HITACHI
Inspire the Next

Hitachi Content Platform S Series Node

HCP S Series Node API Reference

Hitachi Data Systems

MK-HCPS004-01

Contents

Preface xiii
Intended audience xiii
Product version xiii
Syntax notation xiv
Related document xiv
Getting help xiv
Comments xv

Chapter 1: Introduction to HCP S Series Nodes 1
About HCP S Series Nodes 2
HCP S10 Node hardware components 2
User accounts 3
Usernames 5
Passwords 6
User roles 7
Considerations for working with user accounts 8
Objects 9
Buckets 9
Bucket names 10
Bucket owners 10
Considerations for working with buckets 11
HCP S Series Node networks 11
Access network 12
Management network 15
Server interconnect network 18
Considerations for working with S Series Node networks 18
HCP S Series Node identification 19
HCP S Series Node licenses 20

HCP S Series Node access 20
HCP S Series Management Console configuration 20
HCP S Series management API configuration 21
HCP S Series data access protocol configuration 22
Allow and deny lists 22
SSL server certificates 24
HCP S Series Node security 24
DNS servers and time servers 25
HCP S Series Node event log 26
Syslog logging 27
HCP S Series Node internal logs 29
HCP S Series software and firmware maintenance 30
HCP S Series Node update files 31
Considerations for software, license, and firmware updates 31
HCP S10 Node hardware maintenance 33
Maintaining hard disk drives 33
Replacing the enclosure 34

Chapter 2: Introduction to the HCP S Series management API 37
What you can do with the management API 38
Who can use the management API 41
Resources and properties 41
Supported methods for the management API 42
Management API input and output format 43
Management API query parameters 43
prettyprint query parameter 44
Management API error response body 44
X-HCPS-API-VERSION request and response headers 45
HTTP Server response header with the management API 46
X-HCPS-Domain-Name response header 46
X-HCPS-Server-Module-Number response header 46
X-HCPS-ErrorMessage response header 46

Chapter 3: Management API access and authentication 47
URLs for S Series Node access through the management API 48
Considerations for resource URLs 50
Management API authentication 51

Chapter 4: Management API resources 53
Alerts resource 55
Beaconing resources 55
Bucket resources 56
Certificate resources 57
Console resource 58
DNS resource 58
Events resource 59
Hardware resource 59
Identification resource 60
Irreparables resources 60
License resource 61
Log resources 62
Maintenance resources 63
Management API resource 65
Metrics resources 65
Miscellaneous settings resource 66
Network resources 67
Power resources 67
Protocol resources 68
Security resource 69
Status resources 69
Syslog resources 70
Time resource 70
Update resources 71
User account resources 72
Versions resource 73
Managing resource lists 73
Management API count query parameter 74
Management API marker query parameter 75
Management API prefix query parameter 76
Management API owner query parameter 77

Chapter 5: Management API resource details 79
Resource property usage 80
/alerts 80
/alerts properties 81
/alerts query parameters 86

/alert example 87
/buckets 88
/buckets properties 89
/buckets example 90
/buckets/bucket-name 91
/buckets/bucket-name properties 92
/buckets/bucket-name example 93
/buckets/bucket-name/irreparables 93
/buckets/bucket-name/irreparables properties 94
/buckets/bucket-name/irreparables examples 96
/configuration/certificates/system 98
/configuration/certificates/system properties 99
/configuration/certificates/system example 100
/configuration/certificates/system/generate 101
/configuration/certificates/system/generate properties 102
/configuration/certificates/system/generate example 103
/configuration/console 104
/configuration/console properties 105
/configuration/console example 109
/configuration/dns 110
/configuration/dns properties 111
/configuration/dns example 112
/configuration/ident 113
/configuration/ident properties 113
/configuration/ident example 114
/configuration/mapi 115
/configuration/mapi properties 116
/configuration/mapi example 121
/configuration/networks/builtin 122
/configuration/networks/builtin property 122
/configuration/networks/builtin example 122
/configuration/networks/builtin/network-name 123
/configuration/networks/builtin/network-name properties 124
/configuration/networks/builtin/network-name example 128
/configuration/protocols 129
/configuration/protocols property 129
/configuration/protocols example 130
/configuration/protocols/hs3 130
/configuration/protocols/hs3 properties 131

