HITACHI
Inspire the Next

Hitachi Content Platform S Series Node

3.0

HCP S11 and S31 Node API Reference

This book contains all the information you need in order to use the HCP S Series management API with an HCP S11 or S31 Node. This RESTful, HTTP-based API enables you to programmatically configure, monitor, and manage an S11 or S31 Node. The book explains how to use the management API to retrieve information about and manipulate S11 and S31 Node resources. The book also includes an introduction to the S Series Node concepts that underlie the management API resources.

MK-HCPS023-00
April 2019

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara Corporation at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara Corporation.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and

2. Verifying that data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

EXPORT CONTROLS - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API, or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Contents

Preface

This book contains all the information you need in order to use the **Hitachi Content Platform (HCP) S Series management API** with an **HCP S11 or S31 Node**. This RESTful, HTTP-based API enables you to programmatically configure, monitor, and manage an S11 or S31 Node. The book explains how to use the management API to retrieve information about and manipulate S11 and S31 Node resources. The book also includes an introduction to the S Series Node concepts that underlie the management API resources.

Intended audience

This book is intended for people who want to configure, monitor, and manage an HCP S11 or S31 Node programmatically. This audience includes:

- S Series Node administrators and monitors
- Authorized S Series Node service providers

This book assumes that you are familiar with HTTP.

Product version

This book applies to release 3.0 or later of the HCP S Series Node.

Release notes

Read the release notes before installing and using this product. They may contain requirements or restrictions that are not fully described in this document or updates or corrections to this document. Release notes are available on Hitachi Vantara Support Connect:

https://knowledge.hitachivantara.com/Documents

Syntax notation

The table below describes the conventions used for the syntax of URLs in this book.

Notation	Meaning	Example
boldface	Type exactly as it appears in the syntax (if the context is case insensitive, you can vary the case of the letters you type)	This book shows: **https://mapi.***node-domain-name***:9090/***resource-identifier* You enter: https://mapi.s-node-1.example.com:9090/user_accounts
italics	Replace with a value of the indicated type	

Related documents

- *HCP S Series Node Help* (MK-HCPS022) — This Help system contains information about configuring and managing an HCP S11 or S31 Node. The Help includes information you need to effectively use the HCP S Series Management Console. The Help also contains a complete reference for using the HCP S Series management API.

- *HCP S11 and S31 Node Third-Party Copyrights and Licenses* (MK-HCPS024) — This book contains the copyright and license information for third-party software that's incorporated into the HCP S Series operating system and software.

Accessing product documentation

Product documentation is available on Hitachi Vantara Support Connect: https://knowledge.hitachivantara.com/Documents. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Getting help

Hitachi Vantara Support Portal is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to Hitachi Vantara Support Connect for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

Hitachi Vantara Community is a global online community for Hitachi Vantara customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Comments

Please send us your comments on this document:

HCPDocumentationFeedback@HitachiVantara.com

Include the document title and part number, including the revision (for example, -01), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara Corporation.

Thank you!

1

Introduction to HCP S Series Nodes

The **Hitachi Content Platform (HCP) S Series Node** is one of the storage products offered by Hitachi Vantara. This chapter describes the S Series Node concepts you need to understand in order to successfully use the HCP S Series management API. The chapter also includes information about the hardware components of HCP S11 and S31 Nodes.

About HCP S Series Nodes

An HCP S Series Node is a highly efficient, highly available, cost-effective storage device that supports very large amounts of data. Each S Series Node consists of two cooperating server modules and multiple high-density drives in some number of enclosures.

During normal operation, the two server modules in an S Series Node actively share responsibility for all S Series Node functions. Because the server modules are peers, if one module becomes unavailable, the other can still provide full, uninterrupted S Series Node functionality, although performance may be degraded.

The S Series Node data storage implementation ensures that data is well-protected through the use of erasure coding. Additionally, S Series Nodes use several internal processes to continuously check the integrity of both the stored data and the storage media.

S Series Nodes can provide direct-write storage for HCP systems and for HCP for cloud scale systems. S Series Nodes can also serve as storage tiering platforms for HCP systems. A single HCP system or HCP for cloud scale system can seamlessly store data across multiple S Series Nodes, thereby enabling scalability in both capacity and performance.

HCP systems and HCP for cloud scale systems use the S Series Node implementation of the Hitachi API for Amazon S3® to write, retrieve, and otherwise manage objects in an S Series Node. This RESTful, HTTP-based API is compatible with Amazon S3.

For administrative purposes, S Series Nodes provide a web-based Management Console and a RESTful, HTTP-based management API. Using these interfaces, S Series Node administrators and service providers can configure, manage, and monitor an S Series Node. These interfaces can also be used to initiate and verify S Series Node hardware procedures, such as adding and replacing drives.

The newest S Series Node models are the S11 Node and the S31 Node. The S31 Node has more processing power and memory than the S11 Node and also allows for greater storage capacity.

The older S Series Node models are the S10 Node and the S30 Node. The enclosures used in S11 and S31 Nodes can hold more drives than the enclosures used in S10 and S30 Nodes can hold. Also, S11 and S31 Nodes support higher-capacity drives than S10 and S30 Nodes support.

The S11 and S31 Node enclosures are not interchangeable with the S10 and S30 Node enclosures.

HCP S11 and S31 Node hardware components

The main components of an S11 or S31 Node are:

- One **base enclosure** and, optionally, one **expansion enclosure**. An enclosure is a container for the drives used by the S11 or S31 Node. Each enclosure has a main bay and a controller bay.

- Two **server modules**, located in the base-enclosure controller bay. Each server module runs the complete HCP S Series software. This redundancy prevents service interruptions due to either one of the server modules becoming unavailable.

- Two **I/O modules**, located in the expansion-enclosure controller bay. The I/O modules enable communication between the base enclosure and the drives in the expansion enclosure.

- **Data drives**. The data drives store the data written to the S11 or S31 Node and the metadata that describes that data. A base enclosure holds 30, 62, or 94 data drives, all in the main bay. An expansion enclosure holds 42, 74, or 106 data drives, ten in the controller bay and the rest in the main bay.

- Six **database drives**, located in the base enclosure. The database drives store the internal database used by the S11 or S31 Node to hold information such as user-account and bucket definitions and various configuration settings.

  Four of the database drives are in small form factor (SFF) drive carriers in the controller bay. The other two are in large form factor (LFF) drive carriers in the main bay.

User accounts

To access an S Series Node, you need an S Series Node user account. A **user account** is a set of credentials that gives a user permission to use one or more of these interfaces:

- The HCP S Series Management Console
- The HCP S Series management API
- The Hitachi API for Amazon S3 (the S3 compatible API)

Permissions are granted by the roles associated with a user account. For more information on roles, see "User roles" on page 23.

If you have the security role, you can use the HCP S Series Management Console or management API to create, modify, and delete S Series Node user accounts.

An S Series Node can have at most 10,000 user accounts.

User account credentials

User account credentials consist of a username and password. You can use the HCP S Series Management Console or management API to change the password for your own user account at any time. An S Series Node user with the security role can change the password for any user account at any time.

Important: Passwords for S Series Node user accounts created by HCP systems are generated automatically and are not known to administrators of those systems. If you change the password for such a user account, the applicable system will no longer be able to manage or report on its usage of the S Series Node storage.

For you to use the S3 compatible API, your user account must have the data role and additional credentials that consist of an access key and secret key. You can use the HCP S Series Management Console or management API to generate these credentials. Only you can generate the S3 compatible API credentials for your user account.

Note: In release 3.0 of the S Series Node, only an HCP system can be a direct user of the S3 compatible API.

Normally, user account passwords expire after a configurable amount of time. However, security administrators can configure individual user accounts so that the password never expires automatically or so that the password expires immediately. A password that is set to expire immediately expires regardless of whether it's subject to automatic expiration.

If your user account password expires, you can use an interface that requires password access only to change that password. An expired password does not prevent the user account from being used for data access.

Access keys and secret keys do not expire. However, if you lose these keys, you can generate new ones. As soon as you generate new keys, the old keys stop working.

Usernames

When you create an S Series Node user account, you specify a username for the account. The username uniquely identifies that account on the S Series Node.

Usernames:

- Must be 3 through 128 characters long
- Can contain only valid UTF-8 characters
- Cannot contain uppercase letters
- Cannot contain an opening angle bracket (<) or closing angle bracket (>)
- Cannot start with an opening square bracket ([) or closing square bracket (])
- Cannot contain white space
- Must be unique for the current S Series Node

Additionally, the following strings are reserved and cannot be used as usernames:

- *allusers*
- *authenticatedusers*

- *internal*
- *logdelivery*
- *http://acs.amazonaws.com/groups/global/allusers*
- *http://acs.amazonaws.com/groups/global/authenticatedusers*
- *http://acs.amazonaws.com/groups/s3/logdelivery*

You can reuse usernames that are not currently in use. For example, if you delete the account for a user, you can create a new account for that user with the same username as the deleted account had.

Passwords

When you create an S Series Node user account, you specify a password for the account. Users can change their account passwords at any time.

Passwords:

- Can be at most 64 characters long
- Cannot be shorter than the configured minimum password length, which cannot be less than 6
- Can contain any valid UTF-8 characters
- Can include white space
- Are case sensitive
- Must include at least one character from two of these character sets:
  - Letters
  - Numbers
  - Other

The longer the password, the stronger it is likely to be. Using a mix of uppercase and lowercase letters, numbers, and special characters creates an even stronger password.

When changing the password for your own user account, you cannot reuse your current password.

As a security administrator, when you modify a user account, you can reuse the current password.

User account properties

In addition to a username and password, user accounts have these properties:

- Full name. The full name can be used to identify the user for whom the account was created. This name must be 1 through 256 characters long and can contain any valid UTF-8 characters, including white space.

- Description (optional). The description can be at most 1,024 characters long and can contain any valid UTF-8 characters, including white space.

- Roles that determine which interfaces the user can use with the account and what the user can do with those interfaces.

- Whether the account password must be changed before the account can be used for any purpose other than to change the password (that is, whether the password has expired).

- Whether the password for the user account ever expires automatically based on the S Series Node security setting for password expiration.

- Whether the account is enabled or disabled. While a user account is disabled, it cannot be used for any purpose. You might choose to disable an account, for example, while the user for whom you created it is on vacation.

User roles

A **role** is a named collection of permissions that can be associated with an S Series Node user account. The roles associated with a user account determine which S Series Node interfaces the user can use and what the user can do with those interfaces. Roles generally correspond to job functions.

A user account must be associated with one or more roles. The account user has all the permissions granted by each of the associated roles.

The roles that you can associate with a user account are:

- **Administrator** — Grants permission to use the HCP S Series Management Console and management API to:

    - View S Series Node configuration and status.

    - Perform configuration activities (such as changing server module IP addresses).

    - Insert comments into and download the S Series Node internal logs. For information on the internal logs, see "Internal logs" on page 51.

    - View the user account and bucket lists. For information on buckets, see "Buckets" on page 27.

    - Create, modify, and delete buckets and view the list of irreparable objects in those buckets.

    The administrator role does not grant permission to:

    - Configure user accounts.

    - Create, view, or otherwise manage objects in buckets. For information on objects, see "S Series objects" on page 27.

- **Monitor** — Grants permission to use the HCP S Series Management Console and management API to:

    - View S Series Node configuration and status

    - Insert comments into the internal logs

    - View the bucket list and the list of irreparable objects in those buckets

    The monitor role does not grant permission to:

    - View or configure user accounts

    - Create, view, or otherwise manage objects in buckets

- **Security** — Grants permission to use the HCP S Series Management Console and management API to:

  - Create and manage user accounts
  - Configure security settings (such as disabling SSH access to the S Series Node)
  - View security event messages (such as messages about unsuccessful attempts to log in to the HCP S Series Management Console)
  - Insert comments into the internal logs

Tip: Always have at least two user accounts that have the security role. This configuration ensures that if one of the accounts with the security role becomes disabled, another account that can manage user accounts still exists.

- **Service** — Grants permission to use the HCP S Series Management Console and management API to:

  - View S Series Node configuration and status
  - Perform most configuration activities
  - Perform maintenance activities (such as replacing a failed drive)
  - Insert comments into and download the internal logs
  - Update the HCP S Series operating system and software
  - View the bucket list and the list of irreparable objects in those buckets

  The service role does not grant permission to:

  - View or configure user accounts
  - Create, view, or otherwise manage objects in buckets

Note: You should associate the service role only with user accounts created for authorized service providers.

- **Data** — Grants permission to use the Hitachi API for Amazon S3 (the S3 compatible API) to:

  - Create and manage buckets

  - ○ Store and manage objects in buckets

  With this role, you can also use the Management Console and management API to generate your S3 compatible API access key and secret key.

All users can use the HCP S Series Management Console and management API to change their own passwords.

Considerations for working with user accounts

If you have the security role, you can create, modify, and delete user accounts. The following considerations apply to working with user accounts:

- You cannot change the username for an existing user account.

- When changing the password for a user account other than your own, you can reuse the current password. When changing the password for you own user account, you cannot reuse the current password.

- At all times, at least one user account must have the security role. Therefore:

  - ○ You cannot remove the security role from the last user account that has that role.

  - ○ You cannot delete the last user account that has the security role.

- You cannot disable the last user account that has the security role. However, that user account can be disabled automatically due to the configured number of consecutive invalid login attempts.

  For information on setting the limit for invalid login attempts, see "User account and Management Console properties" on page 47.

- If you disable the user account you used to log in to the current HCP S Series Management Console, the Console session immediately ends.

- You cannot delete a user account that owns any buckets. To delete such a user account, you first need to change the owner of each applicable bucket to a different user.

  For information on buckets, see "Buckets" on the next page.

- You cannot delete the user account you're currently using to access the S Series Node.
- Multiple people can use the same user account concurrently to access the same or different S Series Node interfaces. To prevent this from happening, you should create a separate account for each user. Users should keep their passwords confidential.

S Series objects

An S Series Node stores objects. An S Series **object** is a combination of:

- An exact digital reproduction of data as it existed before it was stored on the S Series Node.
- Information that describes the object (for example, the data size and the object creation date). This information is called **metadata**.

When data is written to an S Series Node, the S Series Node creates an object from that data.

S Series objects are not the same as HCP objects, and the two types of objects do not have a one-to-one correspondence with each other. Each HCP object tiered to an S Series Node can result in multiple objects on the S Series Node.

Buckets

An S Series Node stores objects in buckets. A **bucket** is a logical grouping of objects such that the objects in one bucket are not visible in any other bucket.

Buckets have these properties:

- A name.
- An owner.
- A description (optional). The description can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space.

An S Series Node can have at most 10,000 buckets.

If you have the administrator role, you can use the HCP S Series Management Console or management API to create, modify, and delete buckets. If you have the data role, you can use the Hitachi API for Amazon S3 (the S3 compatible API) to create and delete buckets.

Bucket names

When you create a bucket, you specify a name for it. This name uniquely identifies that bucket on the S Series Node.

Bucket names:

- Must be 3 through 63 characters long
- Can contain only lowercase letters, digits, hyphens (-), and periods (.)
- Cannot contain consecutive periods
- Must start and end with a lowercase letter or digit
- Can consist of multiple parts delimited by periods, where each part must start and end with a lowercase letter or digit
- Cannot have the form of an IP address (for example, 192.168.10.4)

Bucket owners

Each S Series Node bucket has an owner that corresponds to an S Series Node user account with the data role. When you create a bucket, you select the bucket owner. Only the owner of a bucket can store and manage objects in that bucket.

If you have the administrator role, you can use the HCP S Series Management Console or management API to change the owner of a bucket to a different user account.

An individual user can own at most 100 buckets.

Considerations for working with buckets

These considerations apply to modifying and deleting buckets:

- You cannot change the name of an existing bucket.
- If you change the owner of a bucket that's used by an HCP system or by an HCP for cloud scale system, you need to provide the applicable system with the credentials for the new owner. Until you provide the new credentials, that system cannot store, retrieve, or otherwise manage objects in the bucket.
- You can delete a bucket only if it's empty (that is, it does not contain any objects).

Networking

An S Series Node makes use of three networks:

- **The access network** is used for external client access to the S Series Node through the Hitachi API for Amazon S3 (the S3 compatible API). This network can also be used for external client access to the S Series Node through the HCP S Series Management Console and management API.

Note: HCP always communicates with S Series Nodes over the access network for both data access and management purposes.

- **The management network** is used for external client access to the S Series Node through the HCP S Series Management Console and management API. This network cannot be used for access to the S Series Node through the S3 compatible API.

  You can use the management network to segregate network traffic for management purposes from network traffic for data access.
- **The server interconnect network** is used exclusively for communication between the two S Series Node server modules. The two server modules are the only devices on this isolated network.

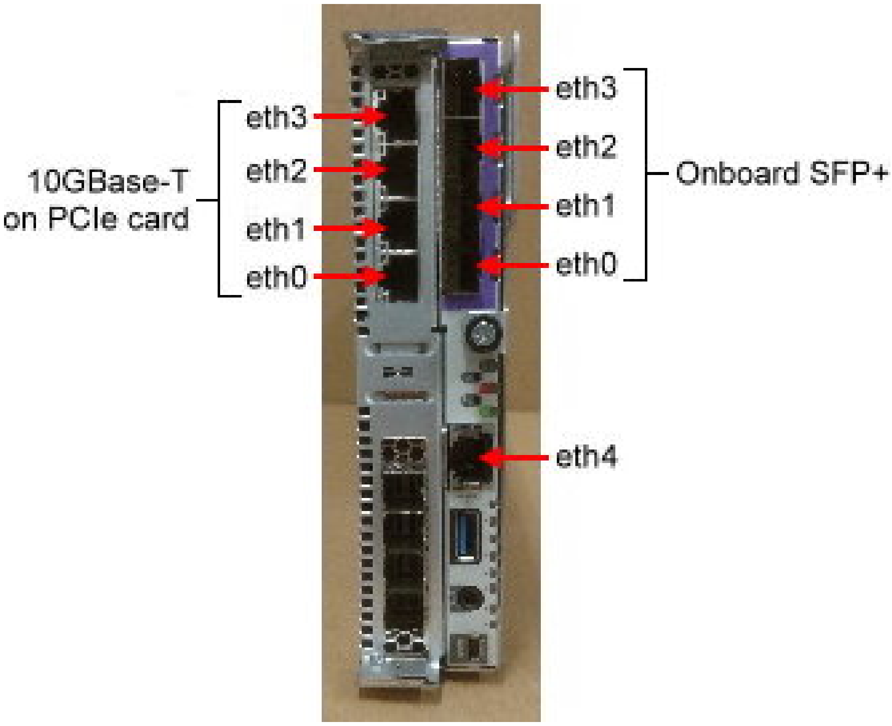
If you have the administrator or service role, you can use the HCP S Series Management Console or management API to modify S Series Node network configurations.

Server module Ethernet ports

Each server module in an S11 or S31 Node has these Ethernet ports:

- Four onboard 10Gb SFP+ ports — Used for the access network if the optional 10GBase-T PCIe card is not present:
  - If these ports are used, the port device names are eth0, eth1, eth2, and eth3.
  - If these ports are unused, they are disabled.
- Optionally, four 10GBase-T ports on a PCIe card — If present, used instead of the onboard SFP+ ports for the access network. The port device names are eth0, eth1, eth2, and eth3.
- One onboard 1000Base-T Ethernet port — Used for the management network. The port device name is eth4.

The figure below shows the locations of these ports on a server module.



Access network

You can configure the access network so that the S Series Node expects one, some, or all of the available ports to be connected to an active switch. However, regardless of the configured expectations, the S Series Node uses each connected port, either as an active port or as a backup port, depending on the bonding mode.

Tip: To prevent the S Series Node from issuing alerts about unexpected or missing port connections, set the connection expectation for each port according to whether the port is actually connected to an active switch.

The access network ports connect the server modules to the customer networking infrastructure through one or two Ethernet switches. The recommended configurations are:

- Both the access network and one or two switches configured for IEEE 802.3ad Link Aggregation Control Protocol (LCAP) bonding:
    - With one switch, all the ports being used on both server modules connect to that switch.
    - With two switches, all the ports being used on one server module connect to one switch. All the ports being used on the other server module connect to the other switch.
- Both the access network and two switches configured for active-backup bonding. In this case, all the even-numbered ports (that is, eth0 and eth2) being used on both server modules connect to one switch. All the odd-numbered ports (that is, eth1 and eth3) being used on both server modules connect to the other switch.

For the appropriate configuration for your S Series Node, consult your network administrator.

Access network IP addresses

Each server module has both physical and virtual access network IP addresses. To ensure that access to the HCP S Series Node is not disrupted by the unavailability of a single server module, clients should use the virtual IP addresses to communicate with the S Series Node. Communications that use a virtual IP address for an unavailable server module are automatically redirected to the available server module. When the unavailable server module becomes available again, communications using the virtual IP address for that module revert back to that module.

The access network can have an IP mode of either IPv4 or IPv6. If the IP mode is IPv4, the two server modules must have access network IPv4 addresses on the same IPv4 subnet. If the IP mode is IPv6, the two server modules must have primary access network IPv6 addresses on the same IPv6 subnet. In all cases, the virtual IP address for a server module must be on the same subnet as the physical IP address.

With an IP mode of IPv6, the server modules can also have secondary physical and virtual access network IPv6 addresses. These addresses must be on the same IPv6 subnet, and that subnet must not overlap the primary access network subnet. If one server module has a secondary access network IPv6 address, the other server module must also have a secondary access network IPv6 address.

The table below shows a sample IPv6 configuration for the access network.

Property	Values
Primary IPv6 properties	
Gateway address	2001:db8::ff:ff:ff:0
Prefix length	64
Physical IP addresses	Server module 1: 2001:db8::1:0:0:1 Server module 2: 2001:db8::1:0:0:2
Virtual IP addresses	Server module 1: 2001:db8::1:0:0:3 Server module 2: 2001:db8::1:0:0:4
Secondary IPv6 properties	
Gateway address	2001:db9::ff:ff:ff:0
Prefix length	64
Physical IP addresses	Server module 1: 2001:db9::1:0:0:1 Server module 2: 2001:db9::1:0:0:2
Virtual IP addresses	Server module 1: 2001:db9::1:0:0:3 Server module 2: 2001:db9::1:0:0:4

The access network subnet or subnets cannot overlap the subnets for the S Series Node management and server interconnect networks.

Note: In the zone definition for the S Series Node in DNS, use the virtual IP addresses of the server modules. For information on configuring an S Series Node in DNS, see the HCP S Series Node Help.

Access network properties

The access network has these properties:

- **IP mode (either IPv4 or IPv6).** By default, the access network has an IP mode of IPv4.
- **If the IP mode is IPv4:**
    - **IPv4 gateway address.** This is the address from which communications initiated by the S Series Node are sent when the access network is the selected network for the particular type of communication and IPv4 addressing is selected.

      By default, the access network has an IPv4 gateway address of 10.0.0.254.
    - **IPv4 subnet.** With the Management Console, you use the IPv4 gateway address and a four-octet subnet mask to specify the IPv4 subnet. With the management API, you use CIDR notation to specify the IPv4 subnet.

      By default, the access network has an IPv4 subnet of 10.0.0.0/24 and a four-octet subnet mask of 255.255.255.0.
    - **Physical IPv4 address for each server module.** By default, the access network has physical IPv4 addresses of 10.0.0.1 for server module 1 and 10.0.0.2 for server module 2.
    - **Virtual IPv4 address for each server module.** By default, the access network virtual IP addresses are not set. These IP addresses must be set during the initial on-site configuration of the S Series Node.
- **If the IP mode is IPv6:**
    - **Primary IPv6 gateway address.** This is the address from which communications initiated by the S Series Node are sent when the access network is the selected network for the particular type of communication and primary IPv6 addressing is selected.

  - **Primary IPv6 subnet.** With the Management Console, you use the primary IPv6 gateway address and an IPv6 prefix length to specify the primary IPv6 subnet. With the management API, you use CIDR notation to specify the primary IPv6 subnet.

  - **Primary physical IPv6 address for each server module.** These IP addresses must be set during the initial on-site configuration of the S Series Node.

  - **Primary virtual IPv6 address for each server module.** These IP addresses must be set during the initial on-site configuration of the S Series Node.

  - **Optionally, secondary IPv6** gateway address, subnet, physical address for each server module, and virtual address for each server module.

    With the Management Console, you use the secondary IPv6 gateway address and an IPv6 prefix length to specify the secondary IPv6 subnet. With the management API, you use CIDR notation to specify the secondary IPv6 subnet.

- **VLAN ID.** If the networking infrastructure supports virtual networking, valid values for the VLAN ID are integers in the range 0 through 4,094. If the networking infrastructure doesn't support virtual networking, the VLAN ID must be 0.

  If the access network has a nonzero VLAN ID, the applicable switches must be configured to support that ID. Additionally, the networking infrastructure must be configured to allow client requests to be routed to the S Series Node through the access network.

  By default, the access network has a VLAN ID of 0.

- **Maximum transmission unit (MTU).** The MTU is the largest packet size supported for data sent on the network.

  The MTU for a network can be 1,500 or 9,000. The larger MTU reduces overhead and increases network throughput. An MTU of 9,000 is possible only if it is supported by the networking infrastructure.

  By default, the access network has an MTU of 1,500.

- **Combined speed and duplex setting.** By default, the access network has a speed and duplex setting of **auto**. With this setting, the S Series

Node detects the speed and duplex settings of the device with which it's communicating. The S Series Node then adjusts its own settings to provide the highest possible data rate.

- **A bonding mode of IEEE 802.3ad or active-backup.** By default, the access network has a bonding mode of active-backup.

  When the bonding mode is active-backup, the active port is the lowest-numbered connected port. All other connected ports are backup ports.

- **A connection expection (ON or OFF) for each of ports 0, 1, 2, and 3.** The setting for a port applies to the port on both server modules.

  The S Series Node issues an alert if:

  - A port is expected to be connected (that is, the port setting is **ON**) but is not connected to an active port on a network switch.

  - A port is not expected to be connected (that is, the port setting is **OFF**) but is connected to an active port on a network switch.

Management network

For the management network, each server module has one 1Gb Ethernet port. These ports connect the server modules to your networking infrastructure through one or two Ethernet switches:

- With one Ethernet switch, the management ports on both server modules connect to the same switch. With this configuration, if connectivity to the switch is lost, access to the S Series Node over the management network is not possible.

- With two Ethernet switches, the management port on each server module connects to a different switch. With this configuration, loss of connectivity to one switch does not prevent access to the S Series Node over the management network.

Use of the management network is not required. If you don't plan to use this network, you can leave the management ports unconnected.

Tip: If you don't connect the management network ports, disable monitoring of the management network. Disabling monitoring prevents the S Series Node from issuing alerts about the network not being connected.

Modifying the management network causes the S Series Node to reboot. Enabling or disabling monitoring of the management network does not cause a reboot.

Management network IP addresses

The management network can have an IP mode of either IPv4 or IPv6. If the IP mode is IPv4, the two server modules must have management IPv4 addresses on the same IPv4 subnet. If the IP mode is IPv6, the two server modules must have primary management IPv6 addresses on the same IPv6 subnet.

With an IP mode of IPv6, the server modules can also have secondary management IPv6 addresses. These addresses must be on the same IPv6 subnet, and that subnet must not overlap the subnet for the primary management IPv6 addresses. If one server module has a secondary management IPv6 address, the other server module must also have a secondary management IPv6 address.

The management network subnet or subnets cannot overlap the subnets for the S Series Node access and server interconnect networks.

Management network properties

The management network has these properties:

- **An IP mode (either IPv4 or IPv6).** By default, the management network for a new S Series Node has an IP mode of IPv4.
- **If the IP mode is IPv4:**
    - **IPv4 gateway address.** This is the address from which communications initiated by the S Series Node are sent when the management network is the selected network for the particular type of communication and IPv4 addressing is selected.

      By default, the management network has an IPv4 gateway address of 10.2.2.254.
    - **IPv4 subnet.** With the Management Console, you use the IPv4 gateway address and a four-octet subnet mask to specify the IPv4 subnet. With the management API, you use CIDR notation to specify the IPv4 subnet.

By default, the management network has an IPv4 subnet of 10.2.2.0/24 and a four-octet subnet mask of 255.255.255.0.

The management network IPv4 subnet cannot start with 192.168.

- **IPv4 address for each server module.** By default, the management network has IPv4 addresses of 10.2.2.1 for server module 1 and 10.2.2.2 for server module 2.

Note: Do not use a value of 10 as the fourth octet for the IPv4 gateway address or server module IPv4 addresses. This value is reserved for use by authorized service providers.

- **If the IP mode is IPv6:**
  - **Primary IPv6 gateway address.** This is the address from which communications initiated by the S Series Node are sent when the management network is the selected network for the particular type of communication and primary IPv6 addressing is selected.
  - **Primary IPv6 subnet.** With the Management Console, you use the primary IPv6 gateway address and an IPv6 prefix length to specify the primary IPv6 subnet. With the management API, you use CIDR notation to specify the primary IPv6 subnet.
  - **Primary IPv6 address for each server module.**
  - **Optionally, secondary IPv6** gateway address, subnet, and address for each server module.

    With the Management Console, you use the secondary IPv6 gateway address and an IPv6 prefix length to specify the secondary IPv6 subnet. With the management API, you use CIDR notation to specify the secondary IPv6 subnet.

Note: Do not use a value of 000A as the last segment for the primary or secondary IPv6 gateway address or primary or secondary server module IPv6 addresses. This value is reserved for use by authorized service providers.

- **VLAN ID.** If the networking infrastructure supports virtual networking, valid values for the VLAN ID are integers in the range 0 through 4,094.

If the networking infrastructure doesn't support virtual networking, the VLAN ID must be 0.

If the management network has a nonzero VLAN ID, the management switches must be configured to support that ID. Additionally, the networking infrastructure must be configured to allow client requests to be routed to the S Series Node through the management network.

By default, the management network has a VLAN ID of 0.

Note: For internal purposes, the S Series Node uses VLAN IDs of either 700 and 800 or 701 and 801. You cannot use the HCP S Series Management Console or management API to change the management network VLAN ID to a VLAN ID that's being used internally. If the management network requires the use of a VLAN ID that's being used internally, contact your authorized service provider to have the VLAN ID changed. In this case, changing the VLAN ID entails rebooting the S Series Node. While the S Series Node reboots, it is unavailable for both management and data access purposes.

- **Maximum transmission unit (MTU).** The MTU is the largest packet size supported for data sent on the network.

  The MTU for a network can be 1,500 or 9,000. The larger MTU reduces overhead and increases network throughput. An MTU of 9,000 is possible only if it is supported by the networking infrastructure.

  By default, the management network has an MTU of 1,500.

- **Combined speed and duplex setting.** By default, the management network has a speed and duplex setting of **auto**. This setting cannot be changed.

  With a setting of **auto**, the S Series Node detects the speed and duplex settings of the device with which it’s communicating. The S Series Node then adjusts its own settings to provide the highest possible data rate.

- **Whether monitoring of the management network is enabled.** If you don't physically connect the management network to the customer networking infrastructure, you should disable monitoring for the network. If monitoring is enabled without the physical connections present, the S Series Node reports that the network is not functioning properly, and the HCP S Series Management Console displays alerts to that effect.

By default, management network monitoring is enabled.

Server interconnect network

Each server module has a single internal Ethernet port for the server interconnect network. An internal link connects these ports to each other.

The server interconnect network has an IP mode of IPv4. By default, the subnet for this network is 10.1.1.0/24.

You can change the subnet for the server interconnect network. However, you should do this only if a conflict exists.

The server interconnect network subnet cannot overlap the subnets for the S Series Node access and management networks. Additionally, the server interconnect network subnet cannot overlap any subnet used in your networking environment.

The number of bits in the server interconnect network subnet prefix must be 24 (indicated by the suffix /24 in CIDR notation).

The server interconnect network subnet cannot start with 192.168.

Modifying the server interconnect network subnet causes the S Series Node to reboot.

Considerations for working with networks

These considerations apply to modifying networks:

- You cannot change the names of the S Series Node networks.

- You can modify all properties of the access network and management network except their names. To modify a subnet, change the applicable gateway address and/or the applicable subnet mask or prefix length.

- When you modify the access network or management network, communication with the S Series Node is briefly disrupted. However, the S Series Node server modules are not rebooted.

- You can change the physical or virtual IP address of the server module that's servicing the change request. If the IP address you change is the one the request is using and you're making the change in the HCP S Series Management Console, the Console session immediately ends.

- You can change the subnet for the server interconnect network, but you cannot change the fourth octet of the server module IP addresses on that network.

- When you change the subnet for the server interconnect network, both S Series Node server modules are automatically rebooted. Until the reboot is complete, no communication can occur between the S Series Node and other devices.

- Two different S Series Nodes can have the same server interconnect network subnet and the same server interconnect network IP addresses for their server modules. This is possible because the server interconnect network for each S Series Node is isolated from the server interconnect network for the other S Series Node.

- When you correctly change the configuration of a network, the HCP S Series Management Console displays a success message. However, this message is displayed before the change is fully implemented. To ensure that the change succeeded, check the S Series Node event log. If you do *not* see the following message, the change succeeded:

  ```
  Network configuration change could not be applied
  ```

Transport Layer Security (TLS)

Transport Layer Security (TLS) is a protocol for secure communication over computer networks. When a client request to an HCP S Series Node specifies HTTPS in the URL, both the client request and the response from the S Series Node are secured by TLS.

S Series Nodes support TLS versions 1.0, 1.1, and 1.2, but you can set the minimum version that the S Series Node can use. For example, if you set the minimum TLS version to 1.1, the S Series Node accepts requests that use version 1.1 or 1.2 but rejects requests that use version 1.0.

By default the minimum TLS version for an S Series Node is 1.0.

If you have the administrator or service role, you can use the HCP S Series Management Console or management API to change the minimum TLS version.

Changing the minimum TLS version causes the S Series Node to reboot.

Note: For a release 7.*x* HCP system to use the S Series Node, the S Series Node must have a minimum TLS version of 1.0.

HCP S Series Node identification

Each S Series Node is identified by both a domain name and a serial number.

Domain name

The domain name for an S Series Node must be a valid DNS domain name that can be used for access to that S Series Node (for example, s-node-1.example.com). Valid domain names:

- Can contain only letters, numbers, and hyphens (-)
- Must consist of at least three segments, separated by periods, where each segment is 1 through 63 characters long
- Can be at most 127 characters long, including the periods between segments

The domain name cannot be rhino-name.domain.com.

For clients to access the S Series Node by domain name, the domain must be defined as a primary zone in DNS.

Even if DNS is not in use, the S Series Node must have a domain name. This dummy domain name must comply with the rules for valid domain names.

If you have the administrator or service role, you can use the HCP S Series Management Console or management API to change the domain name for an S Series Node. If you change the domain name and DNS is in use, be sure to also change the domain name in DNS.

For information on configuring an S Series Node in DNS, see the HCP S Series Node Help.

Serial number

The serial number for an S Series Node uniquely identifies the S Series Node. This number is on a label that's attached to the top of the main-bay cover at the front of each enclosure in an S11 or S31 Node.

The serial number is displayed in the bottom left corner of each page of the HCP S Series Management Console. You can also view the serial number on the **Configuration** ▶ **Identification** page of the Management Console or by using the S Series Node management API.

You cannot change the serial number for an S Series Node.

Licensing

When used in conjunction with an HCP system, S Series Node storage is covered by the HCP license. The S Series Node does not require a license of its own. On the S Series Node, the license status is **External**.

Note: On a release 7.*x* HCP system that's using an S Series Node for storage, the HCP System Management Console reports that HCP cannot find license information for the S Series Node.

Client access

An S Series Node has three interfaces for client access:

- The web-based **HCP S Series Management Console** supports only management functions.
- The RESTful **HCP S Series management API** supports only management functions.
- The RESTful **Hitachi API for Amazon S3** (the S3 compatible API) supports only data access functions.

To support the use of HTTPS with these interfaces, the S Series Node must have an SSL server certificate. Using HTTPS with the S3 compatible API is possible only if the S Series Node is configured to support the use of SSL for data access.

If you have the administrator or service role, you can use the HCP S Series Management Console or management API to configure the interfaces that enable access to the S Series Node.

Management Console configuration

You can enable access to the HCP S Series Management Console on both the access network and the management network. At any given time, at least one of these networks must be enabled for Console access. By default, both networks are enabled for Console access.

By default, for both the access and management networks, only HTTPS is enabled for Management Console access. For each of these networks individually, you can also enable HTTP for Console access. You cannot disable HTTPS for Console access on either network.

Support for HTTP without SSL security is provided so that the Management Console can accept requests passed on by load balancers where the load balancer has terminated the SSL connection. Client requests for access to the Management Console should always use HTTPS, not HTTP.

By default, users can access the Management Console from any IP address. You can choose to allow access only from specific IP addresses. Similarly, you can choose to deny access from specific IP addresses. You control how the S Series Node handles IP addresses that are included in both or neither of the lists of allowed or denied addresses.

You can specify message text to appear on the login page of the Management Console. This text is optional. If specified, it can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space.

The text you specify appears above the fields for the username and password on the login page. You can use this text, for example, for messages such as “Authorized Users Only” or “Welcome to the HCP S Series Management Console.”

Management API configuration

You can enable access to an S Series Node through the HCP S Series management API on both the access network and the management network. At any given time, at least one of these networks must be enabled for management API access. By default, both networks are enabled for management API access.

Note: HCP always communicates with S Series Nodes over the access network. If the access network is disabled for the management API, HCP systems cannot use the S Series Node.

By default, for both the access and management networks, only HTTPS is enabled for access to the S Series Node through the management API. For each of these networks individually, you can also enable HTTP for management API access. You cannot disable HTTPS for management API access on either network.

For security reasons, client requests for access to the S Series Node through the management API should always use HTTPS, not HTTP.

By default, users can use the management API to access an S Series Node from any IP address. You can choose to allow access only from specific IP addresses. Similarly, you can choose to deny access from specific IP addresses. You control how the S Series Node handles IP addresses that are included in both or neither of the lists of allowed or denied addresses.

Data access protocol configuration

Note: In release 3.0 of the S Series Node, the only supported data access protocol is the Hitachi API for Amazon S3 (the S3 compatible API).

You can enable or disable use of the S3 compatible API. If you disable use of this API, clients cannot read, write, modify, or delete data stored on the S Series Node.

By default, if the S Series Node supports the use of SSL for data access, both HTTP and HTTPS are enabled for access to the S Series Node through the S3 compatible API. You can disable the use of HTTP with the S3 compatible API, but you cannot disable the use of HTTPS.

If the S Series Node does not support the use of SSL for data access, HTTP is the only option for access through the S3 compatible API.

By default, clients can use the S3 compatible API to access an S Series Node from any IP address. You can choose to allow access only from specific IP addresses. Similarly, you can choose to deny access from specific IP addresses. You control how the S Series Node handles IP addresses that are included in both or neither of the lists of allowed or denied addresses.

Allow and deny lists

An allow list specifies IP addresses that are allowed access to an S Series Node through a given interface. A deny list specifies IP addresses that are denied access through a given interface.

Each entry in an allow or deny list can be:

- A single IP address

- A range of IPv4 addresses specified as *ip-address/subnet-mask* (for example, 192.168.100.197/255.255.255.0) or in CIDR format (for example, 192.168.100.0/24)

- A range of IPv6 addresses specified in CIDR format (for example, 2001:0db8::/32)

The CIDR entry that matches all IPv4 addresses is 0.0.0.0/0. The CIDR entry that matches all IPv6 addresses is 0::0/0.

The same IP address can be included in neither, one, or both of the allow and deny lists for a given interface. To control how the S Series Node handles this, you use the **Allow requests when same IP is used in both lists** option for the interface. The table below describes the effects of selecting or deselecting this option.

	Allow requests when same IP is used in both lists	
List entries	**Selected**	**Deselected**
Allow list: empty Deny list: empty	All IP addresses have access.	No IP addresses have access.
Allow list: at least one entry Deny list: empty	All IP addresses have access.	Only IP addresses in the allow list have access.
Allow list: empty Deny list: at least one entry	All IP addresses not in the deny list have access. IP addresses in the deny list do not.	No IP addresses have access.
Allow list: at least one entry Deny list: at least one entry	IP addresses included in both or neither of the lists have access.	IP addresses included in both or neither of the lists do not have access.

At all times, at least one IP address must be allowed access to the HCP S Series Management Console, either explicitly or due to the setting for allow-list and deny-list handling.

You cannot add the IP address from which you're currently accessing an S Series Node to the deny list for the interface you're using. Similarly, you cannot change the setting for allow-list and deny-list handling for that interface so that access would be denied from that IP address.

SSL server certificates

For HTTPS access to an S Series Node through the Management Console, management API, or S3 compatible API, the S Series Node must have an SSL server certificate. To meet this need, each S Series Node comes with a self-signed certificate. The common name in this certificate is *.*node-domain-name*, where *node-domain-name* is the configured domain name of the S Series Node.

When an SSL server certificate is close to expiring, the S Series Node displays an alert notifying you of the upcoming expiration. You can use the HCP S Series Management Console or management API to generate a new self-signed certificate. The new certificate has an expiration date of five years from the date on which it was generated.

After you generate a new SSL server certificate for the S Series Node, clients such as HCP that had accepted the old certificate must now accept the new certificate to be able to continue accessing the S Series Node.

Security settings

If you have the security role, you can use the HCP S Series Management Console or management API to control various types of access to an S Series Node.

Ping and SSH

You can allow or prevent the use of these services on the S Series Node server modules:

- **Ping** — Enabling this service lets you use **ping** to check network connectivity to the server modules. This service is enabled by default.
- **SSH login by authorized service providers on the management port** — Enabling this service facilitates troubleshooting when you request support.

  With this service enabled, service providers can use SSH to access the S Series Node on the access and management networks. Disabling this service disables SSH access on the access and management networks.

  This service is enabled by default.

User account and Management Console properties

You can configure these properties that affect S Series Node user accounts and HCP S Series Management Console sessions:

- The minimum password length. Valid values are integers in the range 6 through 64. The default is 6.

  The longer the minimum password length, the stronger user account passwords are likely to be. Encourage users to use a mix of uppercase and lowercase letters, numbers, and special characters to create even stronger passwords.

- The number of days passwords are valid before they automatically expire. Valid values are integers in the range 3 through 180. The default is 90.

Note: An HCP system that's configured to use storage on an S Series Node automatically changes the password for its S Series Node user account every 30 days. If you set the password expiration interval on the S Series Node to fewer than 30 days, the HCP system won't be able to access the S Series Node after the specified number of days have passed. To ensure that the HCP system doesn't lose access to the S Series Node, turn off automatic password expiration for the S Series Node user account created by HCP.

- The consecutive number of times a user can specify an incorrect password before the user account is automatically disabled. Valid values are integers in the range 3 through 999. The default is 10.

  This limit applies both to attempts to log in to the HCP S Series Management Console and to attempts to access the S Series Node through the management API.

  If a user account with the security role is automatically disabled due to an incorrect password, the account is automatically re-enabled after one hour.

- The number of minutes an HCP S Series Management Console session can be inactive before it times out. Valid values are integers in the range 5 through 720. The default is 10.

DNS servers and time servers

If you have the administrator or service role, you can use the HCP S Series Management Console or management API to modify DNS server settings and time server settings.

