
to  
at-  
tach  
to.  
Sho  
be  
ei-  
ther  
of  
be-

image object is present in CONF.ilo.swift\_ilo\_container;

low:

- A Swift object -
- It should be of format swi

It is assumed that the
- A Glance image -
- It should be of format glance / <gl or just <gl
- An HTTP URI

  - ram the options

to  
 be  
 pass  
 to  
 the  
 ram  
 in  
 vir-  
 tual  
 me-  
 dia  
 flopp

**Raises**

Im-  
 age-  
 Cre-  
 ation  
 Failure  
 if  
 it  
 fails  
 while  
 cre-  
 at-  
 ing  
 the  
 flopp  
 im-  
 age.

**Raises**

Ilo-  
 Op-  
 er-  
 a-  
 tion  
 if  
 som  
 op-  
 er-  
 a-  
 tion  
 on  
 iLO  
 fails

ironic.

Sets  
 up

it via virtual floppy image.

the  
 node  
 to  
 boot  
 from  
 the  
 give  
 ISO  
 im-  
 age.  
 This  
 meth  
 at-  
 tach  
 the  
 give  
 boot  
 on  
 the  
 node  
 and  
 pass  
 the  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters  
 to

Paramet

- tas  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
-

image object is present in CONF.ilo.swift\_ilo\_container;

boo  
a  
boot  
ISO  
im-  
age  
to  
at-  
tach  
to.  
Shou  
be  
ei-  
ther  
of  
be-  
low:  
  
-  
A  
Swi  
ob-  
ject  
-  
  
It  
shou  
be  
of  
for-  
mat  
swi  
It  
is  
as-  
sum  
that  
the  
  
-  
A  
Glar  
im-  
age  
-  
  
It  
shou  
be  
for-  
mat

nary. This is optional.

gla  
/  
<gl  
or  
just  
<gl  
-  
An  
HTT  
URI  
•  
**par**  
the  
pa-  
ram-  
e-  
ters  
to  
pass  
in  
the  
vir-  
tual  
flopp  
im-  
age  
in  
a  
dic-  
tio-  
**Raises**  
Im-  
age-  
Cre-  
ation  
Fail  
if  
it  
faile  
whil  
cre-  
at-  
ing  
the  
flopp  
im-  
age.

Raises

Swi
 Op-
 er-
 a-
 tionl
 if
 any
 op-
 er-
 a-
 tion
 with
 Swi
 fails

**Raises**

Ilo-
 Op-
 er-
 a-
 tionl
 if
 at-
 tach
 ing
 vir-
 tual
 me-
 dia
 faile

ironic.

Up-
 date
 in-
 stan
 with
 boot
 mod
 to
 be
 used
 for
 de-
 ploy
 This
 meth
 up-
 date
 in-
 stan
 with

do not have `boot_mode`. It sets the boot mode on the node.

boot  
mod  
to  
be  
used  
for  
de-  
ploy  
if  
node  
prop  
er-  
ties[

**Parameter**  
**task**  
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

**Parameter**  
**task**  
a  
task  
from  
Task  
ager



ironic.

Cha  
se-  
cure  
boot  
mod  
for  
next  
boot  
on  
the  
node  
This  
meth  
char  
se-  
cure  
boot  
mod  
on  
the  
node  
for  
next  
boot  
It  
char  
the  
se-  
cure  
boot  
mod

setting on node only if the deploy has requested for the secure boot. During deploy, this method is used to enable secure boot on the node by passing mode as True. During teardown, this method is used to disable secure boot on the node by passing mode as False.

## Paramet

- **task**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node

to  
act  
on.

- **mod**  
Boo  
valu  
re-  
ques  
ing  
the  
next  
state  
for  
se-  
cure  
boot

**Raises**  
Ilo-  
Op-  
er-  
a-  
tionl  
Sup-  
port  
if  
op-  
er-  
a-  
tion  
is  
not  
sup-  
port  
on  
iLO

**Raises**  
Ilo-  
Op-  
er-  
a-  
tionl  
if  
som  
op-  
er-  
a-  
tion  
on  
iLO

failed
   
 ironic.

Verifies
   
 the
   
 checksum
   
 of the
   
 image
   
 file
   
 against
   
 the
   
 expected
   
 one.

This
   
 method
   
 generates
   
 the
   
 checksum
   
 of the
   
 image
   
 file
   
 on the
   
 fly
   
 and
   
 verifies
   
 it
   
 against

it against the expected checksum provided as argument.

Parameters

- **image\_location**
  
 Location of the image file

fails.

ironic.drivers.modules.ilo.console module

age  
 file  
 who  
 chec  
 sum  
 is  
 ver-  
 i-  
 fied.  
 •  
 exp  
 chec  
 sum  
 to  
 be  
 chec  
 agai  
 Raises  
 Im-  
 ageE  
 f-  
 Val-  
 i-  
 da-  
 tion-  
 Failo  
 if  
 in-  
 valid  
 file  
 path  
 or  
 ver-  
 i-  
 fi-  
 ca-  
 tion  
 iLO  
 De-  
 ploy  
 Driv  
 and  
 sup-  
 port  
 ing

meth  
ods.

**class** i

Base  
*irc*  
*dri*  
*moo*  
*ipm*  
*IPM*

A  
Con  
sole  
ter-  
face  
that  
uses  
ip-  
mi-  
tool  
and  
shel  
linal

**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-  
 e-  
 ter  
 is  
 miss  
 ing

## ironic.drivers.modules.ilo.firmware\_processor module

Firm  
file  
pro-  
ces-  
sor

**class** i

Base  
obj

Firm  
im-  
age  
lo-  
ca-  
tion  
class

This  
class  
acts  
as  
a  
wrap  
per  
class  
for  
the  
firm  
im-  
age  
lo-  
ca-  
tion.  
It  
pri-  
mar-  
ily

helps in removing the firmware files from their respective locations, made available for firmware update operation.

**remove**

Ex-  
pose  
meth  
to  
re-  
mov  
the

location it wraps.

wrap  
 firm  
 file  
 This  
 meth  
 gets  
 over  
 rid-  
 den  
 by  
 the  
 re-  
 mov  
 meth  
 for  
 the  
 re-  
 spec  
 tive  
 type  
 of  
 firm  
 file  
**class** i  
 Base  
 obj  
 Firm  
 file  
 pro-  
 ces-  
 sor  
 This  
 class  
 help  
 in  
 dow  
 load  
 ing  
 the  
 firm  
 file  
 from  
 url,  
 ex-  
 tract  
 ing  
 the  
 firm



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.

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.

file  
(if  
its

## process

Pro-  
cess  
the  
firm  
file  
from  
the  
url  
  
This  
is  
the  
tem-  
plate  
meth  
whic  
dow  
load  
the  
firm  
file  
from  
url,  
ver-  
i-  
fies  
chec  
sum  
and

ironic.

Val-  
i-  
date  
the  
firm  
im-  
age

info  
 and  
 re-  
 turn  
 the  
 re-  
 triev  
 val-  
 ues.

**Paramet**

**fir**  
 dict  
 ob-  
 ject  
 con-  
 tain-  
 ing  
 the  
 firm  
 im-  
 age  
 info

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 for  
 miss  
 ing  
 field  
 (or  
 val-  
 ues)  
 in  
 im-  
 age  
 info

**Raises**

In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 for

un-  
sup-  
port  
firm  
com  
po-  
nent

Returns

tu-  
ple  
of  
firm  
url,  
check  
sum  
com  
po-  
nent  
when  
the  
firm  
up-  
date  
is  
ilo  
base

ironic.

Gets  
swift  
temp  
url.

It  
gen-  
er-  
ates  
a  
temp  
url  
for  
the  
swift  
base  
firm  
url  
to  
the  
tar-  
get  
file.  
Ex-

ing url as swift://containername/objectname.

ironic.drivers.modules.ilo.inspect module

Ins

**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

**inspect**  
 In-  
 spec  
 hard  
 ware  
 to  
 get  
 the  
 hard  
 ware  
 prop  
 er-  
 ties.  
  
 In-  
 spec  
 hard  
 ware  
 to  
 get  
 the  
 es-  
 sen-

fails if any of the essential properties are not received from the node. It doesnt fail if node fails to return any capabilities as the capabilities differ from hardware to hardware mostly.

cessfully.

tial  
and  
ad-  
di-  
tiona  
hard  
ware  
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  
coul  
not  
be  
re-  
triev  
suc-

#### **Raises**

Ilo-  
Op-  
er-  
a-  
tionl  
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-  
 plied  
 task  
 node  
 con-  
 tains  
 the  
 re-  
 quir  
 cre-

dentials information.

Parame

tas  
 a  
 task  
 from  
 Task  
 ager

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 iLO  
 pa-  
 ram-  
 e-  
 ters  
 are  
 not  
 valid

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.



## ironic.drivers.modules.ilo.management module

iLO  
Man  
age-  
men  
In-  
ter-  
face

**class** i

Base  
*iro*  
*dri*  
*mod*  
*ilo*  
*man*  
*Ilo*

**erase\_c**

Eras  
all  
the  
drive  
on  
the  
node

This  
meth  
per-  
form  
out-  
of-  
band  
san-  
i-  
tize  
disk  
eras  
on  
all  
the  
sup-  
port  
phys  
i-  
cal

drives in the node. This erase cannot be performed on logical drives.

Paramete

tas

a

Task

ager

in-

stan

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

any

of

the

ar-

gu-

men

are

in-

valid

Raises

Ilo-

Er-

ror

on

an

er-

ror

from

iLO

one\_but

Eras

the

who

sys-

tem

se-

cure

The

One

butto

se-

cure

eras

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.

pro-  
cess  
re-  
sets  
iLO  
and  
dele  
all  
li-  
cens  
stor  
ther  
re-  
sets

**Parame**  
**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

**Parameters**

**task**

a

Task

ager

ob-

ject.

**Raises**

In-

valid

Pa-

ram-

e-

ter-

Valu

if

any

of

the

ar-

gu-

men

are

in-

valid

**Raises**

Nod

Clea

ing-

Fail-

ure,

on

fail-

ure

to

ex-

e-

cute

of

clea

step

**clear\_i**

Un-

set

iSCS

de-

tails

of

the  
 sys-  
 tem  
 in  
 UEFI  
 boot  
 mod

**Parameters**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 Ilo-  
 Com  
 man  
 Not-  
 Sup-  
 port  
 e-  
 d-  
 In-  
 Bios  
 ror  
 if  
 sys-  
 tem  
 in  
 BIO  
 boot  
 mod

**Raises**  
 Ilo-  
 Er-  
 ror  
 on  
 an  
 er-  
 ror  
 from  
 iLO

**clear\_s**  
 Clea  
 all  
 se-  
 cure  
 boot

Gen9 and above servers.

keys  
 Clea  
 all  
 the  
 se-  
 cure  
 boot  
 keys  
 This  
 op-  
 er-  
 a-  
 tion  
 is  
 sup-  
 port  
 only  
 on  
 HP  
 Pro-  
 liant  
  
**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager  
  
**Raises**  
 Nod  
 Clea  
 ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 clea  
 step  
  
**Raises**  
 In-  
 stan  
 ploy

Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 de-  
 ploy  
 step

**flash\_f**

De-  
 ploy  
 step  
 to  
 Up-  
 date  
 the  
 firm  
 us-  
 ing  
 Sma  
 Up-  
 date  
 Man  
 ager  
 (SU

**Parame**

**tas**  
 a  
 Task  
 ager  
 ob-  
 ject.

**Raises**

In-  
 stan  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute

of  
de-  
ploy  
step

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-



ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 iLO  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

**Raises**

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary

**Returns**

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing:

**boot\_c**

the  
 boot  
 de-  
 vice  
 one  
 of  
 the  
 sup-

it is unknown.

unknown.

port  
de-  
vice  
liste  
in  
*irc*  
*com*  
*boo*  
or  
Non  
if

**persist**  
When  
the  
boot  
de-  
vice  
will  
per-  
sist  
to  
all  
fu-  
ture  
boot  
or  
not,  
Non  
if  
it  
is

**get\_boo**  
Get  
the  
cur-  
rent  
boot  
mod  
for  
a  
node  
  
Pro-  
vide  
the  
cur-  
rent  
boot  
mod

of  
 the  
 node

**Parameters**  
**task**  
 A  
 task  
 from  
 Task  
 ager

**Raises**  
 Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary

**Returns**  
 The  
 boot  
 mod  
 one  
 of  
 iro  
 com  
 boo  
 or  
 Non  
 if  
 it  
 is  
 un-  
 know

**get\_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\_sen

Get  
 sen-  
 sors  
 data

Paramete

tas  
 a  
 Task  
 ager  
 in-  
 stan

Raises

Fail  
 To-  
 Get-  
 Sen-  
 sor-  
 Data  
 whe  
 get-  
 ting  
 the  
 sen-  
 sor  
 data  
 fails

Raises

Fail  
 ToP  
 eSer  
 sor-

Data
 whe
 pars
 ing
 sen-
 sor
 data
 fails

Raises

In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 re-
 quir
 ipmi
 pa-
 ram-
 e-
 ters
 are
 miss
 ing.

Raises

Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 a
 re-
 quir
 pa-
 ram-
 e-
 ter
 is
 miss
 ing.

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-  
finec  
in  
*irc*  
*com*  
*boo*

**get\_sup**  
Get  
a  
list  
of

the  
 sup-  
 port  
 boot  
 de-  
 vice

**Parameters**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 any  
 ex-  
 cep-  
 tion  
 hap-  
 pens  
 in  
 pro-  
 liant  
 tils

**Returns**  
 A  
 list  
 with  
 the  
 sup-  
 port  
 boot  
 de-  
 vice  
 de-  
 fined  
 in  
*irc*  
*com*  
*boo*

**inject\_**  
 In-  
 ject

NM  
 Non  
 Mas  
 able  
 In-  
 ter-  
 rupt  
  
 In-  
 ject  
 NM  
 (Nor  
 Mas  
 able  
 In-  
 ter-  
 rupt  
 for  
 a  
 node  
 im-  
 me-  
 di-  
 ately  
  
**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**Raises**  
 Ilo-  
 Com  
 man  
 Not-  
 Sup  
 port  
 ed-  
 Er-  
 ror  
 if  
 sys-



tion.

tem  
 does  
 not  
 sup-  
 port  
 NM  
 in-  
 jec-

**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  
 cur-  
 rentl

HP Proliant Gen9 and above servers.

sup-  
 port  
 only  
 on

**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 Nod  
 Clea  
 ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 clea  
 step

**Raises**  
 In-  
 stan  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 de-  
 ploy  
 step

**reset\_i**  
 Re-  
 sets  
 the

iLO

**Parameters**

**task**

a

task

from

Task

ager

**Raises**

Node

Clear

ing-

Fail-

ure,

on

fail-

ure

to

ex-

e-

cute

of

clear

step

**Raises**

In-

stan

ploy

Fail-

ure,

on

fail-

ure

to

ex-

e-

cute

of

de-

ploy

step

**reset\_i**

Re-

sets

the

iLO

pass

wor

**Parameters**

- task**  
 a  
 task  
 from  
 Task  
 ager

- challenge**  
 Valu  
 for  
 pass  
 wor  
 to  
 up-  
 date  
 on  
 iLO

**Raises**  
 Nod  
 Clea  
 ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 clea  
 step

**Raises**  
 In-  
 stan  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of

is supported only on HP Proliant Gen9 and above servers.

de-  
 ploy  
 step  
**reset\_s**  
 Re-  
 set  
 se-  
 cure  
 boot  
 keys  
 to  
 man  
 u-  
 fac-  
 tur-  
 ing  
 de-  
 fault  
 Re-  
 sets  
 the  
 se-  
 cure  
 boot  
 keys  
 to  
 man  
 u-  
 fac-  
 tur-  
 ing  
 de-  
 fault  
 This  
 op-  
 er-  
 a-  
 tion  
**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager  
**Raises**  
 Nod  
 Clea

ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 clea  
 step  
  
**Raises**  
 In-  
 stan  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 de-  
 ploy  
 step  
  
**set\_boot**  
 Set  
 the  
 boot  
 de-  
 vice  
 for  
 a  
 node  
  
 Set  
 the  
 boot  
 de-  
 vice  
 to  
 use  
 on  
 next  
 re-  
 boot

of  
 the  
 node

Paramete

- **task**  
 a  
 task  
 from  
 Task  
 ager
- **dev**  
 the  
 boot  
 de-  
 vice  
 one  
 of  
 the  
 sup-  
 port  
 de-  
 vice  
 liste  
 in  
*ironic*  
*common*  
*boot*
- **persist**  
 Boo  
 valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 Fals

not. Default: False.

if

Raises

In-

valid

Pa-

ram-

e-

ter-

Valu

if

an

in-

valid

boot

de-

vice

is

spec

i-

fied.

Raises

Miss

ing-

Pa-

ram-

e-

ter-

Valu

if

a

re-

quir

pa-

ram-

e-

ter

is

miss

ing.

Raises

Ilo-

Op-

er-

a-

tionl

on

an

er-

ror



from  
Ilo-  
Clie  
li-  
brary

#### set\_boot

Set  
the  
boot  
mod  
for  
a  
node

Set  
the  
boot  
mod  
to  
use  
on  
next  
re-  
boot  
of  
the  
node

#### Parameters

- **task**  
A  
task  
from  
Task  
agen
- **mod**  
The  
boot  
mod  
one  
of  
*irc*  
*com*  
*boo*

#### Raises

In-  
valid

Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 boot  
 mod  
 is  
 spec  
 i-  
 fied.

Raises

Illo-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 set-  
 ting  
 boot  
 mod  
 fail

set\_is

Set  
 iSCS  
 de-  
 tails  
 of  
 the  
 sys-  
 tem  
 in  
 UEF  
 boot  
 mod  
  
 The  
 ini-  
 tia-  
 tor  
 is  
 set  
 with  
 the  
 tar-  
 get

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.

ble for invalid input cases.

ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 clea  
 step

**Raises**  
 In-  
 stan  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 de-  
 ploy  
 step

**update\_**  
 Clea  
 step  
 to  
 up-  
 date  
 the  
 firm  
 us-  
 ing  
 Sma  
 Up-  
 date  
 Man  
 ager  
 (SU

**Parame**  
**tas**  
 a  
 Task

ager  
 ob-  
 ject.

**Raises**

Nod  
 Clea  
 ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 of  
 clea  
 step

**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

dentials information.

prop  
erty  
of  
the  
sup-  
plied  
task  
node  
con-  
tains  
the  
re-  
quir  
cre-

Paramet  
tas  
a  
task  
from  
Task  
ager

Raises  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
re-  
quir  
iLO  
pa-  
ram-  
e-  
ters  
are  
not  
valid

Raises  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if

a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

## ironic.drivers.modules.ilo.power module

iLO  
Pow  
Driv

**class** `i`  
Base  
*irc*  
*dri*  
*bas*  
*Pow*

**get\_pow**  
Gets  
the  
cur-  
rent  
pow  
state

### Parame

- **tas**  
a  
Task  
ager  
in-  
stan
- **nod**  
The  
Nod

**Returns**  
one  
of

irc  
 com  
 sta  
 POV  
 POV  
 or  
 ER-  
 ROF

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 iLO  
 cre-  
 den-  
 tials  
 are  
 miss  
 ing.

Raises

Ilo-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 Ilo-  
 Clie  
 li-  
 brary

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

Param

tas  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
 cur-  
 rent  
 not  
 used

Returns

A  
 list  
 with

the  
 sup-  
 port  
 pow  
 state  
 de-  
 finec  
 in  
*irc*  
*com*  
*sta*

**reboot**  
 Re-  
 boot  
 the  
 node

**Parame**

•  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan

•  
**tim**  
 time  
 out  
 (in  
 sec-  
 onds  
 Un-  
 sup-  
 port  
 by  
 this  
 in-  
 ter-  
 face

**Raises**  
 Pow  
 er-  
 State  
 Fail-  
 ure  
 if  
 the  
 fi-

nal  
state  
of  
the  
node  
is  
not  
POV

#### **Raises**

Ilo-  
Op-  
er-  
a-  
tionl  
on  
an  
er-  
ror  
from  
Ilo-  
Clie  
li-  
brary

#### **set\_pow**

Turn  
the  
cur-  
rent  
pow  
state  
on  
or  
off.

#### **Parame**

- **tas**  
a  
Task  
ager  
in-  
stan
- **pow**  
The  
de-  
sired  
pow  
state

POW  
 or  
 RE-  
 BOO  
 from  
*irc*  
*com*  
*sta*

- **time**  
 time  
 out  
 (in  
 sec-  
 onds  
 Un-  
 sup-  
 port  
 by  
 this  
 in-  
 ter-  
 face

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 pow  
 state  
 was  
 spec  
 i-  
 fied.

**Raises**  
 Ilo-  
 Op-  
 er-  
 a-  
 tion  
 on  
 an  
 er-

ror  
from  
Ilo-  
Clie  
li-  
brary

#### Raises

Power-  
State  
Failure  
if  
the  
power  
could  
be  
set  
to  
power

#### validation

Check  
if  
node  
contains  
the  
required  
iLO  
credentials

#### Parameters

- **task**  
a Task object  
instance
- **node**  
Single node object.

Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 iLO  
 cre-  
 den-  
 tials  
 are  
 miss  
 ing.

ironic.drivers.modules.ilo.raid module

iLO.  
 RAI  
 spe-  
 cific  
 meth  
 ods

**class** i  
 Base  
*irc*  
*dri*  
*bas*  
*RAI*  
 Im-  
 ple-  
 men  
 ta-  
 tion  
 of  
 OOI  
 RAI  
 In-  
 ter-  
 face  
 for  
 iLO.

**apply\_c**

Ap-  
plies  
RAL  
con-  
fig-  
u-  
ra-  
tion  
on  
the  
give  
node

**Parame**

- **tas**  
A  
Task  
ager  
in-  
stan
- **rai**  
The  
RAL  
con-  
fig-  
u-  
ra-  
tion  
to  
ap-  
ply.
- **cre**  
Set-  
ting  
this  
to  
Fals  
in-  
di-  
cate  
not  
to  
cre-  
ate

ified in `raid_config`. Default value is `True`.

cept the root volume) in `raid_config`. Default value is `True`.

creating the new configuration.

root  
vol-  
ume  
that  
is  
spec

- **cre**  
Set-  
ting  
this  
to  
`False`  
in-  
di-  
cate  
not  
to  
cre-  
ate  
non-  
root  
vol-  
ume  
(all  
ex-

- **del**  
Set-  
ting  
this  
to  
`True`  
in-  
di-  
cate  
to  
dele  
RAID  
con-  
fig-  
u-  
ra-  
tion  
prior  
to

**Raises**



In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in-  
 valid

Returns

state  
 if  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if  
 it  
 is  
 com

plete.

create\_

Cre-  
 ate  
 a  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion

on  
 a  
 bare  
 meta-  
 us-  
 ing  
 ager  
 rame  
  
 This  
 meth  
 cre-  
 ates  
 a  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give  
 node

Paramete

- **task**  
 a  
 Task  
 ager  
 in-  
 stan
- **create**  
 If  
 True  
 a  
 root  
 vol-  
 ume  
 is  
 cre-  
 ated  
 dur-  
 ing  
 RAI  
 con-  
 fig-  
 u-

erwise, no root volume is created. Default is True.

ated. Default is True.

ter skipping root volume and/or non-root volumes.

ra-  
tion.  
Oth-

- **cre**  
If  
True  
non-  
root  
vol-  
ume  
are  
cre-  
ated  
If  
Fals  
no  
non-  
root  
vol-  
ume  
are  
cre-

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
node  
is  
miss  
ing  
or  
was  
foun  
to  
be  
emp  
af-

**Raises**  
Nod  
Clea  
ing-

Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 clear  
 step

**Raises**

In-  
 stan-  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 de-  
 ploy  
 step

**delete\_**

Dele  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.

**Parame**

**tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act

on.

**Raises**

Nod  
 Clea  
 ing-  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 clea  
 step

**Raises**

In-  
 stan  
 ploy  
 Fail-  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 de-  
 ploy  
 step

**get\_pro**

Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

ironic.drivers.modules.ilo.vendor module

Vendor  
 interface  
 for iLO  
 driver  
 and its  
 support  
 ing meth  
 ods.

**class** i

Base  
*ironic*  
*driver*  
*base*  
*Vendor*

Vendor  
 specific  
 interface  
 for iLO  
 de-  
 ploy  
 driver

**boot\_**ir

At-  
 tach  
 an  
 ISO  
 im-  
 age  
 in  
 glan  
 and  
 re-  
 boot  
 bare  
 meta

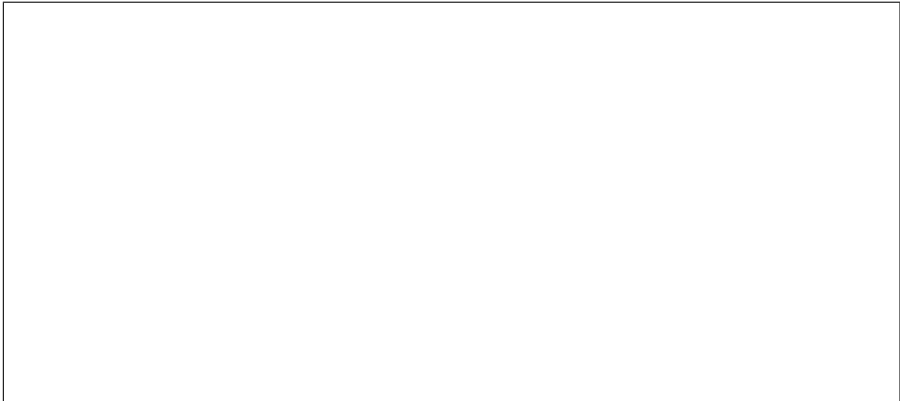
virtual media and then reboots the node. This is useful for debugging purposes. This can be invoked only when the node is in manage state.

Param

- **task**  
 A Task object.
- **kwargs**  
 The arguments sent with vendor password. The expected kwargs are:

(continues on next page)

(continued from previous page)



**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  
 venc  
 spec  
 ac-  
 tions  
 Che  
 if  
 a  
 valid



the vendor passthru method.

ven-  
dor  
pass  
meth  
was  
pass  
and  
val-  
i-  
date  
the  
pa-  
ram-  
e-  
ters  
for

## Parame

- **tas**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **met**  
meth  
to  
be  
val-  
i-  
date
- **kwa**  
kwa  
con-  
tain-  
ing  
the

ven-  
 dor  
 pass  
 meth  
 ods  
 pa-  
 ram-  
 e-  
 ters.

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters  
 were  
 not  
 pass

**Raises**

In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 any  
 of  
 the  
 pa-  
 ram-  
 e-  
 ters  
 have  
 in-  
 valic  
 valu

## Module contents

`ironic.drivers.modules.intel_ipmi` package

## Submodules

`ironic.drivers.modules.intel_ipmi.management` module

In-  
tel  
IPM  
Har  
ware  
  
Sup-  
port  
In-  
tel  
Spee  
Se-  
lect  
Per-  
for-  
man  
Pro-  
file.

**class** i

Base  
*iro*  
*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  
 set-  
 tings  
 from  
 the  
 set-  
 tings  
 para  
 and  
 ap-  
 plies

tion on the given node. After the BIOS configuration is done, `self.cache_bios_settings()` may be called to sync the nodes BIOS-related information with the BIOS configuration applied on the node. It will also validate the given settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the `driver_info` properties.

dictionary as well.

BIO  
con-  
fig-  
u-  
ra-

## Parameters

- **task**  
a  
Task  
ager  
in-  
stan
- **set**  
Dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
BIO  
con-  
fig-  
u-  
ra-  
tion.  
It  
may  
be  
an  
emp

## Raises

IRM  
C-  
Op-  
er-  
a-  
tion  
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
 it
 is
 com
 plete

**factory**

Re-
 set
 BIO
 con-
 fig-
 u-
 ra-
 tion
 to
 fac-
 tory
 de-
 fault
 on
 the
 give
 node

**Parame**

**tas**
 a
 Task
 ager
 in-
 stan

**Raises**

Un-
 sup-
 port.
 ed-
 Driv
 ten-
 sion
 if
 the
 node
 drive
 does
 sup-
 port
 BIO
 re-
 set.

information for this driver to manage the BIOS settings of the node.

**get\_prop**  
 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-  
 plied  
 node  
 con-  
 tains  
 the  
 re-  
 quir  
  
**Paramet**  
**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-  
tribu  
is  
miss  
ing  
or  
in-  
valid

on the node.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing  
in

the driver\_info property.

ironic.drivers.modules.irmc.boot module

links the instance kernel/ramdisk in nodes directory in tftproot and removes the PXE config.

iRM  
 Boo  
 Driv  
  
**class** i  
 Base  
*irc*  
*dri*  
*mod*  
*pxe*  
*PXE*  
  
 iRM  
 PXE  
 boot

**clean\_u**  
 Clea  
 up  
 the  
 boot  
 of  
 in-  
 stan  
  
 This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that  
 was  
 setu  
 for  
 boot  
 ing  
 the  
 in-  
 stan  
 It  
 un-

Parame  
 tas

a  
 task  
 from  
 Task  
 ager

**Raises**

IRM  
 C-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 som  
 op-  
 er-  
 a-  
 tion  
 on  
 iRM  
 faile

**Returns**

Non

**prepare**

Pre-  
 pare  
 the  
 boot  
 of  
 in-  
 stan  
  
 This  
 meth  
 pre-  
 pare  
 the  
 boot  
 of  
 the  
 in-  
 stan  
 af-  
 ter  
 read  
 ing  
 rel-  
 e-  
 vant  
 in-

tion from the nodes instance\_info. In case of netboot, it updates the dhcp entries and switches the PXE config. In case of localboot, it cleans up the PXE config.

**Parameters**  
 task: a Task object representing the task.

**Returns**  
 None

**Raises**  
 IronicException: if some operation fails on iRM.

**prepare\_boot**  
 Prepares the boot of IronRAM using PXE. This method prepares the boot of the de-

formation from the nodes `driver_info` and `instance_info`.

ters as kernel command-line arguments.

ploy  
ker-  
nel/  
af-  
ter  
read  
ing  
rel-  
e-  
vant  
in-

## Parameters

- **task**  
a  
task  
from  
Task  
ager
- **ramdisk**  
the  
pa-  
ram-  
e-  
ters  
to  
be  
pass  
to  
the  
ram  
pxe  
drive  
pass  
thes  
pa-  
ram-  
e-

## Returns

Non

## Raises

Miss  
ing-  
Pa-  
ram-

or instance\_info.

e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 is  
 miss  
 ing  
 in  
 node  
 drive

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 pro-  
 vide  
 is  
 in-  
 valid

Raises

Iron  
 icEx  
 cep-  
 tion.  
 if  
 som  
 pow  
 or  
 set  
 boot  
 de-  
 vice  
 op-  
 er-

node.

a-  
tion  
faile  
on  
the

**class i**

Base  
*irc*  
*dri*  
*bas*  
*Boo*  
*irc*  
*dri*  
*moo*  
*irm*  
*boo*  
*IRM*

iRM  
Vir-  
tual  
Me-  
dia  
boot  
relat  
ac-  
tions

**capabil**

**clean\_u**

Clea  
up  
the  
boot  
of  
in-  
stan

This  
meth  
clea  
up  
the  
en-  
vi-  
ron-  
men  
that

was  
 setu  
 for  
 boot  
 ing  
 the  
 in-  
 stan

**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**Raises**  
 IRM  
 C-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 iRM  
 op-  
 er-  
 a-  
 tion  
 faile

**clean\_u**  
 Clea  
 up  
 the  
 boot  
 of  
 iron  
 rame  
  
 This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that



cue ramdisk.

was  
 setu  
 for  
 boot  
 ing  
 the  
 de-  
 ploy  
 or  
 res-

**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**Raises**  
 IRM  
 C-  
 Op-  
 er-  
 a-  
 tionl  
 if  
 iRM  
 op-  
 er-  
 a-  
 tion  
 faile

**get\_pro**  
 Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

**Returns**  
 dic-  
 tio-

tion from the nodes database.

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  
 ing  
 rel-  
 e-  
 vant  
 in-  
 for-  
 ma-  
 Parame  
 tas  
 a  
 task  
 from  
 Task  
 ager  
 Returns  
 Non  
 prepare

Pre-prepare the deployment or rescue ramdisk using virtual media.

Pre-prepare the options for the deployment or rescue ramdisk sets the node to boot from virtual

tual media cdrom.

Paramete

- **task**
 a Task agent in-stand con-tain-ing the node

fied href.

to  
 act  
 on.  
 •  
 ram  
 the  
 op-  
 tion  
 to  
 be  
 pass  
 to  
 the  
 ram  
 Raises  
 Im-  
 ageE  
 f-  
 Val-  
 i-  
 da-  
 tion-  
 Fail  
 if  
 no  
 im-  
 age  
 ser-  
 vice  
 can  
 han-  
 dle  
 spec  
 i-  
 Raises  
 Im-  
 age-  
 Cre-  
 ation  
 Fail  
 if  
 it  
 faile  
 whil  
 cre-  
 at-  
 ing  
 the  
 flopp

face or ManagementInterface fails.

im-  
age.  
**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
val-  
i-  
da-  
tion  
of  
the  
Pow  
er-  
In-  
ter-  
**Raises**  
IRM  
C-  
Op-  
er-  
a-  
tionl  
if  
som  
op-  
er-  
a-  
tion  
on  
iRM  
fails  
**validat**  
Val-  
i-  
date  
the  
de-  
ploy  
men  
in-  
for-  
ma-

tion  
 for  
 the  
 task  
 node

**Parameters**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 con-  
 fig  
 op-  
 tion  
 has  
 in-  
 valid  
 valu

**Raises**  
 IRM  
 C-  
 Shar  
 File  
 tem-  
 Not-  
 Mou  
 if  
 shar  
 file  
 sys-  
 tem

is  
not  
mou

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
in-  
for-  
ma-  
tion  
is  
in-  
valid

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
ker-  
nel\_  
and  
ram  
are  
miss  
ing  
in  
the  
Glar  
im-

age, or if kernel and ramdisk are missing in the Non Glance image.

#### **validat**

Val-  
i-  
date  
that  
the  
node  
has  
re-

eters

quire  
 prop  
 er-  
 ties  
 for  
 res  
 cue.  
**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  
 mor  
 re-  
 quire  
 pa-  
 ram-  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if



any  
of  
the  
pa-  
ram-  
e-  
ters  
have  
in-  
valid  
valu

**class** i  
Base  
obj  
Mix  
in  
class  
for  
vol-  
ume  
boot  
con-  
fig-  
u-  
ra-  
tion  
to  
iRM  
iRM  
has  
a  
fea-  
ture  
to  
set  
up  
re-  
mote  
boot  
to  
a  
serv  
This  
fea-  
ture  
can  
be  
used

by VIOM (Virtual I/O Manager) library of SCCI client.

info has a value of key `irmc_boot_iso`, it indicates that `boot_option` is `netboot`. Therefore it attaches the boot ISO on the bare metal node and then sets the node to boot from virtual media `cdrom`.

ironic.  
 At-  
 tach  
 boot  
 ISO  
 for  
 a  
 de-  
 ploy  
 node  
 if  
 it  
 ex-  
 ists.  
 This  
 meth  
 chec  
 the  
 in-  
 stan  
 info  
 of  
 the  
 bare  
 meta  
 node  
 for  
 a  
 boot  
 ISO  
 If  
 the  
 in-  
 stan

Paramet  
 tas  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act

on.

**Raises**

IRM  
C-  
Op-  
er-  
a-  
tionl  
if  
at-  
tach  
ing  
vir-  
tual  
me-  
dia  
faile

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
val-  
i-  
da-  
tion  
of  
the  
Man  
age-  
men  
ter-

face fails.

ironic.

Che  
if  
Shar  
File  
Sys-  
tem  
(NF  
or  
CIF  
is  
mou

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
con-  
fig  
op-  
tion  
has  
in-  
valid  
valu

Raises

IRM  
C-  
Shar  
File  
tem-  
Not-  
Mou  
if  
shar  
file  
sys-  
tem  
is  
not  
mou

ironic.drivers.modules.irmc.common module

Com  
mon  
func  
tion-  
al-  
i-  
ties  
shar  
be-  
twec  
dif-  
fer-  
ent  
iRM

on the iRMC.

mod  
ules

ironic.  
Gets  
an  
iRM  
SCC  
clien  
Give  
an  
iron  
node  
ob-  
ject,  
this  
meth  
give  
back  
a  
iRM  
SCC  
clien  
to  
do  
op-  
er-  
a-  
tions

#### Paramet

**nod**  
An  
iron  
node  
ob-  
ject.

#### Returns

sccl  
par-  
tial  
func  
tion  
whic  
take  
a  
SCC  
com  
man

the node

para
 **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-
 ma-
 tion
 is
 miss
 ing
 on
 ironic.
 Gets
 iRM
 SCC
 re-
 port
 Give
 an
 iron
 node
 ob-
 ject,
 this
 meth

give  
back  
a  
iRM  
SCC  
re-  
port

**Parameters**  
**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**  
 Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
man  
tory  
in-  
for-  
ma-  
tion  
is

the node.

miss  
ing  
on

Raises

scci.  
if  
re-  
quir  
pa-  
ram-  
e-  
ters  
are  
in-  
valid

Raises

scci.  
if  
SCC  
faile

ironic.  
Gets  
the  
spe-  
cific  
Nod  
drive  
info  
  
This  
meth  
val-  
i-  
date  
whe  
the  
drive  
prop  
erty  
of  
the  
sup-  
plied  
node  
con-  
tains  
the  
re-  
quir



information for this driver.

**Parameters**

**node**  
 An  
 ironi  
 node  
 ob-  
 ject.

**Returns**

A  
 dict  
 con-  
 tain-  
 ing  
 in-  
 for-  
 ma-  
 tion  
 from  
 drive  
 and  
 de-  
 fault  
 val-  
 ues.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 in-  
 valid  
 valu  
 is  
 con-  
 taine  
 in  
 the  
 drive  
 prop  
 erty.

**Raises**

Miss  
 ing-  
 Pa-

erty.

ironic.

En-  
 able  
 or  
 dis-  
 able  
 UEF  
 Se-  
 cure  
 Boo

Paramet

- **nod**  
 An  
 iron  
 node  
 ob-  
 ject.
- **ena**  
 Boo  
 valu  
 True  
 if  
 the  
 se-  
 cure  
 boot  
 to  
 be  
 en-

able

**Raises**

IRM

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\_prop**

Re-  
 turn  
 the  
 prop-  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

**Returns**

dic-  
 tio-  
 nary  
 of  
 <prop-  
 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-  
 er-  
 ties  
 and  
 mac-  
 ad-  
 dres

**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 Harc  
 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

information for this driver.

spec  
in-  
spec  
tion  
in-  
for-  
ma-  
tion.  
  
This  
meth  
val-  
i-  
date  
whe  
the  
drive  
prop  
erty  
of  
the  
sup-  
plied  
node  
con-  
tains  
the  
re-  
quir

**Parame**  
**tas**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**Raises**  
In-  
valid  
Pa-  
ram-  
e-

on the node.

ter-  
Valu  
if  
re-  
quir  
driv  
at-  
tribu  
is  
miss  
ing  
or  
in-  
valid

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

`ironic.`  
SC2  
sc2U  
re-  
turn  
NIC  
type

#### **sc2UnitN**

SYN  
TAX  
IN-  
TE-  
GEF  
{  
un-  
know

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 }

hardware (MAC) address ::= { sc2ManagementNodes 9 }

pri-  
mar-  
sec-  
ond-  
man-  
blad-  
seco-  
remo-  
seco-  
remo-  
back-

ironic.  
SC2  
sc2U  
re-  
turn  
NIC  
MA  
ad-  
dres

**sc2UnitN**  
SYN  
TAX  
Phys  
dres  
AC-  
CES  
read  
only  
STA  
TUS  
man  
tory  
DE-  
SCR  
TIO  
Man  
age-  
men  
node



## 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\_ser**

Get  
sen-  
sors  
data  
meth  
  
It

dict format.

gets  
 sen-  
 sor  
 data  
 from  
 the  
 task  
 node  
 via  
 SCC  
 and  
 con-  
 vert  
 the  
 data  
 from  
 XM  
 to  
 the

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Fail  
 To-  
 Get-  
 Sen-  
 sor-  
 Data  
 whe  
 get-  
 ting  
 the  
 sen-  
 sor  
 data  
 fails

**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-  
 ters  
 are  
 in-  
 valid

**Raises**

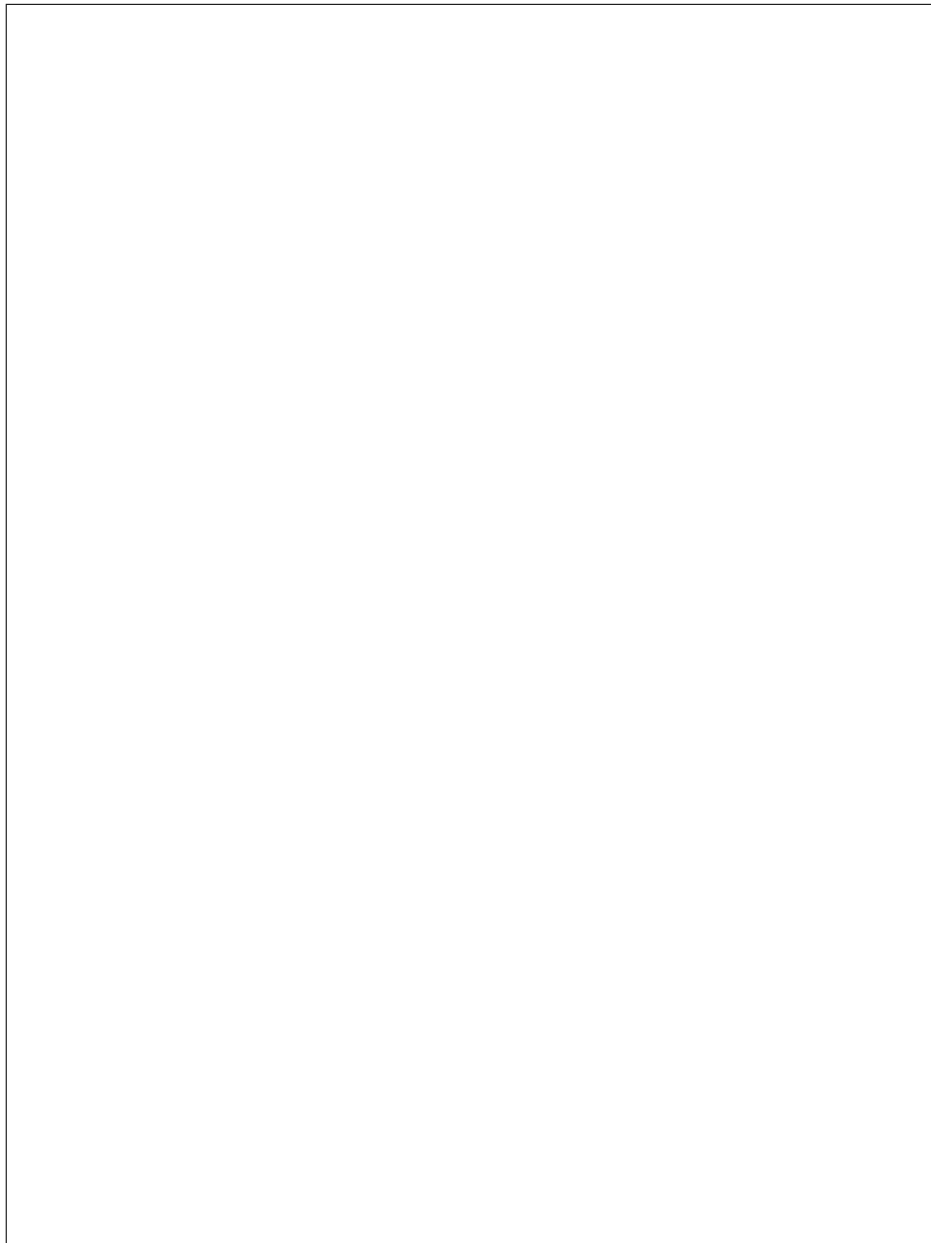
Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

**Returns**

Re-  
 turn  
 a  
 con-  
 sis-

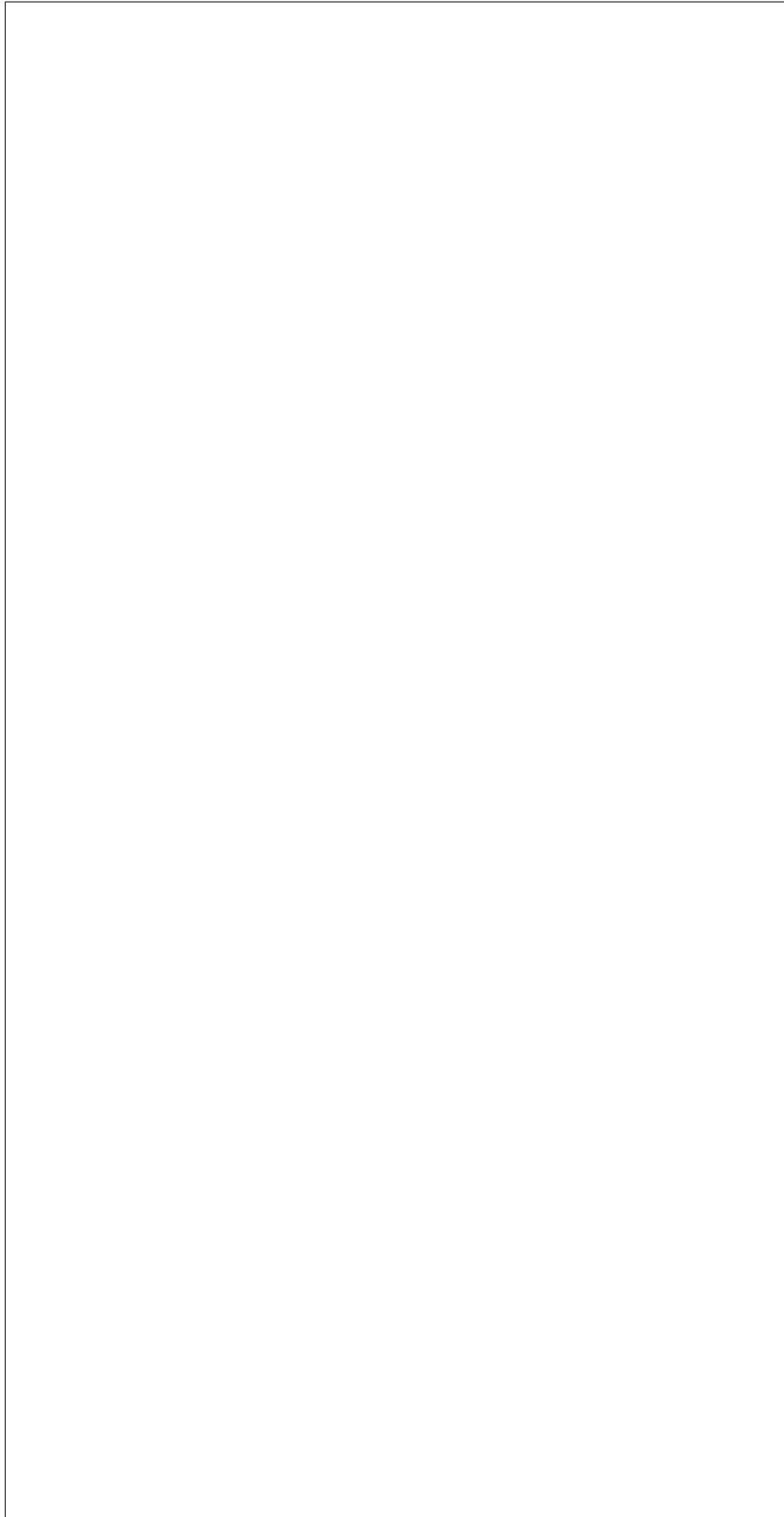
tent  
for-  
mat-  
ted  
dict  
of  
sen-  
sor  
data  
grou  
by  
sen-  
sor  
type  
whic

can be processed by Ceilometer. Example:



(continues on next page)

(continued from previous page)



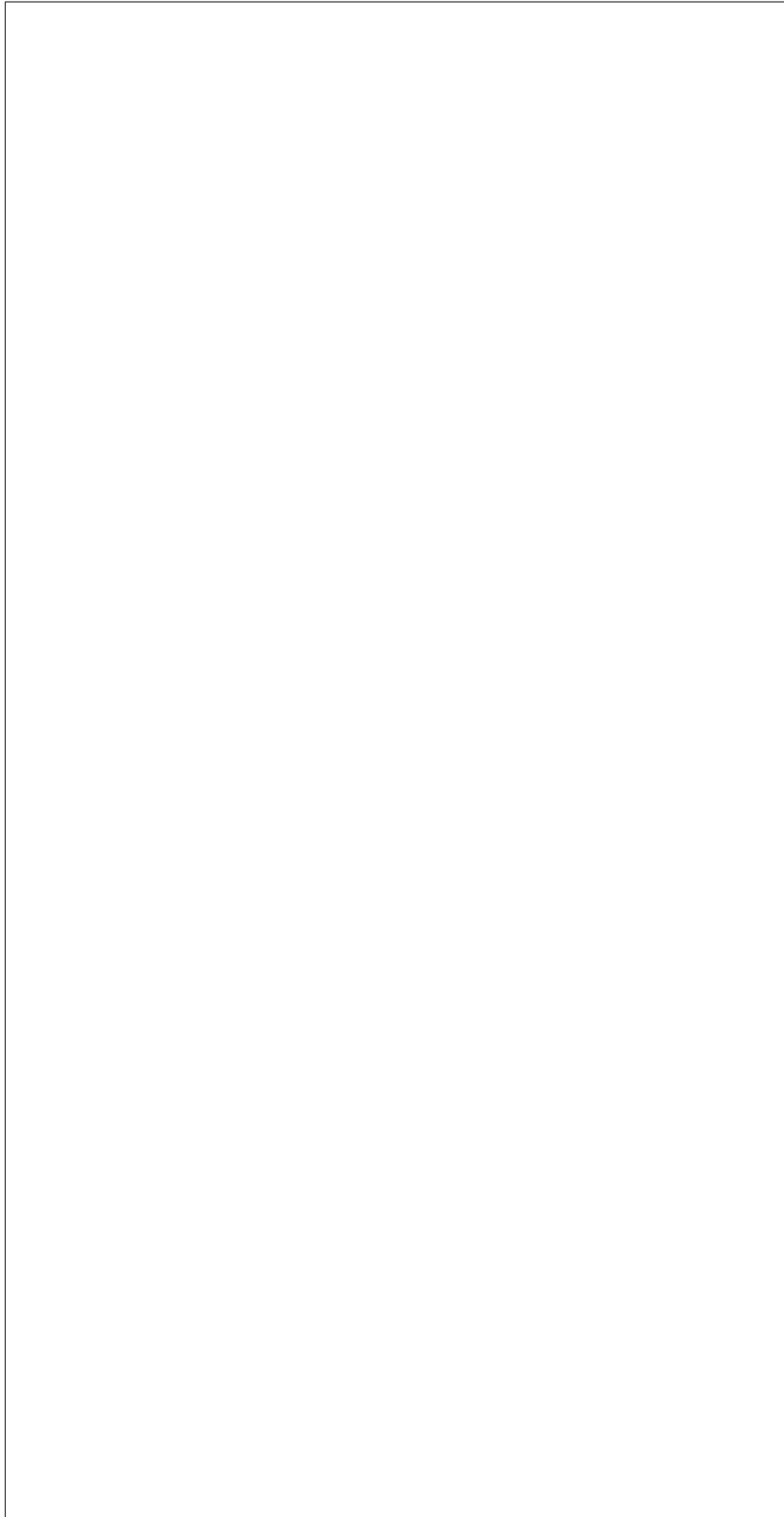
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



**inject\_**  
In-  
ject  
NM  
Non  
Mas  
able  
In-  
ter-  
rupt  
  
In-  
ject  
NM



(Non-  
 Mas-  
 able  
 In-  
 ter-  
 rupt  
 for  
 a  
 node  
 im-  
 me-  
 di-  
 ately

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 IRM  
 C-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 SCC

**Returns**  
 Non-

**restore**  
 Re-  
 store  
 BIO  
 con-  
 fig

for  
 a  
 node

**Parameters**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 Node  
 Clearing  
 ing-  
 Failure,  
 ure,  
 on  
 fail-  
 ure  
 to  
 ex-  
 e-  
 cute  
 step

**Returns**  
 Non

**set\_boot\_device**  
 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 object returned from TaskManager.get\_task()
- device**  
 The device to boot. One of the supported device names listed in [ironic-compute-libs/boothw](#)
- persist**  
 Boolean value. True if the boot device will persist 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  
 spec  
 i-  
 fied.

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

Raises

IP-  
 MI-  
 Fail-  
 ure  
 on  
 an  
 er-  
 ror  
 from  
 ip-  
 mi-  
 tool.

information for this driver.

**validat**  
Val-  
i-  
date  
the  
drive  
spec  
man  
age-  
men  
in-  
for-  
ma-  
tion.  
  
This  
meth  
val-  
i-  
date  
whe  
the  
drive  
prop  
erty  
of  
the  
sup-  
plied  
node  
con-  
tains  
the  
re-  
quir

**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  
pa-  
ram-  
e-  
ters  
are  
in-  
valid

Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

ironic.

Back  
BIO  
con-  
fig  
from  
a  
node

Paramet

tas  
a  
Task  
ager

in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

Raises

IRM  
 C-  
 Op-  
 er-  
 a-  
 tion  
 on  
 fail-  
 ure.

ironic.drivers.modules.irmc.power module

iRM  
 Pow  
 Driv  
 us-  
 ing  
 the  
 Base  
 Serv  
 Pro-  
 file

class i

Base  
*irc*  
*dri*  
*bas*  
*Pow*

In-  
 ter-  
 face  
 for  
 pow  
 relat  
 ac-  
 tions

**get\_pow**  
 Re-  
 turn  
 the  
 pow  
 state  
 of  
 the  
 task  
 node

**Parameter**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 a  
 pow  
 state  
 One  
 of  
*irc*  
*com*  
*sta*

**Raises**  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 ipmi  
 pa-  
 ram-  
 e-  
 ters



are  
miss  
ing.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

**Raises**

IP-  
MI-  
Fail-  
ure  
on  
an  
er-  
ror  
from  
ip-  
mi-  
tool  
(from  
\_pov  
call)

**get\_pro**

Re-  
turn  
the  
prop  
er-  
ties  
of  
the  
in-  
ter-  
face

Returns

dictionary of properties and names describing the entries

get\_supported

Get a list of the supported power states

Parameters

**task**  
 A TaskManager instance containing the node to act on. Currently not used

Returns

A list with the supported power

state  
de-  
fine  
in  
*irc*  
*com*  
*sta*

#### reboot

Per-  
form  
a  
hard  
re-  
boot  
of  
the  
task  
node

#### Parame

- **tas**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **tim**  
time  
out  
(in  
sec-  
onds  
pos-  
i-  
tive  
in-  
te-  
ger  
(>

indicates default timeout.

0)  
 for  
 any  
 pow  
 state  
 Non

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 pow  
 state  
 was  
 spec  
 i-  
 fied.

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

indicates default timeout.

- **task**  
a Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

- **power**  
Any  
power  
state  
from  
*irc*  
*com*  
*sta*

- **timeout**  
time  
out  
(in  
sec-  
onds  
pos-  
i-  
tive  
in-  
te-  
ger  
(>  
0)  
for  
any  
power  
state  
Non

**Raises**  
In-  
valid

Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valid  
pow  
state  
was  
spec  
i-  
fied.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
man  
tory  
in-  
for-  
ma-  
tion  
is  
miss  
ing  
on

the node

**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  
con-  
tain  
the  
re-  
quir

information for this driver to manage the power state of the node.

#### Parame

tas  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

#### Raises

In-  
vali  
Pa-

on the node.

Raises

Missing parameter if a required parameter is missing.

ironic.SC2sc2s returns status of the current boot



## ironic.drivers.modules.irmc.raid module

IroniC  
RAID  
spe-  
cific  
meth  
ods

**class** `ironic.drivers.modules.irmc.raid`

### **create\_raid**

Cre-  
ate  
the  
RAID  
con-  
fig-  
u-  
ra-  
tion.

This  
meth  
cre-  
ates  
the  
RAID  
con-  
fig-  
u-  
ra-  
tion  
on  
the  
give  
node

### **Parameters**

- **task**  
a  
Task  
ager

erwise, no root volume is created. Default is True.

in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
 •  
**cre**  
 If  
 True  
 a  
 root  
 vol-  
 ume  
 is  
 cre-  
 ated  
 dur-  
 ing  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.  
 Oth-  
 •  
**cre**  
 If  
 True  
 non-  
 root  
 vol-  
 ume  
 are  
 cre-  
 ated  
 If  
 Fals  
 no  
 non-  
 root  
 vol-  
 ume  
 are

ated. Default is True.

cre-  
**Returns**  
 state  
 if  
 RAL  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 node  
 is  
 miss  
 ing  
 or  
 emp

**Raises**  
 IRM  
 C-  
 Op-  
 er-  
 a-  
 tionl  
 on  
 an  
 er-  
 ror  
 from  
 sc-  
 ci-  
 clien

**delete\_**  
 Dele  
 the

RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.

**Parameters**  
**tasks**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 state  
 if  
 dele-  
 tion  
 is  
 in  
 prog-  
 asyn-  
 chro-  
 or  
 Non-  
 if  
 it  
 is  
 com-  
 plete

**get\_properties**  
 Re-  
 turn  
 the  
 prop-  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

## Module contents

`ironic.drivers.modules.network` package

## Submodules

`ironic.drivers.modules.network.common` module

```
class i
    Base
    irc
    dri
    moo
    net
    com
    VIF

    VIF
    port
    ID
    mixi
    class
    for
    neu-
    tron
    net-
    worl
    in-
    ter-
    face

    Mix
    class
    that
    pro-
    vide
    VIF
    relat
    net-
    worl
    in-
    ter-
    face
    meth
    ods
    for
    neu-
    tron
    net-
    worl
    in-
```

terfaces. On VIF attach/detach, the associated neutron port will be updated.

VIFs.

get\_noo  
 Get  
 net-  
 worl  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 data  
 for  
 node  
 port  
 Pull  
 net-  
 worl  
 data  
 from  
 iron  
 node  
 ob-  
 ject  
 if  
 pres  
 oth-  
 er-  
 wise  
 col-  
 lect  
 it  
 for  
 Neu  
 tron

Parame  
 tas  
 A  
 Task  
 ager  
 in-  
 stan

Raises  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-

is invalid.

Valu
 if
 the
 net-
 worl
 in-
 ter-
 face
 con-
 fig-
 u-
 ra-
 tion

Raises

Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 som
 pa-
 ram-
 e-
 ters
 are
 miss
 ing.

Returns

a
 dict
 hold
 ing
 net-
 worl
 con-
 fig-
 u-
 ra-
 tion
 in-
 for-
 ma-
 tion
 ad-
 hear
 ing

network metadata layout (*network\_data.json*).



flict

## portgroup

Handle any actions required when a portgroup character

## Parameters

- **task**  
a TaskManager instance
- **portgroup**  
a character Portgroup object from the API before it is saved to data

## Raises

FailureToUpgradeDeviceHardwareNotOnPort

to attach the virtual interface to, the following ordered criteria are applied:

Con-  
 flict  
**vif\_att**  
 At-  
 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-  
 ing  
 a  
 port  
 or  
 port  
 grou  
 •  
 Re-  
 quir  
 port  
 or  
 port  
 grou  
 to  
 have  
 a  
 phys  
 i-  
 cal

or one of the VIFs allowed physical networks.

net-  
worl  
that  
is  
ei-  
ther  
Non

- Pre-  
fer  
port  
or  
port  
grou  
with  
a  
phys  
i-  
cal  
net-  
worl  
field  
whic  
is  
not  
Non
- Pre-  
fer  
port  
grou  
to  
port
- Pre-  
fer  
port  
with  
PXE  
en-  
able

## Parame

- **tas**  
A  
Task

whose value is a unique identifier for that VIF.

ager  
 in-  
 stan  
 •  
**vif**  
 a  
 dic-  
 tio-  
 nary  
 of  
 in-  
 for-  
 ma-  
 tion  
 about  
 a  
 VIF.  
 It  
 mus  
 have  
 an  
 id  
 key,  
**Raises**  
 Net-  
 worl  
 Er-  
 ror,  
 Vi-  
 fAl-  
 read  
 At-  
 tach  
 NoF  
 hys-  
 i-  
 cal-  
 Port  
**Raises**  
 Port  
 grou  
 Phys  
 net-  
 Inco  
 sis-  
 tent  
 if  
 one  
 of

all assigned the same physical network.

the  
 node  
 port  
 grou  
 has  
 port  
 whic  
 are  
 not

**vif\_detach**

De-  
 tach  
 a  
 vir-  
 tual  
 net-  
 worl  
 in-  
 ter-  
 face  
 from  
 a  
 node

**Parameters**

- **task**  
 A  
 Task  
 ager  
 in-  
 stan
- **vif\_id**  
 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  
 class  
 that  
 pro-  
 vide  
 VIF  
 relat  
 net-  
 worl  
 in-  
 ter-  
 face  
 meth  
 ods  
 for  
 non-  
 neut  
 net-  
 worl

terfaces. There are no effects due to VIF attach/detach that are external to ironiC.

vif\_attach, vif\_detach, port\_changed, or portgroup\_changed.

in-

NOT  
This  
does  
not  
yet  
sup-  
port  
the  
full  
set  
of  
VIF  
meth-  
ods,  
as  
it  
does  
not  
pro-  
vide

**get\_cur**

Re-  
turn  
the  
cur-  
rently  
used  
VIF  
as-  
so-  
ci-  
ated  
with  
port  
or  
port-  
group

We  
are  
boot-  
ing  
the  
node  
only  
in  
one  
net-

means were doing cleaning, of provisioning\_vif\_port\_id - provisioning, of rescuing\_vif\_port\_id - rescuing. Otherwise its a tenant network

Parameters

- **task\_manager\_instance**  
 A TaskManager instance
- **port\_obj**  
 IronPort object or portgroup object.

Returns

VIF ID associated with p\_obj or Non

vif\_list

List of VIF IDs for a



node	
<b>Parameters</b>	
<b>task</b>	
A Task	
ager	
in-	
stan-	
<b>Returns</b>	
List	
of	
VIF	
dic-	
tio-	
nar-	
ies,	
each	
dic-	
tio-	
nary	
will	
have	
an	
id	
en-	
try	
with	
the	
ID of the VIF.	
	<code>ironic.</code>

Find	
free	
port	
like	
ob-	
ject	
(por	
grou	
or	
port	
VIF	
will	
be	
at-	
tach	
to.	

or portgroup to attach the virtual interface to, the following ordered criteria are applied:

or one of the VIFs allowed physical networks.

En-  
sure  
that  
the  
VIF  
is  
not  
al-  
read  
at-  
tach  
to  
this  
node  
Whe  
se-  
lect-  
ing  
a  
port

- Re-  
quir  
port  
or  
port  
grou  
to  
have  
a  
phys  
i-  
cal  
net-  
worl  
that  
is  
ei-  
ther  
Non

- Pre-  
fer  
port  
or  
port  
grou  
with

a  
 phys  
 i-  
 cal  
 net-  
 worl  
 field  
 whic  
 is  
 not  
 Non

- Pre-fer  
 port  
 grou  
 to  
 port
- Pre-fer  
 port  
 with  
 PXE  
 en-able

Paramet

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan
- **vif**  
 Nam  
 or  
 UUI  
 of  
 a  
 VIF
- **phy**  
 Set  
 of  
 phys

erned by the segments of the VIFs network. An empty set indicates that the ports physical networks should be ignored.

i-  
cal  
net-  
worl  
on  
whic  
the  
VIF  
may  
be  
at-  
tach  
This  
is  
gov-

- **vif**  
dict  
that  
may  
con-  
tain  
ex-  
tra  
in-  
for-  
ma-  
tion,  
such  
as  
port.

**Raises**  
Vi-  
fAl-  
read  
At-  
tach  
if  
VIF  
is  
al-  
read  
at-  
tach  
to  
the  
node

**Raises**

all assigned the same physical network.

NoF  
 hys-  
 i-  
 cal-  
 Port  
 if  
 there  
 is  
 no  
 port  
 like  
 ob-  
 ject  
 VIF  
 can  
 be  
 at-  
 tach  
 to.

Raises

Port  
 grou  
 Phys  
 net-  
 Inco  
 sis-  
 tent  
 if  
 one  
 of  
 the  
 node  
 port  
 grou  
 has  
 port  
 whic  
 are  
 not

Returns

port  
 like  
 ob-  
 ject  
 VIF  
 will  
 be  
 at-  
 tach

to.

ironic.

Plug  
 port  
 like  
 ob-  
 ject  
 to  
 ten-  
 ant  
 net-  
 worl

Paramet

- **tas**
  
 A
   
 Task
   
 ager
   
 in-
   
 stan
- **por**
  
 port
   
 like
   
 ob-
   
 ject
   
 to
   
 plug
- **cli**
  
 Neu
   
 tron
   
 clien
   
 in-
   
 stan

Raises

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.

**ironic.drivers.modules.network.flat module**

Flat  
net-  
worl  
in-  
ter-  
face  
Use-  
ful  
for  
shar  
flat  
net-  
worl

**class i**

Base  
*irc*  
*dri*  
*mod*  
*net*  
*com*  
Neu  
*irc*  
*com*  
*neu*  
Neu  
*irc*

*dri*  
*bas*  
*Net*  
 Flat  
 net-  
 worl  
 in-  
 ter-  
 face

**add\_cle**  
 Add  
 the  
 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-  
 valio  
 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-  
Valu  
if  
the  
net-  
worl  
in-  
ter-

is invalid.

face  
 con-  
 fig-  
 u-  
 ra-  
 tion  
  
**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  
  
 Flat

ure\_tenant\_network() unbound it.

net-  
 worl  
 does  
 not  
 use  
 the  
 res-  
 cu-  
 ing  
 net-  
 worl  
 Bind  
 the  
 port  
 agai  
 sinc  
 un-  
 con-  
 fig-

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan

**Returns**  
 a  
 dic-  
 tio-  
 nary  
 in  
 the  
 form  
 {por  
 neu-  
 tron

**Raises**  
 Net-  
 worl  
 Er-  
 ror,  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu

**configure**  
 Configure  
 the network  
 for a node

**Parameter: task\_agent\_instance**  
 A task agent instance

**remove\_node**  
 Remove the  
 node from  
 the network

**Parameter: task\_agent\_instance**  
 A task agent instance

**Raises**  
 NetworkError

**remove\_node**  
 Remove the  
 node from  
 the network

is invalid.

worl
 from
 a
 node
 **Parame**
**tas**
 A
 Task
 ager
 in-
 stan
 **Raises**
 In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 the
 net-
 worl
 in-
 ter-
 face
 con-
 fig-
 u-
 ra-
 tion
 **Raises**
 Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 som
 pa-
 ram-
 e-
 ters
 are
 miss
 ing.

remove\_

Re-

mov

the

pro-

vi-

sion

ing

net-

worl

from

a

node

Paramet

tas

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

the

port

agai

sinc

it.

the tenant and cleaning networks at the same time.

add\_
 bound
 Param
 tas
 A
 Task
 ager
 in-
 stan
 Raises
 Net-
 worl
 Er-
 ror
 unconfi
 Un-
 con-
 fig-
 ure
 ten-
 ant
 net-
 worl
 for
 a
 node
 Un-
 bind
 the
 port
 here
 to
 avoi
 the
 pos-
 si-
 bil-
 ity
 of
 the
 iron
 port
 be-
 ing
 bound
 to

is invalid.

tas  
 A  
 Task  
 ager  
 in-  
 stan  
 Raises  
 Net-  
 worl  
 Er-  
 ror  
 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  
 con-  
 fig-  
 u-  
 ra-  
 tion



Mis-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu-  
if  
some  
pa-  
ram-  
e-  
ters  
are  
miss-  
ing.

Bas  
irc  
dri  
mod  
net  
com  
Neu  
irc  
com  
neu  
Neu  
irc  
dri  
bas  
Net

Neu  
tron  
v2  
net-  
wor  
in-  
ter-  
face

tron  
 port  
 for  
 each  
 port  
 on  
 task  
 to  
 boot  
 the  
 ram

**Parameters**  
**task**  
 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

**Parameters**  
**task**  
 A  
 Task

is invalid.

ager  
 in-  
 stan

**Returns**  
 a  
 dic-  
 tio-  
 nary  
 in  
 the  
 form  
 {por  
 neu-  
 tron

**Raises**  
 Net-  
 worl  
 Er-  
 ror

**Raises**  
 In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 net-  
 worl  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion

**add\_pro**  
 Add  
 the  
 pro-  
 vi-  
 sion  
 ing  
 net-  
 worl

to  
 a  
 node

**Parameters**  
**tasks**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Net-  
 worl  
 Er-  
 ror

**add\_res**  
 Cre-  
 ate  
 neu-  
 tron  
 port  
 for  
 each  
 port  
 to  
 boot  
 the  
 res-  
 cue  
 ram

**Parameters**  
**tasks**  
 a  
 Task  
 ager  
 in-  
 stan

**Returns**  
 a  
 dic-  
 tio-  
 nary  
 in  
 the  
 form  
 {por  
 neu-  
 tron

**config**

Con-  
fig-  
ure  
ten-  
ant  
net-  
worl  
for  
a  
node

**Paramete**

**tas**  
A  
Task  
ager  
in-  
stan

**Raises**

Net-  
worl  
Er-  
ror

**need\_po**

Che  
if  
the  
node  
has  
any  
Sma  
NIC  
port

**Paramete**

**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  
 rame

**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**

is invalid.

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
net-  
worl  
in-  
ter-  
face  
con-  
fig-  
u-  
ra-  
tion

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
pa-  
ram-  
e-  
ters  
are  
miss  
ing.

#### **remove\_**

Re-  
mov  
the  
pro-  
vi-  
sion  
ing  
net-  
worl  
from  
a  
node

**Parameters**

**task**

A

Task

ager

in-

stan

**Raises**

Net-

worl

Er-

ror

**remove\_**

Dele

neu-

tron

port

cre-

ated

for

boot

ing

the

res-

cue

ram

**Parameters**

**task**

a

Task

ager

in-

stan

**Raises**

Net-

worl

Er-

ror

**unconfi**

Un-

con-

fig-

ure

ten-

ant

net-

worl

for

a



possibility of the ironiC port being bound to the tenant and cleaning networks at the same time.

node  
Nov  
take  
care  
of  
port  
re-  
mov  
from  
ten-  
ant  
net-  
worl  
we  
un-  
bind  
it  
here  
to  
avoi  
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  
Task  
ager

is invalid.

in-  
 stan-  
**Raises**  
 In-  
 valid-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu-  
 if  
 the  
 net-  
 worl-  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion  
**Raises**  
 Miss-  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu-  
 if  
 som-  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss-  
 ing.  
**validat**  
 Val-  
 i-  
 date  
 the  
 net-  
 worl-  
 in-  
 ter-  
 face

is invalid.

for  
 res-  
 cue  
 op-  
 er-  
 a-  
 tion.  
**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 net-  
 worl  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion  
**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 pa-  
 ram-  
 e-  
 ters  
 are

ironic.drivers.modules.network.noop module

miss  
ing.

**class** i

Base  
*iro*  
*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**

A

Task

ager

in-

stan

**get\_cur**

Re-

turn

the

cur-

rentl

used

VIF

as-

so-

ci-

ated

with

port

or

port

grou

We

are

boot

ing

the

node

only

means were doing cleaning, of provisioning\_vif\_port\_id - provisioning of rescuing\_vif\_port\_id - rescuing. Otherwise its a tenant network

Paramete

- **tas**  
 A Taskager instance
- **p\_o**  
 Ironic port or port group object.

Returns

VIF ID associated with p\_obj or Non

port\_ch

Handle 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

#### portgro

Han  
dle  
any  
ac-  
tions  
re-  
quir  
whe  
a  
port  
grou  
char

#### Parame

- task**  
 a Task  
 ager  
 in-  
 stan

- port**  
 a  
 char  
 Port  
 grou  
 ob-  
 ject.

**Raises**  
 Con  
 flict.  
 Fail  
 ToU  
 dat-  
 eD-  
 HCP  
 tOn-  
 Port

**remove\_**  
 Re-  
 mov  
 the  
 clea  
 ing  
 net-  
 worl  
 from  
 a  
 node

**Paramet**  
**task**  
 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**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan

**validat**  
 Val-  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quir  
 prop  
 er-  
 ties  
 for  
 in-  
 spec

tion.

**Parameters**

**task**

A Taskager instance with the node being checked.

**vif\_attach**

Attach a virtual network world interface to a node.

**Parameters**

•

**task**

A Taskager instance.

•

**vif**

a dictionary of information about

whose value is a unique identifier for that VIF.

a  
 VIF.  
 It  
 mus  
 have  
 an  
 id  
 key,

Raises

Net-  
 worl  
 Er-  
 ror,  
 Vi-  
 fAl-  
 read  
 At-  
 tach  
 NoF  
 hys-  
 i-  
 cal-  
 Port

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

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Returns**  
 List  
 of  
 VIF  
 dic-  
 tio-  
 nar-  
 ies,  
 each  
 dic-  
 tio-  
 nary  
 will  
 have  
 an  
 id  
 en-  
 try  
 with

ID of the VIF.

## Module contents

`ironic.drivers.modules.redfish` package

## Submodules

`ironic.drivers.modules.redfish.bios` module

the

**class** `ironic.drivers.modules.redfish.bios.BIOS`

Base class for BIOS driver.  
*ironic.drivers.modules.redfish.bios.BIOS*

**apply\_config**

Apply BIOS settings to the node.

**Parameters**

- **task**  
a Task object containing the node to act on.
- **set**

a  
 list  
 of  
 BIO  
 set-  
 tings  
 to  
 be  
 up-  
 date

**Raises**

Red  
 fish-  
 Con  
 nec-  
 tion-  
 Erro  
 whe  
 it  
 fails  
 to  
 con-  
 nect  
 to  
 Red  
 fish

**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  
 ta-  
 ble.

**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 Red  
 fish-  
 Con  
 nec-  
 tion-  
 Erro  
 whe  
 it  
 fails  
 to  
 con-  
 nect  
 to  
 Red

fish

**Raises**

Redfish-Error: An error from the Sushli-brary

**Raises**

Unsupported-Drive-tension-sion-if-the-system-does-not-support-BIO-setting

**factory**

Reset the BIO settings of the node to the factory default



Param

tas

a

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

Raises

Red

fish-

Con

nec-

tion-

Erro

whe

it

fails

to

con-

nect

to

Red

fish

Raises

Red

fish-

Er-

ror

on

an

er-

ror

from

the

Sush

li-

brary

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\_cc**

Per-  
form  
post  
con-  
fig-  
u-  
ra-  
tion  
ac-  
tion  
to  
store  
the  
BIO  
set-  
tings  
  
Ex-  
ten-  
sion  
point  
to  
al-  
low  
ven-  
dor  
im-  
ple-  
men  
ta-

override this method to perform a custom action to write the BIOS settings to the Redfish service. The default implementation performs a reboot.

#### Parameters

- **task**  
a Task Agent instance containing the node to act on.
- **settings**  
a list of BIOS settings to be updated.

**post\_reboot**  
Perform post-reboot action to apply

override this method to perform a custom action to apply the BIOS factory reset to the Redfish service. The default implementation performs a reboot.

the  
 BIO  
 fac-  
 tory  
 re-  
 set.  
 Ex-  
 ten-  
 sion  
 poin  
 to  
 al-  
 low  
 ven-  
 dor  
 im-  
 ple-  
 men  
 ta-  
 tions  
 to  
 ex-  
 tend  
 this  
 class  
 and  
  
**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**validat**  
 Val-  
 i-  
 date  
 the  
 driv  
 in-

for-  
 ma-  
 tion  
 need  
 by  
 the  
 red-  
 fish  
 drive

**Parameters**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**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

ironic.drivers.modules.redfish.boot module

the user image that BMC inserts into the drive.

pa-  
ram-  
e-  
ter(s)

class i

Base  
irc  
dri  
bas  
Boo

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-  
tual  
CD/  
drive  
con-  
tain-  
ing

The  
CD/  
im-

tion) could be pulled over HTTP, served as iSCSI targets or NFS volumes.

is only needed for UEFI boot)

ages  
mus  
be  
in  
ISO  
for-  
mat  
and  
(de-  
pend  
ing  
on  
BM  
im-  
ple-  
men  
ta-

The  
base  
line  
boot  
worl  
flow  
look  
like  
this:

1. Pull  
ker-  
nel,  
ram  
and  
ESP  
(FAT  
par-  
ti-  
tion  
im-  
age  
with  
EFI  
boot  
load  
im-  
ages  
(ESI

2. Cre-

Swift temporary URL

push it to Glance and pass to the BMC as Swift temporary URL

ate  
boot  
ISO  
out  
of  
im-  
ages  
(#1)  
push  
it  
to  
Glance  
and  
pass  
to  
the  
BMC  
as

3. Op-  
tion-  
ally  
cre-  
ate  
flopp  
im-  
age  
with  
de-  
sired  
sys-  
tem  
con-  
fig-  
u-  
ra-  
tion  
data

4. In-  
sert  
CD/  
and  
(op-  
tion-  
ally)  
flopp  
im-



*cue\_kernel/rescue\_ramdisk* properties from *[instance\_info]* or *[driver\_info]*.

in the Glance image metadata found in *[instance\_info]image\_source* node property.

ages  
and  
set  
prop  
boot  
mod  
  
For  
buil  
ing  
de-  
ploy  
or  
res-  
cue  
ISO  
red-  
fish  
boot  
in-  
ter-  
face  
uses  
*de-  
ploy*  
or  
*res-*

For  
buil  
ing  
boot  
(use  
ISO  
red-  
fish  
boot  
in-  
ter-  
face  
seek  
*ker-  
nel\_*  
and  
*ram*  
prop  
er-  
ties

**clean\_up**  
 Clean up the boot of in-stand.  
 This meth clean up the en-vi-ron-men that was setu for boot ing the in-stand.

**Parameters**  
**task**  
 A task from Taskager

**Returns**  
 None

**clean\_up**  
 Clean up the boot of iron-rame.  
 This meth clean up the en-

vi-  
 ron-  
 men-  
 that  
 was  
 setu  
 for  
 boot  
 ing  
 the  
 de-  
 ploy  
 rame

**Parameters**  
**task**  
 A  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**get\_properties**  
 Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

**Returns**  
 dic-  
 tio-  
 nary  
 of  
 <prop  
 erty  
 nam  
 de-  
 scrip  
 tion:  
 en-  
 tries

**prepare**  
 Pre-

tion from the nodes `instance_info`.

pare  
the  
boot  
of  
in-  
stan  
over  
vir-  
tual  
me-  
dia.

This  
meth  
pre-  
pare  
the  
boot  
of  
the  
in-  
stan  
af-  
ter  
read  
ing  
rel-  
e-  
vant  
in-  
for-  
ma-

The  
in-  
ter-  
nal  
logic  
is  
as  
fol-  
lows

- If *boot* re-ques for this de-  
ploy

dia boot image

is  
 lo-  
 cal,  
 then  
 set  
 the  
 node  
 to  
 boot  
 from  
 disk

- Un-  
 less  
*boot*  
 re-  
 ques-  
 for  
 this  
 de-  
 ploy  
 is  
 rame  
 pass  
 root  
 disk  
 ID  
 to  
 vir-  
 tual  
 me-
- Oth-  
 er-  
 wise  
 build  
 boot  
 im-  
 age,  
 in-  
 sert  
 it  
 into  
 vir-  
 tual  
 me-  
 dia  
 de-  
 vice

node to boot from CD.

and  
 set

**Parameters**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**Raises**  
 In-  
 stan-  
 ploy  
 Fail-  
 ure,  
 if  
 its  
 try  
 to  
 boot  
 iSCS  
 vol-  
 ume  
 in  
 BIO  
 boot  
 mod

**prepare**  
 Pre-  
 pare  
 the  
 boot  
 of  
 de-  
 ploy  
 or  
 res-  
 cue  
 ram  
 over  
 vir-  
 tual  
 me-  
 dia.  
 This

vant information from the nodes `driver_info` and `instance_info`.

meth  
pre-  
pare  
the  
boot  
of  
the  
de-  
ploy  
or  
res-  
cue  
ram  
af-  
ter  
read  
ing  
rel-  
e-

## Parame

- **task**  
A  
task  
from  
Task  
ager
- **ram**  
the  
pa-  
ram-  
e-  
ters  
to  
be  
pass  
to  
the  
ram

## Returns

Non

## Raises

Miss  
ing-  
Pa-  
ram-

or instance\_info.

e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 is  
 miss  
 ing  
 in  
 node  
 drive

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 pro-  
 vide  
 is  
 in-  
 valid

Raises

Iron  
 icEx  
 cep-  
 tion.  
 if  
 som  
 pow  
 or  
 set  
 boot  
 boot  
 de-  
 vice  
 op-



the node.

validat

Val-  
i-  
date  
the  
de-  
ploy  
men  
in-  
for-  
ma-  
tion  
for  
the  
task  
node

This  
meth  
val-  
i-  
date  
whe  
the  
drive  
and/  
in-  
stan  
prop  
er-  
ties  
of  
the  
task  
node  
con-  
tains

the required information for this interface to function.

Parame

tas  
A  
Task  
ager  
in-  
stan

con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**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)

**validat**

Val-  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quir  
 prop  
 er-

ties  
 for  
 in-  
 spec-  
 tion.

**Parameters**  
**task**  
 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  
 mor  
 re-  
 quir  
 pa-  
 ram-

eters

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

ironic.

thing (default).

Eject  
 vir-  
 tual  
 CDs  
 and  
 DVI

Paramet

•

**tas**  
 A  
 task  
 from  
 Task  
 ager

•

**boo**  
 sush  
 boot  
 de-  
 vice  
 e.g.  
*VIR-*  
*TUA*  
*VIR-*  
*TUA*  
 or  
*VIR-*  
*TUA*  
 or  
*Non*  
 to  
 ejec  
 ev-  
 ery-

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 no  
 suit-  
 able  
 vir-

the node.

### ironic.drivers.modules.redfish.inspect module

tual  
CD  
or  
DVI  
is  
foun  
on

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

sential properties are not received from the node.

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  
 es-  
**Paramete**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
**Raises**  
 Harc  
 ware  
 spec  
 tion-

cessfully.

Fail-  
ure  
if  
es-  
sen-  
tial  
prop-  
er-  
ties  
could  
not  
be  
re-  
triev-  
suc-

#### 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  
when  
the  
drive  
prop-  
er-  
ties  
of

information for this interface to function.

long-running checks.

the  
task  
node  
con-  
tains  
the  
re-  
quir

This  
meth  
is  
of-  
ten  
ex-  
e-  
cute  
syn-  
chro  
in  
API  
re-  
ques  
so  
it  
shou  
not  
con-  
duct

Paramet  
**tas**  
A  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

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)

**ironic.drivers.modules.redfish.management module**

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Man*

**clear\_s**

Clea  
all  
se-  
cure  
boot  
keys

### Paramete

**tas**  
a  
task  
from  
Task

ager

**Raises**

Un-  
 sup-  
 port-  
 ed-  
 Driv-  
 ten-  
 sion  
 if  
 se-  
 cure  
 boot  
 is  
 now  
 sup-  
 port

**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-  
 turer  
 field

**Parame**

state is specified.

tas  
 A  
 task  
 from  
 Task  
 ager  
 Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 com  
 po-  
 nent  
 in-  
 di-  
 ca-  
 tor  
 or  
 Raises  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing  
 Raises  
 Red  
 fish-  
 Er-



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  
Red  
fish

**Raises**

Red  
fish-  
Er-  
ror  
on  
an  
er-

ror  
 from  
 the  
 Sush  
 li-  
 brary

Returns

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing:

boot\_c

the  
 boot  
 de-  
 vice  
 one  
 of  
*irc*  
*com*  
*boo*  
 or  
 Non  
 if  
 it  
 is  
 un-  
 knov

persist

Boo  
 valu  
 or  
 Non  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 per-  
 sists  
 Fals  
 oth-  
 er-  
 wise

unknown.

Non  
if  
its

**get\_boot**  
Get  
the  
current  
boot  
mode  
for  
a  
node

Pro-  
vide  
the  
current  
boot  
mode  
of  
the  
node

**Parameter**  
**task**  
A  
task  
from  
Task  
manager

**Raises**  
Miss-  
ing-  
Parameter  
error  
ValueError  
if  
a  
required  
parameter  
error  
is  
miss

ing

**Raises**

Driv  
Op-  
er-  
a-  
tionl  
or  
its  
deriv  
tive  
in  
case  
of  
drive  
run-  
time  
er-  
ror.

**Returns**

The  
boot  
mod  
one  
of  
iro  
com  
boo  
or  
Non  
if  
it  
is  
un-  
know

**get\_inc**

Get  
cur-  
rent  
state  
of  
the  
in-  
di-  
ca-  
tor  
of  
the  
hard  
ware



com  
po-  
nent

Param

- **task**  
 A  
 task  
 from  
 Task  
 ager
- **component**  
 The  
 hard  
 ware  
 com  
 po-  
 nent  
 one  
 of  
*irc*  
*com*  
*com*
- **indicator**  
 In-  
 di-  
 ca-  
 tor  
 ID  
 (as  
 re-  
 port  
 by  
*get\_*

Raises

Miss  
 ing-  
 Pa-  
 ram  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir

pa-  
ram-  
e-  
ter  
is  
miss  
ing

**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  
*iroc*  
*com*  
*inc*

**get\_proc**

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-  
 e-  
 ter  
 is  
 miss  
 ing

**Raises**

Redfish-Error or its derivative in case of a drive run-time error.

**Raises**

Unsupported Drive-Extension if secure boot is not supported by the hardware

**Returns**

Boolean

**get\_sensors\_data**

Get sensor data

**Parameters**

**task** a Task

ager  
 in-  
 stan

**Raises**

Fail  
 To-  
 Get-  
 Sen-  
 sor-  
 Data  
 whe  
 get-  
 ting  
 the  
 sen-  
 sor  
 data  
 fails

**Raises**

Fail  
 ToP  
 eSer  
 sor-  
 Data  
 whe  
 pars  
 ing  
 sen-  
 sor  
 data  
 fails

**Raises**

In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

**Raises**

Missing parameter value if a requirement parameter is missing.

**Returns**

A dictionary of sensor data groups by sensor type.

**get\_supported\_devices**

Get a list of the supported boot devices.

**Parameter**

**task**
 a task from Task

ager

**Returns**

A  
 list  
 with  
 the  
 sup-  
 port  
 boot  
 de-  
 vice  
 de-  
 fined  
 in  
*irc*  
*com*  
*boo*

**get\_sup**

Get  
 a  
 list  
 of  
 the  
 sup-  
 port  
 boot  
 mod

**Parame**

**tas**  
 A  
 task  
 from  
 Task  
 ager

**Returns**

A  
 list  
 with  
 the  
 sup-  
 port  
 boot  
 mod  
 de-  
 fined  
 in  
*irc*  
*com*  
*boo*

cant be determined, empty list is returned.

If  
boot  
mod  
sup-  
port  
  
**get\_sup**  
Get  
a  
map  
of  
the  
sup-  
port  
in-  
di-  
ca-  
tors  
(e.g.  
LED

Paramete

- **task**  
A  
task  
from  
Task  
ager
- **com**  
If  
not  
*Non*  
re-  
turn  
in-  
di-  
ca-  
tor  
in-  
for-  
ma-  
tion  
for  
just  
this  
com  
po-

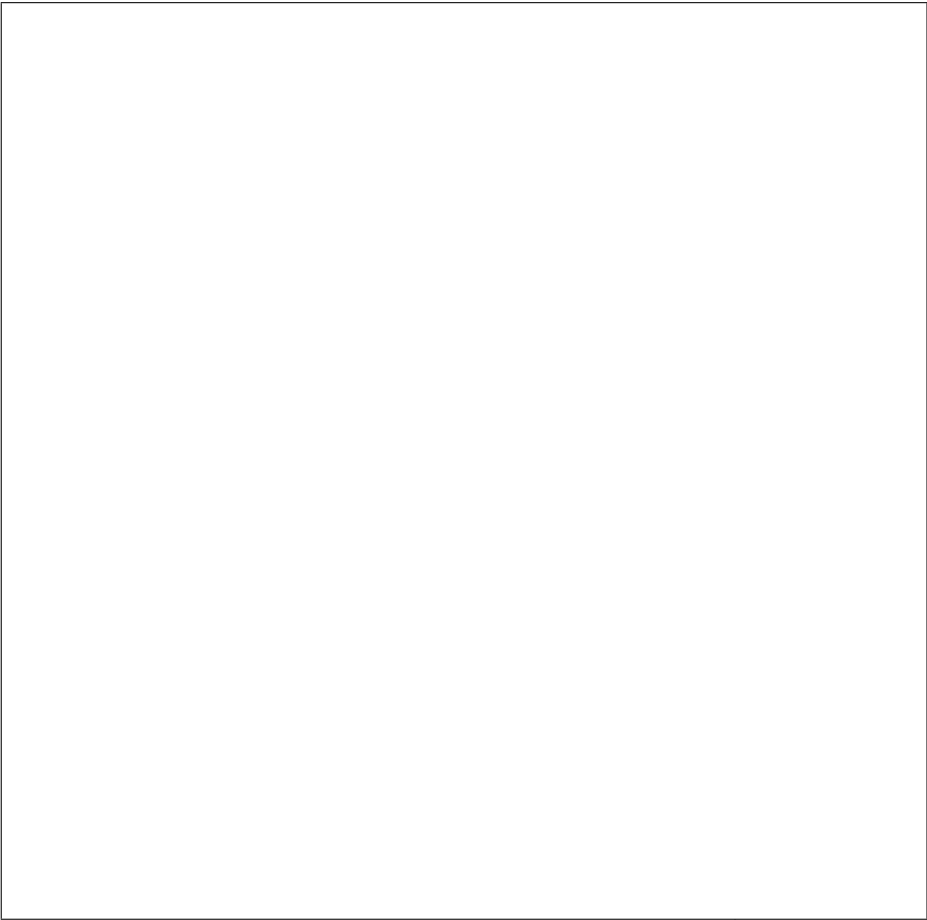


nent, otherwise return indicators for all existing components.

Returns

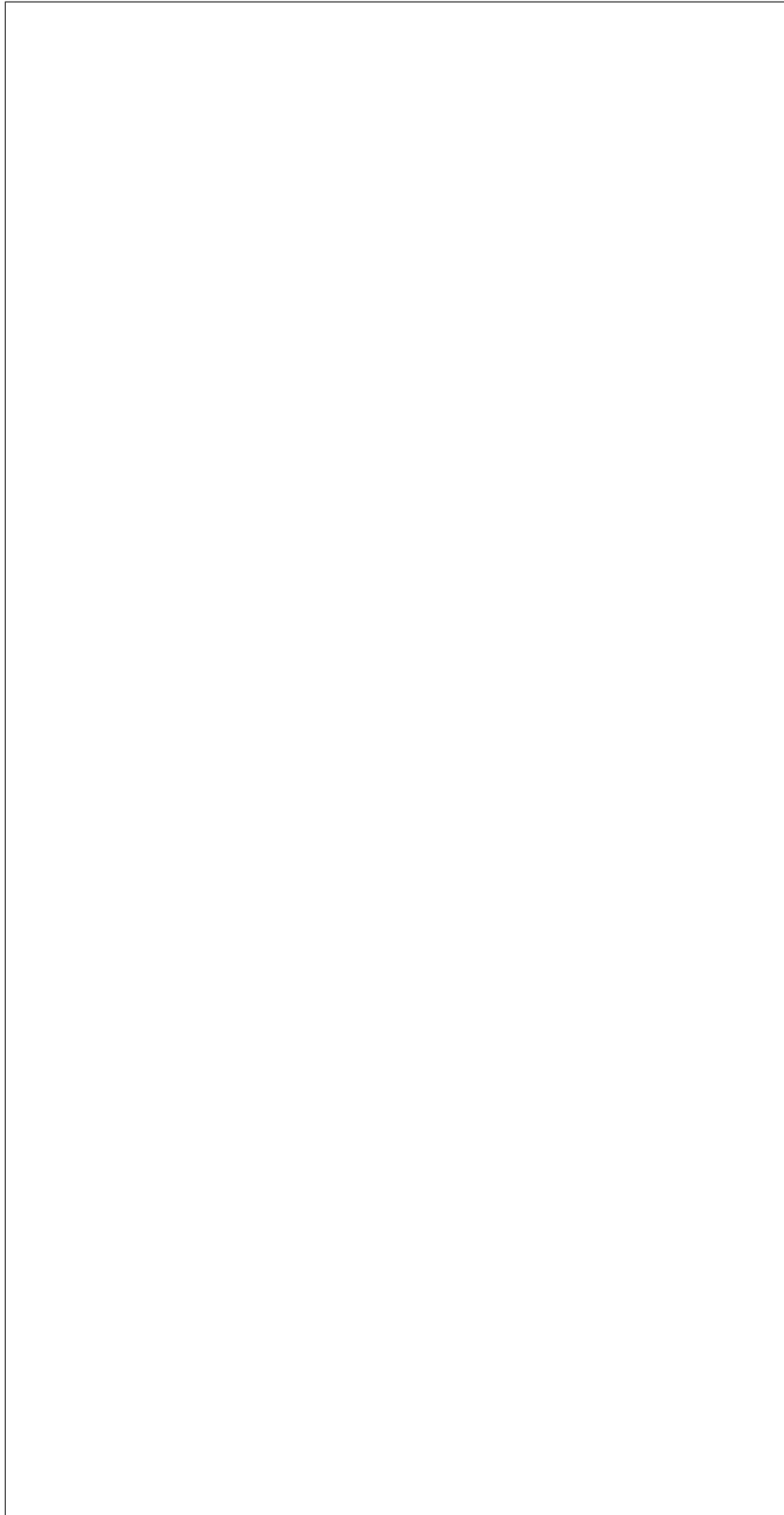
A  
 dic-  
 tio-  
 nary  
 of  
 hard  
 ware  
 com-  
 po-  
 nent  
 (*ir*  
*com*  
*com*  
 as  
 keys  
 with  
 val-  
 ues  
 be-  
 ing

dictionaries having indicator IDs as keys and indicator properties as values.



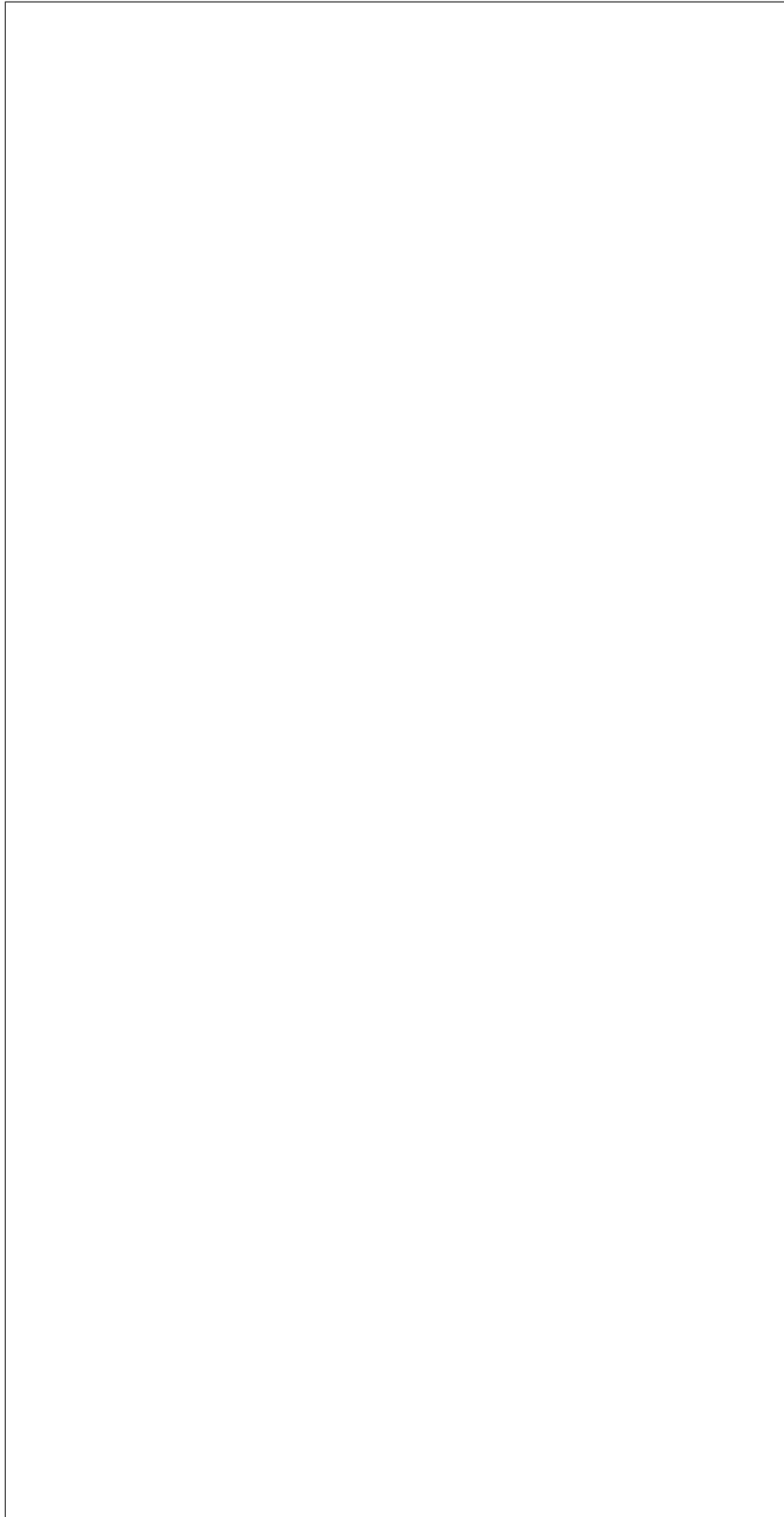
(continues on next page)

(continued from previous page)



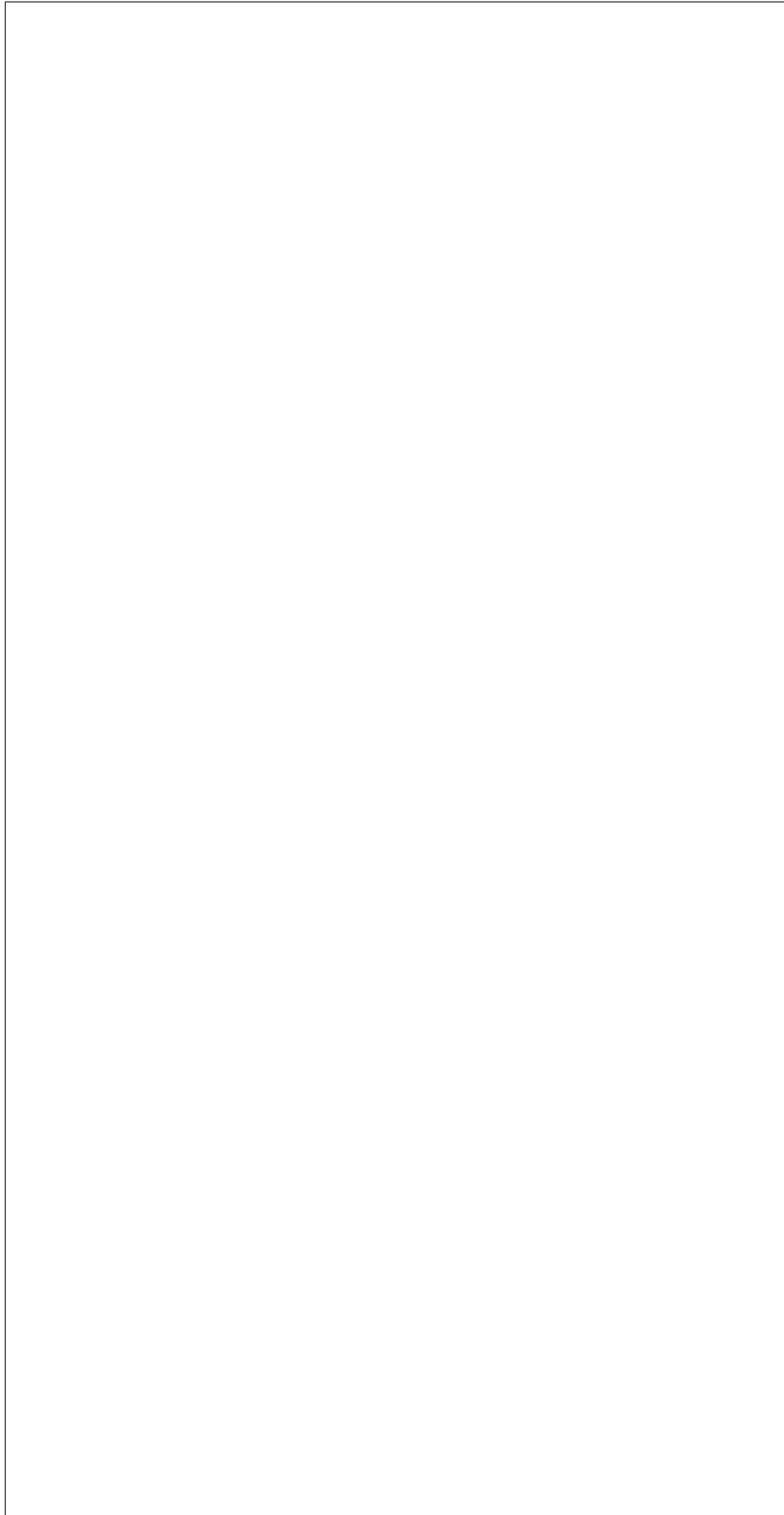
(continues on next page)

(continued from previous page)



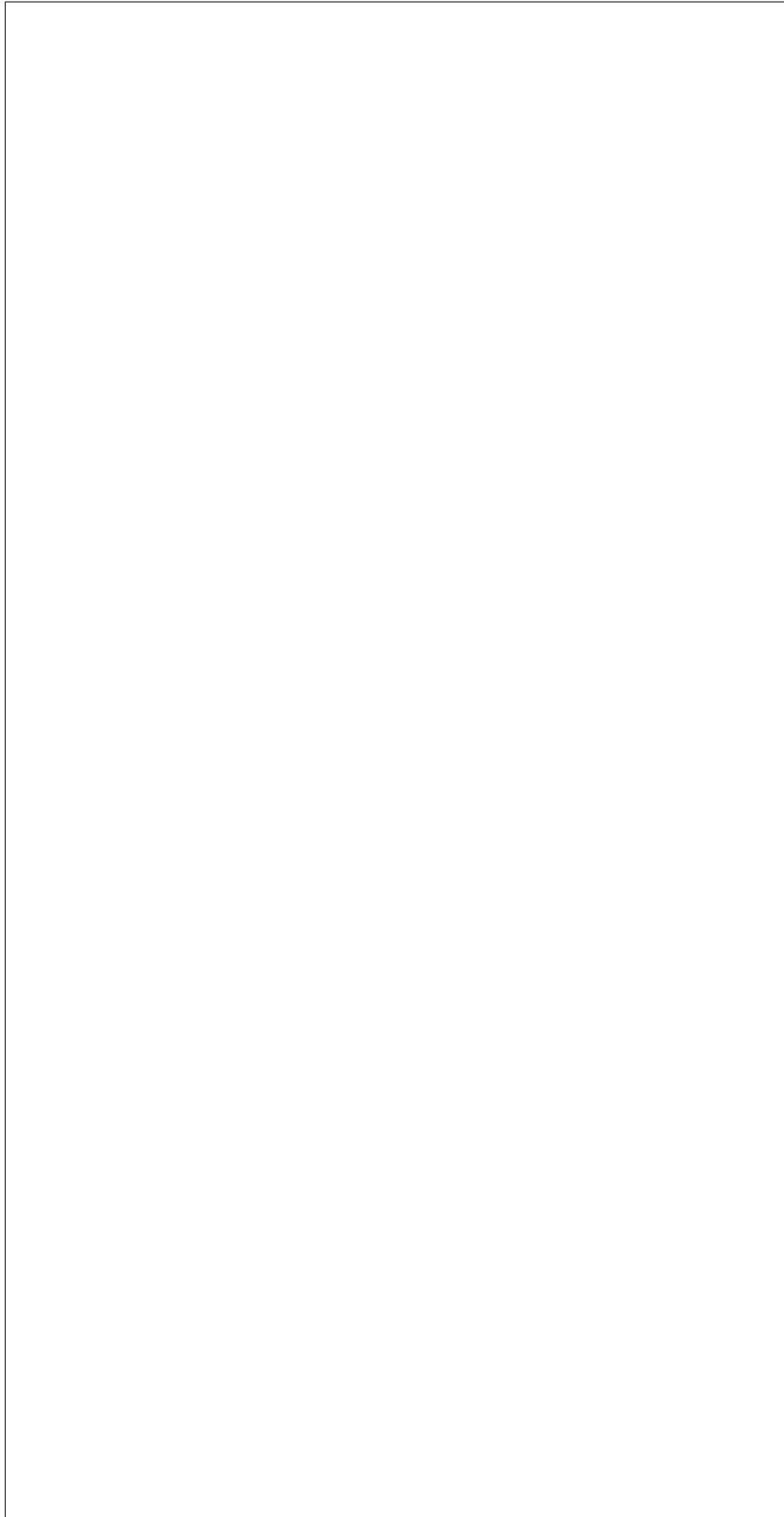
(continues on next page)

(continued from previous page)



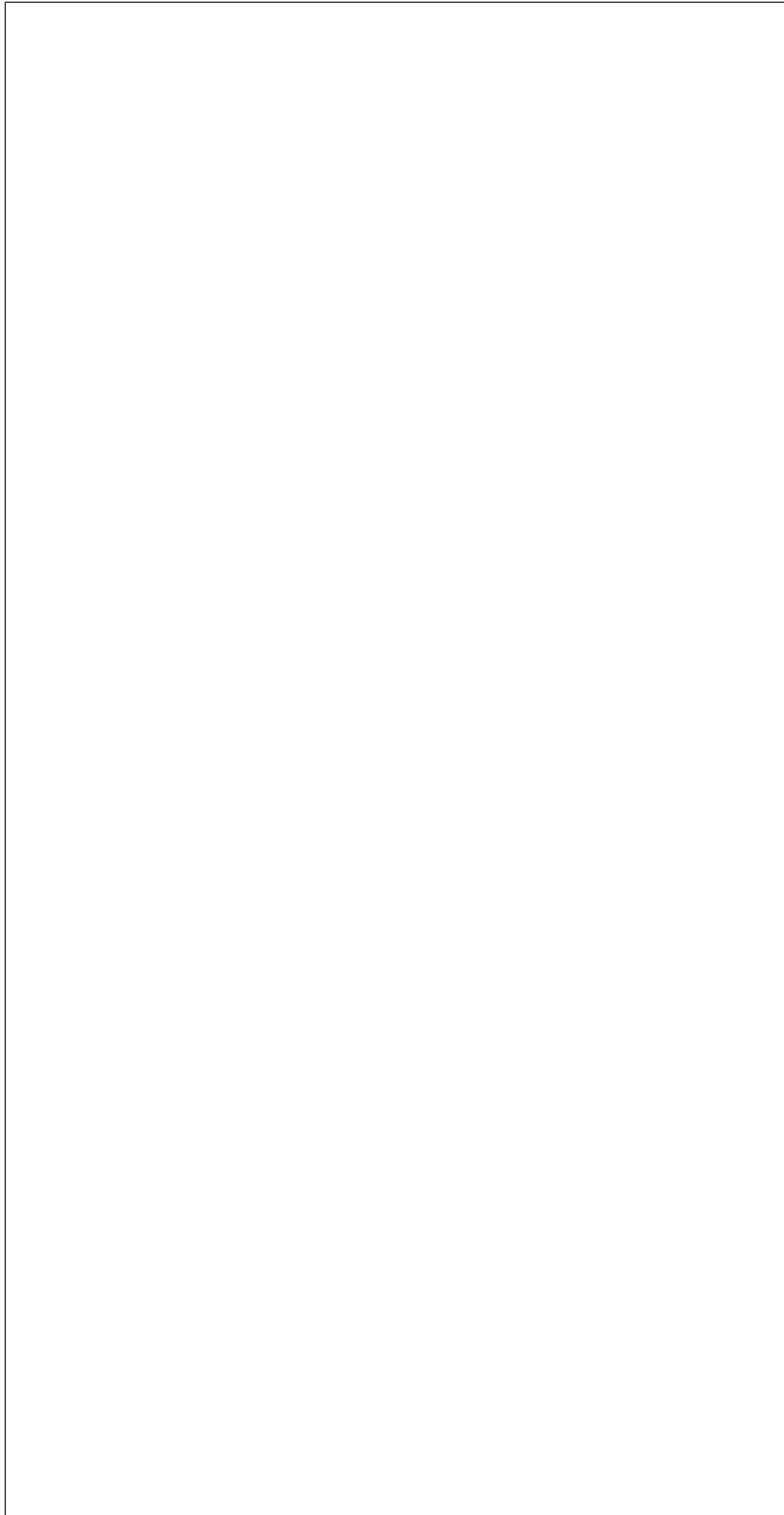
(continues on next page)

(continued from previous page)



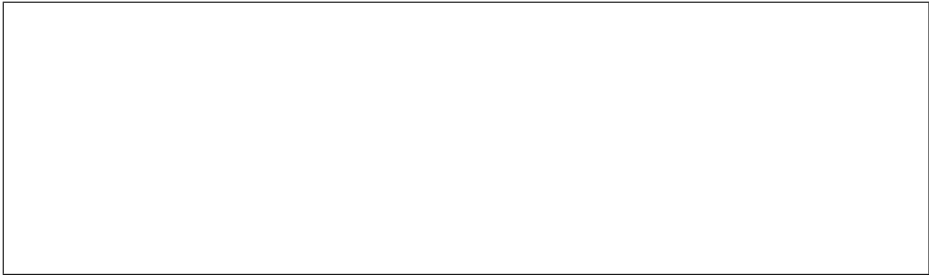
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



inject\_

In-  
 ject  
 NM  
 Non  
 Mas  
 able  
 In-  
 ter-  
 rupt  
  
 In-  
 ject  
 NM  
 (Nor  
 Mas  
 able  
 In-  
 ter-  
 rupt  
 for  
 a  
 node  
 im-  
 me-  
 di-  
 ately

Parame

tas  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**

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  
Red  
fish

**Raises**

Red  
fish-  
Er-



ror  
 on  
 an  
 er-  
 ror  
 from  
 the  
 Sush  
 li-  
 brary

**reset\_s**

Re-  
 set  
 se-  
 cure  
 boot  
 keys  
 to  
 man  
 u-  
 fac-  
 tur-  
 ing  
 de-  
 fault

**Parame**

**tas**  
 a  
 task  
 from  
 Task  
 ager

**Raises**

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 se-  
 cure  
 boot  
 is  
 now  
 sup-  
 port

**Raises**

warning is issued if it fails.