/configuration/protocols/hs3 example ... 133
/configuration/security ... 134
/configuration/security properties ... 135
/configuration/security example ... 136
/configuration/syslog ... 137
/configuration/syslog properties ... 138
/configuration/syslog example ... 141
/configuration/syslog/test ... 142
/configuration/time ... 143
/configuration/time properties ... 143
/configuration/time example ... 144
/events ... 145
/events properties ... 146
/events query parameters ... 152
maxEvents query parameter ... 153
eventsAfter and eventsBefore query parameters ... 153
severity query parameter ... 154
major query parameter ... 155
scopes, scopeRefs, and scopeSubRefs query parameters ... 155
/events example ... 158
/hardware ... 160
/hardware properties ... 160
Hardware: enclosure high-level properties ... 161
Hardware: enclosure alarm properties ... 171
Hardware: enclosure current properties ... 176
Hardware: enclosure detail properties ... 179
Hardware: enclosure service properties ... 183
Hardware: enclosure fan properties ... 187
Hardware: enclosure power supply properties ... 191
Hardware: enclosure SAS connector properties ... 198
Hardware: enclosure SAS expander properties ... 201
Hardware: enclosure slot properties ... 204
Hardware: hard disk drive properties ... 210
Hardware: enclosure temperature properties ... 218
Hardware: enclosure voltage properties ... 224
Hardware: server module properties ... 228
Hardware: server module core hardware properties ... 232
Hardware: server module disk properties ... 235
Hardware: server module file system properties ... 238

Hardware: server module IPMI properties 241
Hardware: server module IPMI sensor properties 243
Hardware: server module mirror state property 244
Hardware: server module mirror set properties 245
Hardware: server module network interface properties 247
Hardware: server module bonded network interface properties 247
Hardware: server module Ethernet interface properties 250
Hardware: server module peer state property 253
Hardware: server module peer properties 254
/hardware example 255
/hardware/beacon/enclosure/enclosure-number 267
/hardware/beacon/enclosure/enclosure-number query parameters 267
/hardware/beacon/enclosure/enclosure-number example 268
/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id 268
/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id query parameters 269
/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id example 269
/hardware/beacon/server_module/server-module-number 270
/hardware/beacon/server_module/server-module-number query parameters 270
/hardware/beacon/server_module/server-module-number example 270
/hardware/maintenance 271
/hardware/maintenance request body property 271
/hardware/maintenance response body properties 272
/hardware/maintenance example 272
/hardware/maintenance/active 273
/hardware/maintenance/active property 274
/hardware/maintenance/active example 274
/hardware/maintenance/history 275
/hardware/maintenance/history property 276
/hardware/maintenance/history example 276
/hardware/maintenance/procedure-id 278
/hardware/maintenance/procedure-id properties 279
/hardware/maintenance/procedure-id example 279
/hardware/maintenance/procedure-id/cancel 280
/hardware/maintenance/procedure-id/cancel properties 280
/hardware/maintenance/procedure-id/cancel example 280
/hardware/maintenance/procedure-id/candidates 282