DNS servers

Optionally, you can make up to three DNS servers known to an S Series Node. You identify each DNS server by its IP address.

You can choose the network (access or management) to be used for communication between the S Series Node and the DNS servers you specify. The default is the access network.

The S Series Node uses the selected network in the IP mode in which the network is configured. If the network is configured for IPv6, you can choose to use the primary or secondary IPv6 gateway. If you choose to use the secondary IPv6 gateway and this gateway is not configured, communications between the S Series Node and the DNS servers fail.

For the S Series Node to communicate with the specified DNS servers, the IP mode of the selected network must match the IP mode of the DNS server IP addresses.

The S Series Node issues an alert if communications to the DNS servers fail.

Time servers

S Series Nodes use external time servers to set and maintain their internal clock times. An S Series Node always needs to know how to access at least one external time server.

You can specify up to three external time servers for use by an S Series Node. You identify each time server by its IP address. You cannot use DNS hostnames to identify time servers to an S Series Node.

The time servers you specify should be the same time servers as those that are used by the clients accessing the S Series Node.

Regardless of the time servers used, S Series Node time is always expressed in UTC.

You can choose the network (access or management) to be used for communication between the S Series Node and the time servers you specify. The default is the access network.

The S Series Node uses the selected network in the IP mode in which the network is configured. If the network is configured for IPv6, you can choose to use the primary or secondary IPv6 gateway. If you choose to use the secondary IPv6 gateway and this gateway is not configured, communications between the S Series Node and the time servers fail.

For the S Series Node to communicate with the specified time servers, the IP mode of the selected network must match the IP mode of the time server IP addresses.

Changing the list of time servers used by an S Series Node causes the S Series Node to restart.

The S Series Node issues an alert in the event of a time synchronization error.

Event log

An S Series Node maintains a log that contains messages about events that occur on the S Series Node. You can view the event log in the HCP S Series Management Console. You can also use the S Series Node management API to retrieve the contents of the log.

The event times associated with log messages are in UTC.

Security event messages report actions that require the security role (such as the creation of a user account). These messages also report attempts to log in to the HCP S Series Management Console with an invalid username or to use the HCP S Series management API with an invalid username. Only users with the security role can see messages about security events.

Alerts

Alerts contain information about the current state of the S Series Node. Typically, an alert requires you to take an action.

The HCP S Series Management Console displays the alerts that are currently in effect at the top of the **Dashboard** page and at the tops of pages that contain information relevant to the alert. You can also use the S Series Node management API to retrieve the alerts that are currently in effect.

Alerts are triggered by events. However, although messages about events are always logged at the time the event occurs, some alerts may not be issued until up to five minutes after the triggering event occurs. Similarly, some alerts may persist up to five minutes past the resolution of the triggering event.

Syslog logging

You can have the S Series Node send event log messages to one or more syslog servers as the messages are logged. You can then use tools in your syslog environment to perform functions such as sorting, querying, and forwarding the messages.

To configure syslog logging, you can use the HCP S Series Management Console or management API.

Configuring syslog logging

You can specify up to ten syslog servers. You identify each one by its IP address (optionally, with an appended port number). If you specify multiple servers, the S Series Node sends each message to all of them.

When you specify syslog servers, you also need to specify the syslog local facility to which the event messages will be directed. This selection applies to all the syslog servers you specify.

Specifying which messages to send

By default, the S Series Node sends messages about all events except security events. However, you can choose to have security event messages sent to the syslog servers along with other event messages. For information about security events, see "Event log" on the previous page.

You can limit the volume of messages sent to the syslog servers in these ways:

- By sending only messages about major events. Major events are those that are displayed on the **Dashboard** page of the HCP S Series Management Console.

- By setting a minimum severity level of **WARNING** or **ERROR** for the messages to be sent.

Selecting a network

You can choose the network (access or management) to be used for communication between the S Series Node and the syslog servers you specify. The default is the access network. For information about networks, see "Networking" on page 29.

The S Series Node uses the selected network in the IP mode in which the network is configured. If the network is configured for IPv6, you can choose to use the primary or secondary IPv6 gateway. If you choose the secondary IPv6 gateway and this gateway is not configured, communications between the S Series Node and the syslog servers fail.

For the S Series Node to communicate with the specified syslog servers, the IP mode of your network selection must match the IP mode of the syslog server IP addresses.

Note: If all these conditions are true, the S Series Node sends messages to the syslog servers over both the access and management network:

- The access and management networks have different IP modes.
- The syslog configuration specifies two or more syslog servers.
- At least one specified syslog server has an IPv4 address, and at least one specified syslog server has an IPv6 address.

Testing syslog logging

After configuring syslog logging and saving your settings, you can test the configuration by having the S Series Node send a test message to the specified syslog servers. For the test message to be sent, the minimum severity level must be set to **NOTICE**, which allows all messages to be sent, and **Send major events only** must be enabled.

Internal logs

S Series Nodes maintain internal logs that record the status and activity of various components of the HCP S Series software. If a problem occurs with the S Series Node, the internal logs can assist support personnel in diagnosing and resolving the problem.

If you have the administrator, monitor, security, or service role, you can use the HCP S Series Management Console or management API to insert comments into the S Series Node internal logs. You can use this capability, for example, to note unusual events that occur on the S Series Node. Comments can later assist support personnel in understanding the symptoms that indicate a possible problem. Comments can also assist support personnel in determining when a problem started.

To help with troubleshooting, if you have the administrator or service role, you can download the internal logs and send them to your HCP support center. You can use the HCP S Series Management Console or management API to download the logs. For ease of handling, the S Series Node downloads the logs into a single packed file. Neither this file nor the logs themselves are encrypted.

An S Series Node generally keeps internal logs for at least 120 days. However, it keeps the logs for a shorter time period if insufficient space is available for them. You can download the logs for any length of time within the period for which logs exist. When downloading the logs, be sure to include all the days on which you observed issues with the S Series Node.

Hotfixes

At times, you may need to apply a hotfix to an S Series Node. A hotfix is an update to the HCP S Series software that resolves a particular problem. Typically, hotfixes are applied only to S Series Nodes that are experiencing that problem. If possible, hotfixes are applied while the S Series Node is running, with no loss of functionality during the process.

To apply a hotfix, you first upload a hotfix file. Then you apply the uploaded hotfix. You can perform this procedure either in the HCP S Series Management Console or by using the HCP S Series management API.

Note: To upload a hotfix file and apply the hotfix, you must be an authorized service provider.

Hotfix files

A **hotfix file** contains the files necessary for applying a hotfix. Hotfix files are a type of update file.

HCP S Series hotfix files are named `HCPS_Hotfix_`*`release-number`*`_HF`*`hotfix-number`*`.bin` (for example, `HCPS_Hotfix_3.0.0.2_HF0001.bin`).

Considerations for applying hotfixes

These considerations apply to applying hotfixes to S Series Nodes:

- Before you start the procedure to apply a hotfix, both S Series Node server modules must be running and healthy.
- When you upload a hotfix file, the file overwrites any previously uploaded hotfix file.
- After uploading a hotfix file, you cannot apply the hotfix while the internal logs are being downloaded or a maintenance procedure is in progress.
- While a hotfix is being applied, you can make changes to the S Series Node configuration. However, most configuration changes don't take effect until the hotfix application is complete.
- Hotfixes are applied on one server module at a time. While a hotfix is being applied on one server module, all S Series Node processing occurs on the other server module.
- When a hotfix application finishes on the first server module, that server module is automatically rebooted. When the reboot is complete, the hotfix application automatically starts on the second server module, and processing fails over from the second server module to the first server module. While this failover is in progress, the HCP S Series Management Console may be briefly unavailable.

  When the hotfix application is complete on the second server module, that server module is automatically rebooted. When the reboot is complete, processing is again distributed across both server modules.
- While a hotfix is being applied on a server module, you cannot use the physical IP address of that module to access the HCP S Series Management Console, make management API requests, or perform data access operations.
- If you accessed the HCP S Series Management Console by using the physical IP address of the second server module while the hotfix was being applied on the first server module, when failover occurs, you lose

your connection to the S Series Node. At that point, you need to access the Management Console again, this time using the S Series Node domain name, a virtual IP address, or the physical IP address of the first server module.

- If an error occurs during the apply step of a hotfix application, you can try restarting that step. If an error occurs again, do not try to restart the step a second time. Instead, contact your HCP support center for help.

Hardware maintenance

For certain HCP S Series Node hardware maintenance procedures, you start and end the procedure either in the HCP S Series Management Console or by using the HCP S Series management API. These procedures are:

- Adding, removing, or replacing data and database drives
- Adding, removing, or replacing an enclosure

Note: To perform a hardware maintenance procedure, you must be an authorized service provider. Customers are not allowed to perform these activities by themselves.

Service providers: For more information about hardware maintenance procedures, see the HCP S11 and S31 Node maintenance documentation.

You can also use either the Management Console or management API to:

- Restart one or both server modules.
- Power off one or both server modules. Powering off both server modules effectively shuts down the S11 or S31 Node.
- Power on an individual server module. You can do this only if the other server module is currently powered on.
- Turn beaconing on or off for an enclosure, server module, or I/O module. When beaconing is on for a component, an LED on the component blinks, enabling the component to be easily identified in the data center.

For you to perform the activities listed above, your user account must include the administrator or service role.

2

Introduction to the HCP S Series management API

The HCP S Series management API is a RESTful HTTP interface to the administrative functions of an S Series Node. Using this API, you can perform tasks such as creating user accounts, modifying S Series Node networks, enabling syslog logging, and viewing current alerts. You can also use the management API to manage maintenance procedures such as replacing data and database drives and updating the HCP S Series software.

Each aspect of an S Series Node that you can work with by using the management API is referred to as a resource. Resources have properties that provide information about them. You use HTTP requests to manipulate resources. Some requests for resources take query parameters that qualify the request.

Note: The HCP S Series management API examples in this book use cURL, which is freely available open-source software. You can download cURL from curl.haxx.se.

What you can do with the management API

The HCP S Series management API lets you work with these aspects of an S Series Node:

- **User accounts** — You can:
  - Create, modify, and delete user accounts
  - Retrieve information about an individual user account
  - Retrieve a list of all user accounts defined on the S Series Node
  - Change the password for your user account
  - Generate the Hitachi API for Amazon S3 (the S3 compatible API) access key and secret key for your user account
- **Buckets** — You can:
  - Create and delete buckets
  - Change the owner of a bucket
  - Retrieve information about an individual bucket
  - Retrieve a list of all buckets defined on the S Series Node
  - Retrieve information about bucket usage
  - Retrieve a list or count of the irreparable objects in a bucket
- **Irreparable objects** — You can retrieve a list or count of the irreparable objects stored on the S Series Node.
- **Networks** — You can:
  - Modify the S Series Node access, management, and server interconnect networks
  - Retrieve information about an individual network
  - Retrieve and modify the setting for management network monitoring

  - Retrieve a list of the networks defined on the S Series Node

- **Transport Layer Security (TLS)** — You can retrieve and modify the minimum TLS version setting.
- **S Series Node identification** — You can:
  - Retrieve and modify the S Series Node domain name
  - Retrieve the S Series Node serial number, software version, and product model
- **S Series Node licensing** — You can retrieve the current S Series Node licensing status.
- **Management Console configuration** — You can retrieve and modify the configuration of the HCP S Series Management Console.
- **Management API configuration** — You can retrieve and modify the configuration of the HCP S Series management API.
- **Data access protocols** — You can:
  - Retrieve and modify the configuration of the Hitachi API for Amazon S3 (the S3 compatible API)
  - Retrieve a list of the data access protocols supported by the S Series Node
- **SSL server certificates** — You can:
  - List the SSL server certificates that are stored on the S Series Node
  - Generate a new self-signed SSL server certificate for the S Series Node
- **Security** — You can retrieve and modify S Series Node security settings.
- **DNS servers** — You can:
  - Retrieve or modify the list of the DNS servers used by the S Series Node

  - Select the network to be used for communication with the DNS servers

- **Time servers** — You can:
  - Retrieve or modify the list of the time servers used by the S Series Node
  - Select the network to be used for communication with the time servers
- **Syslog logging** — You can retrieve and modify the syslog logging settings for the S Series Node.
- **Event log** — You can retrieve a list of messages written to the S Series Node system log.
- **Alerts** — You can retrieve a list of the alerts that currently apply to the S Series Node.
- **S Series Node internal logs** — You can download the S Series Node internal logs.
- **Metrics** — You can retrieve statistics about storage capacity and usage, data access, and object repair.
- **S Series Node status** — You can retrieve complete information about the status of the S Series Node or a subset of that information.
- **Update** — You can:
  - Retrieve the history of the HCP S Series software on the S Series Node, starting from the most recent installation or reinstallation of the software
  - Apply a hotfix to the S Series Node
- **Hardware** — You can:
  - Retrieve all hardware-related information with a single request
  - Reboot or shut down an individual server module or both server modules
  - Power on an individual server module

  - Turn beaconing on or off for an enclosure, server module, or I/O module
  - Manage maintenance procedures
  - Retrieve a list of previous maintenance procedures

- **Management API versions** — You can:
  - Retrieve a list of the management API versions supported by the S Series Node
  - Check whether the S Series Node supports a specific management API version

Who can use the management API

To use the HCP S Series management API, you need a user account that's defined on the S Series Node you're accessing. What you can do with the API depends on the roles associated with that user account. The permissions granted by each role have the same effect with the management API as they do in the HCP S Series Management Console.

For anyone to be able to use the HCP S Series management API, the API must be enabled on at least one network in the HCP S Series Management Console.

Resources and properties

Each aspect of an S Series Node that you can manage independently with the HCP S Series management API is called a **resource**. Examples of resources are user accounts, networks, and hardware.

Some resources have subresources. For example, hs3 is a subresource of the protocols resource.

Some subresources are actions. For example, generating a new self-signed SSL server certificate is a subresource of the configuration resource.

To identify a resource, you use a URL. For example, this URL identifies the server interconnect network for the S Series Node for which the domain name is s-node-1.example.com:

```
https://s-node-1.example.com:9090/mapi/configuration/networks/builtin/interconnect
```

You also use URLs to identify lists of resources. For example, this URL identifies the list of user accounts defined on the same S Series Node as above:

```
https://s-node-1.example.com:9090/mapi/user_accounts
```

Most resources have an unordered set of one or more properties. The properties for a resource describe that resource. For example, the properties for a bucket are bucketName, description, owner, creationTime, and bucketID.

Properties have data types. The data type for a property can be string, integer, short, long, Boolean, timestamp, array, or object (that is, another set of properties). For example, the username property for the user account resource has a data type of string. The roles property for the user account resource has a data type of array.

Supported methods for the management API

The HCP S Series management API supports the HTTP methods listed in the table below.

Method	Description
PUT	Creates a resource
GET	Retrieves information about an individual resource or retrieves a list of resources of a given type
HEAD	Checks whether a particular resource exists
POST	Modifies a resource or performs an action on a resource.
DELETE	Deletes a resource

Each request you submit to the management API can work on only one resource or, for a list, one type of resource. So, for example, you cannot use a single **PUT** request to create two user accounts.

Management API input and output format

When you create or modify a resource through the HCP S Series management API, you use JSON to specify the resource properties in the **PUT** or **POST** request body. In the request itself, you include the HTTP

Content-Type header with a value of **application/json** to indicate the format of the request body.

The response bodies returned by management API requests are also in JSON format.

All responses returned through the management API are UTF-8 encoded. The request bodies you create for input to the API must also be UTF-8 encoded.

Management API query parameters

Some HCP S Series management API requests take query parameters. Query parameters are appended to a resource URL following a question mark (?). Multiple parameters are joined by ampersands (&).

The following considerations apply to management API query parameters:

- Query parameter names are case sensitive.
- If you specify an invalid value for a required or optional query parameter, the S Series Node returns a status code of 400 (Bad Request).
- If you omit a required query parameter, the S Series Node returns a status code of 400 (Bad Request).
- If you specify a query parameter that’s not valid for the request, the S Series Node returns a status code of 400 (Bad Request).
- For query parameters that take a Boolean value, the valid values are **true** and **false**. These values are not case sensitive.

prettyprint query parameter

The **prettyprint** query parameter causes the JSON returned in a response body to be formatted for readability. For example, with the **prettyprint** parameter, the returned JSON for a user account looks like this:

```
{
  "username": "lgreen",
  "description": "Storage management group manager with security privileges",
  "roles": [
    "security",
    "admin"
  ],
  "fullName": "Lee Green",
  "forcePasswordChange": false,
  "enabled": true
}
```

Without the **prettyprint** parameter, the returned JSON looks like this:

```
{"username":"lgreen","description":"Storage management group manager with security
privileges","roles":["security","admin"],"fullName":"Lee
Green","forcePasswordChange":
false,"enabled":true}
```

When the **prettyprint** parameter is used with a request that does not return a response body, the parameter is ignored.

The **prettyprint** parameter increases the time required to process a request. Therefore, you should use this parameter only for testing purposes and not in production applications.

Management API error response body

When a management API request results in an error, the S Series Node returns information about the error in an error response body. Error response bodies are formatted as JSON and can contain one or more error messages.

The JSON in error response bodies is formatted for readability, as in this example:

```
{
  "errorMessages": [
    {
      "message": "Encountered missing or empty required parameter password"
    }
  ]
}
```

X-HCPS-API-VERSION request and response headers

Each HCP S Series management API request must include an X-HCPS-API-VERSION header that specifies which version of the API the S Series Node should use when processing the request. For example, here's a request for a list of user accounts that tells the S Series Node to use the 3.0.0 version of the management API to process the request:

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic bGdyZWVuOkxncmVlbjEh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts?prettyprint"
```

Each HCP S Series management API response also includes an X-HCPS-API-VERSION header. This header specifies the management API version that the S Series Node actually used when processing the request.

Additionally, each HCP S Series management API response includes an X-HCPS-SUPPORTED-API-VERSIONS header. This header specifies the currently supported versions of the management API.

For example, here are the headers returned in response to the request shown above:

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 181
```

HTTP Server response header

Each HCP S Series management API response includes the HTTP Server header. This header identifies the version of the HCP S Series software currently running on the S Series Node that processed the request.

The value of the Server header is always "HCP S Series" followed by the software version number, like this:

```
Server: HCP S Series/3.0.0.2
```

X-HCPS-Domain-Name response header

Each HCP S Series management API response includes the X-HCPS-Domain-Name header. The value of this header is the domain name of the S Series Node that processed the request.

Here's a sample X-HCPS-Domain-Name header:

```
X-HCPS-Domain-Name: s-node-1.example.com
```

For information about S Series Node domain names, see "HCP S Series Node identification" on page 41

X-HCPS-Server-Module-Number response header

Each HCP S Series management API response includes the X-HCPS-Server-Module-Number header. This header identifies the server module that processed the request.

The value of the X-HCPS-Server-Module-Number header is the server module number, like this:

```
X-HCPS-Server-Module-Number: 1
```

X-HCPS-ErrorMessage response header

In some cases, when a management API request contains a query parameter error, the S Series Node returns information about the error as the value of an X-HCPS-ErrorMessage response header. If the request results in an error response body, the value of the X-HCPS-ErrorMessage header is the same as the message in that response body.

3

Management API access and authentication

With the HCP S Series management API, resources are represented by URLs. Each management API request you make must specify one such URL. Each request must also include the credentials for the user account you're using to access HCP through the management API.

This chapter describes resource URLs and explains how to include account credentials in a management API request.

URLs for S Series Node access through the management API

With the HCP S Series management API, you use one of these formats to specify the resource URL in a request:

`https://mapi.`*`node-domain-name`***`:9090/mapi/`***`resource-identifier`*

`https://`*`ip-address`***`:9090/mapi/`***`resource-identifier`*

In these formats:

- *`node-domain-name`* is the fully qualified domain name of the S Series Node, as configured in DNS. When you use a URL with the domain name, the DNS response determines which server module the request is directed to.
- *`ip-address`* is either of:
  - The access network virtual IP address of either server module in the S Series Node
  - The management network IP address of either server module in the S Series Node

  In either case, the applicable network must be enabled in the management API configuration.

Here's an example of a resource URL that uses a domain name:

```
https://mapi.s-node-1.example.com:9090/mapi/user_accounts/lgreen
```

Here's an example of a resource URL that uses an IPv4 address:

```
https://10.0.0.4:9090/mapi/configuration/console
```

Here's an example of a resource URL that uses an IPv6 address:

```
https://[2001:0db8::101]:9090/mapi/configuration/networks/builtin/access
```

When you use an IPv6 address, you need to enclose the address in square brackets.

When you use the S Series Node domain name or an access network virtual IP address, if the server module to which the request is directed is unavailable, the request is automatically redirected to the other server module. If you use a management network IP address, if the server module to which the request is directed is unavailable, the request fails.

If a client uses a `hosts` file to map S Series Node hostnames to IP addresses, the client system has full responsibility for converting any hostnames to IP addresses. In a `hosts` file, you can map any number of IP addresses to a single hostname. The way the client uses multiple IP address mappings for a single hostname depends on the client platform. For information about how your client handles these mappings, see your client documentation.

Regardless of whether you specify the domain name or an IP address in the resource URL, the management API must be configured to allow access from your client IP address.

S Series Nodes can support resource URLs that use HTTP without SSL security (requires port number 9091). However, for security reasons, client requests for access through the management API should always use HTTPS, not HTTP, in the URL.

Note: HTTP access to the S Series Node through the management API without SSL security is possible only if the management API is explicitly configured to allow it.

If the S Series Node uses a self-signed SSL server certificate and the resource URL in a management API request specifies HTTPS, not HTTP, the program submitting the request must include instructions either to trust the SSL certificate or not to perform SSL certificate verification. If the resource URL uses an IP address, the only option is not to perform SSL certificate verification.

With cURL, you specify the instruction not to perform SSL certificate verification by including the **-k** or **--insecure** option in the request command line.

For information about management API configuration, see "Management API configuration" on page 43. For information about SSL server certificates, see "SSL server certificates" on page 46.

Considerations for resource URLs

The following considerations apply to URLs in HCP S Series management API requests.

Case sensitivity

A management API resource URL must always include the *mapi* interface identifier. Both this identifier and the resource identifier in the URL are case sensitive.

URL length

The portion of a resource URL that follows *mapi*, excluding any appended query parameters, is limited to 4,095 bytes. If a request includes a URL that violates that limit, the S Series Node returns a status code of 414 (Request URI Too Large).

Percent-encoding for special characters

Some characters have special meaning when used in a URL and may be interpreted incorrectly when used for other purposes. To avoid ambiguity, percent-encode the special characters listed in the table below.

Character	Percent-encoded value
Space	%20
Tab	%09
New line	%0A
Carriage return	%0D
+	%2B
%	%25
#	%23
?	%3F
&	%26

Percent-encoded values are not case sensitive.

Quotation marks with URLs in command lines
When using the management API, you work in a Windows, Unix, or Mac OS X shell. Some characters in the commands you enter may have special meaning to the shell. For example, the ampersand (&) used in URLs to join multiple query parameters may indicate that a process should be put in the background.

To avoid the possibility of the Windows, Unix, or Mac OS X shell misinterpreting special characters in a URL, always enclose the entire URL in double quotation marks.

Management API authentication

To access an S Series Node through the management API, you need to provide credentials in the form of a username and password. You need to provide credentials with every management API request. If you do not provide credentials or provide invalid credentials, the S Series Node responds with a 401 (Unauthorized) error message.

To provide credentials in a management API request, you use the HTTP Authorization header. The value of this header is **Basic** followed by an authentication token. The authentication token is the Base64 encoding of the username and password, separated by a colon (:).

For example, here's the Authorization header for credentials that consist of the username *lgreen* and the password *Lgreen1!*:

```
Authorization: Basic bGdyZWVuOkxncmVlbjEh
```

The GNU Core Utilities include the **base64** command, which converts text to a Base64-encoded value. With this command, a line like this creates the authentication token to use in the Authorization header:

```
echo -n username:password | base64
```

For example, this line creates the authentication token used in the sample Authorization header shown above:

```
echo -n lgreen:Lgreen1! | base64
```

For more information about the GNU Core Utilities, see http://www.gnu.org/software/coreutils.

Other tools that generate Base64-encoded values are available for download on the web. For security reasons, do not use interactive public web-based tools to generate these values.

Management API resources

The HCP S Series management API has many main types of resources. These resources correspond to entities such as buckets, the HCP S Series Management Console, and S Series Node status.

A **resource identifier** is the portion of a resource URL that follows the mapi interface identifier. Each main type of resource is associated with a set of one or more resource identifiers, each of which identifies one of these:

- A list of resources of that type
- An instance of that type of resource
- A subresource of the resource
- An action to be performed on the resource or on a subresource of the resource

For each main type of resource, this chapter contains a table of the associated resource identifiers. For each resource identifier, the table shows:

- The methods supported by the resource
- The use for each supported method
- The user account roles that allow the user to use each method
- Any additional notes about the resource

For an introduction to resources, see "Resources and properties" on page 59. For information about resource URLs, see "URLs for S Series Node access through the management API" on page 66.

Alerts resource

The alerts resource lets you list the alerts that are currently in effect for the S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/alerts			
GET	Retrieve a list of current alerts	Administrator Monitor Service Security	Alerts about security-related conditions are returned only when the request is made by a user with the security role. These alerts are not returned when the request is made by a user without the security role.

For more information about alerts, see "Alerts" on page 49.

Beaconing resources

Many S11 and S31 Node components have LEDs that can serve as beacons. When lit solid or blinking (depending on the component), the beaconing LED lets you easily identify the applicable component in the data center.

Beaconing resources let you turn beaconing on and off for enclosures, server modules, and I/O modules. The table below provides information about these resources.

Method	Use	Roles	Notes
/hardware/beacon/enclosure/*enclosure-number*			
POST	Turn beaconing on or off for an enclosure	Administrator Service	For information about the query parameters used for turning beaconing on and off, see "/hardware/beacon/enclosure/enclosure-number query parameters" on page 295.

(Continued)

Method	Use	Roles	Notes
/hardware/beacon/enclosure/*enclosure-number*/iom/*io-module-id*			
POST	Turn beaconing on or off for an I/O module	Administrator Service	For information about the query parameters used for turning beaconing on and off, see "/hardware/enclosure/enclosure-number/iom/io-module-id query parameters" on page 296.
/hardware/beacon/server_module/*server-module-number*			
POST	Turn beaconing on or off for a server module	Administrator Service	For information about the query parameters used for turning beaconing on and off, see "/hardware/beacon/server_module/server-module-number query parameters" on page 297.

Bucket resources

Bucket resources let you retrieve a list of existing buckets and add, retrieve information about, modify, and delete buckets. The table below provides information about these resources.

Method	Use	Roles	Notes
/buckets			
PUT	Create a bucket	Administrator	
GET	Retrieve a list of existing buckets	Administrator Monitor Service	The buckets are listed in alphabetical order by bucket name. For information about the query parameters used to limit the bucket list, see "Managing resource lists" on page 91.
/buckets/*bucket-name*			
GET	Retrieve information about an existing bucket	Administrator Monitor	

(Continued)

Method	Use	Roles	Notes
HEAD	Check whether a bucket exists	Administrator Monitor	If the bucket exists, the S Series Node returns a status code of 200 (OK). If the bucket does not exist, the S Series Node returns a status code of 404 (Not Found). If you don't have permission to perform the request, the S Series Node returns a status code of 403 (Forbidden).
POST	Modify a bucket	Administrator	
DELETE	Delete a bucket	Administrator	You can delete only empty buckets (that is, buckets that don't contain any objects).

For more information about buckets, see "Buckets" on page 27 and "Considerations for working with buckets" on page 29.

Certificate resources

Certificates resources let you retrieve information about and generate SSL server certificates. The table below provides information about these resources.

Method	Use	Roles	Notes
/configuration/certificates/system			
GET	Retrieve information about the SSL server certificate currently in use by the S Series Node	Administrator Monitor Service	
/configuration/certificates/system/generate			
POST	Generate a new self-signed SSL server certificate for the S Series Node	Administrator	

For more information about SSL server certificates, see "SSL server certificates" on page 46.

Console resource

The console resource lets you retrieve and modify the configuration of the HCP S Series Management Console. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/console			
GET	Retrieve the Management Console configuration	Administrator Monitor Service	
POST	Modify the Management Console configuration	Administrator Service	

For more information about Management Console configuration, see "Management Console configuration" on page 42.

DNS resource

The DNS resource lets you retrieve and modify the DNS server configuration for the S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/dns			
GET	Retrieve the DNS server configuration	Administrator Monitor Service	
POST	Modify the DNS server configuration	Administrator Service	

For more information about DNS server configuration, see "DNS servers and time servers" on page 48.

Events resource

The events resource lets you list the contents of the S Series Node event log. The table below provides information about this resource.

Method	Use	Roles	Notes
/events			
GET	Retrieve a list of the messages in the event log	Administrator Monitor Service Security	Event messages are listed in descending order by date and time. Messages about security-related events are returned only when the request is made by a user with the security role. For information about the required and optional query parameters used to limit the event message list, see "/events query parameters" on page 179.

For more information about events, see "Event log" on page 49.

Hardware resource

The hardware resource lets you retrieve complete information about the hardware used in the S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/hardware			
GET	Retrieve complete information about the S Series Node hardware	Administrator Monitor Service	

For information about S11 and S31 Node hardware, see "HCP S11 and S31 Node hardware components" on page 19.

Identification resource

The identification resource lets you retrieve and modify information that identifies the S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/ident			
GET	Retrieve identifying information about the S Series Node	Administrator Monitor Service	
POST	Modify the S Series Node domain name	Administrator Service	

For more information about S Series Node identification, see "HCP S Series Node identification" on page 41.

Irreparables resources

Irreparables resources let you retrieve a list of and get a count of the irreparable objects stored on the S Series Node. The table below provides information about these resources.

Method	Use	Roles	Notes
/buckets/*bucket-name*/irreparables			
GET	Retrieve a list of irreparable objects stored in a specified bucket	Administrator Monitor Service	For information about the query parameters used to limit the irreparable object list, see "Managing resource lists" on page 91.
HEAD	Retrieve a count of the irreparable objects stored in a specified bucket	Administrator Monitor Service	The count of irreparable objects is returned in the X-HCPS-Irreparable-Count header.
/system/irreparables			
GET	Retrieve a list of irreparable objects stored on the S Series Node	Administrator Monitor Service	For information about the query parameters used to limit the irreparable object list, see "Managing resource lists" on page 91.

(Continued)

Method	Use	Roles	Notes
HEAD	Retrieve a count of the irreparable objects stored on the S Series Node	Administrator Monitor Service	The count of irreparable objects is returned in the X-HCPS-Irreparable-Count header.

License resource

The license resource lets you retrieve the license status for the S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/system/license			
GET	Retrieve information about the S Series Node license status	Administrator Monitor Security Service	

For more information about S Series Node licensing, see " Licensing" on page 42.

Log resources

Log resources let you insert messages into and download the S Series Node internal logs. The table below provides information about these resources.

Method	Use	Roles	Notes
/system/logs/cancel			
POST	Reset the S Series Node to be ready for a new log download operation	Administrator Service	You cannot cancel a log download operation while the logs are being prepared for downloading.
/system/logs/download			
GET	Pack the prepared logs into a .zip file and download that file	Administrator Service	

(Continued)

Method	Use	Roles	Notes
/system/logs/mark			
POST	Insert a comment into the internal logs	Administrator Monitor Security Service	For information about the query parameter used to specify the message to be inserted, see "/system/logs/mark query parameter" on page 381.
/system/logs/prepare			
POST	Prepare the internal logs for download	Administrator Service	This **POST** request starts a log download operation. For information about the query parameters used to specify the time period for the logs to be downloaded, see "/system/logs/prepare query parameters" on page 382.
/system/logs/status			
GET	Retrieve the status of the current log download operation	Administrator Service	

For more information about the S Series Node internal logs, see "Internal logs" on page 51. For instructions on using the management API to download the internal logs, see "Downloading the internal logs" on page 426.

Maintenance resources

Maintenance resources let you perform one of these hardware maintenance procedures: add drives, remove drives, replace drives, and replace enclosure. The table below provides information about these resources.

Note: To perform a hardware maintenance procedure, you must be an authorized service provider. Customers are not allowed to perform these activities by themselves.

Method	Use	Roles	Notes
hardware/maintenance			
POST	Start a maintenance procedure	Service	The request body specifies the type of procedure to start.
hardware/maintenance/active			
GET	Retrieve a list of active maintenance procedures	Administrator Monitor Service	
hardware/maintenance/history			
GET	Retrieve a list of all completed or canceled maintenance procedures that have been performed on the S Series Node since the HCP S Series software was last installed	Administrator Monitor Service	The maintenance procedures are listed in descending order by procedure ID. For information about the query parameters used to limit the maintenance procedure history list, see "Managing resource lists" on page 91.
hardware/maintenance/*procedure-id*			
GET	Retrieve information about an active or past maintenance procedure	Administrator Monitor Service	
hardware/maintenance/*procedure-id*/cancel			
POST	Cancel an active maintenance procedure	Service	
hardware/maintenance/*procedure-id*/candidates			
GET	Retrieve a list of hardware components that are eligible to be targets of an active maintenance procedure	Service	
hardware/maintenance/*procedure-id*/complete			
POST	Complete an active maintenance procedure	Service	

(Continued)

Method	Use	Roles	Notes
hardware/maintenance/*procedure-id*/confirm			
POST	Specify how you want the S Series Node to handle previously used drives that were inserted into selected slots during an active add or replace drive procedure	Service	
hardware/maintenance/*procedure-id*/perform			
POST	Prepare the S Series Node for the physical portion of an active maintenance procedure	Service	
hardware/maintenance/*procedure-id*/select			
POST	Select the target components for an active maintenance procedure	Service	
hardware/maintenance/*procedure-id*/update			
POST	Add a note to an active maintenance procedure or replace an existing note	Service	
hardware/maintenance/*procedure-id*/verify			
POST	Verify that no errors have occurred in an active maintenance procedure	Service	

For more information about hardware maintenance procedures, see "Hardware maintenance" on page 54. For instructions on using the management API to perform hardware maintenance procedures, see "Performing a hardware maintenance procedure" on page 428.

Management API resource

The management API resource lets you retrieve and modify the configuration of the HCP S Series management API. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/mapi			
GET	Retrieve the management API configuration	Administrator Monitor Service	
POST	Modify the management API configuration	Administrator Service	

For more information about management API configuration, see "Management API configuration" on page 43.

Metrics resources

Metrics resources retrieve statistics about S Series Node usage. The table below provides information about these resources.

Method	Use	Roles	Notes
metrics/buckets			
GET	Retrieve statistics about bucket usage	Administrator Monitor Service	The buckets are listed in alphabetical order by bucket name. For information about the query parameters used to limit the bucket list, see "Managing resource lists" on page 91.
metrics/gateways			
GET	Retrieve statistics about data access protocol usage (that is, usage of the Hitachi API for Amazon S3 (the S3 compatible API))	Administrator Monitor Service	

(Continued)

Method	Use	Roles	Notes
metrics/protection			
GET	Retrieve a count of bytes under repair	Administrator Monitor Service	
metrics/system			
GET	Retrieve statistics about S Series Node capacity usage	Administrator Monitor Service	

Miscellaneous settings resource

The miscellaneous settings resource lets you control monitoring of the management network. The table below provides information about this resource.

Method	Use	Roles	Notes
/system/misc/settings/network/management/monitor			
GET	Retrieve the setting for management network monitoring	Administrator Monitor Service	
POST	Enable or disable management network monitoring	Administrator Service	For information about the query parameter used to enable and disable management network monitoring, see "/system/misc/settings/network/management/monitor query parameter" on page 387.

For more information about management network monitoring, see "Management network" on page 35.

Network resources

Network resources let you retrieve a list of the predefined S Series Node networks and retrieve and modify the configurations of those networks. The table below provides information about these resources.

Method	Use	Roles	Notes
/configuration/networks/builtin			
GET	Retrieve a list of the predefined S Series Node networks	Administrator Monitor Service	
/configuration/networks/builtin/access/ports			
GET	Retrieve a list of the connection expectations for the access network ports	Administrator Monitor Service	
/configuration/networks/builtin/access/ports/*port-number*			
GET	Retrieve the connection expectation for an access network port	Administrator Monitor Service	
POST	Modify the connection expectation for an access network port	Administrator Service	For information about the query parameter used to modify a connection expectation, see "/configuration/networks/builtin/access/ports/port-number query parameter" on page 143.
/configuration/networks/builtin/*network-name*			
GET	Retrieve information about a network	Administrator Monitor Service	
POST	Modify a network	Administrator Service	

For more information about networks, see "Networking" on page 29.

Power resources

Power resources let you power on and off and restart the S Series Node server modules. The table below provides information about these resources.

Method	Use	Roles	Notes
/hardware/power/node			
POST	Power off or restart both server modules	Administrator Service	For information about the query parameters used to specify a power option, see "/hardware/power/node query parameters" on page 356.
/hardware/power/*server-module-number*			
POST	Power on or off or restart a single server module	Administrator Service	For information about the query parameters used to specify power options, see "/hardware/power/server-module-number query parameters" on page 357.

Protocol resources

Protocol resources let you retrieve a list of supported data access protocols and retrieve and modify the configuration of the Hitachi API for Amazon S3 (the S3 compatible API). The table below provides information about these resources.

Method	Use	Roles	Notes
/configuration/protocols			
GET	Retrieve a list of supported data access protocols	Administrator Monitor Service	
/configuration/protocols/hs3			
GET	Retrieve the S3 compatible API configuration	Administrator Monitor Service	
POST	Modify the S3 compatible API configuration	Administrator Service	

For more information about data access protocols, see "Data access protocol configuration" on page 44.

Security resource

The security resource lets you control various aspects of access to an S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/security			
GET	Retrieve the S Series Node security settings	Security	
POST	Modify S Series Node security settings	Security	

For more information about S Series Node security settings, see " Security settings" on page 46.

Status resources

Status resources let you retrieve information about the state of the S Series Node. The table below provides information about these resources.

Method	Use	Roles	Notes
/system/status/full			
GET	Retrieve complete information about the state of the S Series Node, such as the current S Series Node status, hardware identification, and capacity and access statistics	Administrator Monitor Service	
/system/status/health			
GET	Retrieve brief information about the status of the S Series Node	Administrator Monitor Service	

Syslog resources

Syslog resources let you retrieve and modify the configuration of syslog logging. The table below provides information about these resources.

Method	Use	Roles	Notes
/configuration/syslog			
GET	Retrieve the syslog logging configuration	Administrator Monitor Service	
POST	Modify the syslog logging configuration	Administrator Service	
/configuration/syslog/test			
POST	Test whether the configured syslog servers can receive event log messages	Administrator Service	

For more information about syslog logging, see "Syslog logging" on page 50.

Time resource

The time resource lets you retrieve and modify S Series Node time settings. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/time			
GET	Retrieve the S Series Node time settings	Administrator Monitor Service	
POST	Modify the S Series Node time settings	Administrator Service	

For more information about time settings, see "DNS servers and time servers" on page 48.

TLS resource

The TLS resource lets you retrieve and modify the minimum TLS version for the S Series Node. The table below provides information about this resource.

Method	Use	Roles	Notes
/configuration/tls			
GET	Retrieve the minimum TLS version setting	Administrator Monitor Service	
POST	Modify the minimum TLS version setting	Administrator Service	

For more information about the minimum TLS version, see "Transport Layer Security (TLS)" on page 40.

Update resources

The update resource lets you apply hotfixes to the S Series Node and retrieve the history of the HCP S Series software on the S Series Node, starting from the most recent installation or reinstallation of the software. The table below provides information about these resources.

To apply a hotfix, you must be an authorized service provider. Customers are not allowed to perform this activity by themselves.

Method	Use	Roles	Notes
/system/update/apply			
POST	Update the HCP S Series software by applying a hotfix	Service	
/system/update/history			
GET	Retrieve the history of the HCP S Series software on the S Series Node, starting from the most recent installation or reinstallation of the software	Administrator Service	

(Continued)

Method	Use	Roles	Notes
/system/update/manifest			
GET	Retrieve information about the currently uploaded update file	Service	
/system/update/progress			
GET	Retrieve information about the progress of an update operation	Service	
/system/update/restart			
POST	Restart a failed update operation	Service	
/system/update/status			
GET	Retrieve information about the status of update operation on the S Series Node	Service	
/system/update/upload/software			
PUT	Upload and unpack an update file	Service	

For more information about applying hotfixes, see "Hotfixes" on page 52. For instructions on using the management API to apply a hotfix, see "Performing an update operation" on page 435.

User account resources

User account resources let you retrieve a list of existing user accounts and create, retrieve information about, modify, generate access and secret keys for, and delete user accounts. The table below provides information about these resources.

Method	Use	Roles	Notes
/user_accounts			
PUT	Create a user account	Security	

(Continued)

Method	Use	Roles	Notes
GET	Retrieve a list of existing user accounts	Administrator Security	The user accounts are listed in alphabetical order by username. For information about the query parameters used to limit the user account list, see "Managing resource lists" on the next page.
/user_accounts/*username*			
GET	Retrieve information about an existing user account	Security	
HEAD	Check whether a user account exists	Security	If the user account exists, the S Series Node returns a 200 (OK) status code. If the user account does not exist, the S Series Node returns a 404 (Not Found) status code. If you don't have permission to perform the request, the S Series Node returns a 403 (Forbidden) status code.
POST	Modify a user account	Administrator Monitor Security Service Data	Without the security role, you can modify only the password property for your own user account.
DELETE	Delete a user account	Security	
/user_accounts/*username*/access_key/generate			
POST	Generate the access key and secret key for a user account	Data	You can generate the access key and secret key only for your own user account. Be sure to save the access key and secret key returned by the **POST** request. They are not retrievable.

For more information about user accounts, see "User accounts" on page 20 and "Considerations for working with user accounts" on page 26.

Versions resource

The versions resource lets you retrieve information about the supported versions of the HCP S Series management API. The table below provides information about this resource.

Method	Use	Roles	Notes
/versions			
GET	Retrieve information about the supported versions of the management API	Administrator Monitor Security Service Data	
POST	Check whether a specific version of the management API is supported	Administrator Monitor Security Service Data	For information about the query parameter used to check management API version support, see "/versions POST query parameter and properties" on page 421.

Managing resource lists

Some management API requests return a list of resources as the value of a property with a data type of array. In the array, each listed resource is represented by a set of one or more properties.

For certain resources, you can use query parameters to limit the resources included in the response to a request for a resource list. This is especially useful if the number of such resources is very large.

The query parameters you can use are:

- For user accounts: **count**, **marker**, and **prefix**
- For buckets: **count**, **marker**, **prefix**, and **owner**
- For bucket metrics, irreparable objects, and maintenance procedure history: **count** and **marker**

The query parameters for each type of resource can be used alone or in combination with each other.

Other query parameters are available for managing lists of alerts and events. For information about those parameters, see "/alerts query parameters" on page 104 and "/events query parameters" on page 179.

For more information about query parameters, see "Management API query parameters" on page 61.

count query parameter

By default, when you use a management API request to retrieve a list of resources, the returned list includes one thousand of those resources (or fewer if fewer than one thousand satisfy the request criteria). To limit the number of resources in the returned list, you use the **count** query parameter. Valid values for this parameter are integers in the range zero through one thousand.

The **count** query parameter is valid with requests for lists of user accounts, buckets, bucket metrics, irreparable objects, and maintenance procedure history.