Redfish-  
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  
de-  
vice  
ac-  
cord  
ingly.  
A  
This  
meth  
is  
sup-  
pose  
to  
be  
calle  
from  
the  
Red

sidered private to the Redfish hardware type.

fish  
pow  
in-  
ter-  
face  
and  
shou  
be  
con-

## Parame

- **task**  
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

not. Default: False.

the  
 node  
**Parameters**  

- task**  
 a  
 task  
 from  
 Task  
 ager
- dev**  
 the  
 boot  
 de-  
 vice  
 one  
 of  
*ironic*  
*compute*  
*bootstrap*
- persist**  
 Boo  
 valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 Fals  
 if

**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  
 Red  
 fish

**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

Paramete

- **task**  
 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  
 spec  
 i-  
 fied.

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

Raises

Red  
 fish-  
 Con  
 nec-  
 tion-  
 Erro  
 whe  
 it  
 fails  
 to  
 con-  
 nect  
 to  
 Red  
 fish

Raises

Redfish-Error on an error from the Sushli-brary

**set\_inco**  
 Set in-di-ca-tor on the hard ware com-po-nent to the de-sired state

**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*  
*com*  
*inc*

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valid  
com  
po-  
nent  
in-  
di-  
ca-

state is specified.

tor  
or

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**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**

- 

**task**  
A  
task  
from  
Task  
ager

- 

**state**  
A  
new  
state  
as  
a  
bool

#### **Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

#### **Raises**

Red  
fish-  
Er-  
ror  
or  
its  
deri  
tive  
in  
case  
of  
a  
drive

run-time error.

**Raises**

UnsupportedDriverExtensionException if secure boot is not supported by the hardware.

**update\_**

Update the firmware on the node.

**Parameters**

- task**
 a Task object containing the node to act on.

- 

**firm**  
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**  
Val-  
i-  
date  
the  
drive  
in-  
for-  
ma-  
tion  
need  
by  
the  
red-  
fish  
drive

**Parameters**

**task**

a TaskManager instance containing the node to act on.

**Raises**

InvalidParameterException: on malformed parameter(s)

**Raises**

MissingParameterException: on missing parameter(s)

## ironic.drivers.modules.redfish.power module

**class** `i`

Base  
*irc*  
*dri*  
*bas*  
*Pow*

**get\_pow**

Get  
the  
cur-  
rent  
pow  
state  
of  
the  
task  
node

**Parame**

**tas**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**Returns**

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  
 Red  
 fish

Raises

Red  
 fish-  
 Er-  
 ror  
 on  
 an  
 er-



ror  
from  
the  
Sush  
li-  
bran

**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-

at the moment.

tain-  
ing  
the  
node  
to  
act  
on.  
Not  
used  
by  
this  
drive

Returns

A  
list  
with  
the  
sup-  
port  
pow  
state  
de-  
fine  
in  
*irc*  
*com*  
*sta*

reboot

Per-  
form  
a  
hard  
re-  
boot  
of  
the  
task  
node

Parame

- **tas**  
a  
Task  
ager  
in-  
stan  
con-

tain-  
ing  
the  
node  
to  
act  
on.

• **time**  
Time  
to  
wait  
for  
the  
node  
to  
be-  
com  
pow  
ered  
on.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

**Raises**

Red  
fish-  
Con  
nec-  
tion-  
Erro  
whe  
it  
fails

to  
 con-  
 nect  
 to  
 Red  
 fish

**Raises**

Red  
 fish-  
 Er-  
 ror  
 on  
 an  
 er-  
 ror  
 from  
 the  
 Sush  
 li-  
 brary

**set\_pow**

Set  
 the  
 pow  
 state  
 of  
 the  
 task  
 node

**Parame**

•

**tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

•

**pow**  
 Any

pow  
state  
from  
*irc*  
*com*  
*sta*

- **tim**  
Time  
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-  
e-  
ter  
is  
miss  
ing.

**Raises**

Red  
fish-  
Con  
nec-  
tion-  
Erro  
whe  
it  
fails  
to

con-  
nect  
to  
Red  
fish

**Raises**

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-  
fish  
drive

**Parame**

**tas**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**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)

ironic.drivers.modules.redfish.utils module

**class** i

Base  
obj  
  
Cach  
of  
HTT  
ses-  
sion  
cre-  
den-  
tials

**AUTH\_CL**

ironic.

Get  
 a  
 Redfish  
 fish  
 Sys-  
 tem  
 that  
 rep-  
 re-  
 sent  
 a  
 node

**Parameter**  
**node**  
 an  
 Iron  
 node  
 ob-  
 ject

**Raises**  
 Redfish-  
 Con-  
 nec-  
 tion-  
 Error  
 whe  
 it  
 fails  
 to  
 con-  
 nect  
 to  
 Redfish

**Raises**  
 Redfish-  
 Er-  
 ror  
 if  
 the  
 Sys-  
 tem  
 is  
 not  
 reg-  
 is-  
 terec  
 in



Redfish
 ironic.
 Get
 a
 node
 up-
 date
 ser-
 vice

**Parameters**
**node**
an
Ironi
c node
ob-
ject

**Raises**
Redfish
fish-
Connec-
tion-
Error
when
it
fails
to
con-
nect
to
Redfish
fish

**Raises**
Redfish
fish-
Error
when
the
Update
Service
is
not
regis-
tered

in  
 Redfish  
 fish  
 ironic.  
 Pars  
 the  
 in-  
 for-  
 ma-  
 tion  
 re-  
 quir  
 for  
 Iron  
 to  
 con-  
 nect  
 to  
 Red  
 fish.

**Parameters**  
**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-  
 ter(s)  
**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 miss  
 ing  
 pa-  
 ram-  
 e-  
 ter(s)

ironic.drivers.modules.redfish.vendor module

Ven-  
 dor  
 In-  
 ter-  
 face  
 for  
 Red  
 fish  
 drive  
 and  
 its  
 sup-  
 port  
 ing  
 meth  
 ods.

**class** i  
 Base  
*irc*  
*dri*  
*bas*  
*Ven*  
 Ven  
 spec  
 in-  
 ter-  
 face

vice to eject

for  
 Red  
 fish  
 drive

**eject\_v**  
 Eject  
 a  
 vir-  
 tual  
 me-  
 dia  
 de-  
 vice

**Paramete**

- tas**  
 A  
 Task  
 ager  
 ob-  
 ject.
- kwa**  
 The  
 ar-  
 gu-  
 men-  
 sent  
 with  
 ven-  
 dor  
 pass  
 The  
 op-  
 tion:  
 kwa  
 are::  
 boot  
 the  
 boot  
 de-

**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

Che  
if  
a  
valid  
ven-  
dor  
pass  
meth  
was  
pass  
and  
val-  
i-  
date  
the  
pa-  
ram-  
e-  
ters  
for

the vendor passthru method.

Parameters

- **taskManager**  
 A TaskManager instance containing the node to act on.
- **method**  
 The method to be validated.
- **kwargs**  
 kwargs containing the vendor, password, methods, parameters, etc.

**Raises**  
 InvalidParameterException  
 ValueError

if  
any  
of  
the  
pa-  
ram-  
e-  
ters  
have  
in-  
valid  
valu

## Module contents

`ironic.drivers.modules.storage` package

## Submodules

`ironic.drivers.modules.storage.cinder` module

**class** `i`

Base  
*irc*  
*dri*  
*bas*  
*Sto*

A  
stor-  
age\_  
drive  
sup-  
port  
ing  
Cin-  
der.

**attach\_**

In-  
form  
the  
stor-  
age  
sub-  
sys-  
tem  
to

at-  
 tach  
 all  
 vol-  
 ume  
 for  
 the  
 node

**Parame**  
**tas**  
 The  
 task  
 ob-  
 ject.

**Raises**  
 Stor  
 ageE  
 ror  
 If  
 an  
 un-  
 der-  
 ly-  
 ing  
 ex-  
 cep-  
 tion  
 or  
 fail-  
 ure  
 is  
 de-  
 tecte

**detach\_**  
 In-  
 form  
 the  
 stor-  
 age  
 sub-  
 sys-  
 tem  
 to  
 de-  
 tach  
 all  
 vol-  
 ume  
 for  
 the



node  
 This  
 ac-  
 tion  
 is  
 re-  
 tried  
 in  
 case  
 of  
 fail-  
 ure.

Parame

- **task**  
 The  
 task  
 ob-  
 ject.
- **connector**  
 The  
 dic-  
 tio-  
 nary  
 rep-  
 re-  
 sent  
 ing  
 a  
 node  
 con-  
 nec-  
 tiv-  
 ity  
 as  
 de-  
 fined  
 by

\_generate\_connector(). Generated if not passed.

- **about**  
 Boo  
 rep-  
 re-  
 sent  
 ing  
 if

a failed attachment

this  
de-  
tach  
men  
was  
re-  
ques  
to  
han-  
dle  
abor  
ing

Raises

Stor  
ageE  
ror  
If  
an  
un-  
der-  
ly-  
ing  
ex-  
cep-  
tion  
or  
fail-  
ure  
is  
de-  
tecte

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

**should\_**

De-  
ter-  
mine  
if  
de-  
ploy  
shou  
per-  
form  
the  
im-  
age  
write  
out.

**Parame**

**tas**  
The  
task  
ob-  
ject.

**Returns**

True  
if  
the  
de-  
ploy  
men  
write  
out  
pro-  
cess  
shou  
be  
ex-  
e-  
cute

**validat**

Val-  
i-  
date  
stor-

enter the active state, this method performs basic checks of the volume connectors, volume targets, and operator defined capabilities. These checks are to help ensure that we should have a compatible configuration prior to activating the node.

age\_  
con-  
fig-  
u-  
ra-  
tion  
for  
Cin-  
der  
us-  
age.  
  
In  
or-  
der  
to  
pro-  
vide  
fail  
fast  
func-  
tion-  
al-  
ity  
prior  
to  
node  
be-  
ing  
re-  
ques  
to

#### **Parame**

**tas**  
The  
task  
ob-  
ject.

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
If  
a

ists that would prevent storage the cinder storage driver from initializing attachments.

### **ironic.drivers.modules.storage.external module**

**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-  
ume  
for  
the  
node

**Parameters**

**task**

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-

ume

for

the

node

**Parameters**

**task**

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.

This  
en-  
able  
the  
user  
to  
de-  
fine  
a  
vol-  
ume

may already exist and we may be booting to that volume.

and  
Iron  
un-  
der-  
stan  
that  
the  
im-  
age

**Parameters**  
**task**  
The  
task  
ob-  
ject.

**Returns**  
True  
if  
the  
de-  
ploy  
men  
writ  
out  
pro-  
cess  
shou  
be  
ex-  
e-  
cute

**validation**  
Val-  
i-  
date  
the  
drive  
spec  
Nod  
de-  
ploy  
men  
info  
  
This  
meth  
val-  
i-  
date



the required information for this interface to function.

long-running checks.

when  
the  
drive  
and/  
in-  
stan-  
prop-  
er-  
ties  
of  
the  
task-  
node  
con-  
tain-

This  
meth-  
is  
of-  
ten  
ex-  
e-  
cute  
syn-  
chro-  
in  
API  
re-  
ques-  
so  
it  
shou-  
not  
con-  
duct-

#### Paramete

task  
A  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to

act  
on.

**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

ironic.drivers.modules.storage.noop module

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Sto*  
  
No-  
op  
Stor  
age  
In-  
ter-  
face

**attach\_**  
 In-  
 form  
 the  
 stor-  
 age  
 sub-  
 sys-  
 tem  
 to  
 at-  
 tach  
 all  
 vol-  
 ume  
 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-  
 ume

for  
 the  
 node

**Parameters**  
**tasks**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

**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

**should**  
 De-  
 ter-  
 mine  
 if  
 de-

ploy  
 shou  
 per-  
 form  
 the  
 im-  
 age  
 writ  
 out.

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan

**Returns**  
 Boo  
 valu  
 to  
 in-  
 di-  
 cate  
 if  
 the  
 in-  
 ter-  
 face  
 ex-  
 pect  
 the  
 im-  
 age  
 to  
 be  
 writ

ten by Ionic.

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

**validat**  
 Val-  
 i-  
 date

the required information for this interface to function.

long-running checks.

the  
drive  
spec  
Nod  
de-  
ploy  
men  
info

This  
meth  
meth  
val-  
i-  
date  
whe  
the  
drive  
and/  
in-  
stan  
prop  
er-  
ties  
of  
the  
task  
node  
con-  
tain

This  
meth  
is  
of-  
ten  
ex-  
e-  
cute  
syn-  
chro  
in  
API  
re-  
ques  
so  
it  
shou  
not  
con-  
duct

Param

tas

A

Task

ager

in-

stan

con-

tain-

ing

the

node

to

act

on.

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)

## Module contents

`ironic.drivers.modules.xclarity` package

## Submodules

`ironic.drivers.modules.xclarity.common` module

`ironic.`

`ironic.`

Val-

i-

date

node

con-

fig-

u-

ra-

tion

and

re-

turn

xcla

ity

hard

ware

id.

Val-

i-

date

when

node

con-

figut

tion

is

con-

sis-

tent

with

XCI

ity

and

re-

turn

the

XCI

ity 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

```

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
    im-
    port
    xcla
    ity_c
    li-
    brary
    
```

#### Parameter

**node**  
 an  
 iron  
 node  
 ob-  
 ject.

#### Returns

an  
 in-  
 stan  
 of  
 the  
 XCl  
 ity

combination of both.

client  
**Raises**  
 XCI  
 i-  
 ty-  
 Er-  
 ror  
 if  
 cant  
 get  
 to  
 the  
 XCI  
 ity  
 client  
 ironic.  
 Pars  
 a  
 node  
 drive  
 val-  
 ues.  
 Pars  
 the  
 drive  
 of  
 the  
 node  
 read  
 de-  
 fault  
 val-  
 ues  
 and  
 re-  
 turn  
 a  
 dict  
 con-  
 tain-  
 ing  
 the  
**Paramet**  
**nod**  
 an  
 iron  
 node  
 ob-

the node or inputs is invalid.

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-  
 ma-  
 tion  
 is  
 miss  
 ing  
 on  
 ironic.  
 Tran  
 lates  
 iron  
 ics  
 pow  
 ac-  
 tion  
 strin

to  
XCL  
i-  
tys  
for-  
mat.

Paramet

pow  
pow  
ac-  
tion  
strin  
to  
be  
trans  
lated

Returns

the  
pow  
ac-  
tion  
trans  
lated

ironic.  
Tran  
lates  
XCL  
i-  
tys  
pow  
state  
strin  
to  
be  
con-  
sis-  
tent  
with  
Iron

Paramet

pow  
pow  
state  
strin  
to  
be  
trans  
lated

Returns

the  
 trans  
 latec  
 pow  
 state

ironic.drivers.modules.xclarity.management module

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Man*

**get\_boo**

Get  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 for  
 the  
 task  
 node

**Parame**

**tas**  
 a  
 task  
 from  
 Task  
 ager

**Returns**

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing:  
 :boo  
 the  
 boot  
 de-  
 vice  
 one

sistent: Whether the boot device will persist or not It returns None if boot device is unknown.

of  
[PX  
DIS  
CDF  
BIO  
:per

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
boot  
de-  
vice  
is  
un-  
know

#### **Raises**

XCI  
i-  
ty-  
Er-  
ror  
if  
the  
com  
mu-  
ni-  
ca-  
tion  
with  
XCI  
ity  
fails

#### **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  
 de-  
 fine  
 in  
*irc*  
*com*  
*boo*

**set\_boot**

Sets  
 the  
 boot  
 de-  
 vice  
 for  
 a  
 node

**Paramete**

- **task**  
 a  
 task  
 from  
 Task  
 ager
- **dev**  
 the  
 boot  
 de-  
 vice  
 one  
 of  
 the  
 sup-  
 port



de-  
vice  
liste  
in  
*irc*  
*com*  
*boo*

- **per**  
Boo  
valu  
True  
if  
the  
boot  
de-  
vice  
will  
per-  
sist  
to  
all  
fu-  
ture  
boot  
Fals  
if

not. Default: False.

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valid  
boot  
de-  
vice  
is  
spec  
i-  
fied.

#### **Raises**

XCI  
i-

quired information for this driver to manage the node.

ty-  
 Er-  
 ror  
 if  
 the  
 com  
 mu-  
 ni-  
 ca-  
 tion  
 with  
 XCL  
 ity  
 fails  
  
**validat**  
 Val-  
 i-  
 date  
 the  
 drive  
 spec  
 info  
 sup-  
 plied  
  
 This  
 meth  
 val-  
 i-  
 date  
 if  
 the  
 drive  
 prop  
 erty  
 of  
 the  
 sup-  
 plied  
 task  
 node  
 con-  
 tains  
 the  
 re-  
  
**Parame**  
**tas**  
 a  
 task  
 from

Task  
ager

## ironic.drivers.modules.xclarity.power module

**class** `ironic.drivers.modules.xclarity.power.XclarityPower`

**get\_power**  
Gets the current power state

**Parameter:**  
**task**  
a Taskager instance

**Returns:**  
one of `ironic.common.STATES.POWER_ON` or `ironic.common.STATES.POWER_OFF`

**Raises:**  
`XclarityError` if fails to retrieve

pow  
state  
of  
XCL  
ity  
re-  
sour

**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**

Soft  
re-  
boot  
the  
node

**Parame**

- **task**  
a  
Task  
ager  
in-  
stan
-

**tim**  
time  
out  
(in  
sec-  
onds  
Un-  
sup-  
port  
by  
this  
in-  
ter-  
face

**set\_pow**  
Turn  
the  
cur-  
rent  
pow  
state  
on  
or  
off.

#### Parame

- **tas**  
a  
Task  
ager  
in-  
stan
- **pow**  
The  
de-  
sired  
pow  
state  
POV  
POV  
or  
RE-  
BOC  
from  
*irc*  
*com*  
*sta*

• **time**  
 time  
 out  
 (in  
 sec-  
 onds  
 Un-  
 sup-  
 port  
 by  
 this  
 in-  
 ter-  
 face

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 pow  
 state  
 was  
 spec  
 i-  
 fied.

**Raises**  
 XCL  
 i-  
 ty-  
 Er-  
 ror  
 if  
 XCL  
 ity  
 fails  
 set-  
 ting  
 the  
 pow  
 state

**validat**  
 Val-

quired information for this driver to manage the power state of the node.

Module contents

Submodules

ironic.drivers.modules.agent module

i-  
 date  
 the  
 drive  
 spec  
 info  
 sup-  
 plied  
  
 This  
 meth  
 val-  
 i-  
 date  
 if  
 the  
 drive  
 prop  
 erty  
 of  
 the  
 sup-  
 plied  
 task  
 node  
 con-  
 tains  
 the  
 re-

Paramet  
 tas  
 a  
 task  
 from  
 Task  
 agen

class i  
 Base  
 irc  
 dri  
 mod  
 age

*Age*  
*irc*  
*dri*  
*moo*  
*age*  
*Age*  
*irc*  
*dri*  
*bas*  
*Dep*

**In-**  
**ter-**  
**face**  
**for**  
**depl**  
**relat**  
**ac-**  
**tions**

**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  
ahead  
of  
time  
is



this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

pos-  
si-  
ble,

If  
im-  
ple-  
men  
this  
meth  
mus  
be  
iden  
po-  
tent.  
It  
may  
be  
calle  
mul-  
ti-  
ple  
time  
for

This  
meth  
is  
calle  
be-  
fore  
*tear*

**Parame**  
**tas**  
a  
Task  
ager  
in-  
stan

**deploy**  
Per-  
form  
a  
de-  
ploy  
men  
to

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

a  
 node  
  
 Per-  
 form  
 the  
 nec-  
 es-  
 sary  
 work  
 to  
 de-  
 ploy  
 an  
 im-  
 age  
 onto  
 the  
 spec  
 i-  
 fied  
 node  
 This

**Parameters**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan

**Returns**  
 sta-  
 tus  
 of  
 the  
 de-  
 ploy  
 One  
 of  
 iron

**get\_properties**  
 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

new cleaning ports cannot be created.

driver info is specified for power management.

ing  
port  
can-  
not  
be  
re-  
mov  
or  
if

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
pow  
state  
is  
spec  
i-  
fied  
or  
the  
wron

**Raises**

Stor  
ageE  
ror  
If  
the  
stor-  
age  
drive  
is  
un-  
able  
to  
at-  
tach  
the  
con-  
fig-  
ured

umes.

action.

vol-  
**Raises**  
 othe  
 ex-  
 cep-  
 tion:  
 by  
 the  
 node  
 pow  
 drive  
 if  
 som  
 thing  
 wron  
 oc-  
 curre  
 dur-  
 ing  
 the  
 pow  
**Raises**  
 ex-  
 cep-  
 tion.  
 if  
 im-  
 age\_  
 is  
 not  
 Glar  
 href  
 and  
 is  
 not  
 HTT  
 URI  
**Raises**  
 ex-  
 cep-  
 tion.  
 if  
 net-  
 worl  
 val-  
 i-  
 da-  
 tion

fails

**Raises**

any  
 boot  
 in-  
 ter-  
 face  
 pre-  
 pare  
 ex-  
 cep-  
 tions

**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  
 prop  
 er-  
 ties  
 of  
 the  
 sup-  
 plied  
 node  
 con-

information for this driver to deploy images to the node.

ing.

tain  
the  
re-  
quir

**Parame**  
**tas**  
a  
Task  
ager  
in-  
stan

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
any  
of  
the  
re-  
quir  
pa-  
ram-  
e-  
ters  
are  
miss

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
any  
of  
the  
pa-  
ram-  
e-  
ters  
have

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.



**class** **i**  
 Base  
*irc*  
*dri*  
*bas*  
*RAI*  
 Im-  
 ple-  
 men-  
 ta-  
 tion  
 of  
 RAI  
 In-  
 ter-  
 face  
 which  
 uses  
 ager  
 ram

**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-

creating the new configuration.

u-  
 ra-  
 tion  
 to  
 ap-  
 ply.

•

**del**  
 Set-  
 ting  
 this  
 to  
 True  
 in-  
 di-  
 cate  
 to  
 dele  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 prior  
 to

**Raises**  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in-  
 valio

**Returns**  
 state  
 if  
 RAI

plete.

create\_

Cre-  
 ate  
 a  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 a  
 bare  
 meta-  
 us-  
 ing  
 ager  
 rame  
 This  
 meth  
 cre-  
 ates  
 a  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give

otherwise, no root volume is created. Default is True.

node

**Parameters**

- task**  
 a Task object representing the task in the instance
- create\_root\_volume**  
 If True, a root volume is created during RAI configuration. Otherwise,
- create\_non\_root\_volumes**  
 If True, non-root volumes are created. If False, no non-root volumes

ated. Default is True.

ter skipping root volume and/or non-root volumes.

are  
cre-

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  
was  
foun  
to  
be  
emp  
af-

delete\_

Dele  
RAI  
con-  
fig-  
u-  
ra-  
tion  
on  
the  
give  
node

**Parameters**

**task**

a

Task

ager

in-

stan

**Returns**

state

if

op-

er-

a-

tion

was

suc-

cess

fully

in-

voke

**get\_dependencies**

Get

the

list

of

de-

ploy

step

from

the

ager

**Parameters**

**task**

a

Task

ager

ob-

ject

con-

tain-

ing

the

node

**Raises**

*ImportError*

if

the

de-

ploy

has just been enrolled and has not been deployed yet.

step  
are  
not  
yet  
avai  
able  
(cac  
for  
ex-  
am-  
ple,  
whe  
a  
node

#### 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

cleaned if Ironic is managing the ramdisk boot.

of  
 Res-  
 cueI  
 ter-  
 face  
 whic  
 uses  
 ager  
 rame  
  
**clean\_u**  
 Clea  
 up  
 af-  
 ter  
 RES  
 CUE  
 WAI  
 time  
 out/  
 or  
 fin-  
 ish-  
 ing  
 res-  
 cue.  
  
 Res-  
 cue  
 pass  
 wor  
 shou  
 be  
 re-  
 mov  
 from  
 the  
 node  
 and  
 rame  
 boot  
 en-  
 vi-  
 ron-  
 men  
 shou  
 be  
  
**Parame**  
**tas**  
 a



Task  
 ager  
 in-  
 stan-  
 with  
 the  
 node

**Raises**

Net-  
 worl  
 Er-  
 ror  
 if  
 the  
 res-  
 cue  
 port  
 can-  
 not  
 be  
 re-  
 mov

**get\_pro**

Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

**rescue**

Boo  
 a  
 res-  
 cue  
 ranc  
 on  
 the  
 node

**Parame**

**tas**  
 a  
 Task  
 ager  
 in-

driver info is specified for power management.

stan  
**Raises**  
 Net-  
 worl  
 Er-  
 ron  
 if  
 the  
 ten-  
 ant  
 port  
 can-  
 not  
 be  
 re-  
 mov  
**Raises**  
 In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 whe  
 the  
 wron  
 pow  
 state  
 is  
 spec  
 i-  
 fied  
 or  
 the  
 wron  
**Raises**  
 othe  
 ex-  
 cep-  
 tions  
 by  
 the  
 node  
 pow  
 drive  
 if  
 som  
 thing

action.

wron
 oc-
 curr
 dur-
 ing
 the
 pow

Raises

any
 boot
 in-
 ter-
 face
 pre-
 pare
 ex-
 cep-
 tion:

Returns

Re-
 turn
 state

unrescu

At-
 temp
 to
 mov
 a
 res-
 cued
 node
 back
 to
 ac-
 tive
 state

Paramete

tas
 a
 Task
 ager
 in-
 stan

Raises

Net-
 worl
 Er-
 ror

driver info is specified for power management.

if  
the  
res-  
cue  
port.  
can-  
not  
be  
re-  
mov

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
the  
wron  
pow  
state  
is  
spec  
i-  
fied  
or  
the  
wron

**Raises**

othe  
ex-  
cep-  
tions  
by  
the  
node  
pow  
drive  
if  
som  
thing  
wron  
oc-  
curr  
dur-  
ing  
the

action.

pow  
**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  
 ager  
 res-  
 cue.

**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 with  
 the  
 node  
 be-  
 ing  
 chec

**Raises**  
 In-

work UUID config option has an invalid value.

eters

valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
in-  
stan-  
has  
emp  
pass  
wor  
or  
res-  
cu-  
ing  
net-

#### Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
node  
is  
miss  
ing  
one  
or  
more  
re-  
quir  
pa-  
ram-

ironic.

Che  
if  
the  
re-  
ques  
im-  
age  
is

large  
 than  
 the  
 ram  
 size.

Paramet

- **task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

- **image**  
 href  
 of  
 the  
 im-  
 age.

- **image**  
 The  
 disk  
 for-  
 mat  
 of  
 the  
 im-  
 age  
 if  
 pro-  
 vide

Raises

In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-

ram size.

ing.

Valu  
if  
size  
of  
the  
im-  
age  
is  
grea  
than  
the  
avai  
able

ironic.  
Val-  
i-  
date  
con-  
fig-  
u-  
ra-  
tion  
op-  
tions  
re-  
quir  
to  
per-  
form  
HTT  
pro-  
vi-  
sion

Paramet  
nod  
an  
iron  
node  
ob-  
ject

Raises  
Miss  
ing-  
Pa-  
ram  
e-  
ter-  
Valu



if  
 re-  
 quir  
 op-  
 tion  
 is  
 not  
 set.

ironic.  
 Che  
 that  
 the  
 pro-  
 vide  
 prox  
 pa-  
 ram-  
 e-  
 ters  
 are  
 valid

**Paramet**  
**nod**  
 an  
 Iron  
 node

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 any  
 of  
 the  
 pro-  
 vide  
 prox  
 pa-  
 ram-  
 e-  
 ters  
 are

incorrect.

## ironic.drivers.modules.agent\_base module

```
class i
    Base
    obj
    Mix
    with
    base
    meth
    ods
    not
    re-
    ly-
    ing
    on
    any
    de-
    ploy
    step
```

```
clean_u
    Clea
    up
    the
    de-
    ploy
    men
    en-
    vi-
    ron-
    men
    for
    the
    task
    node
    Un-
    links
    TFT
    and
    in-
    stan
    im-
    ages
    and
    trig-
    gers
    im-
    age
    cach
```

figuration files for this node.

clear  
Re-  
mov  
the  
TFT  
con-

#### Parameters

**task**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

#### prepare

Boo  
into  
the  
ager  
to  
pre-  
pare  
for  
clear  
ing.

#### Parameters

**task**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

#### Raises

Nod  
Clea  
ing-

be removed or if new cleaning ports cannot be created.

invalid value.

Fail-  
ure,  
Net-  
worl  
Er-  
ror  
if  
the  
pre-  
vi-  
ous  
clea  
ing  
port  
can-  
not

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
clea  
ing  
net-  
worl  
UUI  
con-  
fig  
op-  
tion  
has  
an

Returns

state  
to  
sig-  
nify  
an  
asyn  
chro  
pre-  
pare

should  
Whe

ager  
boot  
is  
man  
agec  
by  
iron

#### **take\_ov**

Take  
over  
man  
age-  
men  
of  
this  
node  
from  
a  
deac  
con-  
duc-  
tor.

#### **Parame**

**tas**  
a  
Task  
ager  
in-  
stan

#### **tear\_d**

Tear  
dow  
a  
pre-  
vi-  
ous  
de-  
ploy  
men  
on  
the  
task  
node  
  
Pow  
off  
the  
node  
All  
ac-

arately.

tual  
 clea  
 up  
 is  
 done  
 in  
 the  
 clea  
 meth  
 whic  
 shou  
 be  
 calle  
 sep-  
  
**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**Returns**  
 de-  
 ploy  
 state  
 DEL  
  
**Raises**  
 Net-  
 worl  
 Er-  
 ror  
 if  
 the  
 clea  
 ing  
 port  
 can-  
 not  
 be  
 re-  
 mov

info is specified.

Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 whe  
 the  
 wron  
 state  
 is  
 spec  
 i-  
 fied  
 or  
 the  
 wron  
 drive

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  
 oc-  
 curr  
 dur-  
 ing

action.

the  
 pow  
**tear\_down**  
 Clea  
 up  
 the  
 PXE  
 and  
 DHCP  
 files  
 af-  
 ter  
 clea  
 ing.  
**Parameter**  
**task**  
 a  
 Task  
 ager  
 ob-  
 ject  
 con-  
 tain-  
 ing  
 the  
 node  
**Raises**  
 Nod  
 Clea  
 ing-  
 Fail-  
 ure,  
 Net-  
 worl  
 Er-  
 ror  
 if  
 the  
 clea  
 ing  
 port  
 can-  
 not  
 be  
 re-  
 mov  
**class** i  
 Base



*irc*  
*dri*  
*moo*  
*age*  
*Hea*  
*irc*  
*dri*  
*moo*  
*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  
suc-  
cess  
ful  
in-  
stal-

tion of bootloader, this method sets the node to boot from disk.

tion which contains the image deployed or None in case of whole disk images which we expect to already have a bootloader installed.

ware.

tion.  
 This  
 is  
 used  
 only  
 in  
 uefi  
 boot  
 mod

- pre**  
 The  
 UUI  
 of  
 the  
 PRe  
 Boo  
 par-  
 ti-  
 tion.  
 This  
 is  
 used  
 only  
 for  
 boot  
 ing  
 ppc6  
 hard

**Raises**  
 In-  
 stan-  
 ploy  
 Fail-  
 ure  
 if  
 boot  
 load  
 in-  
 stal-  
 la-  
 tion  
 faile  
 or  
 on  
 en-  
 cour  
 ter-  
 ing

error while setting the boot device on the node.

**execute**

Ex-  
e-  
cute  
a  
clea  
step  
asyn  
chro  
on  
the  
ager

**Paramete**

•

**tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

•

**ste**  
a  
clea  
step  
dic-  
tio-  
nary  
to  
ex-  
e-  
cute

**Raises**

Nod  
Clea  
ing-  
Fail-  
ure  
if  
the  
ager  
does

not  
re-  
turn  
a  
com  
man  
sta-  
tus

#### **Returns**

state  
to  
sig-  
nify  
the  
step  
will  
be  
com  
plete  
asyn

#### **execute**

Ex-  
e-  
cute  
a  
de-  
ploy  
step

Wer  
try-  
ing  
to  
find

a  
step  
amo  
both  
out-  
of-  
band  
and  
in-  
band  
step

In  
case  
of  
du-

plicates, out-of-band steps take priority. This property allows having an out-of-band deploy step that calls into a corresponding in-band step after some preparation (e.g. with additional input).

Parameters

- **task**  
 a Task object containing the node
- **step**  
 a deployment step dictionary to execute

Raises

Instance of Failure if the agent does not return a command status

Returns

state to sig-

nify  
the  
step  
will  
be  
com  
plete  
asyn

#### get\_cle

Get  
the  
list  
of  
clea  
step  
from  
the  
ager

#### Parame

**tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

#### Raises

**Noo**  
if  
the  
clea  
step  
are  
not  
yet  
avai  
able  
(cac  
for  
ex-  
am-  
ple,  
whe  
a  
node  
has

just been enrolled and has not been cleaned yet.

Returns

A  
list  
of  
clean  
step  
dic-  
tio-  
nar-  
ies

get\_dep

Get  
the  
list  
of  
de-  
ploy  
step  
from  
the  
ager

Paramet

tas  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

Raises

*Ins*  
if  
the  
de-  
ploy  
step  
are  
not  
yet  
avai  
able  
(cac  
for  
ex-



has just been enrolled and has not been deployed yet.

am-  
ple,  
whe  
a  
node

#### Returns

A  
list  
of  
de-  
ploy  
step  
dic-  
tio-  
nar-  
ies

#### prepare

Pre-  
pare  
in-  
stan-  
to  
boot

#### Parame

- **task**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node
- **root**  
the  
UUID  
for  
root  
par-  
ti-  
tion

• **efi**  
 the  
 UUI  
 for  
 the  
 efi  
 par-  
 ti-  
 tion

**Raises**

In-  
 valid  
 State  
 if  
 fails  
 to  
 pre-  
 pare  
 in-  
 stan

**process**

Star  
 the  
 next  
 clea  
 step  
 if  
 the  
 pre-  
 vi-  
 ous  
 one  
 is  
 com  
 plete  
  
 In  
 or-  
 der  
 to  
 avoi  
 er-  
 rors  
 and  
 mak  
 ager  
 up-  
 grad  
 pain  
 less,

sion 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.

coordinate the reboot once the step is completed.

the  
ager  
com  
pare  
the  
ver-

Ad-  
di-  
tion-  
ally,  
if  
a  
step  
in-  
clud  
the  
re-  
boot  
prop  
erty  
set  
to  
True  
this  
meth  
will

**reboot\_**

Help  
meth  
to  
trig-  
ger  
re-  
boot  
on  
the  
node  
and  
fin-  
ish  
de-  
ploy

This  
meth  
ini-

complete. On failure, it logs the error and marks deploy as failure.

ti-  
ates  
a  
re-  
boot  
on  
the  
node  
On  
suc-  
cess  
it  
mark  
the  
de-  
ploy  
as

**Parameters**  
**task**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

**Raises**  
In-  
stan-  
ce  
Fail-  
ure,  
if  
node  
re-  
boot  
faile

**refresh**  
Re-  
fresh  
the  
node  
cach  
clea  
step  
from

chronous, and should be refreshed as soon as the agent boots to start cleaning/deploy or if cleaning is restarted because of a hardware manager version mismatch.

the  
boot  
agen  
  
Gets  
the  
node  
step  
from  
the  
boot  
agen  
and  
cach  
then  
The  
step  
are  
cach  
to  
mak  
get\_  
calls  
syn-

## Parame

- **tas**  
a  
Task  
agen  
in-  
stan
- **ste**  
clea  
or  
de-  
ploy

## Raises

Nod  
Clea  
ing-  
Fail-  
ure  
or  
In-  
stan

sults

ploy  
Fail-  
ure  
if  
the  
ager  
re-  
turn  
in-  
valid  
re-

**tear\_down**  
A  
de-  
ploy  
step  
to  
tear  
down  
the  
ager

**Parameter**  
**task**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node

**class** **ironic**  
Base  
obj  
  
Mix  
with  
out-  
of-  
band  
de-  
ploy  
step

**boot\_image**  
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.

**continu**

Star  
the  
next  
clea  
ing  
step  
if  
the  
pre-  
vi-  
ous  
one  
is  
com  
plete

**Parame**

**tas**  
a  
Task  
ager  
in-  
stan

**continu**

Con  
tin-  
ues  
the  
de-  
ploy  
men  
of  
bare  
node  
  
This  
meth  
con-



tin-  
 ues  
 the  
 de-  
 ploy  
 men  
 of  
 the  
 bare  
 node  
 af-  
 ter  
 the  
 rame  
 have  
 been  
 boot

**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan

**deploy\_**  
 Che  
 if  
 the  
 de-  
 ploy  
 men  
 has  
 start  
 al-  
 read

**Returns**  
 True  
 if  
 the  
 de-  
 ploy  
 has  
 start  
 Fals  
 oth-  
 er-  
 wise

**deploy\_**  
 Che

if  
 the  
 de-  
 ploy  
 men  
 is  
 al-  
 read  
 com  
 plete

**Returns**

True  
 if  
 the  
 de-  
 ploy  
 men  
 is  
 com  
 plete  
 Fals  
 oth-  
 er-  
 wise

**has\_dec**

Whe  
 the  
 drive  
 sup-  
 port  
 de-  
 com  
 pose  
 de-  
 ploy  
 step  
 Pre-  
 vi-  
 ousl  
 (sinc  
 Roc  
 drive  
 used  
 a  
 sin-  
 gle  
 de-  
 ploy  
 de-  
 ploy

terface. 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.

step  
on  
the  
de-  
ploy  
in-

## heartbe

Pro-  
cess  
a  
hear  
beat

## Paramete

- **task**  
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  
 ager

**property**

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

**Parameter**

**task**  
 a  
 Task  
 ager  
 in-  
 stan

**Returns**

True  
 if  
 the  
 cur-

rent  
de-  
ploy  
step  
is  
de-  
ploy

#### process

Start  
the  
next  
clear  
step  
if  
the  
pre-  
vi-  
ous  
one  
is  
com-  
plete

#### Paramet

- **task**  
a  
Task  
ager  
in-  
stan
- **step**  
clear  
or  
de-  
ploy

#### reboot\_

Met  
in-  
voke  
af-  
ter  
the  
de-  
ploy  
men  
is  
com

plete  
**Paramet**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan

**refresh**  
 Re-  
 fresl  
 the  
 node  
 cach  
 clea  
 step

**Paramet**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan

**refresh**  
 Re-  
 fresl  
 the  
 node  
 cach  
 clea  
 step

**Paramet**

- tas**  
 a  
 Task  
 ager  
 in-  
 stan
- ste**  
 clea  
 or  
 de-  
 ploy

ironic.

ironic.

Ex-  
e-  
cute  
a  
clear  
or  
de-  
ploy  
step  
asyn  
chro  
on  
the  
ager

#### Paramet

- **tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node
- **ste**  
a  
step  
dic-  
tio-  
nary  
to  
ex-  
e-  
cute
- **ste**  
clear  
or  
de-  
ploy
-

does not return a command status.

cli  
 ager  
 clien  
 (if  
 avai  
 able

Raises

Nod  
 Clea  
 ing-  
 Fail-  
 ure  
 (clea  
 step  
 or  
 In-  
 stan  
 ploy  
 Fail-  
 ure  
 (de-  
 ploy  
 step  
 if  
 the  
 ager

Returns

state  
 to  
 sig-  
 nify  
 the  
 step  
 will  
 be  
 com  
 plete  
 asyn

ironic.

Find  
 the  
 give  
 in-  
 banc  
 step

ironic.



Get  
the  
list  
of  
cach  
clea  
or  
de-  
ploy  
step  
from  
the  
ager

The  
step  
cach  
is  
up-  
date  
at  
the  
be-  
gin-  
ning  
of  
clea  
ing  
or  
de-  
ploy

#### Paramet

- **tas**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-  
ing  
the  
node
- **ste**  
clea  
or  
de-

vided, it returns the steps for all interfaces.

ities for them. If a step isnt in this dictionary, the steps original priority is used.

Returns  
 A  
 list  
 of  
 clea  
 step  
 dic-

ploy  
 •  
 int  
 The  
 in-  
 ter-  
 face  
 for  
 whic  
 clea  
 step  
 are  
 to  
 be  
 re-  
 turn  
 If  
 this  
 is  
 not  
 pro-  
 •  
 ove  
 a  
 dic-  
 tio-  
 nary  
 with  
 keys  
 be-  
 ing  
 step  
 nam  
 and  
 val-  
 ues  
 be-  
 ing  
 new  
 pri-  
 or-

tio-  
nar-  
ies  
  
ironic.

Help  
meth  
to  
log  
the  
er-  
ror  
and  
raise  
ex-  
cep-  
tion.

#### Paramet

- **task**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **msg**  
the  
mes-  
sage  
to  
set  
in  
last\_  
of  
the  
node

faults to True. Actual log collection is also affected by CONF.agent.deploy\_logs\_collect config option.

•  
**col**  
Boo  
in-  
di-  
cat  
ing  
whe  
to  
at-  
temp  
to  
col-  
lect  
logs  
from  
IPA-  
base  
ram  
De-  
  
•  
**exc**  
Ex-  
cep-  
tion  
that  
caus  
the  
fail-  
ure.  
  
ironic.  
  
Dec  
o-  
ra-  
tor  
meth  
for  
addi  
a  
post  
clea  
step  
hool  
  
This  
is  
a  
mec  
a-

step. The hook will get executed after the clean step gets executed successfully. The hook is not invoked on failure of the clean step.

terface and step after which the hook should be executed. A TaskManager instance and the object for the last completed command (provided by agent) will be passed to the hook method. The return value of this method will be ignored. Any exception raised by this method will be treated as a failure of the clean step and the node will be moved to CLEANFAIL state.

#### Parameters

- **int**  
name  
of  
the  
in-  
ter-

face

- **steps**  
The name of the step after which it should be executed

**Returns**

A method which registers the given method as a post-clear step hook

`ionic.`

Decorator for adding a post-deploy step hook

This

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.

terface and step after which the hook should be executed. A TaskManager instance and the object for the last completed command (provided by agent) will be passed to the hook method. The return value of this method will be ignored. Any exception raised by this method will be treated as a failure of the deploy step and the node will be moved to DEPLOYFAIL state.

#### Parameters

- **int**  
nam

of  
the  
in-  
ter-  
face

- step**  
The  
name  
of  
the  
step  
af-  
ter  
which  
it  
shou  
be  
ex-  
e-  
cute

**Returns**

A  
meth  
which  
reg-  
is-  
ters  
the  
give  
meth  
as  
a  
post  
de-  
ploy  
step  
hool

ironic.drivers.modules.agent\_client module

```

class i
    Base
    obj
    Clie
    for
    in-
    ter-
    act-

```



ing  
with  
node  
via  
a  
RES  
API

#### collect

Col-  
lect  
and  
pack  
age  
di-  
ag-  
nos-  
tic  
and  
sup-  
port  
data  
from  
the  
rame

#### Parame

nod  
A  
Nod  
ob-  
ject.

#### Raises

Iron  
icEx  
cep-  
tion  
whe  
faile  
to  
is-  
sue  
the  
re-  
ques  
or  
there  
was  
a  
mal-  
form

sponse from the agent.

ing the prior command.

re-  
  
**Raises**  
 Age  
 tAPI  
 ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man  
  
**Raises**  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres  
 ex-  
 e-  
 cut-  
  
**Returns**  
 A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man  
 re-  
 spor  
 from

sample.

ager  
 See  
*get*  
 for  
 a  
 com  
 man  
 re-  
 sult  
  
**execute**  
 Ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 clear  
 step  
  
**Parame**  
  
 •  
  
**ste**  
 A  
 clear  
 step  
 dic-  
 tio-  
 nary  
 to  
 ex-  
 e-  
 cute  
  
 •  
  
**nod**  
 A  
 Nod  
 ob-  
 ject.  
  
 •  
  
**por**  
 Port  
 as-  
 so-  
 ci-  
 ated  
 with  
 the  
 node

sponse from the agent.

Raises  
 Iron  
 icEx  
 cep-  
 tion  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques  
 or  
 there  
 was  
 a  
 mal-  
 form  
 re-

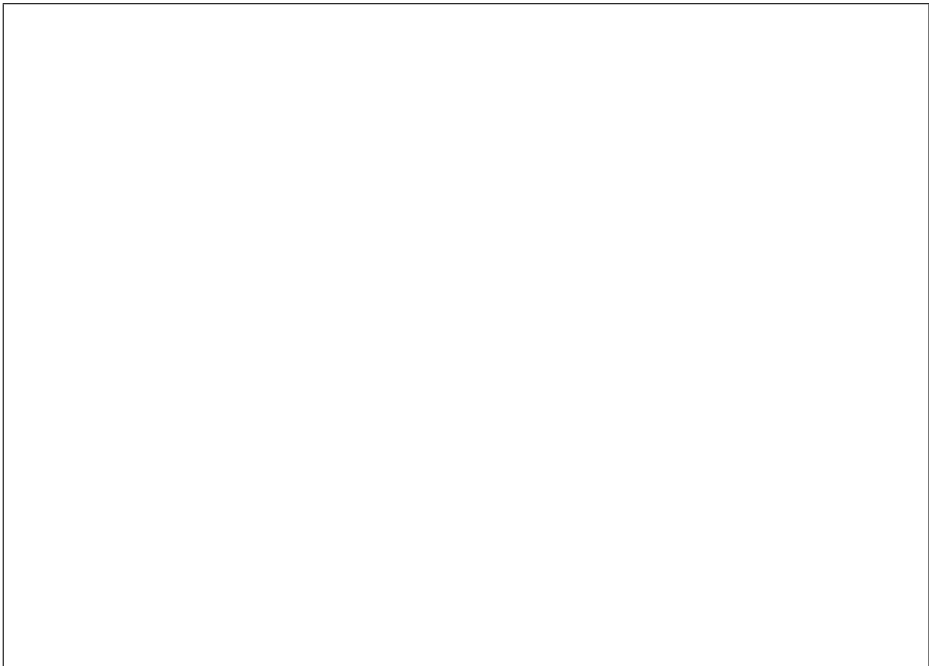
Raises  
 Age  
 tAPI  
 ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man

Raises  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as

ing the prior command.

Returns

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)



**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-

sponse from the agent.

tion  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques  
 or  
 there  
 was  
 a  
 mal-  
 form  
 re-

Raises

Age  
 tAPI  
 ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man

Raises

Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres

ing the prior command.

Returns

A  
dict  
con-  
tain-  
ing  
com  
man  
re-  
spor  
from  
ager  
See  
*get*  
for  
a  
com  
man  
re-  
sult  
sam

ple. The value of key `command_result` is in the form of:



(continues on next page)



(continued from previous page)



**finaliz**

In-  
 struc-  
 the  
 ram  
 to  
 fi-  
 nal-  
 ize  
 en-  
 ter-  
 ing  
 of  
 res-  
 cue  
 mod

**Parame**

nod  
 A  
 Nod  
 ob-  
 ject.

**Raises**

Iron  
 icEx  
 cep-  
 tion  
 if  
 res-  
 cue\_  
 is  
 miss  
 ing,  
 or  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques

or there was a malformed response from the agent.

**Raises**

Age  
 tAPI

ing the prior command.

ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man  
 Raises  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres  
 ex-  
 e-  
 cut-  
 Raises  
 In-  
 stan  
 cue-  
 Fail-  
 ure  
 whe  
 the  
 ager  
 rame  
 is  
 too  
 old  
 to  
 sup-  
 port

the rescue password.

sample.

trans-  
mis-  
sion  
of

#### Returns

A  
dict  
con-  
tain-  
ing  
com  
man  
re-  
spor  
from  
ager  
See  
*get*  
for  
a  
com  
man  
re-  
sult

#### get\_cle

Get  
clea  
step  
from  
ager

#### Parame

- **nod**  
A  
node  
ob-  
ject.
- **por**  
Port  
as-  
so-  
ci-  
ated  
with  
the

sponse from the agent.

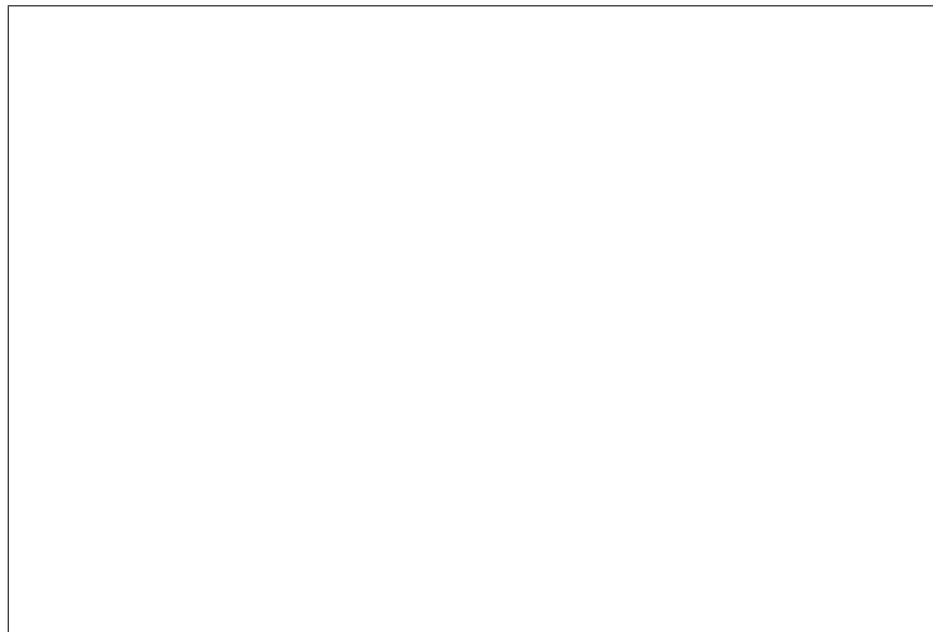
node  
**Raises**  
 Iron  
 icEx  
 cep-  
 tion  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques  
 or  
 there  
 was  
 a  
 mal-  
 form  
 re-

**Raises**  
 Age  
 tAPI  
 ror  
 whe  
 agen  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man

**Raises**  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute

ing the prior command.

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)

--

**get\_com**  
 Get  
 com  
 man  
 sta-  
 tus  
 from  
 ager

**Parame**

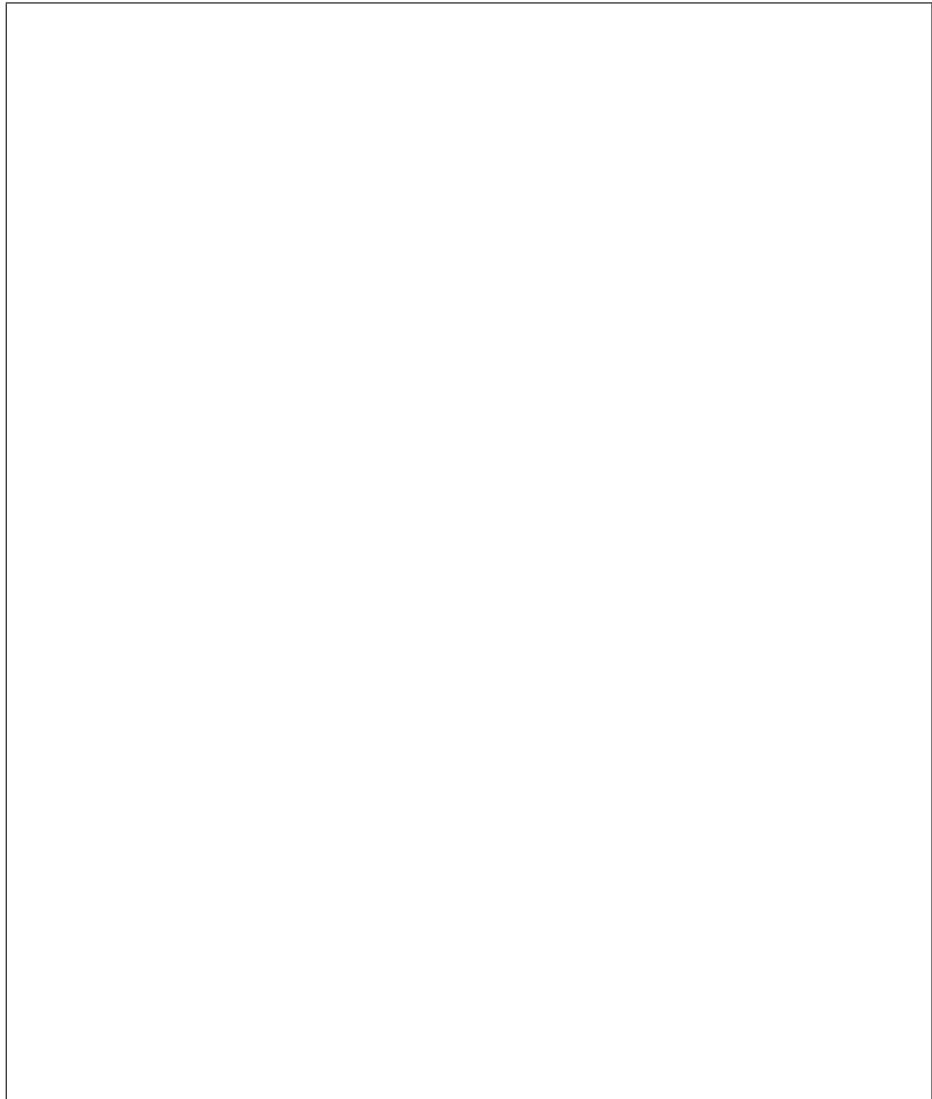
- nod**  
 A  
 Nod  
 ob-  
 ject.
- ret**  
 Whe  
 to  
 retry  
 con-  
 nec-  
 tion  
 prob  
 lems
- exp**  
 If  
 True  
 do  
 not  
 log  
 con-  
 nec-  
 tion  
 prob  
 lems  
 as  
 er-  
 rors.

**Returns**

A

list  
of  
com  
man  
re-  
sults  
each  
re-  
sult  
is  
re-  
lated  
to  
a  
com  
man  
been  
is-  
sued

to agent. A typical result can be:



(continues on next page)

(continued from previous page)

```
→ e.g. 'RUNNING', 'SUCCEEDED', 'FAILED'>
```

→

(continues on next page)



(continued from previous page)

→succeeded, the value **is** command specific,

→e.g.:

(continues on next page)

→\* a dictionary containing keys `clean_result`

(continued from previous page)

```
↪ and clean_step for the command
```

```
↪ clean.execute_clean_step;
```

(continues on next page)

```
↪* a dictionary containing keys deploy_result
```

(continued from previous page)

```
↪ and deploy_step for the command
```

```
↪ deploy.execute_deploy_step;
```

(continues on next page)

(continued from previous page)

```
↪ the command standby.cache_image;
```

```
↪ * None for the command standby.sync.>
```

**get\_dep**  
Get  
de-  
ploy  
step  
from  
agen

**Parame**

- **node**  
A  
node  
ob-  
ject.
- **port**  
Port  
as-  
so-  
ci-  
ated  
with  
the  
node

**Raises**

Iron  
icEx  
cep-  
tion  
when  
failure  
to  
is-  
sue  
the  
re-  
ques  
or  
there  
was  
a  
mal-  
form  
re-

sponse from the agent.

**Raises**

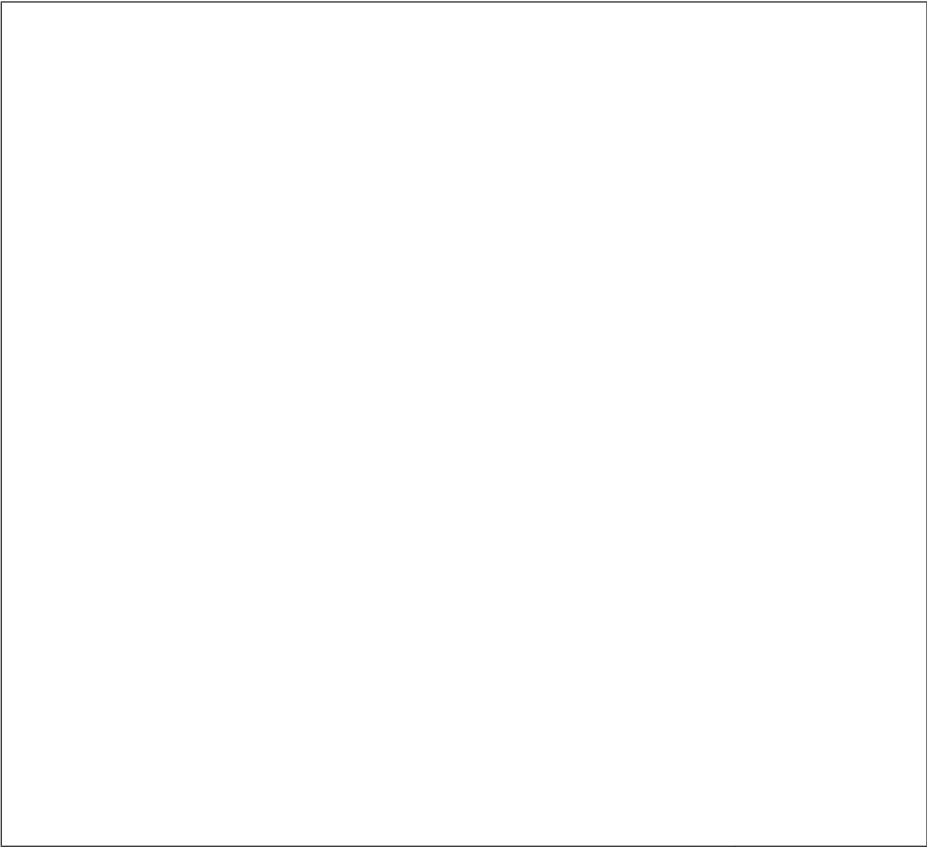
Age  
tAPI  
ror  
when  
agen  
failure  
to  
ex-  
e-  
cute  
spec  
i-

fied  
 com  
 man

Returns

A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man  
 re-  
 spor  
 from  
 agen  
 See  
*get*  
 for  
 a  
 com  
 man  
 re-  
 sult  
 sam

ple. The value of key `command_result` is in the form of:



(continues on next page)

(continued from previous page)

--

**get\_last**  
 Get  
 the  
 last  
 sta-  
 tus  
 for  
 the  
 give  
 com  
 man

**Parame**

- nod**  
 A  
 Nod  
 ob-  
 ject.
- met**  
 Com  
 man  
 nam

**Returns**  
 A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man  
 sta-  
 tus  
 from  
 ager  
 or  
 Non  
 if  
 the  
 com  
 man  
 was  
 not

found.

sponse from the agent.

get\_pai  
 Get  
 de-  
 ploy  
 step  
 from  
 ager  
 Parame  
 nod  
 A  
 node  
 ob-  
 ject.  
 Raises  
 Iron  
 icEx  
 cep-  
 tion  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques  
 or  
 there  
 was  
 a  
 mal-  
 form  
 re-  
 Raises  
 Age  
 tAPI  
 ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man



ing the prior command.

**Raises**

Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres  
 ex-  
 e-  
 cut-

**Returns**

A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man  
 re-  
 spor  
 from  
 ager

**install**

In-  
 stall  
 a  
 boot  
 load  
 on  
 the  
 im-  
 age.

**Parame**

- nod

to, only used for uefi boot mode.

A  
node  
ob-  
ject.

- root**  
The  
UI  
of  
the  
root  
par-  
ti-  
tion.
- target**  
The  
tar-  
get  
de-  
ploy  
men  
boot  
mod
- efi**  
The  
UI  
of  
the  
efi  
sys-  
tem  
par-  
ti-  
tion  
whe  
the  
boot  
load  
will  
be  
in-  
stall
- pre**  
The  
UI

when local booting a partition image on a ppc64\* system.

sponse from the agent.

of  
the  
PRe  
Boo  
par-  
ti-  
tion  
whe  
the  
boot  
load  
will  
be  
in-  
stall  
to

#### **Raises**

Iron  
icEx  
cep-  
tion  
whe  
faile  
to  
is-  
sue  
the  
re-  
ques  
or  
there  
was  
a  
mal-  
form  
re-

#### **Raises**

Age  
tAPI  
ror  
whe  
agen  
faile  
to  
ex-  
e-  
cute  
spec

ing the prior command.

sample.

i-  
 fied  
 com  
 man  
**Raises**  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres  
 ex-  
 e-  
 cut-  
**Returns**  
 A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man  
 re-  
 spon  
 from  
 ager  
 See  
*get*  
 for  
 a  
 com  
 man  
 re-  
 sult  
**power\_c**  
 Soft  
 pow  
 ers

off  
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  
mal-  
form  
re-

sponse from the agent.

#### Raises

Age  
tAPI  
ror  
whe  
ager  
faile  
to  
ex-  
e-  
cute  
spec

ing the prior command.

sample.

i-  
 fied  
 com  
 man  
**Raises**  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres  
 ex-  
 e-  
 cut-  
**Returns**  
 A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man  
 re-  
 spon  
 from  
 ager  
 See  
*get*  
 for  
 a  
 com  
 man  
 re-  
 sult  
**prepare**  
 Call  
 the  
*pre-*

pare  
 meth  
 on  
 the  
 node

Param

- **node**  
 A  
 Node  
 ob-  
 ject.
- **image**  
 A  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 var-  
 i-  
 ous  
 im-  
 age  
 re-  
 latec  
 in-  
 for-  
 ma-  
 tion.
- **wait**  
 True  
 to  
 wait  
 for  
 the  
 com  
 man  
 to  
 fin-  
 ish  
 ex-  
 e-  
 cut-  
 ing,

sponse from the agent.

Fals  
oth-  
er-  
wise  
  
**Raises**  
Iron  
icEx  
cep-  
tion  
whe  
faile  
to  
is-  
sue  
the  
re-  
ques  
or  
there  
was  
a  
mal-  
form  
re-  
  
**Raises**  
Age  
tAPI  
ror  
whe  
ager  
faile  
to  
ex-  
e-  
cute  
spec  
i-  
fied  
com  
man  
  
**Raises**  
Age  
Prog  
whe  
the  
com  
man  
fails  
to



ing the prior command.

sample.

ex-  
e-  
cute  
as  
the  
ager  
is  
pres  
ex-  
e-  
cut-

Returns

A  
dict  
con-  
tain-  
ing  
com  
man  
sta-  
tus  
from  
ager  
See  
*get*  
for  
a  
com  
man  
re-  
sult

reboot

Soft  
re-  
boot  
the  
bare  
meta  
node  
by  
shut  
ting  
dow  
rame  
OS.

Parame

nod  
A

sponse from the agent.

Nod  
 ob-  
 ject.  
 Raises  
 Iron  
 icEx  
 cep-  
 tion  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques  
 or  
 there  
 was  
 a  
 mal-  
 form  
 re-  
 Raises  
 Age  
 tAPI  
 ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man  
 Raises  
 Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-

ing the prior command.

sample.

e-  
cute  
as  
the  
ager  
is  
pres  
ex-  
e-  
cut-

#### Returns

A  
dict  
con-  
tain-  
ing  
com  
man  
re-  
spor  
from  
ager  
See  
[get](#)  
for  
a  
com  
man  
re-  
sult

#### start\_i

Ex-  
pose  
the  
node  
disk  
as  
an  
ISC  
tar-  
get.

#### Parame

- **nod**  
an  
Iron

RAID or filesystem signature.

node

ob-

ject

•

**iqn**

iSCS

tar-

get

IQN

•

**port**

iSCS

por-

tal

port

•

**wip**

True

if

the

ager

shou

wipe

first

the

disk

mag

strin

like

the

par-

ti-

tion

ta-

ble,

**Raises**

Iron

icEx

cep-

tion

whe

faile

to

is-

sue

the

re-

ques

sponse from the agent.

or  
there  
was  
a  
mal-  
form  
re-

Raises

Age  
tAPI  
ror  
whe  
ager  
faile  
to  
ex-  
e-  
cute  
spec  
i-  
fied  
com  
man

Raises

Age  
Prog  
whe  
the  
com  
man  
fails  
to  
ex-  
e-  
cute  
as  
the  
ager  
is  
pres  
ex-  
e-  
cut-

ing the prior command.

Returns

A  
dict  
con-  
tain-

sample.

ing  
 com  
 man  
 re-  
 spor  
 from  
 ager  
 See  
*get*  
 for  
 a  
 com  
 man  
 re-  
 sult

**sync** (*no*  
 Flus  
 file  
 sys-  
 tem  
 buff  
 forc  
 ing  
 char  
 bloc  
 to  
 disk

**Parame**  
**nod**  
 A  
 Nod  
 ob-  
 ject.

**Raises**  
 Iron  
 icEx  
 cep-  
 tion  
 whe  
 faile  
 to  
 is-  
 sue  
 the  
 re-  
 ques  
 or  
 there  
 was

sponse from the agent.

Raises

Age  
 tAPI  
 ror  
 whe  
 ager  
 faile  
 to  
 ex-  
 e-  
 cute  
 spec  
 i-  
 fied  
 com  
 man

Raises

Age  
 Prog  
 whe  
 the  
 com  
 man  
 fails  
 to  
 ex-  
 e-  
 cute  
 as  
 the  
 ager  
 is  
 pres  
 ex-  
 e-  
 cut-

ing the prior command.

Returns

A  
 dict  
 con-  
 tain-  
 ing  
 com  
 man

sample.

re-  
 spon  
 from  
 ager  
 See  
*get*  
 for  
 a  
 com  
 man  
 re-  
 sult

ironic.  
 Ex-  
 tract  
 an  
 er-  
 ror  
 strin  
 from  
 the  
 com  
 man  
 re-  
 sult.

Paramet

com  
 Com  
 man  
 in-  
 for-  
 ma-  
 tion  
 from  
 the  
 ager

Returns

Er-  
 ror  
 strin



## ironic.drivers.modules.agent\_power module

The  
ager  
pow  
in-  
ter-  
face

**class** `ironiC.drivers.modules.agent_power`  
Base  
*ironiC*  
*driv*  
*bas*  
*Power*

Power  
in-  
ter-  
face  
us-  
ing  
the  
run-  
ning  
ager  
for  
pow  
ac-  
tions

**get\_pow**  
Re-  
turn  
the  
pow  
state  
of  
the  
task  
node

Es-  
sen-  
tially  
the  
only  
know  
state  
is  
POV

precisely None).

ON,  
 ev-  
 ery-  
 thing  
 else  
 is  
 an  
 er-  
 ror  
 (or  
 more

Paramete

tas  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

Returns

A  
 pow  
 state  
 One  
 of  
*irc*  
*com*  
*sta*

get\_pro

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-  
 tains  
 RE-  
 BOO

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**

A  
 list  
 with  
 the  
 sup-  
 port

pow  
state  
de-  
fine  
in  
*irc*  
*com*  
*sta*

**reboot**

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  
on.
- **tim**  
time  
out  
(in  
sec-

indicates to use default timeout.

onds  
pos-  
i-  
tive  
in-  
te-  
ger  
(>  
0)  
for  
any  
pow  
state  
Non

**set\_pow**  
Set  
the  
pow  
state  
of  
the  
task  
node

**Parame**

- **tas**  
A  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **pow**  
Pow  
state  
from  
*irc*  
*com*  
*sta*

mous.

indicates to use default timeout.

Only  
 RE-  
 BOC  
 and  
 SOF  
 are  
 sup-  
 port  
 and  
 are  
 syn-  
 ony-  
  
 •  
 tim  
 time  
 out  
 (in  
 sec-  
 onds  
 pos-  
 i-  
 tive  
 in-  
 te-  
 ger  
 (>  
 0)  
 for  
 any  
 pow  
 state  
 Non  
  
 Raises  
 Pow  
 er-  
 State  
 Fail-  
 ure  
 on  
 non-  
 supp  
 pow  
 state  
  
 support  
 Che  
 if  
 pow  
 sync

is  
 sup-  
 port  
 for  
 the  
 give  
 node

Not  
 sup-  
 port  
 for  
 the  
 agen  
 pow  
 sinc  
 it  
 is  
 not  
 pos-  
 si-  
 ble  
 to  
 pow  
 on/o  
 node

Parame

tas  
 A  
 Task  
 agen  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on  
 with  
 a  
 shar  
 lock

Returns

bool  
 whe  
 pow  
 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  
 act  
 on.

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 mal-  
 form  
 pa-  
 ram-  
 e-  
 ter(s)



## ironic.drivers.modules.boot\_mode\_utils module

ironic.  
 Con-  
 fig-  
 ures  
 se-  
 cure  
 boot  
 if  
 it  
 has  
 been  
 re-  
 ques-  
 for  
 the  
 node

ironic.  
 De-  
 con-  
 fig-  
 ures  
 se-  
 cure  
 boot  
 if  
 it  
 has  
 been  
 re-  
 ques-  
 for  
 the  
 node

ironic.  
 Re-  
 turn  
 the  
 boot  
 mod

**Paramet**  
**nod**  
 an  
 iron.  
 node  
 ob-  
 ject.

**Returns**

to node properties/capabilities

bios  
 or  
 uefi  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 node  
 boot  
 mod  
 dis-  
 agre  
 with  
 the  
 boot  
 mod  
 set  
  
 ironic.  
 Re-  
 turn  
 the  
 boot  
 mod  
 that  
 wou  
 be  
 used  
 for  
 de-  
 ploy  
  
 This  
 meth  
 re-  
 turn  
 boot  
 mod  
 to  
 be  
 used  
 for  
 de-  
 ploy  
 It

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).

to node properties/capabilities

re-  
 turn  
 uefi  
 if  
 se-  
 cure  
 is  
  
**Parameters**  
 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-  
 agre  
 with  
 the  
 boot  
 mod  
 set  
  
 ironic.  
 Re-  
 turn

True  
 if  
 se-  
 cure  
 is  
 re-  
 ques  
 for  
 de-  
 ploy  
 This  
 meth  
 chec  
 node  
 prop  
 erty  
 for  
 se-  
 cure  
 and  
 re-  
 turn  
 True  
 if  
 it  
 is  
 re-  
 ques  
 Paramet  
 nod  
 a  
 sin-  
 gle  
 Nod  
 Raises  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 strin

tionary or is malformed.

is  
not  
a  
dic-

Returns

True  
if  
se-  
cure  
is  
re-  
ques

ironic.

Re-  
turn  
True  
if  
trust  
is  
re-  
ques  
for  
de-  
ploy

This  
meth  
check  
in-  
stan  
prop  
erty  
for  
trust  
and  
re-  
turn  
True  
if  
it  
is  
re-  
ques

Paramet

nod  
a  
sin-  
gle  
Nod

tionary or is malformed.

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
ca-  
pa-  
bil-  
i-  
ties  
strin  
is  
not  
a  
dic-

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-  
rentl  
set  
boot  
mod

configuration:

set and apply the logic that follows

off  
the  
bare  
meta-  
ma-  
chin  
Also  
read  
node  
boot  
mod

- If  
BM  
drive  
does  
not  
im-  
ple-  
men  
get-  
ting  
boot  
mod  
as-  
sum  
BM  
boot  
mod  
is  
not

- If  
Iron  
node  
boot  
mod  
is  
not  
set  
and  
BM  
node  
boot  
mod  
is  
not  
set

IroniC boot mode to *[deploy]/default\_boot\_mode*

node boot mode on the IroniC node

boot mode to BM boot mode

-

set

- If IroniC node boot mode is not set and BM node boot mode is set -

set BM

- If IroniC node boot mode is set and BM node boot mode is not set -

set IroniC

- If both IroniC



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**

and  
BM  
node  
boot  
mod  
are  
set  
but  
they  
dif-  
fer  
-

try  
to  
set

In  
the  
end,  
the  
new  
boot  
mod  
may  
be  
set  
in  
drive

**Parameter**  
**task**  
a  
task  
ob-  
ject

Iron  
con-  
sole  
util-  
i-  
ties.

ironic.  
Re-  
turn

a  
 free  
 TCP  
 port  
 on  
 cur-  
 rent  
 host  
  
 Find  
 and  
 re-  
 turn  
 a  
 free  
 TCP  
 port  
 in  
 the  
 rang  
 of  
 COM  
  
 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

old process cannot be stopped.

con-  
sole  
**Raises**  
Con  
sole.  
ror  
if  
the  
di-  
rec-  
tory  
for  
the  
PID  
file  
can-  
not  
be  
cre-  
ated  
or  
an

**Raises**  
Con  
sole  
Sub-  
pro-  
cess  
Fail  
whe  
in-  
vok-  
ing  
the  
sub-  
pro-  
cess  
faile

ironic.

Ope  
the  
se-  
rial  
con-  
sole  
for  
a

sole to the node

node

**Parameters**

- nodeId**  
 the  
 uuid  
 of  
 the  
 node
- port**  
 the  
 ter-  
 mi-  
 nal  
 port  
 for  
 the  
 node
- console**  
 the  
 shel  
 com  
 man  
 that  
 will  
 be  
 ex-  
 e-  
 cute  
 by  
 so-  
 cat  
 to  
 es-  
 tab-  
 lish  
 con-

**Raises**

- ConnectionError*  
 if  
 the  
 di-

rec-  
 tory  
 for  
 the  
 PID  
 file  
 or  
 the  
 PID  
 file  
 can-  
 not  
 be  
 cre-  
 ated

- **Con**  
 whe  
 in-  
 vok-  
 ing  
 the  
 sub-  
 pro-  
 cess  
 faile

ironic.  
 Clos  
 the  
 se-  
 rial  
 con-  
 sole  
 for  
 a  
 node

**Paramet**  
**nod**  
 the  
 UUI  
 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

**Parameter**  
**node**  
 the  
 UUI  
 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.



tion to IPA.

Add  
 re-  
 quir  
 con-  
 fig  
 pa-  
 ram-  
 e-  
 ters  
 to  
 node  
 drive  
 Add  
 the  
 re-  
 quir  
 conf  
 op-  
 tion  
 to  
 node  
 drive  
 It  
 is  
 Re-  
 quir  
 to  
 pass  
 the  
 in-  
 for-  
 ma-

Paramet  
 tas  
 a  
 Task  
 ager  
 in-  
 stan

ironic.  
 Buil  
 the  
 op-  
 tion  
 to  
 be  
 pass  
 to  
 the

ager  
ramo

## Paramet

**node**  
an  
iron  
node  
ob-  
ject

## Returns

a dictionary containing the parameters to be passed to agent framework

ironic.

Build  
in-  
stan-  
ces  
nec-  
es-  
sary  
for  
de-  
ploy-  
ing  
to  
a  
node

## Paramet

**task**  
a  
Task  
ager  
ob-  
ject

con-  
 tain-  
 ing  
 the  
 node

Returns

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 prop  
 er-  
 ties  
 to  
 be  
 up-  
 date  
 in  
 in-  
 stan

Raises

ex-  
 cep-  
 tion.  
 if  
 im-  
 age\_  
 is  
 not  
 Glar  
 href  
 and  
 is  
 not  
 HTT  
 URI

ironic.

Fetc  
 the  
 in-  
 stan  
 im-  
 age  
 from  
 Glar

disk.

This  
 meth  
 pulls  
 the  
 disk  
 im-  
 age  
 and  
 writ  
 then  
 to  
 the  
 ap-  
 pro-  
 pri-  
 ate  
 plac  
 on  
 lo-  
 cal

Paramet

- **ctx**  
con-  
text
- **nod**  
an  
iron  
node  
ob-  
ject
- **for**  
whe  
con-  
vert  
im-  
age  
to  
raw  
for-  
mat

Returns

a  
 tu-  
 ple

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  
 er-  
 ror  
 mes

missing parameters.

sage  
 to  
 pre-  
 fix  
 be-  
 fore  
 prin  
 ing  
 the  
 in-  
 for-  
 ma-  
 tion  
 abou  
 •  
 par  
 Add  
 this  
 pre-  
 fix  
 to  
 each  
 pa-  
 ram-  
 e-  
 ter  
 for  
 er-  
 ror  
 mes  
 sage  
 Raises  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 one  
 or  
 mor  
 pa-  
 ram-  
 e-  
 ters  
 are  
 emp

provided dictionary.

in  
the  
  
ironic.

Eval  
u-  
ate  
in-  
ter-  
face  
to  
de-  
ter-  
mine  
if  
ca-  
pa-  
bil-  
ity  
is  
pres

Paramet

- **int**  
The  
in-  
ter-  
face  
ob-  
ject  
to  
check
- **cap**  
The  
valu  
re-  
re-  
sent  
ing  
the  
ca-  
pa-  
bil-

present.

Returns

True if capability found other-wise False

ironic.

Compute checksum by giving image path and algorithm

ironic.

Delete instance image file and symbolic link refer to



it.  
 ironic.  
 Dele  
 in-  
 stan  
 im-  
 age  
 file.

**Paramet**  
**nod**  
 the  
 uuid  
 of  
 the  
 iron  
 node

ironic.  
 Whe  
 con-  
 vert  
 im-  
 age  
 to  
 raw  
 for-  
 mat  
 for  
 spec  
 i-  
 fied  
 node

**Paramet**  
**nod**  
 iron  
 node  
 ob-  
 ject

**Returns**  
 Boo  
 whe  
 the  
 di-  
 rect  
 de-  
 ploy  
 in-  
 ter-  
 face  
 shou

con-  
 vert  
 im-  
 age  
 to  
 raw.  
 ironic.

Che  
 for  
 avai  
 able  
 disk  
 spac  
 and  
 fetcl  
 im-  
 ages  
 us-  
 ing  
 Im-  
 age-  
 Cacl

Paramet

- **ctx**
  
 con-  
 text
- **cac**
  
 Im-  
 age-  
 Cacl  
 in-  
 stan  
 to  
 use  
 for  
 fetcl  
 ing
- **ima**
  
 list  
 of  
 tu-  
 ples  
 (im-

age  
 href  
 des-  
 ti-  
 na-  
 tion  
 path

- **for**  
 bool  
 valu  
 whe  
 to  
 con-  
 vert  
 the  
 im-  
 age  
 to  
 raw  
 for-  
 mat

**Raises**  
 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

in progress.

in-  
voked

**Parameters**  
**node**  
 an  
 iron-  
 node  
 ob-  
 ject.

**Returns**  
 state  
 if  
 clear  
 ing  
 op-  
 er-  
 a-  
 tion  
 in  
 prog-  
 or  
 state  
 if  
 de-  
 ploy  
 op-  
 er-  
 a-  
 tion

ironic.  
 Gets  
 the  
 boot  
 op-  
 tion.

**Parameters**  
**node**  
 A  
 sin-  
 gle  
 Node

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-

or is malformed.

Valu
 if
 the
 ca-
 pa-
 bil-
 i-
 ties
 strin
 is
 not
 a
 dict

Returns

A
 strin
 rep-
 re-
 sent
 ing
 the
 boot
 op-
 tion
 type
 De-
 fault
 to
 net-
 boot

ironic.
 Gets
 the
 de-
 fault
 boot
 op-
 tion.

ironic.
 Re-
 turn
 the
 disk
 la-
 bel
 re-
 ques
 for
 de-

tionary or is malformed.

ploy  
 if  
 any.  
**Parameters**  
**node**  
 a  
 sin-  
 gle  
 Node  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 strin  
 is  
 not  
 a  
 dic-  
**Returns**  
 the  
 disk  
 la-  
 bel  
 or  
 Non  
 if  
 no  
 disk  
 la-  
 bel  
 was  
 spec  
 i-  
 fied.  
 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  
its  
in-  
stan-  
prop  
erty.

**Paramet**  
**nod**  
a  
sin-  
gle  
Nod

**Returns**  
A

same exception if kernel/ramdisk is missing in instance\_info for non-glance images.

dict  
with  
re-  
quir  
im-  
age  
prop  
er-  
ties  
re-  
triev  
from  
node  
in-  
stan

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
im-  
age\_  
is  
miss  
ing  
in  
node  
in-  
stan  
Also  
raise

ironic.  
Re-  
turn  
the  
iPXE  
boot  
file  
nam  
re-  
ques  
for  
de-  
ploy  
This



boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

[pxe]uefi\_ipxe\_bootfile\_name settings.

meth  
re-  
turn  
iPXE  
boot  
file  
nam  
to  
be  
used  
for  
de-  
ploy  
Ar-  
chi-  
tec-  
ture  
spe-  
cific

If  
no  
valid  
valu  
is  
foun  
the  
de-  
fault  
re-  
verts  
to  
the  
get  
meth  
and  
thus  
the  
[pxe]  
and

#### Parameters

nod  
A  
sin-  
gle  
Node

#### Returns

The  
iPXE

```

boot
file
nam

ironic.
Re-
turn
the
iPXE
con-
fig
tem-
plate
file
nam
re-
ques
of
de-
ploy

This
meth
re-
turn
the
iPXE
con-
fig-
u-
ra-
tion
tem-
plate
file.

Parameters
nodes
A
sin-
gle
Node

Returns
The
iPXE
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  
Ar-  
chi-  
tec-  
ture  
spe-

boot file is searched first. BIOS/UEFI boot file is used if no valid architecture specific file found.

Paramet

nod  
A  
sin-  
gle  
Nod

Returns

The  
PXE  
boot  
file  
nam

ironic.

Re-  
turn  
the  
PXE  
con-  
fig  
tem-  
plate  
file  
nam  
re-  
ques  
for  
de-  
ploy

This  
meth  
re-  
turn  
PXE  
con-  
fig  
tem-  
plate  
file  
to  
be  
used  
for  
de-  
ploy  
First  
spe-  
cific

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.

Paramet
 nod
 A
 sin-
 gle
 Nod

Returns
 The
 PXE
 con-
 fig
 tem-
 plate
 file
 nam

ironic.
 Iden
 tify
 a
 boot
 vol-
 ume
 from
 any
 con-
 fig-
 ured
 vol-
 ume

Returns
 Non
 or
 the
 vol-
 ume
 tar-
 get
 rep-
 re-
 sent
 ing
 the
 vol-
 ume

ironic.
 Get

a  
 root  
 de-  
 vice  
 re-  
 ques  
 for  
 de-  
 ploy  
 men  
 or  
 Non

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 in-  
 valid  
 hints

Returns

Pars  
 root  
 de-  
 vice  
 hints  
 or  
 Non  
 if  
 no  
 hints  
 were  
 pro-  
 vide

ironic.  
 Re-  
 turn  
 the  
 MA  
 ad-  
 dres  
 of  
 a  
 port  
 whic  
 has

a  
 VIF  
 port  
 id.

Paramet

tas  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 port  
 to  
 act  
 on.

Returns

MA  
 ad-  
 dres  
 of  
 the  
 port  
 con-  
 nect  
 to  
 de-  
 ploy  
 men  
 net-  
 worl  
 Non  
 if  
 it  
 can-  
 not

find any port with vif id.

ironic.  
 Re-  
 turn  
 true  
 if  
 boot  
 ing  
 from  
 an  
 iscsi

vol-  
 ume  
  
 ironic.  
 De-  
 ter-  
 mine  
 if  
 soft-  
 ware  
 raid  
 is  
 in  
 use  
 for  
 the  
 de-  
 ploy  
 men

**Paramet**  
**nod**  
 A  
 sin-  
 gle  
 Nod

**Returns**  
 A  
 bool  
 valu  
 of  
 True  
 whe  
 soft-  
 ware  
 raid  
 is  
 in  
 use,  
 oth-  
 er-  
 wise  
 Fals

ironic.  
 Gets  
 the  
 in-  
 stan  
 spe-  
 cific  
 Nod



quired information for this driver to deploy images to the node.

de-  
 ploy  
 men  
 info  
 This  
 meth  
 val-  
 i-  
 date  
 whe  
 the  
 in-  
 stan  
 prop  
 erty  
 of  
 the  
 sup-  
 plied  
 node  
 con-  
 tains  
 the  
 re-

#### Parameters

**node**  
 a  
 sin-  
 gle  
 Node

#### Returns

A  
 dict  
 with  
 the  
 in-  
 stan  
 val-  
 ues.

#### Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if

ing.

any  
of  
the  
re-  
quir  
pa-  
ram-  
e-  
ters  
are  
miss

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
any  
of  
the  
pa-  
ram-  
e-  
ters  
have  
in-  
valid  
valu

ironic.  
Set  
node  
drive  
for  
boot  
from  
vol-  
ume  
pa-  
ram-  
e-  
ters.

Paramet

tas  
a  
Task  
ager

ob-  
 ject  
 con-  
 tain-  
 ing  
 the  
 node

Raises

Stor-  
 ageE-  
 rror  
 whe-  
 a  
 node  
 has  
 an  
 iSCSI  
 or  
 Fi-  
 breC-  
 nel  
 boot  
 vol-  
 ume  
 de-  
 fined  
 but

is not capable to support it.

ironic.

Pre-  
 pare  
 the  
 node  
 to  
 boot  
 into  
 ager  
 for  
 in-  
 band  
 clear  
 ing.

This  
 meth-  
 does  
 the  
 fol-  
 low-

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.

Paramet

- **task\_agent\_obj\_con-  
tain-  
ing  
the  
node**

a Task agent ob-  
ject con-  
tain-  
ing  
the  
node
- **manage\_boot**

If this is set to True this meth-  
calls the pre-  
pare meth-  
of boot in-  
ter-

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.

#### Returns

state  
to  
sig-  
nify  
an  
asyn  
chro  
pre-  
pare

#### Raises

Net-  
worl  
Er-  
ror,  
Nod  
Clea  
ing-  
Fail-  
ure  
if  
the  
pre-  
vi-  
ous  
clea  
ing  
port  
can-  
not

be removed or if new cleaning ports cannot be created.

#### Raises

In-  
vali  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
clea  
ing  
net-  
worl  
UUI  
con-

invalid value.

fig  
 op-  
 tion  
 has  
 an  
  
 ironic.  
  
 ironic.  
  
 ironic.

Sets  
 ap-  
 pro-  
 pri-  
 ate  
 re-  
 boot  
 flags  
 in  
 drive  
 base  
 on  
 op-  
 er-  
 a-  
 tion

Paramet

- **node**  
 an  
 iron  
 node  
 ob-  
 ject.
- **reboot**  
 Boo  
 valu  
 to  
 set  
 for  
 node  
 drive

or deployment operation in progress. If it is None, corresponding reboot flag is not set in nodes driver\_internal\_info.

ment operation in progress. If it is None, corresponding skip step flag is not set in nodes driver\_internal\_info.

flag  
clear  
ing\_  
or  
de-  
ploy  
men  
base  
on  
clear  
ing

- **ski**  
Boo  
valu  
to  
set  
for  
node  
drive  
flag  
skip  
or  
skip  
base  
on  
clear  
ing  
or  
de-  
ploy

- **pol**  
Boo  
valu  
to  
set  
for  
node  
drive  
flag  
de-  
ploy  
men  
or  
clear

the corresponding polling flag is not set in the nodes driver\_internal\_info.

to DEPLOYFAIL and updates last\_error with the given error message. It also powers off the baremetal node.

ing\_  
If  
it  
is  
Non

ironic.

Sets  
the  
de-  
ploy  
sta-  
tus  
as  
faile  
with  
rel-  
e-  
vant  
mes  
sage

This  
meth  
sets  
the  
de-  
ploy  
men  
as  
fail  
with  
the  
give  
mes  
sage  
It  
sets  
node  
pro-  
vi-  
sion

## Paramet

- tas  
a



Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

- **msg**  
the  
mes-  
sage  
to  
set  
in  
logs  
and  
last\_  
of  
the  
node

- **col**  
Boo  
in-  
di-  
cat-  
ing  
whe  
to  
at-  
temp  
to  
col-  
lect  
logs  
from  
IPA-  
base  
rame  
De-

faults to True. Actual log collection is also affected by CONF.agent.deploy\_logs\_collect config option.

ironic.

Switch  
a  
pxe  
con-  
fig  
from  
de-  
ploy  
men  
mod  
to  
ser-  
vice  
mod

Parameters

- **path**  
path  
to  
the  
pxe  
con-  
fig  
file  
in  
tftp-  
boot
- **root**  
root  
uuid  
in  
case  
of  
par-  
ti-  
tion  
im-  
age  
or  
disk

age.

in  
 case  
 of  
 who  
 disk  
 im

- - boo**  
 if  
 boot  
 mod  
 is  
 uefi  
 or  
 bios

- - is\_**  
 if  
 the  
 im-  
 age  
 is  
 a  
 who  
 disk  
 im-  
 age  
 or  
 not.

- - tru**  
 if  
 boot  
 with  
 trust  
 or  
 not.  
 The  
 us-  
 age  
 of  
 is\_w  
 and  
 trust  
 are  
 mu-  
 tu-  
 ally  
 ex-

clusive. You can have one or neither, but not both.

- **iscsi**  
if  
boot  
is  
from  
an  
iSCSI  
vol-  
ume  
or  
not.
- **ramdisk**  
if  
the  
boot  
is  
to  
be  
to  
a  
ramdisk  
con-  
fig-  
u-  
ra-  
tion.
- **ipxe**  
A  
de-  
fault  
Fals  
bool  
valu  
to  
tell  
the  
meth  
if  
the  
calle  
is  
us-  
ing  
iPXE

ironic.

Tear  
dow  
the  
en-  
vi-  
ron-  
men-  
setu  
for  
in-  
band  
clean  
ing.

This  
meth  
does  
the  
fol-  
low-  
ing:  
1.  
Pow  
ers  
off  
the  
bare  
meta  
node  
(un-  
less  
the  
node  
is

fast tracked or there was a cleaning failure). 2. If manage\_boot parameter is set to true, it also calls the clean\_up\_ramdisk method of boot interface to clean up the environment that was set for booting agent ramdisk. 3. Deletes the cleaning ports which were setup as part of cleaning.

## Parameter

- **task**  
a  
Task  
ager  
ob-  
ject  
con-  
tain-

boot the agent ramdisk. If False, it skips this step.

ing  
 the  
 node

- man**  
 If  
 this  
 is  
 set  
 to  
 True  
 this  
 meth  
 calls  
 the  
 clea  
 meth  
 of  
 boot  
 in-  
 ter-  
 face  
 to

**Raises**  
 Net-  
 worl  
 Er-  
 ror,  
 Nod  
 Clea  
 ing-  
 Fail-  
 ure  
 if  
 the  
 clea  
 ing  
 port  
 can-  
 not  
 be  
 re-  
 mov

ironic.  
 Clea  
 up  
 stor-  
 age  
 con-

is done to ensure a clean state for the next boot of the machine.

fig-  
u-  
ra-  
tion.  
  
Re-  
mov  
en-  
tries  
from  
drive  
for  
stor-  
age  
and  
dele  
the  
vol-  
ume  
tar-  
gets  
from  
the  
data  
This

ironic.

Trie  
to  
set  
the  
boot  
de-  
vice  
on  
the  
node  
  
This  
meth  
tries  
to  
set  
the  
boot  
de-  
vice  
on  
the  
node

uefi boot mode, setting of boot device may differ between different machines. IPMI does not work for setting boot devices in uefi mode for certain machines. This method ignores the expected IPMI failure for uefi boot mode and just logs a message. In error cases, it is expected the operator has to manually set the node to boot from the correct device.

Parameters

- **task**  
 a TaskManager object containing the node
- **device**  
 the boot device
- **persistent**  
 Whether to set the boot device persistently

Raises  
 Any exception



ing ipmi is expected to fail).

from  
set\_  
ex-  
cept  
IP-  
MI-  
Fail-  
ure  
(set-  
ting  
of  
boot  
de-  
vice  
us-  
  
ironic.  
Val-  
i-  
date  
that  
spec  
i-  
fied  
sup-  
port  
ca-  
pa-  
bil-  
i-  
ties  
have  
valid  
valu  
  
This  
meth  
chec  
if  
the  
any  
of  
the  
sup-  
port  
ca-  
pa-  
bil-  
ity  
is  
pres

bilities. 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.

value.

in  
 Nod  
 ca-  
 pa-

**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

ironic.

Val-  
 i-  
 date  
 the  
 im-  
 age.  
 For  
 Glan

deployment info contain the properties passed. If its not a Glance image, it checks that deployment info contains needed properties.

#### Parameters

- **ctx**  
se-  
cu-  
rity  
con-  
text
- **dep**  
the  
de-  
ploy  
to  
be  
val-  
i-  
date
- **pro**  
the  
list  
of  
im-  
age  
meta  
prop  
to

tion 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.

ties.

be  
 val-  
 i-  
 date  
 Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if:  
 \*  
 con-  
 nec-  
 tion  
 to  
 glan  
 faile  
 \*  
 au-  
 tho-  
 riza-  
 Raises  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 im-  
 age  
 does  
 con-  
 tain  
 the  
 men  
 tion  
 prop  
 er-

## ironic.drivers.modules.fake module

stance, the `MultipleVendorInterface` class demonstrates how to load more than one interface and wrap them in some logic to route incoming `vendor_passthru` requests appropriately. This can be useful eg. when mixing functionality between a power interface and a deploy interface, when both rely on separate `vendor_passthru` methods.

Fake  
drive  
in-  
ter-  
face  
used  
in  
test-  
ing.  
This  
is  
also  
an  
ex-  
am-  
ple  
of  
some  
kind  
of  
thing  
which  
can  
be  
done  
with  
drive  
For  
in-

```
class i
Base
    irc
    dri
    bas
    BIC

Fake
im-
ple-
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  
 BIO  
 set-  
 ting  
 on  
 the

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver\_info properties. After the BIOS configuration is done, cache\_bios\_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

Paramete

- 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-  
port  
BIO  
con-  
fig-  
u-  
ra-

tion.

**Raises**

In-  
valie  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
val-  
i-

da-  
tion  
of  
set-  
ting  
fails

**Raises**

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
re-  
quir  
pa-  
ram-  
e-  
ters  
are  
miss  
ing.

**Returns**

state  
if  
BIO  
con-  
fig-  
u-  
ra-  
tion  
is  
in  
prog  
asyn  
chro  
or  
Non  
if  
it  
is  
com

plete.

**cache\_k**

Stor  
or  
up-



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.

**Parameters**  
 task : Task object  
 a Task object representing the task.

**Raises**  
 UnsupportedDevice : If the device is not supported.  
 DriveException : If there is an error with the drive.

ties from bare metal.

ter the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings

table with default bios settings.

**Parameters**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 node  
 drive  
 does  
 sup-  
 port  
 BIO  
 re-  
 set.

**Returns**  
 state  
 if  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if  
 it  
 is  
 com

plete.

**get\_proc**  
 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-  
 stan  
 prop  
 er-  
 ties  
 of

the required information for this interface to function.

long-running checks.

the  
task  
node  
con-  
tains

This  
meth  
is  
of-  
ten  
ex-  
e-  
cute  
syn-  
chro  
in  
API  
re-  
ques  
so  
it  
shou  
not  
con-  
duct

#### Parame

**tas**  
A  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

#### Raises

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)

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Doc*

Ex-  
am-  
ple  
im-  
ple-  
men  
ta-  
tion  
of  
a  
sim-  
ple  
boot  
in-  
ter-  
face

**capabil**

**clean\_u**  
Clea

up  
 the  
 boot  
 of  
 in-  
 stan

This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that  
 was  
 setu  
 for  
 boot  
 ing  
 the  
 in-  
 stan

**Parame**  
**tas**  
 A  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**clean\_u**  
 Clea  
 up  
 the  
 boot  
 of  
 iron  
 rame  
 This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-

cue ramdisk.

men  
that  
was  
setu  
for  
boot  
ing  
the  
de-  
ploy  
or  
res-

Paramete

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-



tion from the nodes database.

prepare the boot of in-stand This meth pre-prepare the boot of the in-stand af-ter read ing rel-e-vant in-for-ma

**Parameters**  
**task**  
 A task from Taskager

**Returns**  
 None

**prepare**  
 Prepare the boot of Ironrame This meth pre-prepare the

vant information from the nodes database.

might want to boot the ramdisk in different ways by passing parameters to them. For example,

boot  
 of  
 the  
 de-  
 ploy  
 or  
 res-  
 cue  
 ram  
 af-  
 ter  
 read  
 ing  
 rel-  
 e-  
  
 Parame  
  
 •  
**tas**  
 A  
 task  
 from  
 Task  
 ager  
  
 •  
**ram**  
 The  
 op-  
 tions  
 to  
 be  
 pass  
 to  
 the  
 iron  
 ram  
 Dif-  
 fer-  
 ent  
 im-  
 ple-  
 men  
 ta-  
 tions  
  
 Whe  
 Age  
 ram

etc.

ent implementations of boot interface will have different ways of passing parameters to the ramdisk.

is  
boot  
to  
de-  
ploy  
a  
node  
it  
take  
the  
pa-  
ram-  
e-  
ters  
ipa-  
api-  
url,

Othe  
im-  
ple-  
men-  
ta-  
tions  
can  
mak  
use  
of  
ram  
to  
pass  
such  
in-  
for-  
ma-  
tion.  
Dif-  
fer-

## Returns

Non

## validation

Val-  
i-  
date  
the  
drive  
spec  
Nod  
de-

the required information for this interface to function.

long-running checks.

**Parameters**  
**taskManager**  
 A TaskManager object

in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**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

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Con*  
 Ex-  
 am-  
 ple

im-  
ple-  
men-  
ta-  
tion  
of  
a  
sim-  
ple  
con-  
sole  
in-  
ter-  
face

get\_con

Get  
con-  
nec-  
tion  
in-  
for-  
ma-  
tion  
about  
the  
con-  
sole

This  
meth  
shou  
re-  
turn  
the  
nec-  
es-  
sary  
in-  
for-  
ma-  
tion  
for  
the  
clien  
to  
ac-  
cess  
the

console.

Parame

tas

A Task ager in- stan con- tain- ing the node to act on.

**Returns**

the con- sole con- nec- tion in- for- ma- tion.

**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

**start\_c**  
 Start  
 a  
 re-  
 mote  
 con-  
 sole  
 for  
 the  
 task  
 node  
  
 This  
 meth  
 shou  
 not  
 raise  
 an  
 ex-  
 cep-  
 tion  
 if  
 con-  
 sole  
 al-  
 read  
 start

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**stop\_cc**  
 Stop  
 the  
 re-  
 mote  
 con-  
 sole  
 ses-  
 sion



for  
 the  
 task  
 node

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**validat**

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  
 task  
 node  
 con-

the required information for this interface to function.

long-running checks.

ram-  
 e-  
 ter(s)

**Raises**

Miss-  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu-  
 on  
 miss-  
 ing  
 pa-  
 ram-  
 e-  
 ter(s)

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Dep*

Clas  
 for  
 a  
 fake  
 de-  
 ploy  
 men  
 drive

Ex-  
 am-  
 ple  
 im-  
 ple-  
 men  
 ta-  
 tion  
 of  
 a  
 de-  
 ploy  
 in-  
 ter-  
 face  
 that  
 uses

rate power interface.

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

**clean\_u**

Clea  
up  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
for  
the  
task  
node

If  
prep  
ra-  
tion  
of  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
ahea  
of  
time  
is  
pos-  
si-  
ble,

If  
im-  
ple-  
men  
this  
meth  
mus  
be  
iden

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.

po-  
tent.  
It  
may  
be  
called  
mul-  
ti-  
ple  
time  
for

This  
meth  
is  
called  
be-  
fore  
*tear*

#### Parame

**tas**  
A  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

#### deploy

Per-  
form  
a  
de-  
ploy  
men  
to  
the  
task  
node  
  
Per-  
form  
the

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

nec-  
 es-  
 sary  
 worl  
 to  
 de-  
 ploy  
 an  
 im-  
 age  
 onto  
 the  
 spec  
 i-  
 fied  
 node  
 This

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 sta-  
 tus  
 of  
 the  
 de-  
 ploy  
 One  
 of  
 iron

**get\_properties**  
 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  
node

If  
prep  
ra-  
tion  
of  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
ahead  
of  
time

this method should be implemented by the driver.

the same node on the same conductor.

is  
pos-  
si-  
ble,

If  
im-  
ple-  
men  
this  
meth  
mus  
be  
iden  
po-  
tent.  
It  
may  
be  
calle  
mul-  
ti-  
ple  
time  
for

This  
meth  
is  
calle  
be-  
fore  
*de-*  
*ploy*

Paramet

**task**  
A  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.



take\_ov

Take  
over  
man  
age-  
men  
of  
this  
task  
node  
from  
a  
deac  
con-  
duc-  
tor.

If  
con-  
duc-  
tors  
host  
main  
tain  
a  
stati  
re-  
la-  
tion-  
ship  
to  
node  
this  
meth  
shou  
be  
im-

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

**For exam**

Neu  
tron  
mus  
for-  
ward  
DHCP  
BOOT  
re-  
ques  
to  
a

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.

con-  
duc-  
tor  
which  
has  
pre-  
pare  
the

### Parameter

**Task Agent** An instance containing the node to act on.

## tear\_down

Tear down a previous deployment on the task node. Give a node that has been previously

sary to un-deploy that node.

de-  
 ploy  
 to,  
 do  
 all  
 clea  
 and  
 tear  
 dow  
 nec-  
 es-

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 sta-  
 tus  
 of  
 the  
 de-  
 ploy  
 One  
 of  
 iron

**validat**  
 Val-  
 i-  
 date  
 the  
 drive  
 spec  
 Nod  
 de-  
 ploy  
 men  
 info

the required information for this interface to function.

long-running checks.

This  
meth  
val-  
i-  
date  
whe  
the  
drive  
and/  
in-  
stan  
prop  
er-  
ties  
of  
the  
task  
node  
con-  
tains

This  
meth  
is  
of-  
ten  
ex-  
e-  
cute  
syn-  
chro  
in  
API  
re-  
ques  
so  
it  
shou  
not  
con-  
duct

**Parame**  
**tas**  
A  
Task  
ager  
in-  
stan  
con-

tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**

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)

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Ins*  
 Ex-  
 am-  
 ple  
 im-  
 ple-  
 men

ta-  
tion  
of  
a  
sim-  
ple  
in-  
spec  
in-  
ter-  
face

**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

**inspect**  
In-  
spec  
hard  
ware  
  
In-  
spec  
hard  
ware  
to  
ob-  
tain

the  
 es-  
 sen-  
 tial  
 &  
 ad-  
 di-  
 tion:  
 hard  
 ware  
 prop-  
 er-  
 ties.

**Parameters**  
**task**  
 A  
 task  
 from  
 Task  
 ager

**Raises**  
 Harc  
 ware  
 spec  
 tion-  
 Fail-  
 ure,  
 if  
 un-  
 able  
 to  
 get  
 es-  
 sen-  
 tial  
 hard  
 ware  
 prop-  
 er-  
 ties.

**Returns**  
 Re-  
 sult-  
 ing  
 state  
 of  
 the  
 in-  
 spec  
 tion

the required information for this interface to function.

i.e.  
 state  
 or  
 Non  
**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  
 task  
 node  
 con-  
 tains  
 This  
 meth  
 is  
 of-  
 ten  
 ex-  
 e-  
 cute  
 syn-  
 chro  
 in  
 API  
 re-



long-running checks.

ques  
so  
it  
shou  
not  
con-  
duct

Parame

tas  
A  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

Raises

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)  
  
**class** i  
 Base  
*irc*  
*dri*  
*bas*  
*Man*  
  
 Ex-  
 am-  
 ple  
 im-  
 ple-  
 men  
 ta-  
 tion  
 of  
 a  
 sim-  
 ple  
 man  
 age-  
 men  
 in-  
 ter-  
 face

**get\_boo**  
 Get  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 for  
 a  
 node  
  
 Pro-  
 vide  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 of  
 the

node  
 Be  
 awar  
 that  
 not  
 all  
 drive  
 sup-  
 port  
 this.