/hardware/maintenance/procedure-id/candidates property 283
/hardware/maintenance/procedure-id/candidates example 283
/hardware/maintenance/procedure-id/complete 284
/hardware/maintenance/procedure-id/complete properties 285
/hardware/maintenance/procedure-id/complete example 285
/hardware/maintenance/procedure-id/confirm 286
/hardware/maintenance/procedure-id/confirm request body properties 287
/hardware/maintenance/procedure-id/confirm response body properties 289
/hardware/maintenance/procedure-id/confirm example 289
/hardware/maintenance/procedure-id/perform 291
/hardware/maintenance/procedure-id/perform properties 291
/hardware/maintenance/procedure-id/perform example 291
/hardware/maintenance/procedure-id/select 292
/hardware/maintenance/procedure-id/select request body properties 293
/hardware/maintenance/procedure-id/select response body properties 295
/hardware/maintenance/procedure-id/select example 295
/hardware/maintenance/procedure-id/update 296
/hardware/maintenance/procedure-id/update request body property 297
/hardware/maintenance/procedure-id/update response body properties 297
/hardware/maintenance/procedure-id/update example 297
/hardware/maintenance/procedure-id/verify 299
/hardware/maintenance/procedure-id/verify properties 299
/hardware/maintenance/procedure-id/verify example 299
/hardware/power/node 301
/hardware/power/node query parameters 301
/hardware/power/node example 301
/hardware/power/server-module-number 302
/hardware/power/server-module-number query parameters 302
/hardware/power/server-module-number example 303
/metrics/buckets 303
/metrics/buckets properties 304
/metrics/buckets example 305
/metrics/gateways 306
/metrics/gateways properties 307
/metrics/gateways example 312
/metrics/protection 313
/metrics/protection property 314
/metrics/protection example 314
/metrics/system 315

/metrics/system properties 315
/metrics/system example 316
/system/irreparables 317
/system/irreparables properties 318
/system/irreparables examples 320
/system/license 322
/system/license properties 322
/system/license example 324
/system/logs/cancel 325
/system/logs/download 325
/system/logs/mark 326
/system/logs/mark query parameter 326
/system/logs/mark example 326
/system/logs/prepare 327
/system/logs/prepare query parameters 327
/system/logs/prepare example 328
/system/logs/status 329
/system/logs/status properties 330
/system/logs/status example 332
/system/misc/settings/network/management/monitor 333
/system/misc/settings/network/management/monitor property 333
/system/misc/settings/network/management/monitor query parameter 334
/system/misc/settings/network/management/monitor examples 334
/system/status/full 336
/system/status/full properties 336
/system/status/full example 338
/system/status/health 339
/system/status/health properties 339
/system/status/health example 340
/system/update/apply 341
/system/update/history 341
/system/update/history properties 342
/system/update/history example 343
/system/update/manifest 345
/system/update/manifest properties 345
/system/update/manifest example 346
/system/update/progress 347
/system/update/progress properties 347
/system/update/progress example 349

/system/update/restart 351
/system/update/status 351
/system/update/status property 351
/system/update/status example 353
/system/update/upload/license 354
/system/update/upload/license properties 354
/system/update/upload/license example 355
/system/update/upload/software 356
/system/update/upload/software properties 357
/system/update/upload/software example 358
/user_accounts 359
/user_accounts properties 360
/user_accounts examples 361
/user_accounts/username 363
/user_accounts/username properties 364
/user_accounts/username example 368
/user_accounts/username/access_key/generate 369
/user_accounts/username/access_key/generate properties 369
/user_accounts/username/access_key/generate example 370
/versions 370
/versions GET properties 371
/versions POST query parameter and properties 371
/versions examples 372

Chapter 6: Management API procedures 375
Downloading the internal logs 376
Performing a hardware maintenance procedure 378
Replacing a hard disk drive 379
Maintenance procedure properties 385
Maintenance procedure: target component list property 389
Maintenance procedure: target component properties 390
Maintenance procedure: enclosure or slot properties 402
Maintenance procedure: hard disk drive properties 403
Performing an update 408

Chapter 7: Management API HTTP status codes 413

Glossary 417

Index 421

Preface

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

Note: Throughout this book, the word *Unix* is used to represent all UNIX®-like operating systems (such as UNIX itself or Linux®), except where Linux is specifically required.

Intended audience

This book is intended for people who want to configure, monitor, and manage an S Series 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 the HTTP protocol.

Product version

This book applies to release 1.0.2 of the HCP S Series Node.

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: http://admin.hcp-ma.example.com:9090/user_accounts
italics	Replace with a value of the indicated type	

Related document

HCP S Series Node Help — This Help system contains information about configuring, managing, and maintaining an HCP S Series Node. The Help includes both information you need to effectively use the HCP S Series Management Console and instructions for physical S Series Node maintenance tasks that you manage from the Management Console. The Help also contains a complete reference for using the HCP S Series management API. Additionally, the Help includes release notes for the current release of the product and copyright and license information for third-party software distributed with or embedded in an S Series Node.