Here's a sample **curl** command that limits a list of user accounts to two:

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts?count=2
  &prettyprint"
```

The response body returned by a request for a list of resources includes the count property (bucketCount for bucket metrics). The value of this property is the number you specified as the value of the **count** parameter. If the request did not include the **count** parameter, the count property is included with a value of one thousand.

The response body also includes the isTruncated property. The value of this property is **true** if the returned list does not include all of the resources that satisfy the request criteria. Otherwise, the value is **false**.

Here's an example of a response to the **curl** command shown above:

```
{
  "marker": "",
  "prefix": "",
  "count": 2,
  "isTruncated": true,
  "username": [
    "admin",
    "hcpsrv-hcp-ma"
  ]
}
```

marker query parameter

Resource lists are ordered. By default, when you request a resource list, the returned list includes resources starting from the beginning of the full resource list (for example, user accounts starting with the alphabetically first username or maintenance procedures starting with the most recent procedure). To request a list of resources that starts with a resource that's not the first one in the full list, you use the **marker** query parameter.

The **marker** query parameter is valid with requests for lists of user accounts, buckets, bucket metrics, irreparable objects, and maintenance procedure history.

The **marker** parameter is useful when more than the requested number of resources satisfy the request criteria. If a request does not return the last resource in the full list, the response body includes the isTruncated property with a value of **true**. You can request the next part of the list by including the **marker** parameter in a new request. As the parameter value:

- In a request for a list of user accounts, buckets, or bucket metrics, specify the case-sensitive name of the last resource in the previously returned list.

- In a request for irreparable objects, specify the automatically generated case-sensitive string that identifies the last resource in the previously returned list. This string is returned with that list as the value of a property named nextMarker.

- In a request for the maintenance procedure history list, specify the ID of the last maintenance procedure in the previously returned list.

In any case, the list in the response body starts with the resource in the full list that follows the resource identified by the **marker** parameter.

The response body returned by a request for a list of user accounts, buckets, bucket metrics, irreparable objects, or maintenance procedure history includes the marker property. The value of this property is the character string you specified as the value of the **marker** parameter. If the request did not include the **marker** parameter, the marker property is included with no value for lists of user accounts, buckets, bucket metrics, and irreparable objects and a value of **2147483647** for the maintenance procedure history list.

Here's a sample **curl** command that requests a list of two user accounts starting with the first account with a username that alphabetically follows *lgreen*:

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts?marker=lgreen
  &count=2&prettyprint"
```

Here's an example of a response to the **curl** command shown above:

```
{
  "marker": "lgreen",
  "prefix": "",
  "count": 2,
  "isTruncated": true,
  "username": [
    "mwhite",
    "pblack"
  ]
}
```

prefix query parameter

You use the **prefix** query parameter to request a resource list that includes only resources with names that start with a specified case-sensitive character string. This parameter is valid only with requests for lists of user accounts and buckets.

Here's a sample **curl** command that limits a list of user accounts to those that start with the string *it-*:

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts?prefix=it-"
```

The response body returned by a request for a list of resources includes the prefix property. The value of this property is the character string you specified as the value of the **prefix** parameter. If the request did not include the **prefix** parameter, the prefix property is included with no value.

Here's an example of a response to the **curl** command shown above:

```
{
  "marker": "",
  "prefix": "it-",
  "count": 1000,
  "isTruncated": false,
  "username": [
    "it-pdgrey",
    "it-rbrown",
    "it-sgold"
  ]
}
```

owner query parameter

You use the **owner** query parameter to request a bucket list that includes only buckets that are owned by a specified user. For the value of the **owner** parameter, you specify the username for the applicable user account.

Here's a sample **curl** command that limits a list of buckets to those that are owned by the user with username *lgreen*:

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/buckets?owner=lgreen"
```

The response body returned by a request for a list of buckets includes the owner property. The value of this property is the username you specified as the value of the **owner** parameter. If the request did not include the **owner** parameter, the owner property is included with no value.

Here's an example of a response to the **curl** command shown above:

```
{
  "owner": "lgreen"
  "marker": "",
  "prefix": "",
  "count": 1000,
  "isTruncated": false,
  "bucketName": [
    "lg-testbucket-1",
    "lg-testbucket-2
  ]
}
```

5

Management API resource details

Some HCP S Series management API resource requests require a request body that specifies values for resource properties. Other management API resource requests return a response body that specifies values for resource properties. And, some management API resource requests take query parameters that qualify the request.

This chapter begins with general information about resource properties and information on using query parameters to manage resource lists. The chapter then describes the properties and query parameters, as applicable, for each management API resource. The chapter includes usage examples, each of which shows a sample **curl** command, a sample request body or response body, if applicable, and the corresponding HTTP request and response headers.

The resources in this chapter are presented alphabetically by resource identifier.

For information about resources and properties, see "Resources and properties" on page 59 and "Resource property usage" on the next page.

Resource property usage

When you use a management API request to create a resource, some properties are required in the request body, and some properties are optional. You need to specify a value for each required property. If you omit a required property, the S Series Node returns an error.

When you use a management API request to modify a resource, all properties that are supported for the request type are optional. If you omit a property, the current value of the property remains unchanged.

When you use a management API request to retrieve a resource, the response body includes all the properties for that resource.

When you create or modify a resource, the S Series Node returns an error if the request body includes:

- Properties that are not valid for the resource
- Properties that are not valid for the request type
- Properties that cannot be set with the user account used for the request

For more information about resource properties, see "Resources and properties" on page 59.

/alerts

With the /alerts resource, a **GET** request returns a response body that lists the current alerts for the S Series Node. You can use query parameters to limit the alerts included in the response body.

For more information about the /alerts resource, see "Alerts resource" on page 72.

/alerts properties

The table below describes the properties in /alerts response bodies. For information about the query parameters mentioned in the table, see "/alerts query parameters" on page 104.

Property name	Data type	Description	Notes
alerts	Array	Specifies a comma-separated list of the alerts that satisfy the request criteria. Each alert is represented by the properties described in the next table.	
scopes	Array	Specifies a comma-separated list of the values specified by the **scopes** query parameter included in the **GET** request. If the request did not include the **scopes** parameter, the value of this property is a comma-separated list of all the possible values for the **scopes** parameter.	
scopeRefs	Array	Specifies a comma-separated list of the values specified by the **scopeRefs** query parameter included in the **GET** request. If the request did not include the **scopeRefs** parameter, this property is not included in the response body.	
scopeSubRefs	Array	Specifies a comma-separated list of the values specified by the **scopeSubRefs** query parameter included in the **GET** request. If the request did not include the **scopeSubRefs** parameter, this property is not included in the response body.	

(Continued)

Property name	Data type	Description	Notes
severities	Array	Specifies a comma-separated list of the values specified by the **severities** query parameter included in the **GET** request. If the request did not include the **severities** parameter, the value of this property is a comma-separated list of all the possible values for the **severities** parameter.	

The table below describes the properties used to represent an alert in the array of alerts returned in the response to a **GET** request for the /alerts resource.

Property name	Data type	Description	Notes
alertID	String	Specifies the alert ID.	
level	String	Specifies the effect on the S Series Node of the condition to which the alert applies. Possible values are: • **NORMAL** — The S Series Node is functioning normally. • **DEGRADED** — The S Series Node has one or more noncritical problems that may require attention. • **CRITICAL** — The S Series Node has one or more critical problems that require attention.	
message	String	Specifies the full text of the alert.	

(Continued)

Property name	Data type	Description	Notes
pcode	String	Specifies the Hitachi Vantara part number to use when ordering a replacement for the component indicated by the alert.	This property is returned only when the alert is about a specific, identifiable hardware component.
priority	String	Specifies the level of importance for resolving the condition to which the alert applies. Possible values are: • **1** — High priority • **2** — Medium priority • **3** — Low priority • **4** — Notification only	
scope	String	Specifies the type of component or activity to which the alert applies. Possible values are: • **CERT** — SSL server certificates • **DRIVE** — Data and database drives • **ENCLOSURE** — Enclosures • **FS** — Storage usage • **MAINT** — Maintenance procedures • **SECURITY** — Configuration that requires the security role; failed logins • **SERVER** — Server modules	

(Continued)

Property name	Data type	Description	Notes
		- **SYSTEM** — Configuration that does not require the security role; successful logins; system-initiated events - **UPGRADE** — Software updates and hotfix applications	
scopeRef	Integer	For a scope of DRIVE, specifies the number of the enclosure that contains the data or database drive to which the alert applies. For a scope of ENCLOSURE, specifies the number of the enclosure to which the alert applies. For a scope of SERVER, specifies the number of the server module to which the alert applies. This property is included in the response body only if the scope of the alert is DRIVE, ENCLOSURE, or SERVER.	

(Continued)

Property name	Data type	Description	Notes
scopeSubRef	Integer	For a scope of DRIVE, specifies the ID (not number) of the slot containing the data or database drive to which the alert applies. For a scope of ENCLOSURE, specifies a value that identifies the enclosure component to which the alert applies. For a scope of SERVER, specifies a value that identifies the server module component to which the alert applies. This property is included in the response body only if the scope of the alert is DRIVE, ENCLOSURE, or SERVER and the event applies to a specific drive, enclosure component, or server module component. For information about the possible values of this property for the ENCLOSURE and SERVER scopes, see "scopes, scopeRefs, and scopeSubRefs query parameters" on page 182.	
severity	String	Specifies the severity of the condition described by the alert. Possible values are: • **GOOD** • **WARNING** • **BAD**	

(Continued)

Property name	Data type	Description	Notes
shortName	String	Specifies a brief description of the condition to which the alert applies.	
ticket	Boolean	Specifies whether the Hitachi Remote Ops monitor agent should open a ticket for the condition to which the alert applies (assuming the monitor agent is configured to monitor the S Series Node). The Remote Ops monitor agent is a Hitachi Vantara product that enables remote monitoring of HCP S Series Nodes.	

/alerts query parameters

You can use query parameters to limit the alerts included in the response to a **GET** request for the /alerts resource. The query parameters you can use are:

- **severities**
- **scopes**
- **scopeRefs**
- **scopeSubRefs**

These query parameters can be used alone or in combination with each other.

You use the **severities** query parameter in a **GET** request for the /alerts resource to request alerts with specific severities. Valid values for this parameter are comma-separated lists of one or more of these:

- **GOOD**
- **WARNING**
- **BAD**

These values are case sensitive.

The **scopes**, **scopeRefs**, and **scopeSubRefs** parameters are also used with **GET** requests for the /events resource. For information about these parameters, see "scopes, scopeRefs, and scopeSubRefs query parameters" on page 182.

For more information about query parameters, see "Management API query parameters" on page 61.

/alerts example

Here's a sample **GET** request that retrieves a list of the alerts that apply to server module 1.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/alerts?scope=SERVER
  &scopeRefs=1&prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/alerts?scope=SERVER&scopeRefs=1&prettyprint
HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 703
```

Response body

```
{
  "alerts": [
    {
      "alertId": "2632",
      "shortName": "Server module unavailable",
      "message": "Server module 1 is unavailable.",
      "priority": "1",
      "severity": "BAD",
      "ticket": false,
      "scope": "SERVER",
      "scopeRef": 1,
      "level": "CRITICAL"
    },
    {
      "alertId": "2657",
      "shortName": "Server module powered off",
      "message": "Server module 1 is powered off.",
      "priority": "1",
      "severity": "BAD",
      "ticket": false,
      "scope": "SERVER",
      "scopeRef": 1,
      "level": "CRITICAL"
    }
  ],
  "severities": [
    "GOOD",
    "WARNING",
    "BAD"
  ],
  "scopes": [
    "SERVER"
  ],
  "scopeRefs": [
    1
  ]
}
```

/buckets

With the /buckets resource:

- A **PUT** request requires a request body.
- A **GET** request returns a response body.

For information about the query parameters used to limit the bucket list returned by a **GET** request, see "Managing resource lists" on page 91.

For more information about the /buckets resource, see "Bucket resources" on page 73.

/buckets properties

The table below describes the properties in /buckets resource response bodies. For the properties for /buckets resource request bodies used with **PUT** requests, see "/buckets/bucket-name properties" on page 110.

Property name	Data type	Description	Notes
bucketName	Array	Specifies a comma-separated list of the buckets that satisfy the request criteria. Each bucket is represented by the value of its bucketName property.	
count	Integer	Specifies the value of the **count** query parameter included in the **GET** request or 1,000 if the request did not include the **count** parameter. For more information, see "count query parameter" on page 92.	
isTruncated	Boolean	Specifies whether the returned list of buckets is complete. Possible values are: • **true** — The bucket list is incomplete. • **false** — The bucket list is complete. For more information, see "count query parameter" on page 92.	

(Continued)

Property name	Data type	Description	Notes
marker	String	Specifies the value of the **marker** query parameter included in the **GET** request or no value if the request did not include the **marker** parameter. For more information, see "marker query parameter" on page 93.	
owner	String	Specifies the value of the **owner** query parameter included in the **GET** request or no value if the request did not include the **owner** parameter. For more information, see "owner query parameter" on page 95.	
prefix	String	Specifies the value of the **prefix** query parameter included in the **GET** request or no value if the request did not include the **prefix** parameter. For more information, see "prefix query parameter" on page 94.	

/buckets example

Here's a sample **GET** request that retrieves a list of existing buckets.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/buckets?prettyprint"
```

Request headers

```
GET /mapi/buckets?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 134
```

Response body

```
{
  "owner": "",
  "marker": "",
  "prefix": "",
  "count": 1000,
  "isTruncated": false,
  "bucketName": [
    "hcpsrv-hcp-ma"
  ]
}
```

/buckets/*bucket-name*

With the /buckets/*bucket-name* resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.
- **HEAD** and **DELETE** requests do not take a request body and do not return a response body.

For more information about the /buckets/*bucket-name* resource, see "Bucket resources" on page 73.

/buckets/*bucket-name* properties

The table below describes the properties in /buckets/*bucket-name* resource request and response bodies. These properties apply to an individual bucket. They are also used in the request body for **PUT** requests with the /buckets resource.

Property name	Data type	Description	Notes
bucketID	Integer	Specifies the internal ID for the bucket. The S Series Node generates this ID automatically when the bucket is created.	This property is not valid on a **PUT** or **POST** request.
bucketName	String	Specifies the name for the bucket. For the rules for bucket names, see "Bucket names" on page 28.	This property is required on a **PUT** request. It is not valid on a **POST** request.
creationTime	Timestamp	Specifies the date and time at which the bucket was created, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	This property is not valid on a **PUT** or **POST** request.
description	String	Specifies a description of the bucket. This description is optional. Descriptions can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space. To remove a description from a bucket, specify the description property with no value.	This property is optional on a **PUT** or **POST** request.
owner	String	Specifies the username for the user account that owns the bucket. This user account must have the data role.	This property is required on a **PUT** request. It is optional on a **POST** request.

/buckets/*bucket-name* example

Here's a sample **GET** request that retrieves information about the bucket named *hcpsrv-hcp-ma*.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/buckets/hcpsrv-hcp-ma?prettyprint"
```

Request headers

```
GET /mapi/buckets/hcpsrv-hcp-ma?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 186
```

Response body

```
{
  "bucketName": "hcpsrv-hcp-ma",
  "bucketID": 8,
  "owner": "hcpsrv-hcp-ma",
  "description": "Bucket for HCP system hcp-ma.example.com",
  "creationTime": "2018-09-26 13:18:51 UTC"
}
```

/buckets/*bucket-name*/irreparables

With the /buckets/*bucket-name*/irreparables resource:

- A **GET** request returns a response body.

- A **HEAD** request returns a count of the irreparable objects in the bucket in the X-HCPS-Irreparable-Count response header.

For information about the query parameters used to limit the list of irreparable objects returned by a **GET** request, see "Managing resource lists" on page 91.

For more information about the /buckets/*bucket-name*/irreparables resource, see "Irreparables resources" on page 77.

/buckets/*bucket-name*/irreparables properties

The table below describes the properties in /buckets/*bucket-name*/irreparables resource response bodies.

Property name	Data type	Description	Notes
count	Integer	Specifies the value of the **count** query parameter included in the **GET** request or 1,000 if the request did not include the **count** parameter. For more information, see "count query parameter" on page 92.	
irreparables	Array	Specifies a comma-separated list of the irreparable objects that satisfy the request criteria. Each object is represented by the properties described in the next table.	

(Continued)

Property name	Data type	Description	Notes
isTruncated	Boolean	Specifies whether the returned list of irreparable objects is complete. Possible values are: • **true** — The irreparable object list is incomplete. • **false** — The irreparable object list is complete. For more information, see "count query parameter" on page 92.	
marker	String	Specifies the value of the **marker** query parameter included in the **GET** request or no value if the request did not include the **marker** parameter. For more information, see "marker query parameter" on page 93.	
nextMarker	String	If the value of the isTruncated property is **true**, specifies an automatically generated string that identifies the last irreparable object in the returned list. If the value of the isTruncated property is **false**, this property is not included in the response body.	

The table below describes the properties used to represent an irreparable object in the array of irreparable objects returned in response to a **GET** request for the /bucket/*bucket-name*/irreparables resource.

Property name	Data type	Description	Notes
bucketId	Integer	Specifies the internal ID for the bucket that contains the irreparable object.	
bucketName	String	Specifies the bucket name.	
irreparableTime	String	Specifies the date and time at which the S Series Node first detected that the object was irreparable, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	
partNumber	Integer	Specifies the part number of uploaded content that's an individual part of an in-progress multipart write.	This property is returned by a **GET** request only if the uploaded content is part of an in-progress multipart write.
path	String	Specifies the full path to and name of the object.	
uploadId	Integer	Specifies the ID of the in-progress multipart write that the uploaded content is part of.	This property is returned by a **GET** request only if the uploaded content is part of an in-progress multipart write.

/buckets/*bucket-name*/irreparables examples

The examples below show the use of the /buckets/*bucket-name*/irreparables resource with the **GET** and **HEAD** methods.

/buckets/*bucket-name*/irreparables GET example

Here's a sample **GET** request that retrieves the first irreparable object in the list of irreparable objects in the bucket named *hcpsrv-hcp-ma*.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/buckets/hcpsrv-hcp-ma
  /irreparables?count=1&prettyprint"
```

Request headers

```
GET /mapi/buckets/hcpsrv-hcp-ma/irreparables?count=1&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 244
```

Response body

```
{
  "marker": ""
  "nextMarker": "eyJidWNrZXRJZCI6MSwicGF0aCI6InJoaW5vX2Rpcl9oMV9MMV9kdzEv
cmhpbm9fZmlsZV9oMl9MMV9kdzFfMTAwMCIsInVwbG9hZElkIjotMSwicGFydE51bWJlciI
6LTF9"
  "count": 1,
  "isTruncated": true,
  "irreparables": [
    {"bucketId":1,"bucketName":"hcpsrv-hcp-ma",
"path":"d00/00/00d27c6245a09380c58566158681",
"irreparableTime":"2018-09-08 17:56:02 UTC"
    }
  ]
}
```

/buckets/*bucket-name*/irreparables HEAD example

Here's a sample **HEAD** request that retrieves a count of the irreparable objects in the bucket named *hcpsrv-hcp-ma*.

Request with curl command line

```
curl -k -X HEAD -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/buckets/hcpsrv-hcp-ma
  /irreparables?prettyprint"
```

Request headers

```
HEAD /mapi/buckets/hcpsrv-hcp-ma/irreparables?prettyprint HTTP/1.1
Host: mapi.s-10-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
X-HCPS-Irreparable-Count: 2
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/configuration/certificates/system

With the /configuration/certificates/system resource, a **GET** request returns a response body.

For more information about the /configuration/certificates/system resource, see "Certificate resources" on page 74.

/configuration/certificates/system properties

The table below describes the properties in /configuration/certificates/system resource response bodies. These properties describe the SSL server certificate currently in use by the S Series Node.

Property name	Data type	Description	Notes
commonName	String	The common name (CN) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is the domain name of the S Series Node prefixed with an asterisk and a period (*.).	
country	String	The two-letter ISO 3166-1 abbreviation for the country (C) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *US*.	
created	String	The date and time at which the SSL server certificate was generated, in this format: *DDD MMM dd hh:mm:ss* UTC *yyyy* For example: Wed Oct 09 14:51:57 UTC 2018	
distinguishedName	String	The distinguished name (DN) for the SSL server certificate.	

(Continued)

Property name	Data type	Description	Notes
expires	String	The date and time at which the SSL server certificate expires, in this format: *DDD MMM dd hh:mm:ss* UTC *yyyy* For example: Thu Oct 08 14:51:57 UTC 2023	
locality	String	The location (L) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *Waltham*.	
organization	String	The organization (O) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *Hitachi Data Systems*.	
organizationalUnit	String	The organizational unit (OU) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *HCP S Series Node*.	
state	String	The state or province (ST) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *Massachusetts*.	

/configuration/certificates/system example

Here's a sample **GET** request that retrieves information about the SSL server certificate currently in use by the HCP S Series.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/certificates/system
  ?prettyprint"
```

Request headers

```
GET /mapi/configuration/certificates/system?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 457
```

Response body

```
{
  "distinguishedName": "CN\u003d*.s-node-1.example.com,OU\u003dHCP S Series
    Node,O\u003dHitachi Data Systems,L\u003dWaltham,ST\u003dMassachusetts,
    C\u003dUS",
  "commonName": "*.s-node-1.example.com",
  "organization": "Hitachi Data Systems",
  "organizationalUnit": "HCP S Series Node",
  "locality": "Waltham",
  "state": "Massachusetts",
  "country": "US",
  "created": "Wed Oct 09 14:51:57 UTC 2018",
  "expires": "Thu Oct 08 14:51:57 UTC 2023"
}
```

/configuration/certificates/system/generate

With the /configuration/certificates/system/generate resource, a **POST** request returns a response body. The request does not take a request body.

For more information about the /configuration/certificates/system/generate resource, see "Certificate resources" on page 74.

/configuration/certificates/system/generate properties

The table below describes the properties in /configuration/certificates/system/generate resource response bodies. These properties describe the SSL server certificate generated by the S Series Node.

Property name	Data type	Description	Notes
commonName	String	The common name (CN) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is the domain name of the S Series Node prefixed with an asterisk and a period (*.).	
country	String	The two-letter ISO 3166-1 abbreviation for the country (C) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *US*.	
created	String	The date and time at which the SSL server certificate was generated, in this format: *DDD MMM dd hh:mm:ss* UTC *yyyy* For example: Wed Oct 09 14:51:57 UTC 2018	
distinguishedName	String	The distinguished name (DN) for the SSL server certificate.	

(Continued)

Property name	Data type	Description	Notes
expires	String	The date and time at which the SSL server certificate expires, in this format: *DDD MMM dd hh:mm:ss* UTC *yyyy* For example: Thu Oct 08 14:51:57 UTC 2023	
locality	String	The location (L) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *Waltham*.	
organization	String	The organization (O) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *Hitachi Data Systems*.	
organizationalUnit	String	The organizational unit (OU) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *HCP S Series Node*.	
state	String	The state or province (ST) for the SSL server certificate. For a self-signed certificate generated by the S Series Node, the value of this property is *Massachusetts*.	

/configuration/certificates/system/generate example

Here's a sample **POST** request that generates a new SSL server certificate for the HCP S Series.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/certificates/system
  /generate?prettyprint"
```

Request headers

```
POST /mapi/configuration/certificates/system/generate?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 457
```

Response body

```
{
  "distinguishedName": "CN\u003d*.rhino8.lab.archivas.com,OU\u003dHCP S Series
    Node,O\u003dHitachi Data Systems,L\u003dWaltham,ST\u003dMassachusetts,
    C\u003dUS",
  "commonName": "*.rhino8.lab.archivas.com",
  "organization": "Hitachi Data Systems",
  "organizationalUnit": "HCP S Series Node",
  "locality": "Waltham",
  "state": "Massachusetts",
  "country": "US",
  "created": "Wed Oct 09 08:10:07 UTC 2018",
  "expires": "Thu Oct 08 08:10:07 UTC 2023"
}
```

/configuration/console

With the /configuration/console resource:

- A **GET** request returns a response body.

- A **POST** request requires a request body.

For more information about the /configuration/console resource, see "Console resource" on page 75.

/configuration/console properties

The table below describes the properties in /configuration/console resource request and response bodies.

Property name	Data type	Description	Notes
accessNetworkHttp Enabled	Boolean	Specifies whether HTTP without SSL security can be used for access to the Management Console on the access network. Valid values are: • **true** — HTTP can be used without SSL security. • **false** — HTTP cannot be used without SSL security. The default is **false**. These values are not case sensitive.	You can set the value of this property to **true** only if the value of the accessNetworkHttpsEnabled property is also **true**. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpsEnabled, managementNetworkHttp Enabled, and managementNetworkHttps Enabled properties.

(Continued)

Property name	Data type	Description	Notes
accessNetworkHttps Enabled	Boolean	Specifies whether HTTPS can be used for access to the Management Console on the access network. Valid values are: • **true** — HTTPS can be used. • **false** — HTTPS cannot be used. The default is **true**. These values are not case sensitive.	If the value of this property is **false**, access to the Management Console on the access network is not allowed. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled, managementNetworkHttp Enabled, and managementNetworkHttps Enabled properties. Either this property or the managementNetworkHttps Enabled property must be set to **true**.
allowIfInBothLists	Boolean	Specifies how the S Series Node handles IP addresses that are included in both or neither of the lists of allowed or denied addresses. Valid values are: • **true** — IP addresses included in both lists have access. • **false** — IP addresses included in both lists do not have access. The default is **true**. These values are not case sensitive. For more information about allow and deny list handling, see "Allow and deny lists" on page 44.	

(Continued)

Property name	Data type	Description	Notes
allowList	Array	Specifies a comma-separated list of IP addresses that are allowed access to the Management Console. Each item in the list can be an individual IP address or a range of IP addresses specified either as *ip-address/subnet-mask* (IPv4 only) or in CIDR format. To remove all IP addresses from the allow list, specify an empty array for the allowList property.	With a **POST** request, the list of IP addresses specified in the request body replaces the current list of allowed IP addresses.
denyList	Array	Specifies a comma-separated list of IP addresses that are denied access to the Management Console. Each item in the list can be an individual IP address or a range of IP addresses specified either as *ip-address/subnet-mask* (IPv4 only) or in CIDR format. To remove all IP addresses from the deny list, specify an empty array for the denyList property.	With a **POST** request, the list of IP addresses specified in the request body replaces the current list of denied IP addresses.

(Continued)

Property name	Data type	Description	Notes
loginMessage	String	Specifies message text to appear on the login page of the Management Console. This text is optional. If specified, it can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space. The default is no message. To remove a message, specify the loginMessage property with no value.	
managementNetwork HttpEnabled	Boolean	Specifies whether HTTP without SSL security can be used for access to the Management Console on the management network. Valid values are: • **true** — HTTP can be used without SSL security. • **false** — HTTP cannot be used without SSL security. The default is **false**. These values are not case sensitive.	You can set the value of this property to **true** only if the value of the managementNetworkHttps Enabled property is also **true**. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled, accessNetworkHttpsEnabled, and managementNetworkHttps Enabled properties.

(Continued)

Property name	Data type	Description	Notes
managementNetwork HttpsEnabled	Boolean	Specifies whether HTTPS can be used for access to the Management Console on the management network. Valid values are: • **true** — HTTPS can be used. • **false** — HTTPS cannot be used. The default is **true**. These values are not case sensitive.	If the value of this property is **false**, access to the Management Console on the management network is not allowed. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled, accessNetworkHttpsEnabled, and managementNetworkHttp Enabled properties. Either this property or the accessNetworkHttpsEnabled property must be set to **true**.

/configuration/console example

Here's a sample **GET** request that retrieves the configuration of the HCP S Series Management Console.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/console?prettyprint"
```

Request headers

```
GET /mapi/configuration/console?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 459
```

Response body

```
{
  "loginMessage": "Use of the HCP S Series Management Console is restricted to
members of the IT and storage administration groups.",
  "accessNetworkHttpEnabled": false,
  "accessNetworkHttpsEnabled": true,
  "managementNetworkHttpEnabled": false,
  "managementNetworkHttpsEnabled": true,
  "allowList": [
    10.0.41.13,
    10.0.41.27,
    10.0.41.23,
    10.0.41.56,
    10.0.41.15,
    10.0.41.49
  ],
  "denyList": [],
  "allowIfInBothLists": false
}
```

/configuration/dns

With the /configuration/dns resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/dns resource, see "DNS resource" on page 75.

/configuration/dns properties

The table below describes the properties in /configuration/dns resource request and response bodies.

Property name	Data type	Description	Notes
dnsServers	Array	Specifies a comma-separated list of the IP addresses of up to three DNS servers. To remove all specified DNS servers from the S Series Node, specify an empty array for the dnsServers property.	With a **POST** request, the list of DNS servers specified in the request body replaces the current list of DNS servers.
network	String	Specifies the network to be used for communication between the S Series Node and the specified DNS servers. Valid values are: • **[access]** — Use the access network. • **[management]**— Use the management network. The default is **[access]**. These values are case sensitive.	For the S Series Node to communicate with the specified DNS servers, the IP mode of the specified network must match the IP mode of the DNS server IP addresses.

(Continued)

Property name	Data type	Description	Notes
networkIndex	Integer	Specifies whether to use the primary or secondary IPv6 gateway if the network used for communication with the DNS servers is configured for IPv6. Valid values are: • **1** — Use the primary IPv6 gateway. • **2** — Use the secondary IPv6 gateway. The default is **1**. If the network is configured for IPv4, the only valid value is **1**.	

/configuration/dns example

Here's a sample **GET** request that retrieves the DNS server configuration for the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/dns?prettyprint"
```

Request headers

```
GET /mapi/configuration/dns?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 109
```

Response body

```
{
  "dnsServers": [
    "10.0.201.50",
    "10.0.201.55"
  ],
  "network": "[ACCESS]",
  "networkIndex": 1
}
```

/configuration/ident

With the /configuration/ident resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/ident resource, see "Identification resource" on page 77.

/configuration/ident properties

The table below describes the properties in /configuration/ident resource request and response bodies.

Property name	Data type	Description	Notes
domainName	String	Specifies the domain name of the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
model	String	Specifies the model of the S Series Node. Possible values are: • **S10** — HCP S10 Node • **S10V** — HCP S Series Node Demo Edition • **S11** — HCP S11 Node • **S30** — HCP S30 Node • **S31** — HCP S31 Node	This property is not valid on a **POST** request.
serialNumber	String	Specifies the S Series Node serial number.	This property is not valid on a **POST** request.
softwareVersion	String	Specifies the version of the HCP S Series software currently running on the S Series Node.	This property is not valid on a **POST** request.

/configuration/ident example

Here's a sample **GET** request that retrieves information that identifies the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/ident?prettyprint"
```

Request headers

```
GET /mapi/configuration/ident?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 120
```

Response body

```
{
  "serialNumber": "HHCA310000001",
  "domainName": "s-node-1.example.com",
  "softwareVersion": "3.0.0.2",
  "model": "S31"
}
```

/configuration/mapi

With the /configuration/mapi resource:

- A **GET** request returns a response body.

- A **POST** request requires a request body.

For more information about the /configuration/console resource, see "Management API resource" on page 82.

/configuration/mapi properties

The table below describes the properties in /configuration/mapi resource request and response bodies.

Property name	Data type	Description	Notes
accessNetworkHttp Enabled	Boolean	Specifies whether HTTP without SSL security can be used for access to the S Series Node through the management API on the access network. Valid values are: • **true** — HTTP can be used without SSL security. • **false** — HTTP cannot be used without SSL security. The default is **false**. These values are not case sensitive.	You can set the value of this property to **true** only if the value of the accessNetworkHttpsEnabled property is also **true**. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpsEnabled, managementNetworkHttp Enabled, and managementNetworkHttps Enabled properties.
accessNetworkHttps Enabled	Boolean	Specifies whether HTTPS can be used for access to the S Series Node through the management API on the access network. Valid values are: • **true** — HTTPS can be used. • **false** — HTTPS cannot be used. The default is **true**. These values are not case sensitive.	If the value of this property is **false**, access to the S Series Node through the management API on the access network is not allowed. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled, managementNetworkHttp Enabled, and managementNetworkHttps Enabled properties. Either this property or the managementNetworkHttps Enabled property must be set to **true**.

(Continued)

Property name	Data type	Description	Notes
allowIfInBothLists	Boolean	Specifies how the S Series Node handles IP addresses that are included in both or neither of the lists of allowed or denied addresses. Valid values are: • **true** — IP addresses included in both lists have access. • **false** — IP addresses included in both lists do not have access. The default is **true**. These values are not case sensitive. For more information about allow and deny list handling, see "Allow and deny lists" on page 44.	
allowList	Array	Specifies a comma-separated list of IP addresses that are allowed access to the S Series Node through the management API. Each item in the list can be an individual IP address or a range of IP addresses specified either as *ip-address/subnet-mask* (IPv4 only) or in CIDR format. To remove all IP addresses from the allow list, specify an empty array for the allowList property.	With a **POST** request, the list of IP addresses specified in the request body replaces the current list of allowed IP addresses.

(Continued)

Property name	Data type	Description	Notes
denyList	Array	Specifies a comma-separated list of IP addresses that are denied access to the S Series Node through the management API. Each item in the list can be an individual IP address or a range of IP addresses specified either as *ip-address/subnet-mask* (IPv4 only) or in CIDR format. To remove all IP addresses from the deny list, specify an empty array for the denyList property.	With a **POST** request, the list of IP addresses specified in the request body replaces the current list of denied IP addresses.
managementNetwork HttpEnabled	Boolean	Specifies whether HTTP without SSL security can be used for access to the S Series Node through the management API on the management network. Valid values are: • **true** — HTTP can be used without SSL security. • **false** — HTTP cannot be used without SSL security. The default is **false**. These values are not case sensitive.	You can set the value of this property to **true** only if the value of the managementNetworkHttps Enabled property is also **true**. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled, accessNetworkHttpsEnabled, and managementNetworkHttps Enabled properties.

(Continued)

Property name	Data type	Description	Notes
managementNetwork HttpsEnabled	Boolean	Specifies whether HTTPS can be used for access to the S Series Node through the management API on the management network. Valid values are: • **true** — HTTPS can be used. • **false** — HTTPS cannot be used. The default is **true**. These values are not case sensitive.	If the value of this property is **false**, access to the S Series Node through the management API on the management network is not allowed. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled, accessNetworkHttpsEnabled, and managementNetworkHttp Enabled properties. Either this property or the accessNetworkHttpsEnabled property must be set to **true**.

/configuration/mapi example

Here's a sample **GET** request that retrieves the configuration of the HCP S Series management API.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/mapi?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 324
```

Response body

```
{
  "accessNetworkHttpEnabled": false,
  "accessNetworkHttpsEnabled": true,
  "managementNetworkHttpEnabled": false,
  "managementNetworkHttpsEnabled": true,
  "allowList": [
    10.0.41.13,
    10.0.41.27,
    10.0.41.23,
    10.0.41.56,
    10.0.41.15,
    10.0.41.49
  ],
  "denyList": [],
  "allowIfInBothLists": false
}
```

/configuration/networks/builtin

With the /configuration/networks/builtin resource, a **GET** request returns a response body.

For more information about the /configuration/networks/builtin resource, see "Network resources" on page 84.

/configuration/networks/builtin property

The table below describes the property in /configuration/networks/builtin resource response bodies.

Property name	Data type	Description	Notes
networkName	Array	Specifies a comma-separated list of the predefined S Series Node networks. Each network is represented by the value of its networkName property.	With the management API, the names used for networks are not enclosed in square brackets.

/configuration/networks/builtin example

Here's a sample **GET** request that retrieves a list of the predefined S Series Node networks.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/networks/builtin
  ?prettyprint"
```

Request headers

```
GET /mapi/configuration/networks/builtin?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 77
```

Response body

```
{
  "networkName": [
    "interconnect",
    "access",
    "management"
  ]
}
```

/configuration/networks/builtin/access/ports

With the /configuration/networks/builtin/access/ports resource, a **GET** request returns a response body.

For more information about the /configuration/networks/builtin/access/ports resource, see "Network resources" on page 84.

/configuration/networks/builtin/access/ports properties

The table below describes the property in /configuration/networks/builtin/access/ports resource response bodies.

Property name	Data type	Description	Notes
portExpectations	Array	Specifies a comma-separated list of the access network ports. Each port is represented by the properties described in the next table.	

The table below describes the properties used to represent a port in the array of access network ports returned in the response to a **GET** request for the /configuration/networks/builtin/access/ports resource.

Property name	Data type	Description	Notes
connectionExpected	Boolean	Specifies whether the S Series Node expects the port to be connected to an active port on a network switch. Possible values are: • **true** — The S Series Node expects the port to be connected. • **false** — The S Series Node expects the port not to be connected.	
portNumber	Integer	Specifies the access network port number. Possible values are **0**, **1**, **2**, and **3**.	

/configuration/networks/builtin/access/ports example

Here's a sample **GET** request that retrieves the connection expectations for all four access network ports.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/networks/builtin
  /access/ports?prettyprint"
```

Request headers

```
GET /mapi/configuration/networks/builtin/access/ports?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 308
```

Response body

```
{
  "portExpectations": [
    {
      "portNumber": 0,
      "connectionExpected": true
    },
    {
      "portNumber": 1,
      "connectionExpected": true
    },
    {
      "portNumber": 2,
      "connectionExpected": false
    },
    {
      "portNumber": 3,
      "connectionExpected": false
    }
  ]
}
```

/configuration/networks/builtin/access/ports/*port-number*

With the /configuration/networks/builtin/access/ports/*port-number* resource:

- A **GET** request returns a response body.

- A **POST** request requires a query parameter. The request does not take a request body and does not return a response body.

For more information about the /configuration/networks/builtin/access/ports/*port-number* resource, see "Network resources" on page 84.

/configuration/networks/builtin/access/ports/*port-number* properties

The table below describes the properties in /configuration/networks/builtin/access/ports/*port-number* resource response bodies.

Property name	Data type	Description	Notes
connectionExpected	Boolean	Specifies whether the S Series Node expects the port to be connected to an active port on a network switch. Possible values are: • **true** — The S Series Node expects the port to be connected. • **false** — The S Series Node expects the port not to be connected.	
portNumber	Integer	Specifies the access network port number. Possible values are **0**, **1**, **2**, and **3**.	

/configuration/networks/builtin/access/ports/*port-number* query parameter

To change the connection expectation for an access network port, you use the **connectionExpected** query parameter with a **POST** request for the /configuration/networks/builtin/access/ports/*port-number* resource. Valid values for this parameter are:

- **true** — Tells the S Series Node to expect the port to be connected to an active port on a network switch.
- **false** — Tells the S Series Node to expect the port not to be connected to an active port on a network switch.

These values are not case sensitive.

For more information about query parameters, see "Management API query parameters" on page 61.

/configuration/networks/builtin/access/ports/*port-number* examples

The examples below show the use of the /configuration/networks/builtin/access/ports/*port-number* resource with the **GET** and **POST** methods.

/configuration/networks/builtin/access/ports/*port-number* GET example

Here's a sample **GET** request that retrieves the connection expectation for access network port 3.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/networks/builtin
  /access/ports/3?prettyprint"
```

Request headers

```
GET /mapi/configuration/networks/builtin/access/ports/3?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 52
```

Response body

```
{
  "portNumber": 3,
  "connectionExpected": false
}
```

/configuration/networks/builtin/access/ports/*port-number* POST example

Here's a sample **POST** request that tells the S Series Node to expect access network port 3 to be connected to an active port on a network switch.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/networks/builtin
  /access/ports/3?connectionExpected=true"
```

Request headers

```
POST /mapi/configuration/networks/builtin/access/ports/3?connectionExpected=true
HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/configuration/networks/builtin/*network-name*

With the /configuration/networks/builtin/*network-name* resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

In this resource identifier, valid values for *network-name* are:

- **access** for the access network
- **interconnect** for the server interconnect network
- **management** for the management network

With the management API, the names used for networks are not enclosed in square brackets.

For more information about the /configuration/networks/builtin/*network-name* resource, see "Network resources" on page 84.

/configuration/networks/builtin/*network-name* properties

The table below describes the properties in /configuration/networks/builtin/*network-name* resource request and response bodies. If you want to change a subnet for the access or management network, you need to make all the changes to the applicable subnet, gateway, and IP address properties in a single **POST** request.

Property name	Data type	Description	Notes
bondingMode	String	For the access network, specifies the bonding mode for the network. Valid values are **802.3ad** and **active-backup**. For the management network, the value of this property is always **none**.	This property is not valid on a **POST** request for the management or server interconnect network. It is not returned by a **GET** request for the server interconnect network.
gateway1	String	Specifies the IPv4 gateway address for the network or the primary IPv6 gateway address.	This property is not valid on a **POST** request for the server interconnect network and is not returned by a **GET** request for that network.

(Continued)

Property name	Data type	Description	Notes
gateway2	String	Specifies the secondary IPv6 gateway address for the network. To remove a secondary IPv6 gateway address from the network, specify the gateway2 property with no value.	This property is valid on a **POST** request for the access or management network only when the IP mode for the network is IPv6.It is not valid on a **POST** request for the server interconnect network. This property is returned by a **GET** request for the access or management network only if the IP mode for the network is IPv6. It is not returned by a **GET** request for the server interconnect network.
mtu	String	Specifies the maximum transmission unit (MTU) for the network. Valid values are **9000** and **1500**.	This property is not valid on a **POST** request for the server interconnect network and is not returned by a **GET** request for that network.
networkName	String	Specifies the name of the network.	This property is not valid on a **POST** request.
serverModule1IpAddress1	String	Specifies the IPv4 address for server module 1 on the network or the primary IPv6 address for server module 1.	This property is not valid on a **POST** request for the server interconnect network and is not returned by a **GET** request for that network.

(Continued)

Property name	Data type	Description	Notes
serverModule1IpAddress2	String	Specifies the secondary IPv6 address for server module 1 on the network. To remove a secondary IPv6 address for server module 1 from the network, specify the serverModule1IpAddress2 property with no value.	This property is valid on a **POST** request for the access or management network only if the IP mode for the network is IPv6. It is not valid on a **POST** request for the server interconnect network. This property is returned by a **GET** request for the access or management network only if the IP mode for the network is IPv6. It is not returned by a **GET** request for the server interconnect network.
serverModule1VipAddress1	String	Specifies the virtual IPv4 address for server module 1 on the network or the primary virtual IPv6 address for server module 1.	This property is not valid on a **POST** request for the management or server interconnect network. It is not returned by a **GET** request for the management or server interconnect network.
serverModule1VipAddress2	String	Specifies the secondary virtual IPv6 address for server module 1 on the network. To remove a secondary virtual IPv6 address for server module 1 from the network, specify the serverModule1VipAddress2 property with no value.	This property is valid on a **POST** request for the access network only if the IP mode for the network is IPv6. It is not valid on a **POST** request for the management or server interconnect network. This property is returned by a **GET** request for the access network only if the IP mode for the network is IPv6. It is not returned by a **GET** request for the management or server interconnect network.