**Parame**  
**tas**  
 A  
 task  
 from  
 Task  
 ager

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Returns**  
  
 A  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing:

**boot\_c**  
 Ahe  
 boot  
 de-

unknown.

vice  
 one  
 of  
*ironic*  
*command*  
*book*  
 or  
 Non  
 if  
 it  
 is  
 un-  
 know  
**persist**  
 Whe  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 or  
 not,  
 Non  
 if  
 it  
 is  
**get\_inco**  
 Get  
 cur-  
 rent  
 state  
 of  
 the  
 in-  
 di-  
 ca-  
 tor  
 of  
 the  
 hard  
 ware  
 com

po-  
 nent

Paramete

- **task**  
 A  
 task  
 from  
 Task  
 ager

- **component**  
 The  
 hard  
 ware  
 com  
 po-  
 nent  
 one  
 of  
*ironic*  
*component*  
*component*

- **indicator**  
 In-  
 di-  
 ca-  
 tor  
 ID  
 (as  
 re-  
 port  
 by  
*get\_*

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 com

is specified.

po-  
 nent  
 or  
 in-  
 di-  
 ca-  
 tor

**Raises**

Miss-  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Returns**

Cur-  
 rent  
 state  
 of  
 the  
 in-  
 di-  
 ca-  
 tor,  
 one  
 of  
*irc*  
*com*  
*ino*

**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

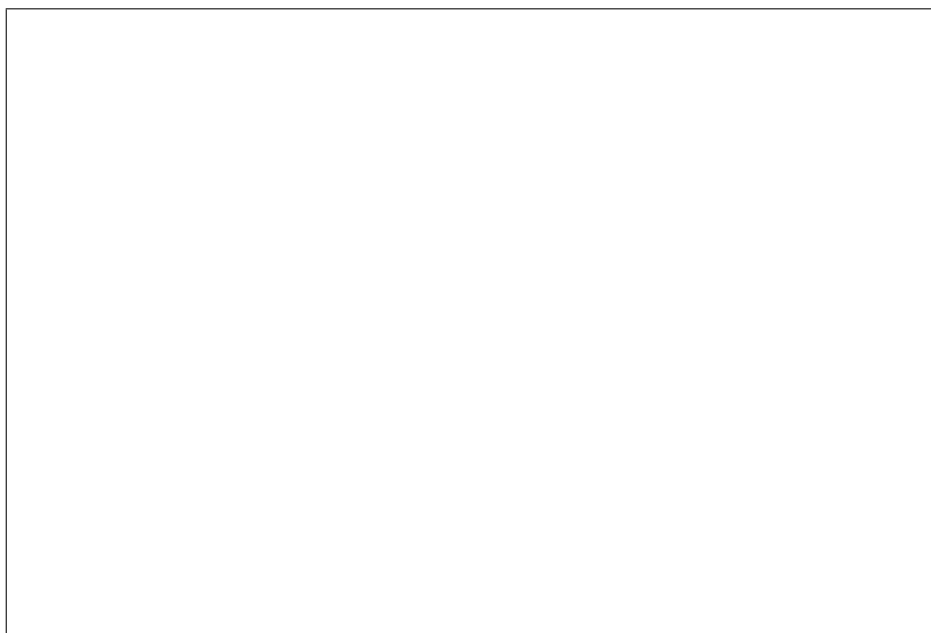
Fail  
 ToP  
 eSen

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  
sen-  
sor  
type  
whic  
can

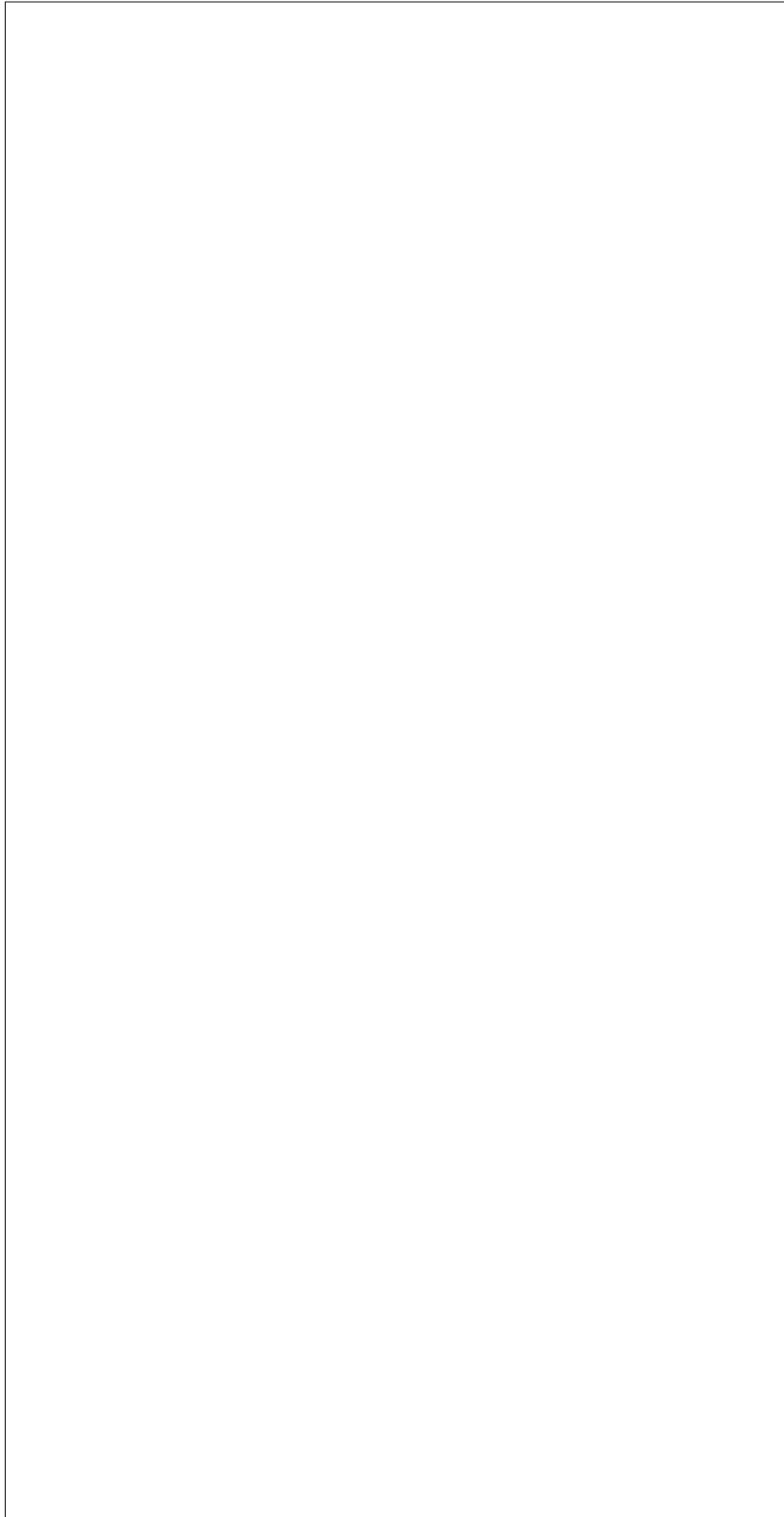
be processed by Ceilometer. eg,



(continues on next page)

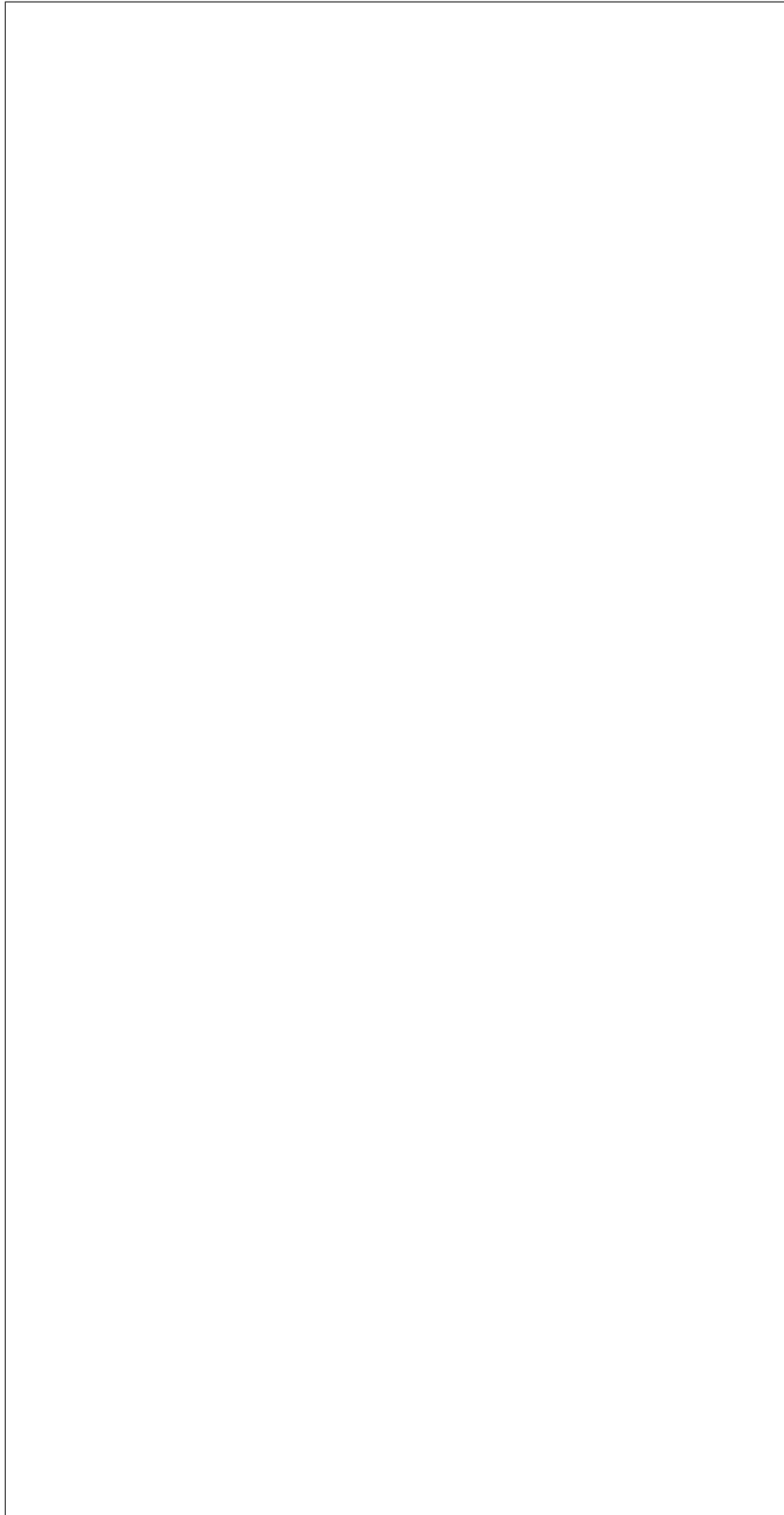


(continued from previous page)



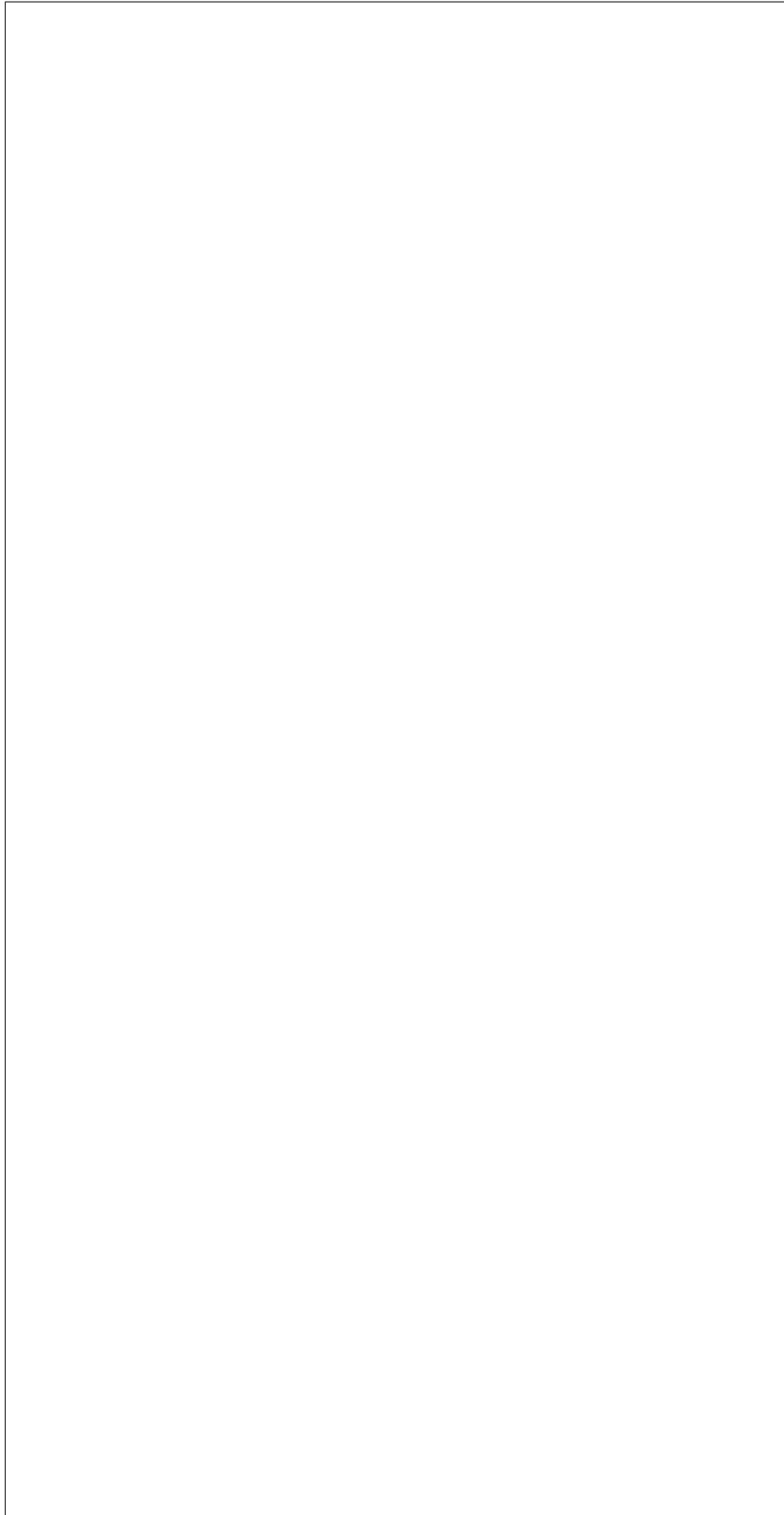
(continues on next page)

(continued from previous page)



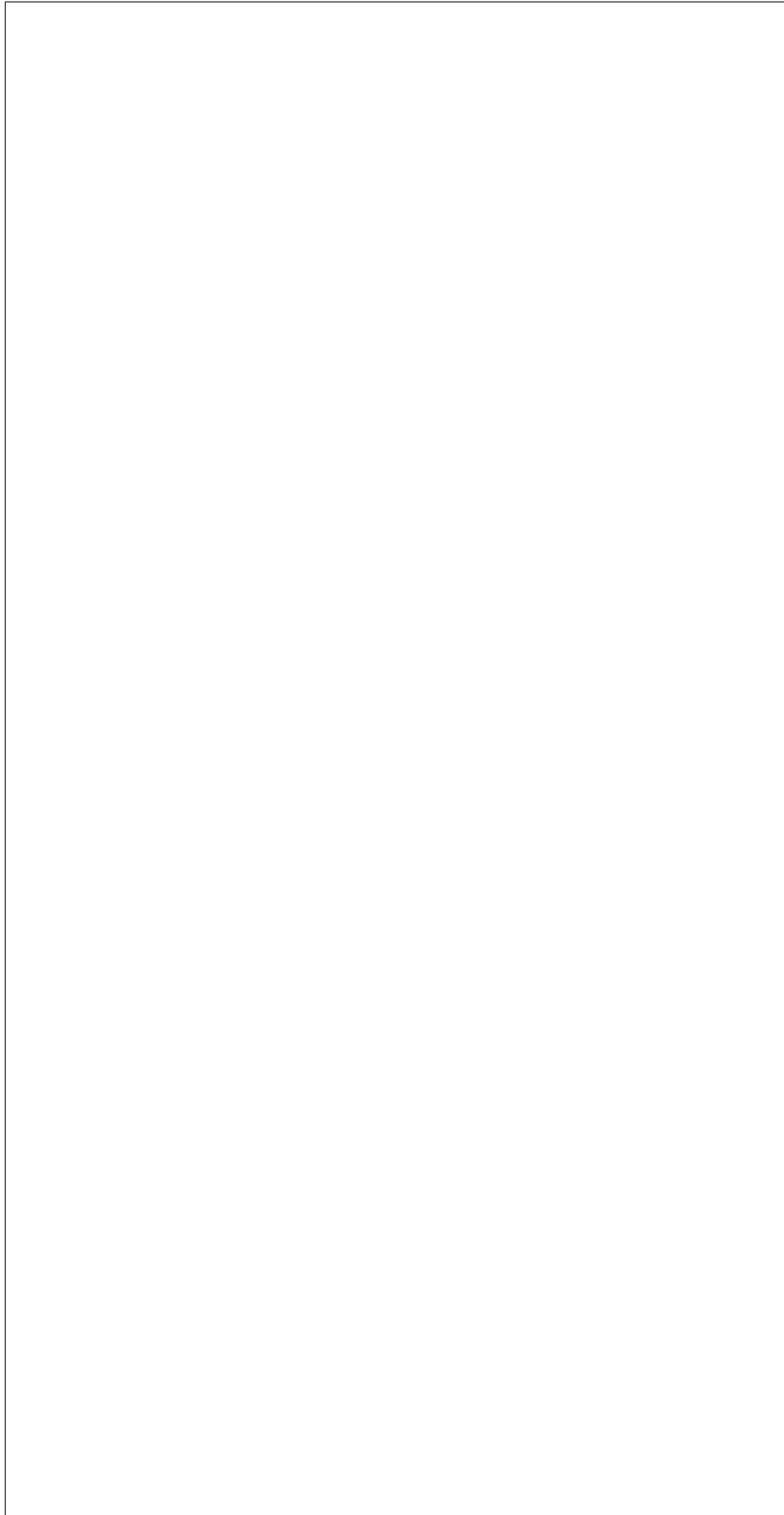
(continues on next page)

(continued from previous page)



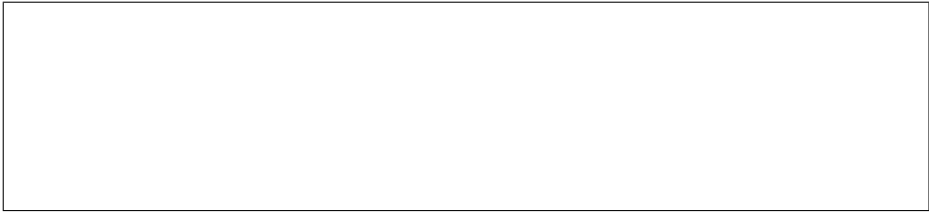
(continues on next page)

(continued from previous page)



(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-  
 finec  
 in  
*irc*  
*com*  
*boo*

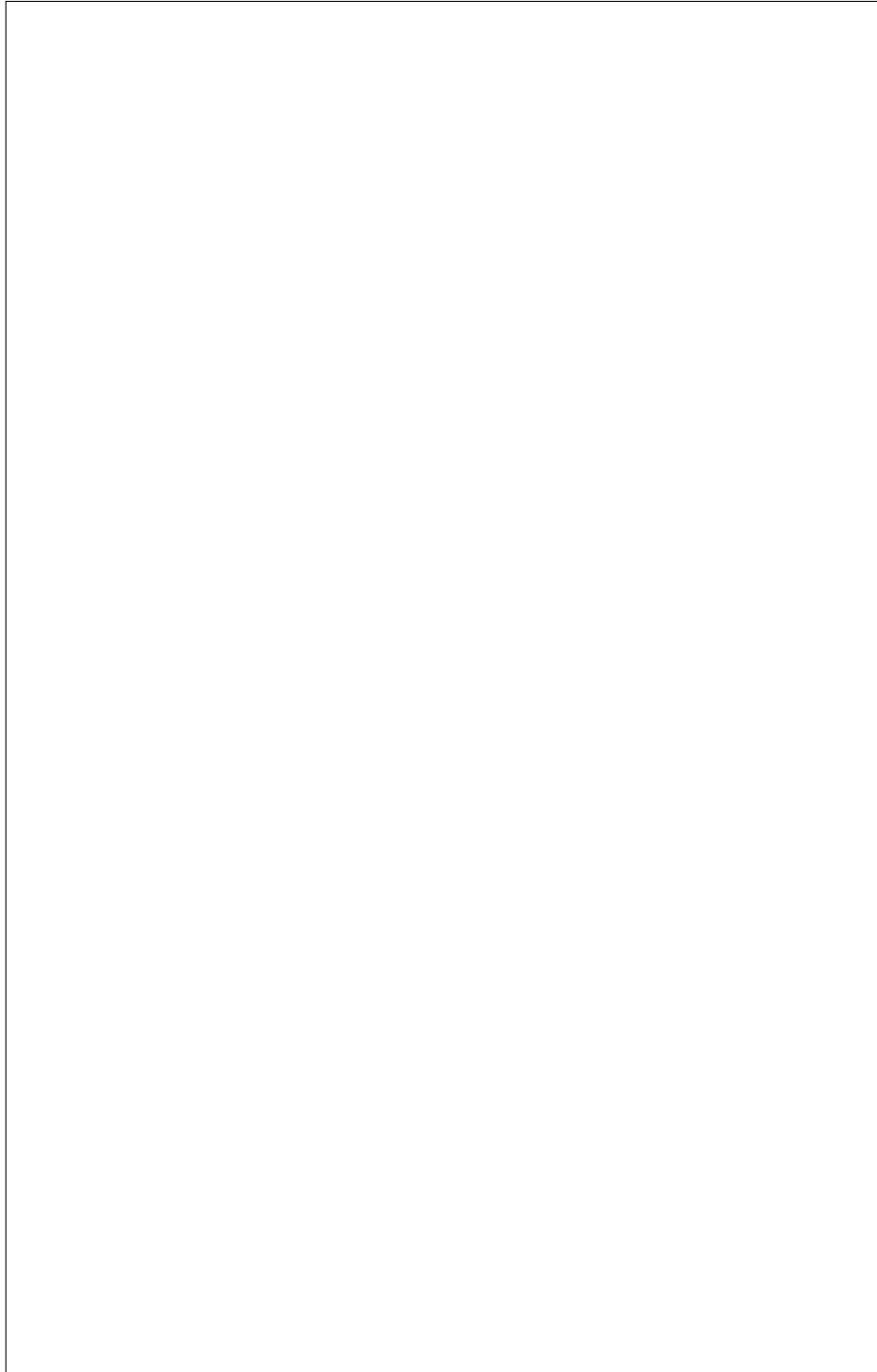
**get\_sup**  
 Get  
 a  
 map  
 of  
 the  
 sup-  
 port

nent, otherwise return indicators for all existing components.

in-  
 di-  
 ca-  
 tors  
 (e.g.  
 LED  
**Parameters**  
 •  
**task**  
 A  
 task  
 from  
 Task  
 ager  
 •  
**com**  
 If  
 not  
*Non*  
 re-  
 turn  
 in-  
 di-  
 ca-  
 tor  
 in-  
 for-  
 ma-  
 tion  
 for  
 just  
 this  
 com  
 po-  
**Returns**  
 A  
 dic-  
 tio-  
 nary  
 of  
 hard  
 ware  
 com  
 po-  
 nent  
 (*ir*

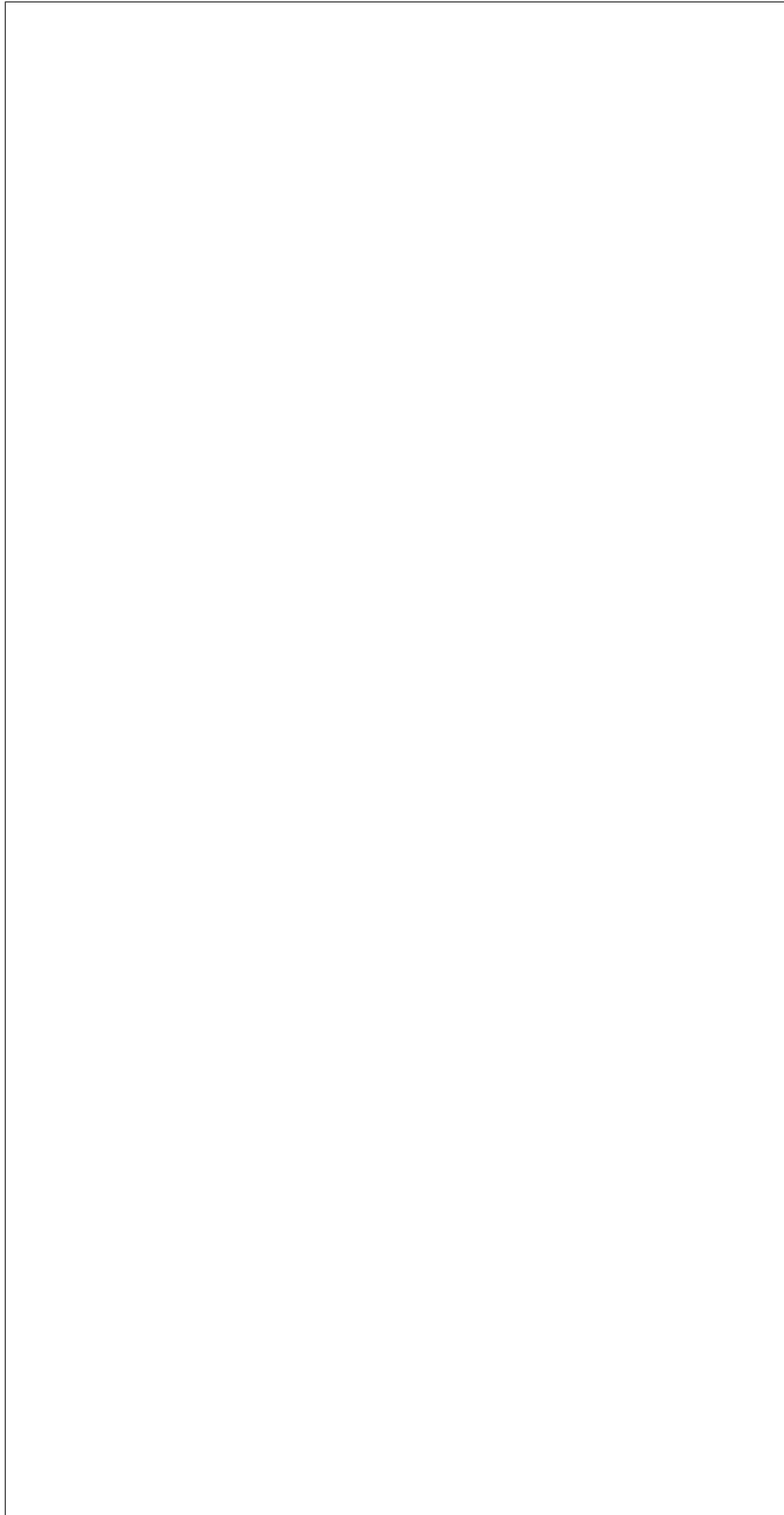
com  
com  
as  
keys  
with  
val  
ues  
be-  
ing

dictionaries having indicator IDs as keys and indicator properties as values.



(continues on next page)

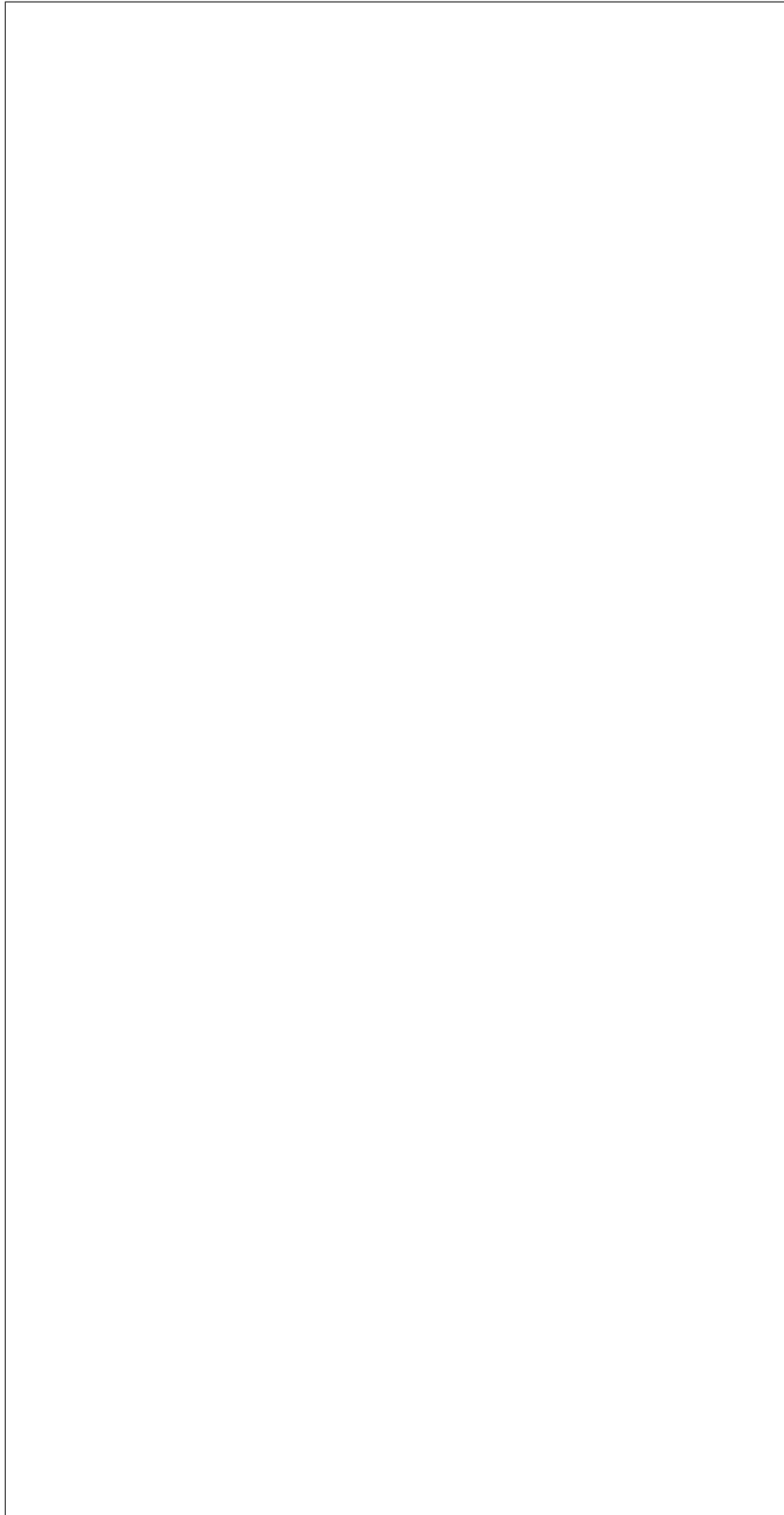
(continued from previous page)



(continues on next page)

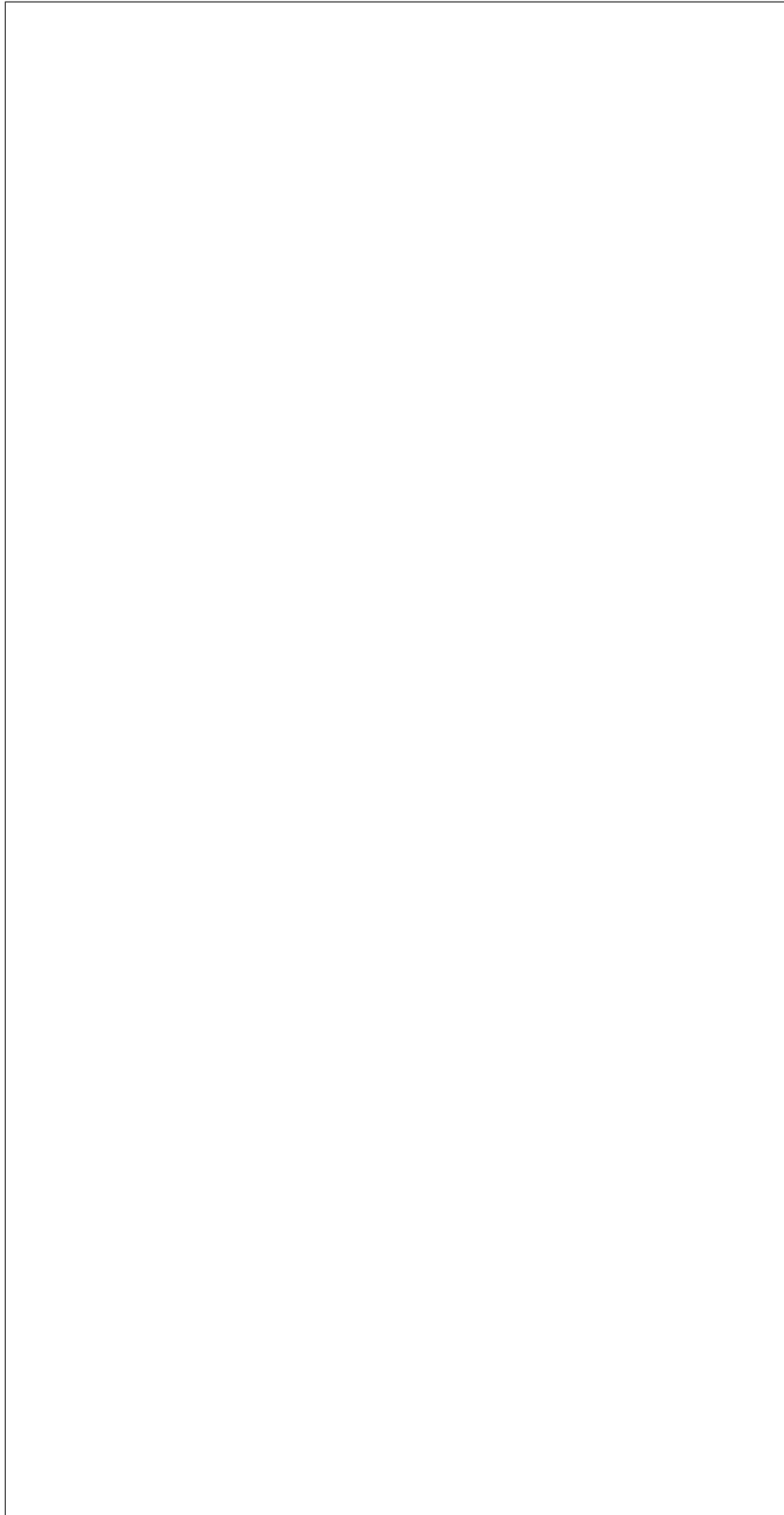


(continued from previous page)



(continues on next page)

(continued from previous page)



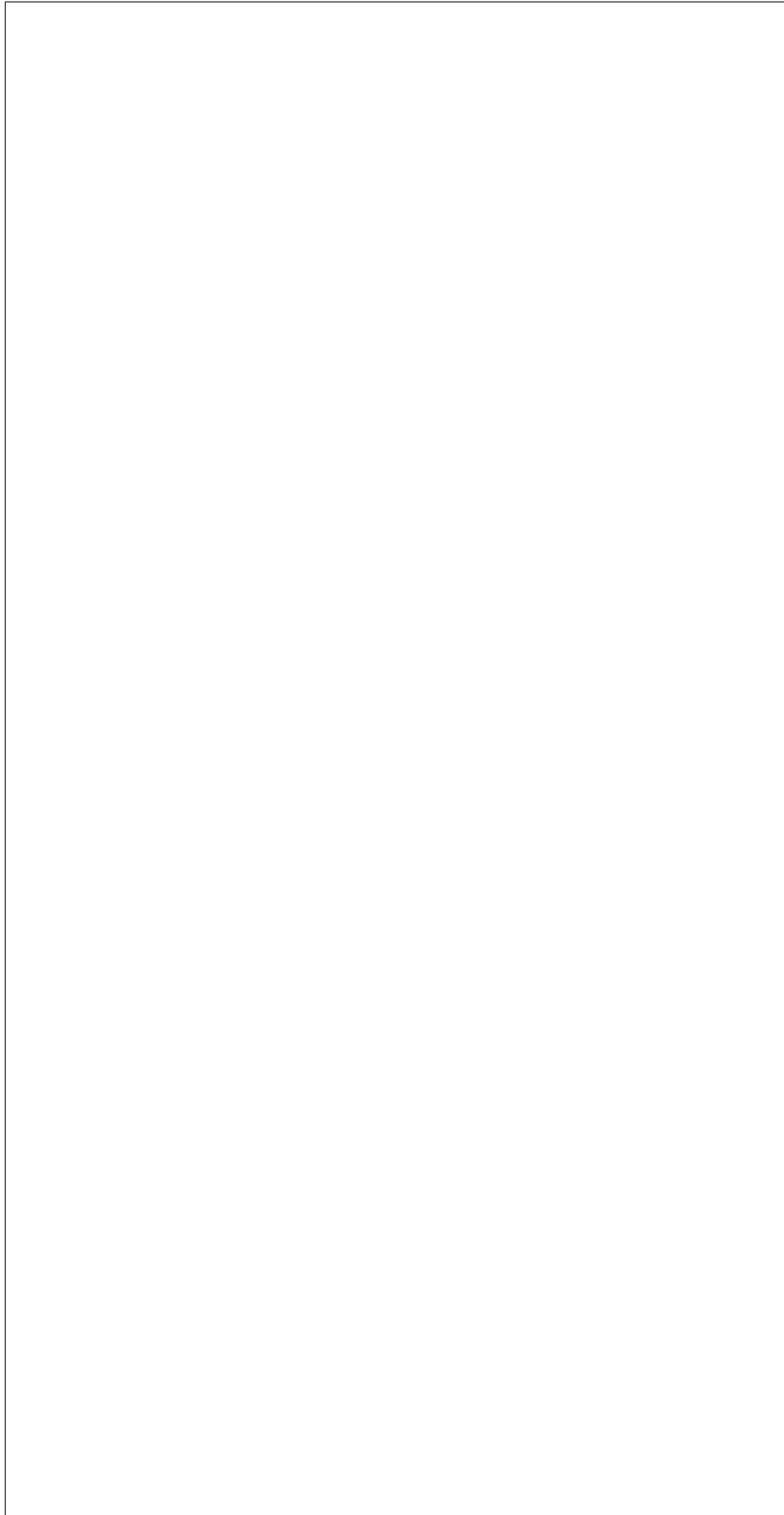
(continues on next page)

(continued from previous page)



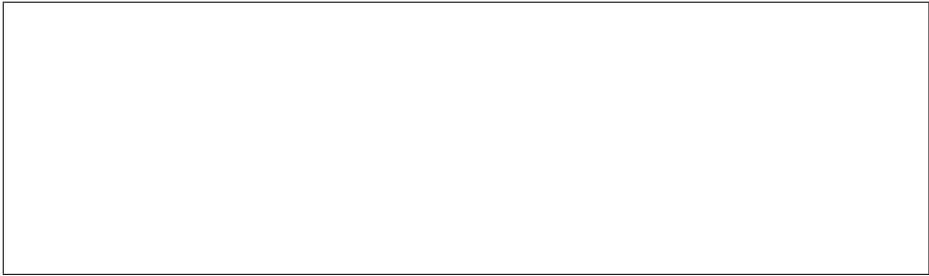
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



**set\_boot**

Set the boot device for a node. Set the boot device to use on next re-boot of the node.

**Parameters**

- task**  
 A task from Task agent.
- device**  
 The boot device one of

not. Default: False.

irc  
com  
boo

- - per**
    - Boo
    - valu
    - True
    - if
    - the
    - boot
    - de-
    - vice
    - will
    - per-
    - sist
    - to
    - all
    - fu-
    - ture
    - boot
    - Fals
    - if

Raises

In-  
valic  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valic  
boot  
de-  
vice  
is  
spec  
i-  
fied.

Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-

the required information for this interface to function.

Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

#### **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  
task  
node  
con-  
tains

This  
meth  
is  
of-  
ten

long-running checks.

ex-  
 e-  
 cute  
 syn-  
 chro  
 in  
 API  
 re-  
 ques  
 so  
 it  
 shou  
 not  
 con-  
 duct

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 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)

**class** i  
Base  
*irc*  
*dri*  
*bas*  
*Pow*

Ex-  
am-  
ple  
im-  
ple-  
men  
ta-  
tion  
of  
a  
sim-  
ple  
pow  
in-  
ter-  
face

**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  
 on.

**Raises**

Miss-  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

**Returns**

A  
 pow  
 state  
 One  
 of  
*irc*  
*com*  
*sta*

**get\_pro**

Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

Returns

dictionary of properties and descriptions of entries.

get\_supported

Get a list of the supported power states.

Parameter

**task**
 A Task object representing the node to act on.

Returns

A list with the supported power states defined in



to  
act  
on.

- **timeout**  
time  
out  
(in  
sec-  
onds  
pos-  
i-  
tive  
in-  
te-  
ger  
(>  
0)  
for  
any  
pow  
state  
Non

indicates to use default timeout.

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

**set\_power**  
Set  
the  
pow  
state  
of  
the

task  
 node

Parameters

- task**  
 A Task object representing the task to be executed. The task object must have a `name` attribute and a `run` method. The `run` method should return a tuple containing the task's status and a dictionary of task data. The task data dictionary should contain the following keys: `exit_code`, `stdout`, `stderr`, `task_id`, and `task_name`. The `task_id` key should contain the task's unique identifier, and the `task_name` key should contain the task's name. The `exit_code` key should contain the task's exit code, and the `stdout` and `stderr` keys should contain the task's standard output and standard error, respectively. The `run` method should also set the `task_id` and `task_name` attributes of the task object.
- power**  
 Any power state from the `ironic-compute-client` library.
- timeout**  
 The timeout in seconds for the task. If the task does not complete within the timeout, the task will be marked as failed. The timeout can be set to `None` to indicate that the task should use the default timeout.

indicates to use default timeout.

Raises

Miss
 ing-
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 a
 re-
 quir
 pa-
 ram-
 e-
 ter
 is
 miss
 ing.

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
 task
 node
 con-

the required information for this interface to function.

long-running checks.

tains  
  
 This  
 meth  
 is  
 of-  
 ten  
 ex-  
 e-  
 cute  
 syn-  
 chro  
 in  
 API  
 re-  
 ques  
 so  
 it  
 shou  
 not  
 con-  
 duct  
  
**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**Raises**  
 In-  
 valio  
 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

### class i

Base  
*irc*  
*dri*  
*bas*  
*RAI*

Ex-  
am-  
ple  
im-  
ple-  
men  
ta-  
tion  
of  
sim-  
ple  
RAI  
In-  
ter-  
face

### create\_

Cre-  
ates  
RAI  
con-  
fig-  
u-  
ra-

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the node.

#### Parameters

- **task**  
A TaskManager instance.
- **create\_raid\_info**  
Setting this to False indicates

ified in the nodes `target_raid_config`. Default value is `True`.

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

not  
to  
cre-  
ate  
root  
vol-  
ume  
that  
is  
spec

- **cre**  
Set-  
ting  
this  
to  
Fals  
in-  
di-  
cate  
not  
to  
cre-  
ate  
non-  
root  
vol-  
ume  
(all  
ex-

- **del**  
Set-  
ting  
this  
to  
True  
in-  
di-  
cate  
to  
dele  
RAI  
con-  
fig-  
u-  
ra-  
tion

creating the new configuration.

chronously, or None if it is complete.

prior  
to  
  
**Returns**  
 state  
(clearing)  
or  
state  
(deploying)  
if  
RAI  
con-  
fig-  
u-  
ra-  
tion  
is  
in  
prog  
asym

**delete\_**  
 Dele  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give  
 node  
  
 This  
 meth  
 dele  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give  
 node

ration is deleted, `node.raid_config` should be cleared by the implementation.

it is complete.

Af-  
ter  
RAI  
con-  
fig-  
u-

#### Parameters

**task**  
A  
Task  
ager  
in-  
stan-

#### Returns

state  
(clea  
ing)  
or  
state  
(de-  
ploy  
men  
if  
dele  
tion  
is  
in  
prog  
asyn  
chro  
or  
Non  
if

#### get\_properties

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** *irc*  
 Base  
*irc*  
*dri*  
*bas*  
*Res*

Ex-  
 am-  
 ple  
 im-  
 ple-  
 men  
 ta-  
 tion  
 of  
 a  
 sim-  
 ple  
 res-  
 cue  
 in-  
 ter-  
 face

**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  
 op-  
 er-  
 a-  
 tion  
 fails

**Returns**

state  
 if  
 res-  
 cue  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 state  
 if  
 it  
 is  
 com  
 plete

**unrescu**

Tear  
 dow  
 the  
 res-  
 cue  
 en-  
 vi-  
 ron-  
 men  
 and  
 re-  
 turn  
 to  
 nor-  
 mal.

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-



ation fails.

tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
**Raises**  
 In-  
 stan-  
 Un-  
 res-  
 cue-  
 Fail-  
 ure  
 if  
 node  
 val-  
 i-  
 da-  
 tion  
 or  
 un-  
 res-  
 cue  
 op-  
 er-  
**Returns**  
 state  
 if  
 it  
 is  
 suc-  
 cess  
 ful  
**validat**  
 Val-  
 i-  
 date  
 the  
 drive  
 spec  
 Nod  
 de-  
 ploy  
 men  
 info  
 This

the required information for this interface to function.

long-running checks.

meth  
 val-  
 i-  
 date  
 whe  
 the  
 driv  
 and/  
 in-  
 stan  
 prop  
 er-  
 ties  
 of  
 the  
 task  
 node  
 con-  
 tains

This  
 meth  
 is  
 of-  
 ten  
 ex-  
 e-  
 cute  
 syn-  
 chro  
 in  
 API  
 re-  
 ques  
 so  
 it  
 shou  
 not  
 con-  
 duct

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-

ing  
the  
node  
to  
act  
on.

#### **Raises**

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)

#### **class i**

Base  
*irc*  
*dri*  
*bas*  
*Sto*

Ex-  
am-  
ple  
im-  
ple-  
men  
ta-

tion  
 of  
 sim-  
 ple  
 stor-  
 age  
 In-  
 ter-  
 face

**attach\_**

In-  
 form  
 the  
 stor-  
 age  
 sub-  
 sys-  
 tem  
 to  
 at-  
 tach  
 all  
 vol-  
 ume  
 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-  
 ume  
 for  
 the  
 node

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

**get\_properties**  
 Re-  
 turn  
 the  
 prop  
 er-  
 ties  
 of  
 the  
 in-  
 ter-  
 face

**Returns**  
 dic-  
 tio-  
 nary  
 of  
 <pro  
 erty  
 nam  
 de-  
 scrip

ten by Ionic.

tion.  
 en-  
 tries  
**should\_**  
 De-  
 ter-  
 mine  
 if  
 de-  
 ploy  
 shou  
 per-  
 form  
 the  
 im-  
 age  
 writ  
 out.  
**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
**Returns**  
 Boo  
 valu  
 to  
 in-  
 di-  
 cate  
 if  
 the  
 in-  
 ter-  
 face  
 ex-  
 pect  
 the  
 im-  
 age  
 to  
 be  
 writ  
**Raises**  
 Un-  
 sup-

port  
ed-  
Driv  
ten-  
sion

**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  
task  
node  
con-  
tains

the required information for this interface to function.

This  
meth  
is  
of-  
ten  
ex-  
e-  
cute  
syn-  
chro  
in  
API

long-running checks.

re-  
 ques  
 so  
 it  
 shou  
 not  
 con-  
 duct

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 In-  
 valie  
 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)

**class** i  
 Base  
*irc*  
*dri*  
*bas*  
*Ven*

Ex-  
 am-  
 ple  
 im-  
 ple-  
 men  
 ta-  
 tion  
 of  
 a  
 ven-  
 dor  
 pass  
 in-  
 ter-  
 face

**first\_m**

**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**

- **tas**  
A  
task  
from  
Task  
ager
- **met**  
Met  
to  
be  
val-  
i-  
date
- **kwa**  
Info  
for

faces.

ac-  
tion.

**Raises**  
Un-  
sup-  
port-  
ed-  
Driv-  
ten-  
sion  
if  
meth  
can  
not  
be  
map  
to  
the  
sup-  
port  
in-  
ter-

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
kwa  
does  
not  
con-  
tain  
meth

**Raises**  
Miss  
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  
 pass

**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\_m

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

faces.

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten  
 sion  
 if  
 meth  
 can  
 not  
 be  
 map  
 to  
 the  
 sup-  
 port  
 in-  
 ter-

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 kwa  
 does  
 not  
 con-  
 tain  
 meth

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu

## ironic.drivers.modules.image\_cache module

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  
keep-  
ing  
cach-  
of  
the  
lat-  
est  
im-  
ages  
Files  
with

ter images after we get listing and before we actually delete files.

reached, even if it is possible to clean up more files

link  
 coun  
 >1  
 are  
 neve  
 dele  
 Pro-  
 tecte  
 by  
 glob  
 lock  
 so  
 that  
 no  
 one  
 mes  
 with  
 mas

Param  
 amo  
 if  
 pres  
 amo  
 of  
 spac  
 to  
 re-  
 clain  
 in  
 byte  
 clear  
 ing  
 will  
 stop  
 if  
 this  
 goal  
 was

fetch\_i  
 Fetc  
 im-  
 age  
 by  
 give  
 href  
 to  
 the  
 des-



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.

#### Parameters

- **href**  
im-  
age  
UUU  
or  
href  
to  
fetc
- **des**  
des-  
ti-  
na-  
tion  
file  
path
- **ctx**  
con-

text

- for**  
 bool  
 valu  
 whe  
 to  
 con-  
 vert  
 the  
 im-  
 age  
 to  
 raw  
 for-  
 mat

ironic.

Ex-  
 plic-  
 itly  
 clea  
 cach  
 base  
 on  
 their  
 pri-  
 or-  
 ity  
 (if  
 re-  
 quir

This  
 clea  
 up  
 the  
 cach  
 to  
 free  
 up  
 the  
 amo  
 of  
 spac  
 re-  
 quir  
 for  
 the  
 im-  
 ages

ages\_info. The caches are cleaned up one after the other in the order of their priority. If we still cannot free up enough space after trying all the caches, this method throws exception.

## Parameters

- **ctx**  
con-  
text
- **dir**  
the  
di-  
rec-  
tory  
(of  
the  
cach  
to  
be  
free  
up.
- **ima**  
a  
list  
of  
tu-  
ples  
of  
the  
form  
(im-  
age\_  
for  
whic  
spac  
is  
to  
be  
cre-  
ated

in cache.

## Raises

In-  
suf-  
fi-  
cien

ing all the caches.

ironic.drivers.modules.image\_utils module

Disk  
 ex-  
 cep-  
 tion,  
 if  
 we  
 can-  
 not  
 free  
 up  
 enou  
 spac  
 af-  
 ter  
 try-  
  
 ironic.  
 Dec  
 o-  
 ra-  
 tor  
 meth  
 for  
 addi  
 clea  
 pri-  
 or-  
 ity  
 to  
 a  
 clas  
  
 class i  
 Base  
 obj  
  
 publish  
 Mak  
 im-  
 age  
 file  
 dow  
 load  
 able  
  
 De-  
 penc  
 ing

servers document root and returns publicly accessible URL leading to the given file.

#### Parameters

- **image\_path**  
path to file to publish
- **object\_name**  
name of the published file

#### Returns

a URL to download published file

#### unpublished

With draw the

published - Swift or local HTTP servers document root.

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  
whe  
it  
has  
been

**Param**  
  
**obj**  
nam  
of  
the  
pub-  
lishe  
file  
(op-  
tiona

**classme**  
With  
draw  
the  
im-  
age  
pre-  
vi-  
ousl  
mad

published - Swift or local HTTP servers document root.

#### Parameters

- **node**  
the node for which image was published
- **prefix**  
object name prefix.
- **suffix**  
object name

suf-  
fix.  
  
 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

**Paramet**

**tas**

an  
iron.  
node  
ob-  
ject.

ironic.

Dele  
the  
ISO  
if  
it  
was  
cre-  
ated  
for  
the  
in-  
stan

**Paramet**

**tas**

A  
task  
from  
Task  
ager

ironic.

Pre-  
pare  
boot  
ISO  
im-  
age

Buil  
boot  
ISO  
out  
of  
[in-  
stan  
[in-  
stan  
and  
[dri  
if  
pres

*ramdisk\_id* from *[instance\_info]/image\_source* Glance image metadata.

age.

Paramet

- **task\_id**
 a Task Identifier containing the node to act on.
- **disk\_id**
 De-

ploy  
men  
in-  
for-  
ma-  
tion  
of  
the  
node

- - roo**  
Roo  
UUI

**Returns**  
boot  
ISO  
HTT  
URI

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
any  
of  
the  
re-  
quir  
pa-  
ram-  
e-  
ters  
are  
miss

ing.

**Raises**  
In-  
valic  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
any

of  
 the  
 pa-  
 ram-  
 e-  
 ters  
 have  
 in-  
 valid  
 valu

**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  
 writ  
 it  
 into  
 a  
 disk  
 im-  
 age  
 that

a virtual USB device. Images stored in Swift are downloaded first.

can  
be  
at-  
tach  
e.g.  
to

## Parameters

- **task**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **config**  
Con-  
fig  
drive  
as  
a  
base  
enco-  
strin

## Raises

Im-  
age-  
Cre-  
ation  
Fail-  
if  
it  
faile  
whil  
cre-  
at-  
ing  
the  
im-

age.

**Raises**

Swi  
Op-  
er-  
a-  
tionl  
if  
any  
op-  
er-  
a-  
tion  
with  
Swi  
fails

**Returns**

im-  
age  
URI  
for  
the  
im-  
age.

ironic.

Pre-  
pare  
de-  
ploy  
or  
res-  
cue  
ISO  
im-  
age

Buil  
boot  
ISO  
out  
of  
*[driv*  
or  
*[driv*  
and  
*[driv*  
then  
push  
built

turn temporary Swift URL to the image.

will be written into an appropriate location on the final ISO.

im-  
age  
up  
to  
Glar  
and  
re-

If  
net-  
worl  
in-  
ter-  
face  
sup-  
plies  
net-  
worl  
con-  
fig-  
u-  
ra-  
tion  
(*net-*  
*worl*  
a  
*net-*  
*worl*

## Paramet

- **tas**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **par**  
a

to kernel command line.

dic-  
tio-  
nary  
con-  
tain-  
ing  
pa-  
ram-  
e-  
ter  
nam  
>val  
map  
ping  
to  
be  
pass

- **mod**  
ei-  
ther  
de-  
ploy  
or  
res-  
cue.

- **d\_i**  
De-  
ploy  
men  
in-  
for-  
ma-  
tion  
of  
the  
node

**Returns**  
boot  
ISO  
HTT  
URI

**Raises**  
Miss  
ing-  
Pa-  
ram-



ing.

e-  
 ter-  
 Valu  
 if  
 any  
 of  
 the  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 any  
 of  
 the  
 pa-  
 ram-  
 e-  
 ters  
 have  
 in-  
 valid  
 valu

Raises

Im-  
 age-  
 Cre-  
 ation  
 Fail  
 if  
 cre-  
 at-  
 ing  
 ISO  
 im-  
 age  
 faile

ironic.

Prepare an image with the given content.

If content is already read an HTTP URI return it unchanged

Parameters

- **task**
 a Task object containing the node to act on.
- **content**
 Content as a

string  
with  
a  
file  
name  
or  
byte  
with  
con-  
tents

- **pre**  
Pre-  
fix  
to  
use  
for  
the  
ob-  
ject  
name

#### **Raises**

Im-  
age-  
Cre-  
ation  
Failure  
if  
it  
fails  
while  
cre-  
at-  
ing  
the  
im-  
age.

#### **Raises**

Swi-  
Op-  
er-  
a-  
tion  
if  
any  
op-  
er-  
a-  
tion  
with

Swift  
 fails.  
**Returns**  
 im-  
 age  
 URL  
 for  
 the  
 im-  
 age.

ironic.

Pre-  
 pare  
 the  
 flopp  
 im-  
 age  
 for  
 pass  
 ing  
 the  
 pa-  
 ram-  
 e-  
 ters.  
 This  
 meth  
 pre-  
 pare  
 a  
 tem-  
 po-  
 rary  
 VFA  
 files  
 tem  
 im-  
 age  
 and  
 adds  
 a  
 file  
 into  
 the  
 im-

age 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.

Paramet

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **par**  
 a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 pa-  
 ram-  
 e-  
 ter  
 nam  
 >val  
 map  
 ping  
 to  
 be  
 pass

to deploy or rescue image via floppy image.

Raises

Im-  
 age-  
 Cre-  
 ation  
 Fail  
 if  
 it  
 faile  
 whil  
 cre-  
 at-

ing  
 the  
 flopp  
 im-  
 age.  
**Raises**  
 Swi  
 Op-  
 er-  
 a-  
 tionl  
 if  
 any  
 op-  
 er-  
 a-  
 tion  
 with  
 Swi  
 fails

**Returns**  
 im-  
 age  
 URI  
 for  
 the  
 flopp  
 im-  
 age.

ironic.drivers.modules.inspect\_utils module

ironic.

Cre-  
 ate  
 iron  
 port  
 from  
 MA  
 ad-  
 dres  
 data  
 dict.  
 Cre-  
 ates  
 iron

operator. Helper argument to detect the MAC address `get_mac_address` defaults to value part of MAC address dict key-value pair.

#### Parameters

- task**  
 A Task manager instance
- mac\_address**  
 A dictionary of MAC addresses returned by node inspection.
- get\_mac\_address**  
 a function

key-value pair of the previous `macs` argument.

**ironic.drivers.modules.inspector module**

tion  
 to  
 get  
 the  
 MA  
 ad-  
 dres  
 from  
 mac  
 item  
 A  
 mac  
 item  
 is  
 the  
 dict

Modules r

<https://pypi.org/project/ironic>

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Ins*

In-  
 banc  
 in-  
 spec  
 tion  
 via  
 iron  
 insp  
 proj

**abort** (*t*

Abos  
 hard  
 ware  
 in-  
 spec



tion.

#### Parameters

**task**

a

task

from

Task

ager

#### get\_properties

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-

sults will be checked in a periodic task.

tic-  
 u-  
 lar  
 im-  
 ple-  
 men  
 ta-  
 tion  
 only  
 start  
 in-  
 spec  
 tion  
 us-  
 ing  
 iron  
 insp  
 Re-

**Parame**  
**tas**  
 a  
 task  
 from  
 Task  
 ager

**Returns**  
 state

**Raises**  
 Harc  
 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

**Parameters**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

ironic.drivers.modules.ipmitool module

IPM  
 pow  
 man  
 ager  
 drive  
 Uses  
 the  
 ip-  
 mi-  
 tool  
 com  
 man  
 ([http](#)

includes setting the boot device, getting a serial-over-LAN console, and controlling the power state of the machine.

PROVIDES DIFFERENT COMMAND-LINE OPTIONS AND *IS NOT SUPPORTED* BY THIS DRIVER.

//  
ipmi  
sour  
net/  
to  
re-  
mote  
man  
age  
hard  
ware  
This

NOT  
THA  
CER  
TAIL  
DIS  
TRO  
MA  
IN-  
STA  
oper  
BY  
DE-  
FAU  
IN-  
STE  
OF  
ip-  
mi-  
tool.  
WH

class i  
Base  
irc  
dri  
bas  
Con  
A  
base  
Con  
sole  
ter-  
face  
that  
uses

ip-  
mi-  
tool.

**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-  
e-  
ter  
is  
miss  
ing

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Man*

**detect\_**

De-  
tects  
and  
re-  
turn  
the  
hard  
ware  
ven-  
dor.

**Parame**

**tas**  
A  
task  
from  
Task  
ager

state is specified.

Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 com  
 po-  
 nent  
 in-  
 di-  
 ca-  
 tor  
 or

Raises  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

Returns  
 Strin  
 rep-  
 re-  
 sent  
 ing  
 the  
 BM  
 re-  
 port

returns None.

Ven-  
 dor  
 or  
 Man  
 u-  
 fac-  
 turer  
 oth-  
 er-  
 wise  
  
**get\_boot**  
 Get  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 for  
 the  
 task  
 node  
  
 Re-  
 turn  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 of  
 the  
 node  
  
**Paramete**  
**task**  
 a  
 task  
 from  
 Task  
 ager  
  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu



if  
 re-  
 quir  
 IPM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

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-  
 e-  
 ter  
 is  
 miss  
 ing.

Returns

a  
 dic-  
 tio-  
 nary

unknown.

con-  
tain-  
ing:

**boot\_c**  
the  
boot  
de-  
vice  
one  
of  
*irc*  
*com*  
*boo*  
or  
Non  
if  
it  
is  
un-  
know

**persist**  
Whe  
the  
boot  
de-  
vice  
will  
per-  
sist  
to  
all  
fu-  
ture  
boot  
or  
not,  
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\_sen

Get  
 sen-  
 sors  
 data

Paramete

tas  
 a  
 Task  
 ager  
 in-  
 stan

Raises

Failu  
 To-  
 Get-  
 Sen-  
 sor-  
 Data  
 whe  
 get-  
 ting  
 the  
 sen-  
 sor  
 data  
 fails

Raises

Failu  
 ToP  
 eSer  
 sor-  
 Data

whe  
 pars  
 ing  
 sen-  
 sor  
 data  
 fails

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 ipmi  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

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

**Returns**  
 A  
 list  
 with  
 the  
 sup-  
 port  
 boot  
 de-  
 vice  
 de-  
 fined  
 in  
*irc*  
*com*  
*boo*

**inject\_**  
 In-  
 ject  
 NM  
 Non  
 Mas

able  
 In-  
 ter-  
 rupt  
  
 In-  
 ject  
 NM  
 (Nor  
 Mas  
 able  
 In-  
 ter-  
 rupt  
 for  
 a  
 node  
 im-  
 me-  
 di-  
 ately

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 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

**Paramete**

- **task**  
a  
task  
from  
Task  
ager

- **dev**  
the  
boot  
de-  
vice  
one  
of  
*irc*  
*com*  
*boo*

- **per**  
Boo

not. Default: False.

valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 Fals  
 if

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 boot  
 de-  
 vice  
 is  
 spec  
 i-  
 fied

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 ipmi  
 pa-



ram-  
e-  
ters  
are  
miss  
ing.

**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-  
plie  
task  
node  
con-  
tains  
the  
re-  
quir  
cre-

dentials information.

**Parameter**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 IPM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

**class** i  
 Base  
*irc*  
*dri*  
*bas*

Pow

**get\_pow**  
 Get  
 the  
 cur-  
 rent  
 pow  
 state  
 of  
 the  
 task.  
 node

**Paramete**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 one  
 of  
 iron.  
 POV  
 POV  
 or  
 ER-  
 ROE

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 ipmi

pa-  
ram-  
e-  
ters  
are  
miss  
ing.

Raises

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

Raises

IP-  
MI-  
Fail-  
ure  
on  
an  
er-  
ror  
from  
ip-  
mi-  
tool  
(from  
\_pov  
call)

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

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
 cur-  
 rentl  
 not  
 used

**Returns**  
 A  
 list

with  
 the  
 sup-  
 port  
 pow  
 state  
 de-  
 fine  
 in  
*irc*  
*com*  
*sta*

**reboot**

Cy-  
 cles  
 the  
 pow  
 to  
 the  
 task  
 node

**Parame**

- **tas**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **tim**  
 time  
 out  
 (in  
 sec-  
 onds  
 pos-  
 i-  
 tive  
 in-

timeout is counted once during power off and once during power on for reboots. `None` indicates that the default timeout will be used.

#### Raises

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu-  
if  
re-  
quir-  
ipmi  
pa-  
ram-  
e-  
ters  
are  
miss-  
ing.

#### Raises

In-  
valid-  
Pa-  
ram-  
e-  
ter-  
Valu-  
if  
an  
in-  
valid-  
pow-  
state  
was  
spec-  
i-  
fied.

#### Raises

Pow-

intermediate state of the node is not `POWER_OFF`.

er-  
 State  
 Fail-  
 ure  
 if  
 the  
 fi-  
 nal  
 state  
 of  
 the  
 node  
 is  
 not  
 POW  
 or  
 the  
 in-

**set\_pow**  
 Turn  
 the  
 pow  
 on,  
 off,  
 soft  
 re-  
 boot  
 or  
 soft  
 pow  
 off.

**Parame**

- tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.



- **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  
(>  
0)  
for  
any  
pow  
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.

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valid  
pow  
state

was  
 spec  
 i-  
 fied.

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 ipmi  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing

**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-  
tains  
IPM  
cre-  
den-  
tials

Param

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  
ipmi  
pa-  
ram-  
e-  
ters  
are  
miss  
ing.

Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if

a  
 re-  
 quire  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

**class** `Ironi`

Base  
*Ironi*  
*Ironi*  
*Ironi*  
*Ironi*  
*Ironi*  
*Ironi*

A  
 Con-  
 sole  
 ter-  
 face  
 that  
 uses  
 ip-  
 mi-  
 tool  
 and  
 shel-  
 lina

**get\_con**

Get  
 the  
 type  
 and  
 con-  
 nec-  
 tion  
 in-  
 for-  
 ma-  
 tion  
 about  
 the  
 con-  
 sole

**start\_c**

Start  
a  
re-  
mote  
con-  
sole  
for  
the  
node

**Parame**  
**tas**  
a  
task  
from  
Task  
ager

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
re-  
quir  
ipmi  
pa-  
ram-  
e-  
ters  
are  
miss  
ing

**Raises**  
Pass  
wor  
File-  
Fail  
ToC  
ate  
if  
un-  
able  
to  
cre-  
ate  
a  
file

word

con-  
 tain-  
 ing  
 the  
 pass

**Raises**  
 Con  
 sole  
 ror  
 if  
 the  
 di-  
 rec-  
 tory  
 for  
 the  
 PID  
 file  
 can-  
 not  
 be  
 cre-  
 ated

**Raises**  
 Con  
 sole  
 Sub-  
 pro-  
 cess  
 Fail  
 whe  
 in-  
 vok-  
 ing  
 the  
 sub-  
 pro-  
 cess  
 faile

**stop\_cc**  
 Stop  
 the  
 re-  
 mote  
 con-  
 sole  
 ses-  
 sion  
 for

the  
node

#### Parameters

**task**  
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  
 abou  
 the  
 con-  
 sole

**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-  
 e-



ters  
 are  
 miss  
 ing

**Raises**

Pass  
 wor  
 File-  
 Fail  
 ToC  
 ate  
 if  
 un-  
 able  
 to  
 cre-  
 ate  
 a  
 file  
 con-  
 tain-  
 ing  
 the  
 pass

word

**Raises**

Con  
 sole  
 ror  
 if  
 the  
 di-  
 rec-  
 tory  
 for  
 the  
 PID  
 file  
 can-  
 not  
 be  
 cre-  
 ated

**Raises**

Con  
 sole  
 Sub-  
 pro-  
 cess  
 Fail

whe  
in-  
vok-  
ing  
the  
sub-  
pro-  
cess  
faile

**stop\_cc**  
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  
 BMC  
 with  
 IPM  
 com  
 man  
 bmc  
 re-  
 set  
 (war

Paramete

- **task**  
 a  
 Task  
 ager  
 in-  
 stan
- **http**  
 the  
 HTTP  
 meth  
 used  
 on  
 the  
 re-  
 ques
- **warn**  
 bool  
 pa-  
 ram-  
 e-  
 ter  
 to  
 de-  
 cide  
 on  
 warn  
 or  
 cold  
 re-  
 set.

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-  
 e-  
 ter  
 is  
 miss  
 ing.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 whe  
 an  
 in-  
 valid  
 valu  
 is  
 spec  
 i-  
 fied

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-  
 e-  
 ter  
 is  
 miss  
 ing.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 whe  
 an  
 in-  
 valid  
 valu  
 is  
 spec  
 i-  
 fied.

validat

Val-  
 i-  
 date  
 venc  
 spec  
 ac-  
 tions  
  
 If  
 in-  
 valid  
 raise  
 an  
 ex-  
 cep-  
 tion.  
 oth-  
 er-  
 wise  
 re-  
 turn  
 Non

Valid me

- send
- bmc

Parame

fied.

- **task**  
 a  
 task  
 from  
 Task  
 ager
- **method**  
 meth  
 to  
 be  
 val-  
 i-  
 date
- **kwargs**  
 info  
 for  
 ac-  
 tion.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 whe  
 an  
 in-  
 valid  
 pa-  
 ram-  
 e-  
 ter  
 valu  
 is  
 spec  
 i-

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-



Valu
 if
 a
 re-
 quir
 pa-
 ram-
 e-
 ter
 is
 miss
 ing.

ironic.
 Dun
 SDR
 data
 to
 a
 file.

Paramet

- **tas**
 a
 Task
 ager
 in-
 stan
- **fil**
 the
 path
 to
 SDR
 dum
 file.

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-  
e-  
ter  
is  
miss  
ing.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
whe  
an  
in-  
valid  
valu  
is  
spec  
i-  
fied.

ironic.

Sen  
raw  
byte  
to  
the  
BM  
Byte  
shou  
be  
a  
strin  
of

byte

Parameters

- **task**  
 a Task object representing the task to be executed.
- **raw**  
 a string representing the raw byte data to be sent. For example, 0x00000000.

Returns

a tuple with the standard output and standard error.

Raises

IPError, IOError, OSError, RuntimeError, ValueError, and other exceptions from the ipmitool module.

Raises

MissingParameterError

ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 whe  
 an  
 in-  
 valid  
 valu  
 is  
 spec  
 i-  
 fied.

ironic.drivers.modules.ipxe module

iPX  
 Boo  
 In-  
 ter-  
 face

class i  
 Base  
*irc*  
*dri*  
*moo*  
*pxe*  
*PXE*  
*irc*

*dri*  
*bas*  
*Boo*

**capabil**

**ipxe\_en**

**ironic.drivers.modules.iscsi\_deploy module**

**class i**

Base  
*iro*  
*dri*  
*mod*  
*age*  
 Age  
*iro*  
*dri*  
*mod*  
*age*  
 Age  
*iro*  
*dri*  
*bas*  
*Dep*  
 iSCS  
 De-  
 ploy  
 In-  
 ter-  
 face  
 for  
 depl  
 relat  
 ac-  
 tions

**clean\_u**

Clea  
 up  
 the  
 de-  
 ploy  
 men  
 en-  
 vi-

figuration files for this node.

ron-  
 men  
 for  
 the  
 task  
 node  
  
 Un-  
 links  
 TFTP  
 and  
 in-  
 stan  
 im-  
 ages  
 and  
 trig-  
 gers  
 im-  
 age  
 cach  
 clea  
 Re-  
 mov  
 the  
 TFTP  
 con-  
  
**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**deploy**  
 Star  
 de-  
 ploy  
 men  
 of  
 the  
 task

boot request to the power driver. This causes the node to boot into the deployment ramdisk and triggers the next phase of PXE-based deployment via agent heartbeats.

node  
Fetc  
in-  
stan  
im-  
age,  
up-  
date  
the  
DHCP  
port  
op-  
tions  
for  
next  
boot  
and  
is-  
sues  
a  
re-

**Parameters**  
**task**  
a  
Task  
agent  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**Returns**  
de-  
ploy  
state  
DE-  
PLC  
WAI

**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

**has\_dec**

Whe-  
the  
drive  
sup-  
port.  
de-  
com-  
pose  
de-  
ploy  
step

Pre-  
vi-  
ousl  
(sinc  
Roc  
drive  
used  
a  
sin-  
gle  
de-  
ploy  
de-  
ploy  
step  
on  
the



terface. 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.

images, fetches the TFTP image from Glance and add it to the local cache.

de-  
ploy  
in-

#### prepare

Pre-  
pare  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
for  
this  
task  
node

Gen  
er-  
ates  
the  
TFTP  
con-  
fig-  
u-  
ra-  
tion  
for  
PXE  
boot  
both  
the  
de-  
ploy  
men  
and  
user

#### Paramet

tas  
a  
Task  
ager  
in-  
stan  
con-  
tain-

ing  
the  
node  
to  
act  
on.

**Raises**

Net-  
work  
Er-  
ror:  
if  
the  
pre-  
vi-  
ous  
clea-  
ing  
port  
can-  
not  
be  
re-  
mov-  
or  
if

new cleaning ports cannot be created.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu-  
whe-  
the  
wron-  
pow-  
state  
is  
spec-  
i-  
fied  
or  
the  
wron-

driver info is specified for power management.

**Raises**

Stor-

umes.

action.

ageE  
ror  
If  
the  
stor-  
age  
drive  
is  
un-  
able  
to  
at-  
tach  
the  
con-  
fig-  
ured  
vol-

**Raises**

othe  
ex-  
cep-  
tion:  
by  
the  
node  
pow  
drive  
if  
som  
thing  
wron  
oc-  
curr  
dur-  
ing  
the  
pow

**Raises**

any  
boot  
in-  
ter-  
face  
pre-  
pare  
ex-  
cep-

or in the process of being deprecated.

tions  
 prepare  
 support  
 In-  
 di-  
 cate  
 if  
 an  
 in-  
 ter-  
 face  
 is  
 sup-  
 port  
 This  
 will  
 be  
 set  
 to  
 Fals  
 for  
 in-  
 ter-  
 face  
 whic  
 are  
 unte  
 in  
 first-  
 or  
 thiro  
 part  
 CI,  
 validat  
 Val-  
 i-  
 date  
 the  
 de-  
 ploy  
 men  
 in-  
 for-  
 ma-  
 tion  
 for

the  
task  
node

#### Parameters

**task**  
a  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

#### Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Value

#### Raises

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Value

#### **write\_i**

Met  
in-  
voke  
when  
de-  
ploy  
us-  
ing  
iSCSI  
  
This  
meth  
is  
in-  
voke

back state. This deploys the image on the node and then configures the node to boot according to the desired boot option (netboot or localboot).

Parameters

- **task**  
 a Task object containing the node
- **kwargs**  
 the kwargs passed from the heartbeat method

Raises  
 InstanceDeploymentFailure, if it

en-  
coun-  
ters  
som-  
er-  
ror  
dur-  
ing  
the  
de-  
ploy

ironic.

En-  
sure  
the  
file  
sys-  
tem  
sees  
the  
iSCS  
bloc  
de-  
vice

ironic.

Che  
if  
the  
re-  
ques  
im-  
age  
is  
large  
than  
the  
root  
par-  
ti-  
tion  
size.

Doe  
noth  
ing  
for  
who

disk  
 im-  
 ages

**Parameters**  
**task**  
 a  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 In-  
 stan-  
 ploy  
 Fail-  
 ure  
 if  
 size  
 of  
 the  
 im-  
 age  
 is  
 grea-  
 than  
 root  
 par-  
 ti-  
 tion.

ironic.  
 Re-  
 sum  
 a  
 de-  
 ploy  
 men  
 upon  
 get-  
 ting  
 POS  
 data  
 from



a callback from the deploy ramdisk.

de-  
ploy  
ram  
  
This  
meth  
raise  
no  
ex-  
cep-  
tions  
be-  
caus  
it  
is  
in-  
tend  
to  
be  
in-  
voke  
asyn  
chro  
as

## Paramet

- **task**  
a  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.
- **kwargs**  
the  
kwargs  
to  
be  
pass  
to

de-  
 ploy

**Raises**

In-  
 valid  
 State  
 if  
 the  
 even  
 is  
 not  
 al-  
 lowe  
 by  
 the  
 as-  
 so-  
 ci-  
 ated  
 state  
 ma-  
 chin

**Returns**

a  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing  
 the  
 fol-  
 low-  
 ing  
 keys

For  
 par-  
 ti-  
 tion  
 im-  
 age:

- root  
 uuid  
 UUI  
 of  
 root  
 par-

is uefi).

ti-  
tion  
  
•  
efi  
sys-  
tem  
par-  
ti-  
tion  
uuid  
UUID  
of  
the  
uefi  
sys-  
tem  
par-  
ti-  
tion  
(if  
boot  
mod  
  
\_\_\_\_\_  
**Note**  
If  
key  
ex-  
ists  
but  
valu  
is  
Non  
it  
mea  
par-  
ti-  
tion  
does  
ex-  
ist.  
  
For  
who  
disk  
im-  
age:  
  
•  
disk

identifier: ID of the disk to which image was deployed

ironic.

Delete the iSCSI target.

ironic.

All-in-one function to deploy a who disk image to a node

Paramet

- add
 The iSCSI IP

ad-  
dres

- **port**  
The  
iSCS  
port  
num  
ber.

- **iqn**  
The  
iSCS  
qual  
i-  
fied  
nam

- **lun**  
The  
iSCS  
log-  
i-  
cal  
unit  
num  
ber.

- **ima**  
Path  
for  
the  
in-  
stan  
disk  
im-  
age.

- **nod**  
node  
uuid

- **con**  
Op-  
tiona  
Base  
en-  
code

disk.

Gzip  
con-  
fig-  
drive  
con-  
tent  
or  
con-  
fig-  
drive  
HTT  
URI

- **con**  
Op-  
tiona  
Add  
a  
flag  
that  
will  
mod  
ify  
the  
be-  
havi  
of  
the  
im-  
age  
copy  
to

**Returns**  
a  
dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
key  
disk  
iden  
ti-  
fier  
to  
iden  
tify

was used for deployment.

the  
disk  
which

ironic

All-  
in-  
one  
func-  
tion  
to  
de-  
ploy  
a  
par-  
ti-  
tion  
im-  
age  
to  
a  
node

## Parameters

- **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.

- **roo**  
 Size  
 of  
 the  
 root  
 par-  
 ti-  
 tion  
 in  
 meg

- **swa**  
 Size  
 of



created.

the  
swap  
par-  
ti-  
tion  
in  
meg

- **eph**  
Size  
of  
the  
ephe  
par-  
ti-  
tion  
in  
meg  
If  
0,  
no  
ephe  
par-  
ti-  
tion  
will  
be

- **eph**  
The  
type  
of  
file  
sys-  
tem  
to  
for-  
mat  
the  
ephe  
par-  
ti-  
tion.

- **nod**  
node  
uuid  
Used  
for

ever content it had (if the partition table has not changed).

log-  
ging  
 •  
**pre**  
 If  
 True  
 no  
 files  
 tem  
 is  
 writ  
 ten  
 to  
 the  
 eph  
 bloc  
 de-  
 vice  
 pre-  
 serv  
 ing  
 wha  
 •  
**con**  
 Op-  
 tion:  
 Base  
 en-  
 code  
 Gzip  
 con-  
 fig-  
 drive  
 con-  
 tent  
 or  
 con-  
 fig-  
 drive  
 HTT  
 URI  
 •  
**boo**  
 Can  
 be  
 lo-  
 cal  
 or  
 net-

boot  
net-  
boot  
by  
de-  
fault

- **boot**  
Can  
be  
bios  
or  
uefi.  
bios  
by  
de-  
fault

- **disk**  
The  
disk  
la-  
bel  
to  
be  
used  
when  
cre-  
at-  
ing  
the  
par-  
ti-  
tion  
ta-  
ble.  
Valid

values are: msdos, gpt or None; If None ironiC will figure it out according to the boot\_mode parameter.

- **cpu**  
Ar-  
chi-  
tec-  
ture  
of  
the  
node  
be-  
ing  
de-

ploy
 to.

**Raises**

In-
 stan-
 dard
 ploy
 Fail-
 ure
 if
 im-
 age
 vir-
 tual
 size
 is
 big-
 ger
 than
 root
 par-
 ti-
 tion
 size.

**Returns**

a
 dic-
 tio-
 nary
 con-
 tain-
 ing
 the
 fol-
 low-
 ing
 keys
 root
 uuid
 UUID
 of
 root
 par-
 ti-

tion efi system partition uuid: UUID of the uefi system partition (if boot mode is uefi). NOTE: If key exists but value is None, it means partition doesnt exist.

ironic.

Do
 iSCSI
 dis-

cov-  
 ery  
 on  
 por-  
 tal.

ironic.

Met  
 in-  
 voke  
 whe  
 de-  
 ploy  
 with  
 the  
 ager  
 rame

This  
 meth  
 is  
 in-  
 voke  
 by  
 drive  
 for  
 do-  
 ing  
 iSCS  
 de-  
 ploy  
 us-  
 ing  
 ager  
 rame  
 This  
 meth  
 as-

sumes that the agent is booted up on the node and is heartbeating.

Paramet

- **task**
  
 a
   
 Task
   
 ager
   
 ob-
   
 ject
   
 con-
   
 tain-

nodes target disk via iSCSI, for install boot loader, etc).

Returns

a dictionary containing the following keys:
 For partitioning image:
 root uuid UUI of

is uefi).

root  
 par-  
 ti-  
 tion  
 •  
 efi  
 sys-  
 tem  
 par-  
 ti-  
 tion  
 uuid  
 UUID  
 of  
 the  
 uefi  
 sys-  
 tem  
 par-  
 ti-  
 tion  
 (if  
 boot  
 mod

Note  
 If  
 key  
 ex-  
 ists  
 but  
 valu  
 is  
 Non  
 it  
 mea  
 par-  
 ti-  
 tion  
 does  
 ex-  
 ist.

For  
 who  
 disk  
 im-  
 age:

• disk  
 iden-  
 ti-  
 fier:  
 ID  
 of  
 the  
 disk  
 to  
 whic  
 im-  
 age  
 was  
 de-  
 ploy

Raises

In-  
 stan-  
 ploy  
 Fail-  
 ure  
 if  
 it  
 en-  
 cour  
 ters  
 som  
 er-  
 ror  
 dur-  
 ing  
 the  
 de-  
 ploy

ironic  
 forc  
 iSCS  
 ini-  
 tia-  
 tor  
 to  
 re-  
 read  
 luns

ironic.

Re-



nary.

Paramet

- **node**  
ironi  
node  
ob-  
ject
- **add**  
iSCS  
ad-  
dres
- **iqn**  
iSCS  
iqn  
for  
the  
tar-  
get  
disk
- **port**  
iSCS  
port  
de-  
fault  
to  
one

spec  
i-  
fied  
in  
the  
con-  
fig-  
u-  
ra-  
tion

- **lun**  
iSCS  
lun,  
de-  
fault  
to  
1

- **con**  
flag  
that  
will  
mod  
ify  
the  
be-  
havi  
of  
the  
im-  
age  
copy  
to  
disk

**Raises**  
Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
re-  
quir  
pa-  
ram-  
e-

ters  
 were  
 not  
 pass

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 any  
 of  
 the  
 pa-  
 ram-  
 e-  
 ters  
 have  
 in-  
 valid  
 valu

ironic.

Lo-  
 gin  
 to  
 an  
 iSCS  
 tar-  
 get.

ironic.

Lo-  
 gout  
 from  
 an  
 iSCS  
 tar-  
 get.

ironic.

Val-  
 i-  
 date  
 the  
 pre-  
 requ  
 for

whether conductor url is available either from CONF file or from keystone.

iSCSI  
 de-  
 ploy  
 Val-  
 i-  
 date  
 whe  
 node  
 in  
 the  
 task  
 pro-  
 vide  
 has  
 som  
 port  
 en-  
 rolle  
 This  
 meth  
 val-  
 i-  
 date

Paramet  
 tas  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 URI

figured in config file and is not accessible via Keystone catalog.

of  
the  
Iron  
API  
ser-  
vice  
is  
not  
con-

#### Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
no  
port  
are  
en-  
rolle  
for  
the  
give  
node

ironic.  
Ver-  
ify  
iscsi  
con-  
nec-  
tion.

**ironic.drivers.modules.noop module**

Dun  
in-  
ter-  
face  
im-  
ple-  
men  
ta-  
tion  
for  
use

exceptions for user-accessible actions.

as  
de-  
fault  
with  
op-  
tion:  
in-  
ter-  
face  
Note  
that  
un-  
like  
fake  
im-  
ple-  
men-  
ta-  
tion:  
thes  
do  
not  
pass  
val-  
i-  
da-  
tion  
and  
raise

**class** i  
Base  
obj  
Mix  
to  
add  
to  
an  
in-  
ter-  
face  
to  
mak  
it  
fail  
val-  
i-  
da-  
tion.

**get\_pro**

**validat**

**class i**

Base  
*irc*  
*dri*  
*mod*  
*noc*  
*Fai*  
*irc*  
*dri*  
*bas*  
*BIO*

BIO  
 in-  
 ter-  
 face  
 im-  
 ple-  
 men-  
 ta-  
 tion  
 that  
 raise  
 er-  
 rors  
 on  
 all  
 re-  
 ques

**apply\_c**

Val-  
 i-  
 date  
 &  
 ap-  
 ply  
 BIO  
 set-  
 ting  
 on  
 the  
 give  
 node  
 This  
 meth

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.

Parameters

- **task**  
 a Task object representing the task instance.
- **settings**  
 Dictionary containing the BIOS configuration to be applied.

Raises  
 UnsupportedError



tion.

ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 node  
 drive  
 does  
 sup-  
 port  
 BIO  
 con-  
 fig-  
 u-  
 ra-

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 val-  
 i-  
 da-  
 tion  
 of  
 set-  
 tings  
 fails

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters

plete.

are  
 miss  
 ing.  
**Returns**  
 state  
 if  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if  
 it  
 is  
 com

**cache\_k**  
 Stor  
 or  
 up-  
 date  
 BIO  
 prop  
 er-  
 ties  
 on  
 the  
 give  
 node  
 This  
 meth  
 store  
 BIO  
 prop  
 er-  
 ties  
 to  
 the  
 bios  
 ta-  
 ble  
 dur-

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.

ties from bare metal.

ing  
clea  
ing  
op-  
er-  
a-  
tion

#### Parame

tas  
a  
Task  
ager  
in-  
stan

#### Raises

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the  
node  
drive  
does  
sup-  
port  
get-  
ting  
BIO  
prop  
er-

#### Returns

Non

#### factory

Re-  
set  
BIO  
con-  
fig-  
u-  
ra-  
tion  
to

ter the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

fac-  
 tory  
 de-  
 fault  
 on  
 the  
 give  
 node  
  
 This  
 meth  
 re-  
 sets  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 to  
 fac-  
 tory  
 de-  
 fault  
 on  
 the  
 give  
 node  
 Af-

**Parame**  
**tas**  
 a  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 the  
 node  
 driv  
 does

sup-  
port  
BIO  
re-  
set.

**Returns**

state  
if  
BIO  
con-  
fig-  
u-  
ra-  
tion  
is  
in  
prog  
asyn  
chro  
or  
Non  
if  
it  
is  
com

plete.

**class** i

Base  
*irc*  
*dri*  
*mod*  
*noc*  
*Fai*  
*irc*  
*dri*  
*bas*  
*Con*

Con  
sole  
in-  
ter-  
face  
im-  
ple-  
men  
ta-  
tion  
that  
raise  
er-

console.

rors  
 on  
 all  
 re-  
 ques  
  
**get\_con**  
 Get  
 con-  
 nec-  
 tion  
 in-  
 for-  
 ma-  
 tion  
 abou  
 the  
 con-  
 sole  
  
 This  
 meth  
 shou  
 re-  
 turn  
 the  
 nec-  
 es-  
 sary  
 in-  
 for-  
 ma-  
 tion  
 for  
 the  
 clien  
 to  
 ac-  
 cess  
 the  
  
**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing

the  
node  
to  
act  
on.

**Returns**

the  
con-  
sole  
con-  
nec-  
tion  
in-  
for-  
ma-  
tion.

**start\_c**

Start  
a  
re-  
mote  
con-  
sole  
for  
the  
task  
node

This  
meth  
shou  
not  
raise  
an  
ex-  
cep-  
tion  
if  
con-  
sole  
al-  
read  
start

**Parame**

**tas**  
A  
Task  
ager  
in-  
stan

con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**stop\_controller**  
 Stop  
 the  
 re-  
 mote  
 con-  
 sole  
 ses-  
 sion  
 for  
 the  
 task  
 node

**Parameter: task\_id**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**class ironic\_driver**  
 Base  
*ironic\_driver*  
*driver*  
*module*  
*node*  
*Failure*  
*ironic\_driver*  
*base*  
*Inspector*  
 In-  
 spec



in-  
 ter-  
 face  
 im-  
 ple-  
 men-  
 ta-  
 tion  
 that  
 raise  
 er-  
 rors  
 on  
 all  
 re-  
 ques

**inspect**

In-  
 spec  
 hard  
 ware

In-  
 spec  
 hard  
 ware  
 to  
 ob-  
 tain  
 the  
 es-  
 sen-  
 tial  
 &  
 ad-  
 di-  
 tion:  
 hard  
 ware  
 prop  
 er-  
 ties.

**Parame**

**tas**  
 A  
 task  
 from  
 Task  
 ager

**Raises**

Hardware specification failure, if unable to get essential hardware properties.

**Returns**

Resulting state of the inspection i.e. state or Non

```

class i
Base
    ird
    dri
    moc
    noc
    Fai
    ird
    dri
    bas
    RAI
    RAI
    in-
    ter-
    face
    im-

```

ple-  
men-  
ta-  
tion  
that  
raise  
er-  
rors  
on  
all  
re-  
ques

#### **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  
It  
as-  
sum  
that  
the

target RAID configuration is already available in `node.target_raid_config`. Implementations of this interface are supposed to read the RAID configuration from `node.target_raid_config`. After the RAID configuration is done (either in this method OR in a call-back method), `ironic.common.raid.update_raid_info()` may be called to sync the nodes RAID-related information with the RAID configuration applied on the

node.

ified in the nodes `target_raid_config`. Default value is `True`.

Paramete

- **task\_agent\_instances**  
 A Task Agent instance
- **create\_root\_volume**  
 Set-ting this to `False` in-dicate not to create root volume that is spec-
- **create\_non\_root\_volume**  
 Set-ting this to `False` in-dicate not to create non-root vol-ume

cept the root volume) in the nodes target\_raid\_config. Default value is True.

creating the new configuration.

chronously, or None if it is complete.

(all  
ex-

- **del**  
Set-  
ting  
this  
to  
True  
in-  
di-  
cate  
to  
dele  
RAI  
con-  
fig-  
u-  
ra-  
tion  
prior  
to

#### Returns

state  
(clea  
ing)  
or  
state  
(de-  
ploy  
men  
if  
RAI  
con-  
fig-  
u-  
ra-  
tion  
is  
in  
prog  
asyn

#### delete\_

Dele  
RAI  
con-  
fig-

ration is deleted, `node.raid_config` should be cleared by the implementation.

u-  
 ra-  
 tion  
 on  
 the  
 give  
 node  
  
 This  
 meth  
 dele  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give  
 node  
 Af-  
 ter  
 RAI  
 con-  
 fig-  
 u-  
  
**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
  
**Returns**  
 state  
 (clea  
 ing)  
 or  
 state  
 (de-  
 ploy  
 men  
 if  
 dele  
 tion  
 is  
 in  
 prog

it is complete.

asyn  
chro  
or  
Non  
if

**validat**  
Val-  
i-  
date  
the  
give  
RAI  
con-  
fig-  
u-  
ra-  
tion.

This  
meth  
val-  
i-  
date  
the  
give  
RAI  
con-  
fig-  
u-  
ra-  
tion.  
Driv  
im-  
ple-  
men  
ta-  
tions  
of

this interface can override this method to support custom parameters for 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-  
 ra-  
 tion  
 is  
 in-  
 valid

**class** i  
 Base  
*irc*  
*dri*  
*mod*  
*noc*  
*Fai*  
*irc*  
*dri*  
*bas*  
*Res*  
 Res-  
 cue  
 in-  
 ter-  
 face  
 im-  
 ple-  
 men



ta-  
 tion  
 that  
 raise  
 er-  
 rors  
 on  
 all  
 re-  
 ques

**rescue**  
 Boo  
 the  
 task  
 node  
 into  
 a  
 res-  
 cue  
 en-  
 vi-  
 ron-  
 men

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 In-  
 stan  
 cue-  
 Fail-  
 ure  
 if  
 node  
 val-  
 i-  
 da-

tion  
 or  
 res-  
 cue  
 op-  
 er-  
 a-  
 tion  
 fails

**Returns**

state  
 if  
 res-  
 cue  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 state  
 if  
 it  
 is  
 com  
 plete

**unrescu**

Tear  
 dow  
 the  
 res-  
 cue  
 en-  
 vi-  
 ron-  
 men  
 and  
 re-  
 turn  
 to  
 nor-  
 mal.

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-

ation fails.

tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
**Raises**  
 In-  
 stan-  
 Un-  
 res-  
 cue-  
 Fail-  
 ure  
 if  
 node  
 val-  
 i-  
 da-  
 tion  
 or  
 un-  
 res-  
 cue  
 op-  
 er-  
**Returns**  
 state  
 if  
 it  
 is  
 suc-  
 cess  
 ful  
**class** i  
 Base  
 irc  
 dri  
 dri  
 mod  
 noc  
 noc  
 Fai  
 irc  
 dri  
 bas  
 Ven  
 Ven-  
 dor

in-  
 ter-  
 face  
 im-  
 ple-  
 men-  
 ta-  
 tion  
 that  
 raise  
 er-  
 rors  
 on  
 all  
 re-  
 ques

**driver\_**

Val-  
 i-  
 date  
 drive  
 vena  
 pass  
 ac-  
 tions

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  
cer-  
tain  
pa-  
ram-  
e-  
ter.

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
pa-  
ram-  
e-  
ter  
does  
not  
matc

ironic.drivers.modules.noop\_mgmt module

No-  
op  
man-  
age-  
men-  
in-  
ter-  
face  
im-  
ple-  
men-  
ta-  
tion.

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Man*

No-  
op  
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

to first try PXE booting, then fall back to hard drives.

to  
 be  
 pre-  
 con-  
 fig-  
 ured

**get\_boot**  
 Get  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 for  
 a  
 node

Pro-  
 vide  
 the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 of  
 the  
 node  
 Be  
 aware  
 that  
 not  
 all  
 drive  
 sup-  
 port  
 this.

**Parameter**  
**task**  
 A  
 task  
 from  
 Task  
 ager

**Raises**  
 Miss-  
 ing-  
 Pa-

ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

Returns

A  
 dic-  
 tio-  
 nary  
 con-  
 tain-  
 ing:

boot\_c

Ahe  
 boot  
 de-  
 vice  
 one  
 of  
*irc*  
*com*  
*boo*  
 or  
 Non  
 if  
 it  
 is  
 un-  
 know

persist

Whe  
 the  
 boot  
 de-  
 vice  
 will  
 per-



unknown.

sist  
 to  
 all  
 fu-  
 ture  
 boot  
 or  
 not,  
 Non  
 if  
 it  
 is

**get\_proc**  
 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**

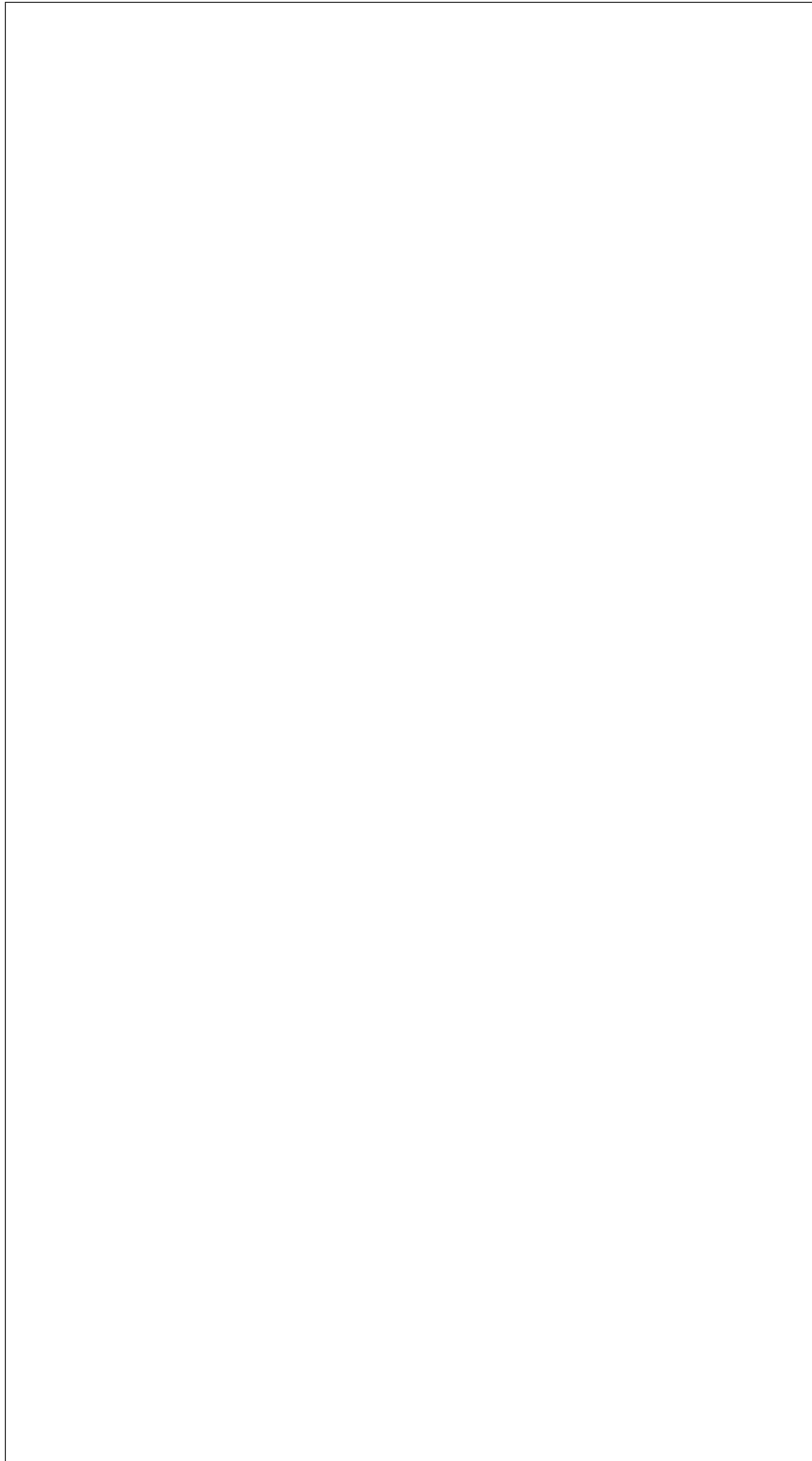
Fail  
 ToP  
 eSer  
 sor-  
 Data  
 whe  
 pars  
 ing  
 sen-  
 sor  
 data  
 fails

**Returns**

Re-  
 turn  
 a  
 con-  
 sis-  
 tent  
 for-  
 mat  
 dict  
 of  
 sen-  
 sor  
 data  
 grou  
 by  
 sen-  
 sor

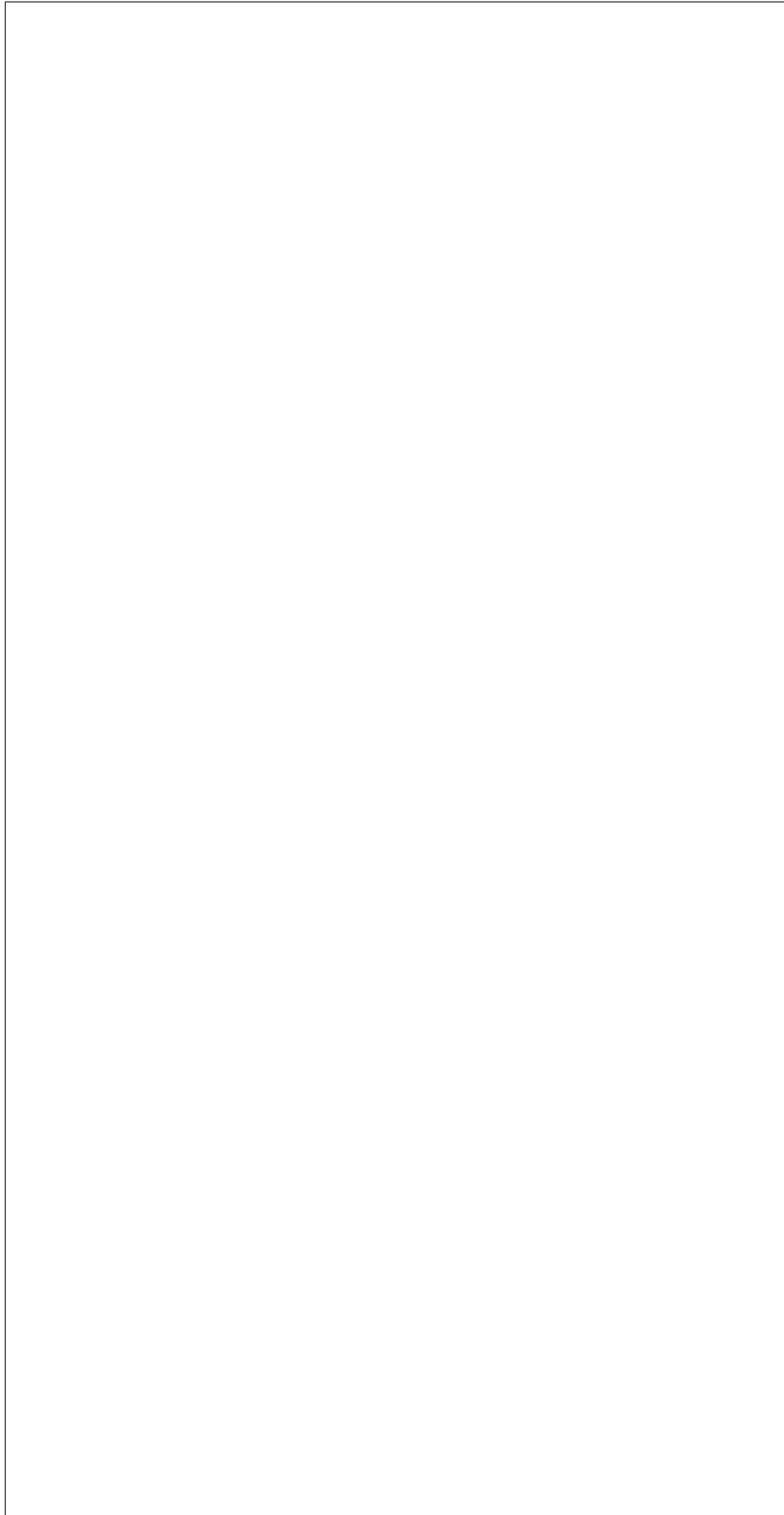
type  
which  
can

be processed by Ceilometer. eg,



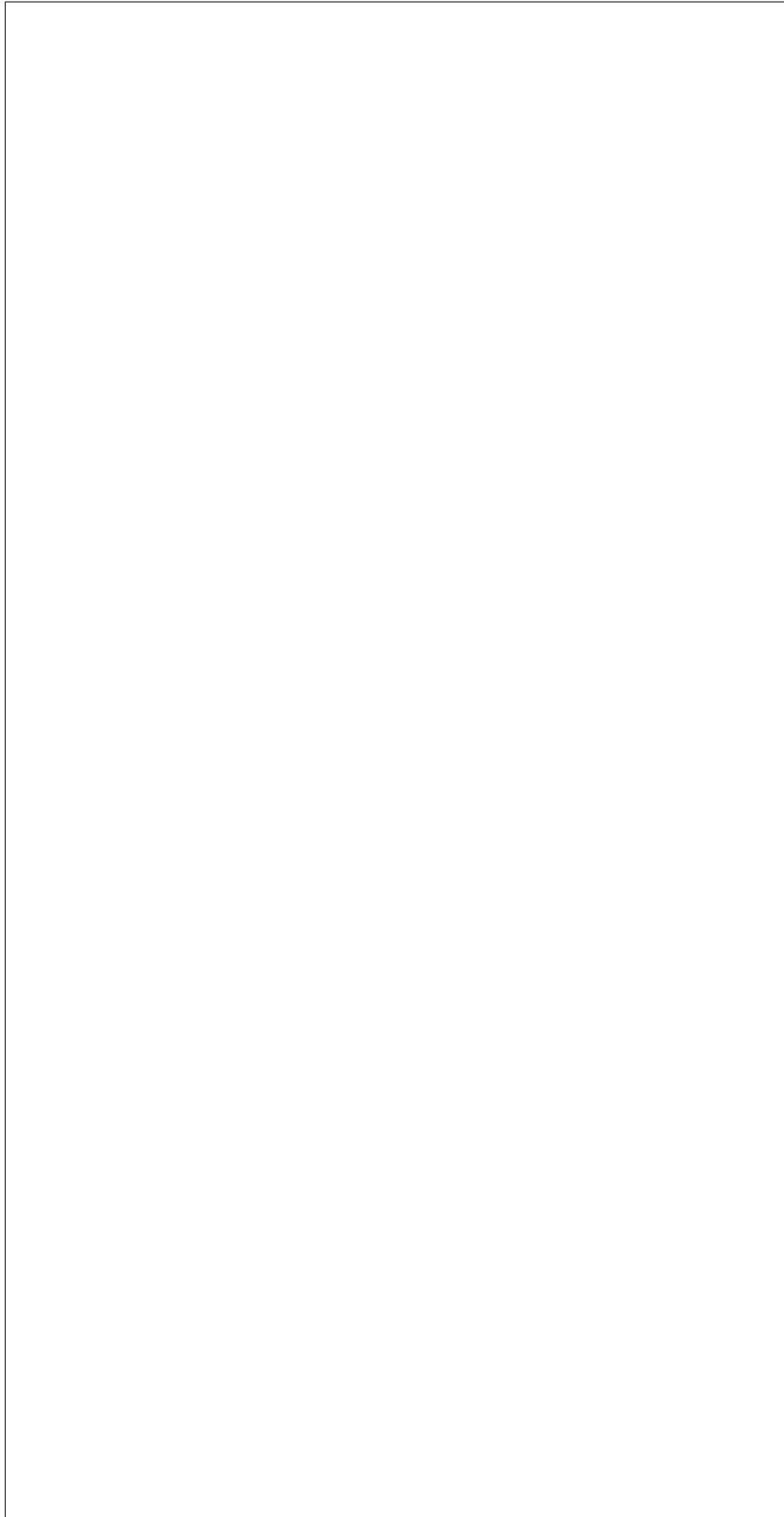
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



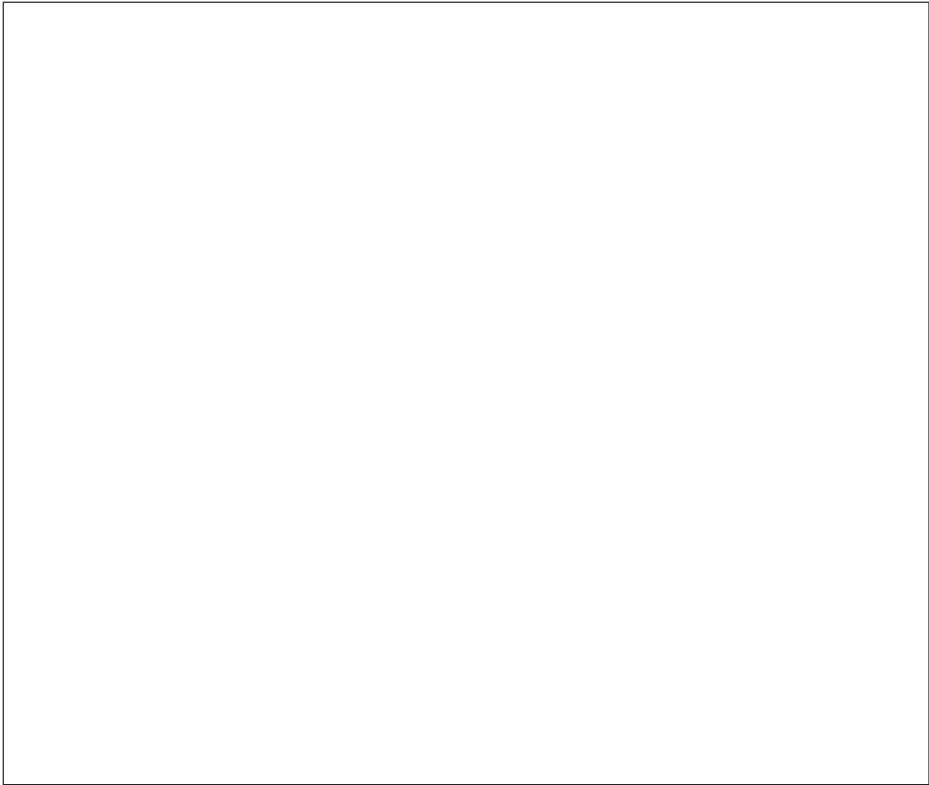
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



**get\_supported\_devices**  
 Get a list of the supported boot devices.

**Parameter: task**  
 A task from Task Manager.

**Returns:**  
 A list with the supported boot devices.

vice  
 de-  
 finec  
 in  
[irc](#)  
[com](#)  
[boo](#)

**set\_boo**  
 Set  
 the  
 boot  
 de-  
 vice  
 for  
 a  
 node  
  
 Set  
 the  
 boot  
 de-  
 vice  
 to  
 use  
 on  
 next  
 re-  
 boot  
 of  
 the  
 node  
  
**Parame**

- tas**  
 A  
 task  
 from  
 Task  
 ager
- dev**  
 The  
 boot  
 de-  
 vice  
 one  
 of  
[irc](#)  
[com](#)



boo

- **per**  
 Boo  
 valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 Fals  
 if

not. Default: False.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 boot  
 de-  
 vice  
 is  
 spec  
 i-  
 fied.

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if

the required information for this interface to function.

a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

**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  
task  
node  
con-  
tains

This  
meth  
is  
of-  
ten  
ex-  
e-

long-running checks.

cute  
 syn-  
 chro  
 in  
 API  
 re-  
 ques  
 so  
 it  
 shou  
 not  
 con-  
 duct

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 mal-  
 form  
 pa-  
 ram-  
 e-  
 ter(s)

**Raises**  
 Mis  
 ing-  
 Pa-  
 ram-  
 e-

ironic.drivers.modules.pxe module

ter-  
 Valu  
 on  
 miss  
 ing  
 pa-  
 ram-  
 e-  
 ter(s)

PXE  
 Boo  
 In-  
 ter-  
 face

**class** i  
 Base  
*irc*  
*dri*  
*mod*  
*pxe*  
*PXE*  
*irc*  
*dri*  
*bas*  
*Boo*

**capabil**

**class** i  
 Base  
*irc*  
*dri*  
*mod*  
*age*  
*Age*  
*irc*  
*dri*  
*mod*  
*age*  
*Hea*  
*irc*  
*dri*  
*bas*  
*Dep*

method will be called after `prepare()`, which may have already performed any preparatory steps, such as pre-caching some data for the node.

**deploy**  
 Per-  
 form  
 a  
 de-  
 ploy  
 men  
 to  
 the  
 task  
 node  
  
 Per-  
 form  
 the  
 nec-  
 es-  
 sary  
 worl  
 to  
 de-  
 ploy  
 an  
 im-  
 age  
 onto  
 the  
 spec  
 i-  
 fied  
 node  
 This

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**  
 sta-

tus  
 of  
 the  
 de-  
 ploy  
 One  
 of  
 iron

**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  
 de-  
 ploy  
 men  
 en-  
 vi-  
 ron-  
 men  
 for  
 the  
 task  
 node  
 If

this method should be implemented by the driver.

the same node on the same conductor.

prep  
ra-  
tion  
of  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
ahea  
of  
time  
is  
pos-  
si-  
ble,

If  
im-  
ple-  
men  
this  
meth  
mus  
be  
iden  
po-  
tent.  
It  
may  
be  
calle  
mul-  
ti-  
ple  
time  
for

This  
meth  
is  
calle  
be-  
fore  
*de-  
ploy*

**Parame**

This  
meth  
is

the required information for this interface to function.



long-running checks.

of-  
 ten  
 ex-  
 e-  
 cute  
 syn-  
 chro  
 in  
 API  
 re-  
 ques  
 so  
 it  
 shou  
 not  
 con-  
 duct

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 mal-  
 form  
 pa-  
 ram-  
 e-  
 ter(s)

**Raises**  
 Mis

ironic.drivers.modules.pxe\_base module

ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 on  
 miss  
 ing  
 pa-  
 ram-  
 e-  
 ter(s)

Base  
 PXE  
 In-  
 ter-  
 face  
 Met  
 ods

**class** i  
 Base  
 obj

**clean\_u**  
 Clea  
 up  
 the  
 boot  
 of  
 in-  
 stan  
  
 This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that  
 was  
 setu  
 for

links the instance kernel/ramdisk in nodes directory in tftproot and removes the PXE config.

rescue ramdisk. It unlinks the deploy/rescue kernel/ramdisk in the nodes directory in tftproot and removes its PXE config.

boot  
ing  
the  
in-  
stan  
It  
un-

**Parame**  
**tas**  
a  
task  
from  
Task  
ager

**Returns**  
Non

**clean\_u**  
Clea  
up  
the  
boot  
of  
iron  
rame  
  
This  
meth  
clea  
up  
the  
PXE  
en-  
vi-  
ron-  
men  
that  
was  
setu  
for  
boot  
ing  
the  
de-  
ploy  
or

**Parame**

ried out on the node. Supported values are deploy and rescue. Defaults to deploy, indicating deploy operation was carried out.

#### 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
	<b>ipxe_en</b>
	<b>prepare</b>
	Pre- pare the boot of in- stan
	This meth pre- pare the boot of the in- stan af- ter read ing rel- e- vant in- for- ma-
tion from the nodes 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

evant information from the nodes `driver_info` and `instance_info`.

**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/  
 af-  
 ter  
 read  
 ing  
 rel-

**Parame**  
  
 •  
**tas**  
 a  
 task  
 from  
 Task  
 ager  
  
 •  
**ram**  
 the  
 pa-  
 ram-  
 e-  
 ters  
 to

ters as kernel command-line arguments.

or instance\_info.

be  
 pass  
 to  
 the  
 ram  
 pxe  
 drive  
 pass  
 thes  
 pa-  
 ram-  
 e-

**Returns**  
 Non

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion  
 is  
 miss  
 ing  
 in  
 node  
 drive

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 in-  
 for-  
 ma-  
 tion

the node.

pro-  
 vide  
 is  
 in-  
 valid  
**Raises**  
 Iron  
 icEx  
 cep-  
 tion,  
 if  
 som  
 pow  
 or  
 set  
 boot  
 boot  
 de-  
 vice  
 op-  
 er-  
 a-  
 tion  
 faile  
 on

**validat**  
 Val-  
 i-  
 date  
 the  
 PXE  
 spec  
 info  
 for  
 boot  
 ing  
 de-  
 ploy  
 im-  
 ages  
 This  
 meth  
 val-  
 i-  
 date  
 the  
 PXE  
 spec  
 info



If invalid, raises an exception; otherwise returns None.

for  
boot  
ing  
the  
ram  
and  
in-  
stan  
on  
the  
node

**Parameters**  
**task**  
 a  
 task  
 from  
 Task  
 ager

**Returns**  
 None

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 pa-  
 ram-  
 e-  
 ters  
 are  
 in-  
 valid

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 re-

quire  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

**validat**

Val-  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quire  
 prop  
 er-  
 ties  
 for  
 in-  
 spec  
 tion.

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan-  
 with  
 the  
 node  
 be-  
 ing  
 chec

**Raises**

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

**validat**

Val-  
 i-

date  
 that  
 the  
 node  
 has  
 re-  
 quir  
 prop  
 er-  
 ties  
 for  
 res-  
 cue.

Paramet

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  
 re-  
 quir  
 pa-  
 ram-

eters

## ironic.drivers.modules.snmp module

driver model to support devices with different SNMP object models.

**class** i

Iron  
SNM  
pow  
man  
ager

Pro-  
vide  
ba-  
sic  
pow  
con-  
trol  
us-  
ing  
an  
SNM  
enab  
sma  
pow  
con-  
troll  
Uses  
a  
plug  
gabl

Base  
obj  
SNM  
clien  
ob-  
ject.  
Per-  
form  
low  
leve  
SNM

tion with PySNMP to simplify dynamic importing and unit testing.

get  
and  
set  
op-  
er-  
a-  
tions  
En-  
cap-  
su-  
lates  
all  
in-  
ter-  
ac-

**get** (*oid*  
Use  
PyS  
NM  
to  
per-  
form  
an  
SNM  
GET  
op-  
er-  
a-  
tion  
on  
a  
sin-  
gle  
ob-  
ject.

**Parame**  
**oid**  
The  
OID  
of  
the  
ob-  
ject  
to  
get.

**Raises**  
SN-  
MP-  
Fail-

ject.

ure  
 if  
 an  
 SNM  
 re-  
 ques  
 fails

**Returns**  
 The  
 valu  
 of  
 the  
 re-  
 ques  
 ob-  
 ject.

**get\_next**  
 Use  
 PyS  
 NM  
 to  
 per-  
 form  
 an  
 SNM  
 GET  
 NEX  
 op-  
 er-  
 a-  
 tion  
 on  
 a  
 ta-  
 ble  
 ob-

**Parameter**  
**oid**  
 The  
 OID  
 of  
 the  
 ob-  
 ject  
 to  
 get.

**Raises**  
 SN-

MP-  
 Fail-  
 ure  
 if  
 an  
 SNM  
 re-  
 ques  
 fails

Returns

A  
 list  
 of  
 val-  
 ues  
 of  
 the  
 re-  
 ques  
 ta-  
 ble  
 ob-  
 ject.

set (oid,

Use  
 PyS  
 NM  
 to  
 per-  
 form  
 an  
 SNM  
 SET  
 op-  
 er-  
 a-  
 tion  
 on  
 a  
 sin-  
 gle  
 ob-  
 ject.