Getting help

Hitachi Data Systems Support Portal is the destination for technical support for your current or previously sold storage systems, midrange and enterprise servers, and combined solution offerings. The Hitachi Data Systems® customer support staff is available 24 hours a day, seven days a week. If you need technical support, log into the Hitachi Data Systems Support Portal for contact information: http://portal.hds.com

Hitachi Data Systems Community is a global online community for HDS customers, partners, independent software vendors, employees, and prospects. It is an open discussion among these groups about the HDS portfolio of products and services. It is the destination to get answers, discover insights, and make connections. The HDS Community

complements our Support Portal and support services by providing an area where you can get answers to noncritical issues and questions. **Join the conversation today!** Go to http://community.hds.com, register, and complete your profile.

Note: If you purchased your HCP S Series Node from a third party, please contact your authorized service provider.

Comments

Please send us your comments on this document:

HCPDocumentationFeedback@hds.com

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

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 Data Systems®. 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 includes information about both hardware and software.

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. The S10 model of the S Series Node consists of two cooperating server modules and multiple high-density disks in a single enclosure. The use of commodity hardware in S Series Nodes ensures that the costs of growth and repair remain low.

During normal operation, the two server modules actively share responsibility for all S Series Node functions. Because the server modules are equals, if one becomes unavailable, the other can still provide full S Series Node functionality.

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

S Series Nodes serve as storage tiering platforms for HCP systems. These systems use the S Series Node HS3 API, which is compatible with Amazon® S3™, to write, retrieve, and otherwise manage objects in an S Series Node. A single HCP system can seamlessly tier data across multiple S Series Nodes, thereby enabling scalability in both capacity and performance.

For administrative purposes, S Series Nodes provide a web-based Management Console and a RESTful 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 disks.

HCP S10 Node hardware components

The primary components of an S10 Node are:

- The **enclosure** that's the container for the other components.
- The two **power and cooling modules** that provide power and cooling for the enclosure and its components.
- The two **server modules** that run the HCP S Series software that manages the S10 Node, provides data access, and ensures data protection.

- The **one-foot CAT6 Ethernet cable** that connects the two server modules to each other, thereby creating an isolated network that enables the modules to communicate with each other for synchronization and resource sharing purposes.

- Twenty-eight or 56 **SATA hard disk drives** that are used to store the data written to the S10 Node.

- Four **SAS hard disk drives** on which the S10 Node stores the internal database that holds object metadata, definitions of user accounts and buckets, and various configuration settings.

All these components are replaceable. The procedures for replacing the enclosure and hard disk drives must be started and finished either in the HCP S Series Management Console or by using the management API.

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 HCP S Series HS3 API

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 HCP S Series HS3 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 HS3 credentials for your user account.

Note: In release 1.0.2 of the S Series Node, only an HCP system can be a direct user of the HCP S Series HS3 API.

Normally, user account passwords expire after a configurable amount of time. However, security administrators can configure individual user accounts such that the password never expires automatically.

Security administrators can also modify individual user accounts such 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.

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

- A full name. The full name can be used to identify the user for whom the account was created. This name must be from one through 256 characters long and can contain any valid UTF-8 characters, including white space.
- A description (optional). The description can be up to 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 is 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.

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

Usernames

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

Usernames:

- Must be three 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. So, 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 it. Users can change their account passwords at any time.

Passwords:

- Can be up to 256 characters long
- Must be at least as long as the configured minimum password length, which cannot be less than eight
- Can contain any valid UTF-8 characters
- Can include white space
- Are case sensitive
- Must include at least one character from each of these character sets:
  - Lowercase letters
  - Uppercase letters
  - Numbers
  - These special characters: ~`!@#$%^&*()_-+={[}]|:;"'<,>.?/\

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 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, and insert comments into and download the internal logs. With this role, you can also view the user account and bucket lists, create, modify, and delete buckets, and view the list of irreparable objects in those buckets. However, you cannot create, view, or otherwise manage objects in buckets.