(Continued)

Property name	Data type	Description	Notes
serverModule2IpAddress1	String	Specifies the IPv4 address for server module 2 on the network or the primary IPv6 address for server module 2.	This property is not valid on a **POST** request for the server interconnect network and is not returned by a **GET** request for that network.
serverModule2IpAddress2	String	Specifies the secondary IPv6 address for server module 2 on the network. To remove a secondary IPv6 address for server module 2 from the network, specify the serverModule2IpAddress2 property with no value.	This property is valid on a **POST** request for the access or management network only if the IP mode for the network is IPv6. It is not valid on a **POST** request for the server interconnect network. This property is returned by a **GET** request for the access or management network only if the IP mode for the network is IPv6. It is not returned by a **GET** request for the server interconnect network.
serverModule2VipAddress1	String	Specifies the virtual IPv4 address for server module 2 on the network or the primary virtual IPv6 address for server module 2.	This property is not valid on a **POST** request for the management or server interconnect network. It is not returned by a **GET** request for the management or server interconnect network.

(Continued)

Property name	Data type	Description	Notes
serverModule2VipAddress2	String	Specifies the secondary virtual IPv6 address for server module 2 on the network. To remove a secondary virtual IPv6 address for server module 2 from the network, specify the serverModule2VipAddress2 property with no value.	This property is valid on a **POST** request for the access network only if the IP mode for the network is IPv6. It is not valid on a **POST** request for the management or server interconnect network. This property is returned by a **GET** request for the access or management network only if the IP mode for the network is IPv6. It is not returned by a **GET** request for the server interconnect network.
speedDuplex	String	Specifies the combined speed and duplex setting for the network. Valid values are the values for the supportedSpeedDuplex property returned by a **GET** request for the network. For the management network, the only valid value is **auto**.	This property is not valid on a **POST** request for the server interconnect network and is not returned by a **GET** request for that network.
subnet1	String	Specifies the IPv4 subnet for the network or the primary IPv6 subnet for the network, in CIDR notation.	

(Continued)

Property name	Data type	Description	Notes
subnet2	String	Specifies the secondary IPv6 subnet for the network, in CIDR notation. To remove a secondary IPv6 subnet from the network, specify the subnet2 property with no value.	This property is valid on a **POST** request for the access or management network only if the IP mode for the network is IPv6. It is not valid on a **POST** request for the server interconnect network. This property is returned by a **GET** request for the access or management network only if the IP mode for the network is IPv6. It is not returned by a **GET** request for the server interconnect network.
supportedSpeedDuplex	Array	Specifies a comma-separated list of the combined speed and duplex settings that are supported for the network. Possible values are **auto**, **10H**, **10F**, **100H**, **100F**, **1000H**, **1000F**, and **10000F**.	This property is not valid on a **POST** request. It is not returned by a **GET** request for the server interconnect network.
vlan	Integer	Specifies the VLAN ID for the network. Valid values are integers in the range zero through 4,094.	This property is not valid on a **POST** request for the server interconnect network and is not returned by a **GET** request for that network.

For more information about network properties, see:

- "Access network" on page 31
- "Management network" on page 35
- "Server interconnect network" on page 39

/configuration/networks/builtin/*network-name* example

Here's a sample **GET** request that retrieves information about the access network.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/networks/builtin
  /access?prettyprint"
```

Request headers

```
GET /mapi/configuration/networks/builtin/access?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 400
```

Response body

```
{
  "supportedSpeedDuplex": [
    "10000F",
    "auto"
  ],
  "networkName": "access",
  "vlan": "0",
  "mtu": "1500",
  "speedDuplex": "auto",
  "bondingMode": "active-backup",
  "subnet1": "10.0.0.0/23",
  "gateway1": "10.0.0.254",
  "serverModule1IpAddress1": "10.0.0.1",
  "serverModule2IpAddress1": "10.0.0.2",
  "serverModule1VipAddress1": "10.0.0.3",
  "serverModule2VipAddress1": "10.0.0.4"
}
```

/configuration/protocols

With the /configuration/protocols resource, a **GET** request returns a response body.

For more information about the /configuration/protocols resource, see "Protocol resources" on page 85.

/configuration/protocols property

The table below describes the property in /configuration/protocols resource response bodies.

Property name	Data type	Description	Notes
protocols	Array	Specifies a comma-separated list of the data access protocols supported by the S Series Node.	Currently, the only supported protocol is the Hitachi API for Amazon S3 (the S3 compatible API), which is listed as **hs3**.

/configuration/protocols example

Here's a sample **GET** request that retrieves a list of the data access protocols supported by the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/protocols?prettyprint"
```

Request headers

```
GET /mapi/configuration/protocols?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 34
```

Response body

```
{
  "protocols": [
    "hs3"
  ]
}
```

/configuration/protocols/hs3

With the /configuration/protocols/hs3 resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/protocols/hs3 resource, see "Protocol resources" on page 85.

/configuration/protocols/hs3 properties

The table below describes the properties in /configuration/protocols/hs3 resource request and response bodies.

Property name	Data type	Description	Notes
accessNetworkHttpEnabled	Boolean	Specifies whether HTTP without SSL security can be used for access to the S Series Node through the Hitachi API for Amazon S3 (the S3 compatible API). Valid values are: • **true** — HTTP can be used without SSL security. • **false** — HTTP cannot be used without SSL security. The default is **true**. These values are not case sensitive.	If your S Series Node supports SSL for data access, you can set the value of this property to **true** only if the value of the accessNetworkHttpsEnabled property is also **true**. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpsEnabled property.

(Continued)

Property name	Data type	Description	Notes
accessNetworkHttpsEnabled	Boolean	Specifies whether HTTPS can be used for access to the S Series Node through the S3 compatible API. Valid values are: • **true** — HTTPS can be used. • **false** — HTTPS cannot be used. If your S Series Node supports SSL for data access, the default is **true**. If your S Series Node does not support SSL for data access, the default is **false**. These values are not case sensitive.	You can set the value of this property to **true** only if your S Series Node supports SSL for data access. If your S Series Node supports SSL for data access and the value of this property is **false**, access to the S Series Node through the S3 compatible API is not allowed. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpEnabled property.

(Continued)

Property name	Data type	Description	Notes
allowIfInBothLists	Boolean	Specifies how the S Series Node handles IP addresses that are included in both or neither of the lists of allowed or denied addresses. Valid values are: • **true** — IP addresses included in both lists have access. • **false** — IP addresses included in both lists do not have access. The default is **true**. These values are not case sensitive. For more information about allow and deny list handling, see "Allow and deny lists" on page 44.	
allowList	Array	Specifies a comma-separated list of IP addresses that are allowed access to the S Series Node through the S3 compatible API. Each item in the list can be an individual IP address or a range of IP addresses specified either as *ip-address/subnet-mask* (IPv4 only) or in CIDR format. To remove all IP addresses from the allow list, specify an empty array for the allowList property.	With a **POST** request, the list of IP addresses specified in the request body replaces the current list of allowed IP addresses.

(Continued)

Property name	Data type	Description	Notes
denyList	Array	Specifies a comma-separated list of IP addresses that are denied access to the S Series Node through the S3 compatible API. Each item in the list can be an individual IP address or a range of IP addresses specified either as *ip-address/subnet-mask* (IPv4 only) or in CIDR format. To remove all IP addresses from the deny list, specify an empty array for the denyList property.	With a **POST** request, the list of IP addresses specified in the request body replaces the current list of denied IP addresses.

/configuration/protocols/hs3 example

Here's a sample **GET** request that retrieves the configuration of the S3 compatible API.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/protocols/hs3
  ?prettyprint"
```

Request headers

```
GET /mapi/configuration/protocols/hs3?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 143
```

Response body

```
{
  "accessNetworkHttpEnabled": false,
  "accessNetworkHttpsEnabled": true,
  "allowList": [],
  "denyList": [],
  "allowIfInBothLists": true
}
```

/configuration/security

With the /configuration/security resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/console resource, see "Security resource" on page 86.

/configuration/security properties

The table below describes the properties in /configuration/security resource request and response bodies.

Property name	Data type	Description	Notes
disableAfterAttempts	Integer	Specifies the consecutive number of times a user can enter an incorrect password before the user account is automatically disabled. Valid values are integers in the range three through ten. The default is ten.	
forcePasswordChangeDays	Integer	Specifies the number of days passwords are valid before they automatically expire. Valid values are integers in the range three through 180. The default is 90.	
logoutOnInactive	Integer	Specifies the number of minutes an HCP S Series Management Console session can be inactive before it times out. Valid values are integers in the range five through 720. The default is ten.	
minimumPasswordLength	Integer	Specifies the minimum length for user account passwords. Valid values are integers in the range six through 64. The default is six.	The longer the minimum password length, the stronger user account passwords are likely to be. Encourage users to use a mix of uppercase and lowercase letters, numbers, and special characters to create even stronger passwords.

(Continued)

Property name	Data type	Description	Notes
pingEnabled	Boolean	Specifies whether ping can be used to check network connectivity to the S Series Node server modules. Valid values are: • **true** — Ping can be used to check network connectivity. • **false** — Ping cannot be used to check network connectivity. The default is **true**. These values are not case sensitive.	
sshEnabled	Boolean	Specifies whether authorized service providers can use SSH to log in to the S Series Node server modules. Valid values are: • **true** — SSH can be used to log in. • **false** — SSH cannot be used to log in. The use of SSH is initially enabled or disabled during the on-site setup of the S Series Node. These values are not case sensitive.	

/configuration/security example

Here's a sample **GET** request that retrieves the security settings for the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/security?prettyprint"
```

Request headers

```
GET /mapi/configuration/security?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 166
```

Response body

```
{
  "minimumPasswordLength": 8,
  "forcePasswordChangeDays": 30,
  "disableAfterAttempts": 3,
  "logoutOnInactive": 20,
  "pingEnabled": true,
  "sshEnabled": true
}
```

/configuration/syslog

With the /configuration/syslog resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/syslog resource, see "Syslog resources" on page 87.

/configuration/syslog properties

The table below describes the properties in /configuration/syslog resource request and response bodies.

Property name	Data type	Description	Notes
enabled	Boolean	Specifies whether the S Series Node sends event log messages to the specified syslog servers as the messages are written to the log. Valid values are: • **true** — Messages are sent to the syslog servers. • **false** — Messages are not sent to the syslog servers. The default is **false**. These values are not case sensitive.	
facility	String	Specifies the syslog local facility to which to direct the event log messages. The specified facility applies to all the specified syslog servers. Valid values are **local0** through **local7**. The default is **local0**. These values are case sensitive.	

(Continued)

Property name	Data type	Description	Notes
majorEventsOnly	Boolean	Specifies whether to send only messages about major events. Major events are those that are displayed on the **Dashboard** page of the HCP S Series Management Console. Valid values are: • **true** — Send only messages about major events. • **false** — Send messages regardless of whether they are about major events. The default is **false**. These values are not case sensitive.	
network	String	Specifies the network to be used for communication between the S Series Node and the specified syslog servers. Valid values are: • **[access]** — Use the access network. • **[management]**— Use the management network. The default is **[access]**. These values are case sensitive.	

(Continued)

Property name	Data type	Description	Notes
networkIndex	Integer	Specifies whether to use the primary or secondary IPv6 gateway if the network used for communication with the syslog servers is configured for IPv6. Valid values are: • **1** — Use the primary IPv6 gateway. • **2** — Use the secondary IPv6 gateway. The default is **1**. If the network is configured for IPv4, the only valid value is **1**.	
sendSecurityEvents	Boolean	Specifies whether to send security event messages to the specified syslog servers along with other event log messages. Valid values are: • **true** — Send messages about security events. • **false** — Do not send messages about security events. The default is **false**. These values are not case sensitive.	
servers	Array	Specifies a comma-separated list of the IP addresses (optionally, with appended port numbers) of up to ten syslog servers to which you want the S Series Node to send event log messages.	

(Continued)

Property name	Data type	Description	Notes
severity	String	Specifies the minimum severity level for event log messages to be sent to the specified syslog servers. Valid values are: • **NOTICE** — Send event log messages regardless of their severity level. • **WARNING** — Send only event log messages with a severity level of warning or error. • **ERROR** — Send only event log messages with a severity level of error. The default is **NOTICE**. These values are case sensitive.	

/configuration/syslog example

Here's a sample **GET** request that retrieves the syslog configuration for the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/syslog?prettyprint"
```

Request headers

```
GET /mapi/configuration/syslog?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 212
```

Response body

```
{
  "enabled": true,
  "servers": [
    "10.0.245.68"
  ],
  "majorEventsOnly": true,
  "sendSecurityEvents": false,
  "severity": "WARNING",
  "facility": "local0",
  "network": "[access]",
  "networkIndex": 1
}
```

/configuration/syslog/test

With the /configuration/syslog/test resource, a **POST** request tests the syslog configuration. The request does not take a request body and does not return a response body.

When you issue a **POST** request for the /configuration/syslog/test resource, the S Series Node tries to send a message to the configured syslog servers. This message can be sent only if syslog logging is enabled, one or more syslog servers are specified, and the minimum severity level for messages to be sent is NOTICE.

If the S Series Node successfully sends the message, this message appears in the S Series Node event log:

```
Syslog test message sent
```

In this case, you can check the specified syslog servers to ensure that they received the test message.

If the test message cannot be sent successfully, the S Series Node returns an error message in response to the **POST** request.

For more information about the /configuration/syslog/test resource, see "Syslog resources" on page 87.

/configuration/time

With the /configuration/time resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/time resource, see "Time resource" on page 87.

/configuration/time properties

The table below describes the properties in /configuration/time resource request and response bodies.

Property name	Data type	Description	Notes
dnsServers	Array	Specifies a comma-separated list of the IP addresses of up to three time servers. At least one time server must be specified.	With a **POST** request, the list of time servers specified in the request body replaces the current list of time servers.

(Continued)

Property name	Data type	Description	Notes
network	String	Specifies the network to be used for communication between the S Series Node and the specified time servers. Valid values are: • **[access]** — Use the access network. • **[management]**— Use the management network. The default is **[access]**. These values are case sensitive.	For the S Series Node to communicate with the specified time servers, the IP mode of the specified network must match the IP mode of the time server IP addresses.
networkIndex	Integer	Specifies whether to use the primary or secondary IPv6 gateway if the network used for communication with the time servers is configured for IPv6. Valid values are: • **1** — Use the primary IPv6 gateway. • **2** — Use the secondary IPv6 gateway. The default is **1**. If the network is configured for IPv4, the only valid value is **1**.	

/configuration/time example

Here's a sample **GET** request that retrieves the time settings for the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/time?prettyprint"
```

Request headers

```
GET /mapi/configuration/time?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 91
```

Response body

```
{
  "timeServers": [
    "10.0.201.65"
  ],
  "network": "[access]",
  "networkIndex": 1
}
```

/configuration/tls

With the /configuration/tls resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.

For more information about the /configuration/tls resource, see "TLS resource" on page 88.

/configuration/tls property

The table below describes the properties in /configuration/tls resource request and response bodies.

Property name	Data type	Description	Notes
minimumTlsVersion	String	Specifies the minimum TLS version that the S Series Node can use for communication with clients. Valid values are: • **TLS 1.0** • **TLS 1.1** • **TLS 1.2** The default is **TLS 1.2**.	

/configuration/tls example

Here's a sample **GET** request that retrieves the minimum TLS version setting for the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/configuration/tls?prettyprint"
```

Request headers

```
GET /mapi/configuration/tls?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 36
```

Response body

```
{
  "minimumTlsVersion": "TLS 1.2"
}
```

/events

With the /events resource, a **GET** request returns a response body that lists event log messages. You can use query parameters to limit the event messages included in the response body.

For more information about the /events resource, see "Events resource" on page 76.

/events properties

The table below describes the properties in /events response bodies. For information about the query parameters mentioned in the table, see "/events query parameters" on page 179.

Property name	Data type	Description	Notes
events	Array	Specifies a comma-separated list of the log messages about events that satisfy the request criteria. Each log message is represented by the properties described in the next table.	The event messages are listed in descending order by the date and time of the event.

(Continued)

Property name	Data type	Description	Notes
eventsAfter	Timestamp	Specifies the value of the **eventsAfter** query parameter included in the **GET** request. If the request did not include the **eventsAfter** parameter, this property is not included in the response body.	
eventsBefore	Timestamp	Specifies the value of the **eventsBefore** query parameter included in the **GET** request. If the request did not include the **eventsBefore** parameter, this property is not included in the response body.	
isTruncated	Boolean	Specifies whether the returned list of event messages is complete. Possible values are: • **true** — The event message list is incomplete. • **false** — The event message list is complete.	
major	Boolean	Specifies the value of the **major** query parameter included in the **GET** request. If the request did not include the **major** parameter, the value of this property is **false**.	
maxEvents	Integer	Specifies the value of the **maxEvents** query parameter included in the **GET** request. If the request did not include the **maxEvents** parameter, the value of this property is **100**.	

(Continued)

Property name	Data type	Description	Notes
scopes	Array	Specifies a comma-separated list of the values specified by the **scopes** query parameter included in the **GET** request. If the request did not include the **scopes** parameter, the value of this property is a comma-separated list of all the possible values for the **scopes** parameter.	
scopeRefs	Array	Specifies a comma-separated list of the values specified by the **scopeRefs** query parameter included in the **GET** request. If the request did not include the **scopeRefs** parameter, this property is not included in the response body.	
scopeSubRefs	Array	Specifies a comma-separated list of the values specified by the **scopeSubRefs** query parameter included in the **GET** request. If the request did not include the **scopeSubRefs** parameter, this property is not included in the response body.	
severity	String	Specifies the value of the **severity** query parameter included in the **GET** request. If the request did not include the **severity** parameter, the value of this property is **NOTICE**.	

The table below describes the properties used to represent an event log message in the array of event messages returned in the response to a **GET** request for the /events resource.

Property name	Data type	Description	Notes
action	String	Specifies the action to take in response to the event.	
eventID	String	Specifies the event message ID.	
localPort	Integer	For events initiated from outside the S Series Node, specifies the port on which on S Series Node was accessed.	
major	Boolean	Specifies whether the event is major. Major events are those that are displayed on the **Dashboard** page in the Management Console. Possible values are: • **true** — The event is major. • **false** — The event is not major.	
message	String	Specifies the full text of the event message.	
originatingIp	String	For events initiated from outside the S Series Node, specifies the IP address from which the request that caused the event was sent.	
reason	String	Specifies the reason for the event message.	

(Continued)

Property name	Data type	Description	Notes
scope	String	Specifies the type of component or activity to which the event applies. Possible values are: • **CERT** — SSL server certificates • **DRIVE** — Data and database drives • **ENCLOSURE** — Enclosures • **FS** — Storage usage • **MAINT** — Maintenance procedures • **SECURITY** — Configuration that requires the security role; failed logins • **SERVER** — Server modules • **SYSTEM** — Configuration that does not require the security role; successful logins; system-initiated events • **UPGRADE** — Software updates and hotfix applications	

(Continued)

Property name	Data type	Description	Notes
scopeRef	Integer	For a scope of DRIVE, specifies the number of the enclosure that contains the data or database drive to which the event applies. For a scope of ENCLOSURE, specifies the number of the enclosure to which the event applies. For a scope of SERVER, specifies the number of the server module to which the event applies. This property is included in the response body only if the scope of the event is DRIVE, ENCLOSURE, or SERVER.	

(Continued)

Property name	Data type	Description	Notes
scopeSubRef	Integer	For a scope of DRIVE, specifies the ID (not number) of the slot containing the data or database drive to which the event applies. For a scope of ENCLOSURE, specifies a value that identifies the enclosure component to which the event applies. For a scope of SERVER, specifies a value that identifies the server module component to which the event applies. This property is included in the response body only if the scope of the event is DRIVE, ENCLOSURE, or SERVER and the event applies to a specific drive, enclosure component, or server module component. For information about the possible values of this property for the ENCLOSURE and SERVER scopes, see "scopes, scopeRefs, and scopeSubRefs query parameters" on page 182.	
severity	String	Specifies the severity of the event. Possible values are: • **NOTICE** • **WARNING** • **ERROR**	

(Continued)

Property name	Data type	Description	Notes
shortName	String	Specifies a brief description of the event.	
timeStamp	Timestamp	Specifies the date and time at which the event occurred, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	
timeStampSuffix	String	Specifies a string that uniquely identifies the date and time at which the event occurred.	
userId	Integer	For events initiated from outside the S Series Node, except login events, specifies the user ID of the user account used to cause the event. For all other events, the value of this property is **0**.	
userName	String	For events initiated from outside the S Series Node, except login events, specifies the username of the user account used to cause the event. For all other events, the value of this property is **[internal]**.	

/events query parameters

You can use query parameters to limit the event messages included in the response to a **GET** request for the /events resource. The query parameters you can use are:

- **maxEvents**
- **eventsAfter**
- **eventsBefore**

- **severity**
- **major**
- **scopes**
- **scopeRefs**
- **scopeSubRefs**

These query parameters can be used alone or in combination with each other.

For more information about query parameters, see "Management API query parameters" on page 61.

maxEvents query parameter

By default, when you issue a **GET** request for the /events resource, the returned list of event messages includes one hundred messages (or fewer if fewer than one hundred satisfy the request criteria). To limit the number of messages in the returned list, you use the **maxEvents** query parameter. Valid values for this parameter are integers in the range one through one hundred.

The response body returned by a **GET** request for the /events resource includes the isTruncated property. The value of this property is **true** if the returned event message list does not include all of the resources that satisfy the request criteria. Otherwise, the value is **false**.

eventsAfter and eventsBefore query parameters

The **eventsAfter** query parameter, used with **GET** requests for the /events resource, limits the list of returned event messages to those about events that happened after a specific date and time. The **eventsBefore** query parameter limits the list of returned event messages to those about events that happened before a specific date and time. A **GET** request for the /events resource must include exactly one of these parameters. You cannot use these parameters in combination with each other.

Valid values for the **eventsAfter** and **eventsBefore** parameters are:

- A date and time, in this format:

  yyyy-MM-dd hh:mm:ss **UTC**

For example:

```
2018-09-20 08:34:29 UTC
```

- The string that uniquely identifies the date and time at which an event occurred. This string is returned as the value of the timestampSuffix property for the event in the response to a **GET** request; for example:

```
"timestampSuffix": "1423074096310.310950000.168"
```

You can use the **eventsAfter** or **eventsBefore** parameter to page through the event messages that satisfy a specified set of criteria. To do this, you repeatedly issue the same request changing only the value of the **eventsAfter** or **eventsBefore** parameter:

- If you're using **eventsAfter**, you change the value to the value of the timestamp or timestampSuffix property for the first (most recent) event message returned by the previous request.

- If you're using **eventsBefore**, you change the value to the value of the timestamp or timestampSuffix property for the last (earliest) event message returned by the previous request.

As long as more messages than are returned satisfy the request criteria, the value of the isTruncated property in the response body is true. When no more messages satisfy the request criteria, the value of the isTruncated property is false.

severity query parameter

The **severity** query parameter, used with **GET** requests for the /events resource, specifies the severity level of the events about which to return event messages. Valid values are:

- **NOTICE** — Return messages about events with any severity level.

- **WARNING** — Return only messages about events with a severity level of warning or error.

- **ERROR** — Return only messages about events with a severity level of error.

The default is **NOTICE**.

These values are case sensitive.

major query parameter

The **major** query parameter, used with **GET** requests for the /events resource, specifies whether to return event messages about all events that satisfy the request criteria or only about major events that satisfy the request criteria. Major events are those that appear on the **Dashboard** page of the HCP S Series Management Console.

Valid values for the major parameter are:

- **true** — Return only messages about major events.
- **false** — Return messages about all events.

The default is **false**.

These values are not case sensitive.

scopes, scopeRefs, and scopeSubRefs query parameters

The **scopes**, **scopeRefs**, and **scopeSubRefs** query parameters limit the event messages returned by a **GET** request for the /events resource to those that apply to one or more specified types of components or activities or, more specifically, to those that apply to one or more particular components or subcomponents.

These query parameters are also used with **GET** requests for the /alerts resource.

scopes
Every event message has a scope that identifies the type of component or activity to which the event applies. You use the **scopes** query parameter in a **GET** request for the /events resource to request messages with specific scopes. Valid values for this parameter are comma-separated lists of one or more of these scopes:

- **CERT** — Returns event messages related to SSL server certificates
- **DRIVE** — Returns event messages related to data and database drives
- **ENCLOSURE** — Returns event messages related to enclosures
- **FS** — Returns event messages related to storage usage
- **MAINT** — Returns event messages related to maintenance procedures

- **SECURITY** — Returns event messages related to configuration activities that require the security role and to failed logins
- **SERVER** — Returns event messages related to server modules
- **SYSTEM** — Returns event messages related to configuration activities that do not require the security role, to successful logins, and to system-initiated events
- **UPGRADE** — Returns event messages related to software updates and hotfix applications

When you include the **scopes** parameter in a **GET** request, the response body includes messages about events that apply to the types of components represented by the scopes specified by the **scopes** parameter and not about events that apply to any other types of components. For example, if the **scopes** parameter specifies ENCLOSURE and SERVER, the response body includes only messages about events that apply to enclosures and servers.

If you don't include the **scopes** parameter in a **GET** request, the response body includes event messages for all scopes.

scopeRefs
You use the **scopeRefs** query parameter in a **GET** request for the /events resource to drill down to particular components within the DRIVE, ENCLOSURE, and SERVER scopes. Valid values for this parameter are comma-separated lists of one or more integers, where:

- For a scope of DRIVE, each integer specifies the number of an enclosure. The response body returned by a **GET** request includes messages for events that apply to drives in the identified enclosures and not to drives in other enclosures.
- For a scope of ENCLOSURE, each integer specifies the number of an enclosure. The response body returned by a **GET** request includes messages for events that apply to the identified enclosures and not to other enclosures.
- For a scope of SERVER, each integer specifies the number of a server module. The response body returned by a **GET** request includes messages for events that apply to the identified server modules and not to other server modules.

If you specify the **scopeRefs** parameter with any other scopes, those scopes are ignored.

If you don't include the **scopeRefs** parameter in a **GET** request, the response body includes event messages for all the components and activities in the specified scopes.

scopeSubRefs
Each hardware component and subcomponent of an S Series Node has an ID. A **GET** request for the /hardware resource returns a list of the S Series Node hardware components and subcomponents. Each component or subcomponent is represented by a set of properties that includes an id property. The value of this property is an integer that's the component or subcomponent ID.

You use the **scopeSubRefs** query parameter in a **GET** request for the /events resource to drill down to data or database drives in particular slots and to particular subcomponents of enclosures and server modules. Valid values for this parameter are comma-separated lists of one or more slot IDs (not numbers) or subcomponent IDs.

The ID of a slot is the slot number minus one. For example, the first slot in an enclosure has a number of one and an ID of zero, the second slot has a number of two and an ID of one, and so forth.

The IDs for the subcomponents of enclosures depend on the enclosure hardware. To know which enclosure subcomponent IDs to use in a **GET** request for the /events resource, you need to check the response body returned by a **GET** request for the /hardware resource.

The table below shows the IDs for the subcomponents of a server module.

ID	Subcomponent
1000	CPUs
1001	Memory
1002	eth0
1003	eth1
1004	eth2
1005	eth3

(Continued)

ID	Subcomponent
1006	bond0
1007	SSDs

If you specify the **scopeSubRefs** parameter with the **scopes** and/or **scopeRefs** parameters, any values of those parameters to which the specified IDs do not apply are ignored.

If you don't include the **scopeSubRefs** parameter in a **GET** request, the messages included int he response body are not limited by slot number or subcomponent ID.

/events example

Here's a sample **GET** request that retrieves a list of event log messages that apply to server module 1.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/events?maxEvents=2
  &eventsAfter=2018-09-20+03:00:00+UTC&scope=SERVER&scopeRefs=1
  &prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/events?maxEvents=2&eventsAfter=2018-09-
20+03:00:00+UTC&scope=SERVER&scopeRefs=1&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 1424
```

Response body

```
{
   "major": false,
   "severity": "NOTICE",
   "scopes": [
      "SERVER"
   ],
   "scopeRefs": [
      1
   ],
   "eventsAfter": "2018-09-20 15:00:00 UTC",
   "isTruncated": false,
   "maxEvents": 2,
   "events": [
      {
         "eventID": 2632,
         "shortName": "Server module unavailable",
         "severity": "ERROR",
         "userId": 0,
         "userName": "[internal]",
         "scope": "SERVER",
         "scopeRef": 1,"major": true,
         "message": "Server module 1 is unavailable.",
         "action": "If this event is unexpected and the server module does not restart
automatically, contact your authorized service provider. Do not try to restart the server
module manually, as that may cause the loss of information needed to diagnose the
problem.",
         "reason": "A server module is unavailable.",
         "timestamp": "2018-09-20 15:18:05 UTC",
         "timestampSuffix": "1423149485295.295462000.28"
      },
      {
         "eventID": 3133,
         "shortName": "Server module shutdown requested",
         "severity": "NOTICE",
         "userId": 1,
         "userName": "admin",
         "scope": "SERVER",
         "scopeRef": 1,
         "major": true,
         "message": "Server module 1 was shut down; reason: Shutting down for
maintenance",
         "action": "No action is required.",
         "reason": "A user shut down a server module.",
         "timestamp": "2018-09-20 15:17:53 UTC",
         "timestampSuffix": "1423149473890.890410000.27"
```

```
            }
        ]
    }
```

/hardware

With the /hardware resource, a **GET** request returns a response body.

For more information about the /hardware resource, see "Hardware resource" on page 76. For information about S Series Node hardware, see "HCP S11 and S31 Node hardware components" on page 19.

/hardware properties

The table below describes the top-level properties in /hardware resource response bodies.

Property name	Data type	Description	Notes
enclosureInfo	Array	Specifies a comma-separated list of the enclosures in the S Series Node, where each enclosure is represented by a set of properties that provide information about that enclosure. For descriptions of these properties, see "Hardware: enclosure high-level properties" on page 210.	
serverModuleInfo	Array	Specifies a comma-separated list of the server modules in the S Series Node, where each module is represented by a set of properties that provide information about that module. For descriptions of these properties, see "Hardware: server module properties" on page 278.	

Hardware: data and database drive properties

The table below describes the properties used to provide detailed information about a data or database drive in a slot in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
ataVersion	String	Unused. This property always has no value.	
capacity	Long	Specifies the drive capacity, in bytes.	
changeTime1	Timestamp	Specifies the date and time at which the value of the state1 property last changed, in this format: *yyyy-MM-dd hh:mm:ss.u* *u* is an integer that, in combination with the specified date and time, makes the change time unique. For example: 2018-09-20 14:39:00.18579 The time is in UTC.	

(Continued)

Property name	Data type	Description	Notes
changeTime2	Timestamp	Specifies the date and time at which the value of the state2 property last changed, in this format: *yyyy-MM-dd hh:mm:ss.u* *u* is an integer that, in combination with the specified date and time, makes the change time unique. For example: 2018-09-22 14:39:00.18579 The time is in UTC.	
error	Boolean	For database drives, specifies whether the drive has an errror-level condition, as indicated by the state (**FAILED** or **MISSING**) or errorsDetected property for the drive. Possible values are: • **true** — For database drives, the drive has an errror-level condition. • **false** — For database drives, the drive does not have an errror-level condition. For data drives, the value of this property may be **true** based on other conditions.	

(Continued)

Property name	Data type	Description	Notes
errorsDetected	Boolean	Specifies whether the drive is unreliable (that is, it has experienced a write error or is in a state in which failure is predicted). Possible values are: • **true** — The drive has experienced a write error or is in a state in which failure is predicted. • **false** — The drive neither has experienced a write error nor is in a state in which failure is predicted.	
evacuate	Boolean	Unused. The value of this property is always **false**.	
failCode	String	For a drive that's marked failed, specifies the reason why the drive is in that condition. Possible values are: • **ADD_FAIL** — The S Series Node could not integrate the drive into the system. • **DRIVE_CORRUPT** — One or more I/O errors occurred on the drive, as a result of which the S Series Node logically removed the drive from the system. • **FORMAT_FAIL** — The S Series Node could not format the drive.	

(Continued)

Property name	Data type	Description	Notes
		- **MAINT_CANCEL** — The drive was a target component for a canceled maintenance procedure. - **MAINT_FAIL** — The drive was a target component for a failed maintenance procedure. - **MAINT_NOT_ACTIVE** — The drive was inserted into its slot while no add or replace drives procedure was active. - **MIRROR_FAULT** — An I/O error occurred on the drive while the S Series Node was protecting the internal database. - **MIRROR_LAST** — The drive is the only remaining active drive in a mirror set, and an attempt was made to repartition the drive. - **MIRROR_SPARE** — The drive is an unused member of a mirror set. - **MISSING** — The drive became unavailable while it was being initialized. - **MOVED** — The drive was moved to its current slot from another slot in the same S Series Node. - **NONE** — The drive is marked failed for an unknown reason.	

(Continued)

Property name	Data type	Description	Notes
		• **REMOVE_FAIL** — During a remove or replace drives procedure, the drive could not be completely removed from the internal database. • **SERIAL_MISMATCH** — The drive serial number does not match the serial number for the drive in the internal database. • **WWID_MISMATCH** — The drive WWID does not match the WWID for the drive in the internal database.	
formFactor	String	For a SAS drive, specifies the physical size of the drive. For a SATA drive, this property is returned with no value.	
fwRev	String	Specifies the revision of the firmware currently installed on the drive.	
fwRevs	Array	Specifies a comma-separated list of the revisions of the firmware that have been installed on the drive, including the current revision. The revisions are listed in order by installation date and time.	

(Continued)

Property name	Data type	Description	Notes
notice	Boolean	For data drives, specifies whether the drive has an informational condition, as indicated by the state (**FAILED** or **MISSING**) or errorsDetected property for the drive. Possible values are: • **true** — For data drives, the drive has an informational condition. • **false** — For data drives, the drive does not have an informational condition. For database drives, the value of this property may be **true** based on other conditions.	
pCode	String	Specifies the Hitachi Vantara part number to use when ordering a replacement drive.	
product	String	Specifies the vendor part number for the drive.	
protocol	String	Specifies the protocol used by the drive. For a SATA drive, the value of this property is always **SATA**. For a SAS drive, the value of this property is always **SAS**.	

(Continued)

Property name	Data type	Description	Notes
reinsert	Boolean	For an unavailable drive, specifies whether the drive will be reincorporated into the storage system when the drive becomes available again. Possible values are: • **true** — The drive will be reincorporated into the storage system. • **false** — The drive will not be reincorporated into the storage system. For an available drive, the value of this property is always **false**.	
rotationRate	Integer	Specifies the disk rotation rate, in RPM.	
sasAddr	String	Specifies the SAS address for the drive.	
sataSpeed	String	Unused. This property always has no value.	
sataVersion	String	Unused. This property always has no value.	
sectorSize	Integer	Specifies the drive sector size, in bytes.	
serial	String	Specifies the vendor serial number for the drive.	
state1	String	Specifies the drive state as it appears to server module 1. Possible value are: • **ADD** — A request has been issued to add the drive to the storage system.	

(Continued)

Property name	Data type	Description	Notes
		• **ADDED** — The drive is part of the storage system. • **ADDING** — The drive is in the process of being added to the storage system. • **DISCOVERED** — The server module has detected that the drive is present. • **FAIL** — A request has been issued to mark the drive failed. • **FAILED** — The drive is marked failed. • **FORMAT** — A request has been issued to format the drive. • **MIRROR** — A request has been issued to add the drive to a mirror set. • **MIRRORED** — The drive is an active member of a mirror set. • **MISSING** — The drive is unavailable. • **NONE** — During a maintenance procedure, the drive was inserted into the slot and then removed. • **PARTITION** — For database drives only, the drive is in the process of being partitioned.	

(Continued)

Property name	Data type	Description	Notes
		- **REMOVE** — A request has been issued to remove the drive from the storage system. - **REMOVED** — The drive has been removed from the storage system. - **REMOVING** — The drive is in the process of being removed from the storage system.	
state2	String	Specifies the drive state as it appears to server module 2. For possible values, see the description of the state1 property.	
type	String	Specifies whether the drive is a data drive or a database drive. Possible values are: - **DATA** — Data drive - **DB** — Database drive	
vendor	String	Specifies the name of the drive vendor.	
wwid	String	Specifies the drive WWID.	

Hardware: enclosure alarm properties

The table below describes the properties used to provide information about an audible alarm in an enclosure in /hardware resource response bodies.

In an S11 or S31 Node enclosure, the audible alarm is not implemented.

Property name	Data type	Description	Notes
code	Integer	Unused. The value of this property is always **1**.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Unused. The value of this property is always **false**.	
fail	Boolean	Unused. The value of this property is always **false**.	
id	Integer	Specifies the component identifier for the alarm. The value of this property is always **172**.	
ident	Boolean	Unused. The value of this property is always **false**.	
location	String	Specifies the name of the alarm.	
muted	Boolean	Unused. The value of this property is always **false**.	
remind	Boolean	Unused. The value of this property is always **false**.	
swap	Boolean	Unused. The value of this property is always **false**.	
urgency	Array	Unused. This property always has no value.	
warning	Boolean	Unused. The value of this property is always **false**.	

Hardware: enclosure current properties

The table below describes the properties used to provide information about the current measured by a current sensor in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the current sensor. Possible values are: • **1** — No error conditions have been detected or requested for the sensor. • **2** — A failure or fault condition has been detected or requested for the sensor. • **3** — A warning or predicted failure condition has been detected or requested for the sensor. • **5** — The sensor is not installed.	
critOver	Boolean	Specifies whether the current is above the critical high-current threshold. Possible values are: • **true** — The current is above the critical high-current threshold. • **false** — The current is not above the critical high-current threshold.	
current	Double	Unused. The value of this property is always **0**.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the current sensor has an error-level condition, as indicated by the code, fail, or critOver property for the sensor. Possible values are: • **true** — The sensor has an error-level condition. • **false** — The sensor does not have an error-level condition.	
fail	Boolean	Specifies whether the current sensor is marked failed. Possible values are: • **true** — The sensor is marked failed. • **false** — The sensor is not marked failed. The sensor is marked failed when the current goes beyond the critical threshold.	
id	Integer	Specifies the component identifier for the current sensor within the enclosure.	
ident	Boolean	Unused. The value of this property is always **0**.	
location	String	Specifies the name of the current sensor.	

(Continued)

Property name	Data type	Description	Notes
swap	Boolean	Specifies whether the current sensor has been removed and then reinserted or replaced. Possible values are: • **true** — The sensor has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The sensor has never been removed and then reinserted or replaced. ○ The sensor was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
warning	Boolean	Specifies whether the current sensor has a warning-level condition, as indicated by the code or warnOver property for the sensor. Possible values are: • **true** — The sensor has a warning-level condition. • **false** — The sensor does not have a warning-level condition.	

(Continued)

Property name	Data type	Description	Notes
warnOver	Boolean	Specifies whether the current is above the warning high-current threshold. Possible values are: • **true** — The current is above the warning high-current threshold. • **false** — The current is not above the warning high-current threshold.	

Hardware: enclosure detail properties

The table below describes the properties used to provide detailed information about an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
baseboardProduct	String	Specifies the vendor part number for the enclosure midplane.	
baseboardSerial	String	Specifies the vendor serial number for the enclosure midplane.	

(Continued)

Property name	Data type	Description	Notes
code	Integer	Specifies the enclosure status code. Possible values are: • **1** — No error conditions have been detected or requested for the enclosure. • **2** — A failure or fault condition has been detected or requested for the enclosure. • **3** — A warning or predicted failure condition has been detected or requested for the enclosure.	
coverOpen	Boolean	Unused. The value of this property is always **false**.	
enclConfigRev	String	Specifies the current revision of the enclosure configuration.	
error	Boolean	Specifies whether the S Series Node currently has any error-level alerts related to the enclosure, as indicated by the code or fail property. Possible values are: • **true** — The S Series Node currently has one or more error-level alerts related to the enclosure. • **false** — The S Series Node currently has no error-level alerts related to the enclosure.	If the HCP S Series software marks the enclosure failed, this property is set to **true** only after 24 hours. If the hardware recognizes that the enclosure has failed, this property is immediately set to **true**.

(Continued)

Property name	Data type	Description	Notes
fail	Boolean	Specifies whether the enclosure or one or more components in the enclosure are marked failed. Possible values are: • **true** — The enclosure or one or more components in the enclosure are marked failed. • **false** — Neither the enclosure nor any of the components in the enclosure are marked failed.	
id	Integer	Specifies the component identifier for the enclosure details.	
ident	Boolean	Specifies whether beaconing is on for the enclosure. Possible values are: • **true** — Beaconing is on for the enclosure. • **false** — Beaconing is off for the enclosure.	
location	String	Unused. This property always has no value.	

(Continued)

Property name	Data type	Description	Notes
notice	Boolean	Specifies whether the S Series Node currently has an information-level condition related to the enclosure, as indicated by the code and fail properties for the enclosure. Possible values are: • **true** — The enclosure has an information-level condition related to the enclosure. • **false** — The enclosure does not have an information-level condition related to the enclosure.	If the HCP S Series software marks the enclosure failed, this property is set to **true** only for the first 24 hours.
predictedFailure	Boolean	Unused. The value of this property is always **false**.	The faultLed property does not seem to exist.
scpAFwRev	String	Unused. The value of this property is always **N/A**.	
scpBFwRev	String	Unused. The value of this property is always **N/A**.	
scpFwRev	String	Unused. The value of this property is always **N/A**.	
swap	Boolean	Unused. The value of this property is always **false**.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the S Series Node currently has any warning-level alerts related to the enclosure. Possible values are: • **true** — The S Series Node currently has one or more warning-level alerts related to the enclosure. • **false** — The S Series Node currently has no warning-level alerts related to the enclosure.	

Hardware: enclosure door properties

The table below describes the properties used to provide information about a cover on an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the cover. Possible values are: • **1** — No error conditions have been detected or requested for the cover. • **5** — The cover is not installed.	
disable	Boolean	Unused. The value of this property is always **false**.	

(Continued)

Property name	Data type	Description	Notes
elementIndex	Integer	Specifies an integer that uniquely identifies the cover: • Possible values for a base enclosure are **170** for the main-bay cover and **171** for the controller-bay cover. • Possible values for an expansion enclosure are **157** for the main-bay cover and **158** for the controller-bay cover.	
elementType	Integer	Specifies the component type for an enclosure cover. The value of this property is always **5**.	
fail	Boolean	Unused. The value of this property is always **false**.	
ident	Boolean	Unused. The value of this property is always **false**.	
open	Boolean	Specifies whether the cover is open. Possible values are: • **true** — The cover is open. • **false** — The cover is closed.	
predictedFailure	Boolean	Unused. The value of this property is always **false**.	
swap	Boolean	Unused. The value of this property is always **false**.	
unlocked	Boolean	Unused. The value of this property is always **false**.	

Hardware: enclosure fan properties

The table below describes the properties used to provide information about an enclosure fan in /hardware resource response bodies.

Property name	Data type	Description	Notes
actualFanSpeed	Integer	Specifies the actual fan speed, in RPMs.	
actualSpeedCode	String	Specifies a code that indicates how fast the fan is rotating relative to the range of possible fan speeds. Possible values are: • **STOPPED 0** — The fan is stopped. • **LOWEST 1** — The fan is rotating at the lowest speed. • **LOWEST_2ND 2** — The fan is rotating at the second-lowest speed. • **LOWEST_3RD 3** — The fan is rotating at the third-lowest speed. • **INTERMEDIATE 4** — The fan is rotating at medium speed. • **HIGHEST_3RD 5** — The fan is rotating at the third-highest speed. • **HIGHEST_2ND 6** — The fan is rotating at the second-highest speed. • **HIGHEST 7** — The fan is rotating at the highest speed.	