Parame

- oid
   
 The  
 OID

of  
the  
ob-  
ject  
to  
set.

- **val**  
The  
valu  
of  
the  
ob-  
ject  
to  
set.

**Raises**  
SN-  
MP-  
Fail-  
ure  
if  
an  
SNM  
re-  
ques  
fails

**class** i

Base  
*irc*  
*dri*  
*moo*  
*snm*  
*SNM*

SNM  
drive  
class  
for  
APC  
Mas  
ter-  
Swit  
PDU  
de-  
vice

SNM  
ob-  
jects



ues: 1=On, 2=Off, 3=PowerCycle, [more options follow]

for  
APC  
SN-  
M-  
P-  
Driv  
APC  
Mas  
ter-  
Swit  
PDU  
1.3.6  
sP-  
DU-  
Out-  
letC  
Val-

**oid\_dev**

**system**

**value\_p**

**value\_p**

**class i**

Base  
*irc*  
*dri*  
*moo*  
*snm*  
*SNM*  
SNM  
drive  
class  
for  
APC  
Mas  
ter-  
Swit  
Plus  
PDU  
de-  
vice  
SNM  
ob-

ControlMSPOutletCommand Values: 1=On, 3=Off, [more options follow]

jects  
 for  
 APC  
 SN-  
 M-  
 P-  
 Driv  
 APC  
 Mas  
 ter-  
 Swit  
 Plus  
 PDU  
 1.3.0  
 sP-  
 DU-  
 Out-  
 let-

oid\_dev

system\_

value\_p

value\_p

class i

Base  
 ird  
 dri  
 mod  
 snm  
 SNM  
 SNM  
 drive  
 class  
 for  
 APC  
 Rack  
 PDU  
 de-  
 vice  
 SNM  
 ob-  
 jects  
 for

letCommand Values: 1=On, 2=Off, 3=PowerCycle, [more options follow]

APC  
SN-  
M-  
P-  
Drive  
APC  
PDU  
PDU  
#  
1.3.6  
rP-  
DU-  
Out-  
let-  
Con  
trolC

**oid\_dev**

**system\_**

**value\_p**

**value\_p**

**class i**

Base  
*irc*  
*dri*  
*mod*  
*snm*  
*SNM*

SNM  
drive  
class  
for  
Ater  
PDU  
de-  
vice

SNM  
ob-  
jects  
for  
Ater  
PDU  
1.3.6  
Out-

let  
 Pow  
 Val-  
 ues:  
 1=O  
 2=O  
 3=P  
 ing,  
 4=R  
 set

oid\_dev

system\_

value\_p

value\_p

class i  
 Base  
 irc  
 dri  
 moc  
 snm  
 SNM

SYS\_OBJ

class i  
 Base  
 obj  
 SNM  
 pow  
 drive  
 base  
 class  
 The  
 SN-  
 M-  
 P-  
 Driv  
 class  
 hi-  
 er-  
 ar-  
 chy

to interface with different smart power controller products.

im-  
ple-  
men  
man  
spec  
MIE  
ac-  
tions  
over  
SNM

**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  
 MR  
 PDU  
 de-  
 vice  
 SNM

0=Off, 1=On, 2=Reboot

ob-  
 jects  
 for  
 Bay  
 MRI  
 PDU  
 4779  
 1,  
 3,  
 5,  
 3,  
 1,  
 3,  
 {uni  
 Out-  
 let  
 Pow  
 Val-  
 ues:

**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-



1=On, 2=Off, 3=PowerCycle, [more options follow]

jects  
for  
Cy-  
ber-  
Pow  
PDU  
1.3.6  
eP-  
DU-  
Out-  
let-  
Con  
trolC  
let-  
Com  
man  
Val-  
ues:

**oid\_dev**

**system\_**

**value\_p**

**value\_p**

**class i**

Base  
*irc*  
*dri*  
*mod*  
*snm*  
*SNM*

SNM  
drive  
class  
for  
Eato  
Pow  
PDU

The  
Eato  
pow  
PDU  
does  
not

multiple SNMP objects.

ing off, 3=pending on 1.3.6.1.4.1.534.6.6.7.6.6.1.3.<outlet ID> outletControlOffCmd Write 0 for immediate power off 1.3.6.1.4.1.534.6.6.7.6.6.1.4.<outlet ID> outletControlOnCmd Write 0 for immediate power on

**status\_**

**status\_**

**status\_**

**system\_**

**value\_p**

**value\_p**

**class i**

Base

*irc*

*dri*

*mo*

*snm*

*SNM*

SNM

drive

base

class

for

sim-

ple

PDU

de-

vice

Here

sim-

ple

refer

to

de-

vice

whic

pro-

vide

a

sin-

gle

SNM

ob-

ject

for

con-

the power state of an outlet.

A different OID may be specified by overriding the `_snmp_oid` method in a subclass.

trol-  
ling

The  
de-  
fault  
OID  
of  
the  
pow  
state  
ob-  
ject  
is  
of  
the  
form  
<en-  
ter-  
prise  
OID  
OID  
ID>

**abstract**  
De-  
vice  
de-  
pen-  
dent  
por-  
tion  
of  
the  
pow  
state  
ob-  
ject  
OID

**abstract**  
Valu  
rep-  
re-  
sent  
ing  
pow  
off  
state

**abstract**

Valu
 rep-
 re-
 sent
 ing
 pow
 on
 state

**class** i

Base
 *irc*
*dri*
*mod*
*snm*
*SNM*

SNM
 drive
 class
 for
 Tel-
 tron
 PDU
 de-
 vice

SNM
 ob-
 jects
 for
 Tel-
 tron
 PDU
 1.3.6
 Out-
 let
 Pow
 Val-
 ues:
 1=O
 2=O

**oid\_dev**

**system\_**

**value\_p**

**value\_p**

of a physical device using an SNMP-enabled smart power controller.

```
class i
    Base
    irc
    dri
    bas
    Pow

    SNM
    Pow
    In-
    ter-
    face

    This
    Pow
    er-
    In-
    ter-
    face
    class
    pro-
    vide
    a
    mec
    a-
    nism
    for
    con-
    trol-
    ling
    the
    pow
    state

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-  
 e-  
 ters  
 are  
 miss  
 ing.

**Raises**  
 In-  
 valic  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 SNM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 in-  
 valic

**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

Paramete

- **task**  
 An  
 in-  
 stan-  
 of  
*iron*
- **time**  
 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-  
 e-  
 ters  
 are  
 miss  
 ing.

Raises

In-  
 valio  
 Pa-

the timeout.

ram-  
 e-  
 ter-  
 Valu  
 if  
 SNM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 in-  
 valid  
**Raises**  
 Pow  
 er-  
 State  
 Fail-  
 ure  
 if  
 the  
 fi-  
 nal  
 pow  
 state  
 of  
 the  
 node  
 is  
 not  
 POW  
 af-  
 ter  
**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

Paramete

- **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-  
 face

Raises  
 Miss  
 ing-  
 Pa-  
 ram-

invalid.

e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 SNM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 SNM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 in-  
 valid  
 or  
*psta*  
 is

Raises

Pow  
 er-  
 State  
 Fail-  
 ure  
 if  
 the  
 fi-  
 nal  
 pow  
 state  
 of  
 the  
 node

after the timeout.

is  
 not  
 as  
 re-  
 ques  
  
**Raises**  
 SN-  
 MP-  
 Fail-  
 ure  
 if  
 an  
 SNM  
 re-  
 ques  
 fails  
  
**validat**  
 Che  
 that  
 node  
 con-  
 tains  
 the  
 req-  
 ui-  
 site  
 field  
  
**Raises**  
 Mis:  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 re-  
 quir  
 SNM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.  
  
**Raises**  
 In-

valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 SNM  
 pa-  
 ram-  
 e-  
 ters  
 are  
 in-  
 valid  
 ironic.  
 ironic.

Module contents

Submodules

ironic.drivers.base module

Ab-  
 strac  
 base  
 class  
 for  
 drive  
 ironic.  
 Con  
 stan  
 hold  
 ing  
 all  
 know  
 in-  
 ter-  
 face  
 class i  
 Base  
*irc*  
*dri*  
*bas*

## abstract

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  
BIO  
set-  
ting  
on  
the

given node. It may also validate the given bios settings before applying any settings and manage failures when setting an invalid BIOS config. In the case of needing password to update the BIOS config, it will be taken from the driver\_info properties. After the BIOS configuration is done, cache\_bios\_settings will be called to update the nodes BIOS setting table with the BIOS configuration applied on the node.

## Paramete

- **task**  
a  
Task  
ager  
in-  
stan

tion.

• **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-  
 port  
 BIO  
 con-  
 fig-  
 u-  
 ra-

**Raises**  
 In-  
 valio  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 val-  
 i-  
 da-  
 tion  
 of  
 set-



ting  
 fails

**Raises**

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

**Returns**

state  
 if  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if  
 it  
 is  
 com

plete.

**abstract**

Stor  
 or  
 up-  
 date  
 BIO  
 prop  
 er-

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.

**Parameters**  
 task: a TaskManager instance

**Raises**  
 UnsupportedDriverExtensionException if the node's driver does not support getting

ties from bare metal.

ter the BIOS reset action is done, `cache_bios_settings` will be called to update the nodes BIOS settings table with default bios settings.

ting  
 BIO  
 prop  
 er-  
  
**Returns**  
 Non  
  
**abstract**  
 Re-  
 set  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 to  
 fac-  
 tory  
 de-  
 fault  
 on  
 the  
 give  
 node  
  
 This  
 meth  
 re-  
 sets  
 BIO  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 to  
 fac-  
 tory  
 de-  
 fault  
 on  
 the  
 give  
 node  
 Af-  
  
**Paramete**  
**tas**

plete.

a  
Task  
ager  
in-  
stan

**Raises**  
Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the  
node  
drive  
does  
sup-  
port  
BIO  
re-  
set.

**Returns**  
state  
if  
BIO  
con-  
fig-  
u-  
ra-  
tion  
is  
in  
prog  
asyn  
chro  
or  
Non  
if  
it  
is  
com

**interfa**  
In-  
ter-  
face  
type  
used

for  
clear  
step  
and  
log-  
ging

```
class i
Base
obj
A
bare
drive
ob-
ject
whic
will
have
in-
ter-
face
at-
tach
later
Any
com
pos-
able
in-
ter-
face
shou
be
adde
as
class
at-
tribu
of
this
class
as
well
as
```

appended to core\_interfaces or standard\_interfaces here.

**property**

```
bios =
Stan
dara
```

at-  
 tribu  
 for  
 BIO  
 re-  
 latec  
 fea-  
 tures  
  
 A  
 ref-  
 er-  
 ence  
 to  
 an  
 in-  
 stan  
 of  
 :clas

**boot =**  
*Stan*  
*dara*  
 at-  
 tribu  
 for  
 boot  
 re-  
 latec  
 fea-  
 tures  
  
 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

**managem**  
*Stan*  
*dard*  
 at-  
 tribu  
 for



man  
 age-  
 men  
 re-  
 latec  
 fea  
 tures.  
  
 A  
 ref-  
 er-  
 ence  
 to  
 an  
 in-  
 stan  
 of  
 :clas

**network**

*Con*  
 at-  
 tribu  
 for  
 net-  
 worl  
 con-  
 nec-  
 tiv-  
 ity.  
  
 A  
 ref-  
 er-  
 ence  
 to  
 an  
 in-  
 stan  
 of  
 :clas

**propert**

**propert**

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*  
*dara*  
 at-  
 tribu  
 for  
 RAI  
 re-  
 latec  
 fea-  
 tures  
  
 A  
 ref-  
 er-  
 ence  
 to  
 an  
 in-  
 stan  
 of  
 :clas

**rescue**  
*Stan*  
*dara*  
 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  
 Driv  
 In-  
 ter-  
 face  
  
**execute**  
 Ex-  
 e-  
 cute  
 the  
 clea  
 step  
 on  
 task  
  
 A  
 clea  
 step  
 mus  
 take  
 a

may take one or more keyword variable arguments (for use with manual cleaning only.)

method has completed synchronously or states.CLEANWAIT if the step will continue to execute asynchronously. If the step executes asynchronously, it should issue a call to the `continue_node_clean` RPC, so the conductor can begin the next clean step.

single  
position  
arguments  
a  
Task  
argument  
object.  
It

A  
step  
can  
be  
executed  
synchronously  
or  
asynchronously  
A  
step  
should  
return  
Non  
if  
the

#### Parameters

- **task**  
A  
Task  
argument  
object
- **step**

ecute asynchronously.

The  
 clear  
 step  
 dic-  
 tio-  
 nary  
 rep-  
 re-  
 sent  
 ing  
 the  
 step  
 to  
 ex-  
 e-  
 cute

Returns

Non  
 if  
 this  
 meth  
 has  
 com  
 plete  
 syn-  
 chro  
 or  
 state  
 if  
 the  
 step  
 will  
 con-  
 tinue  
 to  
 ex-

execute

Ex-  
 e-  
 cute  
 the  
 de-  
 ploy  
 step  
 on  
 task  
 A  
 de-  
 ploy

It may take one or more keyword variable arguments (for use in the future, when deploy steps can be specified via the API).

method has completed synchronously or states.DEPLOYWAIT if the step will continue to execute asynchronously. If the step executes asynchronously, it should issue a call to the `continue_node_deploy` RPC, so the conductor can begin the next deploy step.

step  
mus  
take  
a  
sin-  
gle  
po-  
si-  
tiona  
ar-  
gu-  
men  
a  
Task  
ager  
ob-  
ject.

A  
step  
can  
be  
ex-  
e-  
cute  
syn-  
chro  
or  
asyn  
chro  
A  
step  
shou  
re-  
turn  
Non  
if  
the

## Parame

- **tas**  
A  
Task  
ager  
ob-

ecute asynchronously.

ject
 

- steps**

The deployment step dictionary represents the step to execute.

**Returns**

None if this method has completed synchronously or state if the step will continue to execute.

**get\_clean**

Get a list of (enabled and disabled



in an unordered list.

able  
 clear  
 step  
 for  
 the  
 in-  
 ter-  
 face  
  
 This  
 func  
 tion  
 will  
 re-  
 turn  
 all  
 clear  
 step  
 (bot  
 en-  
 able  
 and  
 dis-  
 able  
 for  
 the  
 in-  
 ter-  
 face

**Parame**  
**tas**  
 A  
 Task  
 ager  
 ob-  
 ject,  
 use-  
 ful  
 for  
 in-  
 ter-  
 face  
 over  
 rid-  
 ing  
 this  
 func  
 tion

**Raises**  
*No*

a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available clean steps.

Returns

A list of clean step dictionaries narries

get\_deploy

Get a list of (en-able and dis-able deploy step for the in-ter-face This func

face, in an unordered list.

tion  
 will  
 re-  
 turn  
 all  
 de-  
 ploy  
 step  
 (bot  
 en-  
 able  
 and  
 dis-  
 able  
 for  
 the  
 in-  
 ter-

**Parame**  
**tas**  
 A  
 Task  
 ager  
 ob-  
 ject,  
 use-  
 ful  
 for  
 in-  
 ter-  
 face  
 over  
 rid-  
 ing  
 this  
 func  
 tion

**Raises**  
*Ins*  
 if  
 there  
 is  
 a  
 prob  
 lem  
 get-  
 ting  
 the  
 step

a node (using an agent driver) has just been enrolled and the agent isnt alive yet to be queried for the available deploy steps.

Returns

A list of deploy step dictionaries.

abstract

Return the properties of the interface.

Returns

dictionary of <property name>: description entries.

interface

Interface.

face  
 type  
 used  
 for  
 clear  
 step  
 and  
 log-  
 ging

**support**

In-  
 di-  
 cate  
 if  
 an  
 in-  
 ter-  
 face  
 is  
 sup-  
 port

This  
 will  
 be  
 set  
 to  
 Fals  
 for  
 in-  
 ter-  
 face  
 whic  
 are  
 unte  
 in  
 first-

or  
 thir  
 part  
 CI,

or in the process of being deprecated.

**abstract**

Val-  
 i-  
 date  
 the  
 drive  
 spec  
 Nod

the required information for this interface to function.

long-running checks.

Parameters  
 task  
 A  
 Task

ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

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)

class i

Base  
*irc*  
*dri*  
*bas*  
*Bas*  
 In-  
 ter-

face  
 for  
 boot  
 relat  
 ac-  
 tions

**capabil**

**abstrac**

Clea  
 up  
 the  
 boot  
 of  
 in-  
 stan

This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that  
 was  
 setu  
 for  
 boot  
 ing  
 the  
 in-  
 stan

**Parame**

**tas**  
 A  
 task  
 from  
 Task  
 ager

**Returns**

Non

**abstrac**

Clea  
 up  
 the



cue ramdisk.

boot  
 of  
 iron  
 ram  
  
 This  
 meth  
 clea  
 up  
 the  
 en-  
 vi-  
 ron-  
 men  
 that  
 was  
 setu  
 for  
 boot  
 ing  
 the  
 de-  
 ploy  
 or  
 res-

**Parame**  
**tas**  
 A  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**interfa**  
 In-  
 ter-  
 face  
 type  
 used  
 for  
 clea  
 step  
 and  
 log-  
 ging

**abstrac**  
 Pre-  
 pare

tion from the nodes database.

the  
boot  
of  
in-  
stan  
  
This  
meth  
pre-  
pare  
the  
boot  
of  
the  
in-  
stan  
af-  
ter  
read  
ing  
rel-  
e-  
vant  
in-  
for-  
ma-

**Parameters**  
**task**  
 A  
 task  
 from  
 Task  
 ager

**Returns**  
 Non

**abstract**  
 Pre-  
 pare  
 the  
 boot  
 of  
 Iron  
 rame  
  
 This  
 meth  
 pre-  
 pare  
 the  
 boot

vant information from the nodes database.

might want to boot the ramdisk in different ways by passing parameters to them. For example,

of  
the  
de-  
ploy  
or  
res  
cue  
ram  
af-  
ter  
read  
ing  
rel-  
e-

## Parameters

- **task**  
A  
task  
from  
Task  
ager
- **ram**  
The  
op-  
tions  
to  
be  
pass  
to  
the  
iron  
ram  
Dif-  
fer-  
ent  
im-  
ple-  
men  
ta-  
tions

Whe  
Age  
ram  
is

etc.

ent implementations of boot interface will have different ways of passing parameters to the ramdisk.

Returns  
 Non  
**validat**  
 Val-  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quir

prop
 er-
 ties
 for
 in-
 spec
 tion.

**Parameters**  
**task**  
 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
 mor
 re-
 quir
 pa-
 ram-

**Raises**  
 Un-
 sup-
 port
 ed-
 Driv
 ten-
 sion

eters

**validate**  
 Val-  
 i-  
 date  
 that  
 the  
 node  
 has  
 re-  
 quir  
 prop  
 er-  
 ties  
 for  
 res-  
 cue.

**Parameter**  
**task**  
 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  
 mor  
 re-  
 quir  
 pa-  
 ram-

eters

**Raises**

Un-  
sup-  
port-  
ed-  
Driv-  
ten-  
sion

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Bas*

In-  
ter-  
face  
for  
cons  
relat  
ac-  
tion

**abstrac**

Get  
con-  
nec-  
tion  
in-  
for-  
ma-  
tion  
about  
the  
con-  
sole

This  
meth  
shou  
re-  
turn  
the  
nec-  
es-  
sary  
in-  
for-  
ma-  
tion

console.

for  
the  
clien  
to  
ac-  
cess  
the

**Parameters**  
**task**  
A  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

**Returns**  
the  
con-  
sole  
con-  
nec-  
tion  
in-  
for-  
ma-  
tion.

**interface**  
In-  
ter-  
face  
type  
used  
for  
clea  
step  
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  
 start

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**abstrac**  
 Stop  
 the  
 re-  
 mote  
 con-  
 sole  
 ses-  
 sion  
 for  
 the  
 task

node

**Parameter**

**task**

A

Task

ager

in-

stan-

con-

tain-

ing

the

node

to

act

on.

**class** *ionic*

Base

*ionic*

*drive*

*base*

*Base*

In-

ter-

face

for

depl

relat

ac-

tions

**abstract**

Clea

up

the

de-

ploy

men

en-

vi-

ron-

men

for

the

task

node

If

prep

ra-

this method should be implemented by the driver. It should erase anything cached by the *prepare* method.

the same node on the same conductor, and it may be called by multiple conductors in parallel. Therefore, it must not require an exclusive lock.

tion  
of  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
ahea  
of  
time  
is  
pos-  
si-  
ble,

If  
im-  
ple-  
men  
this  
meth  
mus  
be  
iden  
po-  
tent.  
It  
may  
be  
calle  
mul-  
ti-  
ple  
time  
for

This  
meth  
is  
calle  
be-  
fore  
*tear*.

**Parame**  
**tas**

A Taskager instance containing the node to act on.

**abstract**

Perform a deployment to the task node.

Perform the necessary work to deploy an image onto 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.

**Parameters**

**task** A Task

ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

Returns

sta-  
 tus  
 of  
 the  
 de-  
 ploy  
 One  
 of  
 iron

heartbeat

Rece  
 a  
 hear  
 beat  
 for  
 the  
 node

Parameters

- **task**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 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

**Returns**  
 Non

**interfa**  
 In-  
 ter-  
 face  
 type  
 used  
 for  
 clea  
 step  
 and  
 log-  
 ging

**abstrac**  
 Pre-  
 pare  
 the  
 de-

this method should be implemented by the driver.

ploy  
men  
en-  
vi-  
ron-  
men  
for  
the  
task  
node

If  
prep  
ra-  
tion  
of  
the  
de-  
ploy  
men  
en-  
vi-  
ron-  
men  
ahea  
of  
time  
is  
pos-  
si-  
ble,

If  
im-  
ple-  
men  
this  
meth  
mus  
be  
iden  
po-  
tent.  
It  
may  
be  
calle  
mul-  
ti-  
ple  
time

the same node on the same conductor.

for

This  
meth  
is  
called  
be-  
fore  
*de-*  
*ploy*

Paramete

tas  
A  
Task  
ager  
in-  
stan  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

prepare

Pre-  
pare  
the  
node  
for  
clear  
ing  
task

For  
ex-  
am-  
ple,  
node  
that  
use  
the  
Iron  
Pyth  
Age  
will  
need  
to  
boot



to do in-band cleaning tasks.

they would be set in `ironic.conductor.manager._do_node_clean`, but cannot be set when this is asynchronous. After, the interface should make an RPC call to `continue_node_cleaning` to start cleaning.

the  
rame  
in  
or-  
der

If  
the  
func  
tion  
is  
asyn  
chro  
the  
drive  
will  
need  
to  
han-  
dle  
set-  
tings  
node  
and  
node  
as

NOT  
this  
shou  
be  
mov  
to  
Boo  
In-  
ter-  
face  
whe  
it  
gets  
im-  
ple-  
men

**Parame**  
**tas**  
A  
Task  
ager  
in-

return *None*. The interface will need to call `_get_cleaning_steps` and then RPC to `continue_node_cleaning`

stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
**Returns**  
 If  
 this  
 func  
 tion  
 is  
 go-  
 ing  
 to  
 be  
 asyn  
 chro  
 shou  
 re-  
 turn  
*state*  
 Oth-  
 er-  
 wise  
 shou  
**abstract**  
 Take  
 over  
 man  
 age-  
 men  
 of  
 this  
 task.  
 node  
 from  
 a  
 deac  
 con-  
 duc-  
 tor.  
 If  
 con-  
 duc-

plemented by the driver to allow conductors to perform the necessary work during the remapping of nodes to conductors when a conductor joins or leaves the cluster.

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.

tors  
host  
main  
tain  
a  
stati  
re-  
la-  
tion-  
ship  
to  
node  
this  
meth  
shou  
be  
im-

#### For exam

Neu  
tron  
mus  
for-  
war  
DHCP  
BOC  
re-  
ques  
to  
a  
con-  
duc-  
tor  
whic  
has  
pre-  
pare  
the

#### Parame

tas  
A  
Task  
ager  
in-  
stan  
con-

sary to un-deploy that node.

tain-  
 ing  
 the  
 node  
 to  
 act  
 on.  
  
**abstract**  
 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  
 clear  
 and  
 tear  
 down  
 nec-  
 es-  
  
**Parameter**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

con-  
tain-  
ing  
the  
node  
to  
act  
on.

#### Returns

sta-  
tus  
of  
the  
de-  
ploy  
One  
of  
iron

#### `tear_down`

Tear  
down  
af-  
ter  
clear  
ing  
is  
com-  
plete  
  
Give  
that  
clear  
ing  
is  
com-  
plete  
do  
all  
clear  
and  
tear  
down  
nec-  
es-  
sary  
to  
al-  
low  
the

node to be deployed to again.

NOT  
 this  
 shou  
 be  
 mov  
 to  
 Boo  
 In-  
 ter-  
 face  
 whe  
 it  
 gets  
 im-  
 ple-  
 men

Parame

**task**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**class** i

Base  
*irc*  
*dri*  
*bas*  
*Bas*  
 In-  
 ter-  
 face  
 for  
 insp  
 relat  
 ac-  
 tions

ESSENTI

The  
 prop

er-  
ties  
re-  
quir  
by  
sche  
uler

**abort** (*t*)

Abo  
asyn  
chro  
nize  
hard  
ware  
in-  
spec  
tion.

Abo  
an  
on-  
go-  
ing  
hard  
ware  
in-  
tro-  
spec  
tion.  
this  
is  
only  
used  
for  
asyn  
chro  
nize  
base

inspect interface.

NOT  
This  
in-  
ter-  
face  
is  
calle  
with  
node  
ex-  
clu-  
sive

mentation is expected to be a quick processing.

spect interface.

lock  
held  
the  
in-  
ter-  
face  
im-  
ple-

**Parame**  
**tas**  
a  
task  
from  
Task  
ager

**Raises**  
Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
the  
meth  
is  
not  
im-  
ple-  
men  
by  
spe-  
cific  
in-

**abstrac**  
In-  
spec  
hard  
ware  
  
In-  
spec  
hard  
ware  
to  
ob-  
tain



the  
 es-  
 sen-  
 tial  
 &  
 ad-  
 di-  
 tion-  
 hard-  
 ware  
 prop-  
 er-  
 ties.

**Parameters**  
**task**  
 A  
 task  
 from  
 Task  
 ager

**Raises**  
 Har-  
 ware  
 spec-  
 tion-  
 Fail-  
 ure,  
 if  
 un-  
 able  
 to  
 get  
 es-  
 sen-  
 tial  
 hard-  
 ware  
 prop-  
 er-  
 ties.

**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** `irc`  
 Base  
*irc*  
*dri*  
*bas*  
*Bas*  
 In-  
 ter-  
 face  
 for  
 man  
 age-  
 men  
 re-  
 latec  
 ac-  
 tions

**detect\_**  
 De-  
 tects  
 store  
 and  
 re-  
 turn  
 the  
 hard  
 ware  
 ven-  
 dor.  
 If  
 the  
 Nod

intended to query Detects the BMC hardware vendor and stores the returned value with-in the Node object `properties` field if detected.

state is specified.

ob-  
ject  
pro  
field  
does  
not  
al-  
read  
con-  
tain  
a  
ven  
field  
then  
this  
meth  
is

#### Parameters

**task**  
A  
task  
from  
Task  
ager

#### Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
an  
in-  
valid  
com  
po-  
nent  
in-  
di-  
ca-  
tor  
or

#### Raises

Miss  
ing-

returns None.

Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

**Returns**  
Strin  
rep-  
re-  
sent  
ing  
the  
BM  
re-  
port  
Ven-  
dor  
or  
Man  
u-  
fac-  
ture  
oth-  
er-  
wise

**abstract**  
Get  
the  
cur-  
rent  
boot  
de-  
vice  
for  
a  
node  
Pro-  
vide

the  
 cur-  
 rent  
 boot  
 de-  
 vice  
 of  
 the  
 node  
 Be  
 awar  
 that  
 not  
 all  
 drive  
 sup-  
 port  
 this.

**Parame**  
**tas**  
 A  
 task  
 from  
 Task  
 ager

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Returns**  
 A  
 dic-  
 tio-

unknown.

nary  
con-  
tain-  
ing:

**boot\_c**  
Ahe  
boot  
de-  
vice  
one  
of  
*irc*  
*com*  
*boo*  
or  
Non  
if  
it  
is  
un-  
know

**persist**  
Whe  
the  
boot  
de-  
vice  
will  
per-  
sist  
to  
all  
fu-  
ture  
boot  
or  
not,  
Non  
if  
it  
is

**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-  
 e-  
 ter  
 is  
 miss  
 ing

**Raises**  
 Driv  
 Op-

the driver

er-  
a-  
tionl  
or  
its  
deri  
tive  
in  
case  
of  
drive  
run-  
time  
er-  
ror.

**Raises**  
Un-  
sup-  
port  
ed-  
Drive  
ten-  
sion  
if  
re-  
ques  
op-  
er-  
a-  
tion  
is  
not  
sup-  
port  
by

**Returns**  
The  
boot  
mod  
one  
of  
iro  
com  
boo  
or  
Non  
if  
it  
is



un-  
know

#### get\_inc

Get  
cur-  
rent  
state  
of  
the  
in-  
di-  
ca-  
tor  
of  
the  
hard  
ware  
com  
po-  
nent

#### Parame

- **tas**  
A  
task  
from  
Task  
ager
- **com**  
The  
hard  
ware  
com  
po-  
nent  
one  
of  
*irc*  
*com*  
*com*
- **ind**  
In-  
di-  
ca-  
tor

is specified.

ID  
 (as  
 re-  
 port  
 by  
 get\_

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 com  
 po-  
 nent  
 or  
 in-  
 di-  
 ca-  
 tor

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Returns**  
 Cur-  
 rent  
 state

of  
 the  
 in-  
 di-  
 ca-  
 tor,  
 one  
 of  
*irc*  
*com*  
*inc*

**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  
 ager

**Raises**

Miss  
 ing-  
 Pa-  
 ram  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir

the hardware

pa-  
ram-  
e-  
ter  
is  
miss  
ing

Raises

Driv  
Op-  
er-  
a-  
tionl  
or  
its  
deriv  
tive  
in  
case  
of  
drive  
run-  
time  
er-  
ror.

Raises

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
se-  
cure  
boot  
is  
not  
sup-  
port  
by  
the  
drive  
or

Returns

Boo

abstract

Get  
 sen-  
 sors  
 data  
 meth

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

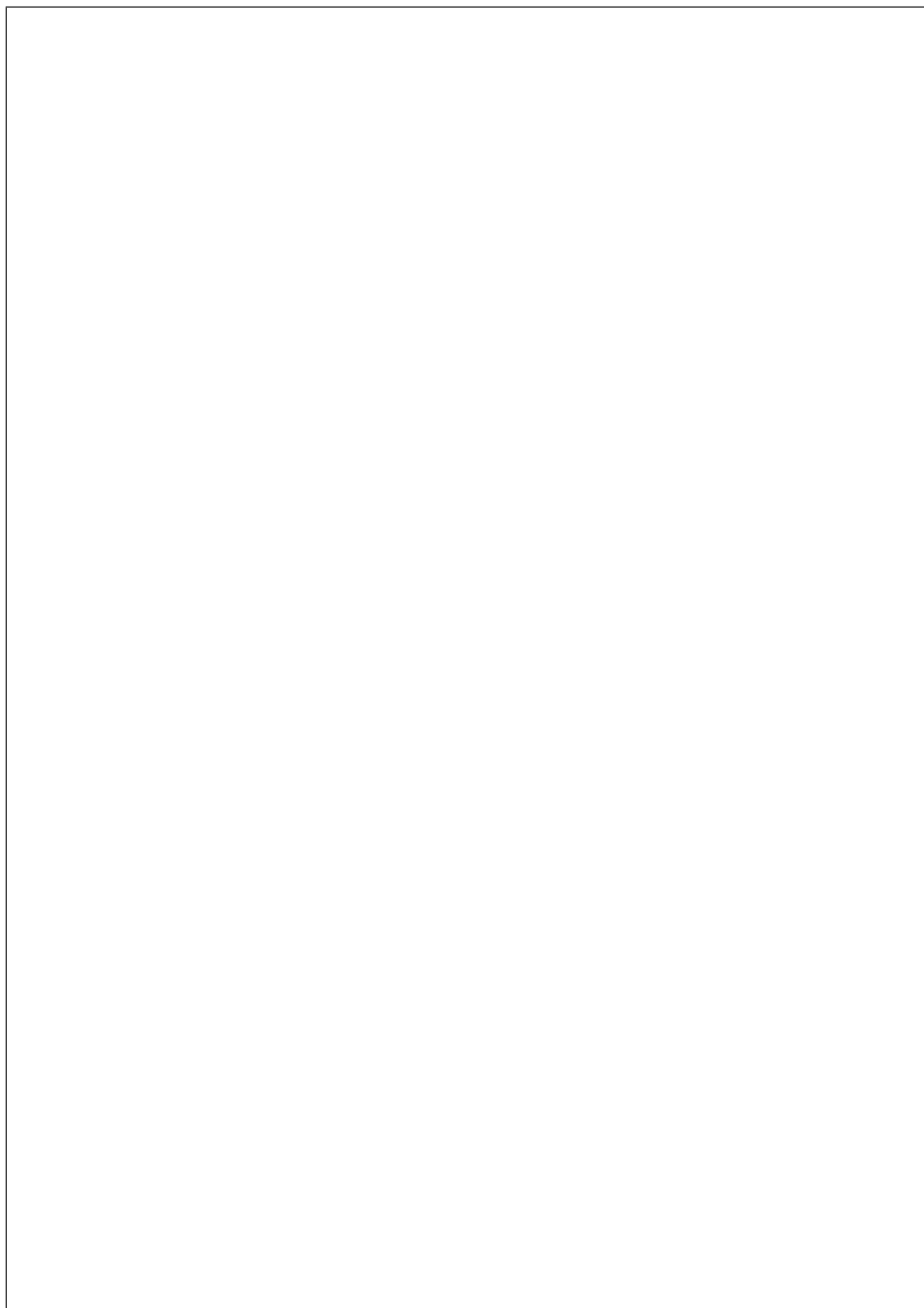
**Raises**  
 Fail  
 To-  
 Get-  
 Sen-  
 sor-  
 Data  
 whe  
 get-  
 ting  
 the  
 sen-  
 sor  
 data  
 fails

**Raises**  
 Fail  
 ToP  
 eSen  
 sor-  
 Data  
 whe  
 pars  
 ing  
 sen-  
 sor  
 data  
 fails

**Returns**  
 Re-  
 turn  
 a  
 con-  
 sis-  
 tent  
 for-

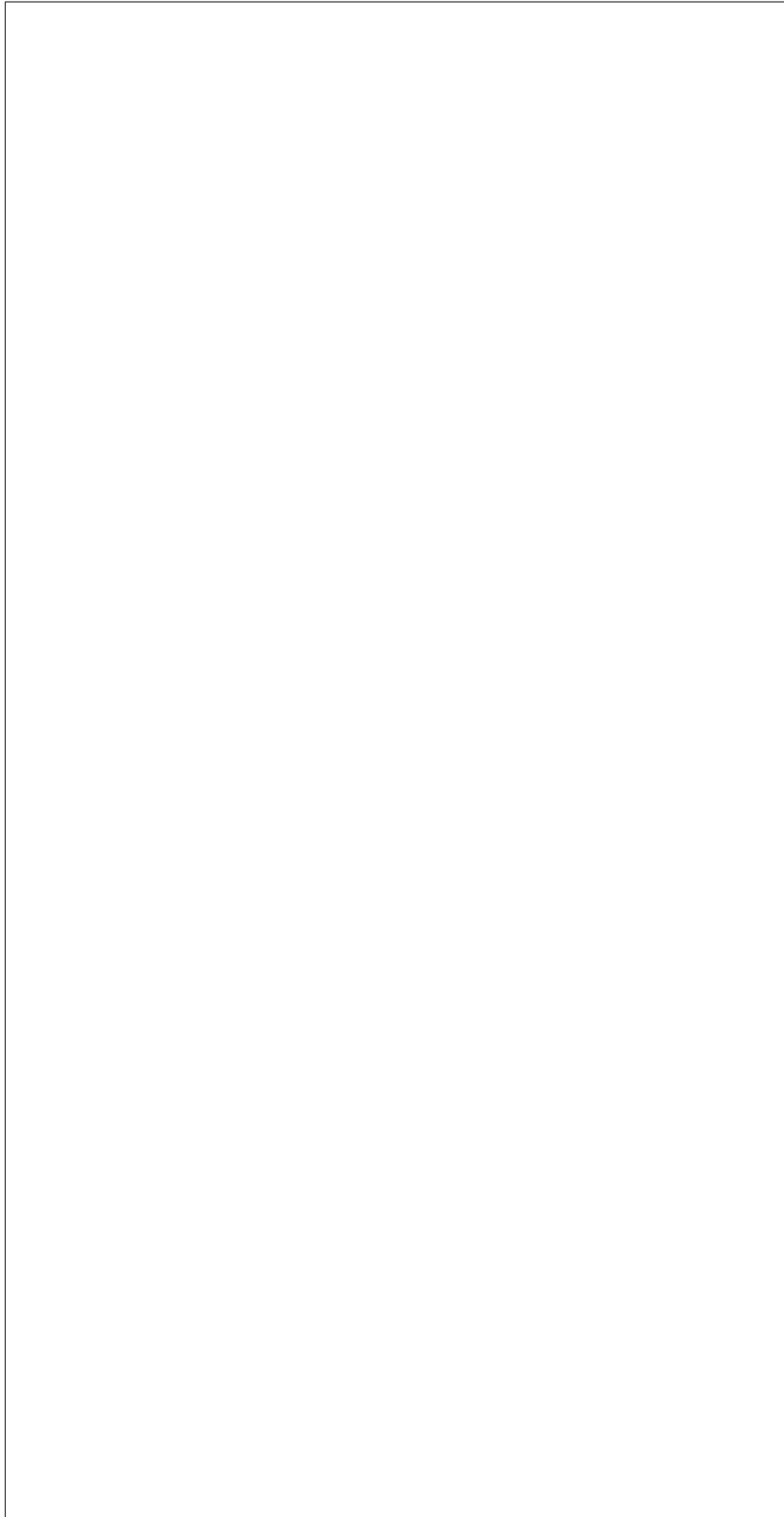
mat  
dict  
of  
sen-  
sor  
data  
grou  
by  
sen-  
sor  
type  
whic  
can

be processed by Ceilometer. eg,



(continues on next page)

(continued from previous page)



(continues on next page)

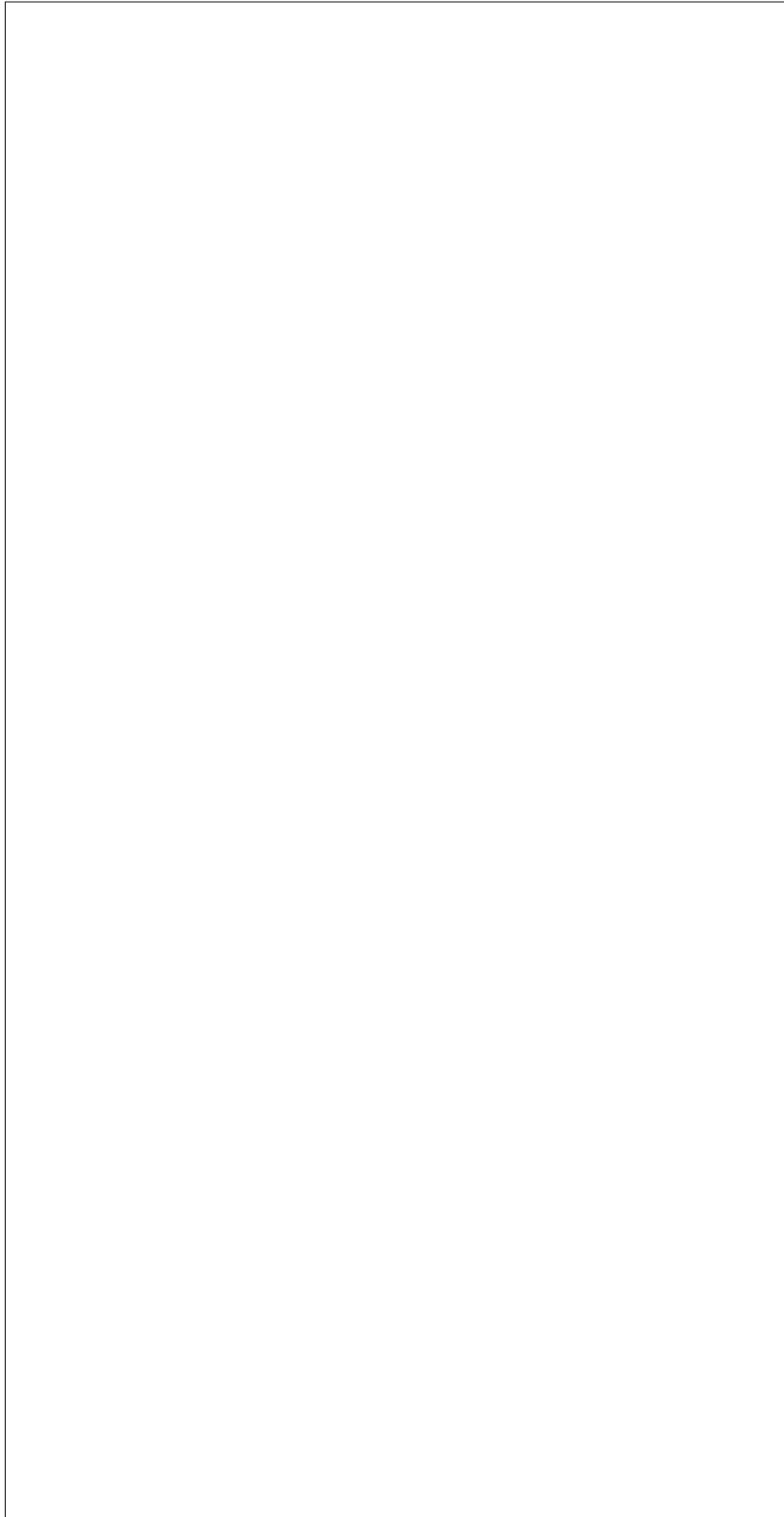
(continued from previous page)



(continues on next page)

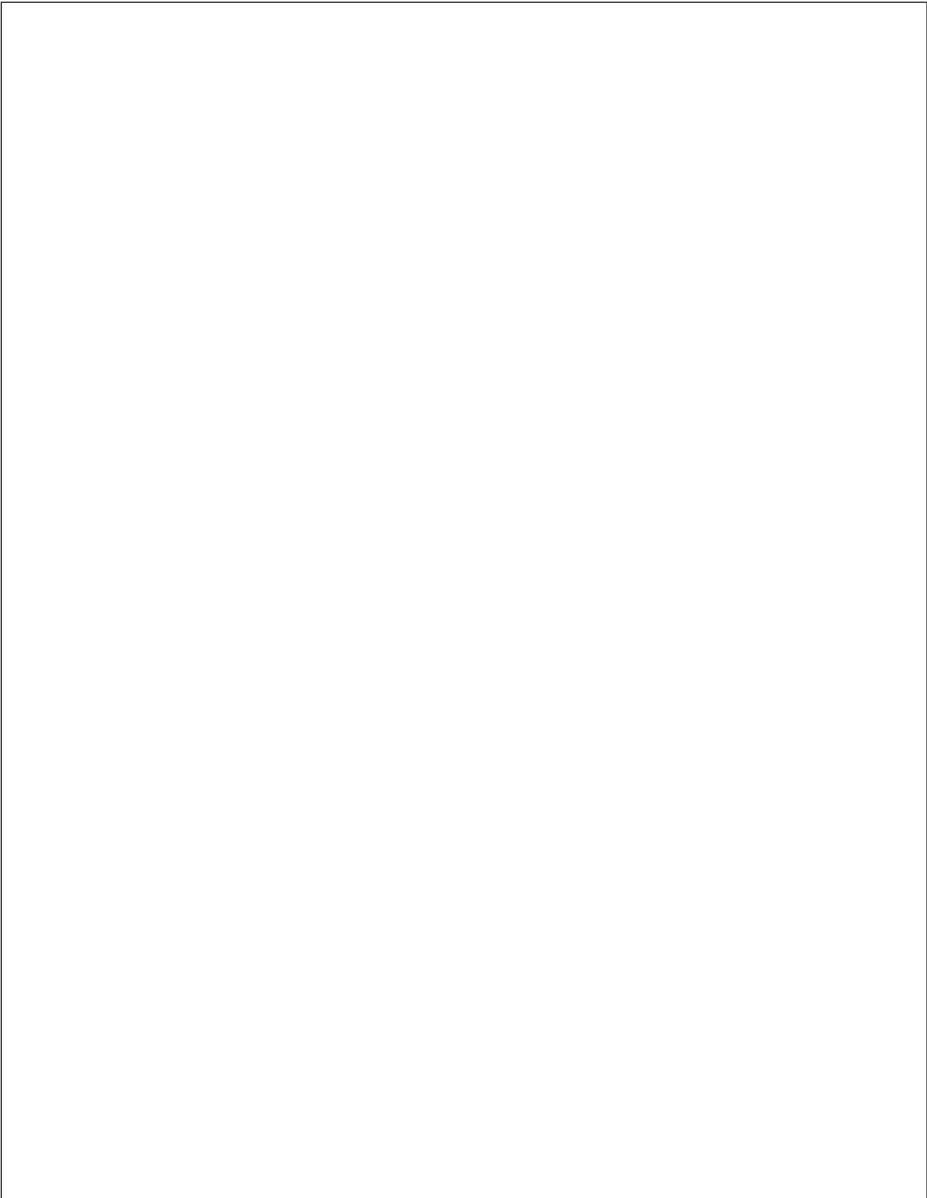


(continued from previous page)



(continues on next page)

(continued from previous page)



**abstract**

Get  
 a  
 list  
 of  
 the  
 sup-  
 port  
 boot  
 de-  
 vice

**Parameter**

**task**  
 A  
 task  
 from

Task  
ager

#### Returns

A  
list  
with  
the  
sup-  
port  
boot  
de-  
vice  
de-  
finec  
in  
*irc*  
*com*  
*boo*

#### get\_sup

Get  
a  
list  
of  
the  
sup-  
port  
boot  
mod

#### NOTE:

may  
not  
im-  
ple-  
men  
that.

#### Parame

**tas**  
A  
task  
from  
Task  
ager

#### Raises

Un-  
sup-  
port  
ed-  
Driv

the driver

ten-  
 sion  
 if  
 re-  
 ques  
 op-  
 er-  
 a-  
 tion  
 is  
 not  
 sup-  
 port  
 by

Raises

Driv  
 Op-  
 er-  
 a-  
 tion  
 or  
 its  
 deriv  
 tive  
 in  
 case  
 of  
 driv  
 run-  
 time  
 er-  
 ror.

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is

miss  
ing

**Returns**

A  
list  
with  
the  
sup-  
port  
boot  
mod  
de-  
finec  
in  
*irc*  
*com*  
*boo*  
If  
boot  
mod  
sup-  
port

cant be determined, empty list is returned.

**get\_sup**

Get  
a  
map  
of  
the  
sup-  
port  
in-  
di-  
ca-  
tors  
(e.g.  
LED

**Parame**

- **tas**  
A  
task  
from  
Task  
agen
- **com**

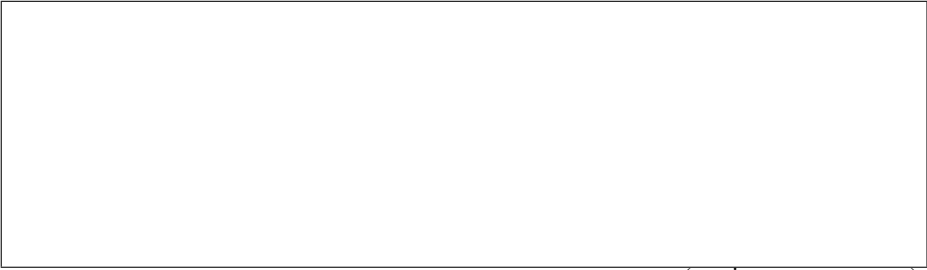
If not *Non* re- turn in- di- ca- tor in- for- ma- tion for just this com po-

nent, otherwise return indicators for all existing components.

Returns

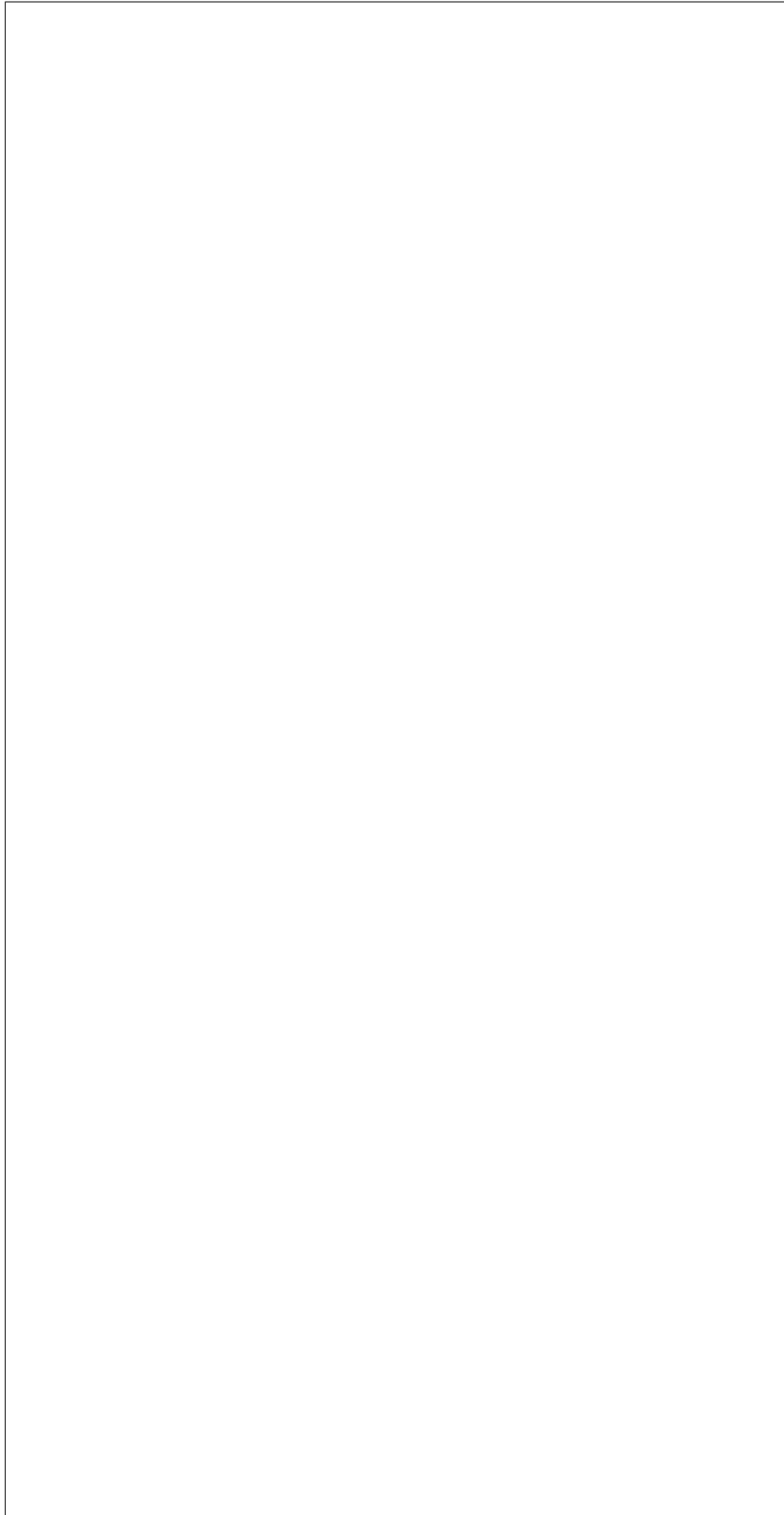
A dic- tio- nary of hard ware com po- nent (*ir com com* as keys with val- ues be- ing

dictionaries having indicator IDs as keys and indicator properties as values.



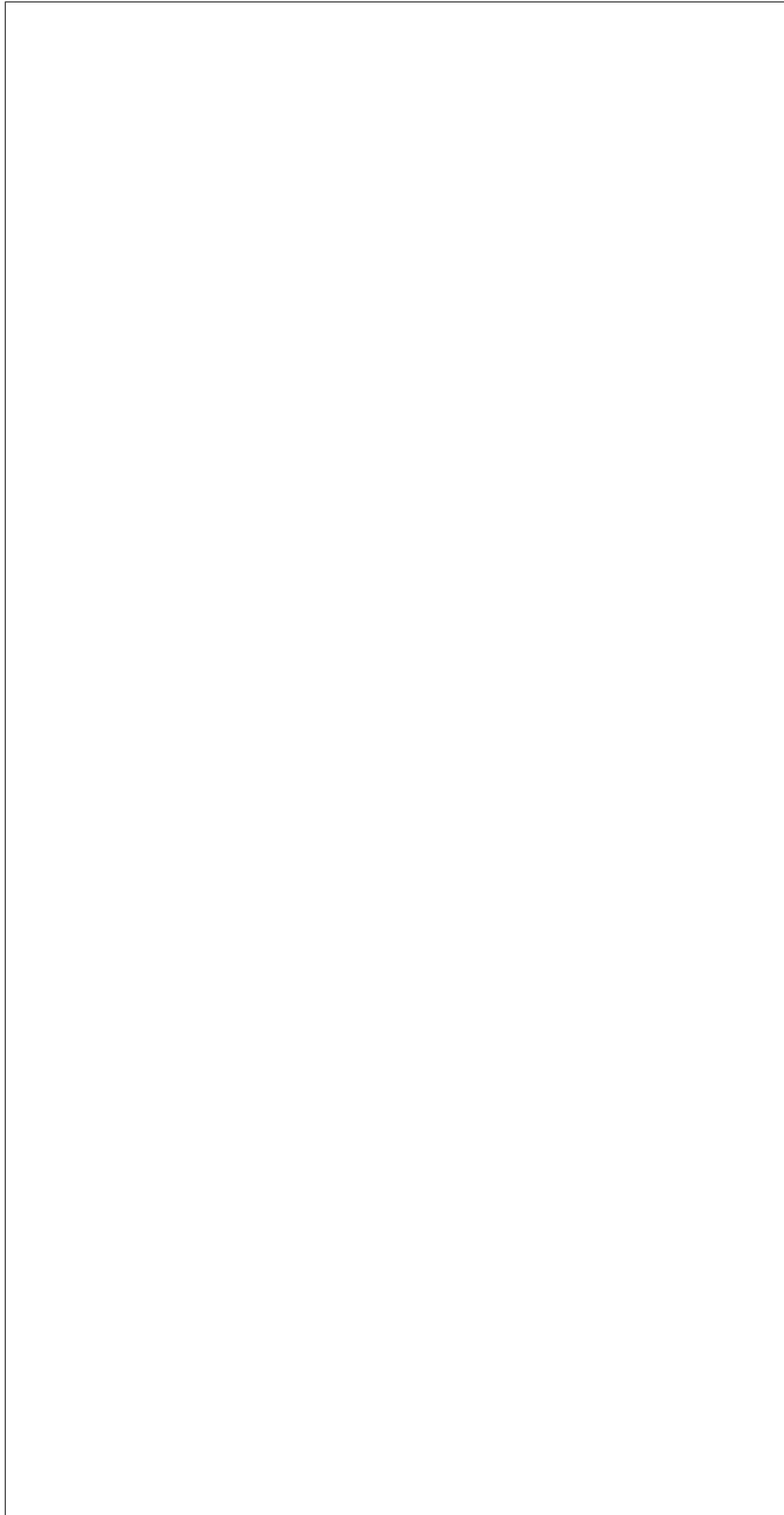
(continues on next page)

(continued from previous page)



(continues on next page)

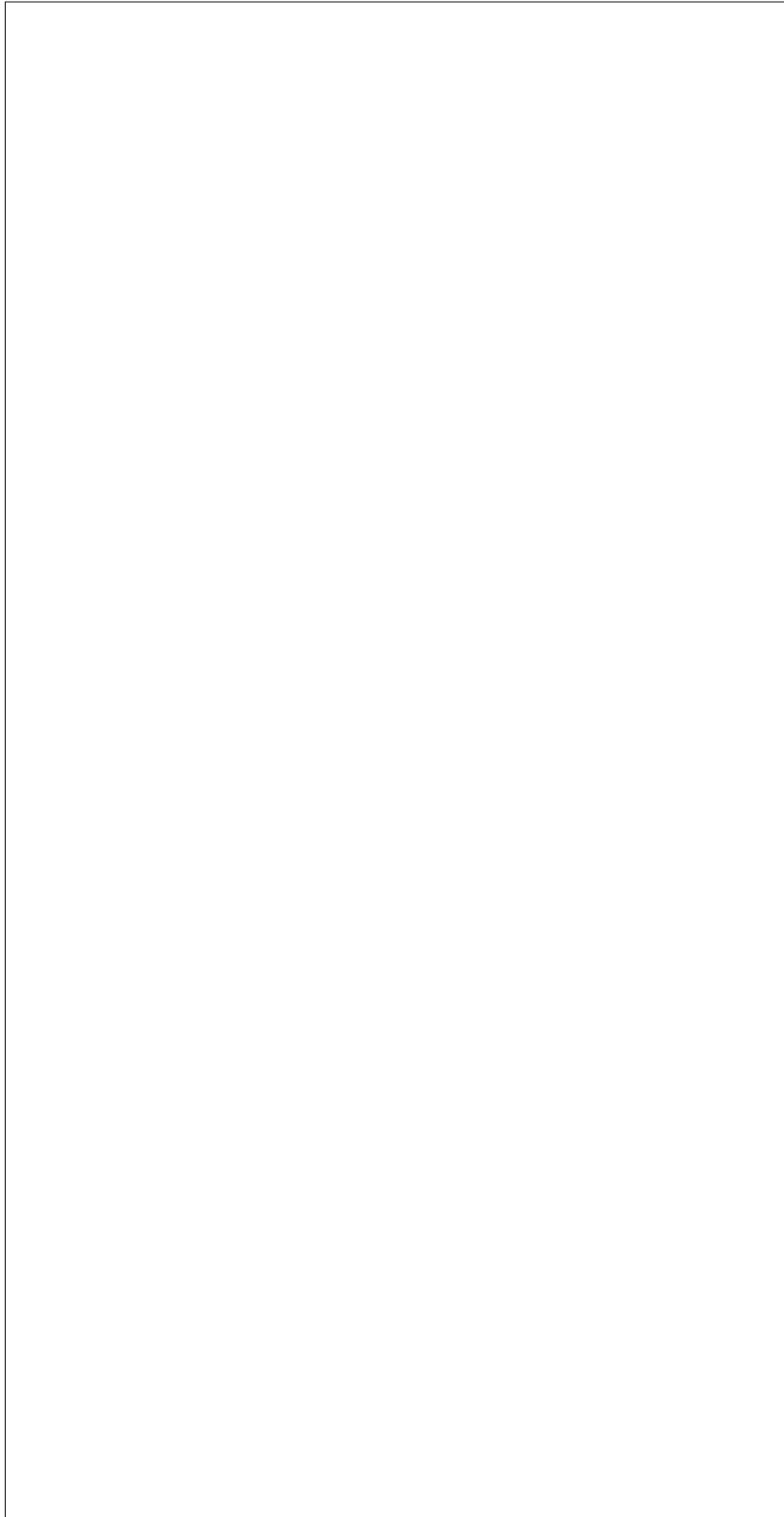
(continued from previous page)



(continues on next page)



(continued from previous page)



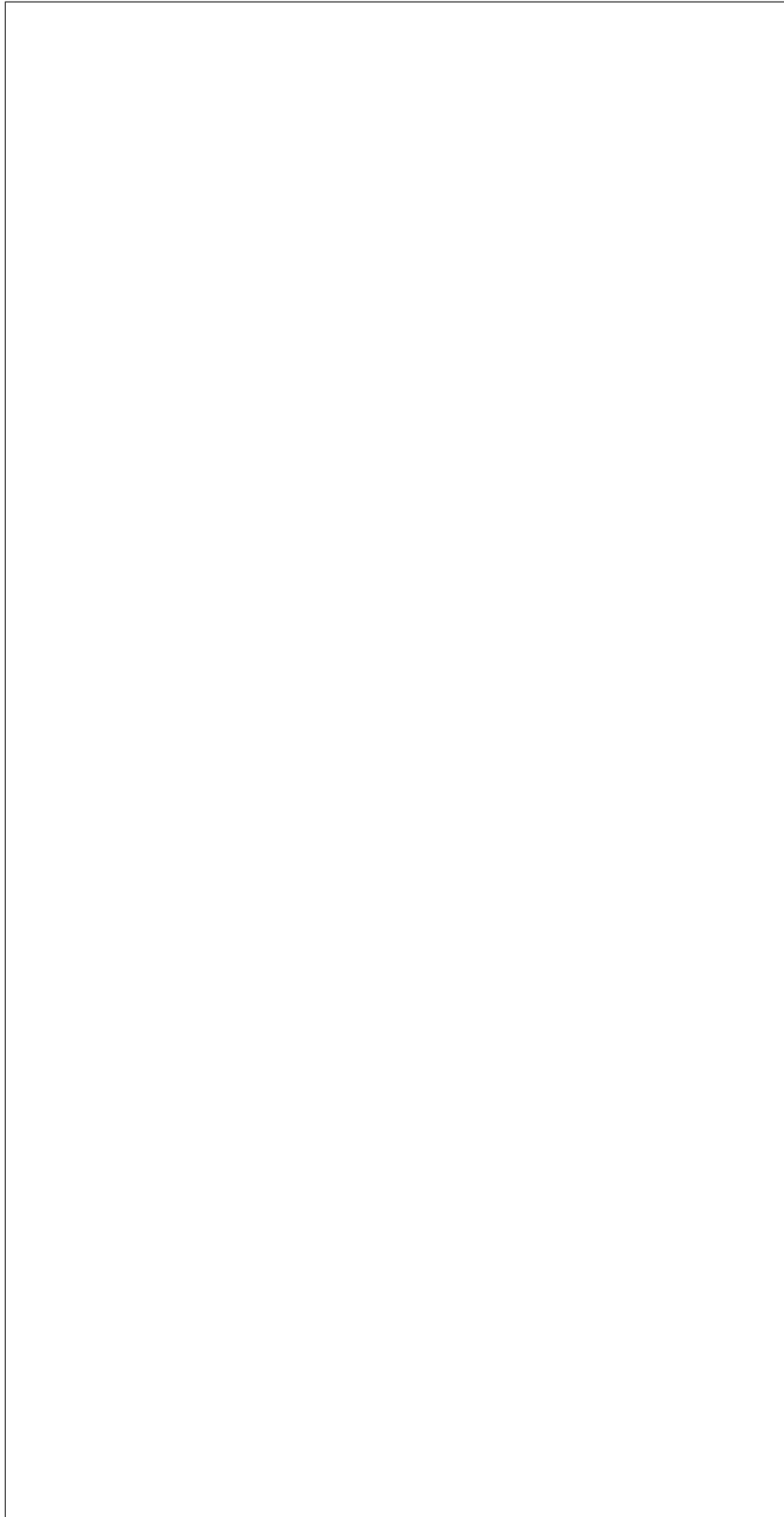
(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



(continues on next page)

(continued from previous page)



**inject\_**  
In-  
ject  
NM  
Non

Mas  
 able  
 In-  
 ter-  
 rupt  
 In-  
 ject  
 NM  
 (Nor  
 Mas  
 able  
 In-  
 ter-  
 rupt  
 for  
 a  
 node  
 im-  
 me-  
 di-  
 ately

**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

**interfa**  
 In-  
 ter-  
 face  
 type

used  
 for  
 clear  
 step  
 and  
 log-  
 ging

**abstract**

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  
 agen
- **device**  
 The  
 boot  
 de-  
 vice  
 one  
 of  
*ironic*  
*common*

boo

- **per**  
 Boo  
 valu  
 True  
 if  
 the  
 boot  
 de-  
 vice  
 will  
 per-  
 sist  
 to  
 all  
 fu-  
 ture  
 boot  
 Fals  
 if

not. Default: False.

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 boot  
 de-  
 vice  
 is  
 spec  
 i-  
 fied.

Raises

Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if

a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

#### **set\_boot**

Set  
the  
boot  
mod  
for  
a  
node

Set  
the  
boot  
mod  
to  
use  
on  
next  
re-  
boot  
of  
the  
node

Driv  
im-  
ple-  
men  
ing  
this  
meth  
are  
re-  
quir  
to  
im-  
ple-  
men  
the  
*get\_*  
meth  
as  
well



**NOTE:** I
 one
 boot
 mod
 may
 not
 im-
 ple-
 men
 that.

**Parame**

- tas**

A
 task
 from
 Task
 ager
- mod**

The
 boot
 mod
 one
 of
 *irc*
*com*
*boo*

**Raises**

In-
 valid
 Pa-
 ram-
 e-
 ter-
 Valu
 if
 an
 in-
 valid
 boot
 mod
 is
 spec
 i-
 fied.

**Raises**

the driver

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing

Raises

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion  
if  
re-  
ques  
op-  
er-  
a-  
tion  
is  
not  
sup-  
port  
by

Raises

Driv  
Op-  
er-  
a-  
tionl  
or  
its  
deri  
tive  
in

case  
 of  
 drive  
 run-  
 time  
 er-  
 ror.

**set\_inco**

Set  
 in-  
 di-  
 ca-  
 tor  
 on  
 the  
 hard  
 ware  
 com  
 po-  
 nent  
 to  
 the  
 de-  
 sired  
 state

**Parame**

- tas**  
 A  
 task  
 from  
 Task  
 ager
- com**  
 The  
 hard  
 ware  
 com  
 po-  
 nent  
 one  
 of  
*irc*  
*com*  
*com*
-

state is specified.

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*  
*inc*  
**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 an  
 in-  
 valid  
 com  
 po-  
 nent  
 in-  
 di-  
 ca-  
 tor  
 or  
**Raises**  
 Miss  
 ing-

Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**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  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss  
 ing

**Raises**

Driv  
 Op-  
 er-  
 a-  
 tionl  
 or  
 its  
 deriv  
 tive  
 in  
 case  
 of  
 drive  
 run-  
 time  
 er-  
 ror.

**Raises**

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if

the hardware

se-  
 cure  
 boot  
 is  
 not  
 sup-  
 port  
 by  
 the  
 drive  
 or

**class** `ironic.drivers.base.BaseDriver`  
 Base  
*ironic.drivers.base*  
*BaseDriver*  
 Base  
 class  
 for  
 net-  
 worl  
 in-  
 ter-  
 face

**abstract** `ironic.drivers.base.BaseDriver`  
 Add  
 the  
 clea  
 ing  
 net-  
 worl  
 to  
 a  
 node

**Parameter** `task`  
**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

**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-



is invalid.

valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 net-  
 worl  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion

**abstract**

Add  
 the  
 pro-  
 vi-  
 sion  
 ing  
 net-  
 worl  
 to  
 a  
 node

**Parameter**

**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**

Net-  
 worl  
 Er-  
 ror

**add\_resource**

Add  
 the  
 res-  
 cu-  
 ing

is invalid.

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  
 con-  
 fig-  
 u-  
 ra-  
 tion

**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  
or  
port  
grou

We  
are  
boot  
ing  
the  
node  
only  
in  
one

means were doing cleaning, of provisioning\_vif\_port\_id - provisioning, of rescuing\_vif\_port\_id - rescuing. Otherwise its a tenant network.

Parameters

- **task**  
 A Task object representing the provisioning or rescuing task.
- **p\_obj**  
 An Ironic port object representing the port to be provisioned or rescued.

Returns

VIF ID associated with the port object or None.

get\_network

Return the network configuration for the

provider, then put together collected data in form of Nova network metadata (*network\_data.json*) dict.

out-of-band.

is invalid.

A  
 Task  
 ager  
 in-  
 stan  
 Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 net-  
 worl  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 Raises  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.  
 Returns  
 a  
 dict  
 hold  
 ing  
 net-  
 worl

network metadata layout (*network\_data.json*).

con-  
fig-  
u-  
ra-  
tion  
in-  
for-  
ma-  
tion  
ad-  
hear  
ing  
Nov

#### **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\_pos**  
 Check  
 if  
 iron  
 node  
 must  
 be  
 power  
 ered  
 on  
 be-  
 fore  
 ap-  
 ply-  
 ing  
 net-  
 work  
 char

**Parameter**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Returns**  
 Boo

**abstract**  
 Han  
 dle  
 any  
 ac-  
 tions  
 re-  
 quir  
 whe  
 a  
 port  
 char

**Parameter**  
 •  
**task**  
 A  
 Task  
 ager  
 in-



stan

- **por**  
 a  
 char  
 Port  
 ob-  
 ject.

**Raises**

Con  
 flict.  
 Fail  
 ToU  
 dat-  
 eD-  
 HCF  
 tOn-  
 Port

**abstract**

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.  
 Faild  
 ToU  
 dat-  
 eD-  
 HCF  
 tOn-  
 Port

**abstract**

Re-  
 mov  
 the  
 clea  
 ing  
 net-  
 worl  
 from  
 a  
 node

**Parameter**

**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**

Net-  
 worl  
 Er-  
 ror

**remove\_**

Re-  
 mov  
 the  
 in-  
 spec  
 tion  
 net-  
 worl  
 from  
 a  
 node

**Parameter**

**task**  
 A  
 Task  
 ager  
 in-

is invalid.

standards

**Raises**

Net-  
world-  
Error-  
ror

**Raises**

In-  
valid-  
Pa-  
ram-  
e-  
ter-  
Valu-  
if  
the  
net-  
world-  
in-  
ter-  
face  
con-  
fig-  
u-  
ra-  
tion

**Raises**

Miss-  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu-  
if  
som-  
pa-  
ram-  
e-  
ters  
are  
miss-  
ing.

**abstract**

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  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Net-  
 worl  
 Er-  
 ror

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the

is invalid.

net-  
 worl  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss  
 ing.

**abstract**  
 Un-  
 con-  
 fig-  
 ure  
 ten-  
 ant  
 net-  
 worl  
 for  
 a  
 node

**Parameter**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**validation**  
 Val-

is invalid.

i-  
 date  
 the  
 net-  
 worl  
 in-  
 ter-  
 face  
 Param  
 tas  
 A  
 Task  
 ager  
 in-  
 stan  
 Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 net-  
 worl  
 in-  
 ter-  
 face  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 Raises  
 Mis  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 pa-  
 ram-  
 e-  
 ters

are  
miss  
ing.

**validat**

Val-  
i-  
date  
that  
the  
node  
has  
re-  
quir  
prop  
er-  
ties  
for  
in-  
spec  
tion.

**Parame**

**tas**  
A  
Task  
ager  
in-  
stan  
with  
the  
node  
be-  
ing  
chec

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
node  
is  
miss  
ing  
one  
or  
more  
re-

eters

quir  
pa-  
ram-

**Raises**

Un-  
sup-  
port  
ed-  
Driv  
ten-  
sion

**validat**

Val-  
i-  
date  
the  
net-  
worl  
in-  
ter-  
face  
for  
res-  
cue  
op-  
er-  
a-  
tion.

**Parame**

**tas**  
A  
Task  
ager  
in-  
stan

**Raises**

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
the  
net-  
worl  
in-



is invalid.

ter-  
face  
con-  
fig-  
u-  
ra-  
tion

Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
pa-  
ram-  
e-  
ters  
are  
miss  
ing.

abstract

At-  
tach  
a  
vir-  
tual  
net-  
worl  
in-  
ter-  
face  
to  
a  
node

Parameters

- tas  
A  
Task  
ager  
in-  
stan
-

whose value is a unique identifier for that VIF.

vif  
 a  
 dic-  
 tio-  
 nary  
 of  
 in-  
 for-  
 ma-  
 tion  
 about  
 a  
 VIF.  
 It  
 mus  
 have  
 an  
 id  
 key,

Raises

Net-  
 worl  
 Er-  
 ror,  
 Vi-  
 fAl-  
 read  
 At-  
 tach  
 NoF  
 hys-  
 i-  
 cal-  
 Port

abstract

De-  
 tach  
 a  
 vir-  
 tual  
 net-  
 worl  
 in-  
 ter-  
 face  
 from  
 a  
 node

Paramete

- 

**taskager**  
A Taskager instance

- 

**vifn\_detach**  
A VIF ID to detach

#### Raises

NetworkWorldError, VifNotFoundAttachError

#### abstract

List of attach VIF IDs for a node

#### Parameters

**taskager**  
A Taskager instance

#### Returns

List of VIF dictionaries, including

ID of the VIF.

each  
 dic-  
 tio-  
 nary  
 will  
 have  
 an  
 id  
 en-  
 try  
 with  
 the

**class** i  
 Base  
*irc*  
*dri*  
*bas*  
*Bas*  
 In-  
 ter-  
 face  
 for  
 pow  
 relat  
 ac-  
 tions

**abstrac**  
 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**

Missing-Parameter-ValueError  
if a requirement parameter is missing.

**Returns**

A power state One of *ironcore.com.standby*

**get\_supported\_power\_state**

Get a list of the supported power states

**Parameter**

**task**  
A TaskManager instance

con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**

A  
 list  
 with  
 the  
 sup-  
 port  
 pow  
 state  
 de-  
 finec  
 in  
*irc*  
*com*  
*sta*

**interfa**

In-  
 ter-  
 face  
 type  
 used  
 for  
 clea  
 step  
 and  
 log-  
 ging

**abstrac**

Per-  
 form  
 a  
 hard  
 re-  
 boot  
 of  
 the  
 task  
 node  
  
 Driv  
 are  
 ex-

it on.

Paramete

- **task**  
 A Task  
 ager  
 in-  
 stan-  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.
- **time**  
 time  
 out  
 (in  
 sec-  
 onds  
 pos-  
 i-  
 tive  
 in-  
 te-  
 ger  
 (>  
 0)

indicates to use default timeout.

for  
any  
pow  
state  
Non

Raises

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
a  
re-  
quir  
pa-  
ram-  
e-  
ter  
is  
miss  
ing.

abstract

Set  
the  
pow  
state  
of  
the  
task  
node

Parameters

- **task**  
A  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to



indicates to use default timeout.

act  
 on.  
 •  
**pow**  
 Any  
 pow  
 state  
 from  
*irc*  
*com*  
*sta*  
 •  
**tim**  
 time  
 out  
 (in  
 sec-  
 onds  
 pos-  
 i-  
 tive  
 in-  
 te-  
 ger  
 (>  
 0)  
 for  
 any  
 pow  
 state  
 Non

**Raises**  
 Miss  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 a  
 re-  
 quir  
 pa-  
 ram-  
 e-  
 ter  
 is  
 miss

of trying to force the expected power state.

ing.  
**support**  
 Che  
 if  
 pow  
 sync  
 is  
 sup-  
 port  
 for  
 the  
 give  
 node  
  
 If  
 Fal  
 the  
 con-  
 duc-  
 tor  
 will  
 sim-  
 ply  
 store  
 wha  
 ever  
 get  
 re-  
 turn  
 in  
 the  
 data  
 in-  
 stea

**Paramet**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

## Returns

bool  
 whe  
 pow  
 sync  
 is  
 sup-  
 port

## class i

Base  
*irc*  
*dri*  
*bas*  
*Bas*

## apply\_c

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-

ified in `raid_config`. Default value is `True`.

cept the root volume) in `raid_config`. Default value is `True`.

ply.
   
 •
   
**create\_root\_volume**
  
 Set-
   
 ting
   
 this
   
 to
   
`False`
  
 in-
   
 di-
   
 cate
   
 not
   
 to
   
 cre-
   
 ate
   
 root
   
 vol-
   
 ume
   
 that
   
 is
   
 spec
   
 •
   
**create\_non\_root\_volume**
  
 Set-
   
 ting
   
 this
   
 to
   
`False`
  
 in-
   
 di-
   
 cate
   
 not
   
 to
   
 cre-
   
 ate
   
 non-
   
 root
   
 vol-
   
 ume
   
 (all
   
 ex-
   
 •
   
**delete\_root\_volume**
  
 Set-
   
 ting
   
 this
   
 to
   
`True`

creating the new configuration.

in-  
 di-  
 cate  
 to  
 dele  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 prior  
 to

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in-  
 valid

Returns

state  
 if  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn  
 chro  
 or  
 Non  
 if

plete.

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.

it  
is  
com

**abstract**

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  
It  
as-  
sum  
that  
the

**Paramete**

- **tas**  
A

ified in the nodes `target_raid_config`. Default value is `True`.

cept the root volume) in the nodes `target_raid_config`. Default value is `True`.

Task  
ager  
in-  
stan

- **cre**  
Set-  
ting  
this  
to  
Fals  
in-  
di-  
cate  
not  
to  
cre-  
ate  
root  
vol-  
ume  
that  
is  
spec
- **cre**  
Set-  
ting  
this  
to  
Fals  
in-  
di-  
cate  
not  
to  
cre-  
ate  
non-  
root  
vol-  
ume  
(all  
ex-
- **del**  
Set-  
ting

creating the new configuration.

chronously, or None if it is complete.

this  
 to  
 True  
 in-  
 di-  
 cate  
 to  
 dele  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 prior  
 to

Returns

state  
 (clea  
 ing)  
 or  
 state  
 (de-  
 ploy  
 men  
 if  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 is  
 in  
 prog  
 asyn

abstract

Dele  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion  
 on  
 the  
 give  
 node



This method deletes the RAID configuration on the node. After RAID configuration is deleted, node.raid\_config should be cleared by the implementation.

**Parameters:**  
 A TaskManager instance.

**Returns:**  
 state (clearing) or state (de- ploy men if dele tion is in prog asyn chro or Non if

it is complete.

fied for logical disks and a textual description for them.

Get  
the  
prop  
er-  
ties  
that  
can  
be  
spec  
i-  
fied  
for  
log-  
i-  
cal  
disk  
  
This  
meth  
re-  
turn  
a  
dic-  
tio-  
nary  
con-  
tain-  
ing  
the  
prop  
er-  
ties  
that  
can  
be  
spec  
i-  
  
**Returns**  
A  
dic-  
tio-  
nary  
con-  
tain-  
ing  
prop  
er-  
ties  
that  
can

disks and a textual description for them.

be  
men  
tion  
for  
log-  
i-  
cal

#### **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

#### **interfa**

In-  
ter-  
face  
type  
used  
for  
clea  
step  
and  
log-  
ging

#### **validat**

Val-  
i-  
date

Driver implementations of this interface can override this method for doing more validations (such as BMCs credentials).

the  
 RAI  
 In-  
 ter-  
 face  
 This  
 meth  
 meth  
 val-  
 i-  
 date  
 the  
 prop  
 er-  
 ties  
 de-  
 fine  
 by  
 Iron  
 for  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.  
**Parame**  
**tas**  
 A  
 Task  
 ager  
 in-  
 stan  
**Raises**  
 In-  
 valid  
 Pa-  
 ram  
 e-  
 ter-  
 Valu  
 if  
 the  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion

is  
 in-  
 valid

**Raises**

Miss-  
 ing-  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu-  
 if  
 som-  
 pa-  
 ram-  
 e-  
 ters  
 are  
 miss-  
 ing.

**validat**

Val-  
 i-  
 date  
 the  
 give  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.

This  
 meth-  
 val-  
 i-  
 date  
 the  
 give  
 RAI  
 con-  
 fig-  
 u-  
 ra-  
 tion.  
 Driv-  
 im-  
 ple-  
 men-  
 ta-

this interface can override this method to support custom parameters for RAID configuration.

Parameters

- task**  
 A TaskManager instance
- raid**  
 The RAID configuration to validate

Raises

InvalidParameterError if the RAID configuration is invalid

ironic.  
 This may be used

menting an `apply_configuration` deploy step.

**class** `ironicclient.drivers.bas`  
 Base  
*ironicclient.drivers.bas*  
*Base*  
 In-  
 ter-  
 face  
 for  
 resc  
 relat  
 ac-  
 tions

**clean\_u**  
 Clea  
 up  
 the  
 res-  
 cue  
 en-  
 vi-  
 ron-  
 men  
 for  
 the  
 task  
 node  
 This  
 is  
 par-  
 tic-

out occurs.

u-  
larly  
use-  
ful  
for  
node  
when  
res-  
cu-  
ing  
is  
asyn-  
chro-  
and  
a  
time

Paramete

tas  
A  
Task  
ager  
in-  
stan-  
con-  
tain-  
ing  
the  
node  
to  
act  
on.

Returns

Non

interf

In-  
ter-  
face  
type  
used  
for  
clea-  
step  
and  
log-  
ging

abstrac

Boo  
the



task  
 node  
 into  
 a  
 res-  
 cue  
 en-  
 vi-  
 ron-  
 men

Param

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  
 op-  
 er-  
 a-  
 tion  
 fails

Returns

state  
 if  
 res-  
 cue

is  
 in  
 prog  
 asyn  
 chro  
 or  
 state  
 if  
 it  
 is  
 com  
 plete

**abstract**

Tear  
 dow  
 the  
 res-  
 cue  
 en-  
 vi-  
 ron-  
 men  
 and  
 re-  
 turn  
 to  
 nor-  
 mal.

**Parameter**

**task**  
 A  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Raises**

In-  
 stan  
 Un-  
 res-  
 cue-  
 Fail.

ation fails.

ure  
if  
node  
val-  
i-  
da-  
tion  
or  
un-  
res-  
cue  
op-  
er-

## Returns

state  
if  
it  
is  
suc-  
cess  
ful.

## class i

Base  
*irc*  
*dri*  
*bas*  
*Bas*

Base  
class  
for  
stor-  
age  
in-  
ter-  
face

## abstract

In-  
form  
the  
stor-  
age  
sub-  
sys-  
tem  
to  
at-  
tach

all  
 vol-  
 ume  
 for  
 the  
 node

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion

**abstract**  
 In-  
 form  
 the  
 stor-  
 age  
 sub-  
 sys-  
 tem  
 to  
 de-  
 tach  
 all  
 vol-  
 ume  
 for  
 the  
 node

**Parameters**  
**task**  
 A  
 Task  
 ager  
 in-  
 stan

**Raises**  
 Un-  
 sup-

port  
 ed-  
 Driv  
 ten-  
 sion

**interfa**

In-  
 ter-  
 face  
 type  
 used  
 for  
 clear  
 step  
 and  
 log-  
 ging

**abstrac**

De-  
 ter-  
 mine  
 if  
 de-  
 ploy  
 shou  
 per-  
 form  
 the  
 im-  
 age  
 writ  
 out.

**Parame**

**tas**  
 A  
 Task  
 ager  
 in-  
 stan

**Returns**

Boo  
 valu  
 to  
 in-  
 di-  
 cate  
 if  
 the  
 in-

ten by Ironic.

ter-  
face  
ex-  
pect  
the  
im-  
age  
to  
be  
writ

**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  
veno

or  
driv  
spec  
ca-  
pa-  
bil-  
i-

method in the class inheriting from this class and use the `@passthru` or `@driver_passthru` decorators.

ties  
shou  
be  
im-  
ple-  
men  
as  
a

Met  
ods  
dec-  
o-  
ratec  
with  
@dr  
shou  
be  
shor  
livec  
be-  
caus  
it  
is  
a  
bloc  
ing  
call.

#### **driver\_**

Val-  
i-  
date  
drive  
vend  
pass  
ac-  
tions  
  
If  
in-  
valid  
raise  
an  
ex-  
cep-  
tion.  
oth-  
er-  
wise  
re-

turn  
 Non

**Parameters**

- **method**  
 method to be validated.  
 i.e. date
- **kwargs**  
 info for action.

**Raises**

Missing parameter.  
 Invalid parameter.  
 ValueError.  
 if kwargs does not contain certain parameter.  
 ValueError.

**Raises**

Invalid parameter.  
 ValueError.  
 if parameter.



e-  
ter  
does  
not  
matc

#### **interfa**

In-  
ter-  
face  
type  
used  
for  
clear  
step  
and  
log-  
ging

#### **abstrac**

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

faces.

- **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  
sup-  
port  
in-  
ter-

Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
kwa  
does  
not  
con-  
tain  
meth

**Raises**

Miss  
ing-  
Pa-  
ram-  
e-  
ter-  
Valu

**class** i

Base  
tup

**metadat**

Alia  
for  
field  
num  
ber  
1

**method**

Alia  
for  
field  
num  
ber  
0

ironic.

A  
dec-  
o-  
ra-  
tor  
to  
cach  
bios  
set-  
ting  
af-  
ter  
run-  
ning  
the  
func  
tion.

**Paramet**

**fun**  
Func  
tion  
or

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.

meth  
to  
wrap  
  
ironic.  
Dec  
o-  
ra-  
tor  
for  
clea  
ing  
step  
  
Clea  
ing  
step  
may  
be  
used  
in  
man  
ual  
or  
au-  
to-  
mate  
clea  
ing.  
  
For  
au-  
to-  
mate  
clea  
ing,  
only  
step  
with  
pri-  
or-  
i-  
ties  
grea  
than  
0  
are  
used  
The  
step

to automated cleaning, but the steps and order of execution must be explicitly specified by the user when invoking the cleaning API.

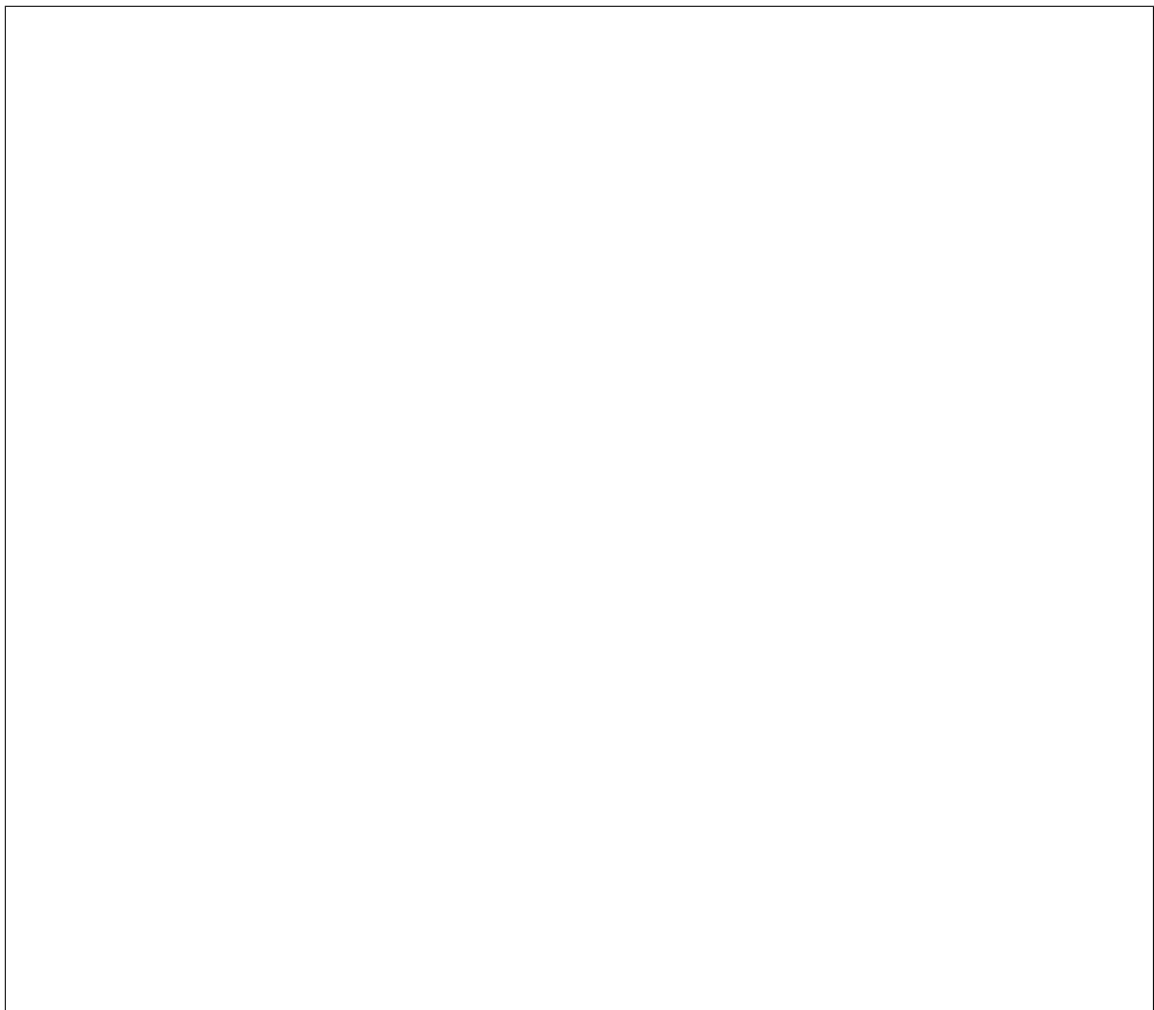
ject. Clean steps used in manual cleaning may also take keyword variable arguments (as described in `argsinfo`).

For  
man  
ual  
clea  
ing,  
the  
clea  
step  
will  
be  
ex-  
e-  
cute  
in  
a  
sim-  
i-  
lar  
fash  
ion

Dec  
o-  
rate  
clea  
step  
mus  
take  
as  
the  
only  
po-  
si-  
tion  
ar-  
gu-  
men  
a  
Task  
ager  
ob-

Clea  
step  
can  
be  
ei-  
ther

turn *None* when finished, and the conductor will continue on to the next step. While the clean step is executing, the node will be in *states.CLEANING* provision state. If the step is asynchronous, the step should return *states.CLEANWAIT* to the conductor before it starts the asynchronous work. When the step is complete, the step should make an RPC call to *continue\_node\_clean* to move to the next step in cleaning. The node will be in *states.CLEANWAIT* provision state during the asynchronous work.



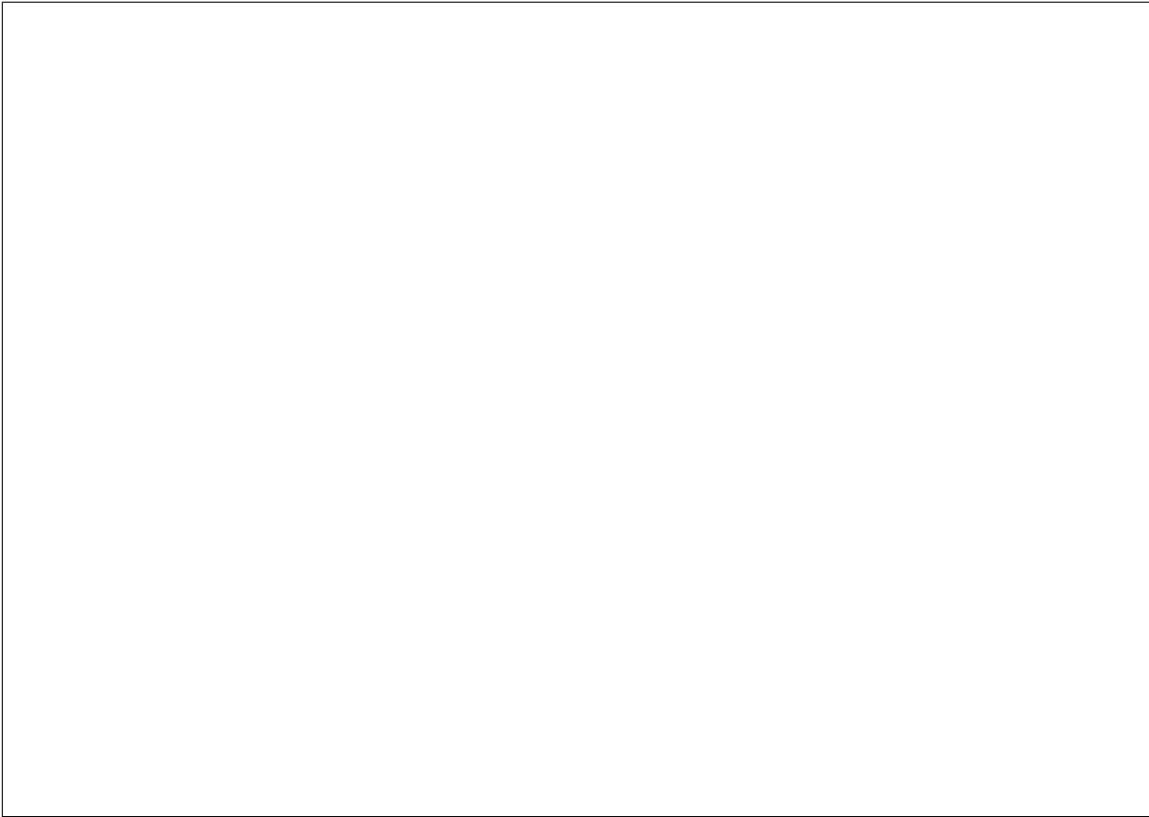
(continues on next page)

(continued from previous page)