  The administrator role does not grant permission to configure user accounts.

- **Monitor** — Grants permission to use the HCP S Series Management Console and management API to view S Series Node configuration and status and insert comments into the internal logs. With this role, you can also view the bucket list and view the list of irreparable objects in those buckets. However, you cannot create, view, or otherwise manage objects in buckets.

  The monitor role does not grant permission to view or configure user accounts.

- **Security** — Grants permission to use the HCP S Series Management Console and management API to view security events, create and manage user accounts, configure security settings, and insert comments into the internal logs.

Tip: Always have at least two user accounts with the security role. This 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, insert comments into and download the internal logs, and update the S Series Node software, firmware, and license. With this role, you can also view the bucket list and view the list of irreparable objects in those buckets. However, you cannot create, view, or otherwise manage objects in buckets.

  The service role does not grant permission to view or configure user accounts.

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

- **Data** — Grants permission to use the HCP S Series HS3 API to create and manage buckets and store and manage objects in buckets. With this role, you can also use the Management Console and management API to generate your access key and secret key.

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

Considerations for working with user accounts

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

These considerations apply to creating, modifying, and deleting 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.
- If you disable the user account you used to log into 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.
- 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, and users should keep their passwords confidential.

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 it.

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.

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 from three 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

If you have the administrator role, you can create, modify, and delete buckets. To perform these actions, you can use either the HCP S Series Management Console or the HCP S Series management API. If you have the data role, you can use the HCP S Series HS3 API to create and delete buckets.

These considerations apply to modifying and deleting buckets:

- You cannot change the name of an existing bucket.
- You should not change the owner of a bucket that is being used by HCP. Doing so prevents HCP from storing and managing objects in the bucket.
- You can delete a bucket only if it's empty (that is, it does not contain any objects).

HCP S Series Node networks

An S Series Node makes use of three networks:

- **The access network** is used for external client access to the S Series Node though the HCP S Series HS3 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 HS3 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 by the two S Series Node server modules to communicate with each other. The two server modules are the only devices on this isolated network.

Caution: Do not disconnect the purple server interconnect cable from either server module while the S Series Node is powered on. Doing so can result in data loss.

Access network

For the access network, each S Series Node server module has two bonded 10Gb Ethernet ports that can be configured as active-active (802.3ad) or active-backup. These ports connect the server modules to the customer network through one or two Ethernet switches. The recommended configurations are:

- One Ethernet switch, with both the S Series Node and the switch configured for active-active bonding. With this configuration, both access network ports on both server modules are connected to the same switch.

- Two Ethernet switches. With this configuration, one access network port on each server module is connected to one of the switches, and the other access network port on each server module is connected to the other switch.

For the appropriate configuration for your, consult your network administrator.

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 must 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.

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 one.

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

The S Series Node access network has these properties:

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

      By default, the access network has an IPv4 gateway address of 10.0.0.254.
    - An IPv4 subnet mask. By default, the access network has an IPv4 subnet mask of 255.255.255.0.
    - An IPv4 subnet. The S Series Node derives this subnet from the access network IPv4 gateway address and access network IPv4 subnet mask.

      By default, the access network has an IPv4 subnet of 10.0.0.0/24.
    - A 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.
    - A virtual IPv4 address for each server module. By default, the access network has virtual IPv4 addresses of 10.0.0.3 for server module 1 and 10.0.0.4 for server module 2.
- If the IP mode is IPv6:
    - A primary IPv6 gateway address. This is the address from which communications initiated by the S Series Node are sent over the network when the access network using the primary IPv6 gateway is the selected network for the particular type of communication.

  - A primary IPv6 prefix length.
  - A primary IPv6 subnet. The S Series Node derives this subnet from the primary access network IPv6 gateway address and primary access network IPv6 prefix length.
  - A primary physical IPv6 address for each server module.
  - A primary virtual IPv6 address for each server module.
  - Optionally, a secondary IPv6 gateway address, a secondary IPv6 prefix length, a secondary IPv6 subnet, a secondary physical IPv6 address for each server module, and a secondary virtual IPv6 address for each server module.