(Continued)

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the fan. Possible values are: • **1** — No error conditions have been detected or requested for the fan. • **2** — A failure or fault condition has been detected or requested for the fan. • **3** — A warning or predicted failure condition has been detected or requested for the fan. • **5** — The fan is not installed.	
error	Boolean	Specifies whether the fan has an error-level condition, as indicated by the code or off property for the fan. Possible values are: • **true** — The fan has an error-level condition. • **false** — The fan does not have an error-level condition.	
fail	Boolean	Specifies whether the fan is marked failed. Possible values are: • **true** — The fan is marked failed. • **false** — The fan is not marked failed.	

(Continued)

Property name	Data type	Description	Notes
id	Integer	Specifies the component identifier for the fan within the enclosure.	
ident	Boolean	Unused. The value of this property is always **0**.	
location	String	Specifies the name of the fan.	
off	Boolean	Specifies whether the fan is one or off. Possible values are: • **true** — The fan is off. • **false** — The fan is on.	
swap	Boolean	Specifies whether the fan has been removed and then reinserted or replaced. Possible values are: • **true** — The fan has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The fan has never been removed and then reinserted or replaced. ○ The fan was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the fan has an warning-level condition, as indicated by the code property for the fan. Possible values are: • **true** — The fan has an warning-level condition. • **false** — The fan does not have an warning-level condition.	

Hardware: enclosure high-level properties

The table below describes the properties used to provide high-level information about an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
alarms	Array	Specifies a comma-separated list of the audible alarms in the enclosure, where each alarm is represented by a set of properties that provide information about that alarm. For descriptions of these properties, see "Hardware: enclosure alarm properties" on page 196.	In an S Series Node enclosure, only the enclosure itself has an audible alarm.
currents	Array	Specifies a comma-separated list of the currents measured by current sensors in the enclosure, where each current is represented by a set of properties that provide information about that current. For descriptions of these properties, see "Hardware: enclosure current properties" on page 198.	

(Continued)

Property name	Data type	Description	Notes
doors	Array	Specifies a comma-separated list of the covers on the enclosure, where each cover is represented by a set of properties that provide information about that cover. For descriptions of these properties, see "Hardware: enclosure door properties" on page 205.	
enclosures	Array	Specifies a comma-separated list of objects, where each object is represented by a set of properties that provide detailed information about the enclosure. For descriptions of these properties, see "Hardware: enclosure detail properties" on page 201.	An S Series Node enclosure has only one set of enclosure detail properties.
enclosureServices	Array	Specifies a comma-separated list of the enclosure service components, where each service component is represented by a set of properties that provide information about that component. For descriptions of these properties, see "Hardware: enclosure service properties" on page 231.	In a base enclosure, an enclosure service component is a server module. In an expansion enclosure, an enclosure service component is an I/O module.

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the S Series Node currently has any error-level alerts related to the enclosure. Possible values are: • **true** — The S Series Node currently has one or more error-level alerts related to the enclosure. • **false** — The S Series Node currently has no error-level alerts related to the enclosure.	
fans	Array	Specifies a comma-separated list of the fans in the enclosure, where each fan is represented by a set of properties that provide information about that fan. For descriptions of these properties, see "Hardware: enclosure fan properties" on page 207.	
fwRev	String	Specifies the revision of the firmware currently installed on the enclosure.	
fwRevs	Array	Specifies a comma-separated list of the revisions of the firmware that have been installed on the enclosure, including the current revision. The revisions are listed in order by installation date and time.	
id	Integer	Specifies the component identifier for the enclosure.	

(Continued)

Property name	Data type	Description	Notes
leds	Array	Specifies a comma-separated list of the LEDs on the enclosure, where each LED is represented by a set of properties that provide information about that LED.	This property is used internally only.
ledStates	Array	Unused. This property always has no value.	
lockdownReason	String	Unused. The value of this property is always **N/A**.	
notice	Boolean	Specifies whether the S Series Node currently has any informational alerts related to the enclosure. Possible values are: • **true** — The S Series Node currently has one or more informational alerts related to the enclosure. • **false** — The S Series Node currently has no informational alerts related to the enclosure.	
pCode	String	Specifies the Hitachi Vantara part number to use when ordering a replacement enclosure.	
powerSupplies	Array	Specifies a comma-separated list of the power supplies in the enclosure, where each power supply is represented by a set of properties that provide information about that power supply. For descriptions of these properties, see "Hardware: enclosure power supply properties" on page 218.	

(Continued)

Property name	Data type	Description	Notes
product	String	Specifies the vendor part number for the enclosure.	
sasConnectors	Array	Specifies a comma-separated list of the SAS connectors in the enclosure, where each connector is represented by a set of properties that provide information about that connector. For descriptions of these properties, see "Hardware: enclosure SAS connector properties" on page 225.	
sasExpanders	Array	Specifies a comma-separated list of the SAS expanders in the enclosure, where each expander is represented by a set of properties that provide information about that expander. For descriptions of these properties, see "Hardware: enclosure SAS expander properties" on page 228	
sbbPowerOnState	String	Unused. The value of this property is always **N/A**.	
serial	String	Specifies the vendor serial number for the enclosure.	
sideplanes	Array	Specifies a comma-separated list of the SAS expanders in the enclosure, where each expander is represented by a set of properties that provide information about that expander. For descriptions of these properties, see "Hardware: enclosure sideplane properties" on page 235.	

(Continued)

Property name	Data type	Description	Notes
slots	Array	Specifies a comma-separated list of the slots in the enclosure, where each slot is represented by a set of properties that provide information about that slot. For descriptions of these properties, see "Hardware: enclosure slot properties" on page 238.	
state1	String	Specifies the status of the enclosure as seen by server module 1. Possible values are: • **ADDED** — The enclosure is functioning normally. • **DISCOVERED** — Upon restarting during a replace enclosure maintenance procedure, the server module has detected the presence of the new enclosure. • **FAILED** — The enclosure is in a failed state. • **MISSING** — The enclosure is unavailable. • **NONE** — The server module cannot detect the presence of the enclosure. • **REMOVED** — The enclosure has been removed from the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
		• **REMOVING** — The enclosure is in the process of being removed from the S Series Node.	
state2	String	Specifies the status of the enclosure as seen be server module 2. For possible values, see the description of the state1 property above.	
status	String	Specifies the status of the enclosure. Possible values are: • **AVAILABLE** — Both server modules can see the enclosure. • **DEGRADED** — Only one server module can see the enclosure. • **UNAVAILABLE** — Neither server module can see the enclosure. • **UNKNOWN** — The S Series Node cannot determine the state of the enclosure.	
temperatures	Array	Specifies a comma-separated list of the temperatures measured by temperature sensors in the enclosure, where each temperature is represented by a set of properties that provide information about that temperature. For descriptions of these properties, see "Hardware: enclosure temperature properties" on page 245.	

(Continued)

Property name	Data type	Description	Notes
vendor	String	Specifies the name of the enclosure vendor.	
voltages	Array	Specifies a comma-separated list of the voltages measured by voltage sensors in the enclosure, where each voltage is represented by a set of properties that provide information about that voltage. For descriptions of these properties, see "Hardware: enclosure voltage properties" on page 251.	
warning	Boolean	Specifies whether the S Series Node currently has any warning-level alerts related to the enclosure. Possible values are: • **true** — The S Series Node currently has one or more warning-level alerts related to the enclosure. • **false** — The S Series Node currently has no warning-level alerts related to the enclosure.	
wwid	String	Specifies the enclosure WWID.	

Hardware: enclosure power supply properties

The table below describes the properties used to provide information about a power supply in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
acFail	Boolean	Specifies whether the power supply is receiving any AC power. Possible values are: • **true** — The power supply is receiving AC power. • **false** — The power supply is not receiving any AC power. Either the power cord is damaged or not plugged in, or the power is not switched on.	
code	Integer	Specifies the status code for the power supply. Possible values are: • **1** — No error conditions have been detected or requested for the power supply. • **2** — A failure or fault condition has been detected or requested for the power supply. • **3** — A warning or predicted failure condition has been detected or requested for the power supply. • **5** — The power supply is not installed.	

(Continued)

Property name	Data type	Description	Notes
dcFail	Boolean	Specifies whether the power supply is generating adequate DC power. Possible values are: • **true** — The power supply is generating adequate DC power. • **false** — The power supply is not generating adequate DC power.	
dcOverCurrent	Boolean	Specifies whether the DC current is above the high-current threshold for the power supply. Possible values are: • **true** — The DC current is above the high-current threshold. • **false** — The DC current is not above the high-current threshold.	
dcOverVoltage	Boolean	Specifies whether the DC voltage is above the high-voltage threshold for the power supply. Possible values are: • **true** — The DC voltage is above the high-voltage threshold. • **false** — The DC voltage is not above the high-voltage threshold.	

(Continued)

Property name	Data type	Description	Notes
dcUnderVoltage	Boolean	Specifies whether the DC voltage is below the low-voltage threshold for the power supply. Possible values are: • **true** — The DC voltage is below the low-voltage threshold. • **false** — The DC voltage is not below the low-voltage threshold.	
error	Boolean	Specifies whether the power supply has an error-level condition, as indicated by the code, fail, acFail, dcFail, overTempFail, dcOverVoltage, dcUnderVoltage, dcOverCurrent, or off property for the power supply. Possible values are: • **true** — The power supply has an error-level condition. • **false** — The power supply does not have an error-level condition.	

(Continued)

Property name	Data type	Description	Notes
fail	Boolean	Specifies whether the power supply is marked failed. Possible values are: • **true** — The power supply is marked failed. • **false** — The power supply is not marked failed. The power supply is marked failed if the value of the fail, acFail, or dcFail property for the power supply is **true**.	
fwRev	String	Specifies the revision of the firmware currently installed on the power supply.	
id	Integer	Specifies the component identifier for the power supply within the enclosure.	
ident	Boolean	Specifies whether beaconing is on for the power and cooling module that contains the power supply. Possible values are: • **true** — Beaconing is on for the module. • **false** — Beaconing is off for the module.	
location	String	Specifies the name of the power supply.	
off	Boolean	Unused. The value of this property is always **false**.	

(Continued)

Property name	Data type	Description	Notes
overTempFail	Boolean	Specifies whether the power supply is marked failed due to a temperature above the critical high-temperature threshold for the power supply. Possible values are: • **true** — The power supply is marked failed due to a critically high temperature. • **false** — The power supply is not marked failed due to a critically high temperature.	
overTempWarn	Boolean	Specifies whether the power supply temperature is above the warning high-temperature threshold. Possible values are: • **true** — The power supply temperature is above the warning high-temperature threshold. • **false** — The power supply temperature is not above the warning high-temperature threshold.	
pCode	String	Specifies the Hitachi Vantara part number to use when ordering a replacement power supply.	
product	String	Specifies the vendor part number for the power supply.	
rev	String	Specifies the power supply hardware revision to use when requesting service.	

(Continued)

Property name	Data type	Description	Notes
serial	String	Specifies the vendor serial number for the power supply.	
supplierProduct	String	Specifies the original-vendor part number for the power supply.	
supplierRev	String	Specifies the original-vendor hardware revision for the power supply.	
supplierSerial	String	Specifies the original-vendor serial number for the power supply.	
swap	Boolean	Specifies whether the power supply has been removed and then reinserted or replaced. Possible values are: • **true** — The power supply has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The power supply has never been removed and then reinserted or replaced. ○ The power supply was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
vendor	String	Specifies the name of the power supply vendor.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the power supply has a warning-level condition, as indicated by the code or overTempWarn property for the power supply. Possible values are: • **true** — The power supply has a warning-level condition. • **false** — The power supply does not have a warning-level condition.	

Hardware: enclosure SAS connector properties

The table below describes the properties used to provide information about a SAS connector in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the SAS connector. Possible values are: • **1** — No error conditions have been detected or requested for the connector. • **2** — A failure or fault condition has been detected or requested for the connector. • **3** — A warning or predicted failure condition has been detected or requested for the connector. • **5** — The connector is not installed.	
connectorPhyLink	Integer	Specifies a value of **255**, indicating that the SAS connector includes all physical links in the connector.	
connectorType	Integer	Specifies the SAS connector type. The value of this property is always **5**, indicating that the connector is a SAS 4x receptacle.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the SAS connector has an error-level condition, as indicated by the code or fail property for the connector. Possible values are: - **true** — The connector has an error-level condition. - **false** — The connector does not have an error-level condition.	
fail	Boolean	Specifies whether the SAS connector is marked failed. Possible values are: - **true** — The connector is marked failed. - **false** — The connector is not marked failed.	
id	Integer	Specifies the component identifier for the SAS connector within the enclosure.	
ident	Boolean	Unused. The value of this property is always **0**.	
location	String	Specifies the name of the SAS connector.	

(Continued)

Property name	Data type	Description	Notes
swap	Boolean	Specifies whether the SAS connector has been removed and then reinserted or replaced. Possible values are: • **true** — The connector has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The connector has never been removed and then reinserted or replaced. ○ The connector was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
warning	Boolean	Specifies whether the SAS connector has a warning-level condition, as indicated by the code property for the connector. Possible values are: • **true** — The connector has a warning-level condition. • **false** — The connector does not have a warning-level condition.	

Hardware: enclosure SAS expander properties

The table below describes the properties used to provide information about a SAS expander in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
arrayIndex	Array	Specifies a comma-separated list of the slots associated with the SAS expander, where each slot is identified by its component identifier.	
code	Integer	Specifies the status code for the SAS expander. Possible values are: • **1** — No error conditions have been detected or requested for the expander. • **2** — A failure or fault condition has been detected or requested for the expander. • **3** — A warning or predicted failure condition has been detected or requested for the expander. • **5** — The expander is not installed.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the SAS expander has an error-level condition, as indicated by the code or fail property for the expander. Possible values are: • **true** — The expander has an error-level condition. • **false** — The expander does not have an error-level condition.	
fail	Boolean	Specifies whether the SAS expander is marked failed. Possible values are: • **true** — The expander is marked failed. • **false** — The expander is not marked failed.	
fwRev	String	Specifies the revision of the firmware currently installed on the SAS expander.	
id	Integer	Specifies the component identifier for the SAS expander within the enclosure.	
ident	Boolean	Unused. The value of this property is always **0**.	
location	String	Specifies the name of the SAS expander.	
product	String	Specifies the vendor part number for the SAS expander.	
sasAddr	String	Specifies the SAS address for the SAS expander.	

(Continued)

Property name	Data type	Description	Notes
serial	String	Specifies the vendor serial number for the SAS expander.	
swap	Boolean	Specifies whether the SAS expander has been removed and then reinserted or replaced. Possible values are: • **true** — The expander has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The expander has never been removed and then reinserted or replaced. ○ The expander was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
warning	Boolean	Specifies whether the SAS expander has a warning-level condition, as indicated by the code property for the expander. Possible values are: • **true** — The expander has a warning-level condition. • **false** — The expander does not have a warning-level condition.	

Hardware: enclosure service properties

The table below describes the properties used to provide information about an enclosure service component in /hardware resource response bodies.

In a base enclosure, an enclosure service component is a server module. In an expansion enclosure, an enclosure service component is an I/O module.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the enclosure service component. Possible values are: • **1** — No error conditions have been detected or requested for the component. • **2** — A failure or fault condition has been detected or requested for the component. • **3** — A warning or predicted failure condition has been detected or requested for the component. • **5** — The component is not installed.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the enclosure service component has an error-level condition, as indicated by the code or fail property for the component. Possible values are: • **true** — The component has an error-level condition. • **false** — The component does not have an error-level condition.	
fail	Boolean	Specifies whether the enclosure service component is marked failed. Possible values are: • **true** — The component is marked failed. • **false** — The component is not marked failed.	
fruProduct	String	Specifies the vendor part number for the enclosure service component.	
fruSerial	String	Specifies the vendor serial number for the enclosure service component.	
fwRev	String	Specifies the revision of the firmware currently installed on the enclosure service component.	
id	Integer	Specifies the component identifier for the enclosure service component within the enclosure.	

(Continued)

Property name	Data type	Description	Notes
ident	Boolean	Specifies whether beaconing is on for the enclosure service component. Possible values are: • **true** — Beaconing is on for the component. • **false** — Beaconing is off for the component.	
location	String	Specifies the name of the enclosure service component.	
notice	Boolean	Specifies whether the enclosure service component has an information-level condition, as indicated by the ident property for the enclosure service component. Possible values are: • **true** — The enclosure service component has an information-level condition. • **false** — The enclosure service component does not have an information-level condition.	

(Continued)

Property name	Data type	Description	Notes
report	Boolean	Specifies whether the enclosure service component is the one that generated the current enclosure status values. Possible values are: • **true** — The component generated the enclosure status values. • **false** — The component did not generate the enclosure status values.	
swap	Boolean	Specifies whether the enclosure service component has been removed and then reinserted or replaced. Possible values are: • **true** — The component has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The component has never been removed and then reinserted or replaced. ○ The component was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the enclosure service component has a warning-level condition, as indicated by the code property for the component. Possible values are: • **true** — The component has a warning-level condition. • **false** — The component does not have a warning-level condition.	

Hardware: enclosure sideplane properties

The table below describes the properties used to provide information about a SAS expander in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
cableFault	Boolean	Specifies whether the internal cable connected to the SAS expander is functioning normally. Possible values are: • **true** — The cable is functioning normally. • **false** — The cable is not functioning normally.	
cableFaultRqst	Boolean	Unused. The value of this property is always **false**.	

(Continued)

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the SAS expander. Possible values are: • **1** — No error conditions have been detected or requested for the expander. • **2** — A failure or fault condition has been detected or requested for the expander. • **3** — A warning or predicted failure condition has been detected or requested for the expander. • **5** — The expander is not installed.	
coverRemoved	Boolean	Unused. The value of this property is always **false**.	
elementIndex	Integer	Specifies an integer that uniquely identifies the SAS expander: • Possible values for a base enclosure are integers in the range **235** through **246**, which correspond to SAS expanders 0 through 11. • Possible values for an expansion enclosure are integers in the range **131** through **138**, which correspond to SAS expanders 0 through 7.	SAS expanders 0 through 7 are in the main bay of the enclosure. SAS expanders 8 through 11 are in the personality modules in a base enclosure.

(Continued)

Property name	Data type	Description	Notes
elementType	Integer	Specifies the component type for a SAS expander. The value of this property is always **144**.	
failRqst	Boolean	Unused. The value of this property is always **false**.	
fault	Boolean	Specifies whether the SAS expander is functioning normally. Possible values are: • **true** — The SAS expander is functioning normally. • **false** — The SAS expander is not functioning normally.	
ident	Boolean	Specifies whether beaconing is on for the SAS expander. Possible values are: • **true** — Beaconing is on for the expander. • **false** — Beaconing is off for the expander.	
powered	Boolean	Specifes whether the SAS expander has power. Possible values are: • **true** — The SAS expander has power. • **false** — The SAS expander does not have power.	
powerFault	Boolean	Unused. The value of this property is always **false**.	
predictedFailure	Boolean	Unused. The value of this property is always **false**.	

(Continued)

Property name	Data type	Description	Notes
swap	Boolean	Specifies whether the SAS expander has been removed and then reinserted or replaced. Possible values are: • **true** — The expander has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The expander has never been removed and then reinserted or replaced. ○ The expander was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	

Hardware: enclosure slot properties

The table below describes the properties used to provide high-level information about a slot in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
attachSasAddr	String	If the slot contains a drive, specifies the SAS address of the SAS expander to which the slot is connected. If the slot is empty, the value of this property is **0x0**.	

(Continued)

Property name	Data type	Description	Notes
code	Integer	Specifies the slot status code. Possible values are: - **1** — No error conditions have been detected or requested for the slot. - **2** — A failure or fault condition has been detected or requested for the slot. - **3** — A warning or predicted failure condition has been detected or requested for the slot. - **5** — The slot is not installed.	
consistencyCheck	Boolean	Unused. The value of this property is always **false**.	
deviceOff	Boolean	Unused. The value of this property is always **false**.	
doNotRemove	Boolean	Unused. The value of this property is always **false**.	
drive	Object	Specifies a set of properties that provide information about the drive in the slot. For descriptions of these properties, see "Hardware: data and database drive properties" on page 188.	This property is not returned if the slot is empty.

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the slot has an error-level condition, as indicated by the code or fail property for the slot or by the errorsDetected property for the drive in the slot. Possible values are: • **true** — The slot has an error-level condition. • **false** — The slot does not have an error-level condition.	
fail	Boolean	If the slot contains a drive, specifies whether the drive is marked failed. Possible values are: • **true** — The drive is marked failed. • **false** — The drive is not marked failed. If the slot is empty, the value of this property is **false**.	
hotSpare	Boolean	Unused. The value of this property is always **false**.	
id	Integer	Specifies the component identifier for the slot within the enclosure. The value of this property is the slot number. For example, the value of this property for the seventh slot from the left in the first row of slots in the main bay of an enclosure is **6**.	In an S Series Node, slot numbering starts at zero.

(Continued)

Property name	Data type	Description	Notes
ident	Boolean	Specifies whether beaconing is on for the slot. Possible values are: • **true** — Beaconing is on for the slot. • **false** — Beaconing is off for the slot.	
inCriticalArray	Boolean	Unused. The value of this property is always **false**.	
inFailedArray	Boolean	Unused. The value of this property is always **false**.	
location	String	Specifies the name of the slot, in this format: Slot*n* *n* is a four-character string consisting of some number of spaces followed by the slot number. For example, the value of this property for the seventh slot from the left in the first row of slots in the main bay of an enclosure is **Slot 6**, where **Slot** and **6** are separated by three spaces.	In an S Series Node, slot numbering starts at zero.
maintProcedure	Boolean	Specifies whether the slot is currently selected for a maintenance procedure. Possible values are: • **true** — The slot is selected for a maintenance procedure. • **false** — The slot is not selected for a maintenance procedure.	

(Continued)

Property name	Data type	Description	Notes
notice	Boolean	Specifies whether the slot has an information-level condition, as indicated by the code or fail property for the slot. Possible values are: • **true** — The slot has an information-level condition. • **false** — The slot does not have an information-level condition.	
prepareForRemoval	Boolean	Unused. The value of this property is always **false**.	
readyToInsert	Boolean	Unused. The value of this property is always **false**.	
rebuildRemap	Boolean	Unused. The value of this property is always **false**.	
rebuildRemapAbort	Boolean	Unused. The value of this property is always **false**.	
reservedDevice	Boolean	Unused. The value of this property is always **false**.	
sasAddr	String	If the slot contains a drive, specifies the SAS address for the drive. If the slot is empty, the value of this property is **0x0**.	
slotNumber	Integer	Specifies the slot number. For example, the value of this property for the seventh slot from the left in the first row of slots in the main bay of an enclosure is **6**.	In an S Series Node, slot numbering starts at zero.

(Continued)

Property name	Data type	Description	Notes
status	String	Specifies the status of the slot. Possible values are: • **AVAILABLE** — The drive in the slot is healthy and available. • **FAILED** — The drive in the slot is marked failed. • **NONE** — The slot is empty. • **UNAVAILABLE** — The drive in the slot is unavailable.	
swap	Boolean	Specifies whether the drive in the slot has been removed and then reinserted or replaced. Possible values are: • **true** — The drive has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The drive has never been removed and then reinserted or replaced. ○ The drive was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the slot has a warning-level condition, as indicated by the code property for the slot. Possible values are: • **true** — The slot has a warning-level condition. • **false** — The slot does not have a warning-level condition.	
wwid	String	If the slot contains a drive, specifies the WWID of the drive.	This property is not returned if the slot is empty.

Hardware: enclosure temperature properties

The table below describes the properties used to provide information about the temperature measured by a temperature sensor in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the temperature sensor. Possible values are: • **1** — No error conditions have been detected or requested for the sensor. • **2** — A failure or fault condition has been detected or requested for the sensor. • **3** — A warning or predicted failure condition has been detected or requested for the sensor. • **5** — The sensor is not installed.	
critOver	Boolean	Specifies whether the temperature is above the critical high-temperature threshold. Possible values are: • **true** — The temperature is above the critical high-temperature threshold. • **false** — The temperature is not above the critical high-temperature threshold.	

(Continued)

Property name	Data type	Description	Notes
critOverThresh	Double	Specifies the critical high-temperature threshold, in degrees Celsius.	
critUnder	Boolean	Specifies whether the temperature is below the critical low-temperature threshold. Possible values are: • **true** — The temperature is below the critical low-temperature threshold. • **false** — The temperature is not below the critical low-temperature threshold.	
critUnderThresh	Double	Specifies critical low-temperature threshold, in degrees Celsius.	
error	Boolean	Specifies whether the temperature sensor has an error-level condition, as indicated by the code, fail, critOver, or critUnder property for the sensor. Possible values are: • **true** — The sensor has an error-level condition. • **false** — The sensor does not have an error-level condition.	

(Continued)

Property name	Data type	Description	Notes
fail	Boolean	Specifies whether the temperature sensor is marked failed. Possible values are: • **true** — The sensor is marked failed. • **false** — The sensor is not marked failed. The sensor is marked failed when the temperature goes beyond a critical threshold.	
id	Integer	Specifies the component identifier for the temperature sensor within the enclosure.	
ident	Boolean	Unused. The value of this property is always **false**.	
location	String	Specifies the name of the temperature sensor.	

(Continued)

Property name	Data type	Description	Notes
swap	Boolean	Specifies whether the temperature sensor has been removed and then reinserted or replaced. Possible values are: • **true** — The sensor has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The sensor has never been removed and then reinserted or replaced. ○ The sensor was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
temperature	Integer	Specifies the actual temperature measured by the temperature sensor, in degrees Celsius.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the temperature sensor has an warning-level condition, as indicated by the code, warnOver, or warnUnder property for the sensor. Possible values are: • **true** — The sensor has an warning-level condition. • **false** — The sensor does not have an warning-level condition.	
warnOver	Boolean	Specifies whether the temperature is above the warning high-temperature threshold. Possible values are: • **true** — The temperature is above the warning high-temperature threshold. • **false** — The temperature is not above the warning high-temperature threshold.	
warnOverThresh	Double	Specifies the warning high-temperature threshold, in degrees Celsius.	

(Continued)

Property name	Data type	Description	Notes
warnUnder	Boolean	Specifies whether the temperature is below the warning low-temperature threshold. Possible values are: • **true** — The temperature is below the warning low-temperature threshold. • **false** — The temperature is not below the warning low-temperature threshold.	
warnUnderThresh	Double	Specifies the warning low-temperature threshold, in degrees Celsius.	

Hardware: enclosure voltage properties

The table below describes the properties used to provide information about the voltage measured by a voltage sensor in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the voltage sensor. Possible values are: • **1** — No error conditions have been detected or requested for the sensor. • **2** — A failure or fault condition has been detected or requested for the sensor. • **3** — A warning or predicted failure condition has been detected or requested for the sensor. • **5** — The sensor is not installed.	
critOver	Boolean	Specifies whether the voltage is above the critical high-voltage threshold. Possible values are: • **true** — The voltage is above the critical high-voltage threshold. • **false** — The voltage is not above the critical high-voltage threshold.	
critOverThresh	Double	Specifies the critical high-voltage threshold as a percent, from zero to 100, above the expected voltage.	

(Continued)

Property name	Data type	Description	Notes
critUnder	Boolean	Specifies whether the voltage is below the critical low-voltage threshold. Possible values are: • **true** — The voltage is below the critical low-voltage threshold. • **false** — The voltage is not below the critical low-voltage threshold.	
critUnderThresh	Double	Specifies the critical low-voltage threshold as a percent, from zero to 100, below the expected voltage.	
error	Boolean	Specifies whether the voltage sensor has an error-level condition, as indicated by the code, fail, critOver, or critUnder property for the sensor. Possible values are: • **true** — The sensor has an error-level condition. • **false** — The sensor does not have an error-level condition.	
fail	Boolean	Specifies whether the voltage sensor is marked failed. Possible values are: • **true** — The sensor is marked failed. • **false** — The sensor is not marked failed. The sensor is marked failed when the voltage goes beyond a critical threshold.	

(Continued)

Property name	Data type	Description	Notes
id	Integer	Specifies the component identifier for the voltage sensor within the enclosure.	
ident	Boolean	Unused. The value of this property is always **false**.	
location	String	Specifies the name of the voltage sensor.	
swap	Boolean	Specifies whether the voltage sensor has been removed and then reinserted or replaced. Possible values are: • **true** — The sensor has been removed and then reinserted or replaced. This is a transient value. • **false** — Either of these: ○ The sensor has never been removed and then reinserted or replaced. ○ The sensor was removed and then reinserted or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
voltage	Double	Specifies the actual voltage measured by the voltage sensor, in volts.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the voltage sensor has an warning-level condition, as indicated by the code, warnOver, or warnUnder property for the sensor. Possible values are: • **true** — The sensor has an warning-level condition. • **false** — The sensor does not have an warning-level condition.	
warnOver	Boolean	Specifies whether the voltage is above the warning high-voltage threshold. Possible values are: • **true** — The voltage is above the warning high-voltage threshold. • **false** — The voltage is not above the warning high-voltage threshold.	
warnOverThresh	Double	Specifies the warning high-voltage threshold as a percent, from zero to 100, above the expected voltage.	
warnUnder	Boolean	Specifies whether the voltage is below the warning low-voltage threshold. Possible values are: • **true** — The voltage is below the warning low-voltage threshold. • **false** — The voltage is not below the warning low-voltage threshold.	

(Continued)

Property name	Data type	Description	Notes
warnUnderThresh	Double	Specifies the warning low-voltage threshold as a percent, from zero to 100, below the expected voltage.	

Hardware: server module bonded network interface properties

The table below describes the properties used to provide information about a bonded network interface for a server module in /hardware resource response bodies.

The only network that uses a bonded interface is the access network.

Property name	Data type	Description	Notes
activeSlave	String	Specifies which of the Ethernet interfaces included in the bonded network interface is currently active. For a bonded interface in active-backup mode, possible values are **eth0** and **eth2**. For a bonded interface in IEEE 802.3ad mode, this property is always returned with no value.	
error	Boolean	Specifies whether the interface has an error condition, as indicated by the value of the ok property for the interface. Possible values are: • **true** — The interface has an error condition. • **false** — The interface does not have an error condition.	

(Continued)

Property name	Data type	Description	Notes
mode	String	Specifies the actual bonding mode being used on the network connected to the interface. Possible values are **802.3ad** and **active-backup**.	
mtu	Long	Specifies the actual maximum transmission unit (MTU) being used on the network connected to the interface. Possible values are **9000** and **1500**.	
name	String	Specifies the name of the interface. The only possible value is **bond0**.	
ok	Boolean	Specifies whether the interface is physically connected to the network and is functional . Possible values are: • **true** — The interface is physically connected to the network and is functional. • **false** — The interface either is not physically connected to the network or is physically connected to the network but not functional .	

(Continued)

Property name	Data type	Description	Notes
slaves	Array	Specifies a comma-separated list of the Ethernet interfaces included in the bonded network interface, where each Ethernet interface is represented by a set of properties that provide information about that interface. For descriptions of these properties, see "Hardware: server module Ethernet interface properties" on page 265.	
type	String	Specifies the type of network that's using the interface. The only possible value is **ACCESS**.	
warning	Boolean	Specifies whether the actual speed, MTU, and bonding mode on the network connected to the interface match the speed, MTU, and bonding mode configured for the network. Possible values are: • **true** — The actual speed, actual MTU, and actual bonding mode match the configured speed, MTU, and bonding mode. • **false** — The actual speed, MTU, or bonding mode does not match the configured speed, MTU, or bonding mode.	

Hardware: server module core hardware properties

The table below describes the properties used to provide information about the core hardware in a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
biosDate	String	Specifies the release date of the server module BIOS, in this format: *MM/dd/yyyy* For example: 11/13/2018	
biosFwRev	String	Specifies the revision of the BIOS firmware currently installed on the server module.	
biosVendor	String	Specifies the name of the server module BIOS vendor.	
bootTime	Long	Specifies the date and time the HCP S Series software last started on the server module, in seconds since January 1, 1970, at 00:00:00.	
cpus	Array	Specifies a comma-separate list of the physical CPUs in the server module.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the server module has an error-level condition, as indicated by the loadAvgError, badDnsServers, or ntpServer property for the server module. Possible values are: • **true** — The server module has an error-level condition. • **false** — The server module does not have an error-level condition.	
failedDnsServer Connections	Array	Unused. This property always has no value.	
fifteenMinuteLoad	String	Specifies the average workload on the server module over the past fifteen minutes.	
fiveMinuteLoad	String	Specifies the average workload on the server module over the past five minutes.	
freeSwap	Long	Specifies the unused amount of the storage allocated for the server module to use for page swapping, in bytes.	
hwSetupToolVersion	String	Specifies the version of the Hardware Setup Tool that was most recently run on the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
ident	Boolean	Specifies whether beaconing is on for the server module. Possible values are: • **true** — Beaconing is on for the server module. • **false** — Beaconing is off for the server module.	
lastUpdate	Long	Specifies the date and time of the last update to the core hardware information for the server module, in seconds since January 1, 1970, at 00:00:00.	
loadAvgError	Boolean	Specifies whether the load on the system is too high. The load is too high if the one-minute load average divided by the number of CPUs in the server module is greater than 50. Possible values are: • **true** — The load average is too high. • **false** — The load average is not too high.	
minuteLoad	String	Specifies the average workload on the server module over the past minute.	
ntpServer	String	Specifies the IP address of the time server to which the time on the server module is currently synced.	
pageSwapIn	Long	Specifies the number of pages currently swapped into memory.	

(Continued)

Property name	Data type	Description	Notes
pageSwapOut	Long	Specifies the number of pages currently swapped out of memory.	
pCode	String	Specifies the Hitachi Vantara part number to use when ordering a replacement server module.	
product	String	Specifies the vendor part number for the server module.	
rev	String	Specifies the server module hardware revision.	
serial	String	Specifies the vendor serial number for the server module.	
swapError	Boolean	Specifies whether the rate of page swapping in the server module is too high. The rate is too high if the values of the pageSwapIn and pageSwapOut properties are both greater than ten. Possible values are: • **true** — The page swap rate is too high. • **false** — The page swap rate is not too high.	
totalMemory	Long	Specifies the total amount of RAM in the server module.	
totalSwap	Long	Specifies the total amount of storage allocated for the server module to use for page swapping, in bytes.	

(Continued)

Property name	Data type	Description	Notes
uptime	Long	Specifies the number of seconds that have passed since the S Series Node was last powered on.	
usedMemory	Long	Specifies the amount of the server module RAM that's currently in use.	
vendor	String	Specifies the name of the server module vendor.	
warning	Boolean	Unused. The value of this property is always **false**.	

Hardware: server module disk properties

The table below describes the properties used to provide information about a disk in a server module in /hardware resource response bodies.

A disk is either an SSD in the server module or a USB flash drive attached to the server module.

Property name	Data type	Description	Notes
capacity	Long	Specifies the capacity of the disk, in bytes.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	For an SSD, specifies whether the disk has an error-level condition, as indicated by the state of the mirror set that includes the disk. Possible values are: • **true** — The disk has an error-level condition. • **false** — The disk does not have an error-level condition. For a USB flash drive, the value of this property is always **false**. For information about mirror set states, see "Hardware: server module mirror set properties" on page 273.	
fwRev	String	Specifies the revision of the firmware currently installed on the disk.	
id	Integer	Specifies the disk ID. For SSD 0, the value of this property is always **1007**. For SSD 1, the value of this property is always **1008**.	This property is not returned for a USB flash drive.
operatingSystem	Boolean	Specifies whether the disk contains the HCP S Series OS. Possible values are: • **true** — The disk contains the OS. • **false** — The disk does not contain the OS.	
pCode	String	Specifies the Hitachi Vantara part number to use when ordering a replacement disk.	

(Continued)

Property name	Data type	Description	Notes
product	String	Specifies the vendor part number for the disk.	
removable	Boolean	Specifies whether the disk is a USB flash drive. Possible values are: • **true** — The disk is a USB flash drive. • **false** — The disk is not a USB flash drive.	
serial	String	Specifies the vendor serial number for the disk.	
vendor	String	Specifies the name of the disk vendor.	
warning	Boolean	For an SSD, specifies whether the disk has a warning-level condition, as indicated by the state of the mirror set that includes the disk. Possible values are: • **true** — The disk has a warning-level condition. • **false** — The disk does not have a warning-level condition. For a USB flash drive, the value of this property is always **false**. For information about mirror set states, see "Hardware: server module mirror set properties" on page 273.	
wwid	String	Specifies the disk WWID.	

Hardware: server module Ethernet interface properties

The table below describes the properties used to provide information about an Ethernet interface for a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
duplex	String	Specifies whether the network connected to the interface is operating in half-duplex or full-duplex mode. Possible values are **half** and **full**.	
error	Boolean	Specifies whether the interface has an error condition, as indicated by the value of the ok property for the interface. Possible values are: • **true** — The interface has an error condition. • **false** — The interface does not have an error condition.	
maxSpeed	String	Specifies the maximum combined transmission rate and duplex mode supported by the interface. Possible values are **10H**, **10F**, **100H**, **100F**, **1000H**, **1000F**, and **10000F**.	
mtu	Long	Specifies the actual maximum transmission unit (MTU) being used on the network connected to the interface. Possible values are **9000** and **1500**.	

(Continued)

Property name	Data type	Description	Notes
name	String	Specifies the name of the interface. Possible values are: • **eth0** — Bonded with eth2 and used for the access network • **eth1** — Used for the management network • **eth2** — Bonded with eth0 and used for the access network • **eth3** — Used for the server interconnect network	
ok	Boolean	Specifies whether the interface is physically connected to the network and is functioning normally. Possible values are: • **true** — The interface is physically connected to the network and is functioning normally. • **false** — The interface either is not physically connected to the network or is physically connected to the network but not functioning normally.	
speed	Long	Specifies the actual transmission rate for data on the network connected to the interface. Possible values are **10**, **100**, **1000**, and **10000**. These measurements are in Mbps.	

(Continued)

Property name	Data type	Description	Notes
supportedSpeedDuplex	Array	Specifies a comma-separated list of the combined transmission rates and duplex modes supported by the interface. Possible values are **10H**, **10F**, **100H**, **100F**, **1000H**, **1000F**, and **10000F**.	
type	String	Specifies the type of network that's using the interface. Possible values are: • **ACCESS** • **INTERCONNECT** • **MANAGEMENT**	
warning	Boolean	Specifies whether the actual speed and MTU on the network connected to the interface match the speed and MTU configured for the network. Possible values are: • **true** — Both the actual speed and the actual MTU match the configured speed and MTU. • **false** — The actual speed or MTU does not match the configured speed or MTU.	

Hardware: server module file system properties

The table below describes the properties used to provide information about a file system on a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
availableSpace	Long	Specifies the amount of the allocated file system space that's currently unused, in bytes.	
cutoff	Boolean	Specifies whether the file system space usage is above 95%. Possible values are: • **true** — The used space in the file system is above 95% of the total allocated space. • **false** — The used space in the file system is not above 95% of the total allocated space.	
error	Boolean	Specifies whether the file system space usage is above 90%. Possible values are: • **true** — The used space in the file system is above 90% of the total allocated space. • **false** — The used space in the file system is not above 90% of the total allocated space.	
mountPoint	String	Specifies the mount point for the file system.	
totalInodes	Long	Specifies the total number of inodes allocated to the file system.	

(Continued)

Property name	Data type	Description	Notes
totalSpace	Long	Specifies the total amount of space allocated to the file system, in bytes.	
usedInodes	Long	Specifies the number of inodes currently in use by the file system.	
usedSpace	Long	Specifies the amount of the allocated file system space that's currently in use, in bytes.	
usedSpacePercentage	Integer	Specifies the percent, from zero to 100, of the allocated file system space that's currently in use.	
warning	Boolean	Specifies whether the file system space usage is above 75%. Possible values are: • **true** — The used space in the file system is above 75% of the total allocated space. • **false** — The used space in the file system is not above 75% of the total allocated space.	

Hardware: server module IPMI properties

The table below describes the properties used to provide information about a type of IPMI sensor in a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
error	Boolean	Specifies whether any of the sensors of the type specified by the sensorType property have measured a value that is critically outside the normal operating range for the component being monitored by the sensor. Possible values are: • **true** — A sensor of this type has measured a value that is critically out of range. • **false** — No sensor of this type has measured a value that is critically out of range.	
sensors	Array	Specifies a comma-separated list of the server module IPMI sensors of the type specified by the sensorType property, where each sensor is represented by a set of properties that provide information about that sensor. For descriptions of these properties, see "Hardware: server module IPMI sensor properties" on page 272.	

(Continued)

Property name	Data type	Description	Notes
sensorType	String	Specifies the sensor type. Possible values are: • **FAN** • **POWER_SUPPLY** • **PROCESSOR** • **TEMPERATURE** • **VOLTAGE_SUPPLY**	
warning	Boolean	Specifies whether any of the sensors of the type specified by the sensorType property have measured a value that is noncritically outside the normal operating range for the component being monitored by the sensor. Possible values are: • **true** — A sensor of this type has measured a value that is noncritically out of range. • **false** — No sensor of this type has measured a value that is noncritically out of range.	

Hardware: server module IPMI sensor properties

The table below describes the properties used to provide information about an IPMI sensor in a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
detailedStatus	String	Specifies the current measurement for the component being monitored by the IPMI sensor, followed by the normal operating range for that component in parentheses.	
error	Boolean	Specifies whether the current measurement specified by the detailedStatus property is critically outside the normal operating range for the component being monitored by the sensor. Possible values are: • **true** — The measurement is critically out of range. • **false** — The measurement is not critically out of range.	
name	String	Specifies the name of the IPMI sensor.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the current measurement specified by the detailedStatus property is noncritically outside the normal operating range for the component being monitored by the sensor. Possible values are: • **true** — The measurement is noncritically out of range. • **false** — The measurement is not noncritically out of range.	

Hardware: server module mirror set properties

The table below describes the properties used to provide information about the mirror sets for a server module in /hardware resource response bodies.

A server module has mirror sets for the HCP S Series OS partitions on its SSDs and for its database drives.

Property name	Data type	Description	Notes
devNum	Integer	Specifies the mirror set device number. Valid values are integers in the range **0** through **9** for the OS partition mirror sets and **11** and **12** for the database drive mirror sets.	

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the mirror set has an error-level condition, as indicated by the status property for the set. Possible values are: • **true** — The value of the status property is **DEGRADED**. • **false** — The value of the status property is not **DEGRADED**.	
mountPoint	String	Specifies the mount point for the mirror set.	
status	String	Specifies the status of the mirror set. Possible values are: • **DEGRADED** — The mirror set includes a failed drive. • **OK** — The mirror set is synchronized and healthy. • **RECOVERING** — The mirror set is in the process of being recovered. • **RESYNCING** — The mirror set is in the process of being resynchronized.	

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the mirror set has a warning-level condition, as indicated by the status property for the set. Possible values are: • **true** — The value of the status property is **RECOVERING**. • **false** — The value of the status property is not **RECOVERING**.	

Hardware: server module mirror state property

The table below describes the property that lists the mirror sets for a server module in /hardware resource response bodies.