```
↪ {'size': {'description': 'size of widget (MB)',
```

(continues on next page)

(continued from previous page)



Paramet

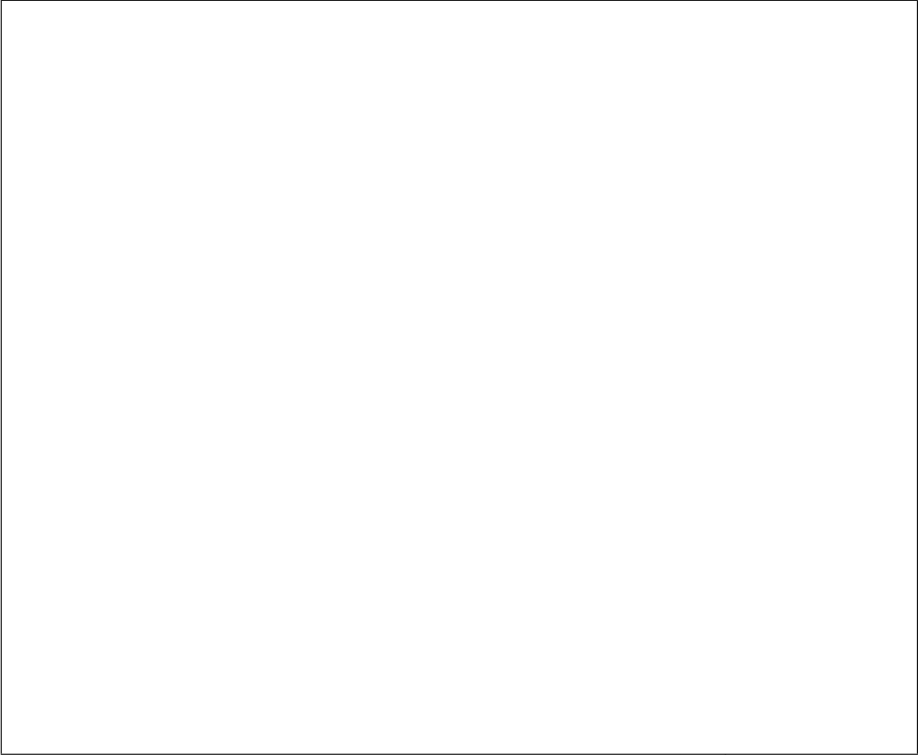
- **pri**  
 an  
 in-  
 te-  
 ger  
 pri-  
 or-  
 ity,  
 shou  
 be  
 a  
 COM  
 op-  
 tion
- **abo**  
 Boo  
 valu  
 Whe  
 the  
 clea  
 step



is  
 abor  
 or  
 not;  
 de-  
 fault  
 to  
 Fals

• **arg**  
 a  
 dic-  
 tio-  
 nary  
 of  
 key-  
 wor  
 ar-  
 gu-  
 men  
 whe  
 key  
 is  
 the  
 nam  
 of  
 the  
 ar-

gument and value is a dictionary as follows:



(continues on next page)

(continued from previous page)

→it must be specified **in**

→optional.

Raises

*Inv*

if

any

of

the

ar-

gu-

men

are

in-

valid

ironic.

Dec

o-

ra-

tor

for

de-

ploy

men

step

Only

step

with

pri-

or-

i-

ties

grea

than

0

are

used

Thes

step

are

or-

dere

by

pri-

or-

ity from highest value to lowest value. For steps with the same priority, they are ordered by driver interface priority (see `conductor.steps.DEPLOYING_INTERFACE_PRIORITY`). `execute_deploy_step()` will be called on each step.

Dec

o-

rated

de-

object.

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.



(continues on next page)

ploy  
step  
mus  
take  
as  
the  
only  
po-  
si-  
tiona  
ar-  
gu-  
men  
a  
Task  
ager

De-  
ploy  
step  
can  
be  
ei-  
ther  
syn-  
chro  
or  
asyn  
chro  
If  
the  
step  
is  
syn-  
chro  
it  
shou

Ex-  
am-  
ples

(continued from previous page)



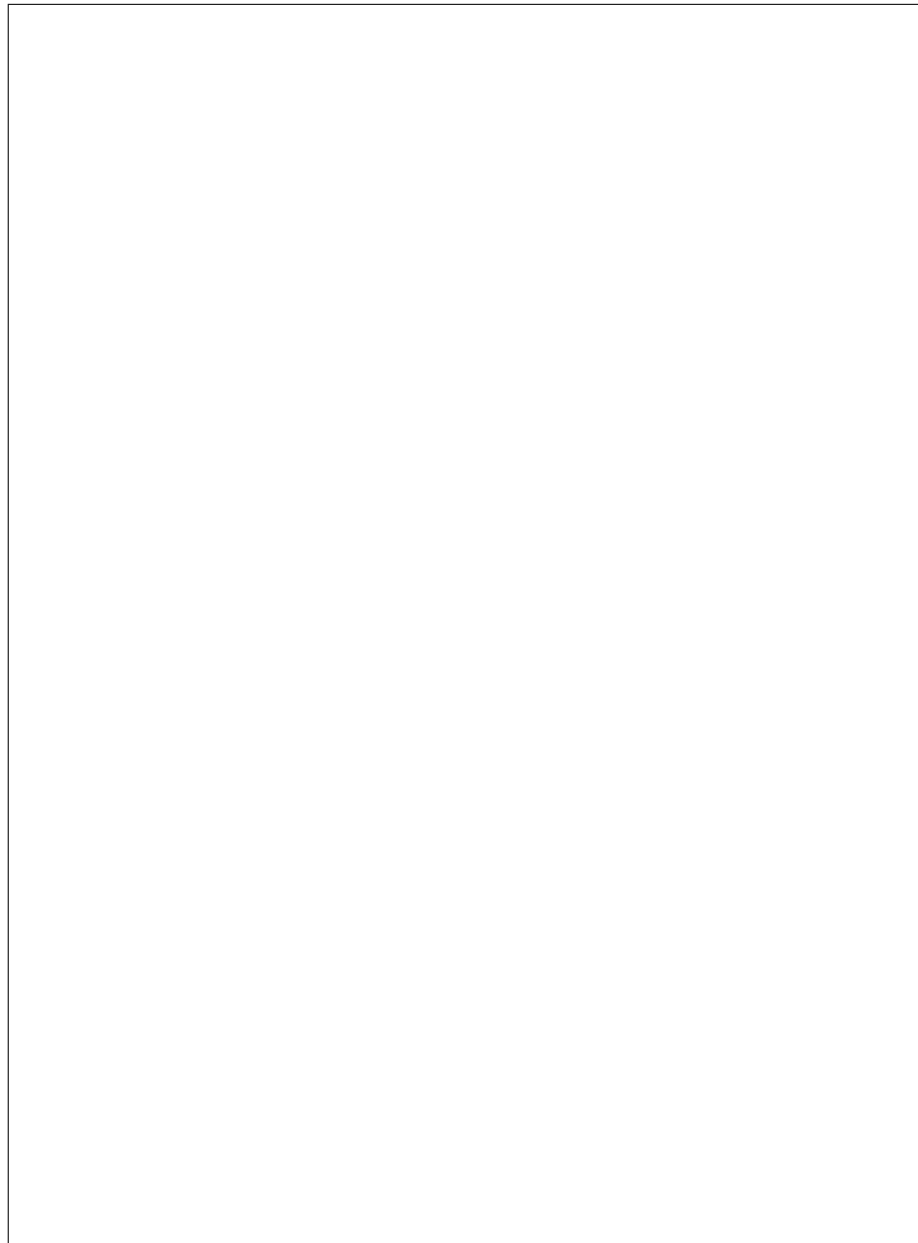
Parameter

- **priority**  
 an integer (>=0) priority; used for determining the order in
- **arguments**  
 a dictionary

which the step is run in the deployment process.

nary  
of  
key-  
wor  
ar-  
gu-  
men  
whe  
key  
is  
the  
nam  
of  
the  
ar-

gument and value is a dictionary as follows:



(continues on next page)

(continued from previous page)

→it must be specified **in**

→optional.

**Raises**

*Inv*

if  
any  
of  
the  
ar-  
gu-  
men-  
are  
in-  
valid

ironic.

ironic.

ironic.drivers.drac module

DRAC  
 Driv  
 for  
 re-  
 mote  
 sys-  
 tem  
 man  
 age-  
 men  
 us-  
 ing  
 Dell  
 Re-  
 mote  
 Ac-  
 cess  
 Card

**class** i  
 Base  
*irc*  
*dri*  
*gen*  
*Gen*  
 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** i

Base  
*irc*  
*dri*  
*har*  
*Abs*

Fake  
 hard  
 ware  
 type

This  
 hard  
 ware  
 type  
 is  
 spec  
 case  
 in  
 the  
 drive  
 fac-  
 tory

verification. Thus, supported\_\* methods here are only for calculating the defaults, not for actual check.

ration.

to  
by-  
pass  
com  
pat-  
i-  
bil-  
ity

All  
fake  
im-  
ple-  
men-  
ta-  
tions  
are  
still  
ex-  
pect  
to  
be  
en-  
able  
in  
the  
con-  
fig-  
u-

#### **property**

List  
of  
class  
of  
sup-  
port  
bios  
in-  
ter-  
face

#### **property**

List  
of  
class  
of  
sup-  
port  
boot  
in-

ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
con-  
sole  
in-  
ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
de-  
ploy  
in-  
ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
in-  
spec  
in-  
ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
man  
age-  
men  
in-

ter-  
face

**property**

List  
of  
sup-  
port  
net-  
work  
in-  
ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
pow  
in-  
ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
raid  
in-  
ter-  
face

**property**

List  
of  
class  
of  
sup-  
port  
res-  
cue  
in-  
ter-  
face

**property**

List  
of

class  
 of  
 sup-  
 port  
 stor-  
 age  
 in-  
 ter-  
 face

**property**  
 List  
 of  
 class  
 of  
 sup-  
 port  
 res-  
 cue  
 in-  
 ter-  
 face

ironic.drivers.generic module

Gen  
 hard  
 ware  
 type

**class** `ironic.drivers.generic.HardwareDriver`  
 Base  
*ironic.drivers.generic.HardwareDriver*  
*AbstractHardwareDriver*  
 Ab-  
 strac  
 base  
 class  
 rep-  
 re-  
 sent  
 ing  
 gene  
 hard  
 ware  
 This  
 class

provide  
 rea-  
 son-  
 able  
 de-  
 fault  
 for  
 all  
 of  
 the  
 in-  
 ter-  
 face

**property**  
 List  
 of  
 sup-  
 port  
 boot  
 in-  
 ter-  
 face

**property**  
 List  
 of  
 sup-  
 port  
 de-  
 ploy  
 in-  
 ter-  
 face

**property**  
 List  
 of  
 sup-  
 port  
 in-  
 spec  
 in-  
 ter-  
 face

**property**  
 List  
 of  
 sup-  
 port

net-  
 worl  
 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** `IronicDriverGenerator`  
 Base  
*ironic*  
*driver*  
*generator*  
*Generator*  
 Harc  
 ware  
 type  
 that  
 uses  
 man  
 ual  
 pow  
 and



ting 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.

boot  
man  
age-  
men  
  
Us-  
ing  
this  
hard  
ware  
type  
as-  
sum  
that  
an  
op-  
er-  
a-  
tor  
man  
ages  
re-  
boot  
and  
set-

#### 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.hardware\_type module

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.

driver interface (power, deploy, etc).

in-  
ter-  
face

A  
hard  
ware  
ware  
type  
de-  
fines  
an  
or-  
dere  
list  
of  
sup-  
port  
im-  
ple-  
men  
ta-  
tions  
for  
each

**get\_prop**

Get  
the  
prop  
er-  
ties  
of  
the  
hard  
ware  
type

Note  
that  
this  
re-  
turn  
prop  
er-  
ties  
for  
the  
de-  
fault  
in-

hardware type. Since this is not node-aware, interface overrides cant be detected.

ter-  
face  
of  
each  
type  
for  
this  
  
**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  
  
**abstrac**  
 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

**property**  
 List  
 of  
 sup-  
 port  
 net-

worl  
in-  
ter-  
face

**abstract**

List  
of  
sup-  
port  
pow  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
raid  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
res-  
cue  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
stor-  
age  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
ven-  
dor

in-  
ter-  
face

**ironic.drivers.ibmC module**

iBM  
Drive  
for  
man  
ag-  
ing  
HUA  
V5  
se-  
ries  
rack  
serv  
such  
as  
2288  
V5,  
CHI  
V5.

**class** i  
Base  
*irc*  
*dri*  
*gen*  
*Gen*  
  
Hua  
iBM  
hard  
ware  
type

**property**  
List  
of  
sup-  
port  
man  
age-  
men  
in-  
ter-  
face

**property**
 List
 of
 sup-
 port
 pow
 in-
 ter-
 face

**property**
 List
 of
 sup-
 port
 raid
 in-
 ter-
 face

**property**
 List
 of
 sup-
 port
 ven-
 dor
 in-
 ter-
 face

ironic.drivers.ilo module

iLO
 Driv
 for
 man
 ag-
 ing
 HP
 Pro-
 liant
 Gen
 and
 abov
 serv

**class** i
 Base
 *irc*
*dri*



*ilc*

*Ilc*

iLO.

hard

ware

type

iLO.

hard

ware

type

is

tar-

gete

for

iLO.

base

Pro-

liant

Gen

serv

#### **property**

List

of

sup-

port

boot

in-

ter-

face

#### **property**

List

of

sup-

port

man

age-

men

in-

ter-

face

#### **property**

List

of

sup-

port

raid

in-

ter-

face

**class** iBase  
 Base class for iLO hardware.  
*iro*  
*dri*  
*gen*  
*Gen*

iLO hardware type

iLO hardware type

is targeted for iLO 4 base Pro-liant Gen and Gen serv

**property** List of supported bios interface

**property** List of supported boot interface

**property**

List  
of  
sup-  
port  
con-  
sole  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
in-  
spec  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
man  
age-  
men  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
pow  
in-  
ter-  
face

**property**

List  
of  
sup-  
port  
pow  
in-  
ter-  
face

ironic.drivers.intel\_ipmi module

tations via `shellinabox` or `socat`. Supports Intel SST-PP feature.

**class** `ironic.drivers.intel_ipmi.IPMI`  
 Base class for IPMI hardware type.  
 Uses `ipmitool` to implement power and management. Provides serial console implementations.

**property** `serial_console`  
 List of supported management interfaces.

## ironic.drivers.ipmi module

Hard-  
ware  
type  
for  
IPM  
(us-  
ing  
ip-  
mi-  
tool)

**class** `ironi`  
Base  
*ironi*  
*driv*  
*gen*  
*Gen*

IPM  
hard-  
ware  
type

Uses  
ipm  
to  
im-  
ple-  
men  
pow  
and  
man  
age-  
men  
Pro-  
vide  
se-  
rial  
con-  
sole  
im-  
ple-  
men

tations 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  
 Driv  
 for  
 man  
 ag-  
 ing  
 FU-  
 JITS  
 PRI  
 BX  
 S4

JITSU PRIMERGY servers, and above servers.

tem.

or  
 RX  
 S8  
 gen-  
 er-  
 a-  
 tion  
 of  
 FU-  
  
**class** i  
 Base  
*irc*  
*dri*  
*gen*  
*Gen*  
  
 iRM  
 hard  
 ware  
 type  
  
 iRM  
 hard  
 ware  
 type  
 is  
 tar-  
 gete  
 for  
 FU-  
 JITS  
 PRI  
 serv  
 whic  
 have  
 iRM  
 S4  
 man  
 age-  
 men  
 sys-  
  
**propert**  
 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 i

Base  
*ironic*  
*drivers*  
*generic*  
*Generic*  
Red  
fish  
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

pow  
 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.snmp module

SNM  
 hard  
 ware  
 type

**class** i  
 Base  
*irc*  
*dri*  
*gen*  
*Gen*

SNM  
Harc  
waro  
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** i  
Base  
*irc*  
*dri*  
*bas*  
*Ven*  
Wra  
per  
arou  
mul-  
ti-  
ple  
Ven-  
dor-  
In-  
ter-  
face

**get\_pro**  
Re-

turn  
 the  
 prop  
 er-  
 ties  
 from  
 all  
 the  
 Ven-  
 dor-  
 In-  
 ter-  
 face

Returns

a  
 dic-  
 tio-  
 nary  
 of  
 <pro  
 erty.  
 en-  
 tries

validation

Call  
 val-  
 i-  
 date  
 on  
 the  
 ap-  
 pro-  
 pri-  
 ate  
 in-  
 ter-  
 face  
 only

Raises

Un-  
 sup-  
 port  
 ed-  
 Driv  
 ten-  
 sion  
 if  
 meth  
 can  
 not

faces.

be  
 map  
 to  
 the  
 sup-  
 port  
 in-  
 ter-

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-  
 ram-  
 e-  
 ters  
 in  
 kwa

ironic.  
 Add  
 ca-  
 pa-  
 bil-  
 ity  
 to  
 node  
 ca-

pa-  
 bil-  
 i-  
 ties  
 prop  
 erty.  
 If  
 ca-  
 pa-  
 bil-  
 ity  
 is  
 al-  
 read  
 pres  
 then  
 a  
 du-  
 pli-  
 cate  
 en-  
 try  
 will  
 be  
 adde

Paramet

- **task**  
 Task  
 ob-  
 ject.
- **capacity**  
 Ca-  
 pa-  
 bil-  
 ity  
 key.
- **value**  
 Ca-  
 pa-  
 bil-  
 ity  
 valu

ironic.  
 Pars

the  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 strin  
 into  
 a  
 dic-  
 tio-  
 nary

Paramet

cap  
 the  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 of  
 the  
 node  
 as  
 a  
 for-  
 mat-  
 ted  
 strin

Raises

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 ca-  
 pa-  
 bil-  
 i-  
 ties  
 is  
 not  
 an  
 strin  
 or  
 has

a malformed value

ramdisk to collect the logs and store it according to the configured storage backend.

ironic.  
 Col-  
 lect  
 and  
 store  
 the  
 sys-  
 tem  
 logs  
 from  
 the  
 IPA  
 ramdisk.  
 Col-  
 lect  
 and  
 store  
 the  
 sys-  
 tem  
 logs  
 from  
 the  
 IPA  
 ramdisk.  
 This  
 meth-  
 od  
 mak-  
 es  
 a  
 call  
 to  
 the  
 IPA

Parameters

- **node**  
 A  
 node  
 ob-  
 ject.
- **label**  
 A  
 string  
 to  
 la-  
 bel



the  
 log  
 file  
 such  
 as  
 a  
 clea  
 step  
 nam

ironic.  
 En-  
 sure  
 boot  
 from  
 cor-  
 rect  
 de-  
 vice  
 if  
 per-  
 sis-  
 tent  
 is  
 True

If  
 ipmi  
 is  
 True  
 and  
 is\_n  
 set  
 to  
 boot  
 from  
 cor-  
 rect  
 de-  
 vice  
 else  
 un-  
 set  
 is\_n  
 field

Paramet

- **tas**  
 Nod  
 ob-

sistent to False, else set is\_next\_boot\_persistent to False.

ject.

- **dri**
  
 Nod
   
 drive

ironic.
   
 Set
   
 per-
   
 sis-
   
 tent
   
 boot
   
 de-
   
 vice
   
 to
   
 drive

If
   
 per-
   
 sis-
   
 tent
   
 is
   
 True
   
 set
   
 per-
   
 sis-
   
 tent
   
 field
   
 to
   
 the
   
 boot
   
 de-
   
 vice
   
 and
   
 re-
   
 set
   
 per-

**Paramet**

- **tas**
  
 Task
   
 ob-
   
 ject.
- **dev**
  
 Boo
   
 de-
   
 vice

- performance**  
 When the next boot is performed, the system is not.

`ionic.`  
 Return the capability value from the node's capabilities information property.

**Parameters**

- node**  
 Node object.
- capability**  
 Capability key.

**Returns**

Capability value.  
 If

ca-  
 pa-  
 bil-  
 ity  
 is  
 not  
 pres  
 then  
 re-  
 turn  
 Non

ironic.  
 Get  
 all  
 MA  
 ad-  
 dres  
 for  
 the  
 port  
 be-  
 long  
 ing  
 to  
 this  
 task  
 node

**Paramet**

**tas**  
 a  
 Task  
 ager  
 in-  
 stan  
 con-  
 tain-  
 ing  
 the  
 node  
 to  
 act  
 on.

**Returns**

A  
 list  
 of  
 MA  
 ad-  
 dres  
 in

the  
 for-  
 mat  
 xx:x

ironic  
 Con  
 struc  
 the  
 log  
 file  
 nam

Paramet

- **node**  
 A  
 node  
 ob-  
 ject.
- **label**  
 A  
 strin  
 to  
 la-  
 bel  
 the  
 log  
 file  
 such  
 as  
 a  
 clea  
 step  
 nam

Returns

The  
 log  
 file  
 nam

ironic.  
 Re-  
 mov  
 -

and  
 :  
 char

ac-  
ters  
and  
low-  
er-  
case  
the  
MA  
strin

Paramet

mac  
MA  
ad-  
dres  
to  
nor-  
mal-  
ize.

Returns

Nor-  
mal-  
ized  
MA  
ad-  
dres  
strin

ironic.

Stor  
the  
rame  
logs

This  
meth  
store  
the  
rame  
logs  
ac-  
cord  
ing  
to  
the  
con-  
fig-  
ured  
stor-  
age  
back  
end.

## Parameters

- **node**  
A node object.
- **log**  
A gzip and base encoded string containing the logs archive.
- **label**  
A string to label the log file such as a clear step name.

## Raises

OS-Error if the directory to

save  
 the  
 logs  
 can-  
 not  
 be  
 cre-  
 ated

**Raises**

IO-  
 Er-  
 ror  
 whe  
 the  
 logs  
 cant  
 be  
 save  
 to  
 the  
 lo-  
 cal  
 file  
 sys-  
 tem.

**Raises**

Swi  
 Op-  
 er-  
 a-  
 tionl  
 if  
 any  
 op-  
 er-  
 a-  
 tion  
 with  
 Swi  
 fails



## ironic.drivers.xclarity module

XCl  
ity  
Driv  
and  
sup-  
port  
ing  
meta  
class

**class** i  
Base  
*irc*  
*dri*  
*gen*  
*Gen*

XCl  
ity  
hard  
ware  
type

**property**  
List  
of  
sup-  
port  
man  
age-  
men  
in-  
ter-  
face

**property**  
List  
of  
sup-  
port  
pow  
in-  
ter-  
face

Module contents

ironic.objects package

Submodules

ironic.objects.allocation module

**class** i
 Base
 *irc*
*obj*
*bas*
*Irc*
 osl
 bas
 Ver

**VERSION**

**property**

**property**

**create**
 Cre-
 ate
 a
 Al-
 lo-
 ca-
 tion
 reco
 in
 the
 DB.

**Parame**
**con**
 Se-
 cu-
 rity
 con-
 text.
 NOT
 This
 shou
 only

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

#### Raises

Al-  
lo-  
ca-  
tion  
pli-  
cate  
Nam  
Al-  
lo-  
ca-  
tion  
Al-  
read  
ists

#### property

#### dbapi =

#### destroy

Dele  
the  
Al-  
lo-  
ca-  
tion  
from  
the  
DB.

#### Parameter

con  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

Raises

AllocationException  
 NotFoundException  
 Four

properties

fields

classmethods

Find  
 an  
 allocation  
 by  
 its  
 ID,  
 UUID  
 or  
 name

Parameters

- **all**  
 The  
 ID,  
 UUID  
 or  
 name  
 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

- **name**  
The  
name  
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  
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.

•

**fil**  
Fil-  
ters  
to  
ap-  
ply.

•

**lim**  
Max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.

•

**mar**  
Pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

•

**sor**  
Col-  
umn  
to  
sort  
re-  
sults  
by.

•

**sort**  
 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**

Load  
 up-  
 date  
 for  
 this  
 Al-  
 lo-  
 ca-  
 tion.

Load  
 an  
 al-  
 lo-  
 ca-  
 tion  
 with  
 the

Updates are applied from the loaded allocation column by column, if there are any updates.

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

same  
uuid  
from  
the  
data  
and  
check  
for  
up-  
date  
at-  
tribu

#### Parameters

con  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

#### Raises

Al-  
lo-  
ca-  
tion-  
Not-  
Foun

#### properties

#### save (context)

Save  
up-  
date  
to

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Allocation(context)

this  
 Al-  
 lo-  
 ca-  
 tion.  
 Up-  
 date  
 will  
 be  
 mad  
 col-  
 umn  
 by  
 col-  
 umn  
 base  
 on  
 the  
 re-  
 sult  
 of  
 self.  
 Parame  
 con  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-  
 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*

No-  
 ti-  
 fi-  
 ca-  
 tion  
 whe  
 iron  
 cre-  
 ates.  
 up-  
 date  
 or  
 dele  
 an  
 al-  
 lo-  
 ca-  
 tion.

**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

property

property

ionic.objects.base module

Iron  
com  
mon  
in-  
ter-  
nal  
ob-  
ject  
mod

**class** i  
Base  
osl  
bas  
Ver  
  
Base  
class  
and  
ob-  
ject  
fac-  
tory.  
  
This  
form  
the  
base  
of  
all  
ob-  
jects  
that  
can  
be  
re-  
mote  
or

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.

in-  
stan-  
ti-  
ated  
via  
RPC

**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-  
vert  
this  
ob-  
ject  
to  
the  
tar-  
get  
ver-  
sion

Con-  
vert  
the



older, or newer than the version of the object. This is used for DB interactions as well as for serialization/deserialization.

ob-  
ject  
to  
the  
tar-  
get  
ver-  
sion  
The  
tar-  
get  
ver-  
sion  
may  
be  
the  
sam

The  
re-  
mov  
flag  
is  
used  
to  
dis-  
tin-  
guis  
thes  
two  
case

- 1) For  
se-  
ri-  
al-  
iza-  
tion  
we  
need  
to  
re-  
mov  
the  
un-  
avai  
able  
field  
be-

service receiving the object may not know about these fields. `remove_unavailable_fields` is set to `True` in this case.

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.

caus  
the

2)  
For  
DB  
in-  
ter-  
ac-  
tions  
we  
need  
to  
set  
the  
un-  
avai-  
able  
field  
to  
their  
ap-  
pro-

\_con  
vert  
does  
the  
ac-  
tual  
worl

#### Parame

- **tar**  
the  
de-  
sired  
ver-  
sion  
of  
the  
ob-  
ject
- **rem**  
True

to True when (de)serializing. False to set the unavailable fields to appropriate values; set this to False for DB interactions.

to  
re-  
mov  
field  
that  
are  
un-  
avai  
able  
in  
the  
tar-  
get  
ver-  
sion  
set  
this

**do\_vers**  
Cha  
the  
ob-  
ject  
to  
the  
ver-  
sion  
need  
for  
the  
data  
  
If  
need  
this  
chan  
the  
ob-  
ject  
(mo  
i-  
fies  
ob-  
ject  
field  
to  
be  
in  
the  
cor-

sion for saving to the database.

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.

rect  
ver-

The  
ver-  
sion  
used  
to  
save  
the  
ob-  
ject  
in  
the  
DB  
is  
de-  
ter-  
min-  
as  
fol-  
lows

- If  
the  
ob-  
ject  
is  
pinn  
we  
save  
the  
ob-  
ject  
in  
the  
pinn  
ver-  
sion  
Sinc  
it  
is

- If  
the  
ob-  
ject  
isnt

must only be called just before saving the object to the DB.

pin  
we  
save  
the  
ob-  
ject  
in  
the  
lat-  
est  
ver-  
sion  
  
Be-  
caus  
the  
ob-  
ject  
may  
be  
con-  
verte  
to  
a  
dif-  
fer-  
ent  
ob-  
ject  
ver-  
sion  
this  
meth

## Returns

a  
dic-  
tio-  
nary  
of  
char  
field  
and  
their  
new  
val-  
ues  
(cou  
be  
an  
emp

These are the fields/values of the object that would be saved to the DB.

wire via RPC or saved in the DB.

dic-  
 tio-  
 nary  
  
**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-  
 u-  
 latec  
 e.g.  
 sent  
 over  
 the  
  
**Returns**  
 if  
 pinn  
 re-  
 turn  
 the  
 ver-  
 sion  
 of  
 this

otherwise, returns the version of the object.

column by column in comparison with the current object.

ob-  
 ject  
 cor-  
 re-  
 spon-  
 ing  
 to  
 the  
 pin.  
 Oth-  
  
**Raises**  
 ovo\_  
**obj\_ref**  
 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-  
 plied  
 from  
 the  
 load  
 ob-  
 ject  
  
**classme**  
 Re-

not be the latest version during an upgrade, when object versions are pinned.

turn  
 whe  
 this  
 ob-  
 ject  
 sup-  
 port  
 a  
 par-  
 tic-  
 u-  
 lar  
 ver-  
 sion  
  
 Che  
 the  
 re-  
 ques  
 ver-  
 sion  
 agai  
 the  
 ob-  
 jects  
 tar-  
 get  
 ver-  
 sion  
 The  
 tar-  
 get  
 ver-  
 sion  
 may  
  
**Parame**  
**ver**  
 A  
 tu-  
 ple  
 rep-  
 re-  
 sent  
 ing  
 the  
 ver-  
 sion  
 to  
 chec

**Returns**



When  
the  
ver-  
sion  
is  
sup-  
ported

**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  
JSO  
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  
the  
en-  
tity.  
This  
se-  
ri-  
al-  
izes  
the  
en-  
tity  
so  
that  
it  
can  
be  
sent  
over  
e.g.  
RPC  
A  
se-  
ri-

alized entity for an IroniCObject is a dictionary with keys: `ironic_object.namespace`, `ironic_object.data`, `ironic_object.name`, `ironic_object.version`, and `ironic_object.changes`.

We  
as-  
sum  
that  
the  
clien  
(iron  
API  
is  
al-  
way

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.

these objects are always in the latest versions.)

talk-  
ing  
to  
a  
serv  
(iron  
conc  
that  
is

(In-  
ter-  
nally  
the  
ser-  
vice  
deal  
with  
the  
lat-  
est  
ver-  
sion  
of  
ob-  
jects  
so  
we  
know  
that

## Parameters

- **con**  
se-  
cu-  
rity  
con-  
text
- **ent**  
the  
en-  
tity  
to  
be

se-ri-al-ized may be an Iron i-cOb-ject

Returns

the se-ri-al-ized en-tity

Raises

ovo\_ (via .get\_

ironic. Re-turn the max i-mun-ver-sion in the list.

Paramet

ver a list of (stri-ver-sion as-sum to have at least

one  
 en-  
 try

**Returns**  
 the  
 max  
 i-  
 mun  
 ver-  
 sion  
 (stri

ironic.objects.bios module

**class** i  
 Base  
*irc*  
*obj*  
*bas*  
*Irc*

**VERSION**

**create**  
 Cre-  
 ate  
 a  
 BIO  
 Set-  
 ting  
 reco  
 in  
 DB.

**Parame**  
**con**  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

nally  
by  
the  
in-  
di-

#### **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  
ex-  
ists.

**property**

**dbapi =**

#### **classme**

Dele  
a  
BIO  
Set-  
ting  
base  
on  
its  
node  
and

nam

**Parameters**

- 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-  
 Four  
 if  
 the  
 bios  
 set-  
 ting  
 nam  
 is

not  
 found

**fields**

**classme**

Get  
 a  
 BIO  
 Set-  
 ting  
 base  
 on  
 its  
 node  
 and  
 nam

**Parame**

- con**  
 Se-  
 cu-  
 rity  
 con-  
 text.
- nod**  
 The  
 node  
 id.
- nam**  
 BIO  
 set-  
 ting  
 nam  
 to  
 be  
 re-  
 triev

**Raises**

Nod  
 Not-  
 Four  
 if  
 the  
 node  
 id



is  
 not  
 found

**Raises**

BIO  
 Set-  
 ting  
 Not-  
 found  
 if  
 the  
 bios  
 set-  
 ting  
 name  
 is  
 not  
 found

**Returns**

A  
 :class:  
 ob-  
 ject.

**property**

**property**

**save** (*code*

Save  
 BIO  
 Set-  
 ting  
 up-  
 date  
 in  
 DB.

**Parameter**

**con**  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

Raises

NodeNotFound if the node id is not found

Raises

BIOSSettingNotFound if the bios setting name is not found

properties

properties

class IronicBase

IronicBase object base class

*IroniC*

## 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-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

- **nod**  
The  
node  
id.

- **set**  
A

list  
 of  
 bios  
 set-  
 ting

**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  
 al-  
 read  
 ex-  
 ists.

**Returns**

A  
 list  
 of  
 BIO  
 Set-  
 ting  
 ob-  
 jects

property

dbapi =

classme

Dele
 BIO
 Set-
 ting
 base
 on
 node
 and
 nam

Parame

- **con**  
 Se-  
 cu-  
 rity  
 con-  
 text.
- **nod**  
 The  
 node  
 id.
- **nam**  
 List  
 of  
 BIO  
 set-  
 ting  
 nam  
 to  
 be  
 dele

Raises

Nod  
 Not-  
 Foun  
 if  
 the  
 node  
 id  
 is  
 not  
 foun

Raises

BIO  
 Set-

ting.  
 Not-  
 Four  
 if  
 any  
 of  
 BIO  
 set-  
 ting  
 fails  
 to  
 dele

**fields**

**classme**

Get  
 BIO  
 Set-  
 ting  
 base  
 on  
 node

**Parame**

- **con**
  
 Se-  
 cu-  
 rity  
 con-  
 text.
- **nod**
  
 The  
 node  
 id.

**Raises**

Nod  
 Not-  
 Four  
 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-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: BIOSSetting(context)

- **node**  
The

node  
id.

- **set**  
A  
list  
of  
bios  
set-  
tings

**Raises**  
Nod  
Not-  
Four  
if  
the  
node  
id  
is  
not  
foun

**Raises**  
BIO  
Set-  
ting-  
Not-  
Four  
if  
any  
of  
the  
bios  
set-  
ting  
nam  
is  
not  
foun

**Returns**  
A  
list  
of  
BIO  
Set-  
ting  
ob-  
jects

**classme**  
Re-



turn  
 lists  
 of  
 cre-  
 ate/  
 set-  
 tings

This  
 meth  
 sync  
 with  
 bios  
 data  
 ta-  
 ble  
 and  
 sorts  
 out  
 four  
 lists  
 of  
 cre-  
 ate/  
 set-  
 tings

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 up-date dele and un-char

property

ironic.objects.chassis module

class iBase  
     *ironic.objects.chassis.IronicOSList*  
     *base*  
     *Version*

VERSION

create  
 Cre-ate a Chassis record in the DB.  
 Colu-wise

it will be checked against the in-database copy of the chassis before updates are made.

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

up-  
 date  
 will  
 be  
 mad  
 base  
 on  
 the  
 re-  
 sult  
 of  
 self.  
 If  
 tar-  
 get\_  
 is  
 pro-  
 vide

**Parame**  
**con**  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-

**propert**

**dbapi =**

**propert**

**destroy**  
 Dele

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

the  
 Cha  
 sis  
 from  
 the  
 DB.  
 Param  
 con  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-  
 propert  
 fields  
 classme  
 Find  
 a  
 chas  
 sis  
 base  
 on  
 its  
 id  
 or  
 uuid  
 and  
 re-  
 turn  
 a  
 Cha  
 sis

ob-  
ject.

Paramete

- **con**  
Se-  
cu-  
rity  
con-  
text
- **cha**  
the  
id  
*or*  
uuid  
of  
a  
chas  
sis.

Returns

a  
*Cha*  
ob-  
ject.

classme

Find  
a  
chas  
sis  
base  
on  
its  
in-  
te-  
ger  
ID  
and  
re-  
turn  
a  
Cha  
sis  
ob-  
ject.

Paramete

- **cls**  
the  
*Cha*
- **con**  
Se-  
cu-  
rity  
con-  
text
- **cha**  
the  
ID  
of  
a  
chas  
sis.

**Returns**

a  
*Cha*  
ob-  
ject.

**classme**

Find  
a  
chas  
sis  
base  
on  
UI  
and  
re-  
turn  
a  
*Cha*  
ob-  
ject.

**Parame**

- **cls**  
the  
*Cha*
- **con**  
Se-

cu-  
rity  
con-  
text

- **uui**  
the  
UUI  
of  
a  
chas  
sis.

#### Returns

a  
*Cha*  
ob-  
ject.

#### property

#### classme

Re-  
turn  
a  
list  
of  
Cha  
sis  
ob-  
jects

#### Parame

- **cls**  
the  
*Cha*

- **con**  
Se-  
cu-  
rity  
con-  
text.

- **lim**  
max  
i-

mun  
 num  
 ber  
 of  
 re-  
 sour  
 to  
 re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult.

- **mar**  
 pag-  
 i-  
 na-  
 tion  
 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
 same
 uuid
 from
 the
 data
 and
 chec
 for
 up-
 date
 at-
 tribu
 Up-
 date
 are

applied from the loaded chassis column by column, if there are any updates.

Param

con
 Se-
 cu-
 rity
 con-
 text.
 NOT
 This
 shou
 only
 be
 used
 in-
 ter-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

save (co
 Save
 up-
 date
 to
 this
 Cha
 sis.
 Up-
 date
 will
 be
 mad
 col-
 umn
 by
 col-
 umn
 base
 on
 the
 re-
 sult
 of
 self.

Parame
 con
 Se-
 cu-
 rity
 con-
 text.
 NOT
 This
 shou
 only
 be
 used
 in-
 ter-
 nally
 by
 the

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Chassis(context)

in-  
di-

property

property

class i

Base  
*iro*  
*obj*  
*not*  
*Not*

No-  
ti-  
fi-  
ca-  
tion  
emit  
ted  
whe  
iron  
cre-  
ates.  
up-  
date  
dele  
a  
chas  
sis.

VERSION

property

property

fields

property

property

property

property

class i
Base
irc
obj
not
Not

SCHEMA

VERSION

property

property

property

fields

property

property

ionic.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
-

online field is ignored if this value is set to None.

onl  
 Spec  
 ify  
 the  
 ex-  
 pect  
 onl  
 field  
 valu  
 for  
 the  
 con-  
 duc-  
 tor  
 to  
 be  
 re-  
 triev  
 The

**Returns**  
 a  
*Con*  
 ob-  
 ject.

**property**

**property**

**classme**

Re-  
 turn  
 a  
 list  
 of  
 Con  
 duc-  
 tor  
 ob-  
 jects

**Parame**

- **cls**  
 the  
*Con*
-

con
 Se-
 cu-
 rity
 con-
 text.

- **lim**
 max
 i-
 mun
 num
 ber
 of
 re-
 sour
 to
 re-
 turn
 in
 a
 sin-
 gle
 re-
 sult.

- **mar**
 pag-
 i-
 na-
 tion
 marl
 for
 large
 data
 sets.

- **son**
 col-
 umn
 to
 sort
 re-
 sults
 by.

- **son**
 di-
 rec-

applied from the loaded chassis column by column, if there are any updates.

tion  
 to  
 sort.  
 asc  
 or  
 desc  
**Returns**  
 a  
 list  
 of  
*Con*  
 ob-  
 ject.  
**refresh**  
 Load  
 and  
 ap-  
 plies  
 up-  
 date  
 for  
 this  
 Con  
 duc-  
 tor.  
 Load  
 a  
*Con*  
 with  
 the  
 sam  
 uuid  
 from  
 the  
 data  
 and  
 chec  
 for  
 up-  
 date  
 at-  
 tribu  
 Up-  
 date  
 are  
**Parame**  
**con**  
 Se-



rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Conductor(context)

cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

#### 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-  
 is-  
 tra-  
 tion  
 will  
 raise  
 an  
 ex-  
 cep-

line record is found. When true, will overwrite the existing record. Default: False.

tion  
 whe  
 a  
 con-  
 flict.  
 ing  
 on-

Raises

Con  
 duc-  
 torA  
 read  
 is-  
 terec

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-  
 try  
 shou

containing `hardware_type`, `interface_type`, `interface_name` and `default`, e.g. `{hardware_type: hardware-type, interface_type: deploy, interface_name: iscsi, default: True}`

be  
a  
dic-  
tio-  
nary

**save** (*co*  
Save  
is  
not  
sup-  
port  
by  
Con  
duc-  
tor  
ob-  
jects

**touch** (*d*  
Touc  
this  
con-  
duc-  
tors  
DB  
reco  
marl  
ing  
it  
as  
up-  
to-  
date

**unregist**  
Re-  
mov  
this  
con-  
duc-  
tor  
from  
the  
ser-  
vice  
reg-  
istry

**unregist**  
Un-

reg-  
 is-  
 ter  
 all  
 hard  
 ware  
 in-  
 ter-  
 face  
 for  
 this  
 con-  
 duc-  
 tor.

**property**

ironic.objects.deploy\_template module

**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

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

Raises

DeployTemplateDuplicateNameError  
 if a deployment template with the same name exists.

Raises

DeployTemplateReadError  
 if a deployment template with

the  
sam  
UUI  
ex-  
ists.

**property**

**dbapi =**

**destroy**

Dele  
the  
De-  
ploy  
plate  
from  
the  
DB.

**Parame**

**con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

**Raises**

De-  
ploy  
plate  
Four  
if  
the  
de-  
ploy

tem-  
plate  
no  
long  
ap-  
pear  
in  
the  
data

**property**

**fields**

**classme**

Find  
a  
de-  
ploy  
tem-  
plate  
base  
on  
its  
in-  
te-  
ger  
ID.

**Parame**

•

**con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-



rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

•

**tem**  
The  
ID  
of  
a  
de-  
ploy  
tem-  
plate

#### Raises

De-  
ploy  
plate  
Four  
if  
the  
de-  
ploy  
tem-  
plate  
no  
long  
ap-  
pear  
in  
the  
data

#### Returns

a  
*Dep*  
ob-  
ject.

#### classme

Find  
a  
de-  
ploy  
tem-  
plate  
base  
on  
its  
nam

#### Parame

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

•  
 con  
 se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-  
 •  
 nam  
 The  
 nam  
 of  
 a  
 de-  
 ploy  
 tem-  
 plate  
 Raises  
 De-  
 ploy  
 plate  
 Four  
 if  
 the  
 de-  
 ploy  
 tem-  
 plate  
 no  
 long  
 ap-  
 pear  
 in  
 the  
 data

## Returns

a  
*Dep*  
 ob-  
 ject.

## classme

Find  
 a  
 de-  
 ploy  
 tem-  
 plate  
 base  
 on  
 its  
 UUI

## Parame

- **con**  
 se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

- **uui**  
 The  
 UUI  
 of  
 a  
 de-  
 ploy  
 tem-

plate

**Raises**

De-  
 ploy  
 plate  
 Four  
 if  
 the  
 de-  
 ploy  
 tem-  
 plate  
 no  
 long  
 ap-  
 pear  
 in  
 the  
 data

**Returns**

a  
*Dep*  
 ob-  
 ject.

**property**

**classmethod**

Re-  
 turn  
 a  
 list  
 of  
 De-  
 ploy  
 plate  
 ob-  
 jects

**Parameters**

- **con**  
 se-  
 cu-  
 rity  
 con-  
 text.  
 NOT

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context).

This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

- **lim**  
max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.

- **mar**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

- **son**  
col-  
umn  
to  
sort

re-  
sults  
by.

- **sort**  
di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc

**Returns**

a  
list  
of  
*Dep*  
ob-  
jects

**classme**

Re-  
turn  
a  
list  
of  
De-  
ploy  
plate  
ob-  
jects  
match-  
ing  
a  
set  
of  
nam

**Parame**

- **con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou

reception\_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)`.

only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

- **name**  
a  
list  
of  
names  
to  
fil-  
ter  
by.

**Returns**  
a  
list  
of  
*Deploy*  
ob-  
jects

**properties**

**refresh**  
Load  
up-  
date  
for  
this  
de-  
ploy  
tem-  
plate  
  
Load  
a  
de-  
ploy  
tem-  
plate  
with  
the

Updates are applied from the loaded template column by column, if there are any updates.

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

same  
uuid  
from  
the  
data  
and  
check  
for  
up-  
date  
at-  
tribu

#### **Parame**

**con**  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

#### **Raises**

De-  
ploy  
plate  
Four  
if  
the  
de-  
ploy  
tem-  
plate  
no  
long  
ap-  
pear



in  
the  
data

**save** (*co*  
Save  
up-  
date  
to  
this  
De-  
ploy  
plate  
Colu  
wise  
up-  
date  
will  
be  
mad  
base  
on  
the  
re-  
sult  
of  
self.

**Parame**  
**con**  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: DeployTemplate(context)

**Raises**

De-  
 ploy  
 plat-  
 eDu  
 pli-  
 cate  
 Nam  
 if  
 a  
 de-  
 ploy  
 tem-  
 plate  
 with  
 the  
 sam  
 nam  
 ex-  
 ists.

Raises

De-  
 ploy  
 plate  
 Four  
 if  
 the  
 de-  
 ploy  
 tem-  
 plate  
 does  
 not  
 ex-  
 ist.

property

property

property

class i

Base  
 irc  
 obj  
 not  
 Not  
 No-

ti-  
 fi-  
 ca-  
 tion  
 emit  
 ted  
 on  
 de-  
 ploy  
 tem-  
 plate  
 API  
 op-  
 er-  
 a-  
 tions

VERSION

property

property

fields

property

property

property

property

class i

Base  
*irc*  
*obj*  
*not*  
*Not*

SCHEMA

VERSION

proper

proper

fields

proper

proper

proper

proper

## ironic.objects.deployment module

```
class i
```

Base

*irc*

obj

bas

*Irc*

osl

bas

Ver

**VERSION**

**create**

Cre-

ate

a

De-

ploy

men

Up-

date

the

cor-

re-

spor

ing

node

un-

der

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Deployment(context)

the  
hood

**Paramete**

- **con**  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-
- **nod**  
Nod  
ob-  
ject  
for  
de-  
ploy  
men

**Raises**

In-  
stan  
As-  
so-  
ci-  
ated  
Nod  
As-  
so-  
ci-  
ated  
Nod  
Not-

Four  
**property**

**dbapi** =

**destroy**  
 Dele  
 the  
 De-  
 ploy  
 men  
 Up-  
 date  
 the  
 cor-  
 re-  
 spor  
 ing  
 node  
 un-  
 der  
 the  
 hood

**Param**

- con**  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

**nod**  
Nod  
ob-  
ject  
for  
de-  
ploy  
men

#### **fields**

**classme**  
Find  
a  
de-  
ploy  
men  
base  
by  
its  
node  
UUI

#### **Parame**

- **cls**  
the  
*Dep*
- **con**  
Se-  
cu-  
rity  
con-  
text
- **nod**  
The  
UUI  
of  
a  
cor-  
re-  
spor  
ing  
node

**Returns**  
An  
*Dep*

ob-  
ject.

**Raises**

Nod  
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



property

property

instance

instance

property

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  
 sin-  
 gle  
 re-  
 sult.

- **mark**  
 Pag-  
 i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets.

- **sort**  
 Col-  
 umn  
 to  
 sort  
 re-  
 sults  
 by.

- **sort**  
 Di-  
 rec-  
 tion  
 to  
 sort.  
 asc  
 or  
 desc

**Returns**  
 A  
 list  
 of

*Dep*  
 ob-  
 ject.

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu

**node\_ma**

**propert**

**propert**

**refresh**  
 Re-  
 fresl  
 the  
 ob-  
 ject  
 by  
 re-  
 fetch  
 from  
 the  
 DB.

**Parame**  
**con**  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

ionic.objects.fields module

if this is not possible.

co-  
erce  
(if  
pos-  
si-  
ble)  
a  
valu  
on  
as-  
sign  
men  
  
This  
meth  
shou  
con-  
vert  
the  
valu  
give  
into  
the  
des-  
ig-  
nate  
type  
or  
thro  
an  
ex-  
cep-  
tion

**Param:**  
The  
Ver-  
sion  
dOb  
ject  
on  
whic  
an  
at-  
tribu  
is  
be-  
ing  
set

**Param::**  
The

nam  
 of  
 the  
 at-  
 tribu  
 be-  
 ing  
 set

**Param:**  
 The  
 valu  
 be-  
 ing  
 set

**Returns**  
 A  
 prop  
 type  
 valu

**class i**  
 Base  
 osl  
 fie  
 Aut

**AUTO\_TY**

**class i**  
 Base  
 osl  
 fie  
 Int

**class i**  
 Base  
 osl  
 fie  
 Aut

**AUTO\_TY**

**class i**  
 Base  
 osl  
 fie  
 Lis

**class i**

if this is not possible.

Param:

The  
Ver-  
sion  
dOb-  
ject  
on  
whic  
an  
at-  
tribu  
is  
be-  
ing  
set

**Param::**

The  
nam  
of  
the  
at-  
tribu  
be-  
ing  
set

**Param:**

The  
valu  
be-  
ing  
set

**Returns**

A  
prop  
type  
valu

**class i**

Base  
osl  
fie  
Aut

**AUTO\_TY**

**class i**

Base  
osl  
fie  
Enu



**ALL =**

**CRITICAL**

**DEBUG =**

**ERROR =**

**INFO =**

**WARNING**

```
class i
    Base
    osl
    fie
    Bas
```

**AUTO\_TY**

```
class i
    Base
    osl
    fie
    Enu
```

**ALL =**

**END =**

**ERROR =**

**START =**

**SUCCESS**

```
class i
    Base
    osl
    fie
    Bas
```

AUTO\_TYP

class i

Base
 osl
 fie
 Obj

class i

Base
 osl
 fie
 Str

static

This
 is
 calle
 to
 co-
 erce
 (if
 pos-
 si-
 ble)
 a
 valu
 on
 as-
 sign
 men

This
 meth
 shou
 con-
 vert
 the
 valu
 give
 into
 the
 des-
 ig-
 nate
 type
 or
 thro
 an
 ex-

if this is not possible.

cep-  
tion

**Param:**  
The  
Ver-  
sion  
dOb  
ject  
on  
whic  
an  
at-  
tribu  
is  
be-  
ing  
set

**Param::**  
The  
nam  
of  
the  
at-  
tribu  
be-  
ing  
set

**Param:**  
The  
valu  
be-  
ing  
set

**Returns**  
A  
prop  
type  
valu

**class i**  
Base  
osl  
fie  
Str

**class i**  
Base  
osl  
fie

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

AUTO\_TYP

class i
 Base
 osl
 fie
 UUI

## 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
    im-
    ple-
    men
    ing
```

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-  
 form  
 the  
 ac-  
 tion

- **obj**  
 The  
 ob-  
 ject  
 in-  
 stan-  
 on  
 whic  
 to  
 per-  
 form  
 the  
 ac-  
 tion

- **obj**  
 The  
 nam  
 of  
 the  
 ac-  
 tion  
 meth  
 to  
 call

- **arg**  
 The  
 po-  
 si-  
 tion:  
 ar-  
 gu-  
 men  
 to  
 the  
 ac-  
 tion  
 meth

•

**kwargs**  
The  
key-  
word  
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-  
stead  
of  
pro-  
vid-

target version for the toplevel object and relying on the service-side mapping to handle sub-objects, this sends a mapping of all the dependent objects and their client-supported versions. The server will backport objects within the tree starting at `objinst` to the versions specified in `object_versions`, removing objects that have no entry. Use `obj_tree_get_versions()` to generate this mapping.

plementedError if you dont implement it. For backports, this method will be tried first, and if unimplemented, will fall back to `object_backport()`.

ing  
a  
spe-  
cific

NOT  
This  
was  
not  
in  
the  
ini-  
tial  
spec  
for  
this  
in-  
ter-  
face  
so  
the  
base  
class  
raise  
NotI

## Parameters

- **context**  
The context with which to perform the backport.
- **objinst**  
An instance of the object to be backported.



stan
 of
 a
 Ver-
 sion
 dOb
 ject
 to
 be
 back
 port

- **obj**  
 A  
 dict  
 of  
 { ob-  
 j-  
 nam  
 ver-  
 sion  
 map  
 ping

**object\_**  
 Dep  
 re-  
 cate  
 sinc  
 ver-  
 sion  
 0.10

Use  
*obj*  
 in-  
 stea

Per-  
 form  
 an  
 ac-  
 tion  
 on  
 a  
 Ver-  
 sion  
 dOb  
 ject  
 class

Whe  
 in-

this interface), classmethod calls on `remotable_classmethod` methods will cause this to be executed to actually make the desired call. This usually involves performing RPC.

**Parame**

- **con**  
The  
con-  
text  
with  
which  
to  
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  
which  
the  
ac-  
tion  
is  
be-  
ing  
take

- **arg**  
The  
po-  
si-  
tion  
ar-  
gu-  
men  
to  
the  
ac-  
tion  
meth

- **kwa**  
The  
key-  
wor  
ar-  
gu-  
men  
to

implementing VersionedObject class.

the  
 ac-  
 tion  
 meth  
**Returns**  
 The  
 re-  
 sult  
 of  
 the  
 ac-  
 tion  
 meth  
 whic  
 may  
 (or  
 may  
 not)  
 be  
 an  
 in-  
 stan  
 of  
 the

**object\_**  
 Per-  
 form  
 an  
 ac-  
 tion  
 on  
 a  
 Ver-  
 sion  
 dOb  
 ject  
 class  
 Whe  
 in-  
 di-  
 rec-  
 tion  
 is  
 set  
 on  
 a  
 Ver-  
 sion

this interface), classmethod calls on remotable\_classmethod methods will cause this to be executed to actually make the desired call. This usually involves performing RPC.

client-side object versions for easier nested backports. The manifest is the result of calling `obj_tree_get_versions()`.

dOb  
ject  
(to  
a  
class  
im-  
ple-  
men-  
ing

This  
dif-  
fers  
from  
ob-  
ject\_  
in  
that  
it  
is  
pro-  
vide  
with  
ob-  
ject\_  
a  
man  
i-  
fest  
of

NOT  
This  
was  
not  
in  
the  
ini-  
tial  
spec  
for  
this  
in-  
ter-  
face  
so  
the  
base

plementedError if you dont implement it. For backports, this method will be tried first, and if unimplemented, will fall back to `object_class_action()`. New implementations should provide this method instead of `object_class_action()`

Parameters

- **context**  
 The context with which to perform the action
- **obj**  
 The registry name of the object
- **obj**  
 The name of the action method to call
- **obj**  
 A dict of {obj-

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  
in-  
stan

implementing VersionedObject class.

## ironic.objects.node module

of  
the

```
class i
    Base
    irc
    obj
    bas
    Irc
    osl
    bas
    Ver
```

## VERSION

## property

```
as_dict
    Re-
    turn
    the
    ob-
    ject
    rep-
    re-
    sent
    as
    a
    dict.

    The
    re-
    turn
    ob-
    ject
    is
    JSO
    seria
```

## property

## property

## property



property

property

property

property

property

property

**create**

Cre-  
ate  
a  
Nod  
reco  
in  
the  
DB.

Colu  
wise  
up-  
date  
will  
be  
mad  
base  
on  
the  
re-  
sult  
of  
self.  
If  
tar-  
get\_  
is  
pro-  
vide

it will be checked against the in-database copy of the node before updates are made.

**Parame**

con  
Se-  
cu-  
rity

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-

**Raises**  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 som  
 prop  
 erty  
 val-  
 ues  
 are  
 in-  
 valid

**property**

**dbapi =**

**property**

**property**

**property**

**destroy**  
 Dele  
 the  
 Nod

from  
the  
DB.

#### Parameters

**context**  
Security context.  
NOT  
This should only be used internally by the in-di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

#### properties

#### properties

#### properties

#### properties

#### properties

#### fields

#### classmethods

Find  
a  
node  
base  
on  
its  
id  
or  
uuid

and  
 re-  
 turn  
 a  
 Node  
 ob-  
 ject.

Paramete

- **con**
  
 Se-
 cu-
 rity
 con-
 text
- **nod**
  
 the
 id
 *or*
 uuid
 of
 a
 node

Returns

a
   
*Node*
  
 ob-
 ject.

classme

Find
 a
 node
 base
 on
 its
 in-
 te-
 ger
 ID
 and
 re-
 turn
 a
 Node
 ob-
 ject.

#### Parameters

- **cls**  
the  
*Nod*
- **con**  
Se-  
cu-  
rity  
con-  
text
- **nod**  
the  
ID  
of  
a  
node

#### Returns

a  
*Nod*  
ob-  
ject.

#### classmethods

Find  
a  
node  
base  
on  
the  
in-  
stan-  
UI  
and  
re-  
turn  
a  
Nod  
ob-  
ject.

#### Parameters

- **cls**  
the  
*Nod*

- **context**  
Security context

- **ui**  
the UUI of the instance

**Returns**  
a *Node* object.

**classmethod**  
Find a node based on name and return a *Node* object.

**Parameter**

- **cls**  
the *Node*

- **context**  
Security context

-

**name**  
the  
log-  
i-  
cal  
nam  
of  
a  
node

#### Returns

a  
*NoC*  
ob-  
ject.

#### classmethods

Get  
a  
node  
by  
as-  
so-  
ci-  
ated  
port  
ad-  
dres

#### Parameters

- **cls**  
the  
*NoC*
- **con**  
Se-  
cu-  
rity  
con-  
text.
- **add**  
A  
list  
of  
port  
ad-  
dres

#### Raises

Node  
 Not-  
 Four  
 if  
 the  
 node  
 is  
 not  
 found

Returns

a  
*Noop*  
 ob-  
 ject.

classmethod

Find  
 a  
 node  
 base  
 on  
 UUI  
 and  
 re-  
 turn  
 a  
 Node  
 ob-  
 ject.

Parameters

- **cls**  
 the  
*Noop*
- **context**  
 Se-  
 cu-  
 rity  
 con-  
 text
- **uui**  
 the  
 UUI  
 of  
 a  
 node



#### Returns

a  
*Noo*  
ob-  
ject.

property

property

property

property

property

property

property

property

classme

Re-  
turn  
a  
list  
of  
Nod  
ob-  
jects

#### Parame

- **cls**  
the  
*Noo*
- **con**  
Se-  
cu-  
rity  
con-  
text.

- **limit**  
 maximum number of resources to return in a single result.
- **marshal**  
 pagination method for large data sets.
- **sortBy**  
 column to sort results by.
- **sortDirection**  
 direction to sort. asc or desc

•

fil  
 Fil-  
 ters  
 to  
 ap-  
 ply.

**Returns**

a  
 list  
 of  
*Noo*  
 ob-  
 ject.

properti

properti

properti

properti

properti

properti

properti

properti

properti

properti

properti

properti

properti

properti

property

property

refresh

Re-  
fresh  
the  
ob-  
ject  
by  
re-  
fetched  
from  
the  
DB.

Parameter

con-  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

classme

Re-  
lease  
the  
rese  
va-  
tion  
on  
a  
node

Parameter

- **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

**property**

**property**

**classme**  
Get  
and  
re-  
serv

performed, mark it reserved by this host.

a  
 node  
 To  
 pre-  
 vent  
 othe  
 Man  
 ager  
 vice  
 from  
 ma-  
 nip-  
 u-  
 lat-  
 ing  
 the  
 give  
 Nod  
 whil  
 a  
 Task  
 is

Paramet

- **cls**  
 the  
*Noa*
- **con**  
 Se-  
 cu-  
 rity  
 con-  
 text.
- **tag**  
 A  
 strin  
 uniq  
 iden  
 ti-  
 fy-  
 ing  
 the  
 rese  
 va-  
 tion

hold

- **node**  
A  
node  
ID  
or  
UUID

#### **Raises**

Nod  
Not-  
Foun  
if  
the  
node  
is  
not  
foun

#### **Returns**

a  
*Nod*  
ob-  
ject.

**property**

**property**

**property**

**save** (*co*

Save  
up-  
date  
to  
this  
Nod  
  
Colu  
wise  
up-  
date  
will  
be  
mad  
base  
on  
the  
re-

it will be checked against the in-database copy of the node before updates are made.

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Node(context)

sult  
of  
self.  
If  
tar-  
get\_  
is  
pro-  
vide

**Parame**  
**con**  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
som  
prop  
erty  
val-  
ues  
are  
in-  
valid

**propert**



**property**

**property**

**property**

**touch\_p**

Touch  
the  
data  
reco  
to  
marl  
the  
pro-  
vi-  
sion  
ing  
as  
alive

**property**

**property**

**property**

**property**

**class i**

Base  
*irc*  
*obj*  
*not*  
*Not*  
  
No-  
ti-  
fi-  
ca-  
tion  
emit  
ted  
whe  
iron  
cre-  
ates.

up-  
date  
or  
dele  
a  
node

VERSION

propert

propert

fields

propert

propert

propert

propert

class i

Base  
*irc*  
*obj*  
*noc*  
*Noc*

Pay-  
load  
sche  
for  
whe  
iron  
cre-  
ates.  
up-  
date  
or  
dele  
a  
node

SCHEMA

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

fields

property

property

property

proper

property

property

property

property

property

property

property

property

property

class i

Base

*irc*

*obj*

*not*

*Not*

No-

ti-

fi-

ca-

tion

emit

ted

wher

node

con-

sole

state

char

VERSION

property

property

fields

property

property

property

property

class i

Base  
*irc*  
*obj*  
*not*  
*Not*  
  
No-  
ti-  
fi-  
ca-  
tion  
for  
when  
a  
node  
pow  
state  
is  
cor-  
rect  
in  
the  
data  
  
This  
no-  
ti-  
fi-  
ca-  
tion  
is  
emit  
ted  
when  
iron  
de-  
tect  
that

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.

the  
ac-  
tual  
pow  
state  
on

**VERSION**

**property**

**property**

**fields**

**property**

**property**

**property**

**property**

**class i**

Base  
*ironiC*  
*obj*  
*node*  
*Node*

No-  
ti-  
fi-  
ca-  
tion  
pay-  
load  
sche  
for  
whe  
a  
node  
pow  
state  
is

dated.

correct  
 from  
 in-  
 di-  
 cate  
 the  
 pre-  
 vi-  
 ous  
 pow  
 state  
 on  
 the  
 iron  
 node  
 be-  
 fore  
 the  
 node  
 was  
 up-

VERSION

property

property

property

property

property

property

property

property

property

property



property  
property  
property  
fields  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property  
property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class i

Base  
*irc*  
*obj*  
*not*  
*Not*  
No-  
ti-  
fi-  
ca-

tion  
emit  
ted  
whe  
main  
te-  
nanc  
state  
char  
via  
API

**VERSION**

**property**

**property**

**fields**

**property**

**property**

**property**

**property**

**class i**

Base  
*irc*  
*obj*  
*not*  
*Not*

Base  
class  
used  
for  
all  
no-  
ti-  
fi-  
ca-  
tion  
pay-  
load

about  
 a  
 Node  
 ob-  
 ject.

SCHEMA

VERSION

property

property

property

property

property

property

property

property

property

property

property

property

property

fields

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

class i

Base  
*irc*  
*obj*  
*not*  
*Not*  
No-  
ti-  
fi-  
ca-  
tion  
emit  
ted  
whe  
iron  
char  
a  
node  
pow  
state

VERSION

property

property

fields

property

property

property

property

class i

Base

*irc*

*obj*

*noo*

*Noo*

Pay-

load

sche

for

whe

iron

char

a

node

pow

state

VERSION

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

**class** i

Base  
*irc*  
*obj*  
*not*  
*Not*

No-  
 ti-  
 fi-  
 ca-  
 tion  
 emit  
 ted  
 whe  
 iron  
 char  
 a  
 node  
 pro-  
 vi-  
 sion  
 state

**VERSION**

**property**

**property**

**fields**

**property**

**property**

**property**

**property**

**class** i

Base  
*irc*  
*obj*

*noo*

*Noo*

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

propert

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

property

ionic.objects.notification module

tion being taken on the notification, and the status of the action.

**class** `ionic.objects.notification.Notification`  
 Base class for the notification object.  
 Defines the event to be sent on the wire.  
 An Event object must specify the object being acted on, a string describing the action.

**VERSION**

**property**

**property**

**fields**

**property**

**property**

**to\_even**

Con  
struc  
strin  
for  
even  
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*

tionPayloadBase.

Base  
 class  
 for  
 ver-  
 sion  
 no-  
 ti-  
 fi-  
 ca-  
 tion.  
  
 Sub-  
 class  
 mus  
 de-  
 fine  
 the  
 pay-  
 load  
 field  
 whic  
 mus  
 be  
 a  
 sub-  
 class  
 of  
 No-  
 ti-  
 fi-  
 ca-

VERSION

property

emit (*co*  
 Sen  
 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

property

**fields**

**populat**

Pop-  
u-  
late  
the  
ob-  
ject  
base  
on  
the  
SCH  
and  
the  
sour  
ob-  
jects

**Parame**

**kwa**  
A  
dict  
con-  
tains  
the  
sour  
ob-  
ject  
and  
the  
keys  
de-  
finec  
in  
the  
SCH

**Raises**

No-  
ti-  
fi-  
ca-  
tion-  
Sche  
jectl  
ror

**Raises**

No-  
ti-  
fi-

ca-  
 tion-  
 Sched-  
 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.

**VERSION**

proper

**Parameters** **Comments**

Security context. NOT This should only be used internally by the indi-

## Raises

MA  
 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  
 only  
 be  
 used  
 in-  
 ter-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

object.

nally  
by  
the  
in-  
di-

#### **Raises**

Port  
Not-  
Four

#### **property**

#### **fields**

#### **classme**

Find  
a  
port  
  
Find  
a  
port  
base  
on  
its  
id  
or  
uuid  
or  
nam  
or  
MA  
ad-  
dres  
and  
re-  
turn  
a  
Port

#### **Parame**

- **con**  
Se-  
cu-  
rity

con-  
text

- - port**  
the  
id  
*or*  
uuid  
*or*  
nam  
*or*  
MA  
ad-  
dres  
of  
a  
port

**Returns**  
a  
*Port*  
ob-  
ject.

**Raises**  
In-  
va-  
li-  
dI-  
den-  
tity

**classmethod**  
Find  
a  
port  
base  
on  
ad-  
dres  
and  
re-  
turn  
a  
*Port*  
ob-  
ject.

**Paramete**

- - cls**

the  
*Port*

- **con**  
 Se-  
 cu-  
 rity  
 con-  
 text
- **add**  
 the  
 ad-  
 dres  
 of  
 a  
 port
- **own**  
 DEF  
 RE-  
 CAT  
 a  
 node  
 own  
 to  
 matc  
 agai
- **pro**  
 a  
 node  
 own  
 or  
 lesse  
 to  
 matc  
 agai

**Returns**  
 a  
*Port*  
 ob-  
 ject.

**Raises**  
 Port  
 Not-  
 Four

**classme**



Find  
 a  
 port  
 base  
 on  
 its  
 in-  
 te-  
 ger  
 ID  
 and  
 re-  
 turn  
 a  
 Port  
 ob-  
 ject.

Paramete

- **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  
*Por*  
 ob-  
 ject.

Paramete

- **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  
 UUI

and  
 re-  
 turn  
 a  
*Port*  
 ob-  
 ject.

Parameters

- **cls**  
 the  
*Port*
- **context**  
 Se-  
 cu-  
 rity  
 con-  
 text
- **ui**  
 the  
 UI  
 of  
 a  
 port

Returns

a  
*Port*  
 ob-  
 ject.

Raises

Port  
 Not-  
 Four

property

property

property

classme

Re-  
 turn

a  
list  
of  
Port  
ob-  
jects

Paramete

- **con**  
Se-  
cu-  
rity  
con-  
text.
- **lim**  
max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.
- **mar**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.
- **son**  
col-  
umn

to  
 sort  
 re-  
 sults  
 by.

- **sort**  
 di-  
 rec-  
 tion  
 to  
 sort.  
 asc  
 or  
 desc

- **own**  
 DEF  
 RE-  
 CAT  
 a  
 node  
 own  
 to  
 matc  
 agai

- **pro**  
 a  
 node  
 own  
 or  
 lesse  
 to  
 matc  
 agai

**Returns**  
 a  
 list  
 of  
*Por*  
 ob-  
 ject.

**Raises**  
 In-  
 valic  
 Pa-  
 ram-  
 e-

ter-  
 Valu  
**classme**  
  
 Re-  
 turn  
 a  
 list  
 of  
 Port  
 ob-  
 jects  
 as-  
 so-  
 ci-  
 ated  
 with  
 a  
 give  
 node  
 ID.

**Parame**

- **con**  
 Se-  
 cu-  
 rity  
 con-  
 text.
- **nod**  
 the  
 ID  
 of  
 the  
 node
- **lim**  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to

re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult.

• **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  
 matc  
 agai

•
   
**pro**
  
 a
   
 node
   
 own
   
 or
   
 lesse
   
 to
   
 matc
   
 agai

**Returns**
  
 a
   
 list
   
 of
   
*Por*
  
 ob-
   
 ject.

**classme**

Re-
   
 turn
   
 a
   
 list
   
 of
   
 Port
   
 ob-
   
 jects
   
 as-
   
 so-
   
 ci-
   
 ated
   
 with
   
 a
   
 give
   
 port
   
 grou
   
 ID.

**Parame**

•
   
**con**
  
 Se-
   
 cu-
   
 rity
   
 con-
   
 text.

•



por
 the
 ID
 of
 the
 port
 grou

- **limit**
 max
 i-
 mun
 num
 ber
 of
 re-
 sour
 to
 re-
 turn
 in
 a
 sin-
 gle
 re-
 sult.

- **max**
 pag-
 i-
 na-
 tion
 mar
 for
 large
 data
 sets.

- **sort**
 col-
 umn
 to
 sort
 re-
 sults
 by.

- **sort**
 di-

rec-  
tion  
to  
sort.  
asc  
or  
desc

- **own**  
DEF  
RE-  
CAT  
a  
node  
own  
to  
mate  
agai

- **pro**  
a  
node  
own  
or  
lesse  
to  
mate  
agai

**Returns**  
a  
list  
of  
*Por*  
ob-  
ject.

**property**

**property**

**property**

**property**

**property**

**property**

**refresh**

Load  
up-  
date  
for  
this  
Port

Load  
a  
port  
with  
the  
same  
uuid  
from  
the  
data  
and  
check  
for  
up-  
date  
at-  
tribu-  
Up-  
date  
are

applied from the loaded port column by column, if there are any updates.

**Parame**

**con**  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

Raises

Port  
Not-  
Four

save (co

Save  
up-  
date  
to  
this  
Port

Up-  
date  
will  
be  
mad  
col-  
umn  
by  
col-  
umn  
base  
on  
the  
re-  
sult  
of  
self.

Parame

con  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Port(context)

**Raises**

Port  
 Not-  
 Four

**Raises**

MA  
 read  
 ists  
 if  
 ad-  
 dres  
 col-  
 umn  
 is  
 not  
 uniq

**classme**

Re-  
 turn  
 whe  
 is\_s  
 field  
 is  
 sup-  
 port

**Returns**

Whe  
 is\_s  
 field  
 is  
 sup-  
 port

**Raises**

ovo\_

**classme**

Re-  
 turn  
 whe  
 the  
 phys  
 i-  
 cal\_  
 field  
 is  
 sup-  
 port

**Returns**

Whe

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  
 whe  
 iron  
 cre-  
 ates.  
 up-  
 date  
 or  
 dele  
 a  
 port

VERSION

property

property

fields

property

property

property

property

class i

Base

*irc*

*obj*

*not*

*Not*

SCHEMA

VERSION

property

property

property

fields

property

property

property

property

property

property

property

proper

proper

## ironic.objects.portgroup module

```
class i
    Base
    irc
    obj
    bas
    Irc
    osl
    bas
    Ver
```

**VERSION**

proper

```
create
    Create
    a
    Port
    group
    record
    in
    the
    DB.
```

**Parameters**  
Security context. NOT This should only be used internally by



reception\_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)

the  
in-  
di-

#### Raises

Du-  
pli-  
cate  
Nam  
MA  
read  
ists,  
Port  
grou  
read  
ists

#### properties

**dbapi** =

#### destroy

Dele  
the  
Port  
grou  
from  
the  
DB.

#### Parameters

**con**  
Se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

#### **Raises**

Portgroup.NotFound  
Portgroup.NotFound  
Portgroup.NotFound  
Portgroup.NotFound  
Portgroup.NotFound

#### **properties**

#### **fields**

#### **classmethods**

Find a portgroup based on its id, uuid, name or address

#### **Parameters**

- 

**portgroup**  
The id, uuid, name or address of a portgroup

- 

**context**  
Security

con-  
text

#### Returns

A  
*Port*  
ob-  
ject.

#### Raises

In-  
va-  
li-  
dI-  
den-  
tity

#### classmethod

Find  
port  
group  
by  
ad-  
dres  
and  
re-  
turn  
a  
*Port*  
ob-  
ject.

#### Parameter

- **cls**  
the  
*Port*
- **con**  
Se-  
cu-  
rity  
con-  
text
- **add**  
The  
MA  
ad-  
dres  
of

a  
 port  
 grou

**Returns**

A  
*Por*  
 ob-  
 ject.

**Raises**

Port  
 grou  
 Not-  
 Four

**classme**

Find  
 a  
 port  
 grou  
 by  
 its  
 in-  
 te-  
 ger  
 ID  
 and  
 re-  
 turn  
 a  
 Port  
 grou  
 ob-  
 ject.

**Parame**

- **cls**  
 the  
*Por*
- **con**  
 Se-  
 cu-  
 rity  
 con-  
 text
- **por**  
 The

ID  
of  
a  
port  
group

#### Returns

A  
*Port*  
ob-  
ject.

#### Raises

Port  
group  
Not-  
Found

#### classmethods

Find  
port  
group  
base  
on  
name  
and  
re-  
turn  
a  
*Port*  
ob-  
ject.

#### Parameters

- **cls**  
the  
*Port*
- **con**  
Se-  
cu-  
rity  
con-  
text
- **nam**  
The  
name  
of  
a

port  
 grou

**Returns**

A  
*Port*  
 ob-  
 ject.

**Raises**

Port  
 grou  
 Not-  
 Foun

**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-

sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.

- **mar**  
Pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

- **sor**  
Col-  
umn  
to  
sort  
re-  
sults  
by.

- **sor**  
Di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc

**Returns**  
A  
list  
of  
*Por*  
ob-  
ject.

**Raises**  
In-



valid
 Pa-
 ram-
 e-
 ter-
 Valu

**classme**

Re-
 turn
 a
 list
 of
 Port
 grou
 ob-
 jects
 as-
 so-
 ci-
 ated
 with
 a
 give
 node
 ID.

**Parame**

- **cls**
 the
 *Por*
- **con**
 Se-
 cu-
 rity
 con-
 text.
- **nod**
 The
 ID
 of
 the
 node
- **lim**

Max  
i-  
mun  
num  
ber  
of  
re-  
sour  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-  
sult.

- **max**  
Pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets.

- **sort**  
Col-  
umn  
to  
sort  
re-  
sults  
by.

- **sort**  
Di-  
rec-  
tion  
to  
sort.  
asc  
or  
desc

Returns  
A

list  
 of  
*Por*  
 ob-  
 ject.

**Raises**

In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu

**property**

**property**

**property**

**property**

**refresh**

Load  
 up-  
 date  
 for  
 this  
 Port  
 grou

Load  
 a  
 port  
 grou  
 with  
 the  
 sam  
 uuid  
 from  
 the  
 data  
 and  
 chec  
 for  
 up-  
 date  
 at-  
 tribu  
 Up-

are applied from the loaded portgroup column by column, if there are any updates.

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

date  
 Paramete  
 con  
 Se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-  
 Raises  
 Port  
 grou  
 Not-  
 Four  
 save (co  
 Save  
 up-  
 date  
 to  
 this  
 Port  
 grou  
 Up-  
 date  
 will  
 be  
 mad  
 col-  
 umn  
 by  
 col-  
 umn  
 base  
 on

the  
re-  
sult  
of  
self.

#### Parameters

**context**  
Security  
context.  
NOT  
This  
should  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Portgroup(context)

#### Raises

Port  
group  
Not-  
Four  
Du-  
pli-  
cate  
Nam  
MA  
read  
ists

**properties**

**properties**

**properties**

**class** i

Base

*irc*  
*obj*  
*not*  
*Not*  
 No-  
 ti-  
 fi-  
 ca-  
 tion  
 whe  
 iron  
 cre-  
 ates.  
 up-  
 date  
 or  
 dele  
 a  
 port  
 grou

VERSION

property

property

fields

property

property

property

property

class i

Base  
*irc*  
*obj*  
*not*  
*Not*

SCHEMA

VERSION

property

property

property

fields

property

property

property

property

property

property

property

ironic.objects.trait module

class i
Base
 ironi
 objects
 base
 Ironi

VERSION

create
Create
a
Trait

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

reco  
in  
the  
DB.  
  
**Parame**  
**con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
addi  
the  
trait  
wou  
ex-  
ceed  
the  
per-  
node  
trait  
limi

**Raises**  
Nod  
Not-  
Foun



if  
the  
node  
no  
long  
ap-  
pear  
in  
the  
data

**property**

**dbapi =**

**classme**

Dele  
the  
Trai  
from  
the  
DB.

**Paramet**

- **con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

- **nod**  
The

id  
 of  
 a  
 node

- **traits**  
 A  
 trait  
 string

**Raises**

Node  
 Not-  
 Four  
 if  
 the  
 node  
 no  
 long  
 ap-  
 pear  
 in  
 the  
 data

**Raises**

Node  
 Trai  
 Not-  
 Four  
 if  
 the  
 trait  
 is  
 not  
 foun

**classme**

Che  
 whe  
 a  
 Trai  
 ex-  
 ists  
 in  
 the  
 DB.

**Parame**

- **con**

reception\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

- **nod**  
The  
id  
of  
a  
node
- **tra**  
A  
trait  
strin

#### 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

property

property

property

class i  
 Base  
 ired  
 obj  
 bas  
 Irc  
 ired  
 obj  
 bas  
 Irc

VERSION

classme  
 Re-  
 plac  
 all  
 ex-  
 ist-  
 ing  
 trait  
 with  
 the  
 spec  
 i-  
 fied  
 list.

Parame

- con

reception\_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)`.

se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

- **nod**  
The  
id  
of  
a  
node

- **tra**  
List  
of  
Strin  
trait  
to  
set.

**Raises**  
In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
addi  
the  
trait  
wou  
ex-  
ceed

the  
 per-  
 node  
 trait  
 limit

**Raises**

Node  
 Not-  
 Four  
 if  
 the  
 node  
 no  
 long  
 ap-  
 pear  
 in  
 the  
 data

**property**

**dbapi =**

**classme**

Dele  
 all  
 trait  
 for  
 the  
 spec  
 i-  
 fied  
 node

**Paramet**

- **con**  
 se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

in-  
ter-  
nally  
by  
the  
in-  
di-

- **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  
traits  
for  
the  
spec  
i-  
fied  
node

**Parame**

- **con**  
se-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: Trait(context).

cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

- **nod**  
The  
id  
of  
a  
node

#### **Raises**

Nod  
Not-  
Four  
if  
the  
node  
no  
long  
ap-  
pear  
in  
the  
data

#### **get\_tra**

Re-  
turn  
a  
list  
of  
nam  
of  
the  
trait



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.

**Parame**

**con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

ready exists with the same type and connector\_id

only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-

**Raises**  
 Vol-  
 ume  
 Con  
 nec-  
 torT  
 pe-  
 An-  
 dI-  
 dAl-  
 read  
 ists  
 if  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 al-

**Raises**  
 Vol-  
 ume  
 Con  
 nec-  
 torA  
 read  
 ists  
 if  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 with  
 the

ready exists

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

same  
 UUI  
 al-  
  
 propert  
  
 dbapi =  
  
 destroy  
 Dele  
 the  
 Vol-  
 ume  
 Con  
 nec-  
 tor  
 from  
 the  
 DB.  
  
 Parame  
 con  
 se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-  
  
 Raises  
 Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 if

the  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 can-  
 not  
 be  
 foun

property

fields

classme

Find  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 base  
 on  
 its  
 ID  
 or  
 UUI

Paramete

- **con**  
 se-  
 cu-  
 rity  
 con-  
 text
- **ide**  
 the  
 data  
 pri-  
 mary  
 key  
 ID  
*or*  
 the  
 UUI

of  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor

Returns

a  
*Vol*  
 ob-  
 ject

Raises

In-  
 va-  
 li-  
 dI-  
 den-  
 tity  
 if  
 iden  
 is  
 nei-  
 ther  
 an  
 in-  
 te-  
 ger  
 ID  
 nor  
 a  
 UUI

Raises

Vol-  
 ume  
 Con  
 nec-  
 torN  
 Four  
 if  
 no  
 vol-  
 ume  
 con-  
 nec-  
 tor  
 ex-  
 ists  
 with  
 the

fied ident

spec  
i-

**classme**  
Find  
a  
vol-  
ume  
con-  
nec-  
tor  
base  
on  
its  
in-  
te-  
ger  
ID.

**Parame**

- cls**  
the  
*Vol*
- con**  
Se-  
cu-  
rity  
con-  
text.
- db\_**  
The  
in-  
te-  
ger  
(data  
pri-  
mar-  
key)  
ID  
of  
a  
vol-  
ume  
con-  
nec-  
tor.

fied ID.

Returns

A
   
*Vol*
  
 ob-
   
 ject.

Raises

Vol-
   
 ume
   
 Con
   
 nec-
   
 torN
   
 Four
   
 if
   
 no
   
 vol-
   
 ume
   
 con-
   
 nec-
   
 tor
   
 ex-
   
 ists
   
 with
   
 the
   
 spec
   
 i-

classme

Find
   
 a
   
 vol-
   
 ume
   
 con-
   
 nec-
   
 tor
   
 base
   
 on
   
 its
   
 UUI

Parame

- **cls**
  
 the
   
*Vol*
- **con**
  
 se-
   
 cu-
   
 rity

fied UUID

con-  
 text  
 •  
**uuid**  
 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  
 ex-  
 ists  
 with  
 the  
 spec  
 i-  
**property**  
**classme**  
 Re-  
 turn  
 a  
 list  
 of  
 Vol-



ume  
 Con  
 nec-  
 tor  
 ob-  
 jects

Paramete

- **con**  
 se-  
 cu-  
 rity  
 con-  
 text
- **lim**  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to  
 re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult
- **mar**  
 pag-  
 i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets
- **son**  
 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

**classme**  
  
 Re-  
 turn  
 a  
 list  
 of  
 Vol-  
 ume  
 Con  
 nec-  
 tor  
 ob-

jects  
 re-  
 latec  
 to  
 a  
 give  
 node  
 ID.

Parame

- **con**  
 se-  
 cu-  
 rity  
 con-  
 text
- **nod**  
 the  
 in-  
 te-  
 ger  
 ID  
 of  
 the  
 node
- **lim**  
 max  
 i-  
 mun  
 num  
 ber  
 of  
 re-  
 sour  
 to  
 re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult
- **mar**  
 pag-

i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets

- **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**

**refresh**

Load  
up-  
date  
for  
this  
Vol-  
ume  
Con-  
nec-  
tor.  
Load  
a  
vol-  
ume  
con-  
nec-  
tor  
with  
the  
same  
UI  
from  
the  
data  
and  
check  
for  
up-  
date  
at-

tributes. If there are any updates, they are applied from the loaded volume connector, column by column.

**Param**

con-  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

the  
in-  
di-  
  
**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-  
sult  
of  
self.  
  
**Parame**  
**con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by

reception\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeConnector(context).

#### **Raises**

Vol-  
ume  
Con  
nec-  
torN  
Four  
if  
the  
vol-  
ume  
con-  
nec-  
tor  
can-  
not  
be  
foun

#### **Raises**

Vol-  
ume  
Con  
nec-  
torT  
pe-  
An-  
dI-  
dAl-  
read  
ists  
if  
an-  
othe  
con-  
nec-  
tor  
al-  
read

exists with the same values for type and connector\_id fields

#### **Raises**

In-  
valid  
Pa-  
ram-  
e-

ter-  
 Valu  
 whe  
 the  
 UUI  
 is  
 be-  
 ing  
 char

**property**

**property**

**property**

**class** **i**

Base  
*irc*  
*obj*  
*not*  
*Not*

No-  
 ti-  
 fi-  
 ca-  
 tion  
 emit  
 ted  
 at  
 CRU  
 of  
 a  
 vol-  
 ume  
 con-  
 nec-  
 tor.

**VERSION**

**property**

**property**

**fields**



property

property

property

property

class i

Base

*irc*

*obj*

*not*

*Not*

Pay-

load

sche

for

CRU

of

a

vol-

ume

con-

nec-

tor.

SCHEMA

VERSION

property

property

property

fields

property

property

property

property

ionic.objects.volume\_target module

class i

Base  
 ird  
 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-  
 ter-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

the same node ID and boot index

nally  
by  
the  
in-  
di-

#### Raises

Vol-  
ume  
get-  
Boo  
dex-  
Al-  
read  
ists  
if  
a  
vol-  
ume  
tar-  
get  
al-  
read  
ex-  
ists  
with

#### Raises

Vol-  
ume  
ge-  
tAl-  
read  
ists  
if  
a  
vol-  
ume  
tar-  
get  
with  
the  
sam  
UUI  
ex-  
ists

propert

**dbapi** =

**destroy**

Dele  
the  
Vol-  
ume  
get  
from  
the  
DB.

**Parame**

**con**  
se-  
cu-  
rity  
con-  
text.  
NOT  
This  
shou  
only  
be  
used  
in-  
ter-  
nally  
by  
the  
in-  
di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

**Raises**

Vol-  
ume  
get-  
Not-  
Foun  
if  
the  
vol-  
ume  
tar-  
get  
can-  
not  
be  
foun

property

fields

classmethod

Find  
a  
vol-  
ume  
tar-  
get  
base  
on  
its  
ID  
or  
UUID

Parameters

- **context**  
se-  
cu-  
rity  
con-  
text

- **id**  
the  
data  
pri-  
mary  
key  
ID  
*or*  
the  
UUID  
of  
a  
vol-  
ume  
tar-  
get

Returns

a  
*Volume*  
ob-  
ject

**Raises**

In-  
va-  
li-  
dI-  
den-  
tity  
if  
iden  
is  
nei-  
ther  
an  
in-  
te-  
ger  
ID  
nor  
a  
UUID

**Raises**

Vol-  
ume  
get-  
Not-  
Four  
if  
no  
vol-  
ume  
tar-  
get  
with  
this  
iden  
ex-  
ists

**classme**

Find  
a  
vol-  
ume  
tar-  
get  
base  
on  
its  
data  
ID.

**Parame**

- **cls**  
the  
*Vol*
- **con**  
se-  
cu-  
rity  
con-  
text
- **db\_**  
the  
data  
pri-  
mar-  
key  
(in-  
te-  
ger)  
ID  
of  
a  
vol-  
ume  
tar-  
get

**Returns**  
a  
*Vol*  
ob-  
ject

**Raises**  
Vol-  
ume  
get-  
Not-  
Four  
if  
no  
vol-  
ume  
tar-  
get  
with  
this  
ID

ex-  
 ists  
**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  
with  
this  
UUI  
ex-  
ists

**property**

**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  
a  
sin-

gle  
 re-  
 sult

- **mark**  
 pag-  
 i-  
 na-  
 tion  
 mark  
 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

**classme**

Re-
 turn
 a
 list
 of
 Vol-
 ume
 get
 ob-
 jects
 re-
 latec
 to
 a
 give
 node
 ID.

**Parame**

- con**
 se-
 cu-
 rity
 con-
 text
- nod**
 the
 in-
 te-
 ger
 ID
 of
 the
 node
- lim**
 max
 i-
 mun
 num

ber  
 of  
 re-  
 sour  
 to  
 re-  
 turn  
 in  
 a  
 sin-  
 gle  
 re-  
 sult

- **mar**  
 pag-  
 i-  
 na-  
 tion  
 marl  
 for  
 large  
 data  
 sets

- **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

**classme**

Re-  
 turn  
 a  
 list  
 of  
 Vol-  
 ume  
 get  
 ob-  
 jects  
 re-  
 latec  
 to  
 a  
 give  
 vol-  
 ume  
 ID.

**Parame**

- **con**  
se-  
cu-  
rity  
con-  
text
- **vol**  
the  
UUI

sult

of  
the  
vol-  
ume

- limit**  
max  
i-  
mun  
num  
ber  
of  
vol-  
ume  
tar-  
gets  
to  
re-  
turn  
in  
a  
sin-  
gle  
re-
- max**  
pag-  
i-  
na-  
tion  
marl  
for  
large  
data  
sets
- sort**  
col-  
umn  
to  
sort  
re-  
sults  
by
- sort**  
di-  
rec-

tion  
to  
sort.  
asc  
or  
desc

#### Returns

a  
list  
of  
*Vol*  
ob-  
jects

#### Raises

In-  
valid  
Pa-  
ram-  
e-  
ter-  
Valu  
if  
sort\_  
does  
not  
ex-  
ist

**property**

**property**

#### refresh

Loa  
up-  
date  
for  
this  
Vol-  
ume  
get.  
  
Loa  
a  
vol-  
ume  
tar-  
get  
with  
the

If there are any updates, they are applied from the loaded volume target, column by column.

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

same  
 UI  
 from  
 the  
 data  
 and  
 chec  
 for  
 up-  
 date  
 at-  
 tribu  
  
**Parame**  
**con**  
 se-  
 cu-  
 rity  
 con-  
 text.  
 NOT  
 This  
 shou  
 only  
 be  
 used  
 in-  
 ter-  
 nally  
 by  
 the  
 in-  
 di-  
  
**Raises**  
 Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 the  
 vol-  
 ume  
 tar-  
 get  
 can-  
 not  
 be



found

**save** (*co*)

Save

up-

date

to

this

Vol-

ume

get.

Up-

date

will

be

mad

col-

umn

by

col-

umn

base

on

the

re-

sult

of

self.

**Parame**

**con**

se-

cu-

rity

con-

text.

NOT

This

shou

only

be

used

in-

ter-

nally

by

the

in-

di-

rection\_api. Unfortunately, RPC requires context as the first argument, even though we dont use it. A context should be set when instantiating the object, e.g.: VolumeTarget(context).

the same node ID and boot index values

Raises  
 In-  
 valid  
 Pa-  
 ram-  
 e-  
 ter-  
 Valu  
 if  
 the  
 UUI  
 is  
 be-  
 ing  
 char

Raises  
 Vol-  
 ume  
 get-  
 Boo  
 dex-  
 Al-  
 read  
 ists  
 if  
 a  
 vol-  
 ume  
 tar-  
 get  
 al-  
 read  
 ex-  
 ists  
 with

Raises  
 Vol-  
 ume  
 get-  
 Not-  
 Four  
 if  
 the  
 vol-  
 ume  
 tar-  
 get  
 can-  
 not

be  
 found

**property**

**property**

**property**

**property**

**class** `ironic`

Base  
*ironic*  
*object*  
*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

property

property

property

fields

property

property

property

property

property

property

## Module contents

ironic.

## Submodules

ironic.version module

## Module contents

### 9.1. the Ironics CI

Its im-  
por-  
tant  
to  
un-  
der-  
stan-  
the  
role  
of  
each  
job  
in  
the  
CI,  
how  
to  
add  
new

jobs and how to debug failures that may arise. To facilitate that, we have created the documentation below.

## Jobs description

*stack/ironic* is visible in *Table. OpenStack IroniC CI jobs description*.

The  
de-  
scrip-  
tion  
of  
each  
jobs  
that  
runs  
in  
the  
CI  
when  
you  
sub-  
mit  
a  
patch  
for  
open

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_nod</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.

## 9.1 Developers Guide

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> and deploys 1 virtual baremetal.
---	--

## **Adding a new Job**

### **Are you familiar with Zuul?**

fig and the Zuul Best Practices.

### **Where can I find the existing jobs?**

that contains three files, whose function is described below.

Be-  
fore  
start  
try-  
ing  
to  
fig-  
ure  
out  
how  
Zuul  
work  
take  
som  
time  
and  
read  
about  
Zuul  
Con

The  
jobs  
for  
the  
Iron  
proj  
are  
de-  
fine  
un-  
der  
the  
zuul  
fold  
in  
the  
root  
di-  
rec-  
tory.

•



verted to Zuul v3 yet.

iron  
jobs  
Con  
tains  
the  
con-  
fig-  
u-  
ra-  
tion  
of  
each  
Iron  
Job  
con-  
verte  
to  
Zuul  
v3.

- lega  
iron  
jobs  
Con  
tains  
the  
con-  
fig-  
u-  
ra-  
tion  
of  
each  
Iron  
Job  
that  
have  
been  
con-

- proj  
Con  
tains  
the  
jobs  
that  
will  
run  
dur-

## Create a new Job

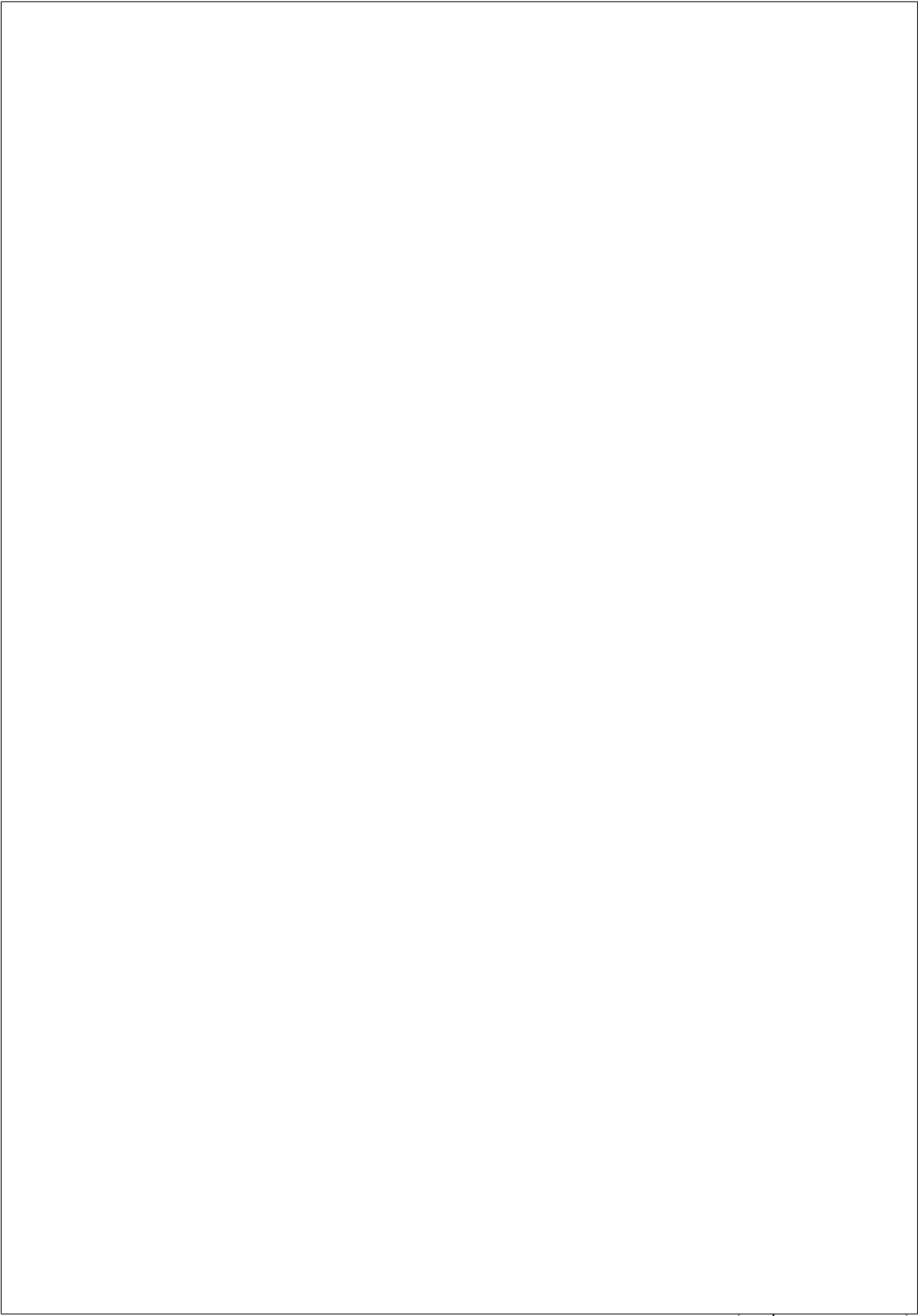
want to test, the existing job will be used as *parent* in your job definition. Now you will only need to either overwrite or add variables to your job definition under the *vars* section to represent the desired scenario.

ing  
chee  
and  
gate  
phas

Ide  
tify  
amo  
the  
ex-  
ist-  
ing  
jobs  
the  
one  
that  
mos  
clos  
re-  
sem  
bles  
the  
sce-  
nari  
you

The  
code  
bloc  
be-  
low  
shov  
the  
min-  
i-  
mal  
struc  
ture  
of  
a  
new  
job  
def-  
i-  
ni-

that you need to add to `ironic-jobs.yaml`.



(continues on next page)

(continued from previous page)

name to the `project.yaml` under *check* and *gate*. Only jobs that are voting should be in the *gate* section.

(continues on next page)

After having the definition of your new job you just need to add the job

If you see **FAILURE** in one or more jobs for your patch,

you to find the initial reason for the failure. When clicking in the failed job you will be redirect to the Zuul web page that contains all the information about the job build.

## **Zuul Web Page**

build failed it will contain a general output of the failure.

plea  
dont  
pani  
This  
guid  
may  
help

The  
page  
has  
three  
tabs  
*Sum*  
*mar*  
*Log*  
and  
*Con*  
*sole*

- Sum  
mar  
Con  
tains  
over  
all  
in-  
for-  
ma-  
tion  
about  
the  
build  
of  
the  
job,  
if  
the  
job

- Log  
Con  
tains  
all

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.

row before each playbook name you can find the roles and commands that were executed.

con-  
fig-  
u-  
ra-  
tions  
and  
log  
files  
about  
all  
ser-  
vice  
that  
were  
used

- Con-  
sole  
Con-  
tains  
all  
the  
play  
book  
that  
were  
ex-  
e-  
cute  
by  
click  
ing  
in  
the  
ar-

9.1.  
pol  
icy  
for  
sta-  
ble  
bra

Sta-  
ble  
bran

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.

tively backport patches from maintained branches. Fixes can still be merged, though, if pushed into review by operators or other downstream developers. It also means that branchless projects (e.g.: `ironiC-tempest-plugin`), may not have configurations that are compatible with those branches.

that  
are  
on  
Ex-  
tend  
Mai  
te-  
nanc  
and  
have  
re-  
ceiv  
back  
port  
in  
a  
whil

Wha  
does  
Unm  
mea  
The  
bran  
still  
ex-  
ists,  
but  
the  
iron  
up-  
strea  
com  
mu-  
nity  
will  
not  
ac-

As  
of  
09  
Mar  
2020  
the  
list  
of



Unm
 bran
 in-
 clud

- Oca
 (Las
 com
 mit
 -

Jun
 28,
 2019

- Pike
 (Las
 com
 mit
 -

Oct
 2,
 2019



## RELEASE NOTES

Release Notes



## PYTHON MODULE INDEX

### a

`ironic.api`, 2224  
`ironic.api.app`, 2207  
`ironic.api.config`, 2208  
`ironic.api.controllers`, 2203  
`ironic.api.controllers.base`, 2198  
`ironic.api.controllers.link`, 2200  
`ironic.api.controllers.root`, 2201  
`ironic.api.controllers.v1`, 2197  
`ironic.api.controllers.v1.allocation`, 1958  
`ironic.api.controllers.v1.bios`, 1964  
`ironic.api.controllers.v1.chassis`, 1966  
`ironic.api.controllers.v1.collection`, 1972  
`ironic.api.controllers.v1.conductor`, 1975  
`ironic.api.controllers.v1.deploy_template`, 1979  
`ironic.api.controllers.v1.driver`, 1986  
`ironic.api.controllers.v1.event`, 1995  
`ironic.api.controllers.v1.node`, 1996  
`ironic.api.controllers.v1.notification_utils`, 2053  
`ironic.api.controllers.v1.port`, 2057  
`ironic.api.controllers.v1.portgroup`, 2068  
`ironic.api.controllers.v1.ramdisk`, 2076  
`ironic.api.controllers.v1.utils`, 2082  
`ironic.api.controllers.v1.versions`, 2166  
`ironic.api.controllers.v1.volume`, 2168  
`ironic.api.controllers.v1.volume_connector`, 2168  
`ironic.api.controllers.v1.volume_target`, 2183  
`ironic.api.controllers.version`, 2201  
`ironic.api.functions`, 2208  
`ironic.api.hooks`, 2215  
`ironic.api.method`, 2223  
`ironic.api.middleware`, 2205  
`ironic.api.middleware.auth_public_routes`, 2203  
`ironic.api.middleware.json_ext`, 2204  
`ironic.api.middleware.parsable_error`, 2204  
`ironic.api.wsgi`, 2224

### c

`ironic.cmd`, 2228  
`ironic.cmd.api`, 2224  
`ironic.cmd.conductor`, 2225  
`ironic.cmd.dbsync`, 2225  
`ironic.cmd.status`, 2227  
`ironic.common`, 2505  
`ironic.common.args`, 2242  
`ironic.common.boot_devices`, 2259  
`ironic.common.boot_modes`, 2261  
`ironic.common.cinder`, 2263  
`ironic.common.components`, 2276  
`ironic.common.config`, 2277  
`ironic.common.context`, 2277  
`ironic.common.dhcp_factory`, 2280  
`ironic.common.driver_factory`, 2283  
`ironic.common.exception`, 2300  
`ironic.common.faults`, 2323  
`ironic.common.fsm`, 2324  
`ironic.common.glance_service`, 2237  
`ironic.common.glance_service.image_service`, 2228

[ironic.common.glance\\_service.service\\_utils, 2235](#)  
[ironic.common.hash\\_ring, 2330](#)  
[ironic.common.il8n, 2330](#)  
[ironic.common.image\\_service, 2331](#)  
[ironic.common.images, 2343](#)  
[ironic.common.indicator\\_states, 2363](#)  
[ironic.common.json\\_rpc, 2242](#)  
[ironic.common.json\\_rpc.client, 2237](#)  
[ironic.common.json\\_rpc.server, 2238](#)  
[ironic.common.keystone, 2363](#)  
[ironic.common.network, 2370](#)  
[ironic.common.neutron, 2376](#)  
[ironic.common.nova, 2403](#)  
[ironic.common.policy, 2404](#)  
[ironic.common.profiler, 2408](#)  
[ironic.common.pxe\\_utils, 2410](#)  
[ironic.common.raid, 2434](#)  
[ironic.common.release\\_mappings, 2442](#)  
[ironic.common.rpc, 2444](#)  
[ironic.common.rpc\\_service, 2448](#)  
[ironic.common.service, 2449](#)  
[ironic.common.states, 2449](#)  
[ironic.common.swift, 2468](#)  
[ironic.common.utils, 2476](#)  
[ironic.common.wsgi\\_service, 2503](#)  
[ironic.conductor, 2749](#)  
[ironic.conductor.allocations, 2505](#)  
[ironic.conductor.base\\_manager, 2510](#)  
[ironic.conductor.cleaning, 2516](#)  
[ironic.conductor.deployments, 2520](#)  
[ironic.conductor.manager, 2525](#)  
[ironic.conductor.notification\\_utils, 2544](#)  
[ironic.conductor.rpcapi, 2550](#)  
[ironic.conductor.steps, 2678](#)  
[ironic.conductor.task\\_manager, 2684](#)  
[ironic.conductor.utils, 2703](#)  
[ironic.conf, 2756](#)  
[ironic.conf.agent, 2749](#)  
[ironic.conf.ansible, 2749](#)  
[ironic.conf.api, 2749](#)  
[ironic.conf.audit, 2749](#)  
[ironic.conf.auth, 2749](#)  
[ironic.conf.cinder, 2751](#)  
[ironic.conf.conductor, 2751](#)  
[ironic.conf.console, 2751](#)  
[ironic.conf.database, 2751](#)  
[ironic.conf.default, 2751](#)  
[ironic.conf.deploy, 2751](#)  
[ironic.conf.dhcp, 2751](#)  
[ironic.conf.drac, 2752](#)  
[ironic.conf.glance, 2752](#)  
[ironic.conf.healthcheck, 2752](#)  
[ironic.conf.ibmcc, 2752](#)  
[ironic.conf.ilo, 2752](#)  
[ironic.conf.inspector, 2752](#)  
[ironic.conf.ipmi, 2752](#)  
[ironic.conf.irmc, 2753](#)  
[ironic.conf.iscsi, 2753](#)  
[ironic.conf.json\\_rpc, 2753](#)  
[ironic.conf.metrics, 2753](#)  
[ironic.conf.metrics\\_statsd, 2753](#)  
[ironic.conf.neutron, 2753](#)  
[ironic.conf.nova, 2753](#)  
[ironic.conf.opts, 2754](#)  
[ironic.conf.pxe, 2756](#)  
[ironic.conf.redfish, 2756](#)  
[ironic.conf.service\\_catalog, 2756](#)  
[ironic.conf.snmp, 2756](#)  
[ironic.conf.swift, 2756](#)  
[ironic.conf.xclarity, 2756](#)

## d

[ironic.db, 3032](#)  
[ironic.db.api, 2906](#)  
[ironic.db.migration, 3031](#)  
[ironic.db.sqlalchemy, 2906](#)  
[ironic.db.sqlalchemy.api, 2757](#)  
[ironic.db.sqlalchemy.migration, 2886](#)  
[ironic.db.sqlalchemy.models, 2890](#)  
[ironic.dhcp, 3053](#)  
[ironic.dhcp.base, 3032](#)  
[ironic.dhcp.neutron, 3039](#)  
[ironic.dhcp.none, 3046](#)  
[ironic.drivers, 4222](#)  
[ironic.drivers.base, 4026](#)  
[ironic.drivers.drac, 4180](#)  
[ironic.drivers.fake\\_hardware, 4182](#)  
[ironic.drivers.generic, 4186](#)  
[ironic.drivers.hardware\\_type, 4190](#)  
[ironic.drivers.ibmcc, 4195](#)  
[ironic.drivers.ilo, 4196](#)  
[ironic.drivers.intel\\_ipmi, 4200](#)  
[ironic.drivers.ipmi, 4201](#)  
[ironic.drivers.irmc, 4202](#)  
[ironic.drivers.modules, 4026](#)  
[ironic.drivers.modules.agent, 3579](#)

ironic.drivers.modules.agent_base,	3206
3606	ironic.drivers.modules.ilo.boot,
ironic.drivers.modules.agent_client,	3212
3644	ironic.drivers.modules.ilo.common,
ironic.drivers.modules.agent_power,	3250
3685	ironic.drivers.modules.ilo.console,
ironic.drivers.modules.ansible,	3288
3062	ironic.drivers.modules.ilo.firmware_processor,
ironic.drivers.modules.ansible.deploy,	3291
3053	ironic.drivers.modules.ilo.inspect,
ironic.drivers.modules.boot_mode_utils,	3296
3693	ironic.drivers.modules.ilo.management,
ironic.drivers.modules.console_utils,	3301
3701	ironic.drivers.modules.ilo.power,
ironic.drivers.modules.deploy_utils,	3331
3708	ironic.drivers.modules.ilo.raid,
ironic.drivers.modules.drac, 3164	3338
ironic.drivers.modules.drac.bios,	ironic.drivers.modules.ilo.vendor,
3062	3346
ironic.drivers.modules.drac.boot,	ironic.drivers.modules.image_cache,
3085	3843
ironic.drivers.modules.drac.common,	ironic.drivers.modules.image_utils,
3090	3848
ironic.drivers.modules.drac.inspect,	ironic.drivers.modules.inspect_utils,
3093	3866
ironic.drivers.modules.drac.job,	ironic.drivers.modules.inspector,
3098	3868
ironic.drivers.modules.drac.management,	ironic.drivers.modules.intel_ipmi,
3102	3351
ironic.drivers.modules.drac.power,	ironic.drivers.modules.intel_ipmi.management,
3116	3351
ironic.drivers.modules.drac.raid,	ironic.drivers.modules.ipmitool,
3125	3871
ironic.drivers.modules.drac.vendor_postinstall,	ironic.drivers.modules.ipxe, 3912
3152	ironic.drivers.modules.irmc, 3417
ironic.drivers.modules.fake, 3753	ironic.drivers.modules.irmc.bios,
ironic.drivers.modules.ibmc, 3206	3352
ironic.drivers.modules.ibmc.management,	ironic.drivers.modules.irmc.boot,
3164	3358
ironic.drivers.modules.ibmc.mappings,	ironic.drivers.modules.irmc.common,
3180	3376
ironic.drivers.modules.ibmc.power,	ironic.drivers.modules.irmc.inspect,
3180	3383
ironic.drivers.modules.ibmc.raid,	ironic.drivers.modules.irmc.management,
3190	3389
ironic.drivers.modules.ibmc.utils,	ironic.drivers.modules.irmc.power,
3198	3403
ironic.drivers.modules.ibmc.vendor,	ironic.drivers.modules.irmc.raid,
3200	3413
ironic.drivers.modules.ilo, 3351	ironic.drivers.modules.iscsi_deploy,
ironic.drivers.modules.ilo.bios,	3913

[ironic.drivers.modules.network,](#)  
[3465](#)

[ironic.drivers.modules.network.common,](#)  
[3417](#)

[ironic.drivers.modules.network.flat,ironic,](#)  
[3435](#)

[ironic.drivers.modules.network.neutron,](#)  
[3445](#)

[ironic.drivers.modules.network.noop,](#)  
[3456](#)

[ironic.drivers.modules.noop,](#)  
[3945](#)

[ironic.drivers.modules.noop\\_mgmt,](#)  
[3970](#)

[ironic.drivers.modules.pxe,](#)  
[3984](#)

[ironic.drivers.modules.pxe\\_base,](#)  
[3990](#)

[ironic.drivers.modules.redfish,](#)  
[3547](#)

[ironic.drivers.modules.redfish.bios,](#)  
[3465](#)

[ironic.drivers.modules.redfish.boot,](#)  
[3474](#)

[ironic.drivers.modules.redfish.inspector,](#)  
[3489](#)

[ironic.drivers.modules.redfish.management,](#)  
[3493](#)

[ironic.drivers.modules.redfish.power,](#)  
[3531](#)

[ironic.drivers.modules.redfish.utils,](#)  
[3539](#)

[ironic.drivers.modules.redfish.vendor,](#)  
[3543](#)

[ironic.drivers.modules.snmp,](#)  
[4000](#)

[ironic.drivers.modules.storage,](#)  
[3564](#)

[ironic.drivers.modules.storage.cinder,](#)  
[3547](#)

[ironic.drivers.modules.storage.external,](#)  
[3553](#)

[ironic.drivers.modules.storage.noop,](#)  
[3558](#)

[ironic.drivers.modules.xclarity,](#)  
[3579](#)

[ironic.drivers.modules.xclarity.common,](#)  
[3564](#)

[ironic.drivers.modules.xclarity.management,](#)  
[3569](#)

[ironic.drivers.modules.xclarity.power,](#)  
[3575](#)

[ironic.drivers.redfish,](#)  
[4205](#)

[ironic.drivers.snmp,](#)  
[4206](#)

[ironic.drivers.utils,](#)  
[4207](#)

[ironic.drivers.xclarity,](#)  
[4221](#)

[ironic.objects,](#)  
[4449](#)

[ironic.objects.allocation,](#)  
[4222](#)

[ironic.objects.base,](#)  
[4235](#)

[ironic.objects.bios,](#)  
[4249](#)

[ironic.objects.chassis,](#)  
[4262](#)

[ironic.objects.conductor,](#)  
[4272](#)

[ironic.objects.deploy\\_template,](#)  
[4281](#)

[ironic.objects.deployment,](#)  
[4296](#)

[ironic.objects.fields,](#)  
[4304](#)

[ironic.objects.indirection,](#)  
[4313](#)

[ironic.objects.node,](#)  
[4324](#)

[ironic.objects.notification,](#)  
[4362](#)

[ironic.objects.port,](#)  
[4368](#)

[ironic.objects.portgroup,](#)  
[4388](#)

[ironic.objects.trait,](#)  
[4403](#)

[ironic.objects.volume\\_connector,](#)  
[4413](#)

[ironic.objects.volume\\_target,](#)  
[4430](#)

[ironic.version,](#)  
[4449](#)



## Symbols

--autogenerate  
     revision command line option,  
     1701  
 --config-dir <DIR>  
     ironic-dbsync command line  
     option, 1699  
 --config-file <PATH>  
     ironic-dbsync command line  
     option, 1699  
 --debug  
     ironic-dbsync command line  
     option, 1699  
 --help  
     create\_schema command line  
     option, 1700  
     ironic-dbsync command line  
     option, 1699  
     online\_data\_migrations command  
     line option, 1700  
     revision command line option,  
     1701  
     stamp command line option, 1701  
     upgrade command line option,  
     1702  
     version command line option,  
     1702  
 --max-count <NUMBER>  
     online\_data\_migrations command  
     line option, 1700  
 --message <MESSAGE>  
     revision command line option,  
     1701  
 --option <MIGRATION.KEY=VALUE>  
     online\_data\_migrations command  
     line option, 1700  
 --revision <REVISION>  
     stamp command line option, 1701  
     upgrade command line option,  
     1702  
 --version

ironic-dbsync command line  
     option, 1699  
 -d  
     ironic-dbsync command line  
     option, 1699  
 -h  
     create\_schema command line  
     option, 1700  
     ironic-dbsync command line  
     option, 1699  
     online\_data\_migrations command  
     line option, 1700  
     revision command line option,  
     1701  
     stamp command line option, 1701  
     upgrade command line option,  
     1702  
     version command line option,  
     1702  
 -m <MESSAGE>  
     revision command line option,  
     1701

## A

abandon\_bios\_config()  
     (*ironic.drivers.modules.drac.vendor\_passthru.DracWSMA*  
     *method*), 3154  
 abandon\_config() (in module  
     *ironic.drivers.modules.drac.bios*), 3070  
 abandon\_config() (in module  
     *ironic.drivers.modules.drac.raid*), 3134  
 abort() (*ironic.drivers.base.InspectInterface*  
     *method*), 4075  
 abort() (*ironic.drivers.modules.inspector.Inspector*  
     *method*), 3868  
 abort\_on\_conductor\_take\_over() (in  
     module *ironic.conductor.utils*), 2703  
 AbstractHardwareType (class in  
     *ironic.drivers.hardware\_type*), 4190  
 acquire() (in module  
     *ironic.conductor.task\_manager*), 2701

`acquire_port()` (in module `ironic.drivers.modules.console_utils`), 3701  
`action()` (`ironic.objects.notification.EventType` property), 4362  
`activate_license()` (`ironic.drivers.modules.ilo.management.IloManagement` method), 3303  
`ACTIVE` (in module `ironic.common.states`), 2450  
`add_allocation_filter_by_conductor()` (in module `ironic.db.sqlalchemy.api`), 2882  
`add_allocation_filter_by_node()` (in module `ironic.db.sqlalchemy.api`), 2882  
`add_auth_opts()` (in module `ironic.conf.auth`), 2749  
`add_cleaning_network()` (`ironic.drivers.base.NetworkInterface` method), 4115  
`add_cleaning_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3436  
`add_cleaning_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3445  
`add_cleaning_network()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 3456  
`add_command_parsers()` (in module `ironic.cmd.dbsync`), 2226  
`add_fieldname()` (`ironic.common.exception.UnknownAttribute` method), 2320  
`add_identity_filter()` (in module `ironic.db.sqlalchemy.api`), 2882  
`add_inspection_network()` (`ironic.drivers.base.NetworkInterface` method), 4116  
`add_inspection_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3437  
`add_inspection_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3446  
`add_node_capability()` (in module `ironic.drivers.utils`), 4209  
`add_node_filter_by_chassis()` (in module `ironic.db.sqlalchemy.api`), 2883  
`add_node_tag()` (`ironic.db.api.Connection` method), 2906  
`add_node_tag()` (`ironic.db.sqlalchemy.api.Connection` method), 2757  
`add_node_trait()` (`ironic.db.api.Connection` method), 2907  
`add_node_trait()` (`ironic.db.sqlalchemy.api.Connection` method), 2758  
`add_node_traits()` (`ironic.conductor.manager.ConductorManager` method), 2528  
`add_node_traits()` (`ironic.conductor.rpcapi.ConductorAPI` method), 2562  
`add_port_filter()` (in module `ironic.db.sqlalchemy.api`), 2883  
`add_port_filter_by_node()` (in module `ironic.db.sqlalchemy.api`), 2884  
`add_port_filter_by_node_owner()` (in module `ironic.db.sqlalchemy.api`), 2884  
`add_port_filter_by_node_project()` (in module `ironic.db.sqlalchemy.api`), 2884  
`add_port_filter_by_portgroup()` (in module `ironic.db.sqlalchemy.api`), 2884  
`add_portgroup_filter()` (in module `ironic.db.sqlalchemy.api`), 2884  
`add_portgroup_filter_by_node()` (in module `ironic.db.sqlalchemy.api`), 2886  
`add_ports_to_network()` (in module `ironic.common.neutron`), 2378  
`add_provisioning_network()` (`ironic.drivers.base.NetworkInterface` method), 4117  
`add_provisioning_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3438  
`add_provisioning_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3447  
`add_provisioning_network()` (`ironic.drivers.modules.network.noop.NoopNetwork` method), 3456  
`add_provisioning_network()` (`ironic.drivers.base.NetworkInterface` method), 4117  
`add_rescuing_network()` (`ironic.drivers.modules.network.flat.FlatNetwork` method), 3438  
`add_rescuing_network()` (`ironic.drivers.modules.network.neutron.NeutronNetwork` method), 3448

[add\\_secret\\_token\(\)](#) (in module [ironic.conductor.utils](#)), 2703  
[add\\_state\(\)](#) ([ironic.common.fsm.FSM](#) method), 2325  
[add\\_transition\(\)](#) ([ironic.common.fsm.FSM](#) method), 2326  
[address](#) ([ironic.db.sqlalchemy.models.Port](#) attribute), 2901  
[address](#) ([ironic.db.sqlalchemy.models.Portgroup](#) attribute), 2902  
[address\(\)](#) ([ironic.objects.port.Port](#) property), 4368  
[address\(\)](#) ([ironic.objects.port.PortCRUDPayload](#) ALL ([ironic.objects.fields.NotificationStatus](#) attribute), 4387  
[address\(\)](#) ([ironic.objects.portgroup.Portgroup](#) ALL ([ironic.objects.fields.NotificationStatus](#) attribute), 4388  
[address\(\)](#) ([ironic.objects.portgroup.PortgroupCRUDPayload](#) ALL ([ironic.objects.fields.NotificationStatus](#) attribute), 4403  
[ADOPTFAIL](#) (in module [ironic.common.states](#)), 2451  
[ADOPTING](#) (in module [ironic.common.states](#)), 2451  
[advanced\\_net\\_fields](#) ([ironic.api.controllers.v1.port.PortsController](#) attribute), 2057  
[after\(\)](#) ([ironic.api.hooks.ContextHook](#) method), 2216  
[after\(\)](#) ([ironic.api.hooks.NoExceptionTracebackHook](#) method), 2219  
[agent\\_add\\_clean\\_params\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), 3708  
[agent\\_is\\_alive\(\)](#) (in module [ironic.conductor.utils](#)), 2704  
[AgentAPIError](#), 2300  
[AgentBaseMixin](#) (class in [ironic.drivers.modules.agent\\_base](#)), 3606  
[AgentClient](#) (class in [ironic.drivers.modules.agent\\_client](#)), 3644  
[AgentCommandTimeout](#), 2300  
[AgentConnectionFailed](#), 2300  
[AgentDeploy](#) (class in [ironic.drivers.modules.agent](#)), 3579  
[AgentDeployMixin](#) (class in [ironic.drivers.modules.agent](#)), 3588  
[AgentDeployMixin](#) (class in [ironic.drivers.modules.agent\\_base](#)), 3612  
[AgentInProgress](#), 2300  
[AgentOobStepsMixin](#) (class in [ironic.drivers.modules.agent\\_base](#)), 3626  
[AgentPower](#) (class in [ironic.drivers.modules.agent\\_power](#)), 3685  
[AgentRAID](#) (class in [ironic.drivers.modules.agent](#)), 3588  
[AgentRescue](#) (class in [ironic.drivers.modules.agent](#)), 3595  
[ALL](#) ([ironic.objects.fields.NotificationLevel](#) attribute), 4309  
[ALL](#) ([ironic.objects.fields.NotificationStatus](#) attribute), 4309  
[ALL\\_INTERFACES](#) (in module [ironic.drivers.base](#)), 4026  
[all\\_interfaces\(\)](#) (in module [ironic.common.driver\\_factory](#)), 2284  
[all\\_interfaces\(\)](#) ([ironic.drivers.base.BareDriver](#) property), 4033  
[all\\_versions\(\)](#) (in module [ironic.api.controllers.version](#)), 2201  
[Allocation](#) (class in [ironic.db.sqlalchemy.models](#)), 2890  
[Allocation](#) (class in [ironic.objects.allocation](#)), 4222  
[allocation\\_id](#) ([ironic.db.sqlalchemy.models.Node](#) attribute), 2896  
[allocation\\_id\(\)](#) ([ironic.objects.node.Node](#) property), 4324  
[allocation\\_sanitize\(\)](#) (in module [ironic.api.controllers.v1.allocation](#)), 1964  
[AllocationAlreadyExists](#), 2300  
[AllocationCRUDNotification](#) (class in [ironic.objects.allocation](#)), 4233  
[AllocationCRUDPayload](#) (class in [ironic.objects.allocation](#)), 4234  
[AllocationDuplicateName](#), 2300  
[AllocationFailed](#), 2300  
[AllocationNotFound](#), 2301  
[AllocationsController](#) (class in [ironic.api.controllers.v1.allocation](#)), 1958  
[allow\\_agent\\_token\(\)](#) (in module [ironic.api.controllers.v1.utils](#)), 2082  
[allow\\_agent\\_version\\_in\\_heartbeat\(\)](#) (in module [ironic.api.controllers.v1.utils](#)), 2082

[allow\\_allocation\\_backfill\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2083  
[allow\\_allocation\\_owner\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2084  
[allow\\_allocation\\_update\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2084  
[allow\\_allocations\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2085  
[allow\\_bios\\_interface\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2086  
[allow\\_build\\_configdrive\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2086  
[allow\\_configdrive\\_vendor\\_data\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2087  
[allow\\_deploy\\_steps\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2087  
[allow\\_deploy\\_templates\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2088  
[allow\\_detail\\_query\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2088  
[allow\\_dynamic\\_drivers\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2089  
[allow\\_dynamic\\_interfaces\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2090  
[allow\\_expose\\_conductors\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2090  
[allow\\_expose\\_events\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2091  
[allow\\_field\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2091  
[allow\\_inject\\_nmi\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2092  
[allow\\_inspect\\_abort\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2092  
[allow\\_inspect\\_wait\\_state\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2093  
[allow\\_links\\_node\\_states\\_and\\_driver\\_properties\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2093  
[allow\\_local\\_link\\_connection\\_network\\_anycast\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2094  
[allow\\_node\\_logical\\_names\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2094  
[allow\\_node\\_rebuild\\_with\\_configdrive\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2094  
[allow\\_port\\_advanced\\_net\\_fields\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2095  
[allow\\_port\\_internal\\_info\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2095  
[allow\\_port\\_is\\_smartrnic\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2096  
[allow\\_port\\_physical\\_network\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2096  
[allow\\_portgroup\\_mode\\_properties\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2097  
[allow\\_portgroups\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2098  
[allow\\_portgroups\\_subcontrollers\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2098  
[allow\\_raid\\_config\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2099  
[allow\\_ramdisk\\_endpoints\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2100  
[allow\\_remove\\_chassis\\_uuid\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2100  
[allow\\_rescue\\_interface\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2101  
[allow\\_reset\\_interfaces\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2101  
[allow\\_soft\\_power\\_off\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2102  
[allow\\_storage\\_interface\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2102  
[allow\\_traits\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2103  
[allow\\_verify\\_ca\\_in\\_heartbeat\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2103  
[allow\\_vifs\\_subcontroller\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2103  
[allow\\_volume\(\)](#) (in module *ironic.api.controllers.v1.utils*), 2104  
[and\\_valid\(\)](#) (in module *ironic.common.args*), 2242  
[AnsibleDeploy](#) (class in *ironic.drivers.modules.ansible.deploy*), 3053  
[apply\\_configuration\(\)](#) (*ironic.drivers.base.BIOSInterface* method), 4027  
[apply\\_configuration\(\)](#) (*ironic.drivers.base.RAIDInterface* method), 4143  
[apply\\_configuration\(\)](#) (*ironic.drivers.modules.agent.AgentRAID*

method), 3589

apply\_configuration() (ironic.drivers.modules.drac.bios.DracWSManBIOS method), 3064

apply\_configuration() (ironic.drivers.modules.drac.raid.DracWSManRAID method), 3126

apply\_configuration() (ironic.drivers.modules.fake.FakeBIOS method), 3754

apply\_configuration() (ironic.drivers.modules.ibm.raid.IbmRAID method), 3190

apply\_configuration() (ironic.drivers.modules.ilo.bios.IloBIOS method), 3206

apply\_configuration() (ironic.drivers.modules.ilo.raid.Ilo5RAID method), 3338

apply\_configuration() (ironic.drivers.modules.irmc.bios.IRMCBIOS method), 3352

apply\_configuration() (ironic.drivers.modules.noop.NoBIOS method), 3947

apply\_configuration() (ironic.drivers.modules.redfish.bios.RedfishBIOS method), 3465

apply\_jsonpatch() (in module ironic.api.controllers.v1.utils), 2105

args (ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 2895

arguments (ironic.api.functions.FunctionDefinition attribute), 2209

as\_dict() (ironic.db.sqlalchemy.models.IronicBase method), 2896

as\_dict() (ironic.objects.base.IronicObject method), 4236

as\_dict() (ironic.objects.base.IronicObjectListBase method), 4245

as\_dict() (ironic.objects.node.Node method), 4324

attach\_boot\_iso\_if\_needed() (in module ironic.drivers.modules.irmc.boot), 3373

attach\_vmedia() (in module ironic.drivers.modules.ilo.common), 3252

attach\_volumes() (in module ironic.common.cinder), 2263

attach\_volumes() (ironic.drivers.base.StorageInterface method), 4159

attach\_volumes() (ironic.drivers.modules.fake.FakeStorage method), 3832

attach\_volumes() (ironic.drivers.modules.storage.cinder.CinderStorage method), 3547

attach\_volumes() (ironic.drivers.modules.storage.external.ExternalStorage method), 3553

attach\_volumes() (ironic.drivers.modules.storage.noop.NoopStorage method), 3559

AUTH\_CLASSES (ironic.drivers.modules.redfish.utils.SessionCache attribute), 3539

auth\_strategy() (in module ironic.common.json\_rpc), 2242

authorize() (in module ironic.common.policy), 2404

AuthPublicRoutes (class in ironic.api.middleware), 2205

AuthPublicRoutes (class in ironic.api.middleware.auth\_public\_routes), 2203

AUTO\_TYPE (ironic.objects.fields.FlexibleDictField attribute), 4306

AUTO\_TYPE (ironic.objects.fields.ListOfFlexibleDictsField attribute), 4306

AUTO\_TYPE (ironic.objects.fields.MACAddressField attribute), 4308

AUTO\_TYPE (ironic.objects.fields.NotificationLevelField attribute), 4309

AUTO\_TYPE (ironic.objects.fields.NotificationStatusField attribute), 4309

AUTO\_TYPE (ironic.objects.fields.StringFieldThatAcceptsCallable attribute), 4312

automated\_clean (ironic.db.sqlalchemy.models.Node attribute), 2896

automated\_clean() (ironic.objects.node.Node property), 4324

AVAILABLE (in module ironic.common.states), 2452

## B

backfill\_allocation() (in module ironic.conductor.allocations), 2505

backup\_bios\_config() (in module ironic.drivers.modules.irmc.management),



- 3402
- BadRequest, 2301
- BareDriver (*class in ironic.drivers.base*), 4033
- BaseConductorManager (*class in ironic.conductor.base\_manager*), 2510
- BaseDHCP (*class in ironic.dhcp.base*), 3032
- BaseDriverFactory (*class in ironic.common.driver\_factory*), 2283
- BaseImageService (*class in ironic.common.image\_service*), 2331
- BaseInterface (*class in ironic.drivers.base*), 4040
- before() (*ironic.api.hooks.ConfigHook method*), 2215
- before() (*ironic.api.hooks.ContextHook method*), 2217
- before() (*ironic.api.hooks.DBHook method*), 2218
- before() (*ironic.api.hooks.PublicUrlHook method*), 2221
- before() (*ironic.api.hooks.RPCHook method*), 2222
- BIOS (*in module ironic.common.boot\_devices*), 2260
- bios (*ironic.drivers.base.BareDriver attribute*), 4033
- bios\_interface (*ironic.db.sqlalchemy.models.Node attribute*), 2896
- bios\_interface() (*ironic.objects.node.Node property*), 4324
- bios\_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload attribute*), 2896
- bios\_interface() (*ironic.objects.node.NodeCRUDPayload property*), 4343
- bios\_interface() (*ironic.objects.node.NodePayload property*), 4352
- bios\_interface() (*ironic.objects.node.NodeSetPowerStatePayload attribute*), 4355
- bios\_interface() (*ironic.objects.node.NodeSetProvisionStatePayload attribute*), 4359
- BIOSInterface (*class in ironic.drivers.base*), 4026
- BIOSSetting (*class in ironic.db.sqlalchemy.models*), 2891
- BIOSSetting (*class in ironic.objects.bios*), 4249
- BIOSSettingAlreadyExists, 2301
- BIOSSettingList (*class in ironic.objects.bios*), 4254
- BIOSSettingListNotFound, 2301
- BIOSSettingNotFound, 2301
- BLINKING (*in module ironic.common.indicator\_states*), 2363
- bmc\_reset() (*ironic.drivers.modules.ipmitool.VendorPassthru method*), 3902
- body() (*in module ironic.api.method*), 2223
- body\_type (*ironic.api.functions.FunctionDefinition attribute*), 2210
- boolean() (*in module ironic.common.args*), 2243
- BooleanField (*class in ironic.objects.fields*), 4304
- boot (*ironic.drivers.base.BareDriver attribute*), 4034
- boot\_device (*ironic.api.controllers.v1.node.NodeManagement attribute*), 2009
- boot\_index (*ironic.db.sqlalchemy.models.VolumeTarget attribute*), 2904
- boot\_index() (*ironic.objects.volume\_target.VolumeTarget property*), 4430
- boot\_index() (*ironic.objects.volume\_target.VolumeTargetCRUD property*), 4448
- boot\_instance() (*ironic.drivers.modules.agent\_base.AgentOobStepsMixin method*), 3626
- boot\_interface (*ironic.db.sqlalchemy.models.Node attribute*), 2896
- boot\_interface() (*ironic.objects.node.Node property*), 4324
- boot\_interface() (*ironic.objects.node.NodeCorrectedPowerStatePayload property*), 4348
- boot\_interface() (*ironic.objects.node.NodeCRUDPayload property*), 4343
- boot\_interface() (*ironic.objects.node.NodePayload property*), 4352
- boot\_interface() (*ironic.objects.node.NodeSetPowerStatePayload attribute*), 4355
- boot\_interface() (*ironic.objects.node.NodeSetProvisionStatePayload attribute*), 4359
- boot\_into\_iso()