- Optionally, if the networking infrastructure supports virtual networking, a VLAN ID other than zero. Valid values for the VLAN ID are integers in the range zero through 4,094. If the networking infrastructure doesn't support virtual networking, the VLAN ID must be zero.

  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 zero.

- A 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, if supported by the networking infrastructure, 9,000. The larger MTU reduces overhead and increases network throughput.

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

- A 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 and then adjusts its settings to provide the highest possible data rate.

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

Note: In the zone definition for the S Series Node in the DNS, use the virtual IP addresses of the server modules.

Management network

For the management network, each S Series Node server module has one 1Gb Ethernet port. These ports connect the server modules to the customer network using either of these configurations:

- Two Ethernet switches, with the management port on each server module connected 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.
- One Ethernet switch, with the management ports on both server modules connected 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.

Use of the management network is not required. If you don't plan to use this network, you have the option of not physically connecting it to the customer network.

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 one.

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

The S Series Node 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:
  - An IPv4 gateway address. This is the address from which communications initiated by the S Series Node are sent over the network when the management network using the IPv4 gateway is the selected network for the particular type of communication.

    By default, the management network has an IPv4 gateway address of 10.2.2.254.
  - An IPv4 subnet mask. By default, the management network has an IPv4 subnet mask of 255.255.255.0.
  - An IPv4 subnet. The S Series Node derives the this subnet from the management network IPv4 gateway address and management network IPv4 subnet mask.

    By default, the management network has an IPv4 subnet of 10.2.2.0/24.
  - An 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 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:
  - A primary IPv6 gateway address. This is the address from which communications initiated by the S Series Node are sent over the network when the access network using the primary IPv6 gateway is the selected network for the particular type of communication.
  - A primary IPv6 prefix length.
  - A primary IPv6 subnet. The S Series Node derives this subnet from the primary management network IPv6 gateway address and primary management network IPv6 prefix length.
  - A primary IPv6 address for each server module.

- Optionally, a secondary IPv6 gateway address, a secondary IPv6 prefix length, a secondary IPv6 subnet, and a secondary IPv6 address for each server module.

Note: Do not use 10 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.

- Optionally, if the networking infrastructure supports virtual networking, a VLAN ID other than zero. Valid values for the VLAN ID are integers in the range zero through 4,094. If the networking infrastructure doesn't support virtual networking, the VLAN ID must be zero.

  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 zero.

- A 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, if supported by the networking infrastructure, 9,000. The larger MTU reduces overhead and increases network throughput.

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

- A combined speed and duplex setting. By default, the management 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 and then adjusts its settings to provide the highest possible data rate.

- Whether monitoring of the management network is enabled or disabled. If you don't make the physical connections for the management network, 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

For the server interconnect network, each S Series Node server module has one 1Gb Ethernet port. These ports connect the server modules 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.

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 corporate networking environment.

Note: You should change the subnet for the server interconnect network only if a conflict exists.

Caution: Do not disconnect the purple server interconnect cable from either server module while the S Series Node is powered on. Doing so can result in data loss.

Considerations for working with S Series Node networks

If you have the administrator or service role, you can modify S Series Node networks. To do this, you can use either the HCP S Series Management Console or the HCP S Series management API.

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 netmask 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 the server interconnect 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 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

HCP S Series Node identification

Each S Series Node can be 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 one through 63 characters long

- Can be up to 127 characters long, including the periods between segments

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

You can use the HCP S Series Management Console or management API to change the domain name for an S Series Node. If you do this, be sure to also change the domain name in the DNS.

Serial number
The serial number for an S Series Node uniquely identifies the S10 Node. You can find the serial number on a label that's attached to the front right corner of the top of the enclosure. You can also view the serial number in the HCP S Series Management Console or by using the S Series Node management API.

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

HCP S Series Node licenses

Every HCP S Series Node must be licensed. Each license is for an individual S Series Node. A license specifies the total storage capacity that can be installed in the S Series Node without violating the license agreement.