A server module has mirror sets for the OS partitions on its SSDs and for its database drives.

Property name	Data type	Description	Notes
sets	Array	Specifies a comma-separated list of the mirror sets for the server module, where each mirror set is represented by a set of properties that provide information about that mirror set. For descriptions of these properties, see "Hardware: server module mirror set properties" on page 273.	

Hardware: server module network interface properties

The table below describes the properties used to represent network interfaces for a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
bond	Object	Specifies a set of properties that provide information about a bonded network interface. For descriptions of these properties, see "Hardware: server module bonded network interface properties" on page 255.	
eth	Object	Specifies a set of properties that provide information about an Ethernet interface. For descriptions of these properties, see "Hardware: server module Ethernet interface properties" on page 265.	

Hardware: server module peer properties

The table below describes the properties used to provide information about the peer for a server module in /hardware resource response bodies.

The peer for a server module is the other server module.

Property name	Data type	Description	Notes
bmcOnline	Boolean	Specifies whether the BMC in the peer server module is online. Possible values are: • **true** — The BMC is online. • **false** — The BMC is offline.	
ipAddress	String	Specifies the BMC IP address of the peer server module.	

(Continued)

Property name	Data type	Description	Notes
powerOn	Boolean	Specifies whether the peer server module is powered on. Possible values are: • **true** — The peer server module is powered on. • **false** — The peer server module is powered off.	

Hardware: server module peer state property

The table below describes the property that lists the peers for a server module in /hardware resource response bodies.

In an S Series Node, a server module has only one peer.

Property name	Data type	Description	Notes
peers	Array	Specifies a comma-separated list of the server modules that are peers for this server module, where each peer is represented by a set of properties that provide information about that peer. For descriptions of these properties, see "Hardware: server module peer properties" on the previous page.	

Hardware: server module properties

The table below describes the properties used to provide detailed information about a server module in /hardware resource response bodies.

Property name	Data type	Description	Notes
bmcOnline	Boolean	Specifies whether the server module BMC is online. Possible values are: • **true** — The BMC is online. • **false** — The BMC is offline.	
coreHardware	Object	Specifies a set of properties that provide information about the core hardware in the server module. For descriptions of these properties, see "Hardware: server module core hardware properties" on page 258.	
disks	Array	Specifies a comma-separated list of the disks in the server module, where each disk is represented by a set of properties that provide information about that disk. For descriptions of these properties, see "Hardware: server module disk properties" on page 262.	A disk is either an SSD in the server module or a USB flash drive attached to the server module.

(Continued)

Property name	Data type	Description	Notes
error	Boolean	Specifies whether the S Series Node currently has any error-level alerts related to the server module. Possible values are: • **true** — The S Series Node currently has one or more error-level alerts related to the server module. • **false** — The S Series Node currently has no error-level alerts related to the server module.	
fileSystems	Array	Specifies a comma-separated list of the file systems on the server module, where each file system is represented by a set of properties that provide information about that file system. For descriptions of these properties, see "Hardware: server module file system properties" on page 268.	
id	Integer	Specifies the server module number. Possible values are **1** and **2**.	
ipmi	Array	Specifies a comma-separated list of the types of IPMI sensors in the server module, where each sensor type is represented by a set of properties that provide information about that sensor type. For descriptions of these properties, see "Hardware: server module IPMI properties" on page 270.	

(Continued)

Property name	Data type	Description	Notes
is_dc	Boolean	Specifies an internal state that currently applies to the server module. Possible values are **true** and **false**.	
mirrorState	Object	Specifies a property that lists the mirror sets for the server module, where each mirror set is represented by a set of properties that provide information about that mirror set. For descriptions of these properties, see "Hardware: server module mirror set properties" on page 273.	A server module has mirror sets for the OS partitions on its SSDs and for its database drives.
networkInterfaces	Array	Specifies a comma-separated list of the network interfaces for the server module, where each interface is represented by a property that provides information about that interface. For descriptions of these properties, see "Hardware: server module network interface properties" on page 276.	
peerState	Object	Specifies a property that lists the peers for the server module, where each peer is represented by a set of properties that provide information about that peer. For descriptions of these properties, see "Hardware: server module peer state property" on page 277.	A server module has only one peer. That peer is the other server module.

(Continued)

Property name	Data type	Description	Notes
powerOn	Boolean	Specifies whether the server module is powered on. Possible values are: • **true** — The server module is powered on. • **false** — The server module is powered off.	
status	String	Specifies whether the server module is available. Possible values are: • **AVAILABLE** — The server module is powered on, and the HCP S Series software is running on it. • **UNAVAILABLE** — Either the server module is powered off, or the server module is powered on but the HCP S Series software is not running on it.	
warning	Boolean	Specifies whether the S Series Node currently has any warning-level alerts related to the server module. Possible values are: • **true** — The S Series Node currently has one or more warning-level alerts related to the server module. • **false** — The S Series Node currently has no warning-level alerts related to the server module.	

/hardware example

Here's a sample **GET** request that retrieves information about the S Series Node hardware.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/hardware?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
```

Response body

```
{
  "enclosureInfo": [
    {
      "doors": [
        {
          "elementType": 5,
          "elementIndex": 170,
          "open": false,
          "unlocked": false,
          "disable": false,
          "fail": false,
          "swap": false,
          "ident": false,
          "code": "1",
          "predictedFailure": false
        },
          .
          .
          .
      ],
      "leds": [
        {
          "type": 28,
          "state": 0,
          "color": 4
        },
          .
          .
          .
      ],
      "sideplanes": [
        {
          "elementType": 144,
          "elementIndex": 235,
          "coverRemoved": false,
          "powered": true,
          "powerFault": false,
          "cableFault": false,
          "fault": false,
          "cableFaultRqst": false,
          "failRqst": false,
          "ident": false,
          "predictedFailure": false,
          "swap": false,
          "code": "1"
        },
          .
          .
          .
      ],
      "enclosures": [
        {
          "scpAFwRev": "N/A",
          "scpBFwRev": "N/A",
          "notice": false,
          "scpFwRev": "N/A",
          "enclConfigRev": "N/A",
          "baseboardProduct": "6004640301",
          "coverOpen": false,
          "baseboardSerial": "SGFTJ18263C8AA8",
          "warning": false,
          "error": false,
          "fail": false,
            "ident": false,
            "predictedFailure": false,
            "swap": false,
            "id": 175,
            "code": 1,
            "location": ""
         }
      ],
      "slots": [
         {
            "warning": false,
            "error": false,
            "notice": false,
            "drive": {
               "type": "DB",
               "failCode": "NONE",
               "errorsDetected": false,
               "error": false,
               "notice": false,
               "vendor": "SEAGATE",
               "product": "XS400LE10003",
               "fwRev": "0003",
               "fwRevs": [
                  "0003"
               ],
               "serial": "ZD4010HF0000822150Z3",
               "wwid": "35000c500bb32d07f",
               "capacity": 400088457216,
               "sectorSize": 512,
               "rotationRate": 1,
               "sasAddr": "0x5000c500bb32d07d",
               "ataVersion": "",
               "sataVersion": "",
               "sataSpeed": "",
               "formFactor": "2.5 inch",
               "protocol": "SAS",
               "state1": "MIRRORED",
               "state2": "MIRRORED",
               "changeTime1": "2018-11-17 01:02:27.40898",
               "changeTime2": "2018-11-19 16:55:58.374413",
               "evacuate": false,
               "reinsert": false,
               "pCode": "SGH-LFSD40-AX.X"
            },
            "maintProcedure": false,
            "status": "AVAILABLE",
            "consistencyCheck": false,
              "doNotRemove": false,
              "hotSpare": false,
              "inCriticalArray": false,
              "inFailedArray": false,
              "prepareForRemoval": false,
              "readyToInsert": false,
              "rebuildRemap": false,
              "rebuildRemapAbort": false,
              "reservedDevice": false,
              "deviceOff": false,
              "attachSasAddr": "0x500c0ff23c8a643f",
              "sasAddr": "0x500c0ff23c8a643f",
              "id": 0,
              "code": 1,
              "location": "Slot 0",
              "swap": false,
              "ident": false,
              "fail": false,
              "slotNumber": 0,
              "wwid": "35000c500bb32d07f"
            },
             .
             .
             .
          ],
          "error": false,
          "warning": false,
          "notice": false,
          "lockdownReason": "N/A",
          "id": 1,
          "product": "SP-34100-E12PM",
          "vendor": "SEAGATE",
          "fwRev": "5241",
          "fwRevs": [
            "5241"
          ],
          "wwid": "3500c0ff03c8aa83c",
          "serial": "SGFTJ18263C8AA8",
          "state1": "ADDED",
          "state2": "ADDED",
          "status": "AVAILABLE",
          "alarms": [
            {
              "error": false,
              "warning": false,
              "muted": false,
              "remind": false,
              "urgency": [],
              "id": 172,
              "code": 1,
              "location": "Ops Panel Buzzer",
              "swap": false,
              "ident": false,
              "fail": false
            }
          ],
          "powerSupplies": [
            {
              "error": false,
              "warning": false,
              "dcUnderVoltage": false,
              "vendor": "SEAGATE",
              "rev": "N/A",
              "off": false,
              "supplierProduct": "N/A",
              "dcFail": false,
              "fwRev": "0109-0120",
              "supplierSerial": "N/A",
              "serial": "M401UR0024ATP",
              "acFail": false,
              "dcOverVoltage": false,
              "product": "727621700",
              "overTempWarn": false,
              "supplierRev": "N/A",
              "overTempFail": false,
              "dcOverCurrent": false,
              "id": 100,
              "code": 1,
              "location": "Power Supply 0A",
              "swap": false,
              "ident": false,
              "fail": false,
              "pCode": "SGH-4U100-PSU-AX.X"
            },
             .
             .
             .
          ],
          "voltages": [
            {
              "error": false,
              "warning": false,
              "warnOver": false,
              "voltage": 12.38,
              "critUnder": false,
              "critOver": false,
              "warnUnder": false,
              "id": 176,
              "code": 1,
              "location": "PCM0:0",
              "swap": false,
              "ident": false,
              "fail": false,
              "critOverThresh": 0.0,
              "warnOverThresh": 0.0,
              "warnUnderThresh": 0.0,
              "critUnderThresh": 0.0
            },
             .
             .
             .
          ],
          "currents": [
            {
              "current": 22.26,
              "error": false,
              "warning": false,
              "warnOver": false,
              "critOver": false,
              "id": 184,
              "code": 1,
              "location": "PCM0:0",
              "swap": false,
              "ident": false,
              "fail": false
            },
             .
             .
             .
          ],
          "sasConnectors": [
            {
              "error": false,
              "warning": false,
              "connectorPhyLink": 255,
              "connectorType": 5,
              "id": 206,
              "code": 1,
              "location": "Server Module 1 Expander A MiniSAS 0",
              "swap": false,
              "ident": false,
              "fail": false
            },
             .
             .
             .
          ],
          "enclosureServices": [
            {
              "notice": false,
              "error": false,
              "warning": false,
              "report": true,
              "fruProduct": "730057802",
              "fruSerial": "SGFTJ18313CC332",
              "fwRev": "05020041",
              "id": 173,
              "code": 1,
              "location": "Server Module 1",
              "swap": false,
              "ident": false,
              "fail": false
            },
             .
             .
             .
          ],
          "fans": [
            {
              "actualFanSpeed": 6990,
              "error": false,
              "warning": false,
              "off": false,
              "actualSpeedCode": "LOWEST 1",
              "id": 104,
              "code": 1,
              "location": "Fan 0A",
              "swap": false,
              "ident": false,
              "fail": false
            },
             .
             .
             .
          ],
          "sasExpanders": [
            {
              "product": "6004590201",
              "error": false,
              "warning": false,
              "sasAddr": "0x500c0ff23c866b3f",
              "fwRev": "05020041",
              "arrayIndex": [
                0,
                1,
                2,
                3,
                4,
                5,
                6,
                7,
                8,
                9,
                10,
                11,
                12,
                13,
                14,
                15,
                16,
                17,
                18,
                19,
                20,
                21,
                22,
                23
              ],
              "id": 192,
              "code": 1,
              "location": "Sideplane 0 Expander A",
              "swap": false,
              "ident": false,
              "fail": false
            },
             .
             .
             .
          ],
          "temperatures": [
            {
              "error": false,
              "warning": false,
              "critOver": false,
              "critUnder": false,
              "warnUnder": false,
              "warnOver": false,
              "temperature": 35,
              "id": 116,
              "code": 1,
              "location": "Mp0:0",
              "swap": false,
              "ident": false,
              "fail": false,
              "critOverThresh": 58.0,
              "warnOverThresh": 53.0,
              "warnUnderThresh": 10.0,
              "critUnderThresh": 5.0
            },
             .
             .
             .
          ],
          "ledStates": [],
          "sbbPowerOnState": "N/A",
          "pCode": "SGH-4U100-NCL-AX.X"
        },
        {
         .
         .
         .
        }
      ],
      "serverModuleInfo": [
        {
          "sasCards": [
            {
              "cardNumber": 1,
              "sasControllers": [
                {
                  "biosVersion": "00.00.00.00",
                  "controllerNumber": 2,
                  "fwRev": "07.00.00.00",
                  "product": "SAS3616",
                  "sasAddress": "0x500c0ffd3cb63100",
                  "sasPorts": [
                    {
                      "active": true,
                      "portNumber": 1
                    },
                    {
                      "active": false,
                      "portNumber": 2
                    }
                  ],
                  "vendor": "LSI Logic / Symbios Logic"
                }
              ]
            },
             .
             .
             .
          ],
          "warning": false,
          "error": false,
          "id": 2,
          "status": "UNAVAILABLE",
          "powerOn": true,
          "bmcOnline": true,
          "peerState": {
            "peers": [
              {
                "ipAddress": "172.16.0.1",
                "bmcOnline": true,
                "powerOn": true
              }
            ]
          },
          "mirrorState": {
            "sets": [
              {
                "status": "OK",
                "warning": false,
                "error": false,
                "devNum": 12,
                "mountpoint": "/rhino/db_local"
              },
               .
               .
               .
            ]
          },
          "coreHardware": {
            "hwSetupToolVersion": "3.0.0.5",
            "ident": false,
            "failedDnsServerConnections": [],
            "warning": false,
            "lastUpdate": 1542675302286,
            "upTime": 29210000,
            "bootTime": 1542646092286,
            "minuteLoad": "46.23",
            "fiveMinuteLoad": "51.11",
            "fifteenMinuteLoad": "52.07",
            "totalMemory": 269553610752,
            "usedMemory": 268016885760,
            "totalSwap": 6438248448,
            "freeSwap": 6104014848,
            "pageSwapIn": 867,
            "pageSwapOut": 80696,
            "vendor": "Seagate",
            "product": "SGHS31CTLB",
            "rev": "TBD by OEM",
            "serial": "SGFTJ18353CC2BD",
            "ntpServer": "172.18.1.2",
            "cpus": [
               "Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz",
               "Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz"
            ],
            "loadAvgError": false,
            "swapError": false,
            "error": false,
            "pCode": "SGH-S31-CTLB-AX.X",
            "biosVendor": "INSYDE Corp.",
            "biosFwRev": "01.22",
            "biosDate": "10/29/2018"
         },
         "networkInterfaces": [
            {
               "bond": {
                  "type": "ACCESS",
                  "warning": false,
                  "name": "bond0",
                  "ok": true,
                  "mtu": 1500,
                  "error": false,
                  "activeSlave": "",
                  "mode": "802.3ad",
                  "slaves": [
                     {
                        "type": "ACCESS",
                        "maxSpeed": "10000F",
                        "supportedSpeedDuplex": [
                           "100F",
                           "1000F",
                           "10000F"
                        ],
                "warning": false,
                "name": "eth0",
                "ok": true,
                "mtu": 1500,
                "error": false,
                "speed": 10000,
                "duplex": "full"
              },
               .
               .
               .
            ]
          }
        },
        {
          "eth": {
            "type": "MANAGEMENT",
            "maxSpeed": "1000F",
            "supportedSpeedDuplex": [],
            "warning": false,
            "name": "eth4",
            "ok": true,
            "mtu": 1500,
            "error": false,
            "speed": 1000,
            "duplex": "full"
          }
        },
         .
         .
         .
      ],
      "fileSystems": [
        {
          "mountPoint": "/boot",
          "availableSpace": 941817856,
          "usedSpace": 43581440,
          "totalSpace": 1039032320,
          "usedSpacePercentage": 4,
          "totalInodes": 65536,
          "usedInodes": 336,
          "cutoff": false,
          "error": false,
          "warning": false
        },
         .
         .
```

```
         .
      ],
      "ipmi": [
        {
          "error": false,
          "warning": false,
          "sensorType": "TEMPERATURE",
          "sensors": [
            {
              "pCode": "SGH-S31-CTLB-AX.X",
              "error": false,
              "warning": false,
              "name": "TEMP Midplane 0",
              "detailedStatus": "35.0C (95.0F); (range 5.0-58.0C)"
            },
             .
             .
             .
          ]
        },
         .
         .
         .
      ],
       "disks": [
          {
            "operatingSystem": true,
            "removable": false,
            "warning": false,
            "error": false,
            "id": 1007,
            "vendor": "ATA",
            "product": "DGM28-B56D81BCBQ",
            "fwRev": "M161225",
            "serial": "BCA11807020560032",
            "wwid": "1ATA   DGM28-B56D81BCBQC-SGA          BCA11807020560032  ",
            "capacity": 256060514304,
            "pCode": "SGH-M2SD25-AX.X(Drive) and SGH-4U100-CPLT-AX.X(Server
Blank)"
        },
         .
         .
         .
      ]
    },
     .
     .
```

```
        .
      ]
    }
```

/hardware/beacon/enclosure/*enclosure-number*

With the /hardware/beacon/enclosure/*enclosure-number* resource, a **POST** request requires a query parameter. The request does not take a request body and does not return a response body.

For more information about the /hardware/beacon/enclosure/*enclosure-id* resource, see "Beaconing resources" on page 72.

/hardware/beacon/enclosure/enclosure-number query parameters

To turn enclosure beaconing on and off, you use query parameters with a **POST** request for the /hardware/beacon/enclosure/*enclosure-number* resource. The query parameters you use are:

- **on** — Turns beaconing on for the specified enclosure
- **off** — Turns beaconing off for the specified enclosure

For more information about query parameters, see "Management API query parameters" on page 61.

/hardware/beacon/enclosure/*enclosure-number* example

Here's a sample **POST** request that turns beaconing on for enclosure 1.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/beacon/enclosure/1?on"
```

Request headers

```
POST /mapi/hardware/beacon/enclosure/1?on HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/hardware/beacon/enclosure/*enclosure-number*/iom/*io-module-id*

With the /hardware/beacon/enclosure/*enclosure-number*/iom/*io-module-id* resource, a **POST** request requires a query parameter. The request does not take a request body and does not return a response body.

For more information about the /hardware/beacon/enclosure/*enclosure-number*/iom/*io-module-id* resource, see "Beaconing resources" on page 72.

/hardware/enclosure/enclosure-number/iom/io-module-id query parameters

To turn I/O module beaconing on and off, you use query parameters with a **POST** request for the /hardware/beacon/enclosure/*enclosure-number*/iom/*io-module-id* resource. The query parameters you use are:

- **on** — Turns beaconing on for the specified I/O module
- **off** — Turns beaconing off for the specified I/O module

For more information about query parameters, see "Management API query parameters" on page 61.

/hardware/beacon/enclosure/*enclosure-number*/iom/*io-module-id* example

Here's a sample **POST** request that turns beaconing on for I/O module 1 (internal ID 160) in enclosure 2.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/beacon/enclosure
  /2/iom/160?on"
```

Request headers

```
POST /mapi/hardware/beacon/enclosure/2/iom/160?on HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/hardware/beacon/server_module/*server-module-number*

With the /hardware/beacon/server_module/*server-module-number* resource, a **POST** request requires a query parameter. The request does not take a request body and does not return a response body.

For more information about the /hardware/beacon/server_module/*server-module-number* resource, see "Beaconing resources" on page 72.

/hardware/beacon/server_module/*server-module-number* query parameters

To turn server module beaconing on and off, you use query parameters with a **POST** request for the /hardware/beacon/server_module/*server-module-number* resource. The query parameters you use are:

- **on** — Turns beaconing on for the specified server module
- **off** — Turns beaconing off for the specified server module

For more information about query parameters, see "Management API query parameters" on page 61.

/hardware/beacon/server_module/*server-module-number* example

Here's a sample **POST** request that turns beaconing on for server module 1.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/beacon/server_module/1
  ?on"
```

Request headers

```
POST /mapi/hardware/beacon/server_module/1?on HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/hardware/maintenance

With the /hardware/maintenance resource, a **POST** request requires a request body and returns a response body.

For more information about the /hardware/maintenance resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance request body property

The table below describes the property in /hardware/maintenance resource request bodies.

Property name	Data type	Description	Notes
maintType	String	Specifies the type of maintenance procedure you want to perform. Valid values are: • **ADD_DRIVE** — Add one or more data or database drives to the S Series Node. • **ADD_ENCLOSURE** — Add one or more enclosures to the S Series Node. • **REMOVE_DRIVE** — Remove one or more data or database drives from the S Series Node. • **REMOVE_ENCLOSURE** — Remove one or more enclosures from the S Series Node. • **REPLACE_DRIVE** — Replace one or more data or database drives in the S Series Node. • **REPLACE_ENCLOSURE** — Replace one enclosure in the S Series Node. These values are case sensitive.	

/hardware/maintenance response body properties

A /hardware/maintenance resource response body contains properties that describe the requested maintenance procedure. These properties include the id property, which specifies an automatically generated ID for the procedure.

For descriptions of the properties in /hardware/maintenance resource response bodies, see "Maintenance procedure properties" below.

Maintenance procedure properties

The table below describes the properties used to provide information about maintenance procedures in response bodies for these resources:

- /hardware/maintenance/
- /hardware/maintenance/active
- /hardware/maintenance/history
- /hardware/maintenance/*procedure-id*
- /hardware/maintenance/*procedure-id*/cancel
- /hardware/maintenance/*procedure-id*/complete
- /hardware/maintenance/*procedure-id*/confirm
- /hardware/maintenance/*procedure-id*/perform
- /hardware/maintenance/*procedure-id*/update
- /hardware/maintenance/*procedure-id*/verify

Property name	Data type	Description	Notes
endTime	Timestamp	Specifies the date and time at which the maintenance procedure ended, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	This property is not returned for an active maintenance procedure.
id	Integer	Specifies the unique identifier for the maintenance procedure.	
maintType	String	Specifies the type of maintenance procedure. Possible values are: • **ADD_DRIVE** — Add one or more data or database drives to the S Series Node. • **ADD_ENCLOSURE** — Add one or more enclosures to the S Series Node. • **REMOVE_DRIVE** — Remove one or more data or database drives from the S Series Node. • **REMOVE_ENCLOSURE** — Remove one or more enclosures from the S Series Node. • **REPLACE_DRIVE** — Replace one or more data or database drives in the S Series Node. • **REPLACE_ENCLOSURE** — Replace one enclosure in the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
notes	String	Specifies user-supplied text that is associated with the maintenance procedure.	This property is returned only if it has a value.
selections	Object	Specifies a property that provides information about the target components for the maintenance procedure. For a description of that property, see "Maintenance procedure: target component list property" on page 313.	
startTime	Timestamp	Specifies the date and time at which the maintenance procedure was started, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	
startTsExtra	Short	Specifies an integer that, in combination with the value specified by the startTime property, makes the maintenance procedure start date and time unique.	
state	String	Specifies the current state of the maintenance procedure. Possible values are: • **ACTION** — For an add drives or replace drives procedure, one or more of the newly inserted drives were previously used in the same or a different S Series Node.	

(Continued)

Property name	Data type	Description	Notes
		• **CANCELED** — The maintenance procedure was canceled before it was completed. • **COMPLETED** — The maintenance procedure finished successfully. • **COMPLETED_ERRORS** — The maintenance procedure finished with errors. • **PERFORMING** — A perform request was issued for the maintenance procedure. While the procedure is in the **PERFORMING** state, you can perform the physical portion of the maintenance procedure on the target components. • **STARTED** — The maintenance procedure was started. • **VERIFIED** — The S Series Node has checked whether the maintenance procedure was performed correctly and that, for an add drives or replace drives procedure, none of the newly inserted drives were previously used in the same or a different S Series Node.	

(Continued)

Property name	Data type	Description	Notes
		- **VERIFYING** — The S Series Node is checking whether the maintenance procedure was performed correctly.	

Maintenance procedure: data or database drive properties

The table below describes the properties used to provide information about the data or database drive, if any, in a target slot for an add, remove, or replace drives procedure in response bodies for these resources:

- /hardware/maintenance/active
- /hardware/maintenance/history
- /hardware/maintenance/*procedure-id*
- /hardware/maintenance/*procedure-id*/cancel
- /hardware/maintenance/*procedure-id*/candidates
- /hardware/maintenance/*procedure-id*/complete
- /hardware/maintenance/*procedure-id*/confirm
- /hardware/maintenance/*procedure-id*/perform
- /hardware/maintenance/*procedure-id*/select
- /hardware/maintenance/*procedure-id*/update
- /hardware/maintenance/*procedure-id*/verify

For a slot that does not contain a drive, the value of the target component drive property is an empty set.

Property name	Data type	Description	Notes
capacity	Long	Specifies the capacity of the drive, in bytes.	

(Continued)

Property name	Data type	Description	Notes
failCode	String	For a drive that's marked failed, specifies the reason why the drive is in that condition. Possible values are: • **ADD_FAIL** — The S Series Node could not integrate the drive into the storage system. • **DRIVE_CORRUPT** — One or more I/O errors occurred on the drive, as a result of which the S Series Node logically removed the drive from the storage system. • **FORMAT_FAIL** — The S Series Node could not format the drive. • **MAINT_CANCEL** — The drive was a target component for a canceled maintenance procedure. • **MAINT_FAIL** — The drive was a target component for a failed maintenance procedure. • **MAINT_NOT_ACTIVE** — The drive was inserted into its slot while no add or replace drives procedure was active. • **MIRROR_FAULT** — An I/O error occurred on the drive while the S Series Node was protecting the internal database.	

(Continued)

Property name	Data type	Description	Notes
		• **MISSING** — The drive became unavailable while it was being initialized. • **MOVED** — The drive was moved to its current slot from another slot in the same S Series Node. • **NONE** — The drive is not marked failed. • **REMOVE_FAIL** — During a remove or replace drives procedure, the drive could not be completely removed from the internal database. • **SERIAL_MISMATCH** — The drive serial number does not match the serial number for the drive in the internal database. • **WWID_MISMATCH** — The drive WWID does not match the WWID for the drive in the internal database.	
product	String	Specifies the name of the drive hardware from the internal database.	

(Continued)

Property name	Data type	Description	Notes
reason	String	Specifies the reason why the drive is a candidate for the maintenance procedure. Possible values are: • **ENCLOSURE_ADDED** — The enclosure containing the drive was added to the S Series Node. • **ENCLOSURE_REMOVED** — The enclosure containing the drive was removed from the S Series Node. • **FAILED** — The drive is marked failed. • **MISSING** — The drive is unavailable. • **NONE** — The drive is not eligible for a maintenance procedure. • **NOT_INSTALLED** — The slot is empty. • **REMOVED** — The drive was previously removed. • **SERVER_DOWN** — A server module is unavailable. • **SPUNDOWN** — The drive is spun down. • **UNMANAGED** — The drive is unresponsive. • **VERIFY** — The drive was not selected for the maintenance procedure.	

(Continued)

Property name	Data type	Description	Notes
serial	String	Specifies the drive serial number from the internal database.	
state	String	Specifies the current state of the drive. Possible values are: • **ADD** — The drive is set to be integrated into the S Series Node. • **ADDED** — The drive is part of the S Series Node. • **ADDING** — The drive is in the process of being integrated into the S Series Node. • **DISCOVERED** — The drive was inserted into the specified slot during a maintenance procedure, and the slot was selected for the procedure. • **FAIL** — The drive is set to be marked failed. • **FAILED** — The drive is marked failed. • **FORMAT** — The drive is being formatted. • **MIRROR** — The drive is a database drive that is set to be integrated into the S Series Node. • **MIRRORED** — The drive is a database drive that is part of the S Series Node. • **MISSING** — The drive is unavailable.	

(Continued)

Property name	Data type	Description	Notes
		• **NONE** — The drive was inserted and then removed from the specified slot during a maintenance procedure. This is a transient state. • **REMOVE** — The drive is set to be removed from the S Series Node. • **REMOVED** — The drive has been logically removed from the S Series Node. • **REMOVING** — The drive is in the process of being logically removed from the S Series Node.	
vendor	String	Specifies the name of the drive vendor from the internal database.	
wwid	String	Specifies the drive WWID from the internal database.	

Maintenance procedure: enclosure or slot properties

The table below describes the properties used to provide information about the target enclosure or about the enclosure that contains the target slot for a maintenance procedure in response bodies for these resources:

- /hardware/maintenance/active
- /hardware/maintenance/history
- /hardware/maintenance/*procedure-id*
- /hardware/maintenance/*procedure-id*/cancel
- /hardware/maintenance/*procedure-id*/candidates
- /hardware/maintenance/*procedure-id*/complete

- /hardware/maintenance/*procedure-id*/confirm
- /hardware/maintenance/*procedure-id*/perform
- /hardware/maintenance/*procedure-id*/select
- /hardware/maintenance/*procedure-id*/update
- /hardware/maintenance/*procedure-id*/verify

Property name	Data type	Description	Notes
failCode	String	For an enclosure that's marked failed, specifies the reason why the enclosure is in that condition. Possible values are: • **MAINT_CANCEL** — The enclosure was a target component for a canceled maintenance procedure. • **MAINT_FAIL** — The enclosure was a target component for a failed maintenance procedure. • **MAINT_NOT_ACTIVE** — The enclosure was detected while no add or replace enclosure procedure was active. • **MISSING** — The enclosure became unavailable while it was being added to the S Series Node. • **NONE** — The enclosure is not marked failed.	

(Continued)

Property name	Data type	Description	Notes
		• **SERIAL_MISMATCH** — The enclosure serial number does not match the serial number for the enclosure in the internal database. • **WWID_MISMATCH** — The enclosure WWID does not match the WWID for the enclosure in the internal database.	
id	Integer	Specifies the number of the target enclosure or of the enclosure that contains the target slot.	
product	String	Specifies the name of the enclosure hardware from the internal database.	
reason	String	Specifies the reason why the enclosure is eligible for the maintenance procedure. Possible values are: • **ADDED** — During a replace enclosure procedure, the enclosure was integrated into the S Series Node. • **FAILED** — The enclosure is marked failed. • **MISSING** — The enclosure is unavailable. • **NONE** — The enclosure is not eligible for a maintenance procedure. • **VERIFY** — The enclosure was not selected for the maintenance procedure.	

(Continued)

Property name	Data type	Description	Notes
serial	String	Specifies the enclosure serial number from the internal database.	
slotNumber	Integer	For an add, remove, or replace drives procedure, specifies the number of the target slot. For an add, remove, or replace enclosure procedure, the value of this property is **-1**.	
state	String	Specifies the current state of the drive. Possible values are: • **ADDED** — The enclosure is part of the S Series Node. • **DISCOVERED** — The enclosure was detected during a maintenance procedure, and the enclosure was selected for the procedure. • **FAILED** — The enclosure is marked failed. • **MISSING** — The enclosure is unavailable. • **NONE** — The enclosure was connected to and then disconnected from the S Series Node during a maintenance procedure. This is a transient state.	

(Continued)

Property name	Data type	Description	Notes
		- **REMOVING** — The enclosure is in the process of being logically removed from the S Series Node.	
wwid	String	Specifies the enclosure WWID from the internal database.	

Maintenance procedure: target component list property

The table below describes the property that lists the target components for a maintenance procedure in response bodies for these resources:

- /hardware/maintenance/
- /hardware/maintenance/active
- /hardware/maintenance/history
- /hardware/maintenance/*procedure-id*
- /hardware/maintenance/*procedure-id*/cancel
- /hardware/maintenance/*procedure-id*/candidates
- /hardware/maintenance/*procedure-id*/complete
- /hardware/maintenance/*procedure-id*/confirm
- /hardware/maintenance/*procedure-id*/perform
- /hardware/maintenance/*procedure-id*/select
- /hardware/maintenance/*procedure-id*/update
- /hardware/maintenance/*procedure-id*/verify

Property name	Data type	Description	Notes
maintSelections	Array	Specifies a comma-separated list of the target components for the maintenance procedure, where each component is represented by a set of properties that provide information about that component. For descriptions of these properties, see "Maintenance procedure: target component properties" below.	If no target components have been selected for the maintenance procedure, the value of this property is an empty list.

Maintenance procedure: target component properties

The table below describes the properties used to provide information about each target component of a maintenance procedure in response bodies for these resources:

- /hardware/maintenance/active (only if target components have already been selected)
- /hardware/maintenance/history (only if target components were selected)
- /hardware/maintenance/*procedure-id* (only if target components were selected)
- /hardware/maintenance/*procedure-id*/cancel (only if target components were selected)
- /hardware/maintenance/*procedure-id*/candidates
- /hardware/maintenance/*procedure-id*/complete
- /hardware/maintenance/*procedure-id*/confirm
- /hardware/maintenance/*procedure-id*/perform
- /hardware/maintenance/*procedure-id*/select

- /hardware/maintenance/*procedure-id*/update (only if target components were selected)
- /hardware/maintenance/*procedure-id*/verify

Property name	Data type	Description	Notes
code	String	Provides additional information about the state of the target component (see the state property later in this table) after the occurrence of an abnormal event during the maintenance procedure. Possible values are: • **CANCELED** — The target component was marked failed because the maintenance procedure was canceled. • **DRIVE_ DECOMMISIONED** — For a drive that was previously used in any S Series Node, one or more I/O errors have occurred on the drive, as a result of which the S Series Node is treating the drive as unusable. • **DRIVE_FOREIGN_ RHINO** — For a drive inserted into a selected slot during an add or replace drives procedure, the drive was previously used in a different S Series Node or in the current S Series Node before the HCP S Series software was upgraded or reinstalled.	This property is not returned until the maintenance procedure has been verified.

(Continued)

Property name	Data type	Description	Notes
		- **DRIVE_INVALID_ SECTOR_SIZE** — For a drive inserted into a selected slot during an add or replace drives procedure, the drive has an invalid sector size. - **DRIVE_IOERROR** — For a drive inserted into a selected slot during an add or replace drives procedure, an I/O error occurred when the S Series Node tried to access the drive. - **DRIVE_MOVED** — For a drive inserted into a selected slot during an add or replace drives procedure, the drive was previously in a different slot in the same S Series Node. - **DRIVE_NOT_ADDED** — For a drive inserted into a selected slot during an add or replace drives procedure, the S Series Node could not integrate the drive into the storage system. - **DRIVE_NOT_ DISCOVERED** — For a drive inserted into a selected slot during an add or replace drives procedure, the S Series Node could not detect that the drive was in the slot.	

(Continued)

Property name	Data type	Description	Notes
		• **DRIVE_NOT_FOUND** — During an add, remove, or replace drives procedure, the S Series Node could not find the target drive in the internal database. • **DRIVE_NOT_REMOVED** — For a selected slot during a remove or replace drives procedure, the old drive was not removed from the slot. • **DRIVE_NOT_REMOVING** — For a selected slot during a remove or replace drives procedure, the S Series Node could not logically remove the drive from the storage system. • **DRIVE_NOT_REPLACED** — For a selected slot during a replace drives procedure, the drive in the slot was not replaced. • **DRIVE_REPLACED** — For a selected slot during a remove drives procedure, a new drive was inserted into the slot. • **DRIVE_UNEXPECTED_ DISCOVER** — During an add, remove, or replace drives procedure, a drive was inserted into the indicated slot, but the slot was not selected for the procedure.	

(Continued)

Property name	Data type	Description	Notes
		• **DRIVE_UNEXPECTED_ FAILED** — During an add, remove, or replace drives procedure, a drive inserted into the indicated slot was marked failed, but the slot was not selected for the procedure. • **DRIVE_UNEXPECTED_ MISSING** — During an add, remove, or replace drives procedure, a drive inserted into the indicated slot became unavailable, but the slot was not selected for the procedure. • **DRIVE_UNEXPECTED_ REPLACEMENT** — During an add, remove, or replace drives procedure, a drive was replaced in the indicated slot, but the slot was not selected for the procedure. • **ENCLOSURE_NOT_ ADDED** — During a replace enclosure procedure, the S Series Node could not integrate the new enclosure into the system. For help recovering from this condition, contact your authorized service provider.	

(Continued)

Property name	Data type	Description	Notes
		• **ENCLOSURE_NOT_FOUND** — During a replace enclosure procedure, the S Series Node could not find the target enclosure in the internal database. For help recovering from this condition, contact your authorized service provider. • **ENCLOSURE_NOT_REMOVED** — During a replace enclosure procedure, the old enclosure was not replaced. For help recovering from this condition, contact your authorized service provider. • **ENCLOSURE_NOT_REMOVING** — During a replace enclosure procedure, the S Series Node could not logically remove target enclosure from the system. For help recovering from this condition, contact your authorized service provider. • **INTERNAL_ERROR** — An unidentified error occurred during the maintenance procedure. For help recovering from this condition, contact your authorized service provider.	

(Continued)

Property name	Data type	Description	Notes
		- **NONE** — The specified component was selected for the maintenance procedure, and, so far, no issues have occurred with during the maintenance procedure. - **SLOT_NOT_EMPTY** — During the verify step of a remove drives procedure, the S Series Node detected the presence of a drive in the specified slot. - **SLOT_NOT_FOUND** — A slot with the specified slot number does not exist in the specified enclosure. - **UNSUPPORTED_ HARDWARE** — The drive in the specified slot is of an unsupported hardware type.	
codeSetInState	String	Specifies the value of the state property (described later in this table) at the time at which the current value of the code property was set.	This property is not returned if the value of the code property is **NONE**.
codeString	String	Specifies a text description of the state of the target component.	This property is not returned until the maintenance procedure has been verified.

(Continued)

Property name	Data type	Description	Notes
confirmAction	Boolean	For an add or replace drives procedure, specifies whether to format the drive in the target slot. Valid values are: • **true** — Format the drive • **false** — Do not format the drive. If the drive is native, reuse it. If the drive is foreign, mark it failed. These values are not case sensitive.	This property is valid in a request body only for a **POST** request for the /hardware/maintenance/*procedure-id*/confirm resource. It is not returned in any response body.
drive	Object	Specifies a set of properties that provide information about the drive, if any, in the target slot. For descriptions of these properties, see "Maintenance procedure: data or database drive properties" on page 304. For a replace enclosure procedure, the value of this property is an empty set.	For a replace drives procedure, this property initially provides information about the drive that's in the target slot when the procedure is started. After the procedure is verified, this property provides information about the newly inserted drive, and the replacedDrive property provides information about the drive that was originally in the slot.
enclosure	Object	Specifies a set of properties that provide information about the target enclosure or about the enclosure that contains the target slot. For descriptions of these properties, see "Maintenance procedure: enclosure or slot properties" on page 309.	

(Continued)

Property name	Data type	Description	Notes
replacedDrive	Object	For a replace drives procedure, specifies a set of properties that provide information about the drive that was originally in the target slot. For descriptions of these properties, see "Maintenance procedure: data or database drive properties" on page 304.	This property is returned only for a replace drives procedure and only after a **POST** request for the /hardware/maintenance/ *procedure-id*/perform resource has been issued.
replacedEnclosure	Object	For a replace enclosure procedure, specifies a set of properties that provide information about the enclosure that was replaced. For descriptions of these properties, see"Maintenance procedure: enclosure or slot properties" on page 309.	This property is returned only for a replace enclosure procedure and only after a **POST** request for the /hardware/maintenance/ *procedure-id*/perform resource has been issued.
state	String	Specifies the state of the target component at the current point in the maintenance procedure. Possible values are: • **ACTION_FOREIGN** — For an add or replace drives procedure, the drive inserted into the specified slot was previously used in another S Series Node or in the current S Series Node before the HCP S Series software was upgraded or reinstalled (that is, the drive is a foreign drive). You can choose to format the drive or treat it as a failed drive.	

(Continued)

Property name	Data type	Description	Notes
		• **ACTION_NATIVE** — For an add or replace drives procedure, the drive inserted into the specified slot was previously used in the current S Series Node, and the HCP S Series software has not been upgraded or reinstalled since the drive was removed (that is, the drive is a native drive). You can choose to format the drive or use it as is. The latter action is useful if the drive was previously unintentionally removed from the enclosure. In this case, reusing the drive may facilitate repairs. • **ADD** — For an add drives procedure, the specified slot was selected for the procedure, and the procedure has not yet ended. For a replace drives procedure, the specified slot was selected for the procedure, the old drive has been removed from the slot, and the procedure has not yet ended. • **ADDED** — For an add or replace drives procedure, a new drive was successfully inserted into the specified slot and is ready to be used by the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
		• **ADDING** — The S Series Node is in the process of completing an add or replace drives procedure in which a new drive was inserted into the specified slot. • **ERROR** — The target component was not successfully added, removed, or replaced, as applicable, during the maintenance procedure. For more information, see the code property (described earlier in this table). • **FAILED** — During an add, remove, or replace drives procedure, the target drive was marked failed. During a replace enclosure operation, the target enclosure was marked failed. For help recovering from the latter condition, contact your authorized service provider. • **FAILING** — During an add, remove, or replace drives procedure, the target drive is in the process of being marked failed. During a replace enclosure operation, the target enclosure is in the process of being marked failed. In either case, this is a transient state.	

(Continued)

Property name	Data type	Description	Notes
		• **NONE** — The specified component was selected for the maintenance procedure, but a perform request has not yet been issued for the procedure. • **REMOVE** — For a remove or replace drives procedure, the specified slot was selected for the procedure, but a perform request has not yet been issued for the procedure. • **REMOVED** — For a remove drives procedure, the drive was removed from the specified slot. • **REMOVING** — For a remove drives procedure, the specified slot was selected for the procedure, and a perform request has been issued for the procedure. For a replace drives procedure, the specified slot was selected for the procedure, a perform request has been issued for the procedure, but the old drive has not yet been removed from the slot.	

(Continued)

Property name	Data type	Description	Notes
		- **WARNING** — An abnormal condition was detected for the specified component, but the component was not selected for the maintenance procedure. If the procedure is active, you can safely continue it.	

/hardware/maintenance example

Here's a sample **POST** request that starts an add drives maintenance procedure.

Request body

```
{
  "maintType": "ADD_DRIVE"
}
```

Request with curl command line

```
curl -k -X POST -d @add_drives_start.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 177
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2018-11-27 09:21:52 UTC",
  "startTsExtra": 77,
  "selections": {
    "maintSelections": []
  }
}
```

/hardware/maintenance/active

With the /hardware/maintenance/active resource, a **GET** request returns a response body.

For more information about the /hardware/maintenance/active resource, see "Maintenance resources" on page 79.

/hardware/maintenance/active property

The table below describes the top-level property in /hardware/maintenance/active resource response bodies.

Property name	Data type	Description	Notes
maintProcedures	Array	Specifies a comma-separated list of the currently active hardware maintenance procedures, where each procedure is represented by a set of properties that provide information about that procedure. For descriptions of these properties,"Maintenance procedure properties" on page 300.	If no maintenance procedures are active, the value of this property is an empty list.

/hardware/maintenance/active example

Here's a sample **GET** request that retrieves a list of the currently active hardware maintenance procedures.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/active
  ?prettyprint"
```

Request headers

```
GET /mapi/hardware/maintenance/active?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 618
```

Response body

```
{
  "isTruncated": false,
  "maintProcedures": [
    "id": 10,
    "maintType": "ADD_DRIVE",
    "state": "STARTED",
    "startTime": "2018-11-27 09:21:52 UTC",
    "startTsExtra": 77,
    "selections": {
      "maintSelections": [
        {
          "state": "ADD",
          "code": "NONE",
          "codeString": "None",
          "enclosure": {
            "wwid": "3500c0ff03c8aa83c",
            "id": 1,
            "product": "SP-34100-E12PM",
            "serial": "SGFTJ18263C8AA8",
            "slotNumber": 7
          },
          "drive": {}
        }
      ]
    }
  ]
}
```

/hardware/maintenance/history

With the /hardware/maintenance/history resource, a **GET** request returns a response body.