(*ironic.drivers.modules.ilo.vendor.VendorPasshine* attribute), 3346  
*method*), 3346  
*boot\_up\_seq()* (*ironic.drivers.modules.ibmc.vendor.IBMCVendor* attribute), 3200  
*method*), 3200  
*BootDeviceController* (class in *ironic.api.controllers.v1.node*), 1996  
*BootInterface* (class in *ironic.drivers.base*), 4051  
*build\_agent\_options()* (in module *ironic.drivers.modules.deploy\_utils*), 3709  
*build\_configdrive()* (in module *ironic.conductor.utils*), 2705  
*build\_deploy\_pxe\_options()* (in module *ironic.common.pxe\_utils*), 2410  
*build\_driver\_for\_task()* (in module *ironic.common.driver\_factory*), 2285  
*build\_extra\_pxe\_options()* (in module *ironic.common.pxe\_utils*), 2410  
*build\_instance\_info\_for\_deploy()* (in module *ironic.drivers.modules.deploy\_utils*), 3710  
*build\_instance\_pxe\_options()* (in module *ironic.common.pxe\_utils*), 2411  
*build\_pxe\_config\_options()* (in module *ironic.common.pxe\_utils*), 2411  
*build\_service\_pxe\_config()* (in module *ironic.common.pxe\_utils*), 2414  
*build\_url()* (in module *ironic.api.controllers.link*), 2200

**C**

*cache\_bios\_settings()* (in module *ironic.drivers.base*), 4167  
*cache\_bios\_settings()* (*ironic.drivers.base.BIOSInterface* method), 4029  
*cache\_bios\_settings()* (*ironic.drivers.modules.drac.bios.DracWSManBIOS* attribute), 3065  
*method*), 3065  
*cache\_bios\_settings()* (*ironic.drivers.modules.fake.FakeBIOS* method), 3756  
*cache\_bios\_settings()* (*ironic.drivers.modules.ilo.bios.IloBIOS* method), 3208  
*cache\_bios\_settings()* (*ironic.drivers.modules.irmc.bios.IRMCBIOS* method), 3354  
*cache\_bios\_settings()* (*ironic.drivers.modules.noop.NoBIOS* method), 3950  
*cache\_instance\_image()* (in module *ironic.drivers.modules.deploy\_utils*), 3711  
*cache\_ramdisk\_kernel()* (in module *ironic.common.pxe\_utils*), 2414  
*call()* (*ironic.common.glance\_service.image\_service.GlanceImageService* method), 2228  
*can\_send\_create\_port()* (*ironic.conductor.rpcapi.ConductorAPI* method), 2564  
*can\_send\_rescue()* (*ironic.conductor.rpcapi.ConductorAPI* method), 2564  
*can\_send\_version()* (*ironic.common.json\_rpc.client.Client* method), 2238  
*candidate\_nodes* (*ironic.db.sqlalchemy.models.Allocation* attribute), 2890  
*candidate\_nodes()* (*ironic.objects.allocation.Allocation* property), 4222  
*candidate\_nodes()* (*ironic.objects.allocation.AllocationCRUDPayload* property), 4234  
*capabilities* (*ironic.drivers.base.BootInterface* attribute), 4052  
*capabilities* (*ironic.drivers.modules.fake.FakeBoot* attribute), 3762  
*capabilities* (*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot* attribute), 3218  
*capabilities* (*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot* attribute), 3229  
*capabilities* (*ironic.drivers.modules.ipxe.iPXEBoot* attribute), 3913  
*capabilities* (*ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot* attribute), 3363  
*capabilities* (*ironic.drivers.modules.pxe.PXEBoot* attribute), 3984  
*capabilities* (*ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot* attribute), 3477  
*capabilities\_to\_dict()* (in module *ironic.drivers.utils*), 4210  
*CatalogNotFound*, 2301  
*CDROM* (in module *ironic.common.boot\_devices*),

2260  
change\_node\_power\_state() (in module *ironic.conductor.manager.ConductorManager* method), 2528  
change\_node\_power\_state() (in module *ironic.conductor.rpcapi.ConductorAPI* method), 2564  
change\_physical\_disk\_state() (in module *ironic.drivers.modules.drac.raid*), 3135  
Chassis (class in *ironic.db.sqlalchemy.models*), 2892  
Chassis (class in *ironic.objects.chassis*), 4262  
CHASSIS (in module *ironic.common.components*), 2276  
chassis\_id (*ironic.db.sqlalchemy.models.Node* attribute), 2897  
chassis\_id() (*ironic.objects.node.Node* property), 4324  
chassis\_uuid() (in module *ironic.objects.node.NodeCRUDPayload* property), 4343  
ChassisAlreadyExists, 2301  
ChassisController (class in *ironic.api.controllers.v1.chassis*), 1966  
ChassisCRUDNotification (class in *ironic.objects.chassis*), 4271  
ChassisCRUDPayload (class in *ironic.objects.chassis*), 4272  
ChassisNotEmpty, 2302  
ChassisNotFound, 2302  
check() (in module *ironic.common.policy*), 2405  
check\_allocation\_policy\_and\_retrieve() (in module *ironic.api.controllers.v1.utils*), 2106  
check\_allow\_configdrive() (in module *ironic.api.controllers.v1.utils*), 2108  
check\_allow\_deploy\_steps() (in module *ironic.api.controllers.v1.utils*), 2108  
check\_allow\_driver\_detail() (in module *ironic.api.controllers.v1.utils*), 2108  
check\_allow\_filter\_by\_conductor() (in module *ironic.api.controllers.v1.utils*), 2109  
check\_allow\_filter\_by\_conductor\_group() (in module *ironic.api.controllers.v1.utils*), 2109  
check\_allow\_filter\_by\_fault() (in module *ironic.api.controllers.v1.utils*), 2110  
check\_allow\_filter\_by\_lessee() (in module *ironic.api.controllers.v1.utils*), 2110  
check\_allow\_filter\_by\_owner() (in module *ironic.api.controllers.v1.utils*), 2111  
check\_allow\_filter\_driver\_type() (in module *ironic.api.controllers.v1.utils*), 2111  
check\_allow\_management\_verbs() (in module *ironic.api.controllers.v1.utils*), 2112  
check\_allow\_specify\_driver() (in module *ironic.api.controllers.v1.utils*), 2112  
check\_allow\_specify\_fields() (in module *ironic.api.controllers.v1.utils*), 2112  
check\_allow\_specify\_resource\_class() (in module *ironic.api.controllers.v1.utils*), 2113  
check\_allowed\_fields() (in module *ironic.api.controllers.v1.utils*), 2114  
check\_allowed\_portgroup\_fields() (in module *ironic.api.controllers.v1.utils*), 2114  
check\_and\_update\_node\_interfaces() (in module *ironic.common.driver\_factory*), 2287  
check\_dir() (in module *ironic.common.utils*), 2476  
check\_file\_system\_for\_iscsi\_device() (in module *ironic.drivers.modules.iscsi\_deploy*), 3923  
check\_for\_invalid\_fields() (in module *ironic.api.controllers.v1.utils*), 2115  
check\_for\_invalid\_state\_and\_allow\_filter() (in module *ironic.api.controllers.v1.utils*), 2116  
check\_for\_missing\_params() (in module *ironic.drivers.modules.deploy\_utils*), 3713  
check\_image\_service() (in module *ironic.common.glance\_service.image\_service*), 2234  
check\_image\_size() (in module *ironic.drivers.modules.agent*), 3602  
check\_image\_size() (in module *ironic.drivers.modules.iscsi\_deploy*), 3923



[check\\_interface\\_capability\(\)](#) [4168](#)  
 (in [module](#) [ironic.drivers.modules.deploy\\_utils](#)), [3715](#)  
[check\\_list\\_policy\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2117](#)  
[check\\_multiple\\_node\\_policies\\_and\\_retrieve\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2118](#)  
[check\\_node\\_list\(\)](#) ([ironic.db.api.Connection](#) [method](#)), [2909](#)  
[check\\_node\\_list\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) [method](#)), [2760](#)  
[check\\_node\\_policy\\_and\\_retrieve\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2120](#)  
[check\\_obj\\_versions\(\)](#) ([ironic.cmd.dbsync.DBCommand](#) [method](#)), [2225](#)  
[check\\_owner\\_policy\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2122](#)  
[check\\_policy\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2123](#)  
[check\\_port\\_list\\_policy\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2124](#)  
[check\\_port\\_policy\\_and\\_retrieve\(\)](#) (in [module](#) [ironic.api.controllers.v1.utils](#)), [2125](#)  
[check\\_share\\_fs\\_mounted\(\)](#) (in [module](#) [ironic.drivers.modules.irmc.boot](#)), [3375](#)  
[check\\_versions\(\)](#) ([ironic.db.api.Connection](#) [method](#)), [2910](#)  
[check\\_versions\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) [method](#)), [2761](#)  
[Checks](#) (class in [ironic.cmd.status](#)), [2227](#)  
[CinderStorage](#) (class in [ironic.drivers.modules.storage.cinder](#)), [3547](#)  
[clean\\_dhcp\(\)](#) ([ironic.common.dhcp\\_factory.DHCPFactory](#) [method](#)), [2280](#)  
[clean\\_dhcp\\_opts\(\)](#) ([ironic.dhcp.base.BaseDHCP](#) [method](#)), [3032](#)  
[CLEAN\\_FAILURE](#) (in [module](#) [ironic.common.faults](#)), [2323](#)  
[clean\\_step\(\)](#) ([ironic.db.sqlalchemy.models.Node](#) [attribute](#)), [2897](#)  
[clean\\_step\(\)](#) (in [module](#) [ironic.drivers.base](#)), [4168](#)  
[clean\\_step\(\)](#) ([ironic.objects.node.Node](#) [property](#)), [4325](#)  
[clean\\_step\(\)](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) [property](#)), [4348](#)  
[clean\\_step\(\)](#) ([ironic.objects.node.NodeCRUDPayload](#) [property](#)), [4343](#)  
[clean\\_step\(\)](#) ([ironic.objects.node.NodePayload](#) [property](#)), [4352](#)  
[clean\\_step\(\)](#) ([ironic.objects.node.NodeSetPowerStatePayload](#) [property](#)), [4355](#)  
[clean\\_step\(\)](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) [property](#)), [4359](#)  
[clean\\_up\(\)](#) ([ironic.drivers.base.DeployInterface](#) [method](#)), [4062](#)  
[clean\\_up\(\)](#) ([ironic.drivers.base.RescueInterface](#) [method](#)), [4155](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.agent.AgentDeploy](#) [method](#)), [3580](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.agent.AgentRescue](#) [method](#)), [3596](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.agent\\_base.AgentBaseMixin](#) [method](#)), [3606](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.ansible.deploy.AnsibleDeploy](#) [method](#)), [3054](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.fake.FakeDeploy](#) [method](#)), [3776](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.image\\_cache.ImageCache](#) [method](#)), [3843](#)  
[clean\\_up\(\)](#) ([ironic.drivers.modules.iscsi\\_deploy.ISCSIDeploy](#) [method](#)), [3913](#)  
[clean\\_up\\_caches\(\)](#) (in [module](#) [ironic.drivers.modules.image\\_cache](#)), [3846](#)  
[clean\\_up\\_instance\(\)](#) ([ironic.drivers.base.BootInterface](#) [method](#)), [4052](#)  
[clean\\_up\\_instance\(\)](#) ([ironic.drivers.modules.fake.FakeBoot](#) [method](#)), [3762](#)  
[clean\\_up\\_instance\(\)](#) ([ironic.drivers.modules.ilo.boot.IloPXEBoot](#) [method](#)), [3241](#)  
[clean\\_up\\_instance\(\)](#) ([ironic.drivers.modules.ilo.boot.IloPXEBoot](#) [method](#)), [3213](#)  
[clean\\_up\\_instance\(\)](#) ([ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot](#) [method](#)), [3218](#)  
[clean\\_up\\_instance\(\)](#) ([ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot](#) [method](#)), [3218](#)

<i>method</i> ), 3229	<code>cleanup_cleanwait_timeout()</code> (in module <i>ironic.conductor.utils</i> ), 2709
<code>clean_up_instance()</code>	<code>cleanup_disk_image()</code> (in module <i>ironic.drivers.modules.image_utils</i> ), 3852
( <i>ironic.drivers.modules.irmc.boot.IRMCPXEBoot</i> <i>method</i> ), 3358	<code>cleanup_floppy_image()</code> (in module <i>ironic.drivers.modules.image_utils</i> ), 3852
<code>clean_up_instance()</code>	<code>cleanup_iso_image()</code> (in module <i>ironic.drivers.modules.image_utils</i> ), 3853
( <i>ironic.drivers.modules.irmc.boot.IRMCPXEBoot</i> <i>method</i> ), 3363	<code>cleanup_vmedia_boot()</code> (in module <i>ironic.drivers.modules.ilo.common</i> ), 3253
<code>clean_up_instance()</code>	<code>CLEANWAIT</code> (in module <i>ironic.common.states</i> ), 2453
( <i>ironic.drivers.modules.pxe_base.PXEBaseMixin</i> <i>method</i> ), 3990	<code>clear_foreign_config()</code> (in module <i>ironic.drivers.modules.drac.raid</i> ), 3138
<code>clean_up_instance()</code>	<code>clear_iscsi_boot_target()</code> ( <i>ironic.drivers.modules.ilo.management.IloManagement</i> <i>method</i> ), 3304
( <i>ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot</i> <i>method</i> ), 3477	<code>clear_job_queue()</code> ( <i>ironic.drivers.modules.drac.management.DracWSManM</i> <i>method</i> ), 3104
<code>clean_up_pxe_config()</code> (in module <i>ironic.common.pxe_utils</i> ), 2414	<code>clear_node_reservations_for_conductor()</code> ( <i>ironic.db.sqlalchemy.api.Connection</i> <i>method</i> ), 2762
<code>clean_up_pxe_env()</code> (in module <i>ironic.common.pxe_utils</i> ), 2415	<code>clear_node_target_power_state()</code> ( <i>ironic.db.sqlalchemy.api.Connection</i> <i>method</i> ), 2762
<code>clean_up_ramdisk()</code>	<code>clear_secure_boot_keys()</code> ( <i>ironic.drivers.modules.ilo.management.IloManagement</i> <i>method</i> ), 3305
( <i>ironic.drivers.base.BootInterface</i> <i>method</i> ), 4052	<code>clear_secure_boot_keys()</code> ( <i>ironic.drivers.modules.redfish.management.RedfishMan</i> <i>method</i> ), 3493
<code>clean_up_ramdisk()</code>	<code>CLEANFAIL</code> (in module <i>ironic.common.states</i> ), 2452
( <i>ironic.drivers.modules.fake.FakeBoot</i> <i>method</i> ), 3763	<code>CLEANING</code> (in module <i>ironic.common.states</i> ), 2453
<code>clean_up_ramdisk()</code>	<code>cleaning_error_handler()</code> (in module <i>ironic.conductor.utils</i> ), 2706
( <i>ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot</i> <i>method</i> ), 3218	<code>cleanup()</code> (in module <i>ironic.common.rpc</i> ), 2447
<code>clean_up_ramdisk()</code>	<code>cleanup()</code> (in module <i>ironic.drivers.modules.image_cache</i> ), 3848
( <i>ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot</i> <i>method</i> ), 3230	<code>cleanup_after_timeout()</code> (in module <i>ironic.conductor.utils</i> ), 2709
<code>clean_up_ramdisk()</code>	
( <i>ironic.drivers.modules.irmc.boot.IRMCPXEBoot</i> <i>method</i> ), 3364	
<code>clean_up_ramdisk()</code>	
( <i>ironic.drivers.modules.pxe_base.PXEBaseMixin</i> <i>method</i> ), 3991	
<code>clean_up_ramdisk()</code>	
( <i>ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot</i> <i>method</i> ), 3478	
<code>CLEANFAIL</code> (in module <i>ironic.common.states</i> ), 2452	
<code>CLEANING</code> (in module <i>ironic.common.states</i> ), 2453	
<code>cleaning_error_handler()</code> (in module <i>ironic.conductor.utils</i> ), 2706	
<code>cleanup()</code> (in module <i>ironic.common.rpc</i> ), 2447	
<code>cleanup()</code> (in module <i>ironic.drivers.modules.image_cache</i> ), 3848	
<code>cleanup_after_timeout()</code> (in module <i>ironic.conductor.utils</i> ), 2709	
	<code>Client</code> (class in <i>ironic.common.json_rpc.client</i> ), 2238
	<code>ClientSideError</code> , 2302
	<code>code</code> ( <i>ironic.common.exception.Conflict</i> attribute), 2303
	<code>code</code> ( <i>ironic.common.exception.Invalid</i> attribute), 2310
	<code>code</code> ( <i>ironic.common.exception.NoFreeConductorWorker</i> attribute), 2314
	<code>code</code> ( <i>ironic.common.exception.NotAcceptable</i> attribute), 2316
	<code>code</code> ( <i>ironic.common.exception.NotAuthorized</i> attribute), 2316

code (*ironic.common.exception.NotFound* attribute), 2316

code (*ironic.common.exception.TemporaryFailure* attribute), 2320

code (*ironic.common.json\_rpc.server.InvalidParams* attribute), 2239

code (*ironic.common.json\_rpc.server.InvalidRequest* attribute), 2239

code (*ironic.common.json\_rpc.server.MethodNotFound* attribute), 2240

code (*ironic.common.json\_rpc.server.ParseError* attribute), 2240

coerce() (*ironic.objects.fields.FlexibleDict* static method), 4304

coerce() (*ironic.objects.fields.MACAddress* static method), 4307

coerce() (*ironic.objects.fields.StringAcceptsCallable* static method), 4310

collect\_ramdisk\_logs() (in module *ironic.drivers.utils*), 4211

collect\_system\_logs() (*ironic.drivers.modules.agent\_client.AgentClient* method), 3645

collection\_from\_list() (in module *ironic.api.controllers.v1.bios*), 1965

commit\_bios\_config() (*ironic.drivers.modules.drac.vendor\_passthru.DracVendorPassthru* method), 3155

commit\_config() (in module *ironic.drivers.modules.drac.bios*), 3071

commit\_config() (in module *ironic.drivers.modules.drac.raid*), 3139

CommunicationError, 2302

compute\_image\_checksum() (in module *ironic.drivers.modules.deploy\_utils*), 3716

Conductor (class in *ironic.db.sqlalchemy.models*), 2892

Conductor (class in *ironic.objects.conductor*), 4272

conductor\_affinity (*ironic.db.sqlalchemy.models.Allocation* attribute), 2890

conductor\_affinity (*ironic.db.sqlalchemy.models.Node* attribute), 2897

conductor\_affinity() (*ironic.objects.allocation.Allocation* property), 4222

conductor\_affinity() (*ironic.objects.node.Node* property), 4325

conductor\_group (*ironic.db.sqlalchemy.models.Conductor* attribute), 2893

conductor\_group (*ironic.db.sqlalchemy.models.Node* attribute), 2897

conductor\_group() (*ironic.objects.conductor.Conductor* property), 4272

conductor\_group() (*ironic.objects.node.Node* property), 4325

conductor\_group() (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4348

conductor\_group() (*ironic.objects.node.NodeCRUDPayload* property), 4343

conductor\_group() (*ironic.objects.node.NodePayload* property), 4352

conductor\_group() (*ironic.objects.node.NodeSetPowerStatePayload* property), 4355

conductor\_group() (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4359

conductor\_id (*ironic.db.sqlalchemy.models.ConductorHardware* attribute), 2894

ConductorAlreadyRegistered, 2302

ConductorAPI (class in *ironic.conductor.rpcapi*), 2550

ConductorHardwareInterfaces (class in *ironic.db.sqlalchemy.models*), 2893

ConductorHardwareInterfacesAlreadyRegistered, 2302

ConductorManager (class in *ironic.conductor.manager*), 2527

ConductorNotFound, 2302

ConductorsController (class in *ironic.api.controllers.v1.conductor*), 1975

config() (in module *ironic.api.controllers.v1.ramdisk*), 2081

ConfigHook (class in *ironic.api.hooks*), 2215

ConfigInvalid, 2303

ConfigNotFound, 2303

configure\_intel\_speedselect() (*ironic.drivers.modules.intel\_ipmi.management.IntelIPMI* method), 3645

[method](#)), [3351](#)  
[configure\\_local\\_boot\(\)](#)  
 ([ironic.drivers.modules.agent\\_base.AgentDeployMixin](#)  
[method](#)), [3613](#)  
[configure\\_secure\\_boot\\_if\\_needed\(\)](#)  
 (in [module](#) [console\\_enabled\(\)](#)  
[ironic.drivers.modules.boot\\_mode\\_utils](#)),  
[3693](#)  
[configure\\_tenant\\_networks\(\)](#) [console\\_interface](#)  
 ([ironic.drivers.base.NetworkInterface](#)  
[method](#)), [4118](#) ([ironic.db.sqlalchemy.models.Node](#)  
[attribute](#)), [2897](#)  
[configure\\_tenant\\_networks\(\)](#) [console\\_interface\(\)](#)  
 ([ironic.drivers.modules.network.flat.FlatNetwork](#) ([ironic.objects.node.Node](#) [property](#)),  
[method](#)), [3439](#) [4325](#)  
[configure\\_tenant\\_networks\(\)](#) [console\\_interface\(\)](#)  
 ([ironic.drivers.modules.network.neutron.NeutronNetwork](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#)  
[method](#)), [3448](#) [property](#)), [4348](#)  
[configure\\_tenant\\_networks\(\)](#) [console\\_interface\(\)](#)  
 ([ironic.drivers.modules.network.noop.NoopNetwork](#) ([ironic.objects.node.NodeCRUDPayload](#)  
[method](#)), [3457](#) [property](#)), [4343](#)  
[Conflict](#), [2303](#) [console\\_interface\(\)](#)  
[Connection](#) (class in [ironic.db.api](#)), [2906](#) ([ironic.objects.node.NodePayload](#) [prop-](#)  
[Connection](#) (class in [ironic.db.sqlalchemy.api](#)), [2757](#) [erty](#)), [4352](#)  
[connection](#) ([ironic.common.swift.SwiftAPI](#) [at-](#)  
[tribute](#)), [2468](#) [console\\_interface\(\)](#)  
[connector\\_id](#) ([ironic.db.sqlalchemy.models.VolumeConnector](#) [console\\_interface\(\)](#)  
[attribute](#)), [2903](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#)  
[property](#)), [4413](#) [property](#)), [4359](#)  
[connector\\_id\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnector](#) [console\\_error](#), [2303](#)  
[property](#)), [4413](#) [ConsoleInterface](#) (class in  
[ironic.drivers.base](#)), [4059](#)  
[connector\\_id\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnector](#) [console\\_error](#), [2303](#)  
[property](#)), [4429](#) [ContextHook](#) (class in [ironic.api.hooks](#)), [2216](#)  
[console](#) ([ironic.api.controllers.v1.node.NodeStatesController](#) [console\\_cleaning\(\)](#)  
[attribute](#)), [2010](#) ([ironic.drivers.modules.agent\\_base.HeartbeatMixin](#)  
[method](#)), [3628](#)  
[console](#) ([ironic.drivers.base.BareDriver](#) [at-](#)  
[tribute](#)), [4034](#) [continue\\_deploy\(\)](#) (in [module](#)  
[console\\_enabled](#) [ironic.drivers.modules.iscsi\\_deploy](#)),  
[ironic.db.sqlalchemy.models.Node](#)  
[attribute](#)), [2897](#) [3924](#)  
[console\\_enabled\(\)](#) ([ironic.objects.node.Node](#) [property](#)),  
[4325](#) [continue\\_deploy\(\)](#)  
 ([ironic.drivers.modules.agent\\_base.HeartbeatMixin](#)  
[method](#)), [3628](#)  
[console\\_enabled\(\)](#) [continue\\_node\\_clean\(\)](#) (in [module](#)  
[ironic.conductor.cleaning](#)), [2516](#)  
 ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) [continue\\_node\\_clean\(\)](#)  
[property](#)), [4348](#) ([ironic.conductor.manager.ConductorManager](#)  
[method](#)), [2528](#)  
[console\\_enabled\(\)](#) ([ironic.objects.node.NodeCRUDPayload](#) [continue\\_node\\_clean\(\)](#)  
[property](#)), [4343](#) ([ironic.conductor.rpcapi.ConductorAPI](#)  
[method](#)), [2566](#)

[continue\\_node\\_deploy\(\)](#) (in module [ironic.conductor.deployments](#)), 2520  
[continue\\_node\\_deploy\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 2530  
[continue\\_node\\_deploy\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), 2567  
[Controller](#) (class in [ironic.api.controllers.v1](#)), 2198  
[convert\(\)](#) (in module [ironic.api.controllers.v1.volume](#)), 2168  
[convert\\_steps\(\)](#) (in module [ironic.api.controllers.v1.deploy\\_template](#)), 1984  
[convert\\_to\\_version\(\)](#) ([ironic.objects.base.IronicObject](#) method), 4236  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.allocation](#)), 1964  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.bios](#)), 1965  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.chassis](#)), 1971  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.conductor](#)), 1978  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.deploy\\_template](#)), 1984  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.driver](#)), 1992  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.port](#)), 2066  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.portgroup](#)), 2075  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.ramdisk](#)), 2081  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.volume\\_connector](#)), 2183  
[convert\\_with\\_links\(\)](#) (in module [ironic.api.controllers.v1.volume\\_target](#)), 2197  
[converted\\_size\(\)](#) (in module [ironic.common.images](#)), 2343  
[copy\\_image\\_to\\_swift\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), 3254  
[copy\\_image\\_to\\_web\\_server\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), 3256  
[core\\_interfaces\(\)](#) ([ironic.drivers.base.BareDriver](#) property), 4035  
[create\(\)](#) ([ironic.objects.allocation.Allocation](#) method), 4222  
[create\(\)](#) ([ironic.objects.bios.BIOSSetting](#) method), 4249  
[create\(\)](#) ([ironic.objects.bios.BIOSSettingList](#) class method), 4255  
[create\(\)](#) ([ironic.objects.chassis.Chassis](#) method), 4262  
[create\(\)](#) ([ironic.objects.deploy\\_template.DeployTemplate](#) method), 4281  
[create\(\)](#) ([ironic.objects.deployment.Deployment](#) method), 4296  
[create\(\)](#) ([ironic.objects.node.Node](#) method), 4325  
[create\(\)](#) ([ironic.objects.port.Port](#) method), 4368  
[create\(\)](#) ([ironic.objects.portgroup.Portgroup](#) method), 4388  
[create\(\)](#) ([ironic.objects.trait.Trait](#) method), 4403  
[create\(\)](#) ([ironic.objects.trait.TraitList](#) class method), 4408  
[create\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnector](#) method), 4413  
[create\(\)](#) ([ironic.objects.volume\\_target.VolumeTarget](#) method), 4430  
[create\\_allocation\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 2532  
[create\\_allocation\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), 2569  
[create\\_allocation\(\)](#) ([ironic.db.api.Connection](#) method), 2911  
[create\\_allocation\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), 2762  
[create\\_bios\\_setting\\_list\(\)](#) ([ironic.db.api.Connection](#) method), 2912  
[create\\_bios\\_setting\\_list\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), 2763



<code>create_boot_iso()</code> (in module <code>ironic.common.images</code> ), 2345	<code>create_node()</code> ( <code>ironic.db.api.Connection</code> method), 2917
<code>create_chassis()</code> ( <code>ironic.db.api.Connection</code> method), 2915	<code>create_node()</code> ( <code>ironic.db.sqlalchemy.api.Connection</code> method), 2768
<code>create_chassis()</code> ( <code>ironic.db.sqlalchemy.api.Connection</code> method), 2766	<code>create_object()</code> ( <code>ironic.common.swift.SwiftAPI</code> method), 2468
<code>create_configuration()</code> ( <code>ironic.drivers.base.RAIDInterface</code> method), 4146	<code>create_port()</code> ( <code>ironic.conductor.manager.ConductorManager</code> method), 2532
<code>create_configuration()</code> ( <code>ironic.drivers.modules.agent.AgentRAID</code> method), 3591	<code>create_port()</code> ( <code>ironic.conductor.rpcapi.ConductorAPI</code> method), 2572
<code>create_configuration()</code> ( <code>ironic.drivers.modules.drac.raid.DracWSManRAID</code> method), 3129	<code>create_port()</code> ( <code>ironic.db.api.Connection</code> method), 2919
<code>create_configuration()</code> ( <code>ironic.drivers.modules.fake.FakeRAID</code> method), 3821	<code>create_port()</code> ( <code>ironic.db.sqlalchemy.api.Connection</code> method), 2770
<code>create_configuration()</code> ( <code>ironic.drivers.modules.ibm.mcraid.IbmMcRAID</code> method), 3193	<code>create_portgroup()</code> ( <code>ironic.db.api.Connection</code> method), 2920
<code>create_configuration()</code> ( <code>ironic.drivers.modules.ilo.raid.Ilo5RAID</code> method), 3341	<code>create_portgroup()</code> ( <code>ironic.db.sqlalchemy.api.Connection</code> method), 2771
<code>create_configuration()</code> ( <code>ironic.drivers.modules.irmc.raid.IRMCRaid</code> method), 3413	<code>create_ports_if_not_exist()</code> (in module <code>ironic.drivers.modules.inspect_utils</code> ), 3866
<code>create_configuration()</code> ( <code>ironic.drivers.modules.noop.NoRAID</code> method), 3959	<code>create_pxe_config()</code> (in module <code>ironic.common.pxe_utils</code> ), 2416
<code>create_deploy_template()</code> ( <code>ironic.db.api.Connection</code> method), 2915	<code>create_schema</code> command line option <code>--help</code> , 1700
<code>create_deploy_template()</code> ( <code>ironic.db.sqlalchemy.api.Connection</code> method), 2766	<code>create_schema</code> command line option <code>-h</code> , 1700
<code>create_esp_image_for_uefi()</code> (in module <code>ironic.common.images</code> ), 2349	<code>create_schema()</code> (in module <code>ironic.db.migration</code> ), 3031
<code>create_ipxe_boot_script()</code> (in module <code>ironic.common.pxe_utils</code> ), 2416	<code>create_schema()</code> (in module <code>ironic.db.sqlalchemy.migration</code> ), 2886
<code>create_isolinux_image_for_bios()</code> (in module <code>ironic.common.images</code> ), 2353	<code>create_schema()</code> ( <code>ironic.cmd.dbsync.DBCommand</code> method), 2226
<code>create_link_without_raise()</code> (in module <code>ironic.common.utils</code> ), 2478	<code>create_schema,</code> <code>ironic-dbsync</code> command line option, 1699
<code>create_node()</code> ( <code>ironic.conductor.manager.ConductorManager</code> method), 2532	<code>create_vfat_image()</code> (in module <code>ironic.common.images</code> ), 2356
<code>create_node()</code> ( <code>ironic.conductor.rpcapi.ConductorAPI</code> method), 2569	<code>create_virtual_disk()</code> (in module <code>ironic.drivers.modules.drac.raid</code> ), 3141
	<code>create_volume_connector()</code> ( <code>ironic.db.api.Connection</code> method), 2921
	<code>create_volume_connector()</code>

[\(ironic.db.sqlalchemy.api.Connection method\), 2772](#)  
[create\\_volume\\_target\(\) \(ironic.db.api.Connection method\), 2923](#)  
[create\\_volume\\_target\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2774](#)  
[created\\_at \(ironic.db.sqlalchemy.models.Allocation attribute\), 2890](#)  
[created\\_at \(ironic.db.sqlalchemy.models.BIOSSetting attribute\), 2891](#)  
[created\\_at \(ironic.db.sqlalchemy.models.Chassis attribute\), 2892](#)  
[created\\_at \(ironic.db.sqlalchemy.models.Conductor attribute\), 2893](#)  
[created\\_at \(ironic.db.sqlalchemy.models.ConductorHardwar attribute\), 2894](#)  
[created\\_at \(ironic.db.sqlalchemy.models.DeployTemplate attribute\), 2894](#)  
[created\\_at \(ironic.db.sqlalchemy.models.DeployTemplate attribute\), 2895](#)  
[created\\_at \(ironic.db.sqlalchemy.models.Node attribute\), 2897](#)  
[created\\_at \(ironic.db.sqlalchemy.models.NodeTag attribute\), 2899](#)  
[created\\_at \(ironic.db.sqlalchemy.models.NodeTrait attribute\), 2900](#)  
[created\\_at \(ironic.db.sqlalchemy.models.Port attribute\), 2901](#)  
[created\\_at \(ironic.db.sqlalchemy.models.Portgroup attribute\), 2902](#)  
[created\\_at \(ironic.db.sqlalchemy.models.VolumeConnecto attribute\), 2903](#)  
[created\\_at \(ironic.db.sqlalchemy.models.VolumeTarget attribute\), 2904](#)  
[created\\_at\(\) \(ironic.objects.allocation.Allocation property\), 4223](#)  
[created\\_at\(\) \(ironic.objects.allocation.AllocationCRUDNotification property\), 4233](#)  
[created\\_at\(\) \(ironic.objects.allocation.AllocationCRUDPayload property\), 4234](#)  
[created\\_at\(\) \(ironic.objects.bios.BIOSSetting property\), 4250](#)  
[created\\_at\(\) \(ironic.objects.bios.BIOSSettingList property\), 4256](#)  
[created\\_at\(\) \(ironic.objects.chassis.Chassis property\), 4263](#)  
[created\\_at\(\) \(ironic.objects.chassis.ChassisCRUDNotification property\), 4271](#)  
[created\\_at\(\) \(ironic.objects.chassis.ChassisCRUDPayload property\), 4272](#)  
[created\\_at\(\) \(ironic.objects.conductor.Conductor property\), 4272](#)  
[created\\_at\(\) \(ironic.objects.deploy\\_template.DeployTemplate property\), 4283](#)  
[created\\_at\(\) \(ironic.objects.deploy\\_template.DeployTemplate property\), 4295](#)  
[created\\_at\(\) \(ironic.objects.deploy\\_template.DeployTemplate property\), 4295](#)  
[created\\_at\(\) \(ironic.objects.deployment.Deployment property\), 4298](#)  
[created\\_at\(\) \(ironic.objects.node.Node property\), 4326](#)  
[created\\_at\(\) \(ironic.objects.node.NodeConsoleNotification property\), 4345](#)  
[created\\_at\(\) \(ironic.objects.node.NodeCorrectedPowerStateN property\), 4347](#)  
[created\\_at\(\) \(ironic.objects.node.NodeCorrectedPowerStateP property\), 4348](#)  
[created\\_at\(\) \(ironic.objects.node.NodeCRUDNotification property\), 4342](#)  
[created\\_at\(\) \(ironic.objects.node.NodeCRUDPayload property\), 4343](#)  
[created\\_at\(\) \(ironic.objects.node.NodeMaintenanceNotificatio property\), 4351](#)  
[created\\_at\(\) \(ironic.objects.node.NodePayload property\), 4352](#)  
[created\\_at\(\) \(ironic.objects.node.NodeSetPowerStateNotificatio property\), 4354](#)  
[created\\_at\(\) \(ironic.objects.node.NodeSetPowerStatePayload property\), 4355](#)  
[created\\_at\(\) \(ironic.objects.node.NodeSetProvisionStateNotif property\), 4358](#)  
[created\\_at\(\) \(ironic.objects.node.NodeSetProvisionStatePaylo property\), 4359](#)  
[created\\_at\(\) \(ironic.objects.notification.EventType property\), 4362](#)  
[created\\_at\(\) \(ironic.objects.notification.NotificationBase property\), 4364](#)  
[created\\_at\(\) \(ironic.objects.notification.NotificationPayloadB property\), 4365](#)  
[created\\_at\(\) \(ironic.objects.notification.NotificationPublisher property\), 4367](#)  
[created\\_at\(\) \(ironic.objects.port.Port property\), 4369](#)  
[created\\_at\(\) \(ironic.objects.port.PortCRUDNotification property\), 4386](#)  
[created\\_at\(\) \(ironic.objects.port.PortCRUDPayload property\), 4387](#)  
[created\\_at\(\) \(ironic.objects.portgroup.Portgroup property\), 4389](#)

created\_at () (*ironic.objects.portgroup.Portgroup* *CRUDNotification* attribute), 4415  
     property), 4402 dbapi (*ironic.objects.volume\_target.VolumeTarget*  
 created\_at () (*ironic.objects.portgroup.Portgroup* *CRUDNotification* attribute), 4431  
     property), 4403 DBCommand (class in *ironic.cmd.dbsync*), 2225  
 created\_at () (*ironic.objects.trait.Trait* property), 4405 DBHook (class in *ironic.api.hooks*), 2218  
 created\_at () (*ironic.objects.trait.TraitList* attribute), 4309 DEBUG (*ironic.objects.fields.NotificationLevel* attribute), 4309  
     property), 4410 deconfigure\_secure\_boot\_if\_needed ()  
 created\_at () (*ironic.objects.volume\_connector.VolumeConnector* module  
     property), 4415 *ironic.drivers.modules.boot\_mode\_utils*),  
 created\_at () (*ironic.objects.volume\_connector.VolumeConnector* *CRUDNotification*  
     property), 4428 default (*ironic.api.functions.FunctionArgument*  
 created\_at () (*ironic.objects.volume\_connector.VolumeConnector* *CRUDPayload*  
     property), 4429 default (*ironic.db.sqlalchemy.models.ConductorHardwareInterf*  
 created\_at () (*ironic.objects.volume\_target.VolumeTarget* attribute), 2894  
     property), 4431 default\_interface () (in module  
 created\_at () (*ironic.objects.volume\_target.VolumeTarget* *CRUDNotification* driver\_factory), 2290  
     property), 4447 default\_version () (in module  
 created\_at () (*ironic.objects.volume\_target.VolumeTarget* *CRUDPayload* controllers.version), 2201  
     property), 4448 del\_host () (*ironic.conductor.base\_manager.BaseConductorManager*  
 CRITICAL (*ironic.objects.fields.NotificationLevel* method), 2511  
     attribute), 4309 delete () (*ironic.api.controllers.v1.allocation.AllocationsController*  
     method), 1958  
**D** delete () (*ironic.api.controllers.v1.allocation.NodeAllocationController*  
     method), 1964  
 DatabaseVersionTooOld, 2303 delete () (*ironic.api.controllers.v1.chassis.ChassisController*  
     method), 1966  
 datatype (*ironic.api.functions.FunctionArgument* attribute), 2208 delete () (*ironic.api.controllers.v1.deploy\_template.DeployTemplateController*  
     method), 1979  
 DateTimeField (class in *ironic.objects.fields*), 4304 delete () (*ironic.api.controllers.v1.node.NodeMaintenanceController*  
     method), 2007  
 dbapi (*ironic.objects.allocation.Allocation* attribute), 4223 delete () (*ironic.api.controllers.v1.node.NodesController*  
     method), 2030  
 dbapi (*ironic.objects.bios.BIOSSetting* attribute), 4250 delete () (*ironic.api.controllers.v1.node.NodeTraitsController*  
     method), 2025  
 dbapi (*ironic.objects.bios.BIOSSettingList* attribute), 4256 delete () (*ironic.api.controllers.v1.node.NodeVIFController*  
     method), 2026  
 dbapi (*ironic.objects.chassis.Chassis* attribute), 4263 delete () (*ironic.api.controllers.v1.port.PortsController*  
     method), 2057  
 dbapi (*ironic.objects.conductor.Conductor* attribute), 4273 delete () (*ironic.api.controllers.v1.portgroup.PortgroupsController*  
     method), 2068  
 dbapi (*ironic.objects.deploy\_template.DeployTemplate* attribute), 4283 delete () (*ironic.api.controllers.v1.volume\_connector.VolumeConnector*  
     method), 2168  
 dbapi (*ironic.objects.deployment.Deployment* attribute), 4298 delete () (*ironic.api.controllers.v1.volume\_target.VolumeTarget*  
     method), 2183  
 dbapi (*ironic.objects.node.Node* attribute), 4326 delete () (*ironic.objects.bios.BIOSSetting* class  
     method), 4250  
 dbapi (*ironic.objects.port.Port* attribute), 4369 delete () (*ironic.objects.bios.BIOSSettingList*  
     class method), 4256  
 dbapi (*ironic.objects.portgroup.Portgroup* attribute), 4389 DELETE\_ALLOWED\_STATES (in module  
     method), 4250  
 dbapi (*ironic.objects.trait.Trait* attribute), 4405 delete () (*ironic.objects.bios.BIOSSettingList*  
     class method), 4256  
 dbapi (*ironic.objects.trait.TraitList* attribute), 4410  
 dbapi (*ironic.objects.volume\_connector.VolumeConnector* *ironic.common.states*), 2455



<code>delete_bios_setting_list()</code> ( <i>ironic.db.api.Connection</i> <i>method</i> ), 2926	DELETING (in module <i>ironic.common.states</i> ), 2455
<code>delete_bios_setting_list()</code> ( <i>ironic.db.sqlalchemy.api.Connection</i> <i>method</i> ), 2777	<code>deploy</code> ( <i>ironic.drivers.base.BareDriver</i> <i>attribute</i> ), 4035
<code>delete_configuration()</code> ( <i>ironic.drivers.base.RAIDInterface</i> <i>method</i> ), 4148	<code>deploy()</code> ( <i>ironic.drivers.base.DeployInterface</i> <i>method</i> ), 4064
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.agent.AgentRAID</i> <i>method</i> ), 3593	<code>deploy()</code> ( <i>ironic.drivers.modules.agent.AgentDeploy</i> <i>method</i> ), 3581
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.drac.raid.DracWSManRAID</i> <i>method</i> ), 3132	<code>deploy()</code> ( <i>ironic.drivers.modules.ansible.deploy.AnsibleDeploy</i> <i>method</i> ), 3054
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.fake.FakeRAID</i> <i>method</i> ), 3824	<code>deploy()</code> ( <i>ironic.drivers.modules.fake.FakeDeploy</i> <i>method</i> ), 3777
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.ibm.raid.IbmCRRAID</i> <i>method</i> ), 3196	<code>deploy()</code> ( <i>ironic.drivers.modules.iscsi_deploy.ISCSIDeploy</i> <i>method</i> ), 3914
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.ilo.raid.Ilo5RAID</i> <i>method</i> ), 3344	<code>deploy()</code> ( <i>ironic.drivers.modules.pxe.PXERamdiskDeploy</i> <i>method</i> ), 3984
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.irmc.raid.IRMCRRAID</i> <i>method</i> ), 3415	<code>deploy_disk_image()</code> (in module <i>ironic.drivers.modules.iscsi_deploy</i> ), 3928
<code>delete_configuration()</code> ( <i>ironic.drivers.modules.noop.NoRAID</i> <i>method</i> ), 3961	<code>deploy_has_started()</code> ( <i>ironic.drivers.modules.agent_base.HeartbeatMixin</i> <i>method</i> ), 3629
<code>delete_iscsi()</code> (in module <i>ironic.drivers.modules.iscsi_deploy</i> ), 3928	<code>deploy_interface</code> ( <i>ironic.db.sqlalchemy.models.Node</i> <i>attribute</i> ), 2897
<code>delete_node_tag()</code> ( <i>ironic.db.api.Connection</i> <i>method</i> ), 2927	<code>deploy_interface()</code> ( <i>ironic.objects.node.Node</i> <i>property</i> ), 4326
<code>delete_node_tag()</code> ( <i>ironic.db.sqlalchemy.api.Connection</i> <i>method</i> ), 2778	<code>deploy_interface()</code> ( <i>ironic.objects.node.NodeCorrectedPowerStatePayload</i> <i>property</i> ), 4348
<code>delete_node_trait()</code> ( <i>ironic.db.api.Connection</i> <i>method</i> ), 2928	<code>deploy_interface()</code> ( <i>ironic.objects.node.NodeCRUDPayload</i> <i>property</i> ), 4343
<code>delete_node_trait()</code> ( <i>ironic.db.sqlalchemy.api.Connection</i> <i>method</i> ), 2779	<code>deploy_interface()</code> ( <i>ironic.objects.node.NodePayload</i> <i>prop-</i> <i>erty</i> ), 4352
<code>delete_object()</code> ( <i>ironic.common.swift.SwiftAPI</i> <i>method</i> ), 2470	<code>deploy_interface()</code> ( <i>ironic.objects.node.NodeSetPowerStatePayload</i> <i>property</i> ), 4356
<code>delete_virtual_disk()</code> (in module <i>ironic.drivers.modules.drac.raid</i> ), 3144	<code>deploy_interface()</code> ( <i>ironic.objects.node.NodeSetProvisionStatePayload</i> <i>property</i> ), 4359
DELETED (in module <i>ironic.common.states</i> ), 2454	<code>deploy_is_done()</code> ( <i>ironic.drivers.modules.agent_base.HeartbeatMixin</i> <i>method</i> ), 3629
	<code>deploy_partition_image()</code> (in module <i>ironic.drivers.modules.iscsi_deploy</i> ), 3931
	<code>deploy_step</code> ( <i>ironic.db.sqlalchemy.models.Node</i> <i>attribute</i> ), 2897

<code>deploy_step()</code>	(in module <a href="#">ironic.drivers.base</a> ), 4175	<code>DeployTemplateStep</code>	(class in <a href="#">ironic.db.sqlalchemy.models</a> ), 2895
<code>deploy_step()</code>	( <a href="#">ironic.objects.node.Node</a> property), 4326	<code>DEPLOYWAIT</code>	(in module <a href="#">ironic.common.states</a> ), 2457
<code>deploy_step()</code>	( <a href="#">ironic.objects.node.NodeCorrectedPowerStatePayload</a> property), 4348	<code>description</code>	( <a href="#">ironic.db.sqlalchemy.models.Chassis</a> attribute), 2892
<code>deploy_step()</code>	( <a href="#">ironic.objects.node.NodeCRUDPayload</a> property), 4343	<code>description()</code>	( <a href="#">ironic.db.sqlalchemy.models.Node</a> attribute), 2897
<code>deploy_step()</code>	( <a href="#">ironic.objects.node.NodePayload</a> property), 4352	<code>description()</code>	( <a href="#">ironic.objects.chassis.Chassis</a> property), 4263
<code>deploy_step()</code>	( <a href="#">ironic.objects.node.NodeSetPowerStatePayload</a> property), 4356	<code>description()</code>	( <a href="#">ironic.objects.chassis.ChassisCRUDPayload</a> property), 4272
<code>deploy_step()</code>	( <a href="#">ironic.objects.node.NodeSetProvisionStatePayload</a> property), 4359	<code>description()</code>	( <a href="#">ironic.objects.node.Node</a> property), 4326
<code>deploy_template</code>	( <a href="#">ironic.db.sqlalchemy.models.DeployTemplateStep</a> attribute), 2895	<code>description()</code>	( <a href="#">ironic.objects.node.NodeCorrectedPowerStatePayload</a> property), 4348
<code>deploy_template_id</code>	( <a href="#">ironic.db.sqlalchemy.models.DeployTemplateStep</a> attribute), 2895	<code>deserialize_context()</code>	( <a href="#">ironic.common.rpc.RequestContextSerializer</a> method), 2444
<code>DEPLOYDONE</code>	(in module <a href="#">ironic.common.states</a> ), 2455	<code>deserialize_entity()</code>	( <a href="#">ironic.common.rpc.RequestContextSerializer</a> method), 2445
<code>DEPLOYFAIL</code>	(in module <a href="#">ironic.common.states</a> ), 2456	<code>destroy()</code>	( <a href="#">ironic.objects.allocation.Allocation</a> method), 4223
<code>DEPLOYING</code>	(in module <a href="#">ironic.common.states</a> ), 2456	<code>destroy()</code>	( <a href="#">ironic.objects.chassis.Chassis</a> method), 4263
<code>deploying_error_handler()</code>	(in module <a href="#">ironic.conductor.utils</a> ), 2710	<code>destroy()</code>	( <a href="#">ironic.objects.deploy_template.DeployTemplate</a> method), 4283
<code>DeployInterface</code>	(class in <a href="#">ironic.drivers.base</a> ), 4062	<code>destroy()</code>	( <a href="#">ironic.objects.deployment.Deployment</a> method), 4298
<code>Deployment</code>	(class in <a href="#">ironic.objects.deployment</a> ), 4296	<code>destroy()</code>	( <a href="#">ironic.objects.node.Node</a> method), 4326
<code>DeployTemplate</code>	(class in <a href="#">ironic.db.sqlalchemy.models</a> ), 2894	<code>destroy()</code>	( <a href="#">ironic.objects.port.Port</a> method), 4369
<code>DeployTemplate</code>	(class in <a href="#">ironic.objects.deploy_template</a> ), 4281	<code>destroy()</code>	( <a href="#">ironic.objects.portgroup.Portgroup</a> method), 4369
<code>DeployTemplateAlreadyExists</code>	2303		
<code>DeployTemplateCRUDNotification</code>	(class in <a href="#">ironic.objects.deploy_template</a> ), 4294		
<code>DeployTemplateCRUDPayload</code>	(class in <a href="#">ironic.objects.deploy_template</a> ), 4295		
<code>DeployTemplateDuplicateName</code>	2304		
<code>DeployTemplateNotFound</code>	2304		
<code>DeployTemplatesController</code>	(class in <a href="#">ironic.api.controllers.v1.deploy_template</a> ), 4369		

method), 4389

destroy() (ironic.objects.trait.Trait class method), 4405

destroy() (ironic.objects.trait.TraitList class method), 4410

destroy() (ironic.objects.volume\_connector.VolumeConnector class method), 4415

destroy() (ironic.objects.volume\_target.VolumeTarget class method), 4432

destroy\_allocation() (ironic.conductor.manager.ConductorManager class method), 2532

destroy\_allocation() (ironic.conductor.rpcapi.ConductorAPI class method), 2573

destroy\_allocation() (ironic.db.api.Connection class method), 2929

destroy\_allocation() (ironic.db.sqlalchemy.api.Connection class method), 2780

destroy\_chassis() (ironic.db.api.Connection class method), 2929

destroy\_chassis() (ironic.db.sqlalchemy.api.Connection class method), 2780

destroy\_deploy\_template() (ironic.db.api.Connection class method), 2929

destroy\_deploy\_template() (ironic.db.sqlalchemy.api.Connection class method), 2780

destroy\_floppy\_image\_from\_web\_server() (in module ironic.drivers.modules.ilo.common), 3258

destroy\_http\_instance\_images() (in module ironic.drivers.modules.deploy\_utils), 3716

destroy\_images() (in module ironic.drivers.modules.deploy\_utils), 3717

destroy\_node() (ironic.conductor.manager.ConductorManager class method), 2532

destroy\_node() (ironic.conductor.rpcapi.ConductorAPI class method), 2574

destroy\_node() (ironic.db.api.Connection class method), 2930

destroy\_node() (ironic.db.sqlalchemy.api.Connection class method), 2781

destroy\_port() (ironic.conductor.manager.ConductorManager class method), 2532

destroy\_port() (ironic.conductor.rpcapi.ConductorAPI class method), 2576

destroy\_port() (ironic.db.api.Connection class method), 2931

destroy\_port() (ironic.db.sqlalchemy.api.Connection class method), 2782

destroy\_portgroup() (ironic.conductor.manager.ConductorManager class method), 2532

destroy\_portgroup() (ironic.conductor.rpcapi.ConductorAPI class method), 2577

destroy\_portgroup() (ironic.db.api.Connection class method), 2931

destroy\_portgroup() (ironic.db.sqlalchemy.api.Connection class method), 2782

destroy\_volume\_connector() (ironic.conductor.manager.ConductorManager class method), 2532

destroy\_volume\_connector() (ironic.conductor.rpcapi.ConductorAPI class method), 2578

destroy\_volume\_connector() (ironic.db.api.Connection class method), 2932

destroy\_volume\_connector() (ironic.db.sqlalchemy.api.Connection class method), 2783

destroy\_volume\_target() (ironic.conductor.manager.ConductorManager class method), 2533

destroy\_volume\_target() (ironic.conductor.rpcapi.ConductorAPI class method), 2580

destroy\_volume\_target() (ironic.db.api.Connection class method), 2933

destroy\_volume\_target() (ironic.db.sqlalchemy.api.Connection class method), 2784

detach_volumes()	(in module <i>ironic.common.cinder</i> ), 2270	discovery()	(in module <i>ironic.drivers.modules.iscsi_deploy</i> ), 3936
detach_volumes()	( <i>ironic.drivers.base.StorageInterface</i> method), 4160	DISK	(in module <i>ironic.common.boot_devices</i> ), 2260
detach_volumes()	( <i>ironic.drivers.modules.fake.FakeStorage</i> method), 3832	DISK	(in module <i>ironic.common.components</i> ), 2277
detach_volumes()	( <i>ironic.drivers.modules.storage.cinder.CinderStorage</i> method), 3548	do_agent_iscsi_deploy()	(in module <i>ironic.drivers.modules.iscsi_deploy</i> ), 3937
detach_volumes()	( <i>ironic.drivers.modules.storage.external.ExternalStorage</i> method), 3554	do_allocate()	(in module <i>ironic.conductor.allocations</i> ), 2508
detach_volumes()	( <i>ironic.drivers.modules.storage.noop.NoopStorage</i> method), 3559	do_clean_step()	(in module <i>ironic.conductor.cleaning</i> ), 2517
detail()	( <i>ironic.api.controllers.v1.chassis.ChassisController</i> method), 1966	do_next_deploy_step()	(in module <i>ironic.conductor.deployments</i> ), 2521
detail()	( <i>ironic.api.controllers.v1.node.NodesController</i> method), 2030	do_node_clean()	(in module <i>ironic.conductor.cleaning</i> ), 2518
detail()	( <i>ironic.api.controllers.v1.port.PortsController</i> method), 2057	do_node_clean()	( <i>ironic.conductor.manager.ConductorManager</i> method), 2533
detail()	( <i>ironic.api.controllers.v1.portgroup.PortgroupsController</i> method), 2068	do_node_clean()	( <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2582
detect_vendor()	( <i>ironic.drivers.base.ManagementInterface</i> method), 4078	do_node_clean_abort()	(in module <i>ironic.conductor.cleaning</i> ), 2519
detect_vendor()	( <i>ironic.drivers.modules.ipmitool.IPMIManagement</i> method), 3874	do_node_deploy()	(in module <i>ironic.conductor.deployments</i> ), 2522
detect_vendor()	( <i>ironic.drivers.modules.redfish.management.RedfishManagement</i> method), 3494	do_node_deploy()	( <i>ironic.conductor.manager.ConductorManager</i> method), 2533
dhcp_options_for_instance()	(in module <i>ironic.common.pxe_utils</i> ), 2418	do_node_deploy()	( <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2585
DHCPFactory	(class in <i>ironic.common.dhcp_factory</i> ), 2280	do_node_rescue()	( <i>ironic.conductor.manager.ConductorManager</i> method), 2533
DHCPLoadError	, 2303	do_node_rescue()	( <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2587
dict_valid()	(in module <i>ironic.common.args</i> ), 2244	do_node_tear_down()	( <i>ironic.conductor.manager.ConductorManager</i> method), 2533
direct_deploy_should_convert_raw_image()	(in module <i>ironic.drivers.modules.deploy_utils</i> ), 3717	do_node_tear_down()	( <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2589
DirectoryNotWritable	, 2304	do_node_unrescue()	( <i>ironic.conductor.manager.ConductorManager</i> method), 2533
disable_secure_boot_if_supported()	(in module <i>ironic.drivers.modules.ilo.boot</i> ), 3245	do_node_unrescue()	( <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2533
disallowed_fields()	(in module <i>ironic.api.controllers.v1.utils</i> ), 2126		

method), 2592

do\_provisioning\_action() (ironic.conductor.manager.ConductorManager method), 2533

do\_provisioning\_action() (ironic.conductor.rpcapi.ConductorAPI method), 2593

do\_sync\_power\_state() (in module ironic.conductor.manager), 2540

do\_version\_changes\_for\_db() (ironic.objects.base.IronicObject method), 4239

doc (ironic.api.functions.FunctionDefinition attribute), 2210

downgrade() (in module ironic.db.sqlalchemy.migration), 2887

downgrade\_lock() (ironic.conductor.task\_manager.TaskManager method), 2693

download() (ironic.common.glance\_service.image\_service.ImageService method), 2230

download() (ironic.common.image\_service.BaseImageService method), 2331

download() (ironic.common.image\_service.FileImageService method), 2334

download() (ironic.common.image\_service.HttpImageService method), 2337

download\_size() (in module ironic.common.images), 2359

DracInspect (class in ironic.drivers.modules.drac.inspect), 3093

DracManagement (class in ironic.drivers.modules.drac.management), 3102

DracOperationError, 2304

DracPower (class in ironic.drivers.modules.drac.power), 3116

DracRAID (class in ironic.drivers.modules.drac.raid), 3125

DracRedfishBIOS (class in ironic.drivers.modules.drac.bios), 3063

DracRedfishInspect (class in ironic.drivers.modules.drac.inspect), 3094

DracRedfishManagement (class in ironic.drivers.modules.drac.management), 3103

DracRedfishPower (class in ironic.drivers.modules.drac.power), 3117

DracRedfishVendorPassthru (class in ironic.drivers.modules.drac.vendor\_passthru), 3152

DracRedfishVirtualMediaBoot (class in ironic.drivers.modules.drac.boot), 3085

DracVendorPassthru (class in ironic.drivers.modules.drac.vendor\_passthru), 3152

DracWSManBIOS (class in ironic.drivers.modules.drac.bios), 3063

DracWSManInspect (class in ironic.drivers.modules.drac.inspect), 3095

DracWSManManagement (class in ironic.drivers.modules.drac.management), 3104

DracWSManPower (class in ironic.drivers.modules.drac.power), 3118

DracWSManRAID (class in ironic.drivers.modules.drac.raid), 3126

DracWSManVendorPassthru (class in ironic.drivers.modules.drac.vendor\_passthru), 3154

driver() (ironic.db.sqlalchemy.models.Node attribute), 2897

driver() (ironic.objects.node.Node property), 4327

driver() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4348

driver() (ironic.objects.node.NodeCRUDPayload property), 4343

driver() (ironic.objects.node.NodePayload property), 4352

driver() (ironic.objects.node.NodeSetPowerStatePayload property), 4356

driver() (ironic.objects.node.NodeSetProvisionStatePayload property), 4359

driver\_info (ironic.db.sqlalchemy.models.Node attribute), 2897

driver\_info() (ironic.objects.node.Node property), 4327

driver\_info() (ironic.objects.node.NodeCRUDPayload property), 4343

driver\_internal\_info (ironic.db.sqlalchemy.models.Node attribute), 2897

driver\_internal\_info() (ironic.objects.node.Node property), 4327



4327 *method*), 4364

`driver_internal_info()` (*in module* `emit_console_notification()` (*in module* `ironic.objects.node.NodeSetProvisionStatePayload` *ule* `ironic.conductor.notification_utils`), 2544

`driver_passthru()` (*in module* `emit_end_notification()` (*in module* `ironic.drivers.base`), 4180 `ironic.api.controllers.v1.notification_utils`), 2053

`driver_validate()` (*in module* `emit_power_set_notification()` (*in module* `ironic.drivers.base.VendorInterface` *method*), 4163 `ironic.conductor.notification_utils`), 2546

`driver_validate()` (*in module* `emit_power_state_corrected_notification()` (*in module* `ironic.drivers.modules.noop.NoVendor` *method*), 3968 `ironic.conductor.notification_utils`), 2547

`driver_vendor_passthru()` (*in module* `emit_provision_set_notification()` (*in module* `ironic.conductor.manager.ConductorManager` *method*), 2533 `ironic.conductor.notification_utils`), 2549

`driver_vendor_passthru()` (*in module* `emit_start_notification()` (*in module* `ironic.conductor.rpcapi.ConductorAPI` *method*), 2596 `ironic.conductor.notification_utils`), 2054

`DriverLoadError`, 2304

`DriverNotFound`, 2304

`DriverNotFoundInEntrypoint`, 2304

`DriverOperationError`, 2305

`DriverPassthruController` (*class in module* `enabled_supported_interfaces()` (*in module* `ironic.api.controllers.v1.driver`), 1986 `ironic.common.driver_factory`), 2293

`DriverRaidController` (*class in module* `END` (*in module* `ironic.api.controllers.v1.driver`), 1988 `ironic.objects.fields.NotificationStatus` *attribute*), 4309

`drivers` (*in module* `ENROLL` (*in module* `ironic.db.sqlalchemy.models.Conductor` *attribute*), 2893 `ironic.common.states`), 2457

`drivers()` (*in module* `ensure_next_boot_device()` (*in module* `ironic.objects.conductor.Conductor` *property*), 4273 `ironic.drivers.utils`), 4213

`DriversController` (*class in module* `ensure_thread_contain_context()` (*in module* `ironic.api.controllers.v1.driver`), 1990 `ironic.common.context.RequestContext` *method*), 2278

`dump_sdr()` (*in module* `EnumField` (*class in module* `ironic.drivers.modules.ipmitool`), 3909 `ironic.objects.fields`), 4304

`Duplicate`, 2305

`duplicate_steps()` (*in module* `erase_devices()` (*in module* `ironic.api.controllers.v1.deploy_template`), 1984 `ironic.drivers.modules.ilo.management.Ilo5Management` *method*), 3301

`DuplicateName`, 2305

`ERROR` (*in module* `ironic.common.states`), 2458

`ERROR` (*in module* `ironic.objects.fields.NotificationLevel` *attribute*), 4309

`ERROR` (*in module* `ironic.objects.fields.NotificationStatus` *attribute*), 4309

`EJECT_VMEDIA` (*in module* `ESSENTIAL_PROPERTIES` (*in module* `ironic.drivers.modules.redfish.boot`), 3487 `ironic.drivers.base.InspectInterface` *attribute*), 4074

`eject_vmedia()` (*in module* `emit_console_notification()` (*in module* `ironic.drivers.modules.redfish.vendor.RedfishVendorPassthru` *method*), 3544 `ironic.objects.node.NodeSetProvisionStatePayload` *property*), 4359

`eject_vmedia_devices()` (*in module* `event_type()` (*in module* `ironic.drivers.modules.ilo.common`), 3258 `ironic.objects.allocation.AllocationCRUDNotification` *property*), 4233

`emit()` (*in module* `event_type()` (*in module* `ironic.objects.notification.NotificationBase` *property*), 4271 `ironic.objects.chassis.ChassisCRUDNotification` *property*), 4271

[event\\_type\(\)](#) ([ironic.objects.deploy\\_template.DeployTemplateCRUDNotification](#) [property](#)), [4295](#) [\(ironic.drivers.modules.agent\\_base.AgentDeployMixin](#)  
[event\\_type\(\)](#) ([ironic.objects.node.NodeConsoleNotification](#) [method](#)), [3617](#)  
[property](#)), [4345](#) [execute\\_deploy\\_step\(\)](#)  
[event\\_type\(\)](#) ([ironic.objects.node.NodeCorrectedPowerStateNotification](#) [property](#)), [4347](#) [\(ironic.drivers.modules.agent\\_client.AgentClient](#)  
[method](#)), [3650](#)  
[event\\_type\(\)](#) ([ironic.objects.node.NodeCRUDNotification](#) [property](#)), [4342](#) [execute\\_step\(\)](#) (in module  
[ironic.drivers.modules.agent\\_base](#)),  
[event\\_type\(\)](#) ([ironic.objects.node.NodeMaintenanceNotification](#) [property](#)), [4351](#) [exists\(\)](#) ([ironic.objects.trait.Trait](#) [class](#)  
[event\\_type\(\)](#) ([ironic.objects.node.NodeSetPowerStateNotification](#) [method](#)), [4406](#)  
[property](#)), [4354](#) [expose\(\)](#) (in module [ironic.api.method](#)), [2223](#)  
[event\\_type\(\)](#) ([ironic.objects.node.NodeSetProvisionStateNotification](#) [property](#)), [4358](#) [\(class in](#)  
[ironic.drivers.modules.storage.external](#)),  
[event\\_type\(\)](#) ([ironic.objects.notification.NotificationBase](#) [property](#)), [4365](#) [extra](#) ([ironic.db.sqlalchemy.models.Allocation](#)  
[event\\_type\(\)](#) ([ironic.objects.port.PortCRUDNotification](#) [attribute](#)), [2890](#)  
[property](#)), [4386](#) [extra](#) ([ironic.db.sqlalchemy.models.Chassis](#) [at-](#)  
[event\\_type\(\)](#) ([ironic.objects.portgroup.PortgroupCRUDNotification](#) [attribute](#)), [2892](#)  
[property](#)), [4402](#) [extra](#) ([ironic.db.sqlalchemy.models.DeployTemplate](#)  
[event\\_type\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnectorCRUDNotification](#) [property](#)), [4428](#) [extra](#) ([ironic.db.sqlalchemy.models.Node](#) [at-](#)  
[event\\_type\(\)](#) ([ironic.objects.volume\\_target.VolumeTargetCRUDNotification](#) [property](#)), [4447](#) [extra](#) ([ironic.db.sqlalchemy.models.Port](#) [at-](#)  
[events\\_valid\(\)](#) (in module [ironic.api.controllers.v1.event](#)), [1995](#) [attribute](#)), [2901](#)  
[EventsController](#) (class in [ironic.api.controllers.v1.event](#)), [1995](#) [extra](#) ([ironic.db.sqlalchemy.models.Portgroup](#)  
[attribute](#)), [2902](#)  
[EventType](#) (class in [ironic.objects.notification](#)), [4362](#) [extra](#) ([ironic.db.sqlalchemy.models.VolumeConnector](#)  
[attribute](#)), [2903](#)  
[ExclusiveLockRequired](#), [2305](#) [extra](#) ([ironic.db.sqlalchemy.models.VolumeTarget](#)  
[attribute](#)), [2904](#)  
[execute\(\)](#) (in module [ironic.common.utils](#)), [2478](#) [extra\(\)](#) ([ironic.objects.allocation.Allocation](#)  
[property](#)), [4224](#)  
[execute\\_clean\\_step\(\)](#) (in module [ironic.drivers.modules.agent\\_base](#)), [3634](#) [extra\(\)](#) ([ironic.objects.allocation.AllocationCRUDPayload](#)  
[property](#)), [4234](#)  
[execute\\_clean\\_step\(\)](#) ([ironic.drivers.base.BaseInterface](#) [method](#)), [4040](#) [extra\(\)](#) ([ironic.objects.chassis.Chassis](#) [prop-](#)  
[property](#)), [4264](#)  
[execute\\_clean\\_step\(\)](#) [extra\(\)](#) ([ironic.objects.chassis.ChassisCRUDPayload](#)  
[property](#)), [4272](#)  
[execute\\_clean\\_step\(\)](#) [extra\(\)](#) ([ironic.objects.deploy\\_template.DeployTemplate](#)  
[property](#)), [4284](#)  
[execute\\_clean\\_step\(\)](#) [extra\(\)](#) ([ironic.objects.deploy\\_template.DeployTemplateCRUDPayload](#)  
[property](#)), [4296](#)  
[execute\\_clean\\_step\(\)](#) ([ironic.drivers.modules.agent\\_client.AgentClient](#) [method](#)), [3647](#) [extra\(\)](#) ([ironic.objects.node.Node](#) [property](#)),  
[4327](#)  
[execute\\_clean\\_step\(\)](#) [extra\(\)](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#)  
[property](#)), [4349](#)  
[execute\\_clean\\_step\(\)](#) ([ironic.drivers.modules.ansible.deploy.AnsibleDeploy](#) [method](#)), [3054](#) [extra\(\)](#) ([ironic.objects.node.NodeCRUDPayload](#)  
[property](#)), [4343](#)  
[execute\\_deploy\\_step\(\)](#) ([ironic.drivers.base.BaseInterface](#) [method](#)), [4042](#) [extra\(\)](#) ([ironic.objects.node.NodePayload](#)  
[property](#)), [4352](#)

[extra\(\) \(ironic.objects.node.NodeSetPowerStatePayload property\), 4356](#)  
[extra\(\) \(ironic.objects.node.NodeSetProvisionStatePayload property\), 4360](#)  
[extra\(\) \(ironic.objects.port.Port property\), 4370](#)  
[extra\(\) \(ironic.objects.port.PortCRUDPayload property\), 4387](#)  
[extra\(\) \(ironic.objects.portgroup.Portgroup property\), 4390](#)  
[extra\(\) \(ironic.objects.portgroup.PortgroupCRUDPayload property\), 4403](#)  
[extra\(\) \(ironic.objects.volume\\_connector.VolumeConnector property\), 4416](#)  
[extra\(\) \(ironic.objects.volume\\_connector.VolumeConnectorCRUDPayload property\), 4429](#)  
[extra\(\) \(ironic.objects.volume\\_target.VolumeTarget property\), 4432](#)  
[extra\(\) \(ironic.objects.volume\\_target.VolumeTargetCRUDPayload property\), 4448](#)  
[extra\\_options \(ironic.api.functions.FunctionDefinition attribute\), 2210](#)  
**F**  
[factory\\_reset\(\) \(ironic.drivers.base.BIOSInterface method\), 4031](#)  
[factory\\_reset\(\) \(ironic.drivers.modules.drac.bios.DracWSManBIOS method\), 3066](#)  
[factory\\_reset\(\) \(ironic.drivers.modules.fake.FakeBIOS method\), 3758](#)  
[factory\\_reset\(\) \(ironic.drivers.modules.ilo.bios.IloBIOS method\), 3209](#)  
[factory\\_reset\(\) \(ironic.drivers.modules.irmc.bios.IRMCBIOS method\), 3355](#)  
[factory\\_reset\(\) \(ironic.drivers.modules.noop.NoBIOS method\), 3951](#)  
[factory\\_reset\(\) \(ironic.drivers.modules.redfish.bios.RedfishBIOS method\), 3468](#)  
[fail\\_on\\_error\(\) \(in module ironic.conductor.utils\), 2711](#)  
[FailedToCleanDHCPOpts, 2305](#)  
[FailedToGetIPAddressOnPort, 2305](#)  
[FailedToGetSensorData, 2305](#)  
[FailedToParseSensorData, 2305](#)  
[FailedToUpdateDHCPOnPort, 2306](#)  
[FailedToUpdateMacOnPort, 2306](#)  
[FailMixin \(class in ironic.drivers.modules.noop\), 3946](#)  
[FakeBIOS \(class in ironic.drivers.modules.fake\), 3753](#)  
[FakeBoot \(class in ironic.drivers.modules.fake\), 3762](#)  
[FakeConsole \(class in ironic.drivers.modules.fake\), 3769](#)  
[FakeDeploy \(class in ironic.drivers.modules.fake\), 3775](#)  
[FakeHardware \(class in ironic.drivers.modules.fake\\_hardware\), 4182](#)  
[FakeInspect \(class in ironic.drivers.modules.fake\), 3785](#)  
[FakeManagement \(class in ironic.drivers.modules.fake\), 3790](#)  
[FakePower \(class in ironic.drivers.modules.fake\), 3813](#)  
[FakeRAID \(class in ironic.drivers.modules.fake\), 3821](#)  
[FakeRescue \(class in ironic.drivers.modules.fake\), 3826](#)  
[FakeStorage \(class in ironic.drivers.modules.fake\), 3831](#)  
[FakeVendorA \(class in ironic.drivers.modules.fake\), 3837](#)  
[FakeVendorB \(class in ironic.drivers.modules.fake\), 3839](#)  
[fast\\_track\\_able\(\) \(in module ironic.conductor.utils\), 2711](#)  
[FASTTRACK\\_LOOKUP\\_ALLOWED\\_STATES \(in module ironic.common.states\), 2458](#)  
[fault \(ironic.db.sqlalchemy.models.Node attribute\), 2897](#)  
[fault\(\) \(ironic.objects.node.Node property\), 4327](#)  
[fault\(\) \(ironic.objects.node.NodeCorrectedPowerStatePayload property\), 4349](#)  
[fault\(\) \(ironic.objects.node.NodeCRUDPayload property\), 4343](#)  
[fault\(\) \(ironic.objects.node.NodePayload property\), 4352](#)  
[fault\(\) \(ironic.objects.node.NodeSetPowerStatePayload property\), 4356](#)  
[fault\(\) \(ironic.objects.node.NodeSetProvisionStatePayload property\), 4360](#)  
[faultstring\(\) \(ironic.common.exception.ClientSideError](#)



property), 2302

faultstring() (ironic.common.exception.InvalidInput property), 2311

faultstring() (ironic.common.exception.UnknownArgument property), 2320

faultstring() (ironic.common.exception.UnknownAttribute property), 2321

fetch() (in module ironic.common.images), 2359

fetch\_image() (ironic.drivers.modules.image\_cache.ImageCache method), 3844

fetch\_images() (in module ironic.drivers.modules.deploy\_utils), 3718

fetch\_into() (in module ironic.common.images), 2359

fields (ironic.objects.allocation.Allocation attribute), 4224

fields (ironic.objects.allocation.AllocationCRUDNotification attribute), 4234

fields (ironic.objects.allocation.AllocationCRUDPayload attribute), 4234

fields (ironic.objects.base.IronicObject attribute), 4242

fields (ironic.objects.bios.BIOSSetting attribute), 4252

fields (ironic.objects.bios.BIOSSettingList attribute), 4258

fields (ironic.objects.chassis.Chassis attribute), 4264

fields (ironic.objects.chassis.ChassisCRUDNotification attribute), 4271

fields (ironic.objects.chassis.ChassisCRUDPayload attribute), 4272

fields (ironic.objects.conductor.Conductor attribute), 4273

fields (ironic.objects.deploy\_template.DeployTemplate attribute), 4284

fields (ironic.objects.deploy\_template.DeployTemplateCRUDNotification attribute), 4295

fields (ironic.objects.deploy\_template.DeployTemplateCRUDPayload attribute), 4296

fields (ironic.objects.deployment.Deployment attribute), 4299

fields (ironic.objects.node.Node attribute), 4327

fields (ironic.objects.node.NodeConsoleNotification attribute), 4345

fields (ironic.objects.node.NodeCorrectedPowerStateNotification attribute), 4347

fields (ironic.objects.node.NodeCorrectedPowerStatePayload attribute), 4349

fields (ironic.objects.node.NodeCRUDNotification attribute), 4342

fields (ironic.objects.node.NodeCRUDPayload attribute), 4343

fields (ironic.objects.node.NodeMaintenanceNotification attribute), 4351

fields (ironic.objects.node.NodePayload attribute), 4352

fields (ironic.objects.node.NodeSetPowerStateNotification attribute), 4355

fields (ironic.objects.node.NodeSetPowerStatePayload attribute), 4356

fields (ironic.objects.node.NodeSetProvisionStateNotification attribute), 4358

fields (ironic.objects.node.NodeSetProvisionStatePayload attribute), 4360

fields (ironic.objects.notification.EventType attribute), 4362

fields (ironic.objects.notification.NotificationBase attribute), 4365

fields (ironic.objects.notification.NotificationPayloadBase attribute), 4365

fields (ironic.objects.notification.NotificationPublisher attribute), 4367

fields (ironic.objects.port.Port attribute), 4370

fields (ironic.objects.port.PortCRUDNotification attribute), 4386

fields (ironic.objects.port.PortCRUDPayload attribute), 4387

fields (ironic.objects.portgroup.Portgroup attribute), 4390

fields (ironic.objects.portgroup.PortgroupCRUDNotification attribute), 4402

fields (ironic.objects.portgroup.PortgroupCRUDPayload attribute), 4403

fields (ironic.objects.trait.Trait attribute), 4408

fields (ironic.objects.trait.TraitList attribute), 4408

fields (ironic.objects.volume\_connector.VolumeConnector attribute), 4416

fields (ironic.objects.volume\_connector.VolumeConnectorCRUDNotification attribute), 4428

fields (ironic.objects.volume\_connector.VolumeConnectorCRUDPayload attribute), 4429

fields (ironic.objects.volume\_target.VolumeTarget attribute), 4433

fields (*ironic.objects.volume\_target.VolumeTargetCRUDNotification* attribute), 2223  
 fields (*ironic.objects.volume\_target.VolumeTargetCRUDNotification* attribute), 4447  
 file\_has\_content() (in module *ironic.common.utils*), 2479  
 file\_mime\_type() (in module *ironic.common.utils*), 2481  
 FileImageService (class in *ironic.common.image\_service*), 2333  
 FileSystemNotSupported, 2306  
 filter\_target\_raid\_config() (in module *ironic.common.raid*), 2434  
 finalize\_rescue() (in module *ironic.drivers.modules.agent\_client*), 3653  
 find\_step() (in module *ironic.conductor.steps*), 2678  
 find\_step() (in module *ironic.drivers.modules.agent\_base*), 3636  
 FirmwareImageLocation (class in *ironic.drivers.modules.ilo.firmware\_processor*), 3291  
 FirmwareProcessor (class in *ironic.drivers.modules.ilo.firmware\_processor*), 3292  
 first\_method() (in module *ironic.drivers.modules.fake.FakeVendorA*), 3837  
 flash\_firmware\_sum() (in module *ironic.drivers.modules.ilo.management.ilo\_management*), 3307  
 FlatNetwork (class in *ironic.drivers.modules.network.flat*), 3435  
 FlexibleDict (class in *ironic.objects.fields*), 4304  
 FlexibleDictField (class in *ironic.objects.fields*), 4306  
 FLOPPY (in module *ironic.common.boot\_devices*), 2260  
 Forbidden, 2306  
 force\_iscsi\_lun\_update() (in module *ironic.drivers.modules.iscsi\_deploy*), 3940  
 force\_persistent\_boot() (in module *ironic.drivers.utils*), 4214  
 force\_raw\_will\_convert() (in module *ironic.common.images*), 2359  
 format\_exception() (in module *ironic.common.utils*), 2479  
 fourth\_method\_shared\_lock() (in module *ironic.drivers.modules.fake.FakeVendorB*), 3840  
 from\_chassis() (in module *ironic.api.controllers.v1.node.NodesController*), 2036  
 from\_power() (in module *ironic.objects.node.NodeCorrectedPowerState*), 4349  
 FSM (class in *ironic.common.fsm*), 2324  
 FunctionArgument (class in *ironic.api.functions*), 2208  
 FunctionDefinition (class in *ironic.api.functions*), 2209  
 GenericClient (class in *ironic.drivers.modules.agent\_client*), 3653  
 GenericHardware (class in *ironic.drivers.generic*), 4186  
 get() (in module *ironic.api.controllers.v1.node.BootDeviceController*), 1996  
 get() (in module *ironic.api.controllers.v1.node.NodeConsoleController*), 2006  
 get() (in module *ironic.api.controllers.v1.node.NodeStatesController*), 2010  
 get() (in module *ironic.api.controllers.v1.volume.VolumeController*), 2168  
 get() (in module *ironic.api.functions.FunctionDefinition*), 2210  
 get() (in module *ironic.drivers.modules.snmp.SNMPClient*), 4001  
 get() (in module *ironic.objects.allocation.Allocation*), 4224  
 get() (in module *ironic.objects.bios.BIOSSetting*), 4252  
 get() (in module *ironic.objects.chassis.Chassis*), 4264  
 get() (in module *ironic.objects.node.Node*), 4327  
 get() (in module *ironic.objects.port.Port*), 4370  
 get() (in module *ironic.objects.portgroup.Portgroup*), 4390  
 get() (in module *ironic.objects.volume\_connector.VolumeConnector*), 4416  
 get() (in module *ironic.objects.volume\_target.VolumeTarget*), 4433  
 get\_active\_hardware\_type\_dict() (in module *ironic.db.api.Connection*), 2934  
 get\_active\_hardware\_type\_dict() (in module *ironic.db.sqlalchemy.api.Connection*), 2785  
 get\_adapter() (in module *ironic.common.utils*), 2479

*ironic.common.keystone*), 2364  
get\_admin\_context() (in module *ironic.common.context*), 2279  
get\_all() (*ironic.api.controllers.v1.allocation.AllocationsController* method), 1958  
get\_all() (*ironic.api.controllers.v1.allocation.NodeAllocationController* method), 1964  
get\_all() (*ironic.api.controllers.v1.bios.NodeBiosController* method), 1965  
get\_all() (*ironic.api.controllers.v1.chassis.ChassisController* method), 1967  
get\_all() (*ironic.api.controllers.v1.conductor.ConductorsController* method), 1975  
get\_all() (*ironic.api.controllers.v1.deploy\_template.DeployTemplatesController* method), 1979  
get\_all() (*ironic.api.controllers.v1.driver.DriversController* method), 1990  
get\_all() (*ironic.api.controllers.v1.node.IndicatorController* method), 2000  
get\_all() (*ironic.api.controllers.v1.node.NodesController* method), 2037  
get\_all() (*ironic.api.controllers.v1.node.NodeTraitsController* method), 2025  
get\_all() (*ironic.api.controllers.v1.node.NodeVIFController* method), 2027  
get\_all() (*ironic.api.controllers.v1.port.PortsController* method), 2061  
get\_all() (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2070  
get\_all() (*ironic.api.controllers.v1.ramdisk.LookupController* method), 2079  
get\_all() (*ironic.api.controllers.v1.volume\_connector.VolumeConnectorsController* method), 2171  
get\_all() (*ironic.api.controllers.v1.volume\_target.VolumeTargetsController* method), 2185  
get\_allocation\_by\_id() (*ironic.db.api.Connection* method), 2935  
get\_allocation\_by\_id() (*ironic.db.sqlalchemy.api.Connection* method), 2786  
get\_allocation\_by\_name() (*ironic.db.api.Connection* method), 2936  
get\_allocation\_by\_name() (*ironic.db.sqlalchemy.api.Connection* method), 2787  
get\_allocation\_by\_uuid() (*ironic.db.api.Connection* method), 2937  
get\_allocation\_by\_uuid() (*ironic.db.sqlalchemy.api.Connection* method), 2788  
get\_allocation\_list() (*ironic.db.sqlalchemy.api.Connection* method), 2937  
get\_allowed\_exmods() (in module *ironic.common.rpc*), 2447  
get\_and\_validate\_firmware\_image\_info() (*ironic.drivers.modules.ilo.firmware\_processor*), 2447  
get\_arg() (*ironic.api.functions.FunctionDefinition* method), 2210  
get\_async\_step\_return\_state() (in module *ironic.drivers.modules.deploy\_utils*), 3719  
get\_attached\_vif() (in module *ironic.conductor.utils*), 2713  
get\_auth() (in module *ironic.common.keystone*), 2365  
get\_backend() (in module *ironic.db.migration*), 3031  
get\_backend() (in module *ironic.db.sqlalchemy.api*), 2886  
get\_bios\_config() (*ironic.drivers.modules.drac.vendor\_passthru.DracWSMan* method), 3158  
get\_bios\_setting() (*ironic.db.api.Connection* method), 2791  
get\_bios\_setting\_list() (*ironic.db.api.Connection* method), 2941  
get\_bios\_setting\_list() (*ironic.db.sqlalchemy.api.Connection* method), 2792  
get\_boot\_device() (*ironic.conductor.manager.ConductorManager* method), 2533  
get\_boot\_device() (*ironic.conductor.rpcapi.ConductorAPI* method), 2601  
get\_boot\_device() (*ironic.drivers.base.ManagementInterface*

<code>method</code> ), 4080	<code>get_by_hostname()</code>
<code>get_boot_device()</code>	<code>(ironic.objects.conductor.Conductor</code>
<code>(ironic.drivers.modules.drac.management.DracWSManManagement</code>	<code>method)</code> , 4273
<code>method</code> ), 3105	<code>get_by_id()</code> <code>(ironic.objects.allocation.Allocation</code>
<code>get_boot_device()</code>	<code>class method)</code> , 4225
<code>(ironic.drivers.modules.fake.FakeManagement</code>	<code>get_by_id()</code> <code>(ironic.objects.chassis.Chassis</code>
<code>method</code> ), 3790	<code>class method)</code> , 4265
<code>get_boot_device()</code>	<code>get_by_id()</code> <code>(ironic.objects.deploy_template.DeployTemplate</code>
<code>(ironic.drivers.modules.ibmc.management.IBMCManagement</code>	<code>class method)</code> , 4284
<code>method</code> ), 3164	<code>get_by_id()</code> <code>(ironic.objects.node.Node class</code>
<code>get_boot_device()</code>	<code>method)</code> , 4328
<code>(ironic.drivers.modules.ilo.management.IloManagement</code>	<code>get_by_id()</code> <code>(ironic.objects.port.Port class</code>
<code>method</code> ), 3308	<code>method)</code> , 4372
<code>get_boot_device()</code>	<code>get_by_id()</code> <code>(ironic.objects.portgroup.Portgroup</code>
<code>(ironic.drivers.modules.ipmitool.IPMIManagement</code>	<code>class method)</code> , 4392
<code>method</code> ), 3876	<code>get_by_id()</code> <code>(ironic.objects.volume_connector.VolumeConnector</code>
<code>get_boot_device()</code>	<code>class method)</code> , 4418
<code>(ironic.drivers.modules.noop_mgmt.NoopManagement</code>	<code>get_by_id()</code> <code>(ironic.objects.volume_target.VolumeTarget</code>
<code>method</code> ), 3971	<code>class method)</code> , 4434
<code>get_boot_device()</code>	<code>get_by_instance_uuid()</code>
<code>(ironic.drivers.modules.redfish.management.RedfishManagement</code>	<code>(ironic.objects.node.Node class method)</code> ,
<code>method</code> ), 3496	4329
<code>get_boot_device()</code>	<code>get_by_name()</code>
<code>(ironic.drivers.modules.xclarity.management.XClarityManagement</code>	<code>(ironic.objects.allocation.Allocation</code>
<code>method</code> ), 3569	<code>class method)</code> , 4226
<code>get_boot_mode()</code> (in module	<code>get_by_name()</code>
<code>ironic.drivers.modules.boot_mode_utils</code> ),	<code>(ironic.objects.deploy_template.DeployTemplate</code>
3693	<code>class method)</code> , 4285
<code>get_boot_mode()</code>	<code>get_by_name()</code> <code>(ironic.objects.node.Node</code>
<code>(ironic.drivers.base.ManagementInterface</code>	<code>class method)</code> , 4330
<code>method</code> ), 4082	<code>get_by_name()</code> <code>(ironic.objects.port.Port class</code>
<code>get_boot_mode()</code>	<code>method)</code> , 4373
<code>(ironic.drivers.modules.ibmc.management.IBMCManagement</code>	<code>(ironic.objects.portgroup.Portgroup</code>
<code>method</code> ), 3167	<code>class method)</code> , 4393
<code>get_boot_mode()</code>	<code>get_by_node_id()</code>
<code>(ironic.drivers.modules.ilo.management.IloManagement</code>	<code>(ironic.objects.bios.BIOSSettingList</code>
<code>method</code> ), 3310	<code>class method)</code> , 4258
<code>get_boot_mode()</code>	<code>get_by_node_id()</code>
<code>(ironic.drivers.modules.redfish.management.RedfishManagement</code>	<code>(ironic.objects.trait.TraitList class</code>
<code>method</code> ), 3499	<code>method)</code> , 4411
<code>get_boot_mode_for_deploy()</code> (in module	<code>get_by_node_uuid()</code>
<code>ironic.drivers.modules.boot_mode_utils</code> ),	<code>(ironic.objects.deployment.Deployment</code>
3694	<code>class method)</code> , 4299
<code>get_boot_option()</code> (in module	<code>get_by_port_addresses()</code>
<code>ironic.drivers.modules.deploy_utils</code> ),	<code>(ironic.objects.node.Node class method)</code> ,
3720	4331
<code>get_by_address()</code> <code>(ironic.objects.port.Port</code>	<code>get_by_uuid()</code>
<code>class method)</code> , 4371	<code>(ironic.objects.allocation.Allocation</code>
<code>get_by_address()</code>	<code>class method)</code> , 4227
<code>(ironic.objects.portgroup.Portgroup</code>	<code>get_by_uuid()</code>
<code>class method)</code> , 4391	

<i>(ironic.objects.chassis.Chassis class method), 4266</i>	<i>(ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3055</i>
<i>get_by_uuid()</i> <i>(ironic.objects.deploy_template.DeployTemplate class method), 4287</i>	<i>get_cleaning_network_uuid()</i> <i>(ironic.common.neutron.NeutronNetworkInterfaceMixin method), 2376</i>
<i>get_by_uuid()</i> <i>(ironic.objects.deployment.Deployment class method), 4300</i>	<i>get_client()</i> (in module <i>ironic.common.cinder</i> ), 2274
<i>get_by_uuid()</i> ( <i>ironic.objects.node.Node class method</i> ), 4332	<i>get_client()</i> (in module <i>ironic.common.neutron</i> ), 2380
<i>get_by_uuid()</i> ( <i>ironic.objects.port.Port class method</i> ), 4374	<i>get_client()</i> (in module <i>ironic.common.rpc</i> ), 2447
<i>get_by_uuid()</i> <i>(ironic.objects.portgroup.Portgroup class method), 4394</i>	<i>get_command_error()</i> (in module <i>ironic.drivers.modules.agent_client</i> ), 3684
<i>get_by_uuid()</i> <i>(ironic.objects.volume_connector.VolumeConnector class method), 4419</i>	<i>get_commands_status()</i> <i>(ironic.drivers.modules.agent_client.AgentClient method), 3658</i>
<i>get_by_uuid()</i> <i>(ironic.objects.volume_target.VolumeTarget class method), 4436</i>	<i>get_conductor()</i> ( <i>ironic.db.api.Connection method</i> ), 2944
<i>get_chassis_by_id()</i> <i>(ironic.db.api.Connection method), 2941</i>	<i>get_conductor()</i> <i>(ironic.db.sqlalchemy.api.Connection method), 2795</i>
<i>get_chassis_by_id()</i> <i>(ironic.db.sqlalchemy.api.Connection method), 2793</i>	<i>get_conductor_for()</i> <i>(ironic.conductor.rpcapi.ConductorAPI method), 2604</i>
<i>get_chassis_by_uuid()</i> <i>(ironic.db.api.Connection method), 2942</i>	<i>get_conductor_list()</i> <i>(ironic.db.api.Connection method), 2945</i>
<i>get_chassis_by_uuid()</i> <i>(ironic.db.sqlalchemy.api.Connection method), 2793</i>	<i>get_conductor_list()</i> <i>(ironic.db.sqlalchemy.api.Connection method), 2796</i>
<i>get_chassis_list()</i> <i>(ironic.db.api.Connection method), 2942</i>	<i>get_config()</i> (in module <i>ironic.drivers.modules.drac.bios</i> ), 3072
<i>get_chassis_list()</i> <i>(ironic.db.sqlalchemy.api.Connection method), 2793</i>	<i>get_console()</i> <i>(ironic.drivers.base.ConsoleInterface method), 4059</i>
<i>get_class()</i> (in module <i>ironic.db.sqlalchemy.models</i> ), 2905	<i>get_console()</i> <i>(ironic.drivers.modules.fake.FakeConsole method), 3770</i>
<i>get_clean_steps()</i> <i>(ironic.drivers.base.BaseInterface method), 4044</i>	<i>get_console()</i> <i>(ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole method), 3896</i>
<i>get_clean_steps()</i> <i>(ironic.drivers.modules.agent_base.AgentDeployMixin method), 3619</i>	<i>get_console()</i> <i>(ironic.drivers.modules.ipmitool.IPMISocatConsole method), 3899</i>
<i>get_clean_steps()</i> <i>(ironic.drivers.modules.agent_client.AgentClient method), 3655</i>	<i>get_console()</i> <i>(ironic.drivers.modules.noop.NoConsole method), 3954</i>
<i>get_clean_steps()</i>	<i>get_console_information()</i> <i>(ironic.conductor.manager.ConductorManager method), 2533</i>



[get\\_console\\_information\(\)](#)  
 ([ironic.conductor.rpcapi.ConductorAPI](#)  
[method](#)), 2605

[get\\_controller\\_reserved\\_names\(\)](#) (in  
[module \[ironic.api.controllers.v1.utils\]\(#\)](#)),  
 2126

[get\\_current\\_boot\\_mode\(\)](#) (in [module](#)  
[ironic.drivers.modules.ilo.common](#)),  
 3260

[get\\_current\\_topic\(\)](#)  
 ([ironic.conductor.rpcapi.ConductorAPI](#)  
[method](#)), 2607

[get\\_current\\_vif\(\)](#)  
 ([ironic.drivers.base.NetworkInterface](#)  
[method](#)), 4119

[get\\_current\\_vif\(\)](#)  
 ([ironic.drivers.modules.network.common.VIFPortIDMonitor](#)  
[method](#)), 3427

[get\\_current\\_vif\(\)](#)  
 ([ironic.drivers.modules.network.noop.NoopNetwork](#)  
[method](#)), 3457

[get\\_default\\_boot\\_option\(\)](#) (in [module](#)  
[ironic.drivers.modules.deploy\\_utils](#)),  
 3721

[get\\_deploy\\_info\(\)](#) (in [module](#)  
[ironic.drivers.modules.iscsi\\_deploy](#)),  
 3940

[get\\_deploy\\_steps\(\)](#)  
 ([ironic.drivers.base.BaseInterface](#)  
[method](#)), 4046

[get\\_deploy\\_steps\(\)](#)  
 ([ironic.drivers.modules.agent.AgentRAID](#)  
[method](#)), 3594

[get\\_deploy\\_steps\(\)](#)  
 ([ironic.drivers.modules.agent\\_base.AgentDeployMixin](#)  
[method](#)), 3620

[get\\_deploy\\_steps\(\)](#)  
 ([ironic.drivers.modules.agent\\_client.AgentClient](#)  
[method](#)), 3664

[get\\_deploy\\_template\\_by\\_id\(\)](#)  
 ([ironic.db.api.Connection](#) [method](#)),  
 2947

[get\\_deploy\\_template\\_by\\_id\(\)](#)  
 ([ironic.db.sqlalchemy.api.Connection](#)  
[method](#)), 2798

[get\\_deploy\\_template\\_by\\_name\(\)](#)  
 ([ironic.db.api.Connection](#) [method](#)), 2948

[get\\_deploy\\_template\\_by\\_name\(\)](#)  
 ([ironic.db.sqlalchemy.api.Connection](#)  
[method](#)), 2799

[get\\_deploy\\_template\\_by\\_uuid\(\)](#)  
 ([ironic.db.api.Connection](#) [method](#)), 2949

[get\\_deploy\\_template\\_by\\_uuid\(\)](#)  
 ([ironic.db.sqlalchemy.api.Connection](#)  
[method](#)), 2800

[get\\_deploy\\_template\\_list\(\)](#)  
 ([ironic.db.api.Connection](#) [method](#)),  
 2949

[get\\_deploy\\_template\\_list\(\)](#)  
 ([ironic.db.sqlalchemy.api.Connection](#)  
[method](#)), 2800

[get\\_deploy\\_template\\_list\\_by\\_names\(\)](#)  
 ([ironic.db.api.Connection](#) [method](#)), 2951

[get\\_deploy\\_template\\_list\\_by\\_names\(\)](#)  
 ([ironic.db.sqlalchemy.api.Connection](#)  
[method](#)), 2802

[get\\_disk\\_label\(\)](#) (in [module](#)  
[ironic.drivers.modules.deploy\\_utils](#)),  
 3721

[get\\_drac\\_client\(\)](#) (in [module](#)  
[ironic.drivers.modules.drac.common](#)),  
 3090

[get\\_driver\(\)](#) ([ironic.common.driver\\_factory.BaseDriverFactory](#)  
[method](#)), 2284

[get\\_driver\\_properties\(\)](#)  
 ([ironic.conductor.manager.ConductorManager](#)  
[method](#)), 2533

[get\\_driver\\_properties\(\)](#)  
 ([ironic.conductor.rpcapi.ConductorAPI](#)  
[method](#)), 2607

[get\\_driver\\_vendor\\_passthru\\_methods\(\)](#)  
 ([ironic.conductor.manager.ConductorManager](#)  
[method](#)), 2533

[get\\_driver\\_vendor\\_passthru\\_methods\(\)](#)  
 ([ironic.conductor.rpcapi.ConductorAPI](#)  
[method](#)), 2608

[get\\_endpoint\(\)](#) (in [module](#)  
[ironic.common.keystone](#)), 2366

[get\\_enforcer\(\)](#) (in [module](#)  
[ironic.common.policy](#)), 2405

[get\\_free\\_port\\_like\\_object\(\)](#)  
 (in [module](#)  
[ironic.drivers.modules.network.common](#)),  
 3429

[get\\_hardware\\_type\(\)](#) (in [module](#)  
[ironic.common.driver\\_factory](#)), 2295

[get\\_ilo\\_object\(\)](#) (in [module](#)  
[ironic.drivers.modules.ilo.common](#)),  
 3261

[get\\_image\\_download\\_source\(\)](#) (in [mod-](#)  
[ule \[ironic.drivers.modules.deploy\\\_utils\]\(#\)](#)),  
 3722

[get\\_image\\_info\(\)](#) (in module [ironic.common.pxe\\_utils](#)), 2420  
[get\\_image\\_instance\\_info\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), 3723  
[get\\_image\\_properties\(\)](#) (in module [ironic.common.images](#)), 2359  
[get\\_image\\_service\(\)](#) (in module [ironic.common.image\\_service](#)), 2341  
[get\\_indicator\\_state\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 2533  
[get\\_indicator\\_state\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), 2610  
[get\\_indicator\\_state\(\)](#) ([ironic.drivers.base.ManagementInterface](#) method), 4085  
[get\\_indicator\\_state\(\)](#) ([ironic.drivers.modules.fake.FakeManagementInterface](#) method), 3792  
[get\\_indicator\\_state\(\)](#) ([ironic.drivers.modules.redfish.management.RedfishManagementInterface](#) method), 3500  
[get\\_inspection\\_network\\_uuid\(\)](#) ([ironic.common.neutron.NeutronNetworkInterfaceMixin](#) method), 2376  
[get\\_instance\(\)](#) (in module [ironic.db.api](#)), 3030  
[get\\_instance\\_image\\_info\(\)](#) (in module [ironic.common.pxe\\_utils](#)), 2423  
[get\\_interface\(\)](#) (in module [ironic.common.driver\\_factory](#)), 2296  
[get\\_ip\\_addresses\(\)](#) ([ironic.dhcp.base.BaseDHCP](#) method), 3033  
[get\\_ip\\_addresses\(\)](#) ([ironic.dhcp.neutron.NeutronDHCPApi](#) method), 3039  
[get\\_ip\\_addresses\(\)](#) ([ironic.dhcp.none.NoneDHCPApi](#) method), 3046  
[get\\_ipxe\\_boot\\_file\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), 3724  
[get\\_ipxe\\_config\\_template\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), 3726  
[get\\_ipxe\\_root\\_dir\(\)](#) (in module [ironic.common.pxe\\_utils](#)), 2425  
[get\\_irmc\\_client\(\)](#) (in module [ironic.drivers.modules.irmc.common](#)), 3377  
[get\\_irmc\\_report\(\)](#) (in module [ironic.drivers.modules.irmc.common](#)), 3378  
[get\\_ironic\\_api\\_url\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), 3726  
[get\\_job\(\)](#) (in module [ironic.drivers.modules.drac.job](#)), 3098  
[get\\_kernel\\_ramdisk\\_info\(\)](#) (in module [ironic.common.pxe\\_utils](#)), 2425  
[get\\_last\\_command\\_status\(\)](#) ([ironic.drivers.modules.agent\\_client.AgentClient](#) method), 3667  
[get\\_local\\_group\\_information\(\)](#) (in module [ironic.common.neutron](#)), 2381  
[get\\_logical\\_disk\\_properties\(\)](#) (in module [ironic.common.raid](#)), 2437  
[get\\_logical\\_disk\\_properties\(\)](#) ([ironic.drivers.base.RAIDInterface](#) method), 4149  
[get\\_logical\\_disks\(\)](#) ([ironic.drivers.modules.drac.raid.DracWSManRAID](#) method), 3133  
[get\\_neutron\\_port\\_data\(\)](#) (in module [ironic.common.neutron](#)), 2383  
[get\\_next\(\)](#) (in module [ironic.api.controllers.v1.collection](#)), 1972  
[get\\_next\(\)](#) ([ironic.drivers.modules.snmp.SNMPClient](#) method), 4002  
[get\\_node\\_by\\_id\(\)](#) ([ironic.db.api.Connection](#) method), 2952  
[get\\_node\\_by\\_id\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), 2803  
[get\\_node\\_by\\_instance\(\)](#) ([ironic.db.api.Connection](#) method), 2952  
[get\\_node\\_by\\_instance\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), 2803  
[get\\_node\\_by\\_name\(\)](#) ([ironic.db.api.Connection](#) method), 2953  
[get\\_node\\_by\\_name\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), 2804  
[get\\_node\\_by\\_port\\_addresses\(\)](#) ([ironic.db.api.Connection](#) method),

2953

`get_node_by_port_addresses()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2804

`get_node_by_uuid()`  
(*ironic.db.api.Connection* method), 2954

`get_node_by_uuid()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2805

`get_node_capability()` (in module *ironic.drivers.utils*), 4215

`get_node_list()` (*ironic.db.api.Connection* method), 2954

`get_node_list()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2805

`get_node_mac_addresses()` (in module *ironic.drivers.utils*), 4216

`get_node_network_data()`  
(*ironic.drivers.base.NetworkInterface* method), 4120

`get_node_network_data()`  
(*ironic.drivers.modules.network.common.NetworkInterface* method), 3418

`get_node_next_clean_steps()` (in module *ironic.conductor.utils*), 2714

`get_node_next_deploy_steps()` (in module *ironic.conductor.utils*), 2714

`get_node_portmap()` (in module *ironic.common.neutron*), 2385

`get_node_tags_by_node_id()`  
(*ironic.db.api.Connection* method), 2957

`get_node_tags_by_node_id()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2808

`get_node_traits_by_node_id()`  
(*ironic.db.api.Connection* method), 2958

`get_node_traits_by_node_id()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2809

`get_node_vendor_passthru_methods()`  
(*ironic.conductor.manager.ConductorManager* method), 2533

`get_node_vendor_passthru_methods()`  
(*ironic.conductor.rpcapi.ConductorAPI* method), 2613

`get_node_vif_ids()` (in module *ironic.common.network*), 2370

`get_node_with_token()`  
(*ironic.conductor.manager.ConductorManager* method), 2533

`get_node_with_token()`  
(*ironic.conductor.rpcapi.ConductorAPI* method), 2614

`get_nodeinfo_list()`  
(*ironic.db.api.Connection* method), 2958

`get_nodeinfo_list()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2809

`get_nodes_controller_reserved_names()`  
(in module *ironic.api.controllers.v1.node*), 2048

`get_not_versions()`  
(*ironic.db.api.Connection* method), 2962

`get_not_versions()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2813

`get_object_versions()` (in module *ironic.common.release\_mappings*), 2442

`get_one_vif_port_id_in_nodeinfo()`  
(*ironic.db.api.Connection* method), 2963

`get_offline_conductors()`  
(*ironic.db.sqlalchemy.api.Connection* method), 2815

`get_one()` (*ironic.api.controllers.v1.allocation.AllocationsController* method), 1961

`get_one()` (*ironic.api.controllers.v1.bios.NodeBiosController* method), 1965

`get_one()` (*ironic.api.controllers.v1.chassis.ChassisController* method), 1969

`get_one()` (*ironic.api.controllers.v1.conductor.ConductorsController* method), 1977

`get_one()` (*ironic.api.controllers.v1.deploy\_template.DeployTemplateController* method), 1982

`get_one()` (*ironic.api.controllers.v1.driver.DriversController* method), 1990

`get_one()` (*ironic.api.controllers.v1.node.IndicatorController* method), 2001

`get_one()` (*ironic.api.controllers.v1.node.NodesController* method), 2044

`get_one()` (*ironic.api.controllers.v1.port.PortsController* method), 2064

`get_one()` (*ironic.api.controllers.v1.portgroup.PortgroupsController* method), 2073

`get_one()` (*ironic.api.controllers.v1.volume\_connector.VolumeConnectorController* method), 2174



[get\\_one\(\) \(ironic.api.controllers.v1.volume\\_target\\_volume\\_target\\_controller method\), 2189](#)  
[get\\_online\\_conductors\(\) \(ironic.db.api.Connection method\), 2964](#)  
[get\\_online\\_conductors\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2816](#)  
[get\\_oslo\\_policy\\_enforcer\(\) \(in module ironic.common.policy\), 2406](#)  
[get\\_partition\\_uuids\(\) \(ironic.drivers.modules.agent\\_client.AgentClient method\), 3667](#)  
[get\\_patch\\_values\(\) \(in module ironic.api.controllers.v1.utils\), 2127](#)  
[get\\_path\\_relative\\_to\\_tftp\\_root\(\) \(in module ironic.common.pxe\\_utils\), 2427](#)  
[get\\_pecan\\_config\(\) \(in module ironic.api.app\), 2208](#)  
[get\\_physnets\\_by\\_port\\_uuid\(\) \(in module ironic.common.neutron\), 2387](#)  
[get\\_physnets\\_by\\_portgroup\\_id\(\) \(in module ironic.common.network\), 2371](#)  
[get\\_physnets\\_for\\_node\(\) \(in module ironic.common.network\), 2373](#)  
[get\\_port\\_by\\_address\(\) \(ironic.db.api.Connection method\), 2965](#)  
[get\\_port\\_by\\_address\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2816](#)  
[get\\_port\\_by\\_id\(\) \(ironic.db.api.Connection method\), 2965](#)  
[get\\_port\\_by\\_id\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2817](#)  
[get\\_port\\_by\\_name\(\) \(ironic.db.api.Connection method\), 2966](#)  
[get\\_port\\_by\\_name\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2817](#)  
[get\\_port\\_by\\_uuid\(\) \(ironic.db.api.Connection method\), 2966](#)  
[get\\_port\\_by\\_uuid\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2818](#)  
[get\\_port\\_list\(\) \(ironic.db.api.Connection method\), 2967](#)  
[get\\_portgroup\\_by\\_address\(\) \(ironic.db.api.Connection method\), 2968](#)  
[get\\_portgroup\\_by\\_address\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2819](#)  
[get\\_portgroup\\_by\\_id\(\) \(in module ironic.common.network\), 2374](#)  
[get\\_portgroup\\_by\\_id\(\) \(ironic.db.api.Connection method\), 2969](#)  
[get\\_portgroup\\_by\\_id\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2820](#)  
[get\\_portgroup\\_by\\_name\(\) \(ironic.db.api.Connection method\), 2969](#)  
[get\\_portgroup\\_by\\_name\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2821](#)  
[get\\_portgroup\\_by\\_uuid\(\) \(ironic.db.api.Connection method\), 2970](#)  
[get\\_portgroup\\_by\\_uuid\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2821](#)  
[get\\_portgroup\\_list\(\) \(ironic.db.api.Connection method\), 2971](#)  
[get\\_portgroup\\_list\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2822](#)  
[get\\_portgroups\\_by\\_node\\_id\(\) \(ironic.db.api.Connection method\), 2972](#)  
[get\\_portgroups\\_by\\_node\\_id\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2823](#)  
[get\\_ports\\_by\\_node\\_id\(\) \(ironic.db.api.Connection method\), 2974](#)  
[get\\_ports\\_by\\_node\\_id\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2825](#)  
[get\\_ports\\_by\\_portgroup\\_id\(\) \(in module ironic.common.network\), 2375](#)  
[get\\_ports\\_by\\_portgroup\\_id\(\) \(ironic.db.api.Connection method\), 2975](#)

2975

`get_ports_by_portgroup_id()`  
*(ironic.db.sqlalchemy.api.Connection method), 2827*

`get_power_state()`  
*(ironic.drivers.base.PowerInterface method), 4136*

`get_power_state()`  
*(ironic.drivers.modules.agent\_power.AgentPower method), 3685*

`get_power_state()`  
*(ironic.drivers.modules.drac.power.DracWSManPower method), 3118*

`get_power_state()`  
*(ironic.drivers.modules.fake.FakePower method), 3813*

`get_power_state()`  
*(ironic.drivers.modules.ibmcpower.IBMCPower method), 3180*

`get_power_state()`  
*(ironic.drivers.modules.ilo.power.IloPower method), 3331*

`get_power_state()`  
*(ironic.drivers.modules.ipmitool.IPMIPower method), 3887*

`get_power_state()`  
*(ironic.drivers.modules.irmc.power.IRMCPower method), 3403*

`get_power_state()`  
*(ironic.drivers.modules.redfish.power.RedfishPower method), 3531*