A license can have an expiration date or can be valid for an indefinite period of time. If the license for an S Series Node expires, using that S Series Node is a violation of the license agreement.

HCP S Series Node access

An S Series Node has three interfaces that provide access to it:

- The web-based **HCP S Series Management Console** supports both management functions and data access.
- The RESTful **HCP S Series management API** supports only management functions.
- The RESTful **HS3 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.

HCP S Series 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.

For each of these networks individually, you can enable HTTPS alone or HTTPS and HTTP together for access to the Management Console. By default for both networks, only HTTPS is enabled for Console access.

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."

HCP S Series 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 S10 Node.

For the access and management networks individually, you can enable HTTPS alone or HTTPS and HTTP together for access to the S Series Node through the management API. By default for both networks, only HTTPS is enabled for management API access.

Although S Series Nodes can support HTTP without SSL security, for security reasons, client requests for access 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.

HCP S Series data access protocol configuration

Note: In release 1.0.2 of the S Series Node, the only supported data access protocol is HS3.

You can enable or disable use of the HCP S Series HS3 API. If you disable use of this API, clients cannot read, write, modify, or delete data stored on the S Series Node.

You can enable HTTPS alone or HTTPS and HTTP together for access to the S Series Node through the HS3 API. By default, both are enabled.

By default, clients can use the HS3 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 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.

List entries	Allow requests when same IP is used in both lists	
	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 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 and deny list handling for that interface such 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 HS3 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.

HCP S Series Node security

As a security administrator, you can control various aspects 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.
- **SSH login by authorized service providers** — Enabling this service facilitates troubleshooting when you request support.

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 eight through 256. The default is eight.
- 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.

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 prevent the from happening, turn off the 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 three through 999. The default is ten.

  This limit applies both to attempts to log into 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 reenabled 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 five through 720. The default is ten.

DNS servers and time servers

You can choose to tell the S Series Node how to access one or more DNS servers. An S Series Node always needs to know how to access at least one external time server.

DNS servers
Optionally, you can make DNS servers known to an S Series Node. You can specify up to three DNS servers. You identify each one 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 your network selection must match the IP mode of the DNS server IP addresses.

Time servers
S Series Nodes use external time servers to set and maintain their internal clock times. Regardless of the time servers used, S Series Node time is always expressed in UTC.

You can specify up to three external times 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.

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 your network selection 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 itself.

HCP S Series Node event log

An S Series Node maintains a log to which it writes 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 Management Console provides several views of the event log:

- In the **Monitor** section of the Console, you can choose to see only messages about events related to a particular aspect of the S Series

Node (for example, security events or events related to hard disk drives).

- On the **Dashboard** page, the event list includes only messages about major events (for example, user account creations but not user logins to the Console).
- On the **Hardware ▶ Overview** page, the event list includes only messages about events related to the S Series Node hardware components.
- On each page for an individual hardware component, the event list includes only messages about events related to that component.
- On the **Hardware ▶ Maintenance** page, the event list includes only messages about events related to the maintenance procedures.
- On the **System ▶ Update** page, the event list includes only messages about events related to software, firmware, and license update procedures.
- On the **Configuration ▶ Certificates** page, the event list includes only messages about events related to SSL server certificates.

The event times shown with the log messages are in UTC.

To see more information about a message in the event log, including what action to take, if any, in response to the event, click on the message.

Tip: The text for alerts displayed in the Management Console is the same as the detailed form of messages in the event log. For more information about an alert, find the corresponding message in a view of the event log and click on the message.

Syslog logging

You can have the S Series Node send event log messages to one or more specified syslog servers as the messages are written to the log. When you do this, you can use tools in your syslog environment to perform functions such as sorting the messages, querying for certain events, or forwarding error messages to a mobile device.

By default, the S Series Node sends messages about all events except security events. Security event messages report actions that require the security role (such as the creation of user accounts) and attempts to log into the HCP S Series Management Console with an invalid username or to use the HCP S Series management API with an invalid username. With the Management Console and management API, only users with the security role can see these messages. However, you can choose to have security event messages sent to the syslog servers along with other event messages.