For information about the query parameters used to limit the maintence procedure history list returned by a **GET** request, see "Managing resource lists" on page 91.

For more information about the /hardware/maintenance/history resource, see "Maintenance resources" on page 79.

/hardware/maintenance/history properties

The table below describes the top-level properties in /hardware/maintenance/history resource response bodies.

Property name	Data type	Description	Notes
count	Integer	Specifies the value of the **count** query parameter included in the **GET** request or 1,000 if the request did not include the **count** parameter. For more information, see "count query parameter" on page 92.	
isTruncated	Boolean	Specifies whether the returned list of maintenance procedures is complete. Possible values are: • **true** — The history list is incomplete. • **false** — The history list is complete. For more information, see "count query parameter" on page 92.	

(Continued)

Property name	Data type	Description	Notes
maintProcedures	Array	Specifies a comma-separated list of all completed or canceled maintenance procedures that have been performed on the S Series Node since the HCP S Series software was last installed, where each procedure is represented by a set of properties that provide information about that procedure. For descriptions of these properties,"Maintenance procedure properties" on page 300.	If no maintenance procedures have been performed, the value of this property is an empty list.
marker	String	Specifies the value of the **marker** query parameter included in the **GET** request or **2147483647** value if the request did not include the **marker** parameter. For more information, see "marker query parameter" on page 93.	

/hardware/maintenance/history example

Here's a sample **GET** request that retrieves a list of the past hardware maintenance procedures.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/history
  ?marker=11&count=2&prettyprint"
```

Request headers

```
GET /mapi/hardware/maintenance/history?marker=11&count=2&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 2083
```

Response body

```
{
  "marker": "11"
  "count": "2"
  "isTruncated": true
  "maintProcedures": [
    {
      "id": 10,
      "maintType": "ADD_DRIVE",
      "state": "COMPLETED",
      "startTime": "2018-11-27 09:21:52 UTC",
      "startTsExtra": 77,
      "endTime": "2018-11-27 09:46:22 UTC",
      "notes": "Reinserting incorrectly removed drive.",
      "selections": {
       "maintSelections": [
          {
            "state": "ADDED",
            "code": "NONE",
            "codeString": "None",
            "enclosure": {
              "wwid": "3500c0ff03c8aa83c",
              "id": 1,
              "product": "SP-34100-E12PM",
              "serial": "SGFTJ18263C8AA8",
              "slotNumber": 7
            },
            "drive": {
              "wwid": "35000c500952bb9cf",
              "vendor": "SEAGATE",
              "product": "ST10000NM0096",
              "serial": "ZA2554A50000C8292SHR",
              "capacity": 10000831348736,
              "state": "ADDED",
              "failCode": "NONE"
            }
          }
        ]
      }
    },
    {
      "id": 9,
      "maintType": "REMOVE_DRIVE",
      "state": "COMPLETED",
      "startTime": "2018-11-27 09:47:41 UTC",
      "startTsExtra": 95,
      "endTime": "2018-11-27 09:50:23 UTC",
      "selections": {
        "maintSelections": [
          {
            "state": "REMOVED",
            "code": "NONE",
            "codeString": "None",
            "enclosure": {
              "wwid": "3500c0ff03c8aa83c",
              "id": 1,
              "product": "SP-34100-E12PM",
              "serial": "SGFTJ18263C8AA8",
              "slotNumber": 7
            },
            "drive": {
              "wwid": "35000c500952bb9cf",
              "vendor": "SEAGATE",
              "product": "ST10000NM0096",
              "serial": "ZA2554A50000C8292SHR",
              "capacity": 10000831348736,
              "state": "REMOVED",
              "failCode": "MISSING"
            }
          }
        ]
      }
    }
  ]
}
```

/hardware/maintenance/*procedure-id*

With the /hardware/maintenance/*procedure-id* resource, a **GET** request returns a response body.

For more information about the /hardware/maintenance/*procedure-id* resource, see "Maintenance resources" on page 79.

/hardware/maintenance/*procedure-id* properties

A /hardware/maintenance/*procedure-id* resource response body contains properties that describe the maintenance procedure specified in the URL. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id* example

Here's a sample **GET** request that retrieves information about the maintenance procedure with ID 9.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/9
  ?prettyprint"
```

Request headers

```
GET /mapi/hardware/maintenance/9?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 766
```

Response body

```
{
  "id": 9,
  "maintType": "REMOVE_DRIVE",
  "state": "PERFORMING",
    "startTime": "2018-11-27 09:47:41 UTC",
    "startTsExtra": 95,
    "selections": {
      "maintSelections": [
        {
          "state": "REMOVING",
          "code": "NONE",
          "codeString": "None",
          "enclosure": {
            "wwid": "3500093d00179b000",
            "id": 1,
            "product": "NDS-4600-JD",
            "serial": "MXE340003ATRB0BB",
            "slotNumber": 7
          },
          "drive": {
            "reason": "MISSING"
            "wwid": "35000c5006990b1d1",
            "vendor": "ATA",
            "product": "ST4000DM000-1F21",
            "serial": "W300DFDK",
            "capacity": 4000787030016,
            "state": "FAILED",
            "failCode": "MISSING"
          }
        }
      ]
    }
  }
```

/hardware/maintenance/*procedure-id*/cancel

With the /hardware/maintenance/*procedure-id*/cancel resource, a **POST** request returns a response body. The request does not take a request body.

For more information about the /hardware/maintenance/*procedure-id*/cancel resource, see "Maintenance resources" on page 79.

/hardware/maintenance/*procedure-id*/cancel properties

A /hardware/maintenance/*procedure-id*/cancel resource response body contains properties that describe the maintenance procedure being canceled. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id*/cancel example

Here's a sample **POST** request that cancels the hardware maintenance procedure with ID 11.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/11/cancel
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/11/cancel?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 897
```

Response body

```
{
  "id": 11,
  "maintType": "REMOVE_DRIVE",
  "state": "CANCELED",
  "startTime": "2018-11-27 15:49:25 UTC",
  "startTsExtra": 128,
  "endTime": "2018-11-27 15:51:21 UTC",
  "selections": {
    "maintSelections": [
      {
        "state": "FAILED",
        "code": "CANCELED",
        "codeString": "Marked failed due to maintenance procedure cancelation",
        "codeSetInState": "PERFORMING",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {
          "reason": "MISSING"
          "wwid": "35000c500952bb9cf",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
          "serial": "ZA2554A50000C8292SHR",
          "capacity": 10000831348736,
          "state": "FAILED",
          "failCode": "MISSING"
        }
      }
    ]
  }
}
```

/hardware/maintenance/*procedure-id*/candidates

With the /hardware/maintenance/*procedure-id*/candidates resource, a **GET** request returns a response body.

For more information about the /hardware/maintenance/*procedure-id*/candidates resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/candidates resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/candidates property

The table below describes the top-level property in /hardware/maintenance/*procedure-id*/candidates resource response bodies.

Property name	Data type	Description	Notes
maintSelections	Array	Specifies a comma-separated list of the hardware components that are eligible to be targets of the maintenance procedure, where each component is represented by a set of properties that provide information about that component. For descriptions of these properties, see "Maintenance procedure: target component properties" on page 314.	

/hardware/maintenance/*procedure-id*/candidates example

Here's a sample **GET** request that retrieves a list of the components that are eligible to be targets of the hardware maintenance procedure with ID 10.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10
  /candidates?prettyprint"
```

Request headers

```
GET /mapi/hardware/maintenance/10/candidates?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 313
```

Response body

```
{
  "maintSelections": [
    {
      "state": "NONE",
      "code": "NONE",
      "codeString": "None",
      "enclosure": {
        "wwid": "3500c0ff03c8aa83c",
        "id": 1,
        "product": "SP-34100-E12PM",
        "serial": "SGFTJ18263C8AA8",
        "slotNumber": 7
      },
      "drive": {}
    }
  ]
}
```

/hardware/maintenance/*procedure-id*/complete

With the /hardware/maintenance/*procedure-id*/complete resource, a **POST** request returns a response body. The request does not take a request body.

For more information about the /hardware/maintenance/*procedure-id*/complete resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/complete resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/complete properties

A /hardware/maintenance/*procedure-id*/complete resource response body contains properties that describe the maintenance procedure being completed. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id*/complete example

Here's a sample **POST** request that completes the hardware maintenance procedure with ID 10.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10/complete
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/10/complete?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 848
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "COMPLETED",
  "startTime": "2018-11-27 09:21:52 UTC",
  "startTsExtra": 77,
  "endTime": "2018-11-27 09:46:22 UTC",
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADDED",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {
          "reason": "NONE"
          "wwid": "35000c500952bb9cf",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
          "serial": "ZA2554A50000C8292SHR",
          "capacity": 10000831348736,
          "state": "ADDED",
          "failCode": "NONE"
        }
      }
    ]
  }
}
```

/hardware/maintenance/*procedure-id*/confirm

With the /hardware/maintenance/*procedure-id*/confirm resource, a **POST** request requires a request body and returns a response body.

For more information about the /hardware/maintenance/*procedure-id*/confirm resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/confirm resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/confirm request body properties

The table below describes the top-level property in /hardware/maintenance/*procedure-id*/confirm resource request bodies.

Property name	Data type	Description	Notes
maintSelections	Array	Specifies a comma-separated list of the native and foreign drives that were inserted into the enclosure during an add or replace drives procedure, where each drive is represented by a set of properties that provide information about that drive and specify what you want to do with the drive. These properties are described in the next table.	A native drive is one that was previously used in the current S Series Node, where the HCP S Series software has not been upgraded or reinstalled since the drive was removed. A foreign drive is one that was previously used in a different S Series Node or in the current S Series Node before the HCP S Series software was upgraded or reinstalled.

The table below describes the properties used to provide information about each native or foreign drive and specify what you want to do with that drive.

Property name	Data type	Description	Notes
confirmAction	Boolean	Specifies how you want to handle the native or foreign drive. Valid values are: • **true** — Format the drive and then use it. • **false** — For a native drive, use the drive as is. For a foreign drive, treat the drive as a failed drive. These values are not case sensitive.	

(Continued)

Property name	Data type	Description	Notes
drive	Object	Specifies a property that provides information about the native or foreign drive. For a description of this property, see "Native or foreign drive property" below.	The drive property is required in the request body for a replace drives procedure. It is invalid in the request body for an add drives procedure.
enclosure	Object	Specifies a set of properties that provide information about the slot that contains the native or foreign drive. For descriptions of these properties, see "Slot properties" below.	

Native or foreign drive property

The table below describes the property used to provide information about a native or foreign drive.

Property name	Data type	Description	Notes
wwid	String	Specifies the WWID of the drive.	

Slot properties

The table below describes the properties used to provide information about each slot that contains a native or foreign drive.

Property name	Data type	Description	Notes
id	Integer	Specifies the number of the enclosure that contains the target slot.	
slotNumber	Integer	Specifies the number of the target slot.	
wwid	String	Specifies the WWID of the enclosure that contains the target slot.	

/hardware/maintenance/*procedure-id*/confirm response body properties

A /hardware/maintenance/*procedure-id*/confirm resource response body contains properties that describe the maintenance procedure for which your specifying how to handle native and foreign drives. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id*/confirm example

Here's a sample **POST** request that tells the S Series Node not to format the drive in slot 7 in enclosure 1.

Request body

```
{
  "maintSelections": [
    {
      "enclosure": {
        "id": 1,
        "slotNumber": 7,
        "wwid": "35000c500952bb9cf"
      },
      "confirmAction": "true"
    }
  ]
}
```

Request with curl command line

```
curl -k -X POST -d @no_format_drive.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10/confirm
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/10/confirm?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 810
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "VERIFIED",
  "startTime": "2018-11-27 09:21:52 UTC",
  "startTsExtra": 77,
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {
          "reason": "MISSING"
          "wwid": "35000c500952bb9cf",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
          "serial": "ZA2554A50000C8292SHR",
          "capacity": 10000831348736,
          "state": "REMOVED",
          "failCode": "MISSING"
        }
      }
    ]
  }
}
```

/hardware/maintenance/*procedure-id*/perform

With the /hardware/maintenance/*procedure-id*/perform resource, a **POST** request returns a response body. The request does not take a request body.

For more information about the /hardware/maintenance/*procedure-id*/perform resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/perform resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/perform properties

A /hardware/maintenance/*procedure-id*/perform resource response body contains properties that describe the maintenance procedure being performed. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id*/perform example

Here's a sample **POST** request that prepares the S Series Node for the physical portion of the maintenance procedure with ID 10.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10/perform
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/10/perform?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 494
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "PERFORMING",
  "startTime": "2018-11-09 10:54:29 UTC",
  "startTsExtra": 77,
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {}
      }
    ]
  }
}
```

/hardware/maintenance/*procedure-id*/select

With the /hardware/maintenance/*procedure-id*/select resource, a **POST** request requires a request body and returns a response body.

For more information about the /hardware/maintenance/*procedure-id*/select resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/select resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/select request body properties

The table below describes the top-level property in /hardware/maintenance/*procedure-id*/select resource request bodies.

Property name	Data type	Description	Notes
maintSelections	Array	Specifies a comma-separated list of the target components for the maintenance procedure, where each component is represented by a set of properties that provide information about that component. For descriptions of these properties, see "Target component properties" below.	

Target component properties

The table below describes the properties used to provide information about each target component for the maintenance procedure.

Property name	Data type	Description	Notes
drive	Object	For a remove or replace drives procedure, specifies a property that provides information about the target drive. For a description of this property, see "Drive property" on the next page.	Do not include this property in the request body for an add drives or replace enclosure procedure.

(Continued)

Property name	Data type	Description	Notes
enclosure	Object	Specifies a set of properties that provide information about the target enclosure or about the enclosure that contains the target slot. For descriptions of these properties, see "Enclosure or slot properties" below.	

Enclosure or slot properties

The table below describes the properties used to provide information about a target enclosure or about the enclosure that contains a target slot.

Property name	Data type	Description	Notes
id	Integer	Specifies the number of the target enclosure or of the enclosure that contains the slot.	
slotNumber	Integer	For an add, remove, or replace drives procedure, specifies the number of the target slot.	Do not include this property in the request body for a replace enclosure procedure.
wwid	String	Specifies the WWID of the target enclosure or of the enclosure that contains the slot.	

Drive property

The table below describes the property used to provide information about a target drive.

Property name	Data type	Description	Notes
wwid	String	Specifies the WWID of the target drive.	

/hardware/maintenance/*procedure-id*/select response body properties

A /hardware/maintenance/*procedure-id*/select resource response body contains properties that describe the maintenance procedure for which you're selecting target components. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id*/select example

Here's a sample **POST** request that selects slot 7 in enclosure 1 to be the target of the maintenance procedure with ID 10.

Request body

```
{
  "maintSelections": [
    {
      "enclosure": {
        "id": 1,
        "slotNumber": 7,
        "wwid": "35000c500952bb9cf"
      }
    }
  ]
}
```

Request with curl command line

```
curl -k -X POST -d @slot_selection.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10/select
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/10/select?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 490
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2018-11-27 09:21:52 UTC",
  "startTsExtra": 77,
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {}
      }
    ]
  }
}
```

/hardware/maintenance/*procedure-id*/update

With the /hardware/maintenance/*procedure-id*/update resource, a **POST** request requires a request body and returns a response body.

For more information about the /hardware/maintenance/*procedure-id*/update resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/update resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/update request body property

The table below describes the property in /hardware/maintenance/*procedure-id*/update resource request bodies.

Property name	Data type	Description	Notes
notes	String	Associates text with the maintenance procedure. This text can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space.	The text specified for this property replaces any text already associated with the maintenance procedure.

/hardware/maintenance/*procedure-id*/update response body properties

A /hardware/maintenance/*procedure-id*/update resource response body contains properties that describe the maintenance procedure you're updating. For descriptions of these properties, see "Maintenance procedure properties" on page 300.

/hardware/maintenance/*procedure-id*/update example

Here's a sample **POST** request that adds a note to the maintenance procedure with ID 10.

Request body

```
{
  "notes": "Reinserting incorrectly removed drive."
}
```

Request with curl command line

```
curl -k -X POST -d @proc_notes.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10/update
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/10/update?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 543
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2018-11-27 09:21:52 UTC",
  "startTsExtra": 77,
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {}
      }
    ]
  }
}
```

/hardware/maintenance/*procedure-id*/verify

With the /hardware/maintenance/*procedure-id*/verify resource, a **POST** request returns a response body. The request does not take a request body.

For more information about the /hardware/maintenance/*procedure-id*/verify resource, see "Maintenance resources" on page 79. For an example of using the /hardware/maintenance/*procedure-id*/verify resource in a maintenance procedure, see "Replacing a data or database drive".

/hardware/maintenance/*procedure-id*/verify properties

A /hardware/maintenance/*procedure-id*/verify resource response body contains properties that describe the maintenance procedure being verified. For descriptions of these properties,.

/hardware/maintenance/*procedure-id*/verify example

Here's a sample **POST** request that verifies the hardware maintenance procedure with ID 10.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/10/verify
  ?prettyprint"
```

Request headers

```
POST /mapi/hardware/maintenance/10/verify?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 817
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "ACTION",
  "startTime": "2018-11-27 09:21:52 UTC",
  "startTsExtra": 77,
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ACTION_NATIVE",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 7
        },
        "drive": {
          "reason": "MISSING"
          "wwid": "35000c500952bb9cf",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
          "serial": "ZA2554A50000C8292SHR",
          "capacity": 10000831348736,
          "state": "REMOVED",
          "failCode": "MISSING"
        }
      }
    ]
  }
}
```

/hardware/power/node

With the /hardware/power/node resource, a **POST** request requires query parameters. The request does not take a request body and does not return a response body.

For more information about the /hardware/power/node resource, see "Power resources" on page 85.

/hardware/power/node query parameters

To turn power off or restart both server modules in an S Series Node, you use query parameters with a **POST** request for the /hardware/power/node resource. The query parameters you use are:

- **reason** — Specifies the reason why you're shutting down the server modules. The value of this parameter is a text string that must be 1 through 1,024 characters long and can contain any valid UTF-8 characters, including percent-encoded white space.
- **shutdown** — Powers off both server modules.
- **reboot** — Restarts both server modules.

The **POST** request must include either the **shutdown** parameter or the **reboot** parameter, but not both. In either case, the request must also include the **reason** parameter.

For more information about query parameters, see "Management API query parameters" on page 61.

/hardware/power/node example

Here's a sample **POST** request that restarts both server modules in an S Series Node.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/power/node?reboot
  &reason=Testing%20node%20restart"
```

Request headers

```
POST /mapi/configuration/mapi/hardware/power/node?reboot&reason=Testing%20
node%20restart HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/hardware/power/*server-module-number*

With the /hardware/power/*server-module-number* resource, a **POST** request requires query parameters. The request does not take a request body and does not return a response body.

For more information about the /hardware/power/*server-module-number* resource, see "Power resources" on page 85.

/hardware/power/*server-module-number* query parameters

To turn power on or off or restart an individual server module in an S Series Node, you use query parameters with a **POST** request for the /hardware/power/*server-module-number* resource. The query parameters you use are:

- **reason** — Specifies the reason why you're shutting down the server module. The value of this parameter is a text string that must be 1 through 1,024 characters long and can contain any valid UTF-8 characters, including percent-encoded white space.

- **on** — Powers on the specified server module. You can use a management API request to power on a single server module only if the other server module is available.

- **shutdown** — Powers off the specified server module.

- **reboot** — Restarts the specified server module.

The **POST** request must include exactly one of the **on**, **shutdown**, and **reboot** parameters. In any case, the request must also include the **reason** parameter.

For more information about query parameters, see "Management API query parameters" on page 61.

/hardware/power/*server-module-number* example

Here's a sample **POST** request that powers off server module 1.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/power/1?shutdown
  &reason=Maintenance%20required"
```

Request headers

```
POST /mapi/configuration/mapi/hardware/power/1?shutdown&reason=Maintenance
%20required HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/metrics/buckets

With the /metrics/buckets resource, a **GET** request returns a response body.

For information about the query parameters used to limit the list of buckets for which statistics are returned by a **GET** request, see "Managing resource lists" on page 91.

For more information about the /metrics/buckets resource, see "Metrics resources" on page 82.

/metrics/buckets properties

The table below describes the properties in /metrics/buckets resource response bodies.

Property name	Data type	Description	Notes
bucketCount	Integer	Specifies the value of the **count** query parameter included in the **GET** request or 1,000 if the request did not include the **count** parameter. For more information, see "count query parameter" on page 92.	
buckets	Array	Specifies a comma-separated list of the buckets that satisfy the request criteria. Each bucket is represented by the properties described in the next table.	
isTruncated	Boolean	Specifies whether the returned list of buckets is complete. Possible values are: • **true** — The bucket list is incomplete. • **false** — The bucket list is complete. For more information, see "count query parameter" on page 92.	

(Continued)

Property name	Data type	Description	Notes
marker	String	Specifies the value of the **marker** query parameter included in the **GET** request or no value if the request did not include the **marker** parameter. For more information, see "marker query parameter" on page 93.	

The table below describes the properties used to represent buckets in the array of buckets returned in response to a **GET** request for the /metrics/buckets resource.

Property name	Data type	Description	Notes
logicalUsedBytes	Long	Specifies the total number of bytes of data written to the S Series Node for all objects currently in the bucket.	
name	String	Specifies the bucket name.	
objectcount	Long	Specifies the total number of objects currently stored in the bucket.	
physicalUsedBytes	Long	Specifies the total number of bytes currently used for storing and protecting data in the bucket.	

/metrics/buckets example

Here's a sample **GET** request that retrieves statistics about bucket usage.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/metrics/buckets?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/metrics/buckets?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 241
```

Response body

```
{
  "buckets": [
    {
      "name": "hcpsrv-hcp-ma",
      "objectcount": 1656631,
      "logicalUsedBytes": 82831550000000,
      "physicalUsedBytes": 107698381462773
    }
  ],
  "marker": "",
  "bucketCount": 1000,
  "isTruncated": false
}
```

/metrics/gateways

With the /metrics/gateways resource, a **GET** request returns a response body.

For more information about the /metrics/gateways resource, see "Metrics resources" on page 82.

/metrics/gateways properties

The table below describes the top-level property in /metrics/gateways resource response bodies.

Property name	Data type	Description	Notes
hs3	Object	Specifies a set of properties that provide information about the use of the Hitachi API for Amazon S3 (the S3 compatible API). These properties are described in the next table.	

The table below describes the properties used to provide information about the use of the S3 compatible API in /metrics/gateways resource response bodies.

Property name	Data type	Description	Notes
bytesCommitted	Long	Specifies the total number of bytes in complete objects created on the S Series Node from parts written during a multipart write since the date and time specified by the startTime property.	
bytesDeleted	Long	Specifies the total number of bytes deleted from the S Series Node since the date and time specified by the startTime property.	
bytesLinked	Long	Specifies the total number of bytes linked on the S Series Node during copy operations since the date and time specified by the startTime property.	Instead of data being duplicated by a copy operation, the object created by the operation can point to the original object data. Linked bytes are the bytes occupied by that data.

(Continued)

Property name	Data type	Description	Notes
bytesRead	Long	Specifies the total number of bytes read from the S Series Node since the date and time specified by the startTime property.	
bytesWritten	Long	Specifies the total number of bytes written to the S Series Node since the date and time specified by the startTime property.	
objectsCommitted	Long	Specifies the total number of complete objects created on the S Series Node from parts written during a multipart write since the date and time specified by the startTime property.	
objectsDeleted	Long	Specifies the total number of objects deleted from the S Series Node since the date and time specified by the startTime property.	
objectsLinked	Long	Specifies the total number of objects linked on the S Series Node during copy operations since the date and time specified by the startTime property.	Instead of data being duplicated by a copy operation, the object created by the operation can point to the original object data. Linked objects are the objects created by these copy operations.
objectsRead	Long	Specifies the total number of objects read from the S Series Node since the date and time specified by the startTime property.	
objectsWritten	Long	Specifies the total number of objects written to the S Series Node since the date and time specified by the startTime property.	

(Continued)

Property name	Data type	Description	Notes
serverModules	Array	Specifies a comma-separated list of server modules. Each server module is represented by the properties described in the next table.	
startTime	Timestamp	Specifies the start time of the interval during which the S Series Node statistics were collected, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC The end time is the time at which the **GET** request was processed.	The start time is the earlier of the times specified by the startTime property for each server module, as described in the next table.

The table below describes the properties used to represent server modules in /metrics/gateways resource response bodies.

Property name	Data type	Description	Notes
bytesCommitted	Long	Specifies the total number of bytes in complete objects created by the server module from parts written during a multipart write since the date and time specified by the startTime property for the server module.	
bytesDeleted	Long	Specifies the total number of bytes deleted from the S Series Node through the server module since the date and time specified by the startTime property for the server module.	

(Continued)

Property name	Data type	Description	Notes
bytesLinked	Long	Specifies the total number of bytes linked by the server module during copy operations since the date and time specified by the startTime property for the server module.	Instead of data being duplicated by a copy operation, the object created by the operation can point to the original object data. Linked bytes are the bytes occupied by that data.
bytesRead	Long	Specifies the total number of bytes read from the S Series Node through the server module since the date and time specified by the startTime property for the server module.	
bytesWritten	Long	Specifies the total number of bytes written to the S Series Node through the server module since the date and time specified by the startTime property for the server module.	
objectsCommitted	Long	Specifies the total number of complete objects created by the server module from parts written during a multipart write since the date and time specified by the startTime property for the server module.	
objectsDeleted	Long	Specifies the total number of objects deleted from the S Series Node through the server module since the date and time specified by the startTime property for the server module.	

(Continued)

Property name	Data type	Description	Notes
objectsLinked	Long	Specifies the total number of objects linked by the server module during copy operations since the date and time specified by the startTime property for the server module.	Instead of data being duplicated by a copy operation, the object created by the operation can point to the original object data. Linked objects are the objects created by these copy operations.
objectsRead	Long	Specifies the total number of objects read from the S Series Node through the server module since the date and time specified by the startTime property for the server module.	
objectsWritten	Long	Specifies the total number of objects written to the S Series Node through the server module since the date and time specified by the startTime property for the server module.	
serverModuleNumber	Integer	Specifies the server module number.	
startTime	Timestamp	Specifies the start time of the interval during which the server module statistics were collected, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC The end time is the time at which the **GET** request was processed.	The start time is reset each time the HCP S Series software restarts on the server module.

/metrics/gateways example

Here's a sample **GET** request that retrieves statistics about use of the S3 compatible API.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/metrics/gateways?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/metrics/gateways?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 1123
```

Response body

```
{
  "hs3": {
    "serverModules": [
      {
        "serverModuleNumber": 2,
        "bytesRead": 0,
        "bytesWritten": 25769803776026,
        "bytesLinked": 0,
        "bytesDeleted": 0,
        "bytesCommitted": 0,
        "objectsRead": 0,
        "objectsWritten": 25323,
        "objectsLinked": 0,
        "objectsDeleted": 0,
        "objectsCommitted": 0,
        "startTime": "2018-09-04 14:50:37 EST"
      },
      {
        "serverModuleNumber": 1,
        "bytesRead": 0,
        "bytesWritten": 25769803805832,
        "bytesLinked": 0,
        "bytesDeleted": 0,
        "bytesCommitted": 0,
        "objectsRead": 0,
        "objectsWritten": 22694,
        "objectsLinked": 0,
        "objectsDeleted": 0,
        "objectsCommitted": 0,
        "startTime": "2018-09-04 14:50:35 EST"
      }
    ],
    "bytesRead": 0,
    "bytesWritten": 51539607581858,
    "bytesLinked": 0,
    "bytesDeleted": 0,
    "bytesCommitted": 0,
    "objectsRead": 0,
    "objectsWritten": 48017,
    "objectsLinked": 0,
    "objectsDeleted": 0,
    "objectsCommitted": 0,
    "startTime": "2018-09-04 14:50:35 EST"
  }
}
```

/metrics/protection

With the /metrics/protection resource, a **GET** request returns a response body.

For more information about the /metrics/protection resource, see "Metrics resources" on page 82.

/metrics/protection property

The table below describes the property in /metrics/protection resource response bodies.

Property name	Data type	Description	Notes
underRepairBytes	Long	Specifies the total number of bytes that need to be written for all stored data to be fully protected.	

/metrics/protection example

Here's a sample **GET** request that retrieves statistics about data being repaired by the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/metrics/protection?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/metrics/protection?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 27
```

Response body

```
{
  "underRepairBytes": 0
}
```

/metrics/system

With the /metrics/system resource, a **GET** request returns a response body.

For more information about the /metrics/system resource, see "Metrics resources" on page 82.

/metrics/system properties

The table below describes the properties in /metrics/system resource response bodies.

Property name	Data type	Description	Notes
bucketCount	Integer	Specifies the total number of buckets that currently exist on the S Series Node.	
efficiency	Double	Specifies a percent representing the ratio between the amount of data ingested for the objects currently stored on the S Series Node and the current amount of used storage on the S Series Node.	Normally, due to metadata and data protection, the amount of storage required to store an object is greater than the size of the data ingested for the object. However, S Series Nodes can single-instance object data. Single-instancing means storing and protecting only one copy of the data for multiple objects that have the same ingested data. With single-instancing, the byte count for ingested data increases as new copies of existing data are written to the S Series Node, but the amount of storage used for that data remains the same. As a result, single-instancing can increase storage efficiency.

(Continued)

Property name	Data type	Description	Notes
			Without single-instancing, storage efficiency is typically around 77%. However, storing mostly small objects without single-instancing can decrease storage efficiency. Storing mostly large objects with single-instancing can result in a storage efficiency that's greater than 100%. The process of repairing storage can temporarily increase storage efficiency. If the percent of used storage out of total storage is less than five percent, the value of the efficiency property is **0.0**.
freeBytes	Long	Specifies the amount of storage, in bytes, that is currently available to be allocated for storing, protecting, and repairing object data and metadata.	Free storage equals total storage minus used storage.
idealEfficiency	Double	Specifies a percent representing the optimal ratio between ingested data and used storage for the S Series Node.	
objectCount	Long	Specifies the total number of objects currently stored in all buckets on the S Series Node.	
partCount	Long	Specifies the total number of parts currently stored for in-progress multipart upload operations in all buckets on the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
percentUsed	Double	Specifies the percent of used storage out of total storage.	If drives are added or become unavailable or objects are stored or deleted while used storage is under repair, this percent can change.
projectedPercentUsed	Double	Specifies the projected percent of used storage out of total storage after all outstanding repairs are complete.	If no storage needs repair, the value of the projectedPercentUsed property equals the value of the percentUsed property.
totalBytes	Long	Specifies the total amount of storage, in bytes, that can be used for storing, protecting, and repairing object data and metadata.	When drives are added to an S Series Node, the total amount of storage increases. If drives become unavailable (for example, due to drive failure), the total amount of storage decreases.
usedBytes	Long	Specifies the amount of storage, in bytes, currently allocated for storing, protecting, and repairing object data and metadata.	The S Series Node preallocates a relatively small amount of storage in anticipation of receiving new data. Therefore, used storage includes both previously allocated storage that now contains object data or metadata and storage that has been allocated but does not yet contain object data or metadata. Used storage can be returned to free storage if object deletions result in a sufficient amount of data and metadata being deleted from the S Series Node. Because storage is allocated in advance, an S Series Node on which no data has been ingested may show a small amount of used storage.

(Continued)

Property name	Data type	Description	Notes
userCount	Integer	Specifies the total number of user accounts that currently exist on the S Series Node.	

/metrics/system example

Here's a sample **GET** request that retrieves statistics about S Series Node capacity usage.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/metrics/system?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/metrics/system?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 192
```

Response body

```
{
  "bucketCount": 1,
  "objectCount": 353884034,
  "partCount": 0,
  "totalBytes": 1993620471152640,
  "usedBytes": 12438091071488,
  "freeBytes": 1981182380081152,
  "percentUsed": 0.0062389463047080075,
  "projectedPercentUsed": 0.0062389463047080075,
  "efficiency": 0.622719876493105,
  "idealEfficiency": 0.7692307692307692,
  "userCount": 3
}
```

/system/irreparables

With the /system/irreparables resource:

- A **GET** request returns a response body.
- A **HEAD** request returns a count of the irreparable objects on the S Series Node in the X-HCPS-Irreparable-Count response header.

For information about the query parameters used to limit the list of irreparable objects returned by a **GET** request, see "Managing resource lists" on page 91.

For more information about the /system/irreparables resource, see "Irreparables resources" on page 77.

/system/irreparables properties

The table below describes the properties in /system/irreparables resource response bodies.

Property name	Data type	Description	Notes
count	Integer	Specifies the value of the **count** query parameter included in the **GET** request or 1,000 if the request did not include the **count** parameter. For more information, see "count query parameter" on page 92.	
irreparables	Array	Specifies a comma-separated list of the irreparable objects that satisfy the request criteria. Each object is represented by the properties described in the next table.	
isTruncated	Boolean	Specifies whether the returned list of irreparable objects is complete. Possible values are: • **true** — The irreparable object list is incomplete. • **false** — The irreparable object list is complete. For more information, see "count query parameter" on page 92.	

(Continued)

Property name	Data type	Description	Notes
marker	String	Specifies the value of the **marker** query parameter included in the **GET** request or no value if the request did not include the **marker** parameter. For more information, see "marker query parameter" on page 93.	
nextMarker	String	If the value of the isTruncated property is **true**, specifies an automatically generated string that identifies the last irreparable object in the returned list. If the value of the isTruncated property is **false**, this property is not included in the response body.	

The table below describes the properties used to represent an irreparable object in the array of irreparable objects returned in response to a **GET** request for the /system/irreparables resource.

Property name	Data type	Description	Notes
bucketId	Integer	Specifies the internal ID for the bucket that contains the irreparable object.	
bucketName	String	Specifies the bucket name.	
irreparableTime	String	Specifies the date and time at which the S Series Node first detected that the object was irreparable, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	

(Continued)

Property name	Data type	Description	Notes
partNumber	Integer	Specifies the part number of uploaded content that's an individual part of an in-progress multipart write.	This property is returned by a **GET** request only if the uploaded content is part of an in-progress multipart write.
path	String	Specifies the full path to and name of the object.	
uploadId	Integer	Specifies the ID of the in-progress multipart write that the uploaded content is part of.	This property is returned by a **GET** request only if the uploaded content is part of an in-progress multipart write.

/system/irreparables examples

The examples below show the use of the /system/irreparables resource with the **GET** and **HEAD** methods.

/system/irreparables GET example

Here's a sample **GET** request that retrieves the first irreparable object in the list of irreparable objects stored on the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/irreparables?count=1
  &prettyprint"
```

Request headers

```
GET /mapi/system/irreparables?count=1&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 244
```

Response body

```
{
  "marker": ""
  "nextMarker": "eyJidWNrZXRJZCI6MSwicGF0aCI6InJoaW5vX2Rpcl9oMV9MMV9kdzEv
cmhpbm9fZmlsZV9oMl9MMV9kdzFfMTAwMCIsInVwbG9hZElkIjotMSwicGFydE51bWJlciI
6LTF9"
  "count": 1,
  "isTruncated": true,
  "irreparables": [
    {"bucketId":1,"bucketName":"hcpsrv-hcp-ma",
"path":"d00/00/00d27c6245a09380c58566158681",
"irreparableTime":"2018-09-08 17:56:02 UTC"
    }
  ]
}
```

/system/irreparables HEAD example

Here's a sample **HEAD** request that retrieves a count of the irreparable objects stored on the S Series Node.

Request with curl command line

```
curl -k -X HEAD -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/irreparables?prettyprint"
```

Request headers

```
HEAD /mapi/system/irreparables?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
X-HCPS-Irreparable-Count: 2
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/system/license

With the /system/license resource, a **GET** request returns a response body.

For more information about the /system/license resource, see "License resource" on page 78.

/system/license properties

The table below describes the properties in /system/license resource response bodies.

Property name	Data type	Description	Notes
capacity	Long	Always **0**.	
expirationDate	Timestamp	Always **1970-01-01 00:00:00 UTC**.	
status	String	Always **EXTERNAL**.	
type	String	Always **EXTERNAL**.	

/system/license example

Here's a sample **GET** request that retrieves information about the S Series Node license.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/license?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/system/license?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 175
```

Response body

```
{
  "type": "EXTERNAL",
  "status": "EXTERNAL",
  "capacity": 0,
  "expirationDate": "1970-01-01 00:00:00 UTC"
}
```

/system/logs/cancel

With the /system/logs/cancel resource, a **POST** request does not take a request body and does not return a response body.

For more information about the /system/logs/cancel resource, see "Internal logs" on page 51"Log resources" on page 78 and "Downloading the internal logs" on page 426.

/system/logs/download

With the /system/logs/download resource, a **GET** request streams the zipped log files to the HttpResponse object, from which you can retrieve the data and write it to a specified file.

For more information about the /system/logs/download resource, see "Log resources" on page 78 and "Downloading the internal logs" on page 426.

/system/logs/mark

With the /system/logs/mark resource, a **POST** request requires a query parameter. The request does not take a request body and does not return a response body.

For more information about the /system/logs/mark resource, see "Log resources" on page 78.

/system/logs/mark query parameter

To insert a comment into the S Series Node internal logs, you use the **message** query parameter with a **POST** request for the /system/logs/mark resource. Valid values for this parameters are text strings. The text must be 1 through 1,024 characters long and can contain any valid UTF-8 characters, including percent-encoded white space.

For more information about query parameters, see "Management API query parameters" on page 61.

/system/logs/mark example

Here's a sample **POST** request that inserts a comment into the S Series Node internal logs.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/logs/mark
  ?message=SM1%20issue%20noted"
```

Request headers

```
POST /mapi/configuration/mapi/system/logs/mark?message=SM1%20issue%20noted
HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/system/logs/prepare

With the /system/logs/prepare resource, a **POST** request requires at least one query parameter. The request does not take a request body and does not return a response body.

For more information about the /system/logs/prepare resource, see "Log resources" on page 78 and "Downloading the internal logs" on page 426.

/system/logs/prepare query parameters

To specify the time period for the S Series Node internal logs you want to download, you use these query parameters with a **POST** request for the /system/logs/prepare resource:

- **startDate** — Specifies the date of the earliest logs you want to include in the download. Logs are included starting from 12:00 a.m.on the specified date.

  This parameter is optional. If the **POST** request does not include this parameter, logs are included starting from 12:00 a.m. on the current day.

  You cannot specify a start date that's later than the date specified by the **endDate** parameter or that's more than 120 days in the past.

- **endDate** — Specifies the date of the latest logs you want to include in the download. Logs are included up to 1:00 a.m., inclusive, on the day following the specified date.

  This parameter is required.

Valid values for the **startDate** and **endDate** parameters are dates in this format:

MM/dd/yyyy

For example:

```
11/09/2018
```

For more information about the /system/logs/prepare resource, see "Log resources" on page 78 and "Downloading the internal logs" on page 426.

/system/logs/prepare example

Here's a sample **POST** request that starts the process of preparing the S Series Node internal logs for download.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/prepare?startDate=11/09/2018
  &endDate=11/10/2018"
```

Request headers

```
POST /mapi/configuration/mapi/system/logs/prepare?startDate=11/09/2018
&endDate=11/10/2018 HTTP/1.1
Host: 10.0.0.3:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

For more information about the /system/logs/prepare resource, see "Log resources" on page 78 and "Downloading the internal logs" on page 426.

/system/logs/status

With the /system/logs/status resource, a **GET** request returns a response body.

For more information about the /system/logs/status resource, see "Log resources" on page 78 and "Downloading the internal logs" on page 426.

/system/logs/status properties

The table below describes the properties in /system/logs/status resource response bodies.

Property name	Data type	Description	Notes
downloadComplete	Boolean	Specifies whether the current log download operation is complete. Possible values are: • **true** — The log download operation finished successfully. • **false** — The log download operation either is in progress or ended with an error.	After the internal logs have been downloaded, the value of this property remains **true** until a **POST** request for the /system/logs/cancel resource is processed.
downloadError	Boolean	Specifies whether an error occurred during the current log download operation. Possible values are: • **true** — An error occurred during the current log download operation. • **false** — No errors have occurred during the current log download operation.	

(Continued)

Property name	Data type	Description	Notes
downloadReadyForStreaming	Boolean	Specifies whether log preparation is complete and the internal logs are ready to be downloaded. Possible values are: • **true** — The internal logs have been prepared and are ready to be downloaded. • **false** — The internal logs are not ready to be downloaded.	
downloadStarted	Boolean	Specifies whether a log download operation is currently in progress. Possible values are: • **true** — A log download operation is in progress. • **false** — No log download operation is in progress.	The value of this property changes from **false** to **true** when a **POST** request for the /system/logs/prepare resource is processed. The value remains **true** until a **POST** request for the /system/logs/cancel resource is processed.
downloadStreamingInProgress	Boolean	Specifies whether the internal logs are in the process of being downloaded. Possible values are: • **true** — The internal are in the process of being downloaded. • **false** — The internal logs are not in the process of being downloaded.	

For an example of using the /system/logs/status resource, see "Downloading the internal logs" on page 426.

/system/logs/status example

Here's a sample **GET** request that retrieves the log download operation status.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/status?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/system/logs/status?prettyprint HTTP/1.1
Host: 10.0.0.3:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 162
```

Response body

```
{
  "downloadReadyForStreaming": false,
  "downloadStreamingInProgress": false,
  "downloadStarted": true,
  "downloadError": false,
  "downloadComplete": false
}
```

For an example of using the /system/logs/status resource, see "Downloading the internal logs" on page 426.

/system/misc/settings/network/management/monitor

With the /system/misc/setting/network/management/monitor resource:

- A **GET** request returns a response body.
- A **POST** request requires a query parameter. The request does not take a request body and does not return a response body.

For more information about the /system/misc/setting/network/management/monitor resource, see "Miscellaneous settings resource" on page 83.

/system/misc/settings/network/management/monitor property

The table below describes the property in /system/misc/settings/network/management/monitor resource response bodies.

Property name	Data type	Description	Notes
monitoringDisabled	Boolean	Specifies whether monitoring is disabled for the management network. Possible values are: • **true** — Management network monitoring is disabled. • **false** — Management network monitoring is enabled. The default is **false**.	

/system/misc/settings/network/management/monitor query parameter

To enable or disable management network monitoring, you use the **disable** query parameter with a **POST** request for the /system/misc/settings/network/management/monitor resource. Valid values for this parameter are:

- **true** — Disables management network monitoring.

- **false** — Enables management network monitoring.

These values are not case sensitive.

For more information about query parameters, see "Management API query parameters" on page 61.

/system/misc/settings/network/management/monitor examples

The examples below show the use of the /system/misc/settings/network/management/monitor resource with the **GET** and **POST** methods.

/system/misc/settings/network/management/monitor GET example
Here's a sample **GET** request that retrieves the current setting for management network monitoring.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/misc/settings/network
  /management/monitor?prettyprint"
```

Request headers

```
GET /mapi/system/misc/settings/network/management/monitor?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 33
```

Response body

```
{
  "monitoringDisabled": false
}
```

/system/misc/settings/network/management/monitor POST example

Here's a sample **POST** request that disables management network monitoring.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/misc/settings/network
  /management/monitor?disable=true"
```

Request headers

```
POST /mapi/system/misc/settings/network/management/monitor?disable=true
HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/system/status/full

With the /system/status/full resource, a **GET** request returns a response body.