`get_power_state()`  
*(ironic.drivers.modules.snmp.SNMPPower method), 4018*

`get_power_state()`  
*(ironic.drivers.modules.xclarity.power.XClarityPower method), 3575*

`get_properties()` (in module *ironic.drivers.modules.xclarity.common*), 3564

`get_properties()`  
*(ironic.drivers.base.BareDriver method), 4035*

`get_properties()`  
*(ironic.drivers.base.BaseInterface method), 4048*

`get_properties()`  
*(ironic.drivers.base.NetworkInterface method), 4123*

`get_properties()`  
*(ironic.drivers.base.RAIDInterface method), 4151*

`get_properties()`  
*(ironic.drivers.hardware\_type.AbstractHardwareType method), 4191*

`get_properties()`  
*(ironic.drivers.modules.agent.AgentDeploy method), 3582*

`get_properties()`  
*(ironic.drivers.modules.agent.AgentRAID method), 3595*

`get_properties()`  
*(ironic.drivers.modules.agent.AgentRescue method), 3597*

`get_properties()`  
*(ironic.drivers.modules.agent\_power.AgentPower method), 3686*

`get_properties()`  
*(ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3055*

`get_properties()`  
*(ironic.drivers.modules.drac.bios.DracWSManBIOS method), 3068*

`get_properties()`  
*(ironic.drivers.modules.drac.inspect.DracWSManInspect method), 3095*

`get_properties()`  
*(ironic.drivers.modules.drac.management.DracWSManManagement method), 3107*

`get_properties()`  
*(ironic.drivers.modules.drac.power.DracWSManPower method), 3120*

`get_properties()`  
*(ironic.drivers.modules.drac.raid.DracWSManRAID method), 3134*

`get_properties()`  
*(ironic.drivers.modules.drac.vendor\_passthru.DracWSManVendorPassthru method), 3159*

`get_properties()`  
*(ironic.drivers.modules.fake.FakeBIOS method), 3759*

`get_properties()`  
*(ironic.drivers.modules.fake.FakeBoot method), 3764*

`get_properties()`  
*(ironic.drivers.modules.fake.FakeConsole method), 3771*

`get_properties()`  
*(ironic.drivers.modules.fake.FakeDeploy method), 3778*

`get_properties()`  
*(ironic.drivers.modules.fake.FakeInspect method), 3785*

method), 3786

get\_properties()  
(*ironic.drivers.modules.fake.FakeManagement*  
method), 3794

get\_properties()  
(*ironic.drivers.modules.fake.FakePower*  
method), 3814

get\_properties()  
(*ironic.drivers.modules.fake.FakeRAID*  
method), 3825

get\_properties()  
(*ironic.drivers.modules.fake.FakeRescue*  
method), 3826

get\_properties()  
(*ironic.drivers.modules.fake.FakeStorage*  
method), 3833

get\_properties()  
(*ironic.drivers.modules.fake.FakeVendorA*  
method), 3837

get\_properties()  
(*ironic.drivers.modules.fake.FakeVendorB*  
method), 3840

get\_properties()  
(*ironic.drivers.modules.ibmcm.management.IBMCManagement*  
method), 3169

get\_properties()  
(*ironic.drivers.modules.ibmcp.power.IBMCPower*  
method), 3182

get\_properties()  
(*ironic.drivers.modules.ibmcr.raid.IbmCRaid*  
method), 3197

get\_properties()  
(*ironic.drivers.modules.ibmcr.vendor.IBMVendor*  
method), 3202

get\_properties()  
(*ironic.drivers.modules.ilo.bios.IloBIOS*  
method), 3210

get\_properties()  
(*ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot*  
method), 3219

get\_properties()  
(*ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot*  
method), 3231

get\_properties()  
(*ironic.drivers.modules.ilo.console.IloConsoleInterface*  
method), 3289

get\_properties()  
(*ironic.drivers.modules.ilo.inspect.IloInspect*  
method), 3297

get\_properties()  
(*ironic.drivers.modules.ilo.management.IloManagement*  
method), 3311

get\_properties()  
(*ironic.drivers.modules.ilo.power.IloPower*  
method), 3332

get\_properties()  
(*ironic.drivers.modules.ilo.raid.Ilo5RAID*  
method), 3345

get\_properties()  
(*ironic.drivers.modules.ilo.vendor.VendorPassthru*  
method), 3348

get\_properties()  
(*ironic.drivers.modules.inspector.Inspector*  
method), 3869

get\_properties()  
(*ironic.drivers.modules.ipmitool.IPMIConsole*  
method), 3873

get\_properties()  
(*ironic.drivers.modules.ipmitool.IPMIManagement*  
method), 3878

get\_properties()  
(*ironic.drivers.modules.ipmitool.IPMIPower*  
method), 3888

get\_properties()  
(*ironic.drivers.modules.ipmitool.VendorPassthru*  
method), 3904

get\_properties()  
(*ironic.drivers.modules.irmc.bios.IRMCBIOS*  
method), 3355

get\_properties()  
(*ironic.drivers.modules.irmc.boot.IRMVirtualMediaBoot*  
method), 3365

get\_properties()  
(*ironic.drivers.modules.irmc.inspect.IRMInspect*  
method), 3384

get\_properties()  
(*ironic.drivers.modules.irmc.management.IRMManagement*  
method), 3389

get\_properties()  
(*ironic.drivers.modules.irmc.power.IRMCPower*  
method), 3405

get\_properties()  
(*ironic.drivers.modules.irmc.raid.IRMCRaid*  
method), 3416

get\_properties()  
(*ironic.drivers.modules.iscsi\_deploy.ISCSIDeploy*  
method), 3915

get\_properties()  
(*ironic.drivers.modules.noop.FailMixin*  
method), 3946

get\_properties()  
(*ironic.drivers.modules.noop\_mgmt.NoopManagement*  
method), 3946

method), 3973

get\_properties() (ironic.drivers.modules.pxe.PXERamdiskDeploy method), 3986

get\_properties() (ironic.drivers.modules.pxe\_base.PXEBaseMixin method), 3992

get\_properties() (ironic.drivers.modules.redfish.bios.RedfishBIOS method), 3469

get\_properties() (ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot method), 3479

get\_properties() (ironic.drivers.modules.redfish.inspect.RedfishInspect method), 3489

get\_properties() (ironic.drivers.modules.redfish.management.RedfishManagement method), 3502

get\_properties() (ironic.drivers.modules.redfish.power.RedfishPower method), 3533

get\_properties() (ironic.drivers.modules.redfish.vendor.RedfishVendorPassthru method), 3544

get\_properties() (ironic.drivers.modules.snmp.SNMPPower method), 4020

get\_properties() (ironic.drivers.modules.storage.cinder.CinderStorage method), 3550

get\_properties() (ironic.drivers.modules.storage.external.ExternalStorage method), 3554

get\_properties() (ironic.drivers.modules.storage.noop.NoopStorage method), 3560

get\_properties() (ironic.drivers.modules.xclarity.management.XClarityManagement method), 3570

get\_properties() (ironic.drivers.modules.xclarity.power.XClarityPower method), 3576

get\_properties() (ironic.drivers.utils.MixinVendorInterface method), 4207

get\_provisioning\_network\_uuid() (ironic.common.neutron.NeutronNetworkInterfaceMixin method), 2376

get\_pxe\_boot\_file() (in module ironic.drivers.modules.deploy\_utils), 3727

get\_pxe\_config\_file\_path() (in module ironic.common.pxe\_utils), 2427

get\_pxe\_config\_template() (in module ironic.drivers.modules.deploy\_utils), 3728

get\_raid\_controller\_list() (ironic.drivers.modules.ibm.ibm.vendor.IBMVendor method), 3202

get\_raid\_logical\_disk\_properties() (ironic.conductor.manager.ConductorManager method), 2533

get\_raid\_logical\_disk\_properties() (ironic.conductor.rpcapi.ConductorAPI method), 2615

get\_ramdisk\_logs\_file\_name() (in module ironic.drivers.utils), 4217

get\_remote\_boot\_volume() (in module ironic.drivers.modules.deploy\_utils), 3729

get\_return\_fields() (in module ironic.api.controllers.v1.utils), 2130

get\_rescuing\_network\_uuid() (ironic.common.neutron.NeutronNetworkInterfaceMixin method), 2376

get\_ring() (ironic.common.hash\_ring.HashRingManager method), 2330

get\_root\_device\_for\_deploy() (in module ironic.drivers.modules.deploy\_utils), 3729

get\_root\_dir() (in module ironic.common.pxe\_utils), 2429

get\_rpc\_allocation() (in module ironic.api.controllers.v1.utils), 2132

get\_rpc\_allocation\_with\_suffix() (in module ironic.api.controllers.v1.utils), 2134

get\_rpc\_deploy\_template() (in module ironic.api.controllers.v1.utils), 2135

get\_rpc\_deploy\_template\_with\_suffix() (in module ironic.api.controllers.v1.utils), 2137

get\_rpc\_node() (in module ironic.api.controllers.v1.utils), 2139

get\_rpc\_node\_with\_suffix() (in module ironic.api.controllers.v1.utils), 2140

get\_rpc\_portgroup() (in module

*ironic.api.controllers.v1.utils*), 2141

`get_rpc_portgroup_with_suffix()` (in module *ironic.api.controllers.v1.utils*), 2142

`get_secure_boot_mode()` (in module *ironic.drivers.modules.ilo.common*), 3262

`get_secure_boot_state()` (*ironic.drivers.base.ManagementInterface* method), 4087

`get_secure_boot_state()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3503

`get_sensors_data()` (*ironic.drivers.base.ManagementInterface* method), 4088

`get_sensors_data()` (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 3107

`get_sensors_data()` (*ironic.drivers.modules.fake.FakeManagement* method), 3795

`get_sensors_data()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3169

`get_sensors_data()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3312

`get_sensors_data()` (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3879

`get_sensors_data()` (*ironic.drivers.modules.irmc.management.IRMCMManagement* method), 3389

`get_sensors_data()` (*ironic.drivers.modules.noop\_mgmt.NoopManagement* method), 3973

`get_sensors_data()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3504

`get_sensors_data()` (*ironic.drivers.modules.xclarity.management.XClarityManagement* method), 3571

`get_sensors_notifier()` (in module *ironic.common.rpc*), 2447

`get_server()` (in module *ironic.common.rpc*), 2447

`get_server_hardware_id()` (in module *ironic.drivers.modules.xclarity.common*), 3564

`get_server_post_state()` (in module *ironic.drivers.modules.ilo.common*), 3264

`get_service_auth()` (in module *ironic.common.keystone*), 2367

`get_session()` (in module *ironic.common.keystone*), 2368

`get_shellinabox_console_url()` (in module *ironic.drivers.modules.console\_utils*), 3702

`get_single_nic_with_vif_port_id()` (*ironic.drivers.modules.deploy\_utils*), 3730

`get_socat_console_url()` (in module *ironic.drivers.modules.console\_utils*), 3702

`get_steps()` (in module *ironic.common.images*), 2360

`get_steps()` (in module *ironic.drivers.modules.agent\_base*), 3636

`get_supported_boot_devices()` (*ironic.agent.conductor.manager.ConductorManager* method), 2533

`get_supported_boot_devices()` (*ironic.conductor.rpcapi.ConductorAPI* method), 2619

`get_supported_boot_devices()` (*ironic.drivers.base.ManagementInterface* method), 4094

`get_supported_boot_devices()` (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 3108

`get_supported_boot_devices()` (*ironic.drivers.modules.fake.FakeManagement* method), 3801

`get_supported_boot_devices()` (*ironic.drivers.modules.ibmcm.management.IBMCMManagement* method), 3170

`get_supported_boot_devices()` (*ironic.drivers.modules.ilo.management.IloManagement* method), 3314

`get_supported_boot_devices()` (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3881

`get_supported_boot_devices()` (*ironic.drivers.modules.noop\_mgmt.NoopManagement* method), 3979

`get_supported_boot_devices()` (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3504



<code>method</code> ), 3506	<code>method</code> ), 3406
<code>get_supported_boot_devices()</code> ( <code>ironic.drivers.modules.xclarity.management.XClarityManagement</code> <code>method</code> ), 3571	<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.redfish.power.RedfishPower</code> <code>method</code> ), 3533
<code>get_supported_boot_modes()</code> ( <code>ironic.drivers.base.ManagementInterface</code> <code>method</code> ), 4095	<code>get_swift_session()</code> (in module <code>ironic.common.swift</code> ), 2476
<code>get_supported_boot_modes()</code> ( <code>ironic.drivers.modules.ibmcm.management.IBMCMManagement</code> <code>method</code> ), 3172	<code>get_swift_url()</code> (in module <code>ironic.drivers.modules.ilo.firmware_processor</code> ), 2325
<code>get_supported_boot_modes()</code> ( <code>ironic.drivers.modules.ilo.management.IloManagement</code> <code>method</code> ), 3314	<code>get_system()</code> (in module <code>ironic.drivers.modules.redfish.utils</code> ), 2539
<code>get_supported_boot_modes()</code> ( <code>ironic.drivers.modules.redfish.management.RedfishManagement</code> <code>method</code> ), 3507	<code>get_target_version()</code> ( <code>ironic.objects.base.IronicObject</code> class <code>method</code> ), 4242
<code>get_supported_indicators()</code> ( <code>ironic.conductor.manager.ConductorManager</code> <code>method</code> ), 2533	<code>get_temp_url()</code> ( <code>ironic.common.swift.SwiftAPI</code> <code>method</code> ), 2471
<code>get_supported_indicators()</code> ( <code>ironic.conductor.rpcapi.ConductorAPI</code> <code>method</code> ), 2621	<code>get_temp_url_for_glance_image()</code> (in module <code>ironic.common.images</code> ), 2360
<code>get_supported_indicators()</code> ( <code>ironic.drivers.base.ManagementInterface</code> <code>method</code> ), 4097	<code>get_tftp_path_prefix()</code> (in module <code>ironic.common.pxe_utils</code> ), 2429
<code>get_supported_indicators()</code> ( <code>ironic.drivers.modules.fake.FakeManagement</code> <code>method</code> ), 3801	<code>get_topic_for()</code> ( <code>ironic.conductor.rpcapi.ConductorAPI</code> <code>method</code> ), 2625
<code>get_supported_indicators()</code> ( <code>ironic.drivers.modules.redfish.management.RedfishManagement</code> <code>method</code> ), 3508	<code>get_topic_for_driver()</code> ( <code>ironic.conductor.rpcapi.ConductorAPI</code> <code>method</code> ), 2625
<code>get_supported_power_states()</code> ( <code>ironic.drivers.base.PowerInterface</code> <code>method</code> ), 4137	<code>get_trait_names()</code> ( <code>ironic.objects.trait.TraitList</code> <code>method</code> ), 4412
<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.agent_power.AgentPower</code> <code>method</code> ), 3687	<code>get_transport_url()</code> (in module <code>ironic.common.rpc</code> ), 2447
<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.fake.FakePower</code> <code>method</code> ), 3815	<code>get_update_service()</code> (in module <code>ironic.drivers.modules.redfish.utils</code> ), 3541
<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.ibmcm.power.IBMCPower</code> <code>method</code> ), 3183	<code>get_updated_capabilities()</code> (in module <code>ironic.common.utils</code> ), 2481
<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.ilo.power.IloPower</code> <code>method</code> ), 3333	<code>get_vendor_passthru_metadata()</code> (in module <code>ironic.conductor.manager</code> ), 2542
<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.ipmitool.IPMIPower</code> <code>method</code> ), 3889	<code>get_versioned_notifier()</code> (in module <code>ironic.common.rpc</code> ), 2447
<code>get_supported_power_states()</code> ( <code>ironic.drivers.modules.irmc.power.IRMCPower</code> <code>method</code> ), 3506	<code>get_volume_connector_by_id()</code> ( <code>ironic.db.api.Connection</code> <code>method</code> ), 2977
	<code>get_volume_connector_by_id()</code> ( <code>ironic.db.sqlalchemy.api.Connection</code> <code>method</code> ), 2829
	<code>get_volume_connector_by_uuid()</code> ( <code>ironic.db.api.Connection</code> <code>method</code> ), 2978
	<code>get_volume_connector_by_uuid()</code>

- (*ironic.db.sqlalchemy.api.Connection* method), 2830
- get\_volume\_connector\_list()* (*ironic.db.api.Connection* method), 2980
- get\_volume\_connector\_list()* (*ironic.db.sqlalchemy.api.Connection* method), 2831
- get\_volume\_connectors\_by\_node\_id()* (*ironic.db.api.Connection* method), 2982
- get\_volume\_connectors\_by\_node\_id()* (*ironic.db.sqlalchemy.api.Connection* method), 2833
- get\_volume\_pxe\_options()* (in module *ironic.common.pxe\_utils*), 2429
- get\_volume\_target\_by\_id()* (*ironic.db.api.Connection* method), 2984
- get\_volume\_target\_by\_id()* (*ironic.db.sqlalchemy.api.Connection* method), 2835
- get\_volume\_target\_by\_uuid()* (*ironic.db.api.Connection* method), 2985
- get\_volume\_target\_by\_uuid()* (*ironic.db.sqlalchemy.api.Connection* method), 2836
- get\_volume\_target\_list()* (*ironic.db.api.Connection* method), 2986
- get\_volume\_target\_list()* (*ironic.db.sqlalchemy.api.Connection* method), 2837
- get\_volume\_targets\_by\_node\_id()* (*ironic.db.api.Connection* method), 2988
- get\_volume\_targets\_by\_node\_id()* (*ironic.db.sqlalchemy.api.Connection* method), 2839
- get\_volume\_targets\_by\_volume\_id()* (*ironic.db.api.Connection* method), 2990
- get\_volume\_targets\_by\_volume\_id()* (*ironic.db.sqlalchemy.api.Connection* method), 2841
- get\_xclarity\_client()* (in module *ironic.drivers.modules.xclarity.common*), 3565
- getargspec()* (in module *ironic.api.functions*), 2211
- GlanceConnectionFailed*, 2306
- GlanceImageService* (class in *ironic.common.glance\_service.image\_service*), 2228
- ## H
- handle\_error\_notification()* (in module *ironic.api.controllers.v1.notification\_utils*), 2055
- handle\_ibmc\_exception()* (in module *ironic.drivers.modules.ibmc\_utils*), 3198
- handle\_patch\_port\_like\_extra\_vif()* (in module *ironic.api.controllers.v1.utils*), 2144
- handle\_post\_port\_like\_extra\_vif()* (in module *ironic.api.controllers.v1.utils*), 2146
- handle\_signal()* (*ironic.common.rpc\_service.RPCService* method), 2448
- handle\_sync\_power\_state\_max\_retries\_exceeded()* (in module *ironic.conductor.manager*), 2542
- hardware\_type* (*ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 2894
- hardware\_types()* (in module *ironic.common.driver\_factory*), 2298
- HardwareInspectionFailure*, 2306
- HardwareTypesFactory* (class in *ironic.common.driver\_factory*), 2284
- has\_decomposed\_deploy\_steps* (*ironic.drivers.modules.agent.AgentDeployMixin* attribute), 3588
- has\_decomposed\_deploy\_steps* (*ironic.drivers.modules.agent\_base.HeartbeatMixin* attribute), 3630
- has\_decomposed\_deploy\_steps* (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* attribute), 3056
- has\_decomposed\_deploy\_steps* (*ironic.drivers.modules.iscsi\_deploy.ISCSIDeploy* attribute), 3916
- has\_next()* (in module *ironic.api.controllers.v1.collection*), 1972
- hash\_password()* (in module *ironic.conductor.utils*), 2714
- HashRingManager* (class in *ironic.common.hash\_ring*), 2330
- head\_object()* (*ironic.common.swift.SwiftAPI* method), 2473

[heartbeat \(\) \(ironic.conductor.manager.ConductorManager method\), 2533](#)  
[heartbeat \(\) \(ironic.conductor.rpcapi.ConductorAPI method\), 2626](#)  
[heartbeat \(\) \(ironic.drivers.base.DeployInterface method\), 4065](#)  
[heartbeat \(\) \(ironic.drivers.modules.agent\\_base.HeartbeatMixin method\), 3631](#)  
[heartbeat\\_allowed\\_states \(\) \(ironic.drivers.modules.agent\\_base.HeartbeatMixin property\), 3632](#)  
[HeartbeatController \(class in ironic.api.controllers.v1.ramdisk\), 2076](#)  
[HeartbeatMixin \(class in ironic.drivers.modules.agent\\_base\), 3627](#)  
[hide\\_fields\\_in\\_newer\\_versions \(\) \(in ironic.api.controllers.v1.allocation\), 1964](#)  
[hide\\_fields\\_in\\_newer\\_versions \(\) \(in module ironic.api.controllers.v1.driver\), 1993](#)  
[hide\\_fields\\_in\\_newer\\_versions \(\) \(in module ironic.api.controllers.v1.node\), 2048](#)  
[hide\\_fields\\_in\\_newer\\_versions \(\) \(in module ironic.api.controllers.v1.port\), 2067](#)  
[host \(\) \(ironic.objects.notification.NotificationPublisher property\), 4367](#)  
[hostname \(ironic.db.sqlalchemy.models.Conductor attribute\), 2893](#)  
[hostname \(\) \(ironic.objects.conductor.Conductor property\), 4274](#)  
[HTTPForbidden, 2306](#)  
[HttpImageService \(class in ironic.common.image\\_service\), 2337](#)  
[HTTPNotFound, 2306](#)  
**I**  
[IBMConnectionError, 2307](#)  
[IBMError, 2307](#)  
[IBMCHardware \(class in ironic.drivers.ibmch\), 4195](#)  
[IBMManagement \(class in ironic.drivers.modules.ibmch.management\), 3164](#)  
[IBMCPower \(class in ironic.drivers.modules.ibmch.power\), 3180](#)  
[IbmCRAID \(class in ironic.drivers.modules.ibmch.raid\), 3190](#)  
[IBMCVendor \(class in ironic.drivers.modules.ibmch.vendor\), 3190](#)  
[id \(ironic.db.sqlalchemy.models.Allocation attribute\), 2890](#)  
[id \(ironic.db.sqlalchemy.models.Chassis attribute\), 2892](#)  
[id \(ironic.db.sqlalchemy.models.Conductor attribute\), 2893](#)  
[id \(ironic.db.sqlalchemy.models.ConductorHardwareInterfaces attribute\), 2894](#)  
[id \(ironic.db.sqlalchemy.models.DeployTemplate attribute\), 2894](#)  
[id \(ironic.db.sqlalchemy.models.DeployTemplateStep attribute\), 2895](#)  
[id \(ironic.db.sqlalchemy.models.Node attribute\), 2897](#)  
[id \(ironic.db.sqlalchemy.models.Port attribute\), 2901](#)  
[id \(ironic.db.sqlalchemy.models.Portgroup attribute\), 2902](#)  
[id \(ironic.db.sqlalchemy.models.VolumeConnector attribute\), 2903](#)  
[id \(ironic.db.sqlalchemy.models.VolumeTarget attribute\), 2904](#)  
[id \(\) \(ironic.objects.allocation.Allocation property\), 4228](#)  
[id \(\) \(ironic.objects.chassis.Chassis property\), 4267](#)  
[id \(\) \(ironic.objects.conductor.Conductor property\), 4274](#)  
[id \(\) \(ironic.objects.deploy\\_template.DeployTemplate property\), 4288](#)  
[id \(\) \(ironic.objects.node.Node property\), 4333](#)  
[id \(\) \(ironic.objects.port.Port property\), 4375](#)  
[id \(\) \(ironic.objects.portgroup.Portgroup property\), 4395](#)  
[id \(\) \(ironic.objects.volume\\_connector.VolumeConnector property\), 4420](#)  
[id \(\) \(ironic.objects.volume\\_target.VolumeTarget property\), 4437](#)  
[IDRACHardware \(class in ironic.drivers.drac\), 4180](#)  
[ignore\\_extra\\_args \(ironic.api.functions.FunctionDefinition attribute\), 2211](#)  
[Ilo5Hardware \(class in ironic.drivers.ilo\), 4180](#)



- 4196
- Ilo5Management (class in *ironic.drivers.modules.ilo.management*), 3301
- Ilo5RAID (class in *ironic.drivers.modules.ilo.raid*), 3338
- IloBIOS (class in *ironic.drivers.modules.ilo.bios*), 3206
- IloConsoleInterface (class in *ironic.drivers.modules.ilo.console*), 3289
- IloHardware (class in *ironic.drivers.ilo*), 4198
- IloInspect (class in *ironic.drivers.modules.ilo.inspect*), 3296
- IloiPXEBoot (class in *ironic.drivers.modules.ilo.boot*), 3240
- IloManagement (class in *ironic.drivers.modules.ilo.management*), 3303
- IloOperationError, 2307
- IloOperationNotSupported, 2307
- IloPower (class in *ironic.drivers.modules.ilo.power*), 3331
- IloPXEBoot (class in *ironic.drivers.modules.ilo.boot*), 3212
- IloUefiHttpsBoot (class in *ironic.drivers.modules.ilo.boot*), 3217
- IloVirtualMediaBoot (class in *ironic.drivers.modules.ilo.boot*), 3228
- image\_checksum() (*ironic.objects.deployment.Deployment* property), 4300
- image\_ref() (*ironic.objects.deployment.Deployment* property), 4301
- image\_show() (in module *ironic.common.images*), 2361
- image\_to\_raw() (in module *ironic.common.images*), 2361
- ImageCache (class in *ironic.drivers.modules.image\_cache*), 3843
- ImageConvertFailed, 2307
- ImageCreationFailed, 2308
- ImageDownloadFailed, 2308
- ImageHandler (class in *ironic.drivers.modules.image\_utils*), 3848
- ImageNotAuthorized, 2308
- ImageNotFound, 2308
- ImageRefValidationFailed, 2308
- ImageUnacceptable, 2308
- ImageUploadFailed, 2308
- in\_core\_deploy\_step() (*ironic.drivers.modules.agent\_base.HeartbeatMixin* method), 3632
- IncompatibleInterface, 2309
- IncompleteLookup, 2309
- index() (*ironic.api.controllers.root.RootController* method), 2201
- index() (*ironic.api.controllers.v1.Controller* method), 2198
- indicator\_convert\_with\_links() (in module *ironic.api.controllers.v1.node*), 2049
- indicator\_list\_from\_dict() (in module *ironic.api.controllers.v1.node*), 2049
- IndicatorAtComponent (class in *ironic.api.controllers.v1.node*), 1999
- IndicatorController (class in *ironic.api.controllers.v1.node*), 1999
- indicators (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 2009
- INFO (*ironic.objects.fields.NotificationLevel* attribute), 4309
- init() (in module *ironic.common.rpc*), 2447
- init\_enforcer() (in module *ironic.common.policy*), 2406
- init\_host() (*ironic.conductor.base\_manager.BaseConductorManager* method), 2511
- initial\_node\_provision\_state() (in module *ironic.api.controllers.v1.utils*), 2147
- initialize() (*ironic.common.fsm.FSM* method), 2327
- initialize\_wsgi\_app() (in module *ironic.api.wsgi*), 2224
- inject\_nmi (*ironic.api.controllers.v1.node.NodeManagementController* attribute), 2009
- inject\_nmi() (*ironic.conductor.manager.ConductorManager* method), 2533
- inject\_nmi() (*ironic.conductor.rpcapi.ConductorAPI* method), 2628
- inject\_nmi() (*ironic.drivers.base.ManagementInterface* method), 4104
- inject\_nmi() (*ironic.drivers.modules.ibmc.management.IBMC* method), 3172
- inject\_nmi() (*ironic.drivers.modules.ilo.management.IloManagement* method), 3315
- inject\_nmi() (*ironic.drivers.modules.ipmitool.IPMIManagement* method), 3881
- inject\_nmi() (*ironic.drivers.modules.irmc.management.IRMC* method), 3881

<code>method</code> ), 3396	<code>(ironic.objects.node.NodePayload prop-</code>
<code>inject_nmi()</code> ( <code>ironic.drivers.modules.redfish.management.RedfishManagement</code>	<code>inspect_interface()</code>
<code>method</code> ), 3515	
<code>InjectNmiController</code> ( <code>class</code> in <code>(ironic.objects.node.NodeSetPowerStatePayload</code>	
<code>ironic.api.controllers.v1.node</code> ), 2003	<code>property</code> ), 4356
<code>InputFileError</code> , 2309	<code>inspect_interface()</code>
<code>inspect</code> ( <code>ironic.drivers.base.BareDriver</code> at-	<code>(ironic.objects.node.NodeSetProvisionStatePayload</code>
<code>tribute</code> ), 4036	<code>property</code> ), 4360
<code>inspect_hardware()</code>	<code>INSPECTFAIL</code> ( <code>in</code> <code>module</code>
<code>(ironic.conductor.manager.ConductorManager</code>	<code>ironic.common.states</code> ), 2459
<code>method</code> ), 2533	<code>INSPECTING</code> ( <code>in</code> <code>module</code> <code>ironic.common.states</code> ),
<code>inspect_hardware()</code>	2459
<code>(ironic.conductor.rpcapi.ConductorAPI</code>	<code>InspectInterface</code> ( <code>class</code> in
<code>method</code> ), 2631	<code>ironic.drivers.base</code> ), 4074
<code>inspect_hardware()</code>	<code>inspection_finished_at</code>
<code>(ironic.drivers.base.InspectInterface</code>	<code>(ironic.db.sqlalchemy.models.Node</code>
<code>method</code> ), 4076	<code>attribute</code> ), 2897
<code>inspect_hardware()</code>	<code>inspection_finished_at()</code>
<code>(ironic.drivers.modules.drac.inspect.DracWSManInspect</code>	<code>(ironic.objects.node.Node</code> <code>property</code> ),
<code>method</code> ), 3095	4333
<code>inspect_hardware()</code>	<code>inspection_finished_at()</code>
<code>(ironic.drivers.modules.fake.FakeInspect</code>	<code>(ironic.objects.node.NodeCorrectedPowerStatePayload</code>
<code>method</code> ), 3786	<code>property</code> ), 4349
<code>inspect_hardware()</code>	<code>inspection_finished_at()</code>
<code>(ironic.drivers.modules.ilo.inspect.IloInspect</code>	<code>(ironic.objects.node.NodeCRUDPayload</code>
<code>method</code> ), 3297	<code>property</code> ), 4343
<code>inspect_hardware()</code>	<code>inspection_finished_at()</code>
<code>(ironic.drivers.modules.inspector.Inspector</code>	<code>(ironic.objects.node.NodePayload</code> <code>prop-</code>
<code>method</code> ), 3869	<code>erty</code> ), 4352
<code>inspect_hardware()</code>	<code>inspection_finished_at()</code>
<code>(ironic.drivers.modules.irmc.inspect.IRMCInspect</code>	<code>(ironic.objects.node.NodeSetPowerStatePayload</code>
<code>method</code> ), 3384	<code>property</code> ), 4356
<code>inspect_hardware()</code>	<code>inspection_finished_at()</code>
<code>(ironic.drivers.modules.noop.NoInspect</code>	<code>(ironic.objects.node.NodeSetProvisionStatePayload</code>
<code>method</code> ), 3957	<code>property</code> ), 4360
<code>inspect_hardware()</code>	<code>inspection_started_at</code>
<code>(ironic.drivers.modules.redfish.inspect.RedfishInspect</code>	<code>(ironic.db.sqlalchemy.models.Node</code>
<code>method</code> ), 3490	<code>attribute</code> ), 2897
<code>inspect_interface</code>	<code>inspection_started_at()</code>
<code>(ironic.db.sqlalchemy.models.Node</code>	<code>(ironic.objects.node.Node</code> <code>property</code> ),
<code>attribute</code> ), 2897	4333
<code>inspect_interface()</code>	<code>inspection_started_at()</code>
<code>(ironic.objects.node.Node</code> <code>property</code> ),	<code>(ironic.objects.node.NodeCorrectedPowerStatePayload</code>
4333	<code>property</code> ), 4349
<code>inspect_interface()</code>	<code>inspection_started_at()</code>
<code>(ironic.objects.node.NodeCorrectedPowerStatePayload</code>	<code>(ironic.objects.node.NodeCRUDPayload</code>
<code>property</code> ), 4349	<code>property</code> ), 4343
<code>inspect_interface()</code>	<code>inspection_started_at()</code>
<code>(ironic.objects.node.NodeCRUDPayload</code>	<code>(ironic.objects.node.NodePayload</code> <code>prop-</code>
<code>property</code> ), 4343	<code>erty</code> ), 4352
<code>inspect_interface()</code>	<code>inspection_started_at()</code>

(*ironic.objects.node.NodeSetPowerStatePayload* attribute), 4356  
*ironic.drivers.modules.deploy\_utils*), 3708  
inspection\_started\_at() (class in *ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4360  
Inspector (class in *ironic.drivers.modules.inspector*), 3868  
INSPECTWAIT (in module *ironic.common.states*), 2459  
install\_bootloader() (in module *ironic.drivers.modules.agent\_client.AgentClient*), 3669  
instance\_info (class in *ironic.db.sqlalchemy.models.Node* attribute), 2897  
instance\_info() (in module *ironic.objects.node.Node* attribute), 4333  
instance\_info() (in module *ironic.objects.node.NodeCRUDPayload* attribute), 4343  
instance\_info() (in module *ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4360  
instance\_info\_mapping (in module *ironic.objects.deployment.Deployment* attribute), 4301  
instance\_info\_mapping\_rev (in module *ironic.objects.deployment.Deployment* attribute), 4301  
instance\_uuid (in module *ironic.db.sqlalchemy.models.Node* attribute), 2898  
instance\_uuid() (in module *ironic.objects.node.Node* attribute), 4333  
instance\_uuid() (in module *ironic.objects.node.NodeCorrectedPowerStatePayload* attribute), 4349  
instance\_uuid() (in module *ironic.objects.node.NodeCRUDPayload* attribute), 4344  
instance\_uuid() (in module *ironic.objects.node.NodePayload* attribute), 4353  
instance\_uuid() (in module *ironic.objects.node.NodeSetPowerStatePayload* attribute), 4356  
instance\_uuid() (in module *ironic.objects.node.NodeSetProvisionStatePayload* attribute), 4360  
InstanceAssociated, 2309  
InstanceDeployFailure, 2309  
InstanceImageCache (class in *ironic.drivers.modules.deploy\_utils*), 3708  
InstanceNotFound, 2309  
InstanceRescueFailure, 2309  
InstanceUnrescueFailure, 2309  
InsufficientMemory, 2310  
InsufficientDiskSpace, 2310  
integer() (in module *ironic.common.args*), 2246  
IntegerField (class in *ironic.objects.fields*), 4306  
IntelIPMIHardware (class in *ironic.drivers.intel\_ipmi*), 4200  
IntelIPMIManagement (class in *ironic.drivers.modules.intel\_ipmi.management*), 3351  
interface (in module *ironic.db.sqlalchemy.models.DeployTemplateStep* attribute), 2895  
interface\_name (in module *ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 2894  
interface\_type (in module *ironic.db.sqlalchemy.models.ConductorHardwareInterface* attribute), 2894  
interface\_type (in module *ironic.drivers.base.BaseInterface* attribute), 4048  
interface\_type (in module *ironic.drivers.base.BIOSInterface* attribute), 4032  
interface\_type (in module *ironic.drivers.base.BootInterface* attribute), 4053  
interface\_type (in module *ironic.drivers.base.ConsoleInterface* attribute), 4060  
interface\_type (in module *ironic.drivers.base.DeployInterface* attribute), 4066  
interface\_type (in module *ironic.drivers.base.InspectInterface* attribute), 4078  
interface\_type (in module *ironic.drivers.base.ManagementInterface* attribute), 4105  
interface\_type (in module *ironic.drivers.base.NetworkInterface* attribute), 4123  
interface\_type (in module *ironic.drivers.base.PowerInterface* attribute), 4123

[attribute](#)), 4138  
[interface\\_type](#)  
     ([ironic.drivers.base.RAIDInterface](#)  
     [attribute](#)), 4151  
[interface\\_type](#)  
     ([ironic.drivers.base.RescueInterface](#)  
     [attribute](#)), 4156  
[interface\\_type](#)  
     ([ironic.drivers.base.StorageInterface](#)  
     [attribute](#)), 4161  
[interface\\_type](#)  
     ([ironic.drivers.base.VendorInterface](#)  
     [attribute](#)), 4165  
[InterfaceNotFoundInEntrypoint](#), 2310  
[interfaces\(\)](#) (in [module](#)  
     [ironic.common.driver\\_factory](#)), 2298  
[internal\\_info](#)  
     ([ironic.db.sqlalchemy.models.Port](#) [at-](#)  
     [tribute](#)), 2901  
[internal\\_info](#)  
     ([ironic.db.sqlalchemy.models.Portgroup](#)  
     [attribute](#)), 2902  
[internal\\_info\(\)](#) ([ironic.objects.port.Port](#)  
     [property](#)), 4375  
[internal\\_info\(\)](#)  
     ([ironic.objects.portgroup.Portgroup](#)  
     [property](#)), 4395  
[Invalid](#), 2310  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.allocation.AllocationController](#)  
     [attribute](#)), 1962  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.allocation.NodeAllocationController](#)  
     [attribute](#)), 1964  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.chassis.ChassisController](#)  
     [attribute](#)), 1970  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.conductor.ConductorsController](#)  
     [attribute](#)), 1978  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.deploy\\_template.DeployTemplateController](#)  
     [attribute](#)), 1983  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.node.NodesController](#)  
     [attribute](#)), 2045  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.port.PortsController](#)  
     [attribute](#)), 2065  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.portgroup.PortgroupsController](#)  
     [attribute](#)), 2074  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.volume\\_connector.VolumeConnector](#)  
     [attribute](#)), 2176  
[invalid\\_sort\\_key\\_list](#)  
     ([ironic.api.controllers.v1.volume\\_target.VolumeTargetsController](#)  
     [attribute](#)), 2191  
[InvalidConductorGroup](#), 2310  
[InvalidDatapathID](#), 2310  
[InvalidDeployTemplate](#), 2310  
[InvalidEndpoint](#), 2311  
[InvalidIdentity](#), 2311  
[InvalidImageRef](#), 2311  
[InvalidInput](#), 2311  
[InvalidIPAddress](#), 2311  
[InvalidIPv4Address](#), 2311  
[InvalidMAC](#), 2311  
[InvalidName](#), 2312  
[InvalidParameterValue](#), 2312  
[InvalidParams](#), 2239  
[InvalidRequest](#), 2239  
[InvalidState](#), 2312  
[InvalidStateRequested](#), 2312  
[InvalidSwitchID](#), 2312  
[InvalidUUID](#), 2312  
[InvalidUuidOrName](#), 2312  
[IPMIConsole](#) (class in [ironic.drivers.modules.ipmitool](#)), 3872  
[IPMIFailure](#), 2307  
[IPMIController](#) (class in [ironic.drivers.ipmi](#)), 4201  
[IPMIManagement](#) (class in [ironic.drivers.modules.ipmitool](#)), 3874  
[IPMIPower](#) (class in [ironic.drivers.modules.ipmitool](#)), 3886  
[IPMIShellinaboxConsole](#) (class in [ironic.drivers.modules.ipmitool](#)), 3896  
[IPMISocatConsole](#) (class in [ironic.drivers.modules.ipmitool](#)), 3899  
[ipxe\\_enabled](#) ([ironic.drivers.modules.ipxe.iPXEBoot](#)  
     [attribute](#)), 3913  
[iPXEBoot](#) (class in [ironic.drivers.modules.ipxe](#)), 3912  
[IRMCBIOS](#) (class in [ironic.drivers.modules.irmc.bios](#)), 3352  
[IRMCHardware](#) (class in [ironic.drivers.irmc](#)), 4203  
[IRMCInspect](#) (class in [ironic.drivers.modules.irmc.inspect](#)), 4203

<a href="#">3383</a>			<a href="#">ironic.api.controllers.v1.driver</a>
IRMCManagement	(class	in	<a href="#">module, 1986</a>
	<a href="#">ironic.drivers.modules.irmc.management</a> ),		<a href="#">ironic.api.controllers.v1.event</a>
<a href="#">3389</a>			<a href="#">module, 1995</a>
IRMCOperationError,	<a href="#">2307</a>		<a href="#">ironic.api.controllers.v1.node</a>
IRMCPower	(class	in	<a href="#">module, 1996</a>
	<a href="#">ironic.drivers.modules.irmc.power</a> ),		<a href="#">ironic.api.controllers.v1.notification_utils</a>
<a href="#">3403</a>			<a href="#">module, 2053</a>
IRMCPIXEBoot	(class	in	<a href="#">ironic.api.controllers.v1.port</a>
	<a href="#">ironic.drivers.modules.irmc.boot</a> ),		<a href="#">module, 2057</a>
<a href="#">3358</a>			<a href="#">ironic.api.controllers.v1.portgroup</a>
IRMCRAID	(class	in	<a href="#">module, 2068</a>
	<a href="#">ironic.drivers.modules.irmc.raid</a> ),	<a href="#">3413</a>	<a href="#">ironic.api.controllers.v1.ramdisk</a>
IRMCSharedFilesystemNotMounted,			<a href="#">module, 2076</a>
<a href="#">2307</a>			<a href="#">ironic.api.controllers.v1.utils</a>
IRMCVirtualMediaBoot	(class	in	<a href="#">module, 2082</a>
	<a href="#">ironic.drivers.modules.irmc.boot</a> ),		<a href="#">ironic.api.controllers.v1.versions</a>
<a href="#">3363</a>			<a href="#">module, 2166</a>
IRMCVolumeBootMixin	(class	in	<a href="#">ironic.api.controllers.v1.volume</a>
	<a href="#">ironic.drivers.modules.irmc.boot</a> ),		<a href="#">module, 2168</a>
<a href="#">3373</a>			<a href="#">ironic.api.controllers.v1.volume_connector</a>
ironic			<a href="#">module, 2168</a>
	<a href="#">module, 4449</a>		<a href="#">ironic.api.controllers.v1.volume_target</a>
ironic.api			<a href="#">module, 2183</a>
	<a href="#">module, 2224</a>		<a href="#">ironic.api.controllers.version</a>
ironic.api.app			<a href="#">module, 2201</a>
	<a href="#">module, 2207</a>		<a href="#">ironic.api.functions</a>
ironic.api.config			<a href="#">module, 2208</a>
	<a href="#">module, 2208</a>		<a href="#">ironic.api.hooks</a>
ironic.api.controllers			<a href="#">module, 2215</a>
	<a href="#">module, 2203</a>		<a href="#">ironic.api.method</a>
ironic.api.controllers.base			<a href="#">module, 2223</a>
	<a href="#">module, 2198</a>		<a href="#">ironic.api.middleware</a>
ironic.api.controllers.link			<a href="#">module, 2205</a>
	<a href="#">module, 2200</a>		<a href="#">ironic.api.middleware.auth_public_routes</a>
ironic.api.controllers.root			<a href="#">module, 2203</a>
	<a href="#">module, 2201</a>		<a href="#">ironic.api.middleware.json_ext</a>
ironic.api.controllers.v1			<a href="#">module, 2204</a>
	<a href="#">module, 2197</a>		<a href="#">ironic.api.middleware.parsable_error</a>
ironic.api.controllers.v1.allocation			<a href="#">module, 2204</a>
	<a href="#">module, 1958</a>		<a href="#">ironic.api.wsgi</a>
ironic.api.controllers.v1.bios			<a href="#">module, 2224</a>
	<a href="#">module, 1964</a>		<a href="#">ironic.cmd</a>
ironic.api.controllers.v1.chassis			<a href="#">module, 2228</a>
	<a href="#">module, 1966</a>		<a href="#">ironic.cmd.api</a>
ironic.api.controllers.v1.collection			<a href="#">module, 2224</a>
	<a href="#">module, 1972</a>		<a href="#">ironic.cmd.conductor</a>
ironic.api.controllers.v1.conductor			<a href="#">module, 2225</a>
	<a href="#">module, 1975</a>		<a href="#">ironic.cmd.dbsync</a>
ironic.api.controllers.v1.deploy_template			<a href="#">module, 2225</a>
	<a href="#">module, 1979</a>		<a href="#">ironic.cmd.status</a>

module, <a href="#">2227</a>	ironic.common.network
ironic.common	module, <a href="#">2370</a>
module, <a href="#">2505</a>	ironic.common.neutron
ironic.common.args	module, <a href="#">2376</a>
module, <a href="#">2242</a>	ironic.common.nova
ironic.common.boot_devices	module, <a href="#">2403</a>
module, <a href="#">2259</a>	ironic.common.policy
ironic.common.boot_modes	module, <a href="#">2404</a>
module, <a href="#">2261</a>	ironic.common.profiler
ironic.common.cinder	module, <a href="#">2408</a>
module, <a href="#">2263</a>	ironic.common.pxe_utils
ironic.common.components	module, <a href="#">2410</a>
module, <a href="#">2276</a>	ironic.common.raid
ironic.common.config	module, <a href="#">2434</a>
module, <a href="#">2277</a>	ironic.common.release_mappings
ironic.common.context	module, <a href="#">2442</a>
module, <a href="#">2277</a>	ironic.common.rpc
ironic.common.dhcp_factory	module, <a href="#">2444</a>
module, <a href="#">2280</a>	ironic.common.rpc_service
ironic.common.driver_factory	module, <a href="#">2448</a>
module, <a href="#">2283</a>	ironic.common.service
ironic.common.exception	module, <a href="#">2449</a>
module, <a href="#">2300</a>	ironic.common.states
ironic.common.faults	module, <a href="#">2449</a>
module, <a href="#">2323</a>	ironic.common.swift
ironic.common.fsm	module, <a href="#">2468</a>
module, <a href="#">2324</a>	ironic.common.utils
ironic.common.glance_service	module, <a href="#">2476</a>
module, <a href="#">2237</a>	ironic.common.wsgi_service
ironic.common.glance_service.image_service	module, <a href="#">2503</a>
module, <a href="#">2228</a>	ironic.conductor
ironic.common.glance_service.service_utils	module, <a href="#">2749</a>
module, <a href="#">2235</a>	ironic.conductor.allocations
ironic.common.hash_ring	module, <a href="#">2505</a>
module, <a href="#">2330</a>	ironic.conductor.base_manager
ironic.common.i18n	module, <a href="#">2510</a>
module, <a href="#">2330</a>	ironic.conductor.cleaning
ironic.common.image_service	module, <a href="#">2516</a>
module, <a href="#">2331</a>	ironic.conductor.deployments
ironic.common.images	module, <a href="#">2520</a>
module, <a href="#">2343</a>	ironic.conductor.manager
ironic.common.indicator_states	module, <a href="#">2525</a>
module, <a href="#">2363</a>	ironic.conductor.notification_utils
ironic.common.json_rpc	module, <a href="#">2544</a>
module, <a href="#">2242</a>	ironic.conductor.rpcapi
ironic.common.json_rpc.client	module, <a href="#">2550</a>
module, <a href="#">2237</a>	ironic.conductor.steps
ironic.common.json_rpc.server	module, <a href="#">2678</a>
module, <a href="#">2238</a>	ironic.conductor.task_manager
ironic.common.keystone	module, <a href="#">2684</a>
module, <a href="#">2363</a>	ironic.conductor.utils



- module, [2703](#)
- ironic.conf
  - module, [2756](#)
- ironic.conf.agent
  - module, [2749](#)
- ironic.conf.ansible
  - module, [2749](#)
- ironic.conf.api
  - module, [2749](#)
- ironic.conf.audit
  - module, [2749](#)
- ironic.conf.auth
  - module, [2749](#)
- ironic.conf.cinder
  - module, [2751](#)
- ironic.conf.conductor
  - module, [2751](#)
- ironic.conf.console
  - module, [2751](#)
- ironic.conf.database
  - module, [2751](#)
- ironic.conf.default
  - module, [2751](#)
- ironic.conf.deploy
  - module, [2751](#)
- ironic.conf.dhcp
  - module, [2751](#)
- ironic.conf.drac
  - module, [2752](#)
- ironic.conf.glance
  - module, [2752](#)
- ironic.conf.healthcheck
  - module, [2752](#)
- ironic.conf.ibmcl
  - module, [2752](#)
- ironic.conf.ilo
  - module, [2752](#)
- ironic.conf.inspector
  - module, [2752](#)
- ironic.conf.ipmi
  - module, [2752](#)
- ironic.conf.irmc
  - module, [2753](#)
- ironic.conf.iscsi
  - module, [2753](#)
- ironic.conf.json\_rpc
  - module, [2753](#)
- ironic.conf.metrics
  - module, [2753](#)
- ironic.conf.metrics\_statsd
  - module, [2753](#)
- ironic.conf.neutron
  - module, [2753](#)
- ironic.conf.nova
  - module, [2753](#)
- ironic.conf.opts
  - module, [2754](#)
- ironic.conf.pxe
  - module, [2756](#)
- ironic.conf.redfish
  - module, [2756](#)
- ironic.conf.service\_catalog
  - module, [2756](#)
- ironic.conf.snmp
  - module, [2756](#)
- ironic.conf.swift
  - module, [2756](#)
- ironic.conf.xclarity
  - module, [2756](#)
- ironic.db
  - module, [3032](#)
- ironic.db.api
  - module, [2906](#)
- ironic.db.migration
  - module, [3031](#)
- ironic.db.sqlalchemy
  - module, [2906](#)
- ironic.db.sqlalchemy.api
  - module, [2757](#)
- ironic.db.sqlalchemy.migration
  - module, [2886](#)
- ironic.db.sqlalchemy.models
  - module, [2890](#)
- ironic.dhcp
  - module, [3053](#)
- ironic.dhcp.base
  - module, [3032](#)
- ironic.dhcp.neutron
  - module, [3039](#)
- ironic.dhcp.none
  - module, [3046](#)
- ironic.drivers
  - module, [4222](#)
- ironic.drivers.base
  - module, [4026](#)
- ironic.drivers.drac
  - module, [4180](#)
- ironic.drivers.fake\_hardware
  - module, [4182](#)
- ironic.drivers.generic
  - module, [4186](#)
- ironic.drivers.hardware\_type

- module, [4190](#)
- ironic.drivers.ibmc
  - module, [4195](#)
- ironic.drivers.ilo
  - module, [4196](#)
- ironic.drivers.intel\_ipmi
  - module, [4200](#)
- ironic.drivers.ipmi
  - module, [4201](#)
- ironic.drivers.irmc
  - module, [4202](#)
- ironic.drivers.modules
  - module, [4026](#)
- ironic.drivers.modules.agent
  - module, [3579](#)
- ironic.drivers.modules.agent\_base
  - module, [3606](#)
- ironic.drivers.modules.agent\_client
  - module, [3644](#)
- ironic.drivers.modules.agent\_power
  - module, [3685](#)
- ironic.drivers.modules.ansible
  - module, [3062](#)
- ironic.drivers.modules.ansible.deploy
  - module, [3053](#)
- ironic.drivers.modules.boot\_mode\_utils
  - module, [3693](#)
- ironic.drivers.modules.console\_utils
  - module, [3701](#)
- ironic.drivers.modules.deploy\_utils
  - module, [3708](#)
- ironic.drivers.modules.drac
  - module, [3164](#)
- ironic.drivers.modules.drac.bios
  - module, [3062](#)
- ironic.drivers.modules.drac.boot
  - module, [3085](#)
- ironic.drivers.modules.drac.common
  - module, [3090](#)
- ironic.drivers.modules.drac.inspect
  - module, [3093](#)
- ironic.drivers.modules.drac.job
  - module, [3098](#)
- ironic.drivers.modules.drac.management
  - module, [3102](#)
- ironic.drivers.modules.drac.power
  - module, [3116](#)
- ironic.drivers.modules.drac.raid
  - module, [3125](#)
- ironic.drivers.modules.drac.vendor\_passthru
  - module, [3152](#)
- ironic.drivers.modules.fake
  - module, [3753](#)
- ironic.drivers.modules.ibmc
  - module, [3206](#)
- ironic.drivers.modules.ibmc.management
  - module, [3164](#)
- ironic.drivers.modules.ibmc.mappings
  - module, [3180](#)
- ironic.drivers.modules.ibmc.power
  - module, [3180](#)
- ironic.drivers.modules.ibmc.raid
  - module, [3190](#)
- ironic.drivers.modules.ibmc.utils
  - module, [3198](#)
- ironic.drivers.modules.ibmc.vendor
  - module, [3200](#)
- ironic.drivers.modules.ilo
  - module, [3351](#)
- ironic.drivers.modules.ilo.bios
  - module, [3206](#)
- ironic.drivers.modules.ilo.boot
  - module, [3212](#)
- ironic.drivers.modules.ilo.common
  - module, [3250](#)
- ironic.drivers.modules.ilo.console
  - module, [3288](#)
- ironic.drivers.modules.ilo.firmware\_processor
  - module, [3291](#)
- ironic.drivers.modules.ilo.inspect
  - module, [3296](#)
- ironic.drivers.modules.ilo.management
  - module, [3301](#)
- ironic.drivers.modules.ilo.power
  - module, [3331](#)
- ironic.drivers.modules.ilo.raid
  - module, [3338](#)
- ironic.drivers.modules.ilo.vendor
  - module, [3346](#)
- ironic.drivers.modules.image\_cache
  - module, [3843](#)
- ironic.drivers.modules.image\_utils
  - module, [3848](#)
- ironic.drivers.modules.inspect\_utils
  - module, [3866](#)
- ironic.drivers.modules.inspector
  - module, [3868](#)
- ironic.drivers.modules.intel\_ipmi
  - module, [3351](#)
- ironic.drivers.modules.intel\_ipmi.management
  - module, [3351](#)
- ironic.drivers.modules.ipmitool



[module, 3871](#)  
[ironic.drivers.modules.ipxe](#)  
[module, 3912](#)  
[ironic.drivers.modules.irmc](#)  
[module, 3417](#)  
[ironic.drivers.modules.irmc.bios](#)  
[module, 3352](#)  
[ironic.drivers.modules.irmc.boot](#)  
[module, 3358](#)  
[ironic.drivers.modules.irmc.common](#)  
[module, 3376](#)  
[ironic.drivers.modules.irmc.inspect](#)  
[module, 3383](#)  
[ironic.drivers.modules.irmc.management](#)  
[module, 3389](#)  
[ironic.drivers.modules.irmc.power](#)  
[module, 3403](#)  
[ironic.drivers.modules.irmc.raid](#)  
[module, 3413](#)  
[ironic.drivers.modules.iscsi\\_deploy](#)  
[module, 3913](#)  
[ironic.drivers.modules.network](#)  
[module, 3465](#)  
[ironic.drivers.modules.network.common](#)  
[module, 3417](#)  
[ironic.drivers.modules.network.flat](#)  
[module, 3435](#)  
[ironic.drivers.modules.network.neutron](#)  
[module, 3445](#)  
[ironic.drivers.modules.network.noop](#)  
[module, 3456](#)  
[ironic.drivers.modules.noop](#)  
[module, 3945](#)  
[ironic.drivers.modules.noop\\_mgmt](#)  
[module, 3970](#)  
[ironic.drivers.modules.pxe](#)  
[module, 3984](#)  
[ironic.drivers.modules.pxe\\_base](#)  
[module, 3990](#)  
[ironic.drivers.modules.redfish](#)  
[module, 3547](#)  
[ironic.drivers.modules.redfish.bios](#)  
[module, 3465](#)  
[ironic.drivers.modules.redfish.boot](#)  
[module, 3474](#)  
[ironic.drivers.modules.redfish.inspect](#)  
[module, 3489](#)  
[ironic.drivers.modules.redfish.management](#)  
[module, 3493](#)  
[ironic.drivers.modules.redfish.power](#)  
[module, 3531](#)  
[ironic.drivers.modules.redfish.utils](#)  
[module, 3539](#)  
[ironic.drivers.modules.redfish.vendor](#)  
[module, 3543](#)  
[ironic.drivers.modules.snmp](#)  
[module, 4000](#)  
[ironic.drivers.modules.storage](#)  
[module, 3564](#)  
[ironic.drivers.modules.storage.cinder](#)  
[module, 3547](#)  
[ironic.drivers.modules.storage.external](#)  
[module, 3553](#)  
[ironic.drivers.modules.storage.noop](#)  
[module, 3558](#)  
[ironic.drivers.modules.xclarity](#)  
[module, 3579](#)  
[ironic.drivers.modules.xclarity.common](#)  
[module, 3564](#)  
[ironic.drivers.modules.xclarity.management](#)  
[module, 3569](#)  
[ironic.drivers.modules.xclarity.power](#)  
[module, 3575](#)  
[ironic.drivers.redfish](#)  
[module, 4205](#)  
[ironic.drivers.snmp](#)  
[module, 4206](#)  
[ironic.drivers.utils](#)  
[ironic.drivers.xclarity](#)  
[module, 4221](#)  
[ironic.objects](#)  
[module, 4449](#)  
[ironic.objects.allocation](#)  
[module, 4222](#)  
[ironic.objects.base](#)  
[module, 4235](#)  
[ironic.objects.bios](#)  
[module, 4249](#)  
[ironic.objects.chassis](#)  
[module, 4262](#)  
[ironic.objects.conductor](#)  
[module, 4272](#)  
[ironic.objects.deploy\\_template](#)  
[module, 4281](#)  
[ironic.objects.deployment](#)  
[module, 4296](#)  
[ironic.objects.fields](#)  
[module, 4304](#)  
[ironic.objects.indirection](#)  
[module, 4313](#)  
[ironic.objects.node](#)

[module](#), [4324](#)  
[ironic.objects.notification](#)  
     [module](#), [4362](#)  
[ironic.objects.port](#)  
     [module](#), [4368](#)  
[ironic.objects.portgroup](#)  
     [module](#), [4388](#)  
[ironic.objects.trait](#)  
     [module](#), [4403](#)  
[ironic.objects.volume\\_connector](#)  
     [module](#), [4413](#)  
[ironic.objects.volume\\_target](#)  
     [module](#), [4430](#)  
[ironic.version](#)  
     [module](#), [4449](#)  
[ironic-dbsync](#) command line option  
     [--config-dir <DIR>](#), [1699](#)  
     [--config-file <PATH>](#), [1699](#)  
     [--debug](#), [1699](#)  
     [--help](#), [1699](#)  
     [--version](#), [1699](#)  
     [-d](#), [1699](#)  
     [-h](#), [1699](#)  
     [create\\_schema](#), [1699](#)  
     [online\\_data\\_migrations](#), [1699](#)  
     [revision](#), [1699](#)  
     [stamp](#), [1699](#)  
     [upgrade](#), [1699](#)  
     [version](#), [1699](#)  
[IroniCBase](#) (class in [ironic.db.sqlalchemy.models](#)), [2896](#)  
[IroniCCORS](#) (class in [ironic.api.app](#)), [2207](#)  
[IroniCObject](#) (class in [ironic.objects.base](#)), [4235](#)  
[IroniCObjectIndirectionAPI](#) (class in [ironic.objects.indirection](#)), [4313](#)  
[IroniCObjectListBase](#) (class in [ironic.objects.base](#)), [4245](#)  
[IroniCObjectRegistry](#) (class in [ironic.objects.base](#)), [4245](#)  
[IroniCObjectSerializer](#) (class in [ironic.objects.base](#)), [4245](#)  
[is\\_agent\\_token\\_pregenerated\(\)](#) (in module [ironic.conductor.utils](#)), [2714](#)  
[is\\_agent\\_token\\_present\(\)](#) (in module [ironic.conductor.utils](#)), [2716](#)  
[is\\_agent\\_token\\_valid\(\)](#) (in module [ironic.conductor.utils](#)), [2717](#)  
[is\\_equivalent\(\)](#) (in module [ironic.conductor.steps](#)), [2678](#)  
[is\\_fast\\_track\(\)](#) (in module [ironic.conductor.utils](#)), [2718](#)  
[is\\_glance\\_image\(\)](#) (in module [ironic.common.glance\\_service.service\\_utils](#)), [2235](#)  
[is\\_hostname\\_safe\(\)](#) (in module [ironic.common.utils](#)), [2483](#)  
[is\\_image\\_active\(\)](#) (in module [ironic.common.glance\\_service.service\\_utils](#)), [2235](#)  
[is\\_image\\_available\(\)](#) (in module [ironic.common.glance\\_service.service\\_utils](#)), [2235](#)  
[is\\_ipxe\\_enabled\(\)](#) (in module [ironic.common.pxe\\_utils](#)), [2430](#)  
[is\\_iscsi\\_boot\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), [3731](#)  
[is\\_memory\\_insufficient\(\)](#) (in module [ironic.common.utils](#)), [2484](#)  
[is\\_path\\_removed\(\)](#) (in module [ironic.api.controllers.v1.utils](#)), [2147](#)  
[is\\_path\\_updated\(\)](#) (in module [ironic.api.controllers.v1.utils](#)), [2148](#)  
[is\\_regex\\_string\\_in\\_file\(\)](#) (in module [ironic.common.utils](#)), [2487](#)  
[is\\_secure\\_boot\\_requested\(\)](#) (in module [ironic.drivers.modules.boot\\_mode\\_utils](#)), [3695](#)  
[is\\_smartrnic](#) ([ironic.db.sqlalchemy.models.Port](#) attribute), [2901](#)  
[is\\_smartrnic\(\)](#) ([ironic.objects.port.Port](#) property), [4375](#)  
[is\\_smartrnic\(\)](#) ([ironic.objects.port.PortCRUDPayload](#) property), [4387](#)  
[is\\_smartrnic\\_port\(\)](#) (in module [ironic.common.neutron](#)), [2389](#)  
[is\\_software\\_raid\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), [3732](#)  
[is\\_stable\(\)](#) ([ironic.common.fsm.FSM](#) method), [2328](#)  
[is\\_trusted\\_boot\\_requested\(\)](#) (in module [ironic.drivers.modules.boot\\_mode\\_utils](#)), [3697](#)  
[is\\_valid\\_datapath\\_id\(\)](#) (in module [ironic.common.utils](#)), [2487](#)  
[is\\_valid\\_logical\\_name\(\)](#) (in module [ironic.api.controllers.v1.utils](#)), [2149](#)  
[is\\_valid\\_logical\\_name\(\)](#) (in module

*ironic.common.utils*), 2488

*is\_valid\_no\_proxy()* (in module *ironic.common.utils*), 2489

*is\_valid\_node\_name()* (in module *ironic.api.controllers.v1.utils*), 2150

*is\_volume\_attached()* (in module *ironic.common.cinder*), 2274

*is\_volume\_available()* (in module *ironic.common.cinder*), 2275

*is\_whole\_disk\_image()* (in module *ironic.common.images*), 2361

ISCSIBOOT (in module *ironic.common.boot\_devices*), 2260

ISCSIDeploy (class in *ironic.drivers.modules.iscsi\_deploy*), 3913

*issue\_startup\_warnings()* (in module *ironic.cmd.conductor*), 2225

*iswsmefunction()* (in module *ironic.api.functions*), 2211

*items()* (*ironic.common.driver\_factory.BaseDriverFactory* method), 2284

*iter\_nodes()* (*ironic.conductor.base\_manager.BaseConductorManager* method), 2513

## J

*JsonExtensionMiddleware* (class in *ironic.api.middleware*), 2206

*JsonExtensionMiddleware* (class in *ironic.api.middleware.json\_ext*), 2204

*JsonRpcError*, 2239

## K

*kernel\_ref()* (*ironic.objects.deployment.Deployment* property), 4301

*KeystoneFailure*, 2312

*KeystoneUnauthorized*, 2313

*known\_good\_state()* (*ironic.drivers.modules.drac.management.DracWSManManagement* method), 3109

*ks\_exceptions()* (in module *ironic.common.keystone*), 2369

## L

*last\_error()* (*ironic.db.sqlalchemy.models.Allocation* attribute), 2890

*last\_error()* (*ironic.db.sqlalchemy.models.Node* attribute), 2898

*last\_error()* (*ironic.objects.allocation.Allocation* property), 4228

*last\_error()* (*ironic.objects.allocation.AllocationCRUDPayload* property), 4234

*last\_error()* (*ironic.objects.node.Node* property), 4333

*last\_error()* (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4349

*last\_error()* (*ironic.objects.node.NodeCRUDPayload* property), 4344

*last\_error()* (*ironic.objects.node.NodePayload* property), 4353

*last\_error()* (*ironic.objects.node.NodeSetPowerStatePayload* property), 4356

*last\_error()* (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4360

LEGACY\_BIOS (in module *ironic.common.boot\_modes*), 2262

*lessee()* (*ironic.db.sqlalchemy.models.Node* attribute), 2898

*lessee()* (*ironic.objects.node.Node* property), 4333

*lessee()* (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4349

*lessee()* (*ironic.objects.node.NodeCRUDPayload* property), 4344

*lessee()* (*ironic.objects.node.NodePayload* property), 4353

*lessee()* (*ironic.objects.node.NodeSetPowerStatePayload* property), 4356

*lessee()* (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4360

*level()* (*ironic.objects.allocation.AllocationCRUDNotification* property), 4234

*level()* (*ironic.objects.chassis.ChassisCRUDNotification* property), 4271

*level()* (*ironic.objects.deploy\_template.DeployTemplateCRUDNotification* property), 4295

*level()* (*ironic.objects.node.NodeConsoleNotification* property), 4346

*level()* (*ironic.objects.node.NodeCorrectedPowerStateNotification* property), 4347

*level()* (*ironic.objects.node.NodeCRUDNotification* property), 4342

*level()* (*ironic.objects.node.NodeMaintenanceNotification* property), 4351

*level()* (*ironic.objects.node.NodeSetPowerStateNotification* property), 4355

*level()* (*ironic.objects.node.NodeSetProvisionStateNotification* property), 4358

*level()* (*ironic.objects.notification.NotificationBase* property), 4365

*level()* (*ironic.objects.port.PortCRUDNotification* property), 4386

*level()* (*ironic.objects.portgroup.PortgroupCRUDNotification* property), 4386

<i>property</i> ), 4402	<i>ironic.api.controllers.v1.allocation</i> ),
<i>level()</i> ( <i>ironic.objects.volume_connector.VolumeConnector</i> <i>CRUDNotification</i>	
<i>property</i> ), 4428	<i>list_convert_with_links()</i> (in module
<i>level()</i> ( <i>ironic.objects.volume_target.VolumeTarget</i> <i>CRUDNotification</i>	<i>ironic.api.controllers.v1.chassis</i> ), 1972
<i>property</i> ), 4447	<i>list_convert_with_links()</i> (in module
<i>list()</i> ( <i>ironic.objects.allocation.Allocation</i>	<i>ironic.api.controllers.v1.collection</i> ),
<i>class method</i> ), 4228	1972
<i>list()</i> ( <i>ironic.objects.chassis.Chassis</i> <i>class</i> <i>list_convert_with_links()</i> (in module	<i>ironic.api.controllers.v1.conductor</i> ),
<i>method</i> ), 4267	1979
<i>list()</i> ( <i>ironic.objects.conductor.Conductor</i>	<i>list_convert_with_links()</i> (in module
<i>class method</i> ), 4274	<i>ironic.api.controllers.v1.deploy_template</i> ),
<i>list()</i> ( <i>ironic.objects.deploy_template.DeployTemplate</i>	1985
<i>class method</i> ), 4288	
<i>list()</i> ( <i>ironic.objects.deployment.Deployment</i> <i>list_convert_with_links()</i> (in module	<i>ironic.api.controllers.v1.driver</i> ), 1994
<i>class method</i> ), 4301	
<i>list()</i> ( <i>ironic.objects.node.Node</i> <i>class method</i> ),	<i>list_convert_with_links()</i> (in module
4333	<i>ironic.api.controllers.v1.port</i> ), 2067
<i>list()</i> ( <i>ironic.objects.port.Port</i> <i>class method</i> ),	<i>list_convert_with_links()</i> (in module
4375	<i>ironic.api.controllers.v1.portgroup</i> ),
<i>list()</i> ( <i>ironic.objects.portgroup.Portgroup</i>	2075
<i>class method</i> ), 4395	<i>list_convert_with_links()</i> (in module
<i>list()</i> ( <i>ironic.objects.volume_connector.VolumeConnector</i> <i>ironic.api.controllers.v1.volume_connector</i> ),	
<i>class method</i> ), 4420	2183
<i>list()</i> ( <i>ironic.objects.volume_target.VolumeTarget</i> <i>list_convert_with_links()</i> (in module	<i>ironic.api.controllers.v1.volume_target</i> ),
<i>class method</i> ), 4437	2197
<i>list_by_names()</i>	
( <i>ironic.objects.deploy_template.DeployTemplate</i> <i>list_convert_with_links()</i> (in module	<i>ironic.db.api.Connection</i> <i>method</i> ), 2993
<i>class method</i> ), 4290	
<i>list_by_node_id()</i> ( <i>ironic.objects.port.Port</i> <i>list_convert_with_links()</i> (in module	<i>ironic.db.sqlalchemy.api.Connection</i>
<i>class method</i> ), 4378	<i>method</i> ), 2844
<i>list_by_node_id()</i>	
( <i>ironic.objects.portgroup.Portgroup</i> <i>list_convert_with_links()</i> (in module	<i>ironic.conf.cinder</i> ),
<i>class method</i> ), 4397	2751
<i>list_by_node_id()</i>	
( <i>ironic.objects.volume_connector.VolumeConnector</i> <i>list_convert_with_links()</i> (in module	<i>ironic.conf.glance</i> ),
<i>class method</i> ), 4422	2752
<i>list_by_node_id()</i>	
( <i>ironic.objects.volume_target.VolumeTarget</i> <i>list_convert_with_links()</i> (in module	<i>ironic.conf.inspector</i> ),
<i>class method</i> ), 4439	2752
<i>list_by_portgroup_id()</i>	
( <i>ironic.objects.port.Port</i> <i>class method</i> ),	<i>list_opts()</i> (in module <i>ironic.conf.neutron</i> ),
4380	2753
<i>list_by_volume_id()</i>	
( <i>ironic.objects.volume_target.VolumeTarget</i> <i>list_convert_with_links()</i> (in module	<i>ironic.conf.nova</i> ),
<i>class method</i> ), 4441	2753
<i>list_conductor_hardware_interfaces()</i> <i>list_opts()</i> (in module	<i>ironic.conf.json_rpc</i> ),
( <i>ironic.db.api.Connection</i> <i>method</i> ), 2992	2753
<i>list_conductor_hardware_interfaces()</i> <i>list_opts()</i> (in module	<i>ironic.conf.service_catalog</i> ), 2756
( <i>ironic.db.sqlalchemy.api.Connection</i>	
<i>method</i> ), 2843	2756
<i>list_convert_with_links()</i> (in module	<i>list_physical_disks()</i> (in module
	<i>ironic.drivers.modules.drac.raid</i> ), 3146

list\_policies() (in module *ironic.common.policy*), 2407

list\_raid\_controllers() (in module *ironic.drivers.modules.drac.raid*), 3146

list\_raid\_settings() (in module *ironic.drivers.modules.drac.raid*), 3147

list\_unfinished\_jobs() (in module *ironic.drivers.modules.drac.job*), 3099

list\_unfinished\_jobs() (in module *ironic.drivers.modules.drac.vendor\_passthru*), 3159

list\_virtual\_disks() (in module *ironic.drivers.modules.drac.raid*), 3148

ListOfFlexibleDictsField (class in *ironic.objects.fields*), 4306

ListOfObjectsField (class in *ironic.objects.fields*), 4306

ListOfStringsField (class in *ironic.objects.fields*), 4306

load\_driver() (in module *ironic.conductor.task\_manager.TaskManager*), 2693

local\_link\_connection (in module *ironic.db.sqlalchemy.models.Port* attribute), 2901

local\_link\_connection() (in module *ironic.objects.port.Port* property), 4382

local\_link\_connection() (in module *ironic.objects.port.PortCRUDPayload* property), 4387

local\_link\_normalize() (in module *ironic.api.controllers.v1.utils*), 2151

log\_and\_raise\_deployment\_error() (in module *ironic.drivers.modules.agent\_base*), 3639

logical\_disk\_properties() (in module *ironic.api.controllers.v1.driver.DriverRaidController*), 1988

login\_iscsi() (in module *ironic.drivers.modules.iscsi\_deploy*), 3943

logout\_iscsi() (in module *ironic.drivers.modules.iscsi\_deploy*), 3943

LOOKUP\_ALLOWED\_STATES (in module *ironic.common.states*), 2460

lookup\_allowed\_states() (in module *ironic.api.controllers.v1.ramdisk.LookupController*), 2081

LookupController (class in *ironic.api.controllers.v1.ramdisk*), 2079

## M

mac\_address() (in module *ironic.common.args*), 2247

MACAddress (class in *ironic.objects.fields*), 4307

MACAddressField (class in *ironic.objects.fields*), 4308

MACAlreadyExists (in module *ironic.cmd.api*), 2224

main() (in module *ironic.cmd.conductor*), 2225

main() (in module *ironic.cmd.dbsync*), 2226

main() (in module *ironic.cmd.status*), 2227

maintenance (in module *ironic.api.controllers.v1.node.NodesController* attribute), 2045

maintenance (in module *ironic.db.sqlalchemy.models.Node* attribute), 2898

maintenance() (in module *ironic.objects.node.Node* property), 4335

maintenance() (in module *ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4349

maintenance() (in module *ironic.objects.node.NodeCRUDPayload* property), 4344

maintenance() (in module *ironic.objects.node.NodePayload* property), 4353

maintenance() (in module *ironic.objects.node.NodeSetPowerStatePayload* property), 4356

maintenance() (in module *ironic.objects.node.NodeSetProvisionStatePayload* property), 4360

maintenance\_reason (in module *ironic.db.sqlalchemy.models.Node* attribute), 2898

maintenance\_reason() (in module *ironic.objects.node.Node* property), 4335

maintenance\_reason() (in module *ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4349

maintenance\_reason() (in module *ironic.objects.node.NodeCRUDPayload* property), 4344

maintenance\_reason() (in module *ironic.objects.node.NodePayload* property), 4353



erty), 4353

maintenance\_reason() (ironic.objects.node.NodeSetPowerStatePayload property), 4356

maintenance\_reason() (ironic.objects.node.NodeSetProvisionStatePayload property), 4360

make\_link() (in module ironic.api.controllers.link), 2200

make\_persistent\_password\_file() (in module ironic.drivers.modules.console\_utils), 3703

make\_salt() (in module ironic.conductor.utils), 2719

MANAGEABLE (in module ironic.common.states), 2460

management (ironic.api.controllers.v1.node.NodesController attribute), 2045

management (ironic.drivers.base.BareDriver attribute), 4036

management\_interface (ironic.db.sqlalchemy.models.Node attribute), 2898

management\_interface() (ironic.objects.node.Node property), 4335

management\_interface() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4349

management\_interface() (ironic.objects.node.NodeCRUDPayload property), 4344

management\_interface() (ironic.objects.node.NodePayload property), 4353

management\_interface() (ironic.objects.node.NodeSetPowerStatePayload property), 4356

management\_interface() (ironic.objects.node.NodeSetProvisionStatePayload property), 4360

ManagementInterface (class in ironic.drivers.base), 4078

mandatory (ironic.api.functions.FunctionArgument attribute), 2209

ManualManagementHardware (class in ironic.drivers.generic), 4188

mask\_secrets() (in module ironic.objects.notification), 4367

max\_string (ironic.api.controllers.base.Version attribute), 2198

max\_version() (in module ironic.objects.base), 4248

max\_version\_string() (in module ironic.api.controllers.v1.versions), 2166

memoize() (in module ironic.drivers.modules.snmp), 4026

metadata (ironic.db.sqlalchemy.models.IronicBase attribute), 2896

metadata (ironic.drivers.base.VendorMetadata attribute), 4167

method (ironic.drivers.base.VendorMetadata attribute), 4167

MethodNotFound, 2239

methods() (ironic.api.controllers.v1.driver.DriverPassthruController method), 1987

methods() (ironic.api.controllers.v1.node.NodeVendorPassthruController method), 2028

METRICS (in module ironic.drivers.modules.irmc.inspect), 3387

METRICS (in module ironic.drivers.modules.irmc.power), 3412

migrate\_from\_iscsi\_deploy() (ironic.db.api.Connection method), 2993

migrate\_from\_iscsi\_deploy() (ironic.db.sqlalchemy.api.Connection method), 2845

min\_string (ironic.api.controllers.base.Version attribute), 2198

min\_version\_string() (in module ironic.api.controllers.v1.versions), 2167

MissingParameterValue, 2313

MixinVendorInterface (class in ironic.drivers.utils), 4207

mode (ironic.db.sqlalchemy.models.Portgroup attribute), 2902

mode() (ironic.objects.portgroup.Portgroup property), 4399

mode() (ironic.objects.portgroup.PortgroupCRUDPayload property), 4403

model\_query() (in module ironic.db.sqlalchemy.api), 2886

module ironic, 4449

module ironic.api, 2224

module ironic.api.app, 2207

[ironic.api.config, 2208](#)  
[ironic.api.controllers, 2203](#)  
[ironic.api.controllers.base, 2198](#)  
[ironic.api.controllers.link, 2200](#)  
[ironic.api.controllers.root, 2201](#)  
[ironic.api.controllers.v1, 2197](#)  
[ironic.api.controllers.v1.allocation, 1958](#)  
[ironic.api.controllers.v1.bios, 1964](#)  
[ironic.api.controllers.v1.chassis, 1966](#)  
[ironic.api.controllers.v1.collection, 1972](#)  
[ironic.api.controllers.v1.conductor, 1975](#)  
[ironic.api.controllers.v1.deploy\\_template, 1979](#)  
[ironic.api.controllers.v1.driver, 1986](#)  
[ironic.api.controllers.v1.event, 1995](#)  
[ironic.api.controllers.v1.node, 1996](#)  
[ironic.api.controllers.v1.notification\\_utils, 2053](#)  
[ironic.api.controllers.v1.port, 2057](#)  
[ironic.api.controllers.v1.portgroup, 2068](#)  
[ironic.api.controllers.v1.ramdisk, 2076](#)  
[ironic.api.controllers.v1.utils, 2082](#)  
[ironic.api.controllers.v1.versions, 2166](#)  
[ironic.api.controllers.v1.volume, 2168](#)  
[ironic.api.controllers.v1.volume\\_connections, 2168](#)  
[ironic.api.controllers.v1.volume\\_target, 2183](#)  
[ironic.api.controllers.version, 2201](#)  
[ironic.api.functions, 2208](#)  
[ironic.api.hooks, 2215](#)  
[ironic.api.method, 2223](#)  
[ironic.api.middleware, 2205](#)  
[ironic.api.middleware.auth\\_public\\_routes, 2203](#)  
[ironic.api.middleware.json\\_ext, 2204](#)  
[ironic.api.middleware.parsable\\_error, 2204](#)  
[ironic.api.wsgi, 2224](#)  
[ironic.cmd, 2228](#)  
[ironic.cmd.api, 2224](#)  
[ironic.cmd.conductor, 2225](#)  
[ironic.cmd.dbsync, 2225](#)  
[ironic.cmd.status, 2227](#)  
[ironic.common, 2505](#)  
[ironic.common.args, 2242](#)  
[ironic.common.boot\\_devices, 2259](#)  
[ironic.common.boot\\_modes, 2261](#)  
[ironic.common.cinder, 2263](#)  
[ironic.common.components, 2276](#)  
[ironic.common.config, 2277](#)  
[ironic.common.context, 2277](#)  
[ironic.common.dhcp\\_factory, 2280](#)  
[ironic.common.driver\\_factory, 2283](#)  
[ironic.common.exception, 2300](#)  
[ironic.common.faults, 2323](#)  
[ironic.common.fsm, 2324](#)  
[ironic.common.glance\\_service, 2337](#)  
[ironic.common.glance\\_service.image\\_service, 2228](#)  
[ironic.common.glance\\_service.service\\_utils, 2235](#)  
[ironic.common.hash\\_ring, 2330](#)  
[ironic.common.i18n, 2330](#)  
[ironic.common.image\\_service, 2331](#)  
[ironic.common.images, 2343](#)  
[ironic.common.indicator\\_states, 2363](#)  
[ironic.common.json\\_rpc, 2242](#)  
[ironic.common.json\\_rpc.client, 2237](#)  
[ironic.common.json\\_rpc.server, 2238](#)  
[ironic.common.keystone, 2363](#)  
[ironic.common.network, 2370](#)  
[ironic.common.neutron, 2376](#)  
[ironic.common.nova, 2403](#)  
[ironic.common.policy, 2404](#)  
[ironic.common.profiler, 2408](#)  
[ironic.common.pxe\\_utils, 2410](#)

- `ironic.common.raid`, 2434
- `ironic.common.release_mappings`, 2442
- `ironic.common.rpc`, 2444
- `ironic.common.rpc_service`, 2448
- `ironic.common.service`, 2449
- `ironic.common.states`, 2449
- `ironic.common.swift`, 2468
- `ironic.common.utils`, 2476
- `ironic.common.wsgi_service`, 2503
- `ironic.conductor`, 2749
- `ironic.conductor.allocations`, 2505
- `ironic.conductor.base_manager`, 2510
- `ironic.conductor.cleaning`, 2516
- `ironic.conductor.deployments`, 2520
- `ironic.conductor.manager`, 2525
- `ironic.conductor.notification_utils`, 2544
- `ironic.conductor.rpcapi`, 2550
- `ironic.conductor.steps`, 2678
- `ironic.conductor.task_manager`, 2684
- `ironic.conductor.utils`, 2703
- `ironic.conf`, 2756
- `ironic.conf.agent`, 2749
- `ironic.conf.ansible`, 2749
- `ironic.conf.api`, 2749
- `ironic.conf.audit`, 2749
- `ironic.conf.auth`, 2749
- `ironic.conf.cinder`, 2751
- `ironic.conf.conductor`, 2751
- `ironic.conf.console`, 2751
- `ironic.conf.database`, 2751
- `ironic.conf.default`, 2751
- `ironic.conf.deploy`, 2751
- `ironic.conf.dhcp`, 2751
- `ironic.conf.drac`, 2752
- `ironic.conf.glance`, 2752
- `ironic.conf.healthcheck`, 2752
- `ironic.conf.ibmcl`, 2752
- `ironic.conf.ilo`, 2752
- `ironic.conf.inspector`, 2752
- `ironic.conf.ipmi`, 2752
- `ironic.conf.irmc`, 2753
- `ironic.conf.iscsi`, 2753
- `ironic.conf.json_rpc`, 2753
- `ironic.conf.metrics`, 2753
- `ironic.conf.metrics_statsd`, 2753
- `ironic.conf.neutron`, 2753
- `ironic.conf.nova`, 2753
- `ironic.conf.opts`, 2754
- `ironic.conf.pxe`, 2756
- `ironic.conf.redfish`, 2756
- `ironic.conf.service_catalog`, 2756
- `ironic.conf.snmp`, 2756
- `ironic.conf.swift`, 2756
- `ironic.conf.xclarity`, 2756
- `ironic.db`, 3032
- `ironic.db.api`, 2906
- `ironic.db.migration`, 3031
- `ironic.db.sqlalchemy`, 2906
- `ironic.db.sqlalchemy.api`, 2757
- `ironic.db.sqlalchemy.migration`, 2886
- `ironic.db.sqlalchemy.models`, 2890
- `ironic.dhcp`, 3053
- `ironic.dhcp.base`, 3032
- `ironic.dhcp.neutron`, 3039
- `ironic.dhcp.none`, 3046
- `ironic.drivers`, 4222
- `ironic.drivers.base`, 4026
- `ironic.drivers.drac`, 4180
- `ironic.drivers.fake_hardware`, 4182
- `ironic.drivers.generic`, 4186
- `ironic.drivers.hardware_type`, 4190
- `ironic.drivers.ibmcl`, 4195
- `ironic.drivers.ilo`, 4196
- `ironic.drivers.intel_ipmi`, 4200
- `ironic.drivers.ipmi`, 4201
- `ironic.drivers.irmc`, 4202
- `ironic.drivers.modules`, 4026
- `ironic.drivers.modules.agent`, 3579
- `ironic.drivers.modules.agent_base`, 3606
- `ironic.drivers.modules.agent_client`, 3644
- `ironic.drivers.modules.agent_power`, 3685
- `ironic.drivers.modules.ansible`, 3062
- `ironic.drivers.modules.ansible.deploy`, 3053
- `ironic.drivers.modules.boot_mode_utils`, 3693



ironic.drivers.modules.console_utils,	ironic.drivers.modules.ilo.inspect,
3701	3296
ironic.drivers.modules.deploy_utils,	ironic.drivers.modules.ilo.management,
3708	3301
ironic.drivers.modules.drac,	ironic.drivers.modules.ilo.power,
3164	3331
ironic.drivers.modules.drac.bios,	ironic.drivers.modules.ilo.raid,
3062	3338
ironic.drivers.modules.drac.boot,	ironic.drivers.modules.ilo.vendor,
3085	3346
ironic.drivers.modules.drac.common,	ironic.drivers.modules.image_cache,
3090	3843
ironic.drivers.modules.drac.inspect,	ironic.drivers.modules.image_utils,
3093	3848
ironic.drivers.modules.drac.job,	ironic.drivers.modules.inspect_utils,
3098	3866
ironic.drivers.modules.drac.management,	ironic.drivers.modules.inspector,
3102	3868
ironic.drivers.modules.drac.power,	ironic.drivers.modules.intel_ipmi,
3116	3351
ironic.drivers.modules.drac.raid,	ironic.drivers.modules.intel_ipmi.management,
3125	3351
ironic.drivers.modules.drac.vendor_utils,	ironic.drivers.modules.ipmitool,
3152	3871
ironic.drivers.modules.fake,	ironic.drivers.modules.ipxe,
3753	3912
ironic.drivers.modules.ibmc,	ironic.drivers.modules.irmc,
3206	3417
ironic.drivers.modules.ibmc.management,	ironic.drivers.modules.irmc.bios,
3164	3352
ironic.drivers.modules.ibmc.mappings,	ironic.drivers.modules.irmc.boot,
3180	3358
ironic.drivers.modules.ibmc.power,	ironic.drivers.modules.irmc.common,
3180	3376
ironic.drivers.modules.ibmc.raid,	ironic.drivers.modules.irmc.inspect,
3190	3383
ironic.drivers.modules.ibmc.utils,	ironic.drivers.modules.irmc.management,
3198	3389
ironic.drivers.modules.ibmc.vendor,	ironic.drivers.modules.irmc.power,
3200	3403
ironic.drivers.modules.ilo,	ironic.drivers.modules.irmc.raid,
3351	3413
ironic.drivers.modules.ilo.bios,	ironic.drivers.modules.iscsi_deploy,
3206	3913
ironic.drivers.modules.ilo.boot,	ironic.drivers.modules.network,
3212	3465
ironic.drivers.modules.ilo.common,	ironic.drivers.modules.network.common,
3250	3417
ironic.drivers.modules.ilo.console,	ironic.drivers.modules.network.flat,
3288	3435
ironic.drivers.modules.ilo.firmware_processor,	ironic.drivers.modules.network.neutron,
3291	

[3445](#)  
[ironic.drivers.modules.network.noop](#), [ironic.objects.bios](#), [4249](#)  
[3456](#)  
[ironic.drivers.modules.noop](#), [ironic.objects.chassis](#), [4262](#)  
[3945](#)  
[ironic.drivers.modules.noop\\_mgmt](#), [ironic.objects.conductor](#), [4272](#)  
[3970](#)  
[ironic.drivers.modules.pxe](#), [3984](#)  
[ironic.drivers.modules.pxe\\_base](#), [ironic.objects.deploy\\_template](#), [4281](#)  
[3990](#)  
[ironic.drivers.modules.redfish](#), [ironic.objects.deployment](#), [4296](#)  
[3547](#)  
[ironic.drivers.modules.redfish.bios](#), [ironic.objects.fields](#), [4304](#)  
[3465](#)  
[ironic.drivers.modules.redfish.boot](#), [ironic.objects.indirection](#), [4313](#)  
[3474](#)  
[ironic.drivers.modules.redfish.inspect](#), [ironic.objects.node](#), [4324](#)  
[3489](#)  
[ironic.drivers.modules.redfish.management](#), [ironic.objects.notification](#), [4362](#)  
[3493](#)  
[ironic.drivers.modules.redfish.power](#), [ironic.objects.port](#), [4368](#)  
[3531](#)  
[ironic.drivers.modules.redfish.utils](#), [ironic.objects.portgroup](#), [4388](#)  
[3539](#)  
[ironic.drivers.modules.redfish.vendor](#), [ironic.objects.trait](#), [4403](#)  
[3543](#)  
[ironic.drivers.modules.snmp](#), [ironic.objects.volume\\_connector](#), [4413](#)  
[4000](#)  
[ironic.drivers.modules.storage](#), [ironic.objects.volume\\_target](#), [4430](#)  
[3564](#)  
[ironic.drivers.modules.storage.chainer](#), [ironic.objects.version](#), [4449](#)  
[3547](#)  
[ironic.drivers.modules.storage.extra](#), [mount \(\) \(in module ironic.common.utils\)](#), [2490](#)  
[3553](#)  
[ironic.drivers.modules.storage.noop](#), [N](#)  
[3558](#)  
[ironic.drivers.modules.xclarity](#), [name \(ironic.api.functions.FunctionArgument attribute\)](#), [2209](#)  
[3579](#)  
[ironic.drivers.modules.xclarity.name](#), [name \(ironic.api.functions.FunctionDefinition attribute\)](#), [2211](#)  
[3564](#)  
[ironic.drivers.modules.xclarity.management](#), [name \(ironic.db.sqlalchemy.models.Allocation attribute\)](#), [2890](#)  
[3569](#)  
[ironic.drivers.modules.xclarity.power](#), [name \(ironic.db.sqlalchemy.models.BIOSSetting attribute\)](#), [2891](#)  
[3575](#)  
[ironic.drivers.redfish](#), [name \(ironic.db.sqlalchemy.models.DeployTemplate attribute\)](#), [2894](#)  
[ironic.drivers.snmp](#), [name \(ironic.db.sqlalchemy.models.Node attribute\)](#), [2898](#)  
[ironic.drivers.utils](#), [name \(ironic.db.sqlalchemy.models.Port attribute\)](#), [2901](#)  
[ironic.drivers.xclarity](#), [name \(ironic.db.sqlalchemy.models.Portgroup attribute\)](#), [2902](#)  
[ironic.objects](#), [name \(in module ironic.common.args\)](#), [2249](#)  
[ironic.objects.allocation](#), [name \(\) \(ironic.objects.allocation.Allocation property\)](#), [4230](#)  
[ironic.objects.base](#), [name \(\) \(ironic.objects.allocation.AllocationCRUDPayload property\)](#), [4234](#)  
[ironic.objects.bios](#), [name \(\) \(ironic.objects.bios.BIOSSetting property\)](#), [4253](#)  
[ironic.objects.chassis](#), [name \(\) \(ironic.objects.deploy\\_template.DeployTemplate property\)](#), [4291](#)  
[ironic.objects.conductor](#), [name \(\) \(ironic.objects.deploy\\_template.DeployTemplateCRUDPA property\)](#), [4296](#)  
[ironic.objects.deploy\\_template](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.deployment](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.fields](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.indirection](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.node](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.notification](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.port](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.portgroup](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.trait](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.volume\\_connector](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.volume\\_target](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)  
[ironic.objects.version](#), [name \(\) \(ironic.objects.node.Node property\)](#), [4335](#)

name () (*ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4360  
 property), 4349 NetworkError, 2313  
 name () (*ironic.objects.node.NodeCRUDPayload* property), 4344 NetworkInterface (class in *ironic.drivers.base*), 4115  
 name () (*ironic.objects.node.NodePayload* property), 4353 NetworkInterfaceFactory (class in *ironic.common.driver\_factory*), 2284  
 name () (*ironic.objects.node.NodeSetPowerStatePayload* property), 4356 NeutronDHCPApi (class in *ironic.dhcp.neutron*), 3039  
 name () (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4360 NeutronNetwork (class in *ironic.drivers.modules.network.neutron*), 3445  
 name () (*ironic.objects.port.Port* property), 4382  
 name () (*ironic.objects.port.PortCRUDPayload* property), 4387 NeutronNetworkInterfaceMixin (class in *ironic.common.neutron*), 2376  
 name () (*ironic.objects.portgroup.Portgroup* property), 4399 NeutronVIFPortIDMixin (class in *ironic.drivers.modules.network.common*), 3417  
 name () (*ironic.objects.portgroup.PortgroupCRUDPayload* property), 4403 NIC (in module *ironic.common.components*), 2277  
 names () (*ironic.common.driver\_factory.BaseDriverFactory* property), 2284 Neutron (class in *ironic.drivers.modules.noop*), 3947  
 need\_power\_on () NoConsole (class in *ironic.drivers.modules.noop*), 3953  
 (*ironic.drivers.base.NetworkInterface* method), 4124 NoConsolePid, 2313  
 need\_power\_on () Node (class in *ironic.db.sqlalchemy.models*), 2896  
 (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3449 node (class in *ironic.objects.node*), 4324  
 network (*ironic.drivers.base.BareDriver* attribute), 4037 node (*ironic.db.sqlalchemy.models.NodeTag* attribute), 2900  
 network\_data (*ironic.db.sqlalchemy.models.Node* attribute), 2898 node (*ironic.db.sqlalchemy.models.NodeTrait* attribute), 2900  
 network\_data () (*ironic.objects.node.Node* property), 4335 node () (*ironic.conductor.task\_manager.TaskManager* property), 2693  
 network\_interface (*ironic.db.sqlalchemy.models.Node* attribute), 2898 node\_cache\_bios\_settings () (in module *ironic.conductor.utils*), 2720  
 network\_interface () NODE\_CLASS\_OID (in module *ironic.drivers.modules.irmc.inspect*), 3388  
 (*ironic.objects.node.Node* property), 4335  
 network\_interface () node\_convert\_with\_links () (in module *ironic.objects.node.NodeCorrectedPowerStatePayload* property), 4349 *ironic.api.controllers.v1.node*), 2049  
 network\_interface () node\_get\_boot\_mode () (in module *ironic.conductor.utils*), 2720  
 (*ironic.objects.node.NodeCRUDPayload* property), 4344 node\_id (*ironic.db.sqlalchemy.models.Allocation* attribute), 2890  
 network\_interface () node\_id (*ironic.db.sqlalchemy.models.BIOSSetting* attribute), 2891  
 (*ironic.objects.node.NodePayload* property), 4353 node\_id (*ironic.db.sqlalchemy.models.NodeTag* attribute), 2900  
 network\_interface () node\_id (*ironic.db.sqlalchemy.models.NodeTrait* attribute), 2900  
 (*ironic.objects.node.NodeSetPowerStatePayload* property), 4356  
 network\_interface () node\_id (*ironic.db.sqlalchemy.models.Port* attribute), 2901  
 (*ironic.objects.node.NodeSetProvisionStatePayload* attribute), 2901

[node\\_id\(ironic.db.sqlalchemy.models.Portgroup attribute\), 2902](#)  
[node\\_id\(ironic.db.sqlalchemy.models.VolumeConnector attribute\), 2903](#)  
[node\\_id\(ironic.db.sqlalchemy.models.VolumeTarget attribute\), 2904](#)  
[node\\_id\(\) \(ironic.objects.allocation.Allocation property\), 4230](#)  
[node\\_id\(\) \(ironic.objects.bios.BIOSSetting property\), 4253](#)  
[node\\_id\(\) \(ironic.objects.port.Port property\), 4382](#)  
[node\\_id\(\) \(ironic.objects.portgroup.Portgroup property\), 4399](#)  
[node\\_id\(\) \(ironic.objects.trait.Trait property\), 4408](#)  
[node\\_id\(\) \(ironic.objects.volume\\_connector.VolumeConnector property\), 4424](#)  
[node\\_id\(\) \(ironic.objects.volume\\_target.VolumeTarget property\), 4443](#)  
[node\\_list\\_convert\\_with\\_links\(\) \(in module ironic.api.controllers.v1.node\), 2050](#)  
[node\\_mapping\(ironic.objects.deployment.Deployment attribute\), 4303](#)  
[node\\_power\\_action\(\) \(in module ironic.conductor.utils\), 2722](#)  
[node\\_sanitize\(\) \(in module ironic.api.controllers.v1.node\), 2050](#)  
[node\\_set\\_boot\\_device\(\) \(in module ironic.conductor.utils\), 2725](#)  
[node\\_set\\_boot\\_mode\(\) \(in module ironic.conductor.utils\), 2727](#)  
[node\\_states\\_convert\(\) \(in module ironic.api.controllers.v1.node\), 2051](#)  
[node\\_tag\\_exists\(\) \(ironic.db.api.Connection method\), 2995](#)  
[node\\_tag\\_exists\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2846](#)  
[node\\_trait\\_exists\(\) \(ironic.db.api.Connection method\), 2996](#)  
[node\\_trait\\_exists\(\) \(ironic.db.sqlalchemy.api.Connection method\), 2847](#)  
[node\\_uuid\(\) \(ironic.objects.allocation.Allocation property\), 4234](#)  
[node\\_uuid\(\) \(ironic.objects.deployment.Deployment property\), 4303](#)  
[node\\_uuid\(\) \(ironic.objects.port.PortCRUDPayload property\), 4387](#)  
[node\\_uuid\(\) \(ironic.objects.portgroup.PortgroupCRUDPayload property\), 4403](#)  
[node\\_uuid\(\) \(ironic.objects.volume\\_connector.VolumeConnector property\), 4429](#)  
[node\\_uuid\(\) \(ironic.objects.volume\\_target.VolumeTargetCRUDPayload property\), 4448](#)  
[node\\_wait\\_for\\_power\\_state\(\) \(in module ironic.conductor.utils\), 2729](#)  
[NodeAllocationController \(class in ironic.api.controllers.v1.allocation\), 1963](#)  
[NodeAlreadyExists, 2314](#)  
[NodeAssociated, 2314](#)  
[NodeBiosController \(class in ironic.api.controllers.v1.bios\), 1964](#)  
[NodeCleaningFailure, 2314](#)  
[NodeConsoleController \(class in ironic.api.controllers.v1.node\), 2006](#)  
[NodeConsoleNotEnabled, 2315](#)  
[NodeConsoleNotification \(class in ironic.objects.node\), 4345](#)  
[NodeCorrectedPowerStateNotification \(class in ironic.objects.node\), 4346](#)  
[NodeCorrectedPowerStatePayload \(class in ironic.objects.node\), 4347](#)  
[NodeCRUDNotification \(class in ironic.objects.node\), 4341](#)  
[NodeCRUDPayload \(class in ironic.objects.node\), 4342](#)  
[NodeInMaintenance, 2315](#)  
[NodeIsRetired, 2315](#)  
[NodeLocked, 2315](#)  
[NodeMaintenanceController \(class in ironic.api.controllers.v1.node\), 2007](#)  
[NodeMaintenanceFailure, 2315](#)  
[NodeMaintenanceNotification \(class in ironic.objects.node\), 4350](#)  
[NodeManagementController \(class in ironic.api.controllers.v1.node\), 2009](#)  
[NodeNotFound, 2315](#)  
[NodeNotLocked, 2315](#)  
[NodePayload \(class in ironic.objects.node\), 4351](#)  
[NodeProtected, 2316](#)  
[nodes \(ironic.api.controllers.v1.chassis.ChassisController attribute\), 1970](#)  
[NodesController \(class in ironic.api.controllers.v1.node\), 2029](#)  
[NodeSetPowerStateNotification \(class](#)

- in ironic.objects.node*), 4354
  - NodeSetPowerStatePayload (class *in ironic.objects.node*), 4355
  - NodeSetProvisionStateNotification (class *in ironic.objects.node*), 4357
  - NodeSetProvisionStatePayload (class *in ironic.objects.node*), 4358
  - NodeStatesController (class *in ironic.api.controllers.v1.node*), 2010
  - NodeTag (class *in ironic.db.sqlalchemy.models*), 2899
  - NodeTagNotFound, 2316
  - NodeTrait (class *in ironic.db.sqlalchemy.models*), 2900
  - NodeTraitNotFound, 2316
  - NodeTraitsController (class *in ironic.api.controllers.v1.node*), 2024
  - NodeVendorPassthruController (class *in ironic.api.controllers.v1.node*), 2028
  - NodeVIFController (class *in ironic.api.controllers.v1.node*), 2026
  - NoDriversLoaded, 2313
  - NoExceptionTracebackHook (class *in ironic.api.hooks*), 2219
  - NoFreeConductorWorker, 2313
  - NoFreeIPMITerminalPorts, 2314
  - NoFreePhysicalPorts, 2314
  - NoInspect (class *in ironic.drivers.modules.noop*), 3956
  - non\_vendor\_interfaces() (ironic.drivers.base.BareDriver property), 4037
  - NoneDHCPApi (class *in ironic.dhcp.none*), 3046
  - NoopManagement (class *in ironic.drivers.modules.noop\_mgmt*), 3970
  - NoopNetwork (class *in ironic.drivers.modules.network.noop*), 3456
  - NoopStorage (class *in ironic.drivers.modules.storage.noop*), 3558
  - NoRAID (class *in ironic.drivers.modules.noop*), 3958
  - NoRescue (class *in ironic.drivers.modules.noop*), 3964
  - normalize\_mac() (in module *ironic.drivers.utils*), 4217
  - NOSTATE (in module *ironic.common.states*), 2461
  - NotAcceptable, 2316
  - NotAuthorized, 2316
  - NotFound, 2316
  - NotificationBase (class *in ironic.objects.notification*), 4363
  - NotificationLevel (class *in ironic.objects.fields*), 4308
  - NotificationLevelField (class *in ironic.objects.fields*), 4309
  - NotificationPayloadBase (class *in ironic.objects.notification*), 4365
  - NotificationPayloadError, 2316
  - NotificationPublisher (class *in ironic.objects.notification*), 4367
  - NotificationSchemaKeyError, 2317
  - NotificationSchemaObjectError, 2317
  - NotificationStatus (class *in ironic.objects.fields*), 4309
  - NotificationStatusField (class *in ironic.objects.fields*), 4309
  - notify\_conductor\_resume\_clean() (in module *ironic.conductor.utils*), 2731
  - notify\_conductor\_resume\_deploy() (in module *ironic.conductor.utils*), 2731
  - notify\_conductor\_resume\_operation() (in module *ironic.conductor.utils*), 2731
  - NoValidDefaultForInterface, 2314
  - NoValidHost, 2314
  - NoVendor (class *in ironic.drivers.modules.noop*), 3967
- ## O
- obj (ironic.api.controllers.v1.utils.PassthruResponse attribute), 2082
  - OBJ\_BASE\_CLASS (ironic.objects.base.IronicObjectSerializer attribute), 4246
  - OBJ\_PROJECT\_NAMESPACE (ironic.objects.base.IronicObject attribute), 4236
  - obj\_refresh() (ironic.objects.base.IronicObject method), 4243
  - OBJ\_SERIAL\_NAMESPACE (ironic.objects.base.IronicObject attribute), 4236
  - object() (ironic.objects.notification.EventType property), 4362
  - object\_action() (ironic.conductor.manager.ConductorManager method), 2534
  - object\_action() (ironic.conductor.rpcapi.ConductorAPI





owner() (*ironic.objects.node.NodeCorrectedPowerStatePayload* module *ironic.common.utils*), 2491  
 property), 4349 ParseError, 2240

owner() (*ironic.objects.node.NodeCRUDPayload* passthru() (in module *ironic.drivers.base*),  
 property), 4344 4180

owner() (*ironic.objects.node.NodePayload* PassthruResponse (class in  
 property), 4353 *ironic.api.controllers.v1.utils*), 2082

owner() (*ironic.objects.node.NodeSetPowerStatePayload* PasswordFileFailedToCreate, 2317  
 property), 4356 patch() (in module *ironic.common.args*), 2251

owner() (*ironic.objects.node.NodeSetProvisionStatePayload* (in module *ironic.api.controllers.v1.allocation.AllocationsController*  
 property), 4360 method), 1962

patch() (*ironic.api.controllers.v1.chassis.ChassisController*  
 method), 1970

patch() (*ironic.api.controllers.v1.deploy\_template.DeployTemplateController*  
 method), 1983

patch() (*ironic.api.controllers.v1.node.NodesController*  
 method), 2045

patch() (*ironic.api.controllers.v1.port.PortsController*  
 method), 2065

patch() (*ironic.api.controllers.v1.portgroup.PortgroupsController*  
 method), 2074

patch() (*ironic.api.controllers.v1.volume\_connector.VolumeConnectorController*  
 method), 2176

patch() (*ironic.api.controllers.v1.volume\_target.VolumeTargetsController*  
 method), 2191

patch\_update\_changed\_fields() (in  
 module *ironic.api.controllers.v1.utils*),  
 2154

patch\_validate\_allowed\_fields() (in  
 module *ironic.api.controllers.v1.utils*),  
 2156

patched\_validate\_with\_schema() (in  
 module *ironic.api.controllers.v1.utils*),  
 2157

PatchError, 2317

PathNotFound, 2317

payload() (*ironic.objects.allocation.AllocationCRUDNotification*  
 property), 4234

payload() (*ironic.objects.chassis.ChassisCRUDNotification*  
 property), 4271

payload() (*ironic.objects.deploy\_template.DeployTemplateCRUDNotification*  
 property), 4295

payload() (*ironic.objects.node.NodeConsoleNotification*  
 property), 4346

payload() (*ironic.objects.node.NodeCorrectedPowerStateNotification*  
 property), 4347

payload() (*ironic.objects.node.NodeCRUDNotification*  
 property), 4342

payload() (*ironic.objects.node.NodeMaintenanceNotification*  
 property), 4351

payload() (*ironic.objects.node.NodeSetPowerStateNotification*  
 property), 4355

## P

ParsableErrorMiddleware (class in  
*ironic.api.middleware*), 2207

ParsableErrorMiddleware (class in  
*ironic.api.middleware.parsable\_error*),  
 2205

parse\_args() (in module  
*ironic.common.config*), 2277

parse\_driver\_info() (in module  
*ironic.common.pxe\_utils*), 2430

parse\_driver\_info() (in module  
*ironic.drivers.modules.drac.common*),  
 3091

parse\_driver\_info() (in module  
*ironic.drivers.modules.ibmc.utils*),  
 3198

parse\_driver\_info() (in module  
*ironic.drivers.modules.ilo.boot*), 3246

parse\_driver\_info() (in module  
*ironic.drivers.modules.ilo.common*),  
 3266

parse\_driver\_info() (in module  
*ironic.drivers.modules.irmc.common*),  
 3380

parse\_driver\_info() (in module  
*ironic.drivers.modules.redfish.utils*),  
 3542

parse\_driver\_info() (in module  
*ironic.drivers.modules.xclarity.common*),  
 3566

parse\_headers() (*ironic.api.controllers.base.Version*  
 static method), 2199

parse\_image\_id() (in module  
*ironic.common.glance\_service.service\_utils*),  
 2236

parse\_instance\_info() (in module  
*ironic.drivers.modules.deploy\_utils*),  
 3732

parse\_instance\_info\_capabilities()

[payload\(\) \(ironic.objects.node.NodeSetProvisionStateNotification.payload\), 2067](#)  
[property\), 4358](#)  
[payload\(\) \(ironic.objects.port.PortCRUDNotification.payload\), 4387](#)  
[property\), 4387](#)  
[payload\(\) \(ironic.objects.portgroup.PortgroupCRUDNotification.payload\), 4402](#)  
[property\), 4402](#)  
[payload\(\) \(ironic.objects.volume\\_connector.VolumeConnectorCRUDNotification.payload\), 4429](#)  
[property\), 4429](#)  
[payload\(\) \(ironic.objects.volume\\_target.VolumeTargetCRUDNotification.payload\), 4447](#)  
[property\), 4447](#)  
[physical\\_network \(ironic.db.sqlalchemy.models.Port attribute\), 2901](#)  
[physical\\_network \(ironic.objects.port.Port property\), 4382](#)  
[physical\\_network \(ironic.objects.port.PortCRUDPayload property\), 4387](#)  
[PHYSNET\\_PARAM\\_NAME \(in module ironic.common.neutron\), 2378](#)  
[PlaybookNotFound, 3062](#)  
[plug\\_port\\_to\\_tenant\\_network \(in module ironic.drivers.modules.network.common\), 3434](#)  
[policy\\_deprecation\\_check \(in module ironic.api.hooks\), 2223](#)  
[pop\\_node\\_nested\\_field \(in module ironic.common.utils\), 2493](#)  
[populate\\_node\\_uuid \(in module ironic.api.controllers.v1.utils\), 2159](#)  
[populate\\_schema \(ironic.objects.notification.NotificationPayloadBase method\), 4366](#)  
[populate\\_storage\\_driver\\_internal\\_in \(in module ironic.drivers.modules.deploy\\_utils\), 3734](#)  
[Port \(class in ironic.db.sqlalchemy.models\), 2900](#)  
[Port \(class in ironic.objects.port\), 4368](#)  
[port\\_changed \(ironic.drivers.base.NetworkInterface method\), 4124](#)  
[port\\_changed \(ironic.drivers.modules.network.common.NetworkVIFPort method\), 3420](#)  
[port\\_changed \(ironic.drivers.modules.network.noop.NoopNetwork method\), 3458](#)  
[port\\_sanitize \(in module](#)

[PortAlreadyExists, 2317](#)  
[PortCRUDNotification \(class in ironic.objects.port\), 4386](#)  
[PortCRUDNotificationPayload \(class in ironic.objects.portgroup\), 4387](#)  
[Portgroup \(class in ironic.objects.portgroup\), 4388](#)  
[PortgroupAlreadyExists, 2318](#)  
[PortgroupCRUDNotification \(class in ironic.objects.portgroup\), 4401](#)  
[PortgroupCRUDPayload \(class in ironic.objects.portgroup\), 4402](#)  
[PortgroupDuplicateName, 2318](#)  
[PortgroupMACAlreadyExists, 2318](#)  
[PortgroupNotEmpty, 2318](#)  
[PortgroupNotFound, 2318](#)  
[PortgroupPhysnetInconsistent, 2319](#)  
[PortgroupsController \(class in ironic.api.controllers.v1.portgroup\), 2068](#)  
[PortNotFound, 2318](#)  
[PortsController \(class in ironic.api.controllers.v1.port\), 2057](#)  
[post \(ironic.api.controllers.v1.allocation.AllocationsController method\), 1963](#)  
[post \(ironic.api.controllers.v1.chassis.ChassisController method\), 1971](#)  
[post \(ironic.api.controllers.v1.deploy\\_template.DeployTemplate method\), 1984](#)  
[post \(ironic.api.controllers.v1.event.EventsController method\), 1995](#)