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.

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 to direct the event messages. This selection applies to all the syslog servers you specify.

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.

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 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.

After configuring syslog logging, 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 minmum severity level must be set to **NOTICE**, which allows all messages to be sent.

Note: If the access and management networks have different IP modes and two or more syslog servers are configured to receive event log messages, where at least one syslog server has an IPv4 address and one has an IPv6 address, the S Series Node sends messages to the syslog servers over both the access and management network.

HCP S Series Node internal logs

In addition to the event log, which is displayed in the HCP S Series Management Console and available through the HCP S Series management API, an S Series Node maintains internal logs. These logs record the status and activity of various components of the software running on the S Series Node. If a problem occurs with the S Series Node, the internal logs can help support personnel diagnose and resolve it.

At any time, you can insert a comment into the S Series Node internal logs. You can use this capability, for example, to note unusual events that occur in the S Series Node. This can later help support personnel understand the symptoms that indicate a possible problem. It can also help them determine when a problem started.

To help with troubleshooting, 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 them for a shorter time period if not enough 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 problems with the S Series Node.

Downloading the S Series Node internal logs is a two-part procedure. In the first part, the S Series Node prepares the logs for download. In the second part, the S Series Node packs the prepared logs into a .zip file and performs

the actual download. The amount of time the S Series Node takes to prepare the logs depends both on the length of time for which you request the logs and on the size of the log files.

Note: Downloading the internal logs puts a heavy load on the S Series Node. Do not take this action unless explicitly told to do so by support personnel.

HCP S Series software and firmware maintenance

When a new release of the HCP S Series software becomes available, you can upgrade the currently installed version of the HCP S Series software to that release. Software upgrades, which can also include an upgrade of the HCP S Series OS, are performed while the S Series Node is running. The S Series Node remains fully functional during an upgrade.

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

When capacity is added to an S Series Node or when the product license is extended, you need to upload a new license to the S Series Node. License uploads also occur while the S Series Node is running, with no loss of functionality during the process.

You use the same procedure for upgrading the software, applying a hotfix, and uploading a new license. The first step of this procedure is to upload an update file. The second step is to apply the uploaded update. You can perform this procedure either in the HCP S Series Management Console or by using the HCP S Series management API.

Occasionally, you may need to update the S Series Node component firmware. The first two steps for updating component firmware are the same as the steps for the other update procedures and are performed while the S Series Node is running.

The third step of the firmware update procedure must be performed by an authorized service provider. During this step, the S Series Node is rebooted one or more times. Therefore, while this step is in progress, the S Series Node is unavailable for both management purposes and data access.

HCP S Series Node update files

You make updates to the HCP S Series software, firmware, or license by uploading and applying the contents of a single update file. This can be a software upgrade file, a hotfix file, or a license file.

A **software upgrade file** contains the files necessary for upgrading the HCP S Series software and, if applicable, the HCP S Series OS. This file can also contain the tools necessary for performing the third step of a firmware update.

Software upgrade files are named `HS437_release-number.bin` (for example, `HS437_1.1.0.16.bin`).

A **hotfix file** contains one or both of these:

- The files necessary for applying the hotfix
- The tools necessary for updating component firmware

Hotfix files are named `HCPS-release-number_HFhotfix-number.bin` (for example, `HCPS-1.0.0.12_HF0001.bin`).

A **license file** contains an S Series Node license. Each license is specific to a particular S Series Node and cannot be applied to any other S Series Node.

License files are named `HCPSLic_SNserial-number-digits_Qquote-number_Clicensed-capacityTB_expiration-date.plk` (for example, `HCPSLic_SN12345_Q9876543_C112TB_03-17-2016.plk`). New license files for an S Series Node are sent to the customer site as needed.

Considerations for software, license, and firmware updates

These considerations apply to maintaining the HCP S Series software and firmware:

- Before you can start the procedure to upgrade the HCP S Series software, apply a hotfix, upload a new license, or update component firmware, both S