For more information about the /system/full health resource, see "Status resources" on page 86.

/system/status/full properties

The table below describes the properties in /system/status/full resource response bodies.

Property name	Data type	Description	Notes
alertInfo	Object	Specifies a set of properties that provide information about the alerts currently in effect for the S Series Node. For descriptions of these properties, see "/alerts properties" on page 99.	The response body includes current alerts with all severities and scopes.
bucketMetrics	Object	Specifies a set of properties that provide information about bucket usage. For descriptions of these properties, see "/metrics/buckets properties" on page 359.	
gatewayMetrics	Object	Specifies a set of properties that provide information about use of the Hitachi API for Amazon S3 (the S3 compatible API). For descriptions of these properties, see "/metrics/gateways properties" on page 362.	
hardware	Object	Specifies a set of properties that provide information about the S Series Node hardware. For descriptions of these properties, see "/hardware properties" on page 187.	
protectionMetrics	Object	Specifies a property that provides a count of bytes under repair. For a description of this property, see "/metrics/protection property" on page 369.	

(Continued)

Property name	Data type	Description	Notes
serverModuleIpAddress	String	Specifies the IP address of the server module that responded to the **GET** request for the /system/status/full resource.	
systemHealth	Object	Specifies a set of properties that provide brief information about the status of the S Series Node. For descriptions of these properties, see "/system/status/health properties" on page 393.	
systemIdentification	Object	Specifies a set of properties that provide identifying information about the S Series Node. For descriptions of these properties, see "/configuration/ident properties" on page 131.	
systemMetrics	Object	Specifies a set of properties that provide information about S Series Node capacity usage. For descriptions of these properties, see "/metrics/system properties" on page 370.	

/system/status/full example

Here's a sample **GET** request that retrieves complete information about the current state of the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/status/full?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/system/status/full?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
```

Response body

```
{
  "systemStatus": {
    "serverModuleIpAddress": "10.0.0.4",
    "systemIdentification": {
      See the response body in "/configuration/ident example" on page 132.
    },
    "systemHealth": {
      See the response body in "/system/status/health example" on page 394.
    },
    "alertInfo": {
      See the response body in "/alerts example" on page 105.
    },
    "systemMetrics": {
      See the response body in "/metrics/system example" on page 373.
    },
    "bucketMetrics": {
      See the response body in "/metrics/buckets example" on page 360.
    },
    "gatewayMetrics": {
      See the response body in "/metrics/gateways example" on page 367.
    },
    "protectionMetrics": {
      See the response body in "/metrics/protection example" on page 369.
    },
    "hardware": {
      See the response body in "/hardware example" on page 282.
```

```
        }
      }
    }
```

/system/status/health

With the /system/status/health resource, a **GET** request returns a response body.

For more information about the /system/status health resource, see "Status resources" on page 86.

/system/status/health properties

The table below describes the properties in /system/status/health resource response bodies.

Property name	Data type	Description	Notes
message	String	Specifies the text of all current alerts at the level indicated by the state property.	
state	String	Specifies the current state of the S Series Node. Possible values are: • **NORMAL** — The S Series Node has no alerts at the warning or error level. • **DEGRADED** — The S Series Node has at least one alert at the warning level and no alerts at the error level. • **CRITICAL** — The S Series Node has at least one alert at the error level.	

/system/status/health example

Here's a sample **GET** request that retrieves brief information about the current state of the S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/status/health?prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/system/status/health?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 75
```

Response body

```
{
  "state": "CRITICAL",
  "message": "Server module 1 is unavailable."
}
```

/system/update/apply

With the /system/update/apply resource, a **POST** request does not take a request body and does not return a response body.

For more information about the /system/update/apply resource, see:

- "Hotfixes" on page 52
- "Update resources" on page 88

- "Performing an update operation" on page 435

/system/update/history

With the /system/update/history resource, a **GET** request returns a response body.

For more information about the /system/update/history resource, see "Update resources" on page 88.

/system/update/history properties

The table below describes the top-level property in /system/update/history resource response bodies.

Property name	Data type	Description	Notes
historyList	Array	Specifies a comma-separated list of the software updates that have been made to the S Series Node, in descending order by time. Each update is represented by the properties described in the next table.	

The table below describes the properties used to represent an update in the array of updates returned in the response to a **GET** request for the /system/update/history resource.

Property name	Data type	Description	Notes
manifest	Object	Specifies a set of properties that represent the update file manifest. These properties are described in the next table.	

(Continued)

Property name	Data type	Description	Notes
timestamp	Timestamp	Specifies the time at which the update finished, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	
title	String	Specifies the title of the update from the update file.	

The table below describes the properties used to represent an update file manifest in the response to a **GET** request for the /system/update/history resource.

Property name	Data type	Description	Notes
description	String	Specifies a description of the update.	
impact	String	Specifies the level of impact making the update had on the S Series Node. Possible values are: • **NONE** • **LOW** • **MEDIUM** • **HIGH**	
impactDescription	String	Specifies a description of the impact making the update had on the S Series Node.	
title	String	Specifies the title of the update from the update file manifest.	
version	String	Specifies the version of the HCP S Series software in the update file.	

/system/update/history example

Here's a sample **GET** request that retrieves the update history for the HCP S Series Node.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/history?prettyprint"
```

Request headers

```
GET /system/update/history?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 1247
```

Response body

```
{
  "historyList": [
    {
      "timestamp": "2019-05-01 09:23:56 UTC",
      "title": "HCP S Series 3.0.0 Hotfix HF001",
      "manifest": {
        "title": "HCP S Series 3.0.0 Hotfix HF001",
        "description": "Installs hotfix 0001 for release 3.0.0.This hotfix resolves several
issues on the HCP S Series Node.",
        "impact": "MEDIUM",
        "impactDescription": "You can apply this hotfix only to a release 3.0.0 S Series
Node. Server modules are updated one at a time. While a server module is being
updated, it is unavailable. You can make configuration changes while the S Series Node is
being updated, but certain changes won’t take effect until the update is complete.",
        "version": "3.0.0.2"
      }
    },
    {
      "timestamp": "2019-04-25 13:50:44 UTC",
      "title": "HCP S Series 3.0.0.2",
      "manifest": {
        "title": "HCP S Series 3.0.0.2",
        "description": "System installed to version 3.0.0.2",
        "impact": "NONE",
        "impactDescription": "No impact.",
        "version": "3.0.0.2"
      }
    }
  ]
}
```

/system/update/manifest

With the /system/update/manifest resource, a **GET** request returns a response body.

For more information about the /system/update/manifest resource, see "Update resources" on page 88.

/system/update/manifest properties

The table below describes the properties in /system/update/manifest resource response bodies.

Property name	Data type	Description	Notes
description	String	Specifies a description of the update.	
impact	String	Specifies the level of impact the update will have on the S Series Node. Possible values are: • **NONE** • **LOW** • **MEDIUM** • **HIGH**	
impactDescription	String	Specifies a description of the impact the update will have on the S Series Node.	
title	String	Specifies the title of the update from the update file manifest.	
version	String	Specifies the version of the HCP S Series software in the update file.	

/system/update/manifest example

Here's a sample **GET** request that retrieves the manifest for the currently uploaded update file.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/manifest
  ?prettyprint"
```

Request headers

```
GET /system/update/manifest?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.0.18
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 599
```

Response body

```
{
  "title": "HCP S Series 3.0.0 Hotfix HF001",
  "description": "Installs hotfix 0001 for release 3.0.0.This hotfix resolves several issues
on the HCP S Series Node.",
  "impact": "MEDIUM",
  "impactDescription": "You can apply this hotfix only to a release 3.0.0 S Series Node.
Server modules are updated one at a time. While a server module is being updated, it is
unavailable. You can make configuration changes while the S Series Node is being
updated, but certain changes won’t take effect until the update is complete.",
  "version": "3.0.0.2"
}
```

/system/update/progress

With the /system/update/progress resource, a **GET** request returns a response body.

For more information about the /system/update/progress resource, see:

- "Hotfixes" on page 52
- "Update resources" on page 88
- "Performing an update operation" on page 435

/system/update/progress properties

The table below describes the top-level property in /system/update/progress resource response bodies.

For the first few seconds after you apply an update, a **GET** request for the /system/update/progress resource returns a status code of 503 (Service Unavailable).

If the state of an update operation, as indicated by the state property of the /system/update/status resource, is not UPGRADING or ERROR, a **GET** request for the /system/update/progress resource returns a status code of 400 (Bad Request).

Property name	Data type	Description	Notes
updateProgress	Object	Specifies a set of properties that provide information about the progress of a current update operation. These properties are described in the next table.	

The table below describes the properties used to provide information about the progress of a current update operation in the response to a **GET** request for the /system/update/progress resource.

Property name	Data type	Description	Notes
preupdateProgress	Object	Specifies a set of properties that describe the progress of the process of preparing the server modules to be updated. For descriptions of these properties, see "Preparation progress properties" on the next page.	

(Continued)

Property name	Data type	Description	Notes
serverModules	Array	Specifies a comma-separated list of the server modules in the S Series Node, where each module is represented by a set of properties that provide information about the progress of the current update operation on that module. For descriptions of these properties, see "Server module update progress properties" on the next page.	

Preparation progress properties

The table below describes the properties used to represent the progress of the process of preparing the server modules to be updated in the response to a **GET** request for the /system/update/progress resource.

Property name	Data type	Description	Notes
message	String	Specifies a description of the current state of the update preparation process.	
percentComplete	Integer	Specifies how much of the preparation processing has been completed, as a percent, from zero to 100, of the total amount of processing required to prepare the server modules for the update.	

Server module update progress properties

The table below describes the properties used to represent the progress of the current update operation on a server module in the response to a **GET** request for the /system/update/progress resource.

Property name	Data type	Description	Notes
message	String	Specifies a description of the action the server module is currently performing.	
percentComplete	Integer	Specifies how much of the update processing the server module has completed, as a percent, from zero to 100, of the total amount of processing required to complete the update on that server module.	When an update is 100% complete on a server module, the module automatically reboots.
serverModuleNumber	Integer	Specifies the server module number.	

/system/update/progress example

Here's a sample **GET** request that retrieves the information about the progress of the current update operation.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/progress
  ?prettyprint"
```

Request headers

```
GET /system/update/progress?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 459
```

Response body

```
{
  "updateProgress": {
    "preUpdateProgress: {
      "message": "Complete",
      "percentComplete": 100
    },
    "serverModules": [
      {
        "serverModuleNumber": "2",
        "message": "Waiting for other module.",
        "percentComplete": 0
      },
     {
        "serverModuleNumber": "1",
        "message": "Rebooting server module",
        "percentComplete": 100
      }
    ]
  }
}
```

/system/update/restart

With the /system/update/restart resource, a **POST** request does not take a request body and does not return a response body.

For more information about the /system/update/restart resource, see:

- "Hotfixes" on page 52
- "Update resources" on page 88
- "Performing an update operation" on page 435

/system/update/status

With the /system/update/status resource, a **GET** request returns a response body.

For more information about the /system/update/status resource, see:

- "Hotfixes" on page 52
- "Update resources" on page 88
- "Performing an update operation" on page 435

/system/update/status property

The table below describes the property in /system/update/status resource response bodies.

Property name	Data type	Description	Notes
state	String	Specifies the current status of an update operation on the S Series Node. Possible values are: • **READY** — No update is currently in progress. You can now upload an update file to start an update operation. • **EXTRACTING** — The S Series Node is currently extracting files from an update file. This is part of the upload step of an update operation. • **EXTRACTED** — An update file has been uploaded and files have been extracted from it. You can now start the apply step of the update operation.	

(Continued)

Property name	Data type	Description	Notes
		• **PREUPGRADE** — The S Series Node is performing prechecks to ensure that it is ready for a software update. This is part of the apply step of an update operation. • **UPGRADING** — The S Series Node is in the process of updating the HCP S Series software. • **COMPLETE** — An update operation has finished successfully. This is a transient state that will eventually change to **CLEANUP**. • **ERROR** — An update operation ended with an error. You can try restarting the update operation. • **CLEANUP** — The S Series Node is deleting the temporary files it created during a successful update operation. This is a transient state that will eventually change to **READY**.	

/system/update/status example

Here's a sample **GET** request that retrieves the status of the current update operation.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/status?prettyprint"
```

Request headers

```
GET /system/update/status?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 27
```

Response body

```
{
  "state": "EXTRACTED"
}
```

/system/update/upload/software

With the /system/update/upload/software resource, a **PUT** request requires an update file as input and returns a response body.

For more information about the /system/update/upload/software resource, see:

- "Hotfixes" on page 52
- "Update resources" on page 88
- "Performing an update operation" on page 435

/system/update/upload/software properties

The table below describes the properties in /system/update/upload/software resource response bodies.

Property name	Data type	Description	Notes
description	String	Specifies a description of the update.	
impact	String	Specifies the level of impact the update will have on the S Series Node. Possible values are: • **NONE** • **LOW** • **MEDIUM** • **HIGH**	
impactDescription	String	Specifies a description of the impact the update will have on the S Series Node.	
title	String	Specifies the title of the update from the update file manifest.	
version	String	Specifies the version of the HCP S Series software in the update file.	

/system/update/upload/software example

Here's a sample **PUT** request that uploads a hotfix file.

Request with curl command line

```
curl -k -T HCPS_Hotfix_3.0.0.2_HF0001.bin
  -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/upload/software
  ?prettyprint"
```

Request headers

```
PUT /system/update/upload/software?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
Content-Length: 501155840
```

Response headers

```
HTTP/1.1 201 Created
Server: HCP S Series/1.0.0.18
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 599
```

Response body

```
{
  "title": "HCP S Series 3.0.0 Hotfix HF001",
  "description": "Installs hotfix 0001 for release 3.0.0.This hotfix resolves several issues
on the HCP S Series Node.",
  "impact": "MEDIUM",
  "impactDescription": "You can apply this hotfix only to a release 3.0.0 S Series Node.
Server modules are updated one at a time. While a server module is being updated, it is
unavailable. You can make configuration changes while the S Series Node is being
updated, but certain changes won't take effect until the update is complete.",
  "version": "3.0.0.2"
}
```

/user_accounts

With the /user_accounts resource:

- A **PUT** request requires a request body.
- A **GET** request returns a response body.

For information about the query parameters used to limit the user account list returned by a **GET** request, see "Managing resource lists" on page 91.

For more information about the /user_accounts resource, see "User account resources" on page 89.

/user_accounts properties

The table below describes the properties in /user_accounts resource response bodies. For the properties for /user_accounts resource request bodies used with **PUT** requests, see "/user_accounts/username properties" on page 414.

Property name	Data type	Description	Notes
count	Integer	Specifies the value of the **count** query parameter included in the **GET** request or 1,000 if the request did not include the **count** parameter. For more information, see "count query parameter" on page 92.	
isTruncated	Boolean	Specifies whether the returned list of user accounts is complete. Possible values are: • **true** — The user account list is incomplete. • **false** — The user account list is complete. For more information, see "count query parameter" on page 92.	
marker	String	Specifies the value of the **marker** query parameter included in the **GET** request or no value if the request did not include the **marker** parameter. For more information, see "marker query parameter" on page 93.	

(Continued)

Property name	Data type	Description	Notes
prefix	String	Specifies the value of the **prefix** query parameter included in the **GET** request or no value if the request did not include the **prefix** parameter. For more information, see "prefix query parameter" on page 94.	
username	Array	Specifies a comma-separated list of the user accounts that satisfy the request criteria. Each user account is represented by the value of its username property.	

/user_accounts examples

The examples below show the use of the /user_accounts resource with the **PUT** and **GET** methods.

/user_accounts PUT example

Here's a sample **PUT** request that creates a user account.

Request body

```
{
  "username": "lgreen",
  "password": "Welcome1!",
  "fullName": "Lee Green",
  "description": "Storage management group manager with security
privileges",
  "roles": [
    "admin",
    "security"
  ],
  "forcePasswordChange": true,
  "enabled": true
}
```

Request with curl command line

```
curl -k -T user_create.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts?prettyprint"
```

Request headers

```
PUT /mapi/user_accounts?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
Content-Type: application/json
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
Content-Length: 255
```

Response headers

```
HTTP/1.1 201 Created
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/user_accounts GET example

Here's a sample **GET** request that retrieves a list of existing user accounts.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts?prettyprint"
```

Request headers

```
GET /mapi/user_accounts?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 203
```

Response body

```
{
  "marker": "",
  "prefix": "",
  "count": 1000,
  "isTruncated": false,
  "username": [
    "admin",
    "hcpsrv-hcp-ma",
    "it-rbrown",
    "lgreen",
    "mwhite",
    "pblack",
    "pdgrey"
  ]
}
```

/user_accounts/*username*

With the /user_accounts/*username* resource:

- A **GET** request returns a response body.
- A **POST** request requires a request body.
- **HEAD** and **DELETE** requests do not take a request body and do not return a response body.

Note: If you do not have the security role, the only method you can use with this resource identifier is **POST**, the only user account you can modify is the one identified by the credentials specified in the request, and the only valid property for the request body is password.

For more information about the /user_accounts/*username* resource, see "User account resources" on page 89.

/user_accounts/*username* properties

The table below describes the properties in /user_accounts/*username* resource request and response bodies. These properties apply to an individual user account. They are also used in the request body for **PUT** requests with the /user_accounts resource.

Property name	Data type	Description	Notes
creationTime	Timestamp	Specifies the date and time at which the user account was created, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	This property is not valid on a **PUT** or **POST** request.
description	String	Specifies a description of the user account. This description is optional. Descriptions can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space. To remove a description from a user account, specify the description property with no value.	This property is optional on a **PUT** request. It is valid on a **POST** request only if the user making the request has the security role.

(Continued)

Property name	Data type	Description	Notes
enabled	Boolean	Indicates whether the user account is enabled. Valid values are: • **true** — The user account is enabled. • **false** — The user account is disabled. These values are not case sensitive.	This property is required on a **PUT** request. It is valid on a **POST** request only if the user making the request has the security role.
failedLoginAttempts	Integer	Specifies the number of times an attempt to access the Management Console or management API with the user account username has failed since the last successful access.	This property is not valid on a **PUT** or **POST** request.
forcePasswordChange	Boolean	Indicates whether the password for the user account must be changed before the account can be used with the Management Console or management API for any purpose other than to change the password (that is, whether the password is expired). Valid values are: • **true** — The password must be changed. • **false** — The password does not need to be changed. These values are not case sensitive.	This property is required on a **PUT** request. It is valid on a **POST** request only if the user making the request has the security role.

(Continued)

Property name	Data type	Description	Notes
fullName	String	Specifies the full name of the intended user of the account. This name must be 1 through 256 characters long and can contain any valid UTF-8 characters, including white space.	This property is required on a **PUT** request. It is valid on a **POST** request only if the user making the request has the security role.
lastLoginTime	String	Specifies the last time the user account was used to access the Management Console or management API, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	This property is not valid on a **PUT** or **POST** request. It is returned by a **GET** request only if the user account has already been used to access the Management Console or management API at least once.
lastPasswordChangeTime	Timestamp	Specifies the last time the password for the user account was changed, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2018-09-20 18:28:57 UTC	This property is not valid on a **PUT** or **POST** request.

(Continued)

Property name	Data type	Description	Notes
password	String	Specifies the password for the user account. For the rules for passwords, see "Passwords" on page 22.	This property is required on a **PUT** request. It is valid on a **POST** request only if either of these is true: • The user making the request has the security role. • The user account being modified is the one used for the request credentials. This property is not returned by a **GET** request.
passwordExpires	Boolean	Specifies whether the password for the user account ever expires automatically based on the S Series Node security setting for password expiration.	This property is required on a **PUT** request. It is valid on a **POST** request only if the user making the request has the security role.
roles	Array	Associates one or more comma-separated roles with the user account. Valid values for roles are: • admin • data • monitor • security • service These values are case sensitive.	This property is required on a **PUT** request. It is valid on a **POST** request only if the user making the request has the security role. With a **POST** request, the set of roles specified in the request body replaces the set of roles currently associated with the user account.

(Continued)

Property name	Data type	Description	Notes
userID	Integer	Specifies the internal ID for the user account. The S Series Node generates this ID automatically when the user account is created.	This property is not valid on a **PUT** or **POST** request.
username	String	Specifies the username for the user account. For the rules for usernames, see "Usernames" on page 21.	This property is required on a **PUT** request. It is not valid on a **POST** request.

/user_accounts/*username* example

Here's a sample **GET** request that retrieves information about the user account with username *lgreen*.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts/lgreen?prettyprint"
```

Request headers

```
GET /mapi/user_accounts/lgreen?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 445
```

Response body

```
{
  "passwordExpires": true,
  "lastPasswordChangeTime": "2018-10-17 15:02:35 UTC",
  "username": "lgreen",
  "userID": 3,
  "description": "Storage management group manager with security privileges",
  "roles": [
    "security",
    "admin"
  ],
  "fullName": "Lee Green",
  "forcePasswordChange": false,
  "enabled": true,
  "creationTime": "2018-09-23 13:48:28 UTC",
  "failedLoginAttempts": 0,
  "lastLoginTime": "2018-10-17 16:52:03 UTC"
}
```

/user_accounts/*username*/access_key/generate

With the /user_accounts/*username*/access_key/generate resource, a **POST** request returns a response body. The request does not take a request body.

Note: The username specified in the /user_accounts/*username*/access_key/generate resource identifier must match the username for the credentials used in the **POST** request.

For more information about the /user_accounts/*username*/access_key/generate resource, see "User account resources" on page 89.

/user_accounts/*username*/access_key/generate properties

The table below describes the properties in /user_accounts/*username*/access_key/generate resource response bodies.

Property name	Data type	Description	Notes
accessKeyID	String	Specifies the access key for the user account.	
secretAccessKey	String	Specifies the secret key for the user account.	

/user_accounts/*username*/access_key/generate example

Here's a sample **POST** request that generates new access and secret keys for the user account with username *mwhite*.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic bXdoaXRlOk13aGl0ZTEh"
  "https://mapi.s-node-1.example.com:9090/mapi/user_accounts/mwhite/access_key
  /generate?prettyprint"
```

Request headers

```
POST /mapi/user_accounts/mwhite/access_key/generate?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 108
```

Response body

```
{
  "accessKeyID": "fwRbupbFRY46VhAWPkUz",
  "secretAccessKey": "awZfRok8EjJEdd1pOAlQmSnrshgLG4U6MVAvCzPM"
}
```

/versions

With the /versions resource:

- A **GET** request returns a response body.
- A **POST** request requires a query parameter and returns a response body. The request does not take a request body.

The response bodies returned by **GET** and **POST** requests differ from each other.

For more information about the /versions resource, see "Versions resource" on page 91.

/versions GET properties

The table below describes the properties in /versions resource response bodies that are returned in response to **GET** requests.

Property name	Data type	Description	Notes
latestVersion	String	Specifies the most current version of the HCP S Series management API supported by the S Series Node.	
supportedVersions	Array	Specifies a comma-separated list of the supported versions of the HCP S Series management API.	

/versions POST query parameter and properties

To check whether a specific version of the HCP S Series management API is supported by the current release of the S Series Node, you use the **version** query parameter with a **POST** request for the /versions resource. The value of this query parameter is the version you're checking.

If the request is valid, the S Series Node returns a 200 (OK) status code, regardless of whether the version you're checking is supported.

The table below describes the properties in the response body for a **POST** request with the **version** query parameter.

Property name	Data type	Description	Notes
supported	Boolean	Indicates whether the version specified by the **version** query parameter is a supported version of the HCP S Series management API. Possible values are: • **true** — The specified version is supported. • **false** — The specified version is not supported. These values are not case sensitive.	
version	String	Specifies the version being checked (that is, the version specified by the **version** query parameter).	

For more information about query parameters, see "Management API query parameters" on page 61.

/versions examples

The examples below show the use of the /versions resource with the **GET** and **POST** methods.

/versions GET example

Here's a sample **GET** request that retrieves information about the supported versions of the HCP S Series management API.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/versions?prettyprint"
```

Request headers

```
GET /mapi/versions?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 85
```

Response body

```
{
  "supportedVersions": [
    "1.0.0",
    "1.0.1",
    "2.0.0",
    "2.1.0",
    "2.2.0",
    "3.0.0"
  ],
  "latestVersion": "3.0.0"
}
```

/versions POST example

Here's a sample **POST** request that checks whether version 2.1.0 of the HCP S Series management API is supported by the S Series Node.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/versions?version=2.1.0&prettyprint"
```

Request headers

```
POST /mapi/versions?version=2.1.0&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 3.0.0
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/3.0.0.2
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 1
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1, 2.0.0, 2.1.0, 2.2.0, 3.0.0]
X-HCPS-API-VERSION: 3.0.0
Content-Type: application/json;charset=UTF-8
Content-Length: 46
```

Response body

```
{
  "version": "2.1.0",
  "supported": true
}
```

6

Management API procedures

Most management API resources are used to perform a standalone action, such as retrieving hardware information, creating a user account, reconfiguring a network, or turning on beaconing for a server module. A few resources, however, are used for performing procedures that entail multiple steps. The procedures you can perform with these resources are:

- Downloading the S Series Node internal logs
- Managing these hardware maintenance procedures:
  - Adding, removing, or replacing data and database drives
  - Adding, removing, or replacing enclosures
- Performing an update operation to apply a hotfix

Note: To perform the procedures listed above, other than downloading the internal logs, you must be an authorized service provider. Customers are not allowed to perform these procedures by themselves.

Downloading the internal logs

A log download operation starts with a request to prepare the S Series Node internal logs for download. The operation ends with a request to reset the S Series Node to be ready for a new log download operation.

Considerations for downloading the internal logs

When using the management API to perform a log download operation:

- In the resource URL in each request you issue as part of a log download procedure, use the physical IP address for a server module instead of the S Series Node domain name or a virtual IP address.

- Use the same IP address in all the requests that make up a single log download operation.

- Do *not* try to cancel a log download operation while the logs are being prepared for download or while the download is in progress. You can, however, cancel the operation at the point when the log preparation is complete and you have not yet started the actual download.

- When using cURL, do not include the **-i** or **-v** parameter in the **GET** request for the /system/logs/download resource.

Performing a log download operation

To use the management API to download the S Series Node internal logs:

1. Use a **GET** request for the /system/logs/status resource to verify that the S Series Node is in a state in which you can start a log download operation:

```
curl -k -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/status?prettyprint"
```

You can start a log download operation if the value of each property in the response body is **false**:

```
{
  "downloadReadyForStreaming": false,
  "downloadStreamingInProgress": false,
  "downloadStarted": false,
  "downloadError": false,
  "downloadComplete": false
}
```

2. Use a **POST** request for the /system/logs/prepare resource to start the process of preparing the logs for download:

   ```
   curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://10.0.0.3:9090/mapi/system/logs/prepare?startDate=11/09/2018
     &endDate=11/10/2018"
   ```

3. Use a **GET** request for the /system/logs/status resource to monitor the log preparation. When the value of the downloadReadyForStreaming property is **true**, you can proceed to the next step.

   If the value of the downloadError property is **true**, the preparation process resulted in an error. The only option at this point is to issue a **POST** request for the /system/logs/cancel resource.

4. Use a **GET** request for the /system/logs/download resource to start the process of downloading the prepared logs and writing them to a specified .zip file:

   ```
   curl -k -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://10.0.0.3:9090/mapi/system/logs/download" > logs-11-10-18.zip
   ```

5. Use a **GET** request for the /system/logs/status resource to monitor the log download. When the value of either the downloadComplete property or the downloadError property is **true**, you can proceed to the next step.

   If the value of the downloadError property is **true**, the download process resulted in an error.

6. Use a **POST** request for the /system/logs/cancel resource to reset the S Series Node to be ready for a new log download operation:

   ```
   curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://10.0.0.3:9090/mapi/system/logs/cancel"
   ```

For more information about the S Series Node internal logs, see "Internal logs" on page 51. For more information about the resources used during a log download operation, see "Log resources" on page 78.

Performing a hardware maintenance procedure

A hardware maintenance procedure starts when you issue a **POST** request for the /hardware/maintenance resource specifying the type of procedure you want to perform. It ends when you issue a request to cancel or complete the procedure. At any time while the procedure is active, you can associate notes with it.

To perform a hardware maintenance procedure, you must be an authorized service provider. Customers are not allowed to perform these activities by themselves.

Maintenance procedure steps

The basic steps for performing a hardware maintenance procedure are:

1. Start the procedure (/hardware/maintenance).
2. Retrieve a list of hardware components that are eligible to be targets for the procedure (/hardware/maintenance/*procedure-id*/candidates).
3. Select the target components for the procedure (/hardware/maintenance/*procedure-id*/select).
4. Prepare the S Series Node for the physical portion of the procedure (/hardware/maintenance/*procedure-id*/perform).
5. Physically add, remove, or replace the target components. For instructions on performing these physical tasks, see the applicable HCP S Series Node service documentation.
6. Verify that no errors have occurred during the procedure (/hardware/maintenance/*procedure-id*/verify).

7. For an add drives, replace drives, or add enclosures procedure, if you used any native or foreign drives, specify how you want the S Series Node to handle each one (/hardware/maintenance/*procedure-id*/confirm). A native drive is one that was previously used in the current S Series Node, where the HCP S Series software has not been reinstalled since the drive was removed. A foreign drive is one that was previously used in a different S Series Node or in the current S Series Node before the HCP S Series software was reinstalled.

8. Complete the procedure (/hardware/maintenance/*procedure-id*/complete). You need to perform this step even if errors occurred during the procedure.

Note: When adding or replacing drives or enclosures, the maintenance procedure entails one additional step that cannot be performed with the management API — after verifying the drive or enclosure addition or replacement, you need to verify the firmware on the new drives or enclosure.

Replacing a data or database drive

To use the management API to perform a replace drives procedure:

1. Use a **POST** request for the /hardware/maintenance resource to start the replace drives procedure:

```
curl -k -X POST -d @replace_drives_start.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance
  ?prettyprint"
```

The `replace_drives_start.json` file contains the request body:

```
{
  "maintType": "REPLACE_DRIVE"
}
```

Here's the response body returned by the request:

```
{
  "id": 12,
  "maintType": "REPLACE_DRIVE",
  "state": "STARTED",
  "startTime": "2018-11-02 14:13:13 UTC",
  "startTsExtra": 189,
  "selections": {
    "maintSelections": []
  }
}
```

2. Make a note of the value of the id property in the response body. This is the maintenance procedure ID.

3. Optionally, use a **POST** request for the /hardware/maintenance/*procedure-id*/update resource to add a note to the procedure:

   ```
   curl -k -X POST -d @proc_notes.json -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Content-Type: application/json"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
     /update?prettyprint"
   ```

   The `proc_notes.json` file contains the request body:

   ```
   {
     "notes": "Replacing corrupt drive."
   }
   ```

4. Use a **GET** request for the /hardware/maintenance/*procedure-id*/candidates resource to retrieve a list of slots that are eligible for the replace drives procedure:

   ```
   curl -k -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
     /candidates?prettyprint"
   ```

   Here's the response body returned by the request:

   ```
   {
     "maintSelections": [
       {
       "state": "NONE",
   ```

```
    "code": "NONE",
    "codeString": "None",
    "enclosure": {
      "wwid": "3500c0ff03c8aa83c",
      "id": 1,
      "product": "SP-34100-E12PM",
      "serial": "SGFTJ18263C8AA8",
      "slotNumber": 2
    },
    "drive": {
      "reason": "FAILED"
      "wwid": "35000c500952117eb",
      "vendor": "SEAGATE",
      "product": "ST10000NM0096",
      "serial": "ZA250Z2A0000C825AUZM",
      "capacity": 10000831348736,
      "state": "FAILED",
      "failCode": "DRIVE_CORRUPT"
      }
    }
  ]
}
```

5. Use a **POST** request for the /hardware/maintenance/*procedure-id*/select resource to select the target components for the procedure (in this case, only slot 2):

```
curl -k -X POST -d @slot_selection.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
  /select?prettyprint"
```

The `slot_selection.json` file contains the request body:

```
{
  "maintSelections": [
    {
      "enclosure": {
        "id": 1,
        "slotNumber": 2,
        "wwid": "3500c0ff03c8aa83c"
      },
      "drive": {
        "wwid": "35000c500952117eb"
      }
    }
  ]
}
```

Tip: The easiest way to create the request body for the /hardware/maintenance/*procedure-id*/select resource is to edit the response body returned by the **GET** request for the /hardware/maintenance/*procedure-id*/candidates resource.

6. Use a **POST** request for the /hardware/maintenance/*procedure-id*/perform resource to prepare the S Series Node for the physical portion of the procedure:

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
  /perform?prettyprint"
```

Here's the response body returned by the request:

```
{
  "id": 12,
  "maintType": "REPLACE_DRIVE",
  "state": "PERFORMING",
  "startTime": "2018-11-02 14:13:26 UTC",
  "startTsExtra": 189,
  "notes": "Replacing corrupt drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 2
        },
        "drive": {},
        "replacedDrive": {
          "reason": "FAILED"
          "wwid": "35000c500952117eb",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
          "serial": "ZA250Z2A0000C825AUZM",
          "capacity": 10000831348736,
          "state": "FAILED",
          "failCode": "DRIVE_CORRUPT"
        }
      }
    ]
  }
}
```

You can perform the next step of the replace drives procedure if the value of the state property for the procedure (the first occurrence of a state property in the response body) is **PERFORMING**.

7. Physically replace the old drive in the selected slot with a new drive. For instructions on doing this, see the applicable HCP S Series Node service documentation.

8. Use a **POST** request for the /hardware/maintenance/*procedure-id*/verify resource to check whether the replace drives procedure was performed correctly and whether the newly inserted drive was previously used in the same or a different S Series Node:

   ```
   curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
     /verify?prettyprint"
   ```

   Here's the response body returned by the request:

   ```
   {
     "id": 12,
     "maintType": "REPLACE_DRIVE",
     "state": "ACTION",
   ```

```
  "startTime": "2018-11-02 14:13:13 UTC",
  "startTsExtra": 189,
  "notes": "Replacing corrupt drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ACTION_FOREIGN",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500c0ff03c8aa83c",
          "id": 1,
          "product": "SP-34100-E12PM",
          "serial": "SGFTJ18263C8AA8",
          "slotNumber": 2
        },
        "drive": {
          "reason": "NONE"
          "wwid": "35000c50095210acb",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
           "serial": "ZA250Z9Q0000C8261VP7",
           "capacity": 10000831348736,
           "state": "DISCOVERED",
           "failCode": "NONE"
        },
        "replacedDrive": {
          "reason": "FAILED"
          "wwid": "35000c500952117eb",
          "vendor": "SEAGATE",
          "product": "ST10000NM0096",
          "serial": "ZA250Z2A0000C825AUZM",
          "capacity": 10000831348736,
          "state": "FAILED",
          "failCode": "DRIVE_CORRUPT"
        }
      }
    ]
  }
}
```

Notice that the value of the state property for the procedure is **ACTION** and the value of the state property for the target slot is **ACTION_ FOREIGN**. This means that the new drive was previously used in a different S Series Node. You need to tell the current S Series Node whether to format this drive or mark it failed.

9. Use a **POST** request for the /hardware/maintenance/*procedure-id*/confirm resource to tell the S Series Node to format the new drive:

```
curl -k -X POST -d @format_drive.json -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Content-Type: application/json"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
  /confirm?prettyprint"
```

The `format_drive.json` file contains the request body:

```
{
  "maintSelections": [
    {
      "enclosure": {
        "id": 1,
        "slotNumber": 2,
        "wwid": "3500c0ff03c8aa83c"
      },
      "drive": {
        "wwid": "35000c50095210acb"
      },
      "confirmAction": "true"
    }
  ]
}
```

10. Use a **POST** request for the /hardware/maintenance/*procedure-id*/complete resource to end the replace drives procedure:

```
curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
  /complete?prettyprint"
```

Performing an update operation

An update operation entails an upload step and an apply step.

Considerations for performing an update operation
When using the management API to perform an update operation:

- You can issue a **PUT** request for the /system/update/upload/software resource only while the state of the current update operation is READY, as reported in the response body returned by a **GET** request for the /system/update/status request.

- You can issue a **POST** request for the /system/update/apply resource only while the state of the current update operation is EXTRACTED, as reported in the response body returned by a **GET** request for the /system/update/status request.

- You can issue a **POST** request for the /system/update/restart resource only while the state of the current update operation is ERROR, as reported in the response body returned by a **GET** request for the /system/update/status request.

For additional considerations for performing update operations, see "Considerations for applying hotfixes" on page 53.

Performing an update operation

To use the management API to perform an update operation on an S Series Node:

1. Use a **GET** request for the /system/update/status resource to verify that the S Series Node is in a state in which you can start an update operation:

   ```
   curl -k -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/system/update/status
     ?prettyprint"
   ```

   You can start an update operation while the value of the state property in the response body is **READY**:

   ```
   {
     "state": "READY"
   }
   ```

2. Use a **PUT** request for the /system/update/upload/software resource to upload the update file:

   ```
   curl -k -T HCPS_Hotfix_3.0.0.2_HF0001.bin
     -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/system/update/upload/software
     ?prettyprint"
   ```

The response body returned by the **PUT** request contains the manifest for the uploaded update file.

3. Review the manifest in the response body to ensure that you uploaded the correct file.

   If the uploaded file is not correct, repeat the **PUT** request with the correct file.

4. Use a **GET** request for the /system/update/status resource to determine whether the uploaded update file is in a state in which you can perform the apply step of the update operation.

   You can perform the apply step while the value of the state property in the response body is **EXTRACTED**:

   ```
   {
     "state": "EXTRACTED"
   }
   ```

5. Use a **POST** request for the /system/update/apply resource to start applying the update:

   ```
   curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/system/update/apply"
   ```

6. Optionally, use repeated **GET** requests for the /system/update/progress resource to view the progress of the update operation:

   ```
   curl -k -H "X-HCPS-API-VERSION: 3.0.0"
     -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
     "https://mapi.s-node-1.example.com:9090/mapi/system/update/progress
     ?prettyprint"
   ```

While the update operation is in progress, the response body returned by the **GET** request looks something like this:

```
{
  "updateProgress": {
    "preUpdateProgress: {
      "message": "Complete",
      "percentComplete": 100
    },
    "serverModules": [
      {
        "serverModuleNumber": "2",
        "message": "Waiting for other module.",
        "percentComplete": 0
      },
      {
        "serverModuleNumber": "1",
        "message": "Rebooting server module",
        "percentComplete": 0
      }
    ]
  }
}
```

After the update operation finishes, either successfully or unsuccessfully, the response body returned by the **GET** request looks like this:

```
{}
```

7. Use a **GET** request for the /system/update/status resource to determine the outcome of the of the update operation:

    - If the value of the state property in the response body is **COMPLETE** or **READY**, the update operation finished successfully.

    - If the value of the state property in the response body is **ERROR**, the update operation ended with an error. In this case, use a **POST** request with the /system/update/restart resource to try to restart the update operation from its last good checkpoint:

      ```
      curl -k -X POST -H "X-HCPS-API-VERSION: 3.0.0"
        -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
        "https://mapi.s-node-1.example.com:9090/mapi/system/update/restart"
      ```

If the update operation ends with an error again, do not try to restart the operation a second time. Instead, contact your authorized service provider for help.

If the value of the state property is anything else, the update operation is still in progress. In this case, repeat the **GET** request until the value of the state property is **COMPLETE**, **READY**, or **ERROR**.

7

Management API HTTP status codes

Code	Meaning	Methods	Description
200	OK	POST GET HEAD DELETE	The S Series Node successfully performed the requested operation.
201	Created	PUT	The S Series Node successfully created the requested resource.
302	Found	HEAD	The resource identified by the URL exists, but the user account identified by the Authorization header doesn't have permission to access the resource.

(Continued)

Code	Meaning	Methods	Description
400	Bad Request	All	The request was not valid. These are some, but not all, of the possible reasons: • The management API version specified by the X-HCPS-API-VERSION request header is not supported. • The URL in the request is not well-formed. • The request is missing a required query parameter. • The request contains a required or optional query parameter with an invalid value. • For a **PUT** or **POST** request, the request body: ○ Is missing a required property ○ Includes a property that is invalid for the resource ○ Has a property with an invalid value ○ Contains JSON that is not well-formed
401	Unauthorized	All	The S Series Node was unable to handle the request. If this happens repeatedly, contact your authorized service provider for help.

(Continued)

Code	Meaning	Methods	Description
403	Forbidden	All	The requested operation is not allowed. These are some, but not all, of the possible reasons: • The URL in the request is missing the port number (9090). • The request does not include an Authorization header. • The Authorization header specifies invalid credentials. • The user account identified by the Authorization header doesn’t have permission to perform the requested operation. • For a **PUT** or **POST** request, the request body includes a property that is valid for the resource but that cannot be modified by the requested operation. • For a **DELETE** request for a user account, you cannot delete the user account because it is the account you're using to make the request.
404	Not Found	All	The resource identified by the URL does not exist.
405	Method Not Allowed	PUT POST DELETE	The requested operation is not valid for the resource identified by the URL.
406	Not Acceptable	All	A request that normally returns a response body contains an Accept request header that specifies an Internet media type other than application/json.
409	Conflict	PUT	The S Series Node could not create the specified resource because the resource already exists.
414	Request URI Too Large	All	The portion of the URL following `mapi` is longer than 4,095 bytes.

(Continued)

Code	Meaning	Methods	Description
500	Internal Server Error	All	One of these happened: • A request that expects a request body contains a Content-Type request header that specifies an Internet media type other than application/json. • A request that expects a response body contains an Accept request header that specifies an Internet media type other than application/json. • An internal error occurred. If this happens repeatedly, contact your authorized service provider for help.
503	Service Unavailable	All	The S Series Node is temporarily unable to handle the request. Possible reasons include: • The S Series Node is unavailable due to system overload. • The S Series Node is in the process of being upgraded. • The S Series Node is offline for maintenance purposes. Try the request again in a little while.
505	Version Not Supported	All	The request is malformed in such a way that the X-HCPS-API-VERSION request header cannot be correctly parsed.

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA
HitachiVantara.com | community.HitachiVantara.com

Contact Information
USA: 1-800-446-0744
Global: 1-858-547-4526
HitachiVantara.com/contact