[post \(\) \(ironic.api.controllers.v1.node.NodesController method\), 2010](#)  
[method\), 2046](#) [POWER\\_FAILURE \(in module](#)  
[post \(\) \(ironic.api.controllers.v1.node.NodeVIFController ironic.common.faults\), 2323](#)  
[method\), 2027](#) [power\\_interface](#)  
[post \(\) \(ironic.api.controllers.v1.port.PortsController \(ironic.db.sqlalchemy.models.Node](#)  
[method\), 2066](#) [attribute\), 2898](#)  
[post \(\) \(ironic.api.controllers.v1.portgroup.PortgroupController](#)  
[method\), 2075](#) [power\\_interface \(\)](#)  
[\(ironic.objects.node.Node property\),](#)  
[post \(\) \(ironic.api.controllers.v1.ramdisk.HeartbeatController\), 4335](#)  
[method\), 2076](#) [power\\_interface \(\)](#)  
[post \(\) \(ironic.api.controllers.v1.volume\\_connector.VolumeConnectorNodeCorrectedPowerStatePayload](#)  
[method\), 2181](#) [property\), 4349](#)  
[post \(\) \(ironic.api.controllers.v1.volume\\_target.VolumeTargetController](#)  
[method\), 2195](#) [\(ironic.objects.node.NodeCRUDPayload](#)  
[property\), 4344](#)  
[post\\_clean\\_step\\_hook \(\) \(in module](#) [power\\_interface \(\)](#)  
[ironic.drivers.modules.agent\\_base\),](#) [\(ironic.objects.node.NodePayload prop-](#)  
[3640](#) [erty\), 4353](#)  
[post\\_configuration \(\)](#)  
[\(ironic.drivers.modules.redfish.bios.RedfishBIOS](#) [power\\_interface \(\)](#)  
[method\), 3470](#) [\(ironic.objects.node.NodeSetPowerStatePayload](#)  
[property\), 4356](#)  
[post\\_deploy\\_step\\_hook \(\) \(in module](#) [power\\_interface \(\)](#)  
[ironic.drivers.modules.agent\\_base\),](#) [\(ironic.objects.node.NodeSetProvisionStatePayload](#)  
[3642](#) [property\), 4360](#)  
[POST\\_FINISHEDPOST\\_STATE \(in module](#) [POWER\\_OFF \(in module ironic.common.states\),](#)  
[ironic.drivers.modules.ilo.common\),](#) [2461](#)  
[3250](#)  
[POST\\_INPOST\\_STATE \(in module](#) [power\\_off \(\) \(ironic.drivers.modules.agent\\_client.AgentClient](#)  
[ironic.drivers.modules.ilo.common\),](#) [method\), 3672](#)  
[3250](#) [power\\_off \(\) \(ironic.drivers.modules.snmp.SNMPDriverBase](#)  
[method\), 4009](#)  
[POST\\_INPOSTDISCOVERY\\_STATE \(in mod-](#) [POWER\\_ON \(in module ironic.common.states\),](#)  
[ule ironic.drivers.modules.ilo.common\),](#) [2461](#)  
[3250](#)  
[POST\\_NULL\\_STATE \(in module](#) [power\\_on \(\) \(ironic.drivers.modules.snmp.SNMPDriverBase](#)  
[ironic.drivers.modules.ilo.common\),](#) [method\), 4009](#)  
[3250](#) [power\\_on\\_node\\_if\\_needed \(\) \(in module](#)  
[POST\\_POWEROFF\\_STATE \(in module](#) [ironic.conductor.utils\), 2731](#)  
[ironic.drivers.modules.ilo.common\),](#) [power\\_reset \(\)](#)  
[3251](#) [\(ironic.drivers.modules.snmp.SNMPDriverBase](#)  
[method\), 4010](#)  
[post\\_reset \(\) \(ironic.drivers.modules.redfish.bios.RedfishBIOS](#)  
[method\), 3471](#) [power\\_state \(ironic.db.sqlalchemy.models.Node](#)  
[attribute\), 2898](#)  
[POST\\_RESET\\_STATE \(in module](#) [power\\_state \(\)](#)  
[ironic.drivers.modules.ilo.common\),](#) [\(ironic.drivers.modules.snmp.SNMPDriverBase](#)  
[3251](#) [method\), 4010](#)  
[POST\\_UNKNOWN\\_STATE \(in module](#) [power\\_state \(\) \(ironic.objects.node.Node](#)  
[ironic.drivers.modules.ilo.common\),](#) [property\), 4335](#)  
[3251](#)  
[POWER \(in module ironic.common.components\),](#) [power\\_state \(\)](#)  
[2277](#) [\(ironic.objects.node.NodeCorrectedPowerStatePayload](#)  
[power \(ironic.drivers.base.BareDriver attribute\),](#) [property\), 4349](#)  
[4037](#) [power\\_state \(\)](#)  
[power \(\) \(ironic.api.controllers.v1.node.NodeStatesController\), 4349](#) [\(ironic.objects.node.NodeCRUDPayload](#)

<code>property)</code> , 4344	<code>prepare_disk_image()</code> (in module <code>ironic.drivers.modules.image_utils</code> ), 3861
<code>power_state()</code> ( <code>ironic.objects.node.NodePayload</code> property), 4353	<code>prepare_floppy_image()</code> (in module <code>ironic.drivers.modules.image_utils</code> ), 3864
<code>power_state()</code> ( <code>ironic.objects.node.NodeSetPowerStatePayload</code> property), 4357	<code>prepare_host()</code> ( <code>ironic.conductor.base_manager.BaseConductorManager</code> method), 2515
<code>power_state()</code> ( <code>ironic.objects.node.NodeSetProvisionStatePayload</code> property), 4360	<code>prepare_image()</code> ( <code>ironic.drivers.modules.agent_client.AgentClient</code> method), 3674
<code>power_state_error_handler()</code> (in module <code>ironic.conductor.utils</code> ), 2733	<code>prepare_inband_cleaning()</code> (in module <code>ironic.drivers.modules.deploy_utils</code> ), 3735
<code>power_state_for_network_configuration()</code> (in module <code>ironic.conductor.utils</code> ), 2734	<code>prepare_instance()</code> ( <code>ironic.drivers.base.BootInterface</code> method), 4053
<code>power_update()</code> (in module <code>ironic.common.nova</code> ), 2403	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.fake.FakeBoot</code> method), 3764
<code>PowerInterface</code> (class in <code>ironic.drivers.base</code> ), 4136	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.ilo.boot.IloIPXEBoot</code> method), 3242
<code>PowerStateFailure</code> , 2319	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.ilo.boot.IloIPXEBoot</code> method), 3214
<code>prepare()</code> ( <code>ironic.common.json_rpc.client.Client</code> method), 2238	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot</code> method), 3220
<code>prepare()</code> ( <code>ironic.drivers.base.DeployInterface</code> method), 4066	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.irmc.boot.IRMCPXEBoot</code> method), 3359
<code>prepare()</code> ( <code>ironic.drivers.modules.agent.AgentDeploy</code> method), 3583	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot</code> method), 3366
<code>prepare()</code> ( <code>ironic.drivers.modules.ansible.deploy.AnsibleDeploy</code> method), 3056	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.pxe_base.PXEBaseMixin</code> method), 3993
<code>prepare()</code> ( <code>ironic.drivers.modules.fake.FakeDeploy</code> method), 3779	<code>prepare_instance()</code> ( <code>ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot</code> method), 3479
<code>prepare()</code> ( <code>ironic.drivers.modules.iscsi_deploy.ISCSIDeploy</code> method), 3917	<code>prepare_instance_boot()</code> ( <code>ironic.drivers.modules.agent.AgentDeployMixin</code> method), 3588
<code>prepare()</code> ( <code>ironic.drivers.modules.pxe.PXERamdiskDeploy</code> method), 3986	<code>prepare_instance_boot()</code> ( <code>ironic.drivers.modules.iscsi_deploy.ISCSIDeploy</code> method), 3920
<code>prepare_boot_iso()</code> (in module <code>ironic.drivers.modules.image_utils</code> ), 3853	
<code>prepare_cleaning()</code> ( <code>ironic.drivers.base.DeployInterface</code> method), 4068	
<code>prepare_cleaning()</code> ( <code>ironic.drivers.modules.agent_base.AgentBaseMixin</code> method), 3607	
<code>prepare_cleaning()</code> ( <code>ironic.drivers.modules.ansible.deploy.AnsibleDeploy</code> method), 3057	
<code>prepare_configdrive_image()</code> (in module <code>ironic.drivers.modules.image_utils</code> ), 3856	
<code>prepare_deploy_iso()</code> (in module <code>ironic.drivers.modules.image_utils</code> ), 3858	

[prepare\\_instance\\_pxe\\_config\(\)](#) (in [module `ironic.common.pxe\_utils`](#)), 2432 [process\\_fw\\_on\(\)](#) ([ironic.drivers.modules.ilo.firmware\\_processor.FirmwareProcessor](#) [method](#)), 2693  
[prepare\\_instance\\_to\\_boot\(\)](#) ([ironic.drivers.modules.agent\\_base.AgentDeployMixin](#) [method](#)), 3293  
[prepare\\_node\\_for\\_deploy\(\)](#) (in [module `ironic.drivers.modules.ilo.boot`](#)), 3248 [process\\_launcher\(\)](#) (in [module `ironic.common.service`](#)), 2449  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.base.BootInterface](#) [method](#)), 4054 [process\\_next\\_step\(\)](#) ([ironic.drivers.modules.agent\\_base.AgentDeployMixin](#) [method](#)), 3622  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.fake.FakeBoot](#) [method](#)), 3765 [process\\_next\\_step\(\)](#) ([ironic.drivers.modules.agent\\_base.HeartbeatMixin](#) [method](#)), 3633  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.ilo.boot.IloIPXEBoot](#) [method](#)), 3243 [process\\_next\\_step\(\)](#) ([ironic.drivers.modules.ansible.deploy.AnsibleDeploy](#) [method](#)), 3058  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.ilo.boot.IloIPXEBoot](#) [method](#)), 3215 [properties](#) ([ironic.db.sqlalchemy.models.Node](#) [attribute](#)), 2898  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.ilo.boot.IloIPXEBoot](#) [method](#)), 3215 [properties](#) ([ironic.db.sqlalchemy.models.Portgroup](#) [attribute](#)), 2902  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot](#) [method](#)), 3222 [properties](#) ([ironic.db.sqlalchemy.models.VolumeTarget](#) [attribute](#)), 2904  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot](#) [method](#)), 3234 [properties](#) ([ironic.api.controllers.v1.driver.DriversController](#) [method](#)), 1990  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.irmc.boot.IRMCPXEBoot](#) [method](#)), 3360 [properties](#) ([ironic.objects.node.Node](#) [property](#)), 4335  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.irmc.boot.IRMCPXEBoot](#) [method](#)), 3360 [properties](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) [property](#)), 4349  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.irmc.boot.IRMCPXEBoot](#) [method](#)), 3360 [properties](#) ([ironic.objects.node.NodeCRUDPayload](#) [property](#)), 4344  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.irmc.boot.IRMCPXEBoot](#) [method](#)), 3366 [properties](#) ([ironic.objects.node.NodePayload](#) [property](#)), 4353  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.pxe\\_base.PXEBaseMixin](#) [method](#)), 3993 [properties](#) ([ironic.objects.node.NodeSetPowerStatePayload](#) [property](#)), 4357  
[prepare\\_ramdisk\(\)](#) ([ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot](#) [method](#)), 3482 [properties](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) [property](#)), 4361  
[prepare\\_service\(\)](#) (in [module `ironic.common.service`](#)), 2449 [properties](#) ([ironic.objects.portgroup.Portgroup](#) [property](#)), 4399  
[previous\\_provision\\_state\(\)](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) [property](#)), 4360 [properties](#) ([ironic.objects.portgroup.PortgroupCRUDPayload](#) [property](#)), 4403  
[previous\\_target\\_provision\\_state\(\)](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) [property](#)), 4360 [properties](#) ([ironic.objects.volume\\_target.VolumeTarget](#) [property](#)), 4443  
[priority](#) ([ironic.db.sqlalchemy.models.DeployTemplateStep](#) [attribute](#)), 2895 [properties](#) ([ironic.objects.volume\\_target.VolumeTargetCRUDPayload](#) [property](#)), 4448  
[process\\_event\(\)](#) ([ironic.common.fsm.FSM](#) [method](#)), 2329 [protected](#) ([ironic.db.sqlalchemy.models.Node](#) [attribute](#)), 2898  
[process\\_event\(\)](#) ([ironic.conductor.task\\_manager.TaskManager](#) [method](#)), 2329 [protected](#) ([ironic.objects.node.Node](#) [property](#)), 4335  
[process\\_event\(\)](#) ([ironic.conductor.task\\_manager.TaskManager](#) [method](#)), 2329 [protected](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) [property](#)), 4349  
[process\\_event\(\)](#) ([ironic.conductor.task\\_manager.TaskManager](#) [method](#)), 2329 [protected](#) ([ironic.objects.node.NodeCRUDPayload](#) [property](#)), 4344

protected() (ironic.objects.node.NodePayload property), 4361  
 property), 4353  
 provision\_updated\_at  
 protected() (ironic.objects.node.NodeSetPowerStatePayload), 4357  
 (ironic.db.sqlalchemy.models.Node attribute), 2898  
 protected() (ironic.objects.node.NodeSetProvisionStatePayload), 4361  
 provision\_updated\_at() (ironic.objects.node.Node property), 4335  
 protected\_reason (ironic.db.sqlalchemy.models.Node provision\_updated\_at() attribute), 2898  
 (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350  
 protected\_reason() (ironic.objects.node.Node property), provision\_updated\_at() 4335  
 (ironic.objects.node.NodeCRUDPayload property), 4344  
 protected\_reason() (ironic.objects.node.NodeCorrectedPowerStatePayload), 4350  
 provision\_updated\_at() (ironic.objects.node.NodePayload property), 4353  
 protected\_reason() (ironic.objects.node.NodeCRUDPayload provision\_updated\_at() property), 4344  
 (ironic.objects.node.NodeSetPowerStatePayload property), 4357  
 protected\_reason() (ironic.objects.node.NodePayload property), provision\_updated\_at() 4353  
 (ironic.objects.node.NodeSetProvisionStatePayload property), 4361  
 protected\_reason() (ironic.objects.node.NodeSetPowerStatePayload), 4357  
 provisioning\_error\_handler() (in module ironic.conductor.utils), 2735  
 protected\_reason() PublicUrlHook (class in ironic.api.hooks), 2220  
 (ironic.objects.node.NodeSetProvisionStatePayload property), 4361  
 publish\_image()  
 provider() (ironic.common.dhcp\_factory.DHCPFactory (ironic.drivers.modules.image\_utils.ImageHandler property), 2280  
 method), 3848  
 provision() (ironic.api.controllers.v1.node.NodeStateController(ironic.objects.allocation.AllocationCRUDNotification method), 2013  
 property), 4234  
 provision\_state publisher() (ironic.objects.chassis.ChassisCRUDNotification property), 4271  
 (ironic.db.sqlalchemy.models.Node attribute), 2898  
 publisher() (ironic.objects.deploy\_template.DeployTemplateCRUDNotification property), 4295  
 provision\_state() (ironic.objects.node.Node property), publisher() (ironic.objects.node.NodeConsoleNotification property), 4346  
 4335  
 publisher() (ironic.objects.node.NodeCorrectedPowerStateNotification property), 4347  
 provision\_state() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350  
 publisher() (ironic.objects.node.NodeCRUDNotification property), 4342  
 provision\_state() (ironic.objects.node.NodeCRUDPayload property), 4344  
 publisher() (ironic.objects.node.NodeMaintenanceNotification property), 4351  
 provision\_state() (ironic.objects.node.NodePayload property), publisher() (ironic.objects.node.NodeSetPowerStateNotification property), 4355  
 4353  
 publisher() (ironic.objects.node.NodeSetProvisionStateNotification property), 4358  
 provision\_state() (ironic.objects.node.NodeSetPowerStatePayload), 4357  
 publisher() (ironic.objects.notification.NotificationBase property), 4365  
 provision\_state() publisher() (ironic.objects.port.PortCRUDNotification property), 4387  
 (ironic.objects.node.NodeSetProvisionStatePayload property), 4387

[publisher\(\)](#) ([ironic.objects.portgroup.PortgroupCRUDPayload](#) attribute), 4402  
[publisher\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnectorCRUDPayload](#) attribute), 4429  
[publisher\(\)](#) ([ironic.objects.volume\\_target.VolumeTargetCRUDPayload](#) attribute), 4447  
[put\(\)](#) ([ironic.api.controllers.v1.node.BootDeviceController](#) property), 4350  
[put\(\)](#) ([ironic.api.controllers.v1.node.IndicatorController](#) method), 2002  
[put\(\)](#) ([ironic.api.controllers.v1.node.InjectNmiController](#) method), 2003  
[put\(\)](#) ([ironic.api.controllers.v1.node.NodeConsoleController](#) method), 2007  
[put\(\)](#) ([ironic.api.controllers.v1.node.NodeMaintenanceController](#) method), 2008  
[put\(\)](#) ([ironic.api.controllers.v1.node.NodeTraitsController](#) method), 2025  
[PXE](#) (in module [ironic.common.boot\\_devices](#)), 2260  
[pxe\\_enabled\(\)](#) ([ironic.db.sqlalchemy.models.Port](#) attribute), 2901  
[pxe\\_enabled\(\)](#) ([ironic.objects.port.Port](#) property), 4382  
[pxe\\_enabled\(\)](#) ([ironic.objects.port.PortCRUDPayload](#) property), 4387  
[PXEBaseMixin](#) (class in [ironic.drivers.modules.pxe\\_base](#)), 3990  
[PXEBoot](#) (class in [ironic.drivers.modules.pxe](#)), 3984  
[PXERamdiskDeploy](#) (class in [ironic.drivers.modules.pxe](#)), 3984  
**R**  
[raid\(\)](#) ([ironic.api.controllers.v1.driver.DriversController](#) attribute), 1991  
[raid\(\)](#) ([ironic.drivers.base.BareDriver](#) attribute), 4038  
[raid\(\)](#) ([ironic.api.controllers.v1.node.NodeStatesController](#) method), 2022  
[RAID\\_APPLY\\_CONFIGURATION\\_ARGSINFO](#) (in module [ironic.drivers.base](#)), 4154  
[RAID\\_APPLY\\_CONFIGURATION\\_ARGSINFO](#) ([ironic.drivers.modules.ibm.mc.raid.IbmCRaid](#) attribute), 3190  
[raid\\_config\(\)](#) ([ironic.db.sqlalchemy.models.Node](#) attribute), 2898  
[raid\\_config\(\)](#) ([ironic.objects.node.Node](#) property), 4335  
[raid\\_interface\(\)](#) ([ironic.objects.node.Node](#) property), 2898  
[raid\\_interface\(\)](#) ([ironic.objects.node.NodeCRUDPayload](#) property), 4336  
[raid\\_interface\(\)](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) property), 4344  
[raid\\_interface\(\)](#) ([ironic.objects.node.NodePayload](#) property), 4353  
[raid\\_interface\(\)](#) ([ironic.objects.node.NodeSetPowerStatePayload](#) property), 4357  
[raid\\_interface\(\)](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) property), 4361  
[RAIDInterface](#) (class in [ironic.drivers.base](#)), 4143  
[ramdisk\\_ref\(\)](#) ([ironic.objects.deployment.Deployment](#) property), 4303  
[REBOOT](#) (in module [ironic.common.states](#)), 2462  
[reboot\(\)](#) ([ironic.drivers.base.PowerInterface](#) method), 4138  
[reboot\(\)](#) ([ironic.drivers.modules.agent\\_client.AgentClient](#) method), 3677  
[reboot\(\)](#) ([ironic.drivers.modules.agent\\_power.AgentPower](#) method), 3688  
[reboot\(\)](#) ([ironic.drivers.modules.drac.power.DracWSManPower](#) method), 3120  
[reboot\(\)](#) ([ironic.drivers.modules.fake.FakePower](#) method), 3816  
[reboot\(\)](#) ([ironic.drivers.modules.ibm.mc.power.IBMCPower](#) method), 3184  
[reboot\(\)](#) ([ironic.drivers.modules.ilo.power.IloPower](#) method), 3334  
[reboot\(\)](#) ([ironic.drivers.modules.ipmitool.IPMIPower](#) method), 3890  
[reboot\(\)](#) ([ironic.drivers.modules.irmc.power.IRMCPower](#) method), 3407  
[reboot\(\)](#) ([ironic.drivers.modules.redfish.power.RedfishPower](#) method), 3534  
[reboot\(\)](#) ([ironic.drivers.modules.snmp.SNMPPower](#) method), 4020  
[reboot\(\)](#) ([ironic.drivers.modules.xclarity.power.XClarityPower](#) method), 3576  
[reboot\\_and\\_finish\\_deploy\(\)](#) ([ironic.drivers.modules.agent\\_base.AgentDeployMixin](#) method), 4537



[method](#)), [3623](#)  
[reboot\\_to\\_instance\(\)](#)  
     ([ironic.drivers.modules.agent\\_base.HeartbeatMixin](#)  
     [method](#)), [3633](#)  
[REBUILD](#) (in module [ironic.common.states](#)), [2462](#)  
[RedfishBIOS](#) (class in [ironic.drivers.modules.redfish.bios](#)),  
     [3465](#)  
[RedfishConnectionError](#), [2319](#)  
[RedfishError](#), [2319](#)  
[RedfishHardware](#) (class in [ironic.drivers.redfish](#)), [4205](#)  
[RedfishInspect](#) (class in [ironic.drivers.modules.redfish.inspect](#)),  
     [3489](#)  
[RedfishManagement](#) (class in [ironic.drivers.modules.redfish.management](#)),  
     [3493](#)  
[RedfishPower](#) (class in [ironic.drivers.modules.redfish.power](#)),  
     [3531](#)  
[RedfishVendorPassthru](#) (class in [ironic.drivers.modules.redfish.vendor](#)),  
     [3543](#)  
[RedfishVirtualMediaBoot](#) (class in [ironic.drivers.modules.redfish.boot](#)),  
     [3474](#)  
[refresh\(\)](#) ([ironic.objects.allocation.Allocation](#)  
     [method](#)), [4230](#)  
[refresh\(\)](#) ([ironic.objects.chassis.Chassis](#)  
     [method](#)), [4269](#)  
[refresh\(\)](#) ([ironic.objects.conductor.Conductor](#)  
     [method](#)), [4276](#)  
[refresh\(\)](#) ([ironic.objects.deploy\\_template.DeployTemplate](#)  
     [method](#)), [4291](#)  
[refresh\(\)](#) ([ironic.objects.deployment.Deployment](#)  
     [method](#)), [4303](#)  
[refresh\(\)](#) ([ironic.objects.node.Node](#) [method](#)),  
     [4336](#)  
[refresh\(\)](#) ([ironic.objects.port.Port](#) [method](#)),  
     [4382](#)  
[refresh\(\)](#) ([ironic.objects.portgroup.Portgroup](#)  
     [method](#)), [4399](#)  
[refresh\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnector](#)  
     [method](#)), [4425](#)  
[refresh\(\)](#) ([ironic.objects.volume\\_target.VolumeTarget](#)  
     [method](#)), [4443](#)  
[refresh\\_clean\\_steps\(\)](#)  
     ([ironic.drivers.modules.agent\\_base.HeartbeatMixin](#)  
     [method](#)), [3634](#)  
[refresh\\_steps\(\)](#)

    ([ironic.drivers.modules.agent\\_base.AgentDeployMixin](#)  
     [method](#)), [3624](#)  
     [refresh\\_steps\(\)](#)  
     ([ironic.drivers.modules.agent\\_base.HeartbeatMixin](#)  
     [method](#)), [3634](#)  
     [register\(\)](#) ([ironic.objects.conductor.Conductor](#)  
     class [method](#)), [4277](#)  
     [register\\_all\(\)](#) (in module [ironic.objects](#)),  
     [4449](#)  
     [register\\_auth\\_opts\(\)](#) (in module  
     [ironic.conf.auth](#)), [2750](#)  
     [register\\_conductor\(\)](#)  
     ([ironic.db.api.Connection](#) [method](#)),  
     [2997](#)  
     [register\\_conductor\(\)](#)  
     ([ironic.db.sqlalchemy.api.Connection](#)  
     [method](#)), [2848](#)  
     [register\\_conductor\\_hardware\\_interfaces\(\)](#)  
     ([ironic.db.api.Connection](#) [method](#)), [2999](#)  
     [register\\_conductor\\_hardware\\_interfaces\(\)](#)  
     ([ironic.db.sqlalchemy.api.Connection](#)  
     [method](#)), [2850](#)  
     [register\\_hardware\\_interfaces\(\)](#)  
     ([ironic.objects.conductor.Conductor](#)  
     [method](#)), [4279](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.agent](#)), [2749](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.ansible](#)), [2749](#)  
     [register\\_opts\(\)](#) (in module [ironic.conf.api](#)),  
     [2749](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.audit](#)), [2749](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.cinder](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.conductor](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.console](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.database](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.default](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.deploy](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.dhcp](#)), [2751](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.drac](#)), [2752](#)  
     [register\\_opts\(\)](#) (in module  
     [ironic.conf.glance](#)), [2752](#)

[register\\_opts\(\)](#) (in module [ironic.conf.healthcheck](#)), 2752  
[register\\_opts\(\)](#) (in module [ironic.conf.ibmcc](#)), 2752  
[register\\_opts\(\)](#) (in module [ironic.conf.ilo](#)), 2752  
[register\\_opts\(\)](#) (in module [ironic.conf.inspector](#)), 2752  
[register\\_opts\(\)](#) (in module [ironic.conf.ipmi](#)), 2752  
[register\\_opts\(\)](#) (in module [ironic.conf.irmc](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.iscsi](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.json\\_rpc](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.metrics](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.metrics\\_statsd](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.neutron](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.nova](#)), 2753  
[register\\_opts\(\)](#) (in module [ironic.conf.pxe](#)), 2756  
[register\\_opts\(\)](#) (in module [ironic.conf.redfish](#)), 2756  
[register\\_opts\(\)](#) (in module [ironic.conf.service\\_catalog](#)), 2756  
[register\\_opts\(\)](#) (in module [ironic.conf.snmp](#)), 2756  
[register\\_opts\(\)](#) (in module [ironic.conf.swift](#)), 2756  
[register\\_opts\(\)](#) (in module [ironic.conf.xclarity](#)), 2756  
[registration\\_hook\(\)](#) ([ironic.objects.base.IronicObjectRegistry](#) method), 4245  
[reject\\_fields\\_in\\_newer\\_versions\(\)](#) (in module [ironic.api.controllers.v1.node](#)), 2051  
[reject\\_patch\\_in\\_newer\\_versions\(\)](#) (in module [ironic.api.controllers.v1.node](#)), 2051  
[release\(\)](#) ([ironic.objects.node.Node](#) class method), 4336  
[release\\_node\(\)](#) ([ironic.db.api.Connection](#) method), 3001  
[release\\_node\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), 2852  
[release\\_port\(\)](#) (in module [ironic.drivers.modules.console\\_utils](#)), 3703  
[release\\_resources\(\)](#) ([ironic.conductor.task\\_manager.TaskManager](#) method), 2696  
[remove\(\)](#) ([ironic.drivers.modules.ilo.firmware\\_processor.FirmwareProcessor](#) method), 3291  
[remove\\_agent\\_url\(\)](#) (in module [ironic.conductor.utils](#)), 2737  
[remove\\_cleaning\\_network\(\)](#) ([ironic.drivers.base.NetworkInterface](#) method), 4126  
[remove\\_cleaning\\_network\(\)](#) ([ironic.drivers.modules.network.flat.FlatNetwork](#) method), 3440  
[remove\\_cleaning\\_network\(\)](#) ([ironic.drivers.modules.network.neutron.NeutronNetwork](#) method), 3450  
[remove\\_cleaning\\_network\(\)](#) ([ironic.drivers.modules.network.noop.NoopNetwork](#) method), 3460  
[remove\\_http\\_instance\\_symlink\(\)](#) (in module [ironic.drivers.modules.deploy\\_utils](#)), 3738  
[remove\\_image\\_from\\_swift\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), 3268  
[remove\\_image\\_from\\_web\\_server\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), 3269  
[remove\\_inspection\\_network\(\)](#) ([ironic.drivers.base.NetworkInterface](#) method), 4126  
[remove\\_inspection\\_network\(\)](#) ([ironic.drivers.modules.network.flat.FlatNetwork](#) method), 3440  
[remove\\_inspection\\_network\(\)](#) ([ironic.drivers.modules.network.neutron.NeutronNetwork](#) method), 3450  
[remove\\_neutron\\_ports\(\)](#) (in module [ironic.common.neutron](#)), 2390  
[remove\\_node\\_rescue\\_password\(\)](#) (in module [ironic.conductor.utils](#)), 2737  
[remove\\_node\\_traits\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), 2540  
[remove\\_node\\_traits\(\)](#)

(*ironic.conductor.rpcapi.ConductorAPI*  
 method), 2642  
*remove\_ports\_from\_network()* (in mod-  
 ule *ironic.common.neutron*), 2391  
*remove\_provisioning\_network()*  
 (*ironic.drivers.base.NetworkInterface*  
 method), 4127  
*remove\_provisioning\_network()*  
 (*ironic.drivers.modules.network.flat.FlatNetwork*  
 method), 3441  
*remove\_provisioning\_network()*  
 (*ironic.drivers.modules.network.neutron.NeutronNetwork*  
 method), 3451  
*remove\_provisioning\_network()*  
 (*ironic.drivers.modules.network.noop.NoopNetwork*  
 method), 3460  
*remove\_rescuing\_network()*  
 (*ironic.drivers.base.NetworkInterface*  
 method), 4128  
*remove\_rescuing\_network()*  
 (*ironic.drivers.modules.network.flat.FlatNetwork*  
 method), 3442  
*remove\_rescuing\_network()*  
 (*ironic.drivers.modules.network.neutron.NeutronNetwork*  
 method), 3452  
*remove\_single\_or\_list\_of\_files()*  
 (in module *ironic.drivers.modules.ilo.common*),  
 3270  
*remove\_vifs\_from\_node()* (in module  
*ironic.common.network*), 2376  
*render\_template()* (in module  
*ironic.common.utils*), 2495  
*replace\_node\_id\_with\_uuid()* (in mod-  
 ule *ironic.api.controllers.v1.utils*), 2161  
*replace\_node\_uuid\_with\_id()* (in mod-  
 ule *ironic.api.controllers.v1.utils*), 2162  
*RequestContext* (class in  
*ironic.common.context*), 2277  
*RequestContextSerializer* (class in  
*ironic.common.rpc*), 2444  
*require\_exclusive\_lock()* (in module  
*ironic.conductor.task\_manager*), 2702  
*RESCUE* (in module *ironic.common.states*), 2462  
*rescue* (*ironic.drivers.base.BareDriver* at-  
 tribute), 4038  
*rescue()* (*ironic.drivers.base.RescueInterface*  
 method), 4156  
*rescue()* (*ironic.drivers.modules.agent.AgentRescue*  
 method), 3597  
*rescue()* (*ironic.drivers.modules.fake.FakeRescue*  
 method), 3827  
*rescue()* (*ironic.drivers.modules.noop.NoRescue*  
 method), 3965  
*RESCUE\_ABORT\_FAILURE* (in module  
*ironic.common.faults*), 2324  
*rescue\_interface*  
 (*ironic.db.sqlalchemy.models.Node*  
 attribute), 2898  
*rescue\_interface()*  
 (*ironic.objects.node.Node* property),  
 4337  
*rescue\_interface()*  
 (*ironic.objects.node.NodeCorrectedPowerStatePayload*  
 property), 4350  
*rescue\_interface()*  
 (*ironic.objects.node.NodeCRUDPayload*  
 property), 4344  
*rescue\_interface()*  
 (*ironic.objects.node.NodePayload* prop-  
 erty), 4353  
*rescue\_interface()*  
 (*ironic.objects.node.NodeSetPowerStatePayload*  
 property), 4357  
*rescue\_interface()*  
 (*ironic.objects.node.NodeSetProvisionStatePayload*  
 property), 4361  
*rescue\_or\_deploy\_mode()* (in module  
*ironic.drivers.modules.deploy\_utils*),  
 3738  
*RESCUEFAIL* (in module *ironic.common.states*),  
 2462  
*RescueInterface* (class in  
*ironic.drivers.base*), 4155  
*RESCUEWAIT* (in module *ironic.common.states*),  
 2462  
*RESCUING* (in module *ironic.common.states*),  
 2463  
*rescuing\_error\_handler()* (in module  
*ironic.conductor.utils*), 2738  
*reservation* (*ironic.db.sqlalchemy.models.Node*  
 attribute), 2899  
*reservation()* (*ironic.objects.node.Node*  
 property), 4337  
*reserve()* (*ironic.objects.node.Node* class  
 method), 4337  
*reserve\_node()* (*ironic.db.api.Connection*  
 method), 3003  
*reserve\_node()*  
 (*ironic.db.sqlalchemy.api.Connection*  
 method), 2854  
*reset()* (*ironic.common.hash\_ring.HashRingManager*



<code>class method), 2330</code>	<code>resource_class()</code>
<code>reset() (ironic.common.json_rpc.server.WSGIService method), 2240</code>	<code>(ironic.objects.node.NodeSetPowerStatePayload property), 4357</code>
<code>reset() (ironic.common.wsgi_service.WSGIService resource_class() method), 2504</code>	<code>(ironic.objects.node.NodeSetProvisionStatePayload property), 4361</code>
<code>reset_bios_to_default() (ironic.drivers.modules.ilo.management.IloManagement boot_device() method), 3317</code>	<code>(ironic.drivers.modules.redfish.management.RedfishManagement config() method), 3518</code>
<code>reset_idrac() (ironic.drivers.modules.drac.management.DracWSManManagement config() method), 3110</code>	<code>(ironic.drivers.modules.irmc.management.IRMCMManagement config() method), 3197</code>
<code>reset_ilo() (ironic.drivers.modules.ilo.management.IloManagement boot_device() method), 3318</code>	<code>restore_power_state_if_needed() (in module ironic.conductor.utils), 2739</code>
<code>reset_ilo_credential() (ironic.drivers.modules.ilo.management.IloManagement boot_device() method), 3319</code>	<code>(ironic.db.sqlalchemy.models.Node attribute), 2899</code>
<code>reset_secure_boot_keys_to_default() (ironic.drivers.modules.ilo.management.IloManagement boot_device() method), 3321</code>	<code>retired() (ironic.objects.node.Node property), 4339</code>
<code>reset_secure_boot_keys_to_default() (ironic.drivers.modules.redfish.management.RedfishManagement config() method), 3517</code>	<code>retired() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350</code>
<code>resolve_type() (ironic.api.functions.FunctionArgument method), 2209</code>	<code>(ironic.objects.node.NodeCRUDPayload property), 4344</code>
<code>resolve_types() (ironic.api.functions.FunctionDefinition method), 2211</code>	<code>retired() (ironic.objects.node.NodePayload property), 4353</code>
<code>resource_class (ironic.db.sqlalchemy.models.Allocation attribute), 2891</code>	<code>retired() (ironic.objects.node.NodeSetPowerStatePayload property), 4357</code>
<code>resource_class (ironic.db.sqlalchemy.models.Node attribute), 2899</code>	<code>retired() (ironic.objects.node.NodeSetProvisionStatePayload property), 4361</code>
<code>resource_class() (ironic.objects.allocation.Allocation property), 4231</code>	<code>retired_reason (ironic.db.sqlalchemy.models.Node attribute), 2899</code>
<code>resource_class() (ironic.objects.allocation.AllocationCRUDPayload property), 4234</code>	<code>retired_reason() (ironic.objects.node.Node property), 4339</code>
<code>resource_class() (ironic.objects.node.Node property), 4339</code>	<code>retired_reason() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350</code>
<code>resource_class() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350</code>	<code>retired_reason() (ironic.objects.node.NodeCRUDPayload property), 4345</code>
<code>resource_class() (ironic.objects.node.NodeCRUDPayload property), 4344</code>	<code>retired_reason() (ironic.objects.node.NodePayload property), 4354</code>
<code>resource_class() (ironic.objects.node.NodePayload property), 4353</code>	<code>retired_reason() (ironic.objects.node.NodeSetPowerStatePayload property), 4357</code>
	<code>retired_reason() (ironic.objects.node.NodeSetProvisionStatePayload property), 4361</code>
	<code>retry_interval (ironic.drivers.modules.snmp.SNMPDriverBase attribute), 4011</code>

[retry\\_on\\_outdated\\_cache\(\)](#) (in module [ironic.drivers.modules.snmp](#)), 4026  
[return\\_type](#) ([ironic.api.functions.FunctionDefinition](#) attribute), 2211  
[revert\\_dictionary\(\)](#) (in module [ironic.drivers.modules.ibmclutils](#)), 3200  
[revision](#)  
     [ironic-dbsync](#) command line option, 1699  
[revision](#) command line option  
     --autogenerate, 1701  
     --help, 1701  
     --message <MESSAGE>, 1701  
     -h, 1701  
     -m <MESSAGE>, 1701  
[revision\(\)](#) (in module [ironic.db.migration](#)), 3031  
[revision\(\)](#) (in module [ironic.db.sqlalchemy.migration](#)), 2887  
[revision\(\)](#) ([ironic.cmd.dbsync.DBCommand](#) method), 2226  
[ring\(\)](#) ([ironic.common.hash\\_ring.HashRingManager](#) property), 2330  
[rmtree\\_without\\_raise\(\)](#) (in module [ironic.common.utils](#)), 2496  
[rollback\\_ports\(\)](#) (in module [ironic.common.neutron](#)), 2391  
[root\(\)](#) (in module [ironic.api.controllers.root](#)), 2201  
[root\\_device\(\)](#)  
     ([ironic.objects.deployment.Deployment](#) property), 4304  
[root\\_gib\(\)](#) ([ironic.objects.deployment.Deployment](#) property), 4304  
[RootController](#) (class in [ironic.api.controllers.root](#)), 2201  
[RPC\\_API\\_VERSION](#)  
     ([ironic.conductor.manager.ConductorManager](#) attribute), 2528  
[RPC\\_API\\_VERSION](#)  
     ([ironic.conductor.rpcapi.ConductorAPI](#) attribute), 2562  
[RPCHook](#) (class in [ironic.api.hooks](#)), 2221  
[RPCService](#) (class in [ironic.common.rpc\\_service](#)), 2448  
  
**S**  
[SAFE](#) (in module [ironic.common.boot\\_devices](#)), 2261  
[safe\\_rstrip\(\)](#) (in module [ironic.common.utils](#)), 2496  
[sanitize\\_dict\(\)](#) (in module [ironic.api.controllers.v1.utils](#)), 2163  
[save\(\)](#) ([ironic.objects.allocation.Allocation](#) method), 4231  
[save\(\)](#) ([ironic.objects.bios.BIOSSetting](#) method), 4253  
[save\(\)](#) ([ironic.objects.bios.BIOSSettingList](#) class method), 4259  
[save\(\)](#) ([ironic.objects.chassis.Chassis](#) method), 4270  
[save\(\)](#) ([ironic.objects.conductor.Conductor](#) method), 4280  
[save\(\)](#) ([ironic.objects.deploy\\_template.DeployTemplate](#) method), 4293  
[save\(\)](#) ([ironic.objects.node.Node](#) method), 4339  
[save\(\)](#) ([ironic.objects.port.Port](#) method), 4384  
[save\(\)](#) ([ironic.objects.portgroup.Portgroup](#) method), 4400  
[save\(\)](#) ([ironic.objects.volume\\_connector.VolumeConnector](#) method), 4426  
[save\(\)](#) ([ironic.objects.volume\\_target.VolumeTarget](#) method), 4445  
[SCHEMA](#) ([ironic.objects.allocation.AllocationCRUDPayload](#) attribute), 4234  
[SCHEMA](#) ([ironic.objects.chassis.ChassisCRUDPayload](#) attribute), 4272  
[SCHEMA](#) ([ironic.objects.deploy\\_template.DeployTemplateCRUDPayload](#) attribute), 4295  
[SCHEMA](#) ([ironic.objects.node.NodeCRUDPayload](#) attribute), 4342  
[SCHEMA](#) ([ironic.objects.node.NodePayload](#) attribute), 4352  
[SCHEMA](#) ([ironic.objects.node.NodeSetProvisionStatePayload](#) attribute), 4359  
[SCHEMA](#) ([ironic.objects.notification.NotificationPayloadBase](#) attribute), 4365  
[SCHEMA](#) ([ironic.objects.port.PortCRUDPayload](#) attribute), 4387  
[SCHEMA](#) ([ironic.objects.portgroup.PortgroupCRUDPayload](#) attribute), 4402  
[SCHEMA](#) ([ironic.objects.volume\\_connector.VolumeConnectorCRUDPayload](#) attribute), 4429  
[SCHEMA](#) ([ironic.objects.volume\\_target.VolumeTargetCRUDPayload](#) attribute), 4448  
[schema\(\)](#) (in module [ironic.common.args](#)), 2252  
[second\\_method\(\)](#)  
     ([ironic.drivers.modules.fake.FakeVendorB](#) method), 3840  
[send\\_raw\(\)](#) (in module [ironic.drivers.modules.ipmitool](#)), 3910

[send\\_raw\(\)](#) ([ironic.drivers.modules.ipmitool.VendorPassthru](#) method), [3322](#)  
[method](#)), [3905](#) [set\\_boot\\_device\(\)](#)  
[serialize\\_context\(\)](#) ([ironic.drivers.modules.ipmitool.IPMIManagement](#)  
[\(ironic.common.rpc.RequestContextSerializer](#) method), [3883](#)  
[method](#)), [2446](#) [set\\_boot\\_device\(\)](#)  
[serialize\\_entity\(\)](#) ([ironic.drivers.modules.irmc.management.IRMCManagement](#)  
[\(ironic.common.rpc.RequestContextSerializer](#) method), [3398](#)  
[method](#)), [2446](#) [set\\_boot\\_device\(\)](#)  
[serialize\\_entity\(\)](#) ([ironic.drivers.modules.noop\\_mgmt.NoopManagement](#)  
[\(ironic.objects.base.IronicObjectSerializer](#) method), [3980](#)  
[method](#)), [4246](#) [set\\_boot\\_device\(\)](#)  
[service\(\)](#) ([ironic.objects.notification.NotificationPublisher](#) property), [4367](#) ([ironic.drivers.modules.redfish.management.RedfishManagement](#)  
[method](#)), [3519](#)  
[ServiceUnavailable](#), [2319](#) [set\\_boot\\_device\(\)](#)  
[SessionCache](#) (class in [ironic.drivers.modules.xclarity.management.XClarityManagement](#)  
[ironic.drivers.modules.redfish.utils](#)), [method](#)), [3572](#)  
[3539](#) [set\\_boot\\_mode\(\)](#) (in module  
[set\(\)](#) ([ironic.drivers.modules.snmp.SNMPClient](#) [ironic.drivers.modules.ilo.common](#)),  
[method](#)), [4003](#) [3271](#)  
[set\\_arg\\_types\(\)](#) [set\\_boot\\_mode\(\)](#)  
[\(ironic.api.functions.FunctionDefinition](#) [\(ironic.drivers.base.ManagementInterface](#)  
[method](#)), [2211](#) [method](#)), [4108](#)  
[set\\_async\\_step\\_flags\(\)](#) (in module [set\\_boot\\_mode\(\)](#)  
[ironic.drivers.modules.deploy\\_utils](#)), [\(ironic.drivers.modules.ibmcm.management.IBMCMManagement](#)  
[3738](#) [method](#)), [3177](#)  
[set\\_bios\\_config\(\)](#) [set\\_boot\\_mode\(\)](#)  
[\(ironic.drivers.modules.drac.vendor\\_passthru.DracVendorPassthru](#) [WSMnVendorPassthru](#)  
[method](#)), [3161](#) [method](#)), [3325](#)  
[set\\_boot\\_device\(\)](#) (in module [set\\_boot\\_mode\(\)](#)  
[ironic.drivers.modules.drac.management](#)), [\(ironic.drivers.modules.redfish.management.RedfishManagement](#)  
[3114](#) [method](#)), [3522](#)  
[set\\_boot\\_device\(\)](#) [set\\_config\(\)](#) (in module  
[\(ironic.conductor.manager.ConductorManager](#) [ironic.drivers.modules.drac.bios](#)), [3083](#)  
[method](#)), [2540](#) [set\\_console\\_mode\(\)](#)  
[set\\_boot\\_device\(\)](#) ([ironic.conductor.manager.ConductorManager](#)  
[\(ironic.conductor.rpcapi.ConductorAPI](#) method), [2540](#)  
[method](#)), [2643](#) [set\\_console\\_mode\(\)](#)  
[set\\_boot\\_device\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#)  
[\(ironic.drivers.base.ManagementInterface](#) method), [2646](#)  
[method](#)), [4106](#) [set\\_defaults\(\)](#) (in module  
[set\\_boot\\_device\(\)](#) ([ironic.drivers.modules.drac.management.DracWSMnManagement](#)), [\(in module](#)  
[method](#)), [3111](#) [ironic.common.rpc](#)), [2448](#)  
[3740](#)  
[set\\_boot\\_device\(\)](#) ([ironic.drivers.modules.deploy\\_utils](#)),  
[\(ironic.drivers.modules.fake.FakeManagement](#) [set\\_indicator\\_state\(\)](#)  
[method](#)), [3809](#) [\(ironic.conductor.manager.ConductorManager](#)  
[method](#)), [2540](#)  
[set\\_boot\\_device\(\)](#) ([ironic.drivers.modules.ibmcm.management.IBMCMManagement](#)  
[method](#)), [3174](#) [set\\_indicator\\_state\(\)](#)  
[\(ironic.conductor.rpcapi.ConductorAPI](#)  
[method](#)), [2649](#)  
[set\\_boot\\_device\(\)](#) ([ironic.drivers.modules.ilo.management.IloManagement](#) [set\\_indicator\\_state\(\)](#)  
[method](#)), [3111](#)

(*ironic.drivers.base.ManagementInterface* method), 3408  
 (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3524  
 (*ironic.drivers.modules.ilo.management.IloManagement* method), 3326  
 (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 3577  
 (*ironic.conductor.steps*), 2679  
 (*ironic.conductor.steps*), 2680  
 (*ironic.common.utils*), 2497  
 (*ironic.db.api.Connection* method), 3004  
 (*ironic.db.sqlalchemy.api.Connection* method), 2855  
 (*ironic.db.api.Connection* method), 3005  
 (*ironic.db.sqlalchemy.api.Connection* method), 2857  
 (*ironic.api.functions.FunctionDefinition* method), 2211  
 (*ironic.drivers.base.PowerInterface* method), 4140  
 (*ironic.drivers.modules.agent\_power.AgentPower* method), 3689  
 (*ironic.drivers.modules.drac.power.DracWSManPower* method), 3121  
 (*ironic.drivers.modules.fake.FakePower* method), 3817  
 (*ironic.drivers.modules.ibmcpower.IBMCPower* method), 3186  
 (*ironic.drivers.modules.ilo.power.IloPower* method), 3335  
 (*ironic.drivers.modules.ipmitool.IPMIPower* method), 3892  
 (*ironic.drivers.modules.irmc.power.IRMCPower* method), 3608  
 (*ironic.drivers.modules.redfish.power.RedfishPower* method), 3536  
 (*ironic.drivers.modules.snmp.SNMPPower* method), 4022  
 (*ironic.drivers.modules.xclarity.power.XClarityPower* method), 3577  
 (*ironic.drivers.modules.drac.raid*), 3149  
 (*ironic.drivers.modules.ilo.common*), 3272  
 (*ironic.drivers.modules.irmc.common*), 3382  
 (*ironic.drivers.base.ManagementInterface* method), 4113  
 (*ironic.drivers.modules.redfish.management.RedfishManagement* method), 3526  
 (*ironic.conductor.task\_manager.TaskManager* method), 2697  
 (*ironic.conductor.manager.ConductorManager* method), 2540  
 (*ironic.conductor.rpcapi.ConductorAPI* method), 2651  
 (*ironic.common.profiler*), 2408  
 (*ironic.api.app*), 2208  
 (*ironic.drivers.modules.ilo.common*), 3274  
 (*ironic.drivers.modules.ilo.common*), 3276  
 (*ironic.drivers.modules.ilo.common*), 3279  
 (*ironic.drivers.modules.agent.AgentDeploy* method), 3586  
 (*ironic.drivers.modules.agent\_base.AgentBaseMixin* method), 3608

<code>should_write_image()</code> ( <i>ironic.drivers.base.StorageInterface</i> method), 4161	<code>SNMPDriverTeltronix</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4017
<code>should_write_image()</code> ( <i>ironic.drivers.modules.fake.FakeStorage</i> method), 3834	<code>SNMPFailure</code> , 2319
<code>should_write_image()</code> ( <i>ironic.drivers.modules.storage.cinder.CinderStorage</i> method), 3551	<code>SNMPHardware</code> (class in <i>ironic.drivers.snmp</i> ), 4206
<code>should_write_image()</code> ( <i>ironic.drivers.modules.storage.cinder.CinderStorage</i> method), 3551	<code>SNMPPower</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4018
<code>should_write_image()</code> ( <i>ironic.drivers.modules.storage.external.ExternalStorage</i> method), 3555	<code>SNMP_POWER_OFF</code> (in module <i>ironic.common.states</i> ), 2463
<code>should_write_image()</code> ( <i>ironic.drivers.modules.storage.noop.NoopStorage</i> method), 3560	<code>SOFT_REBOOT</code> (in module <i>ironic.common.states</i> ), 2464
<code>show()</code> ( <i>ironic.common.glance_service.image_service.GlanceImageService</i> method), 2230	<code>spawn_after()</code> ( <i>ironic.conductor.task_manager.TaskManager</i> method), 2698
<code>show()</code> ( <i>ironic.common.image_service.BaseImageService</i> method), 2332	<code>spawn_cleaning_error_handler()</code> (in <i>ironic.conductor.utils</i> ), 2740
<code>show()</code> ( <i>ironic.common.image_service.FileImageService</i> method), 2335	<code>spawn_deploying_error_handler()</code> (in module <i>ironic.conductor.utils</i> ), 2740
<code>show()</code> ( <i>ironic.common.image_service.HttpImageService</i> method), 2339	<code>spawn_rescue_error_handler()</code> (in module <i>ironic.conductor.utils</i> ), 2740
<code>sig</code> (in module <i>ironic.api.functions</i> ), 2212	<code>STABLE_STATES</code> (in module <i>ironic.common.states</i> ), 2464
<code>signature</code> (class in <i>ironic.api.functions</i> ), 2212	<code>stamp</code> ironic-dbsync command line option, 1699
<code>simple_headers</code> ( <i>ironic.api.app.IronicCORS</i> attribute), 2208	<code>stamp</code> command line option --help, 1701
<code>skip_automated_cleaning()</code> (in module <i>ironic.conductor.utils</i> ), 2739	--revision <REVISION>, 1701
<code>SNMPClient</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4000	-h, 1701
<code>SNMPDriverAPCMasterSwitch</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4004	<code>stamp()</code> (in module <i>ironic.db.migration</i> ), 3031
<code>SNMPDriverAPCMasterSwitchPlus</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4005	<code>stamp()</code> (in module <i>ironic.db.sqlalchemy.migration</i> ), 2888
<code>SNMPDriverAPCRackPDU</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4006	<code>stamp()</code> ( <i>ironic.cmd.dbsync.DBCommand</i> method), 2226
<code>SNMPDriverAten</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4007	<code>standalone_ports_supported</code> ( <i>ironic.db.sqlalchemy.models.Portgroup</i> attribute), 2902
<code>SNMPDriverAuto</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4008	<code>standalone_ports_supported()</code> ( <i>ironic.objects.portgroup.Portgroup</i> property), 4401
<code>SNMPDriverBase</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4008	<code>standalone_ports_supported()</code> ( <i>ironic.objects.portgroup.PortgroupCRUDPayload</i> property), 4403
<code>SNMPDriverBaytechMRP27</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4011	<code>START</code> ( <i>ironic.objects.fields.NotificationStatus</i> at- tribute), 4309
<code>SNMPDriverCyberPower</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4012	<code>start()</code> ( <i>ironic.common.json_rpc.server.WSGIService</i> method), 2241
<code>SNMPDriverEatonPower</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4013	<code>start()</code> ( <i>ironic.common.rpc_service.RPCService</i> method), 2448
<code>SNMPDriverSimple</code> (class in <i>ironic.drivers.modules.snmp</i> ), 4015	<code>start()</code> ( <i>ironic.common.wsgi_service.WSGIService</i> method), 2504



[start\\_console\(\)](#) ([ironic.drivers.base.ConsoleInterface](#) [method](#)), 4060  
[start\\_console\(\)](#) ([ironic.drivers.modules.fake.FakeConsole](#) [method](#)), 3771  
[start\\_console\(\)](#) ([ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole](#) [method](#)), 3896  
[start\\_console\(\)](#) ([ironic.drivers.modules.ipmitool.IPMISocatConsole](#) [method](#)), 3900  
[start\\_console\(\)](#) ([ironic.drivers.modules.noop.NoConsole](#) [method](#)), 3955  
[start\\_deploy\(\)](#) (in module [ironic.conductor.deployments](#)), 2523  
[start\\_iscsi\\_target\(\)](#) ([ironic.drivers.modules.agent\\_client.AgentClient](#) [method](#)), 3679  
[start\\_shellinabox\\_console\(\)](#) (in module [ironic.drivers.modules.console\\_utils](#)), 3704  
[start\\_socat\\_console\(\)](#) (in module [ironic.drivers.modules.console\\_utils](#)), 3705  
[state](#) ([ironic.db.sqlalchemy.models.Allocation](#) [attribute](#)), 2891  
[state\(\)](#) ([ironic.objects.allocation.Allocation](#) [property](#)), 4233  
[state\(\)](#) ([ironic.objects.allocation.AllocationCRUDPayload](#) [method](#)), 3956  
[state\(\)](#) ([ironic.objects.allocation.AllocationCRUDPayload](#) [property](#)), 4234  
[state\(\)](#) ([ironic.objects.deployment.Deployment](#) [property](#)), 4304  
[states](#) ([ironic.api.controllers.v1.node.NodesController](#) [attribute](#)), 2047  
[status\(\)](#) ([ironic.objects.notification.EventType](#) [property](#)), 4363  
[status\\_code](#) ([ironic.api.controllers.v1.utils.PassthruResponse](#) [attribute](#)), 2082  
[status\\_code](#) ([ironic.api.functions.FunctionDefinition](#) [attribute](#)), 2211  
[status\\_off](#) ([ironic.drivers.modules.snmp.SNMPDriverEatonPower](#) [attribute](#)), 4014  
[status\\_on](#) ([ironic.drivers.modules.snmp.SNMPDriverEatonPower](#) [attribute](#)), 4014  
[status\\_pending\\_off](#) ([ironic.drivers.modules.snmp.SNMPDriverEatonPower](#) [attribute](#)), 4015  
[status\\_pending\\_on](#) ([ironic.drivers.modules.snmp.SNMPDriverEatonPower](#) [attribute](#)), 4015  
[step](#) ([ironic.db.sqlalchemy.models.DeployTemplateStep](#) [attribute](#)), 2895  
[step\\_sanitize\(\)](#) (in module [ironic.api.controllers.v1.deploy\\_template](#)), 1985  
[steps](#) ([ironic.objects.deploy\\_template.DeployTemplate](#) [property](#)), 4294  
[steps](#) ([ironic.objects.deploy\\_template.DeployTemplateCRUDPayload](#) [property](#)), 4296  
[stop\\_console](#) ([ironic.common.json\\_rpc.server.WSGIService](#) [method](#)), 2241  
[stop\(\)](#) ([ironic.common.rpc\\_service.RPCService](#) [method](#)), 2448  
[stop\(\)](#) ([ironic.common.wsgi\\_service.WSGIService](#) [method](#)), 2504  
[stop\\_console\(\)](#) ([ironic.drivers.base.ConsoleInterface](#) [method](#)), 4061  
[stop\\_console\(\)](#) ([ironic.drivers.modules.fake.FakeConsole](#) [method](#)), 3772  
[stop\\_console\(\)](#) ([ironic.drivers.modules.ipmitool.IPMIShellinaboxConsole](#) [method](#)), 3898  
[stop\\_console\(\)](#) ([ironic.drivers.modules.ipmitool.IPMISocatConsole](#) [method](#)), 3902  
[stop\\_console\(\)](#) ([ironic.drivers.modules.noop.NoConsole](#) [method](#)), 3956  
[stop\\_shellinabox\\_console\(\)](#) (in module [ironic.drivers.modules.console\\_utils](#)), 3707  
[stop\\_socat\\_console\(\)](#) (in module [ironic.drivers.modules.console\\_utils](#)), 3708  
[storage](#) ([ironic.drivers.base.BareDriver](#) [attribute](#)), 4039  
[storage\\_interface](#) ([ironic.db.sqlalchemy.models.Node](#) [attribute](#)), 2899  
[storage\\_interface](#) ([ironic.objects.node.Node](#) [property](#)), 4349  
[storage\\_interface\(\)](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) [property](#)), 4350  
[storage\\_interface\(\)](#) ([ironic.objects.node.NodeCRUDPayload](#) [property](#)), 4345

[storage\\_interface\(\)](#) (*ironic.objects.node.NodePayload* property), 4354  
[storage\\_interface\(\)](#) (*ironic.objects.node.NodeSetPowerStatePayload* property), 4357  
[storage\\_interface\(\)](#) (*ironic.objects.node.NodeSetProvisionStatePayload* property), 4361  
[StorageError](#), 2319  
[StorageInterface](#) (class in *ironic.drivers.base*), 4159  
[StorageInterfaceFactory](#) (class in *ironic.common.driver\_factory*), 2284  
[store\\_agent\\_certificate\(\)](#) (in module *ironic.conductor.utils*), 2741  
[store\\_ramdisk\\_logs\(\)](#) (in module *ironic.drivers.utils*), 4218  
[string](#) (*ironic.api.controllers.base.Version* attribute), 2200  
[string\(\)](#) (in module *ironic.common.args*), 2252  
[string\\_list\(\)](#) (in module *ironic.common.args*), 2253  
[StringAcceptsCallable](#) (class in *ironic.objects.fields*), 4310  
[StringField](#) (class in *ironic.objects.fields*), 4311  
[StringFieldThatAcceptsCallable](#) (class in *ironic.objects.fields*), 4311  
[STUCK\\_STATES\\_TREATED\\_AS\\_FAIL](#) (in module *ironic.common.states*), 2464  
[SUCCESS](#) (*ironic.objects.fields.NotificationStatus* attribute), 4309  
[supported](#) (*ironic.drivers.base.BaseInterface* attribute), 4049  
[supported](#) (*ironic.drivers.hardware\_type.AbstractHardwareType* attribute), 4192  
[supported](#) (*ironic.drivers.modules.iscsi\_deploy.ISCSIDeploy* attribute), 3920  
[supported\(\)](#) (*ironic.api.controllers.v1.node.BootDeviceController* method), 1999  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.drac.IDRACHardware* property), 4180  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.fake\_hardware.FakeHardware* property), 4183  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.hardware\_type.AbstractHardwareType* property), 4192  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.ilo.IloHardware* property), 4198  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.irmc.IRMCHardware* property), 4203  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.redfish.RedfishHardware* property), 4205  
[supported\\_bios\\_interfaces\(\)](#) (*ironic.drivers.drac.IDRACHardware* property), 4181  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.fake\_hardware.FakeHardware* property), 4183  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.generic.GenericHardware* property), 4187  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.hardware\_type.AbstractHardwareType* property), 4192  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.ilo.Ilo5Hardware* property), 4197  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.ilo.IloHardware* property), 4198  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.irmc.IRMCHardware* property), 4204  
[supported\\_boot\\_interfaces\(\)](#) (*ironic.drivers.redfish.RedfishHardware* property), 4205  
[SUPPORTED\\_BOOT\\_MODE\\_LEGACY\\_BIOS\\_AND\\_UEFI](#) (in module *ironic.drivers.modules.ilo.common*), 3251  
[SUPPORTED\\_BOOT\\_MODE\\_LEGACY\\_BIOS\\_ONLY](#) (in module *ironic.drivers.modules.ilo.common*), 3252  
[SUPPORTED\\_BOOT\\_MODE\\_UEFI\\_ONLY](#) (in module *ironic.drivers.modules.ilo.common*), 3252  
[supported\\_console\\_interfaces\(\)](#) (*ironic.drivers.fake\_hardware.FakeHardware* property), 4184  
[supported\\_console\\_interfaces\(\)](#) (*ironic.drivers.hardware\_type.AbstractHardwareType* property), 4193  
[supported\\_console\\_interfaces\(\)](#) (*ironic.drivers.ilo.IloHardware* property), 4198

<i>(ironic.drivers.ilo.IloHardware property)</i> , 4198	<i>(ironic.drivers.ibmcm.IBMCHardware property)</i> , 4195
<code>supported_console_interfaces()</code> <i>(ironic.drivers.ipmi.IPMIHardware property)</i> , 4201	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ilo.Ilo5Hardware property)</i> , 4197
<code>supported_console_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware property)</i> , 4204	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ilo.IloHardware property)</i> , 4199
<code>supported_deploy_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4184	<code>supported_management_interfaces()</code> <i>(ironic.drivers.intel_ipmi.IntelIPMIHardware property)</i> , 4200
<code>supported_deploy_interfaces()</code> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4187	<code>supported_management_interfaces()</code> <i>(ironic.drivers.ipmi.IPMIHardware property)</i> , 4202
<code>supported_deploy_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4193	<code>supported_management_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware property)</i> , 4204
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.drac.IDRACHardware property)</i> , 4181	<code>supported_management_interfaces()</code> <i>(ironic.drivers.redfish.RedfishHardware property)</i> , 4206
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4184	<code>supported_management_interfaces()</code> <i>(ironic.drivers.snmp.SNMPHardware property)</i> , 4207
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4187	<code>supported_management_interfaces()</code> <i>(ironic.drivers.xclarity.XClarityHardware property)</i> , 4221
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4193	<code>supported_network_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4185
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.ilo.IloHardware property)</i> , 4199	<code>supported_network_interfaces()</code> <i>(ironic.drivers.generic.GenericHardware property)</i> , 4187
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.irmc.IRMCHardware property)</i> , 4204	<code>supported_network_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4193
<code>supported_inspect_interfaces()</code> <i>(ironic.drivers.redfish.RedfishHardware property)</i> , 4205	<code>supported_power_interfaces()</code> <i>(ironic.drivers.drac.IDRACHardware property)</i> , 4181
<code>supported_management_interfaces()</code> <i>(ironic.drivers.drac.IDRACHardware property)</i> , 4181	<code>supported_power_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4185
<code>supported_management_interfaces()</code> <i>(ironic.drivers.fake_hardware.FakeHardware property)</i> , 4184	<code>supported_power_interfaces()</code> <i>(ironic.drivers.generic.ManualManagementHardware property)</i> , 4189
<code>supported_management_interfaces()</code> <i>(ironic.drivers.generic.ManualManagementHardware property)</i> , 4189	<code>supported_power_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4194
<code>supported_management_interfaces()</code> <i>(ironic.drivers.hardware_type.AbstractHardwareType property)</i> , 4193	<code>supported_power_interfaces()</code> <i>(ironic.drivers.ibmcm.IBMCHardware property)</i> , 4195
<code>supported_management_interfaces()</code>	<code>supported_power_interfaces()</code>



<code>(ironic.drivers.ilo.IloHardware property), 4199</code>	<code>(ironic.drivers.generic.GenericHardware property), 4188</code>
<code>supported_power_interfaces()</code> <code>(ironic.drivers.ipmi.IPMIHardware property), 4202</code>	<code>supported_storage_interfaces()</code> <code>(ironic.drivers.hardware_type.AbstractHardwareType property), 4194</code>
<code>supported_power_interfaces()</code> <code>(ironic.drivers.irmc.IRMCHardware property), 4204</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.drac.IDRACHardware property), 4182</code>
<code>supported_power_interfaces()</code> <code>(ironic.drivers.redfish.RedfishHardware property), 4206</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.fake_hardware.FakeHardware property), 4186</code>
<code>supported_power_interfaces()</code> <code>(ironic.drivers.snmp.SNMPHardware property), 4207</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.generic.ManualManagementHardware property), 4189</code>
<code>supported_power_interfaces()</code> <code>(ironic.drivers.xclarity.XClarityHardware property), 4221</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.hardware_type.AbstractHardwareType property), 4194</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.drac.IDRACHardware property), 4181</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.ibm.ibmchardware property), 4196</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.fake_hardware.FakeHardware property), 4185</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.ilo.IloHardware property), 4199</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.generic.GenericHardware property), 4188</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.ipmi.IPMIHardware property), 4202</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.hardware_type.AbstractHardwareType property), 4194</code>	<code>supported_vendor_interfaces()</code> <code>(ironic.drivers.redfish.RedfishHardware property), 4206</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.ibm.ibmchardware property), 4196</code>	<code>supports_is_smartnic()</code> <code>(ironic.objects.port.Port class method), 4385</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.ilo.Ilo5Hardware property), 4197</code>	<code>supports_physical_network()</code> <code>(ironic.objects.port.Port class method), 4385</code>
<code>supported_raid_interfaces()</code> <code>(ironic.drivers.irmc.IRMCHardware property), 4205</code>	<code>supports_power_sync()</code> <code>(ironic.drivers.base.PowerInterface method), 4142</code>
<code>supported_rescue_interfaces()</code> <code>(ironic.drivers.fake_hardware.FakeHardware property), 4185</code>	<code>supports_power_sync()</code> <code>(ironic.drivers.modules.agent_power.AgentPower method), 3690</code>
<code>supported_rescue_interfaces()</code> <code>(ironic.drivers.generic.GenericHardware property), 4188</code>	<code>supports_version()</code> <code>(ironic.objects.base.IronicObject class method), 4243</code>
<code>supported_rescue_interfaces()</code> <code>(ironic.drivers.hardware_type.AbstractHardwareType property), 4194</code>	<code>swap_mib()</code> <code>(ironic.objects.deployment.Deployment property), 4304</code>
<code>supported_storage_interfaces()</code> <code>(ironic.drivers.fake_hardware.FakeHardware property), 4185</code>	<code>swift_temp_url()</code> <code>(ironic.common.glance_service.image_service.GlanceImage service method), 2231</code>
<code>supported_storage_interfaces()</code>	<code>SwiftAPI (class in ironic.common.swift), 2468</code>
	<code>SwiftObjectNotFoundError, 2319</code>

SwiftOperationError, 2320  
switch\_pxe\_config() (in module ironic.drivers.modules.deploy\_utils), target (ironic.conductor.manager.ConductorManager attribute), 2540  
3741  
switch\_to\_tenant\_network() target\_power\_state (ironic.drivers.modules.agent\_base.AgentOobStepsMixin.ironic.db.sqlalchemy.models.Node method), 3627  
sync() (ironic.drivers.modules.agent\_client.AgentClient.target\_power\_state() method), 3682 (ironic.objects.node.Node property), 4341  
sync\_boot\_mode() (in module ironic.drivers.modules.boot\_mode\_utils), target\_power\_state() 3698 (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350  
sync\_node\_setting() (ironic.objects.bios.BIOSSettingList target\_power\_state() class method), 4260 (ironic.objects.node.NodeCRUDPayload attribute), 4345  
SYS\_OBJ\_OID (ironic.drivers.modules.snmp.SNMPDriverAttribute), 4008 target\_power\_state() (ironic.objects.node.NodePayload property), 4354  
SYSTEM (in module ironic.common.components), 2277  
system\_id(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch attribute), 4005 (ironic.objects.node.NodeSetPowerStatePayload attribute), 4341  
system\_id(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitchPlus attribute), 4006 target\_power\_state() (ironic.objects.node.NodeSetProvisionStatePayload property), 4361  
system\_id(ironic.drivers.modules.snmp.SNMPDriverAPCRackPD attribute), 4007  
system\_id(ironic.drivers.modules.snmp.SNMPDriverArista attribute), 4008 target\_provision\_state (ironic.db.sqlalchemy.models.Node attribute), 2899  
system\_id(ironic.drivers.modules.snmp.SNMPDriverCyberPower attribute), 4013 target\_provision\_state() (ironic.objects.node.Node property), 4341  
system\_id(ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute), 4015  
system\_id(ironic.drivers.modules.snmp.SNMPDriverEatonTeltronix attribute), 4017 target\_provision\_state() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350  
T target\_provision\_state() (ironic.objects.node.NodeCRUDPayload property), 4345  
table\_args() (in module ironic.db.sqlalchemy.models), 2905  
tag (ironic.db.sqlalchemy.models.NodeTag attribute), 2900 (ironic.objects.node.NodePayload property), 4354  
take\_over() (ironic.drivers.base.DeployInterface method), 4070 target\_provision\_state() (ironic.objects.node.NodeSetPowerStatePayload property), 4357  
take\_over() (ironic.drivers.modules.agent\_base.AgentBaseMixin.ironic.objects.node.NodeSetPowerStatePayload method), 3609  
take\_over() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3058 (ironic.objects.node.NodeSetProvisionStatePayload property), 4361  
take\_over() (ironic.drivers.modules.fake.FakeDeploy method), 3780 target\_raid\_config (ironic.db.sqlalchemy.models.Node attribute), 2899  
take\_over\_allocation() (ironic.db.api.Connection method), 3007 target\_raid\_config() (ironic.objects.node.Node property),

4341

target\_state() (*ironic.common.fsm.FSM* *method*), 4363

property), 2330 (*ironic.common.context.RequestContext*

TaskManager (class in *method*), 2278

*ironic.conductor.task\_manager*), 2692 to\_power() (*ironic.objects.node.NodeSetPowerStatePayload*

tear\_down() (*ironic.drivers.base.DeployInterface* *property*), 4357

*method*), 4072 touch() (*ironic.objects.conductor.Conductor*

tear\_down() (*ironic.drivers.modules.agent\_base.AgentBase* *method*), 4280

*method*), 3609 touch\_conductor()

tear\_down() (*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3009

*method*), 3060 touch\_conductor()

tear\_down() (*ironic.drivers.modules.fake.FakeDeploy* *method*), 3782

*method*), 3609 (*ironic.db.sqlalchemy.api.Connection* *method*),

tear\_down\_agent() *method*), 2860

(*ironic.drivers.modules.agent\_base.AgentBase* *method*), 3626

tear\_down\_agent() *method*), 3010

(*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3060

tear\_down\_cleaning() *method*), 2861

(*ironic.drivers.base.DeployInterface* *method*), 4073

tear\_down\_cleaning() *method*), 4341

(*ironic.drivers.modules.agent\_base.AgentBase* *method*), 3612

tear\_down\_cleaning() Trait (class in *ironic.objects.trait*), 4403

(*ironic.drivers.modules.ansible.deploy.AnsibleDeploy* *method*), 3061

tear\_down\_inband\_cleaning() (in module *ironic.drivers.modules.deploy\_utils*), 3744

tear\_down\_storage\_configuration() (in module *ironic.drivers.modules.deploy\_utils*), 3746

tempdir() (in module *ironic.common.utils*), 2498

template\_sanitize() (in module *ironic.api.controllers.v1.deploy\_template*), 1985

TemporaryFailure, 2320

TempUrlCacheElement (class in *ironic.common.glance\_service.image\_service*), 2234

TFTPImageCache (class in *ironic.common.pxe\_utils*), 2410

third\_method\_sync() (*ironic.drivers.modules.fake.FakeVendorB* *method*), 3840

to\_event\_type\_field() (*ironic.objects.notification.EventType*

*method*), 4363

to\_policy\_values() (*ironic.common.context.RequestContext* *method*), 2278

to\_power() (*ironic.objects.node.NodeSetPowerStatePayload* *property*), 4357

touch() (*ironic.objects.conductor.Conductor* *method*), 4280

touch\_conductor()

touch\_provisioning() (*ironic.db.sqlalchemy.api.Connection* *method*), 2860

touch\_provisioning() (*ironic.objects.node.Node* *method*), 4341

touch\_provisioning() (in module *ironic.common.profiler*), 2409

Trait (class in *ironic.objects.trait*), 4403

TraitList (class in *ironic.objects.trait*), 4408

traits (*ironic.db.sqlalchemy.models.Allocation* *attribute*), 2891

traits() (*ironic.objects.allocation.Allocation* *property*), 4233

traits() (*ironic.objects.allocation.AllocationCRUDPayload* *property*), 4235

traits() (*ironic.objects.node.Node* *property*), 4341

traits() (*ironic.objects.node.NodeCorrectedPowerStatePayload* *property*), 4350

traits() (*ironic.objects.node.NodeCRUDPayload* *property*), 4345

traits() (*ironic.objects.node.NodePayload* *property*), 4354

traits() (*ironic.objects.node.NodeSetPowerStatePayload* *property*), 4357

traits() (*ironic.objects.node.NodeSetProvisionStatePayload* *property*), 4361

translate\_from\_glance() (in module *ironic.common.glance\_service.service\_utils*),

2237  
translate\_xclarity\_power\_action() (in module *ironic.drivers.modules.xclarity.common*), 3567  
translate\_xclarity\_power\_state() (in module *ironic.drivers.modules.xclarity.common*), 3568  
try\_set\_boot\_device() (in module *ironic.drivers.modules.deploy\_utils*), 3747  
type (*ironic.db.sqlalchemy.models.VolumeConnector* attribute), 2903  
type () (*ironic.objects.volume\_connector.VolumeConnector* property), 4428  
type () (*ironic.objects.volume\_connector.VolumeConnector* property), 4429  
types () (in module *ironic.common.args*), 2255

**U**

UEFI (in module *ironic.common.boot\_modes*), 2262  
umount () (in module *ironic.common.utils*), 2498  
Unauthorized, 2320  
unbind\_neutron\_port () (in module *ironic.common.neutron*), 2392  
unconfigure\_tenant\_networks () (*ironic.drivers.base.NetworkInterface* method), 4129  
unconfigure\_tenant\_networks () (*ironic.drivers.modules.network.flat.FlatNetwork* method), 3443  
unconfigure\_tenant\_networks () (*ironic.drivers.modules.network.neutron.NeutronNetwork* method), 3452  
unconfigure\_tenant\_networks () (*ironic.drivers.modules.network.noop.NoopNetwork* method), 3461  
unit\_id (*ironic.drivers.modules.snmp.SNMPDriverBase* attribute), 4012  
unix\_file\_modification\_datetime () (in module *ironic.common.utils*), 2499  
UNKNOWN (in module *ironic.common.indicator\_states*), 2363  
UnknownArgument, 2320  
UnknownAttribute, 2320  
unpublish\_image () (*ironic.drivers.modules.image\_utils.ImageHandler* method), 3849  
unpublish\_image\_for\_node () (*ironic.drivers.modules.image\_utils.ImageHandler* class method), 3850  
unregister () (*ironic.objects.conductor.Conductor* method), 4280  
unregister\_all\_hardware\_interfaces () (*ironic.objects.conductor.Conductor* method), 4280  
unregister\_conductor () (*ironic.db.api.Connection* method), 3011  
unregister\_conductor () (*ironic.db.sqlalchemy.api.Connection* method), 2862  
unregister\_conductor\_hardware\_interfaces () (*ironic.db.api.Connection* method), 3011  
unregister\_conductor\_hardware\_interfaces () (*ironic.db.sqlalchemy.api.Connection* method), 2863  
unrescue () (*ironic.drivers.base.RescueInterface* method), 4158  
unrescue () (*ironic.drivers.modules.agent.AgentRescue* method), 3599  
unrescue () (*ironic.drivers.modules.fake.FakeRescue* method), 3828  
unrescue () (*ironic.drivers.modules.noop.NoRescue* method), 3966  
UNRESCUEFAIL (in module *ironic.common.states*), 2465  
UNRESCUING (in module *ironic.common.states*), 2465  
unset\_node\_tags () (*ironic.db.api.Connection* method), 3012  
unset\_node\_tags () (*ironic.db.sqlalchemy.api.Connection* method), 2863  
unset\_node\_traits () (*ironic.db.api.Connection* method), 3013  
unset\_node\_traits () (*ironic.db.sqlalchemy.api.Connection* method), 2864  
UNSTABLE\_STATES (in module *ironic.common.states*), 2465  
UnsupportedDriverExtension, 2321  
update\_allocation () (*ironic.db.api.Connection* method), 3013  
update\_allocation () (*ironic.db.sqlalchemy.api.Connection* method), 2864

UPDATE_ALLOWED_STATES (in module <i>ironic.common.states</i> ), 2465	update_next_step_index() (in module <i>ironic.conductor.utils</i> ), 2741
update_bios_setting_list() (in module <i>ironic.db.api.Connection</i> method), 3015	update_node() (in module <i>ironic.conductor.manager.ConductorManager</i> method), 2540
update_bios_setting_list() (in module <i>ironic.db.sqlalchemy.api.Connection</i> method), 2866	update_node() (in module <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2654
update_boot_mode() (in module <i>ironic.drivers.modules.ilo.common</i> ), 3283	update_node() (in module <i>ironic.db.api.Connection</i> method), 3020
update_chassis() (in module <i>ironic.db.api.Connection</i> method), 3017	update_node() (in module <i>ironic.db.sqlalchemy.api.Connection</i> method), 2871
update_chassis() (in module <i>ironic.db.sqlalchemy.api.Connection</i> method), 2869	update_object_meta() (in module <i>ironic.common.swift.SwiftAPI</i> method), 2474
update_deploy_template() (in module <i>ironic.db.api.Connection</i> method), 3018	update_opt_defaults() (in module <i>ironic.conf.opts</i> ), 2755
update_deploy_template() (in module <i>ironic.db.sqlalchemy.api.Connection</i> method), 2869	update_port() (in module <i>ironic.conductor.manager.ConductorManager</i> method), 2540
update_dhcp() (in module <i>ironic.common.dhcp_factory.DHCPFactory</i> method), 2280	update_port() (in module <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2657
update_dhcp_opts() (in module <i>ironic.dhcp.base.BaseDHCP</i> method), 3033	update_port() (in module <i>ironic.db.api.Connection</i> method), 3022
update_dhcp_opts() (in module <i>ironic.dhcp.neutron.NeutronDHCPApi</i> method), 3040	update_port() (in module <i>ironic.db.sqlalchemy.api.Connection</i> method), 2873
update_dhcp_opts() (in module <i>ironic.dhcp.none.NoneDHCPApi</i> method), 3047	update_port_address() (in module <i>ironic.common.neutron</i> ), 2395
update_firmware() (in module <i>ironic.drivers.modules.ilo.management.IloManagement</i> method), 3327	update_port_dhcp_opts() (in module <i>ironic.dhcp.base.BaseDHCP</i> method), 3036
update_firmware() (in module <i>ironic.drivers.modules.redfish.management.RedfishManagement</i> method), 3528	update_port_dhcp_opts() (in module <i>ironic.dhcp.neutron.NeutronDHCPApi</i> method), 3043
update_firmware_sum() (in module <i>ironic.drivers.modules.ilo.management.IloManagement</i> method), 3328	update_port_dhcp_opts() (in module <i>ironic.dhcp.none.NoneDHCPApi</i> method), 3050
update_ipmi_properties() (in module <i>ironic.drivers.modules.ilo.common</i> ), 3284	update_portgroup() (in module <i>ironic.conductor.manager.ConductorManager</i> method), 2540
update_ipmi_properties() (in module <i>ironic.drivers.modules.irmc.common</i> ), 3383	update_portgroup() (in module <i>ironic.conductor.rpcapi.ConductorAPI</i> method), 2658
update_neutron_port() (in module <i>ironic.common.neutron</i> ), 2394	update_portgroup() (in module <i>ironic.db.api.Connection</i> method), 3023
	update_portgroup() (in module <i>ironic.db.sqlalchemy.api.Connection</i> method), 2873



[method](#)), [2874](#)  
[update\\_raid\\_info\(\)](#) (in module [ironic.common.raid](#)), [2438](#)  
[update\\_secure\\_boot\\_mode\(\)](#) (in module [ironic.drivers.modules.ilo.common](#)), [3284](#)  
[update\\_state\\_in\\_older\\_versions\(\)](#) (in module [ironic.api.controllers.v1.node](#)), [2051](#)  
[update\\_to\\_latest\\_versions\(\)](#) ([ironic.db.api.Connection](#) method), [3024](#)  
[update\\_to\\_latest\\_versions\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), [2875](#)  
[update\\_volume\\_connector\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), [2540](#)  
[update\\_volume\\_connector\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), [2660](#)  
[update\\_volume\\_connector\(\)](#) ([ironic.db.api.Connection](#) method), [3026](#)  
[update\\_volume\\_connector\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), [2877](#)  
[update\\_volume\\_target\(\)](#) ([ironic.conductor.manager.ConductorManager](#) method), [2540](#)  
[update\\_volume\\_target\(\)](#) ([ironic.conductor.rpcapi.ConductorAPI](#) method), [2663](#)  
[update\\_volume\\_target\(\)](#) ([ironic.db.api.Connection](#) method), [3028](#)  
[update\\_volume\\_target\(\)](#) ([ironic.db.sqlalchemy.api.Connection](#) method), [2879](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.Allocation](#) attribute), [2891](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.BIOSSetting](#) attribute), [2891](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.Chassis](#) attribute), [2892](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.Conductor](#) attribute), [2893](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.ConductorHardwareProfile](#) attribute), [2894](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.DeployTemplate](#) attribute), [2895](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.DeployTemplateStep](#) attribute), [2896](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.Node](#) attribute), [2899](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.NodeTag](#) attribute), [2900](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.NodeTrait](#) attribute), [2900](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.Port](#) attribute), [2901](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.Portgroup](#) attribute), [2903](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.VolumeConnector](#) attribute), [2903](#)  
[updated\\_at](#) ([ironic.db.sqlalchemy.models.VolumeTarget](#) attribute), [2904](#)  
[updated\\_at](#) ([ironic.objects.allocation.Allocation](#) property), [4233](#)  
[updated\\_at](#) ([ironic.objects.allocation.AllocationCRUDNotification](#) property), [4234](#)  
[updated\\_at](#) ([ironic.objects.allocation.AllocationCRUDPayload](#) property), [4235](#)  
[updated\\_at](#) ([ironic.objects.bios.BIOSSetting](#) property), [4254](#)  
[updated\\_at](#) ([ironic.objects.bios.BIOSSettingList](#) property), [4262](#)  
[updated\\_at](#) ([ironic.objects.chassis.Chassis](#) property), [4271](#)  
[updated\\_at](#) ([ironic.objects.chassis.ChassisCRUDNotification](#) property), [4272](#)  
[updated\\_at](#) ([ironic.objects.chassis.ChassisCRUDPayload](#) property), [4272](#)  
[updated\\_at](#) ([ironic.objects.conductor.Conductor](#) property), [4281](#)  
[updated\\_at](#) ([ironic.objects.deploy\\_template.DeployTemplate](#) property), [4294](#)  
[updated\\_at](#) ([ironic.objects.deploy\\_template.DeployTemplate](#) property), [4295](#)  
[updated\\_at](#) ([ironic.objects.deploy\\_template.DeployTemplate](#) property), [4296](#)  
[updated\\_at](#) ([ironic.objects.deployment.Deployment](#) property), [4304](#)  
[updated\\_at](#) ([ironic.objects.node.Node](#) property), [4341](#)  
[updated\\_at](#) ([ironic.objects.node.NodeConsoleNotification](#) property), [4346](#)  
[updated\\_at](#) ([ironic.objects.node.NodeCorrectedPowerStateNotification](#) property), [4347](#)  
[updated\\_at](#) ([ironic.objects.node.NodeCorrectedPowerStatePayload](#) property), [4350](#)  
[updated\\_at](#) ([ironic.objects.node.NodeCRUDNotification](#) property), [4351](#)

property), 4342 upgrade  
updated\_at() (ironic.objects.node.NodeCRUDPayload, ironic-dbsync command line  
property), 4345 option, 1699  
updated\_at() (ironic.objects.node.NodeMaintenanceNotification, ironic-dbsync command line option  
property), 4351 --help, 1702  
updated\_at() (ironic.objects.node.NodePayload, --revision <REVISION>, 1702  
property), 4354 -h, 1702  
updated\_at() (ironic.objects.node.NodeSetPowerStateNotification, in module ironic.db.migration),  
property), 4355 3031  
updated\_at() (ironic.objects.node.NodeSetPowerStatePayload, (in module  
property), 4357 ironic.db.sqlalchemy.migration), 2889  
updated\_at() (ironic.objects.node.NodeSetProvisionStateNotification, ironic.cmd.dbsync.DBCommand  
property), 4358 method), 2226  
updated\_at() (ironic.objects.node.NodeSetProvisionStatePayload,   
property), 4361 (ironic.conductor.task\_manager.TaskManager  
updated\_at() (ironic.objects.notification.EventType, method), 2700  
property), 4363 url (ironic.common.glance\_service.image\_service.TempUrlCache  
updated\_at() (ironic.objects.notification.NotificationBase, attribute), 2234  
property), 4365 url\_expires\_at  
updated\_at() (ironic.objects.notification.NotificationPayload, in module ironic.common.glance\_service.image\_service.TempUrlCache  
property), 4367 attribute), 2234  
updated\_at() (ironic.objects.notification.NotificationRule, in module ironic.db.sqlalchemy.models.Allocation at-  
property), 4367 tribute), 2891  
updated\_at() (ironic.objects.port.Port, uuid (ironic.db.sqlalchemy.models.Chassis at-  
erty), 4386 tribute), 2892  
updated\_at() (ironic.objects.port.PortCRUDNotification, (in module ironic.db.sqlalchemy.models.DeployTemplate  
property), 4387 attribute), 2895  
updated\_at() (ironic.objects.port.PortCRUDPayload, uuid (ironic.db.sqlalchemy.models.Node at-  
property), 4387 tribute), 2899  
updated\_at() (ironic.objects.portgroup.Portgroup, uuid (ironic.db.sqlalchemy.models.Port at-  
property), 4401 tribute), 2901  
updated\_at() (ironic.objects.portgroup.PortgroupCRUDNotification, in module ironic.db.sqlalchemy.models.Portgroup at-  
property), 4402 tribute), 2903  
updated\_at() (ironic.objects.portgroup.PortgroupCRUDPayload, uuid (ironic.db.sqlalchemy.models.VolumeConnector  
property), 4403 attribute), 2903  
updated\_at() (ironic.objects.trait.Trait, uuid (ironic.db.sqlalchemy.models.VolumeTarget  
erty), 4408 attribute), 2904  
updated\_at() (ironic.objects.trait.TraitList, uuid() (in module ironic.common.args), 2256  
property), 4413 uuid() (ironic.objects.allocation.Allocation  
updated\_at() (ironic.objects.volume\_connector.VolumeConnector, property), 4233  
property), 4428 uuid() (ironic.objects.allocation.AllocationCRUDPayload  
updated\_at() (ironic.objects.volume\_connector.VolumeConnectorCRUDNotification,   
property), 4429 uuid() (ironic.objects.chassis.Chassis property),  
updated\_at() (ironic.objects.volume\_connector.VolumeConnectorCRUDPayload, 4271  
property), 4429 uuid() (ironic.objects.chassis.ChassisCRUDPayload  
updated\_at() (ironic.objects.volume\_target.VolumeTarget, property), 4272  
property), 4447 uuid() (ironic.objects.deploy\_template.DeployTemplate  
updated\_at() (ironic.objects.volume\_target.VolumeTargetCRUDNotification,   
property), 4448 uuid() (ironic.objects.deploy\_template.DeployTemplateCRUDPa  
updated\_at() (ironic.objects.volume\_target.VolumeTargetCRUDPayload, 4271  
property), 4448 uuid() (ironic.objects.deployment.Deployment

property), 4304

uuid() (ironic.objects.node.Node property), 4341

uuid() (ironic.objects.node.NodeCorrectedPowerStatePayload property), 4350

uuid() (ironic.objects.node.NodeCRUDPayload property), 4345

uuid() (ironic.objects.node.NodePayload property), 4354

uuid() (ironic.objects.node.NodeSetPowerStatePayload property), 4357

uuid() (ironic.objects.node.NodeSetProvisionStatePayload property), 4361

uuid() (ironic.objects.port.Port property), 4386

uuid() (ironic.objects.port.PortCRUDPayload property), 4388

uuid() (ironic.objects.portgroup.Portgroup property), 4401

uuid() (ironic.objects.portgroup.PortgroupCRUDPayload property), 4403

uuid() (ironic.objects.volume\_connector.VolumeConnector property), 4428

uuid() (ironic.objects.volume\_connector.VolumeConnectorCRUDPayload property), 4430

uuid() (ironic.objects.volume\_target.VolumeTarget property), 4447

uuid() (ironic.objects.volume\_target.VolumeTargetCRUDPayload property), 4448

uuid\_or\_name() (in module ironic.common.args), 2257

UUIDField (class in ironic.objects.fields), 4312

**V**

validate() (in module ironic.common.args), 2258

validate() (in module ironic.drivers.modules.iscsi\_deploy), 3943

validate() (ironic.api.controllers.v1.node.NodesController method), 2047

validate() (ironic.drivers.base.BaseInterface method), 4049

validate() (ironic.drivers.base.NetworkInterface method), 4129

validate() (ironic.drivers.base.RAIDInterface method), 4151

validate() (ironic.drivers.base.VendorInterface method), 4165

validate() (ironic.drivers.modules.agent.AgentDeploy method), 3586

validate() (ironic.drivers.modules.agent.AgentRescue method), 3601

validate() (ironic.drivers.modules.agent\_power.AgentPower method), 3692

validate() (ironic.drivers.modules.ansible.deploy.AnsibleDeploy method), 3062

validate() (ironic.drivers.modules.drac.bios.DracWSManBIOS method), 3069

validate() (ironic.drivers.modules.drac.inspect.DracWSManInspect method), 3097

validate() (ironic.drivers.modules.drac.management.DracWSManManagement method), 3113

validate() (ironic.drivers.modules.drac.power.DracWSManPower method), 3123

validate() (ironic.drivers.modules.drac.vendor\_passthru.DracWSManVendorPassthru method), 3162

validate() (ironic.drivers.modules.fake.FakeBIOS method), 3760

validate() (ironic.drivers.modules.fake.FakeBoot method), 3767

validate() (ironic.drivers.modules.fake.FakeConsole method), 3773

validate() (ironic.drivers.modules.fake.FakeDeploy method), 3783

validate() (ironic.drivers.modules.fake.FakeInspect method), 3788

validate() (ironic.drivers.modules.fake.FakeManagement method), 3811

validate() (ironic.drivers.modules.fake.FakePower method), 3819

validate() (ironic.drivers.modules.fake.FakeRescue method), 3829

validate() (ironic.drivers.modules.fake.FakeStorage method), 3835

validate() (ironic.drivers.modules.fake.FakeVendorA method), 3838

validate() (ironic.drivers.modules.fake.FakeVendorB method), 3841

validate() (ironic.drivers.modules.ibmcm.management.IBMCMManagement method), 3179

validate() (ironic.drivers.modules.ibmcm.power.IBMCPower method), 3188

validate() (ironic.drivers.modules.ibmcm.vendor.IBMCVendor method), 3204

validate() (ironic.drivers.modules.ilo.bios.IloBIOS method), 3210

validate() (ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot method), 3225

validate() (ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot method), 3237

validate() (ironic.drivers.modules.ilo.console.IloConsoleInterface method), 3290



[validate\(\) \(ironic.drivers.modules.ilo.inspect.IloInspect method\), 3538](#)  
[method\), 3299](#) [validate\(\) \(ironic.drivers.modules.redfish.vendor.RedfishVendorData method\), 3545](#)  
[validate\(\) \(ironic.drivers.modules.ilo.management.IloManagement method\), 3329](#) [validate\(\) \(ironic.drivers.modules.snmp.SNMPPower method\), 4025](#)  
[method\), 3337](#) [validate\(\) \(ironic.drivers.modules.storage.cinder.CinderStorage method\), 3551](#)  
[validate\(\) \(ironic.drivers.modules.ilo.vendor.VendorPassThrough method\), 3348](#) [validate\(\) \(ironic.drivers.modules.storage.external.ExternalStorage method\), 3556](#)  
[method\), 3870](#) [validate\(\) \(ironic.drivers.modules.storage.noop.NoopStorage method\), 3561](#)  
[validate\(\) \(ironic.drivers.modules.ipmitool.IPMIConsole method\), 3873](#) [validate\(\) \(ironic.drivers.modules.xclarity.management.XClarityManagement method\), 3574](#)  
[method\), 3885](#) [validate\(\) \(ironic.drivers.modules.xclarity.power.XClarityPower method\), 3578](#)  
[validate\(\) \(ironic.drivers.modules.ipmitool.IPMIPower method\), 3894](#) [validate\(\) \(ironic.drivers.utils.MixinVendorInterface method\), 4208](#)  
[method\), 3907](#) [validate\\_and\\_normalize\\_datapath\\_id\(\) \(in module ironic.common.utils\), 2499](#)  
[validate\(\) \(ironic.drivers.modules.irmc.bios.IRMCBIOS method\), 3356](#) [validate\\_and\\_normalize\\_mac\(\) \(in module ironic.common.utils\), 2500](#)  
[validate\(\) \(ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot method\), 3369](#) [validate\\_boot\\_parameters\\_for\\_trusted\\_boot\(\) \(in module ironic.common.pxe\\_utils\), 2434](#)  
[validate\(\) \(ironic.drivers.modules.irmc.inspect.IRMCInspect method\), 3385](#) [validate\\_driver\\_capabilities\(\) \(in module ironic.drivers.modules.deploy\\_utils\), 2749](#)  
[validate\(\) \(ironic.drivers.modules.irmc.management.IRMCManagement method\), 3400](#) [validate\\_conductor\\_group\(\) \(in module ironic.common.utils\), 2502](#)  
[validate\(\) \(ironic.drivers.modules.irmc.power.IRMCPower method\), 3410](#) [validate\\_configuration\(\) \(in module ironic.common.raid\), 2440](#)  
[method\), 3920](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.network.flat.FlatNetwork method\), 3444](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.network.neutron.NeutronNetwork method\), 3453](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3947](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.noop.FailMix method\), 3982](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3982](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.noop\\_mgmt.NoopManagement method\), 3982](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3982](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.pxe.PXERamdiskDeploy method\), 3988](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3988](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.pxe\\_base.PXEBaseMixin method\), 3996](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3996](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.redfish.bios.RedfishBIOS method\), 3472](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3472](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot method\), 3485](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3485](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.redfish.inspect.RedfishInspect method\), 3491](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3491](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.redfish.management.RedfishManagement method\), 3529](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3529](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[validate\(\) \(ironic.drivers.modules.redfish.power.RedfishPower method\), 3538](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)  
[method\), 3538](#) [validate\\_driver\\_interfaces\(\) \(in module ironic.common.network\), 2540](#)

3750	
validate_image_proxies() (in module ironic.drivers.modules.agent), 3605	validate_rescue() (ironic.drivers.base.BootInterface method), 4057
validate_inspection() (ironic.common.neutron.NeutronNetworkInterfaceMixin method), 2376	validate_rescue() (ironic.drivers.base.NetworkInterface method), 4132
validate_inspection() (ironic.drivers.base.BootInterface method), 4056	validate_rescue() (ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot method), 3227
validate_inspection() (ironic.drivers.base.NetworkInterface method), 4131	validate_rescue() (ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot method), 3239
validate_inspection() (ironic.drivers.modules.ilo.boot.IloUefiHttpsBoot method), 3226	validate_rescue() (ironic.drivers.modules.irmc.boot.IRMCVirtualMediaBoot method), 3371
validate_inspection() (ironic.drivers.modules.ilo.boot.IloVirtualMediaBoot method), 3238	validate_rescue() (ironic.drivers.modules.network.neutron.NeutronNetwork method), 3454
validate_inspection() (ironic.drivers.modules.network.noop.NoopNetwork method), 3461	validate_rescue() (ironic.drivers.modules.pxe_base.PXEBaseMixin method), 3998
validate_inspection() (ironic.drivers.modules.pxe_base.PXEBaseMixin method), 3998	validate_sort_dir() (in module ironic.api.controllers.v1.utils), 2164
validate_inspection() (ironic.drivers.modules.redfish.boot.RedfishVirtualMediaBoot method), 3486	validate_user_deploy_steps_and_templates() (in module ironic.conductor.steps), 2681
validate_instance_info_traits() (in module ironic.conductor.utils), 2742	value() (ironic.objects.bios.BIOSSetting prop- erty), 4254
validate_job_queue() (in module ironic.drivers.modules.drac.job), 3100	value_power_off (ironic.drivers.modules.snmp.SNMPDriverAPCMasterSw attribute), 4005
validate_limit() (in module ironic.api.controllers.v1.utils), 2164	value_power_off (ironic.drivers.modules.snmp.SNMPDriverAPCMasterSw attribute), 4006
validate_network() (in module ironic.common.neutron), 2396	value_power_off (ironic.drivers.modules.snmp.SNMPDriverAPCRackPDU attribute), 4007
validate_network_data() (in module ironic.api.controllers.v1.node), 2052	value_power_off (ironic.drivers.modules.snmp.SNMPDriverAten attribute), 4008
validate_network_port() (in module ironic.common.utils), 2502	value_power_off (ironic.drivers.modules.snmp.SNMPDriverBaytechMRP2 attribute), 4012
validate_node() (in module ironic.conductor.deployments), 2524	value_power_off (ironic.drivers.modules.snmp.SNMPDriverCyberPower attribute), 4013
validate_port_info() (in module ironic.common.neutron), 2398	value_power_off (ironic.drivers.modules.snmp.SNMPDriverEatonPower attribute), 4015
validate_port_physnet() (in module ironic.conductor.utils), 2743	value_power_off
validate_raid_config() (ironic.drivers.base.RAIDInterface method), 4153	
validate_raid_config() (ironic.drivers.modules.noop.NoRAID method), 3963	

[\(ironic.drivers.modules.snmp.SNMPDriverTelecranic\\_vendor\\_interface\(\)](#)  
[attribute\), 4017](#) [\(ironic.objects.node.NodeSetPowerStatePayload](#)  
[value\\_power\\_off\(\)](#) [property\), 4357](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverSimpleVendor\\_interface\(\)](#)  
[property\), 4016](#) [\(ironic.objects.node.NodeSetProvisionStatePayload](#)  
[value\\_power\\_on](#) [property\), 4361](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitch](#)  
[attribute\), 4005](#) [\(ironic.api.controllers.v1.driver.DriversController](#)  
[value\\_power\\_on](#) [attribute\), 1991](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverAPCMasterSwitchPlus](#)  
[attribute\), 4006](#) [\(ironic.api.controllers.v1.node.NodesController](#)  
[value\\_power\\_on](#) [attribute\), 2048](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverAPCRackPDUs](#) [\(in module](#)  
[attribute\), 4007](#) [ironic.api.controllers.v1.utils\), 2164](#)  
[value\\_power\\_on](#) [vendor\\_passthru\(\)](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverAten](#) [\(ironic.conductor.manager.ConductorManager](#)  
[attribute\), 4008](#) [method\), 2540](#)  
[value\\_power\\_on](#) [vendor\\_passthru\(\)](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverBaytechMRP27C](#) [ironic.conductor.rpcapi.ConductorAPI](#)  
[attribute\), 4012](#) [method\), 2668](#)  
[value\\_power\\_on](#) [VendorInterface \(class in](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverCyberPower](#) [ironic.drivers.base\), 4162](#)  
[attribute\), 4013](#) [VendorMetadata \(class in ironic.drivers.base\),](#)  
[value\\_power\\_on](#) [4167](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverEatonPowerPassthru](#) [\(class in](#)  
[attribute\), 4015](#) [ironic.drivers.modules.ilo.vendor\),](#)  
[value\\_power\\_on](#) [3346](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverTelecranicPassthru](#) [\(class in](#)  
[attribute\), 4017](#) [ironic.drivers.modules.ipmitool\), 3902](#)  
[value\\_power\\_on\(\)](#) [VendorPassthruException, 2321](#)  
[\(ironic.drivers.modules.snmp.SNMPDriverSimpleSLES](#) [\(in module ironic.common.states\), 2466](#)  
[property\), 4016](#) [verify\\_firmware\\_update\\_args\(\)](#)  
[value\\_within\\_timeout\(\)](#) [\(in module](#) [\(in module](#)  
[ironic.conductor.utils\), 2747](#) [ironic.drivers.modules.ilo.firmware\\_processor\),](#)  
[vendor](#) [\(ironic.drivers.base.BareDriver](#) [3296](#)  
[tribute\), 4039](#) [verify\\_image\\_checksum\(\)](#) [\(in module](#)  
[vendor\\_interface](#) [ironic.drivers.modules.ilo.common\),](#)  
[\(ironic.db.sqlalchemy.models.Node](#) [3287](#)  
[attribute\), 2899](#) [verify\\_iscsi\\_connection\(\)](#) [\(in module](#)  
[vendor\\_interface\(\)](#) [ironic.drivers.modules.iscsi\\_deploy\),](#)  
[\(ironic.objects.node.Node](#) [property\),](#) [3945](#)  
[4341](#) [verify\\_node\\_for\\_deallocation\(\)](#) [\(in](#)  
[vendor\\_interface\(\)](#) [module ironic.conductor.allocations\),](#)  
[\(ironic.objects.node.NodeCorrectedPowerStatePayload](#) [2709](#)  
[property\), 4350](#) [VERIFYING](#) [\(in module ironic.common.states\),](#)  
[vendor\\_interface\(\)](#) [2467](#)  
[\(ironic.objects.node.NodeCRUDPayload](#) [version](#)  
[property\), 4345](#) [ironic-dbsync command line](#)  
[vendor\\_interface\(\)](#) [option, 1699](#)  
[\(ironic.objects.node.NodePayload](#) [Version](#) [\(class in ironic.api.controllers.base\),](#)  
[property\), 4354](#) [2198](#)

version (ironic.db.sqlalchemy.models.Allocation attribute), 2891

version (ironic.db.sqlalchemy.models.BIOSSetting attribute), 2892

version (ironic.db.sqlalchemy.models.Chassis attribute), 2892

version (ironic.db.sqlalchemy.models.Conductor attribute), 2893

version (ironic.db.sqlalchemy.models.ConductorHardwareInterface attribute), 2894

version (ironic.db.sqlalchemy.models.DeployTemplate attribute), 2895

version (ironic.db.sqlalchemy.models.DeployTemplateStep attribute), 2896

version (ironic.db.sqlalchemy.models.IronicBase attribute), 2896

version (ironic.db.sqlalchemy.models.Node attribute), 2899

version (ironic.db.sqlalchemy.models.NodeTag attribute), 2900

version (ironic.db.sqlalchemy.models.NodeTrait attribute), 2900

version (ironic.db.sqlalchemy.models.Port attribute), 2902

version (ironic.db.sqlalchemy.models.Portgroup attribute), 2903

version (ironic.db.sqlalchemy.models.VolumeConnector attribute), 2904

version (ironic.db.sqlalchemy.models.VolumeTarget attribute), 2904

VERSION (ironic.objects.allocation.Allocation attribute), 4222

VERSION (ironic.objects.allocation.AllocationCRUDNotification attribute), 4233

VERSION (ironic.objects.allocation.AllocationCRUDPayload attribute), 4234

VERSION (ironic.objects.bios.BIOSSetting attribute), 4249

VERSION (ironic.objects.bios.BIOSSettingList attribute), 4255

VERSION (ironic.objects.chassis.Chassis attribute), 4262

VERSION (ironic.objects.chassis.ChassisCRUDNotification attribute), 4271

VERSION (ironic.objects.chassis.ChassisCRUDPayload attribute), 4272

VERSION (ironic.objects.conductor.Conductor attribute), 4272

VERSION (ironic.objects.deploy\_template.DeployTemplate attribute), 4281

VERSION (ironic.objects.deploy\_template.DeployTemplateCRUDNotification attribute), 4295

VERSION (ironic.objects.deploy\_template.DeployTemplateCRUDPayload attribute), 4295

VERSION (ironic.objects.deployment.Deployment attribute), 4296

VERSION (ironic.objects.node.Node attribute), 4324

VERSION (ironic.objects.node.NodeConsoleNotification attribute), 4345

VERSION (ironic.objects.node.NodeCorrectedPowerStateNotification attribute), 4347

VERSION (ironic.objects.node.NodeCorrectedPowerStatePayload attribute), 4348

VERSION (ironic.objects.node.NodeCRUDNotification attribute), 4342

VERSION (ironic.objects.node.NodeCRUDPayload attribute), 4342

VERSION (ironic.objects.node.NodeMaintenanceNotification attribute), 4351

VERSION (ironic.objects.node.NodePayload attribute), 4352

VERSION (ironic.objects.node.NodeSetPowerStateNotification attribute), 4354

VERSION (ironic.objects.node.NodeSetPowerStatePayload attribute), 4355

VERSION (ironic.objects.node.NodeSetProvisionStateNotification attribute), 4358

VERSION (ironic.objects.node.NodeSetProvisionStatePayload attribute), 4359

VERSION (ironic.objects.notification.EventType attribute), 4362

VERSION (ironic.objects.notification.NotificationBase attribute), 4364

VERSION (ironic.objects.notification.NotificationPayloadBase attribute), 4365

VERSION (ironic.objects.notification.NotificationPublisher attribute), 4367

VERSION (ironic.objects.port.Port attribute), 4368

VERSION (ironic.objects.port.PortCRUDNotification attribute), 4386

VERSION (ironic.objects.port.PortCRUDPayload attribute), 4387

VERSION (ironic.objects.portgroup.Portgroup attribute), 4388

VERSION (ironic.objects.portgroup.PortgroupCRUDNotification attribute), 4402

VERSION (ironic.objects.portgroup.PortgroupCRUDPayload attribute), 4403

VERSION (ironic.objects.trait.Trait attribute), 4403



VERSION (*ironic.objects.trait.TraitList* attribute), *vif\_list()* (*ironic.drivers.modules.network.common.VIFPortIDMixin* method), 3428  
 4408  
 VERSION (*ironic.objects.volume\_connector.VolumeConnector* attribute), *vif\_detach()* (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3464  
 4413  
 VERSION (*ironic.objects.volume\_connector.VolumeConnector* attribute), *VifInvalidForAttach*, 2321  
 4428  
 VERSION (*ironic.objects.volume\_connector.VolumeConnector* attribute), *VIFPortIDMixin* (class in *ironic.objects.volume\_target.VolumeTarget*), 4429  
 4429  
 VERSION (*ironic.objects.volume\_target.VolumeTarget* attribute), *ironic.drivers.modules.network.common*, 3426  
 4430  
 VERSION (*ironic.objects.volume\_target.VolumeTarget* attribute), *ironic.db.sqlalchemy.models.VolumeTarget* (class in *ironic.objects.volume\_target.VolumeTarget*), 2904  
 4447  
 VERSION (*ironic.objects.volume\_target.VolumeTarget* attribute), *VolumeTargetCRUDPayload* (class in *ironic.objects.volume\_target.VolumeTarget*), 4448  
 4448  
 version command line option *volume\_id()* (*ironic.objects.volume\_target.VolumeTarget* property), 4448  
 --help, 1702  
 -h, 1702  
 version() (in module *ironic.db.migration*), *volume\_type()* (*ironic.db.sqlalchemy.models.VolumeTarget* attribute), 2904  
 3031  
 version() (in module *ironic.db.sqlalchemy.migration*), *volume\_type()* (*ironic.objects.volume\_target.VolumeTarget* property), 4447  
 2889  
 version() (*ironic.cmd.dbsync.DBCommand* method), 2226  
 2226  
 VersionSelectorApplication (class in *ironic.api.app*), 2208  
 2208  
 vif\_attach() (*ironic.conductor.manager.ConductorManager* method), 2540  
 2540  
 vif\_attach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2672  
 2672  
 vif\_attach() (*ironic.drivers.base.NetworkInterface* method), 4133  
 4133  
 vif\_attach() (*ironic.drivers.modules.network.common.NoopNetworkInterface* method), 3422  
 3422  
 vif\_attach() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3462  
 3462  
 vif\_detach() (*ironic.conductor.manager.ConductorManager* method), 2540  
 2540  
 vif\_detach() (*ironic.conductor.rpcapi.ConductorAPI* method), 2675  
 2675  
 vif\_detach() (*ironic.drivers.base.NetworkInterface* method), 4134  
 4134  
 vif\_detach() (*ironic.drivers.modules.network.common.NoopNetworkInterface* method), 3425  
 3425  
 vif\_detach() (*ironic.drivers.modules.network.noop.NoopNetworkInterface* method), 3463  
 3463  
 vif\_list() (*ironic.conductor.manager.ConductorManager* method), 2540  
 2540  
 vif\_list() (*ironic.conductor.rpcapi.ConductorAPI* method), 2676  
 2676  
 vif\_list() (*ironic.drivers.base.NetworkInterface* method), 4135  
 4135

VolumeTargetCRUDPayload (class in WSGIService (class in  
*ironic.objects.volume\_target*), 4448 *ironic.common.wsgi\_service*), 2503

VolumeTargetNotFound, 2322

VolumeTargetsController (class in X  
*ironic.api.controllers.v1.volume\_target*), XClarityError, 2323  
2183 XClarityHardware (class in  
*ironic.drivers.xclarity*), 4221

**W** XClarityManagement (class in  
*ironic.drivers.modules.xclarity.management*),  
wait() (*ironic.common.json\_rpc.server.WSGIService* method), 2241 3569

wait() (*ironic.common.wsgi\_service.WSGIServiceXClarityPower* (class in  
method), 2504 *ironic.drivers.modules.xclarity.power*),  
3575

wait\_for\_host\_agent() (in module  
*ironic.common.neutron*), 2399

wait\_for\_job\_completion() (in module  
*ironic.drivers.modules.drac.job*), 3101

wait\_for\_port\_status() (in module  
*ironic.common.neutron*), 2401

WANBOOT (in module  
*ironic.common.boot\_devices*), 2261

warn\_about\_deprecated\_extra\_vif\_port\_id()  
(in module *ironic.common.utils*), 2503

warn\_about\_unsafe\_shred\_parameters()  
(in module *ironic.cmd.conductor*), 2225

WARNING (*ironic.objects.fields.NotificationLevel*  
attribute), 4309

wipe\_cleaning\_internal\_info() (in  
module *ironic.conductor.utils*), 2747

wipe\_deploy\_internal\_info() (in mod-  
ule *ironic.conductor.utils*), 2748

wipe\_internal\_info\_on\_power\_off()  
(in module *ironic.conductor.utils*), 2748

wipe\_token\_and\_url() (in module  
*ironic.conductor.utils*), 2748

wrap\_ipv6() (in module *ironic.common.utils*),  
2503

wrapfunc() (in module *ironic.api.functions*),  
2214

write\_image()  
(*ironic.drivers.modules.agent.AgentDeployMixin*  
method), 3588

write\_image()  
(*ironic.drivers.modules.ansible.deploy.AnsibleDeploy*  
method), 3062

write\_image()  
(*ironic.drivers.modules.iscsi\_deploy.ISCSIDeploy*  
method), 3921

write\_to\_file() (in module  
*ironic.common.utils*), 2503

WSGIService (class in  
*ironic.common.json\_rpc.server*), 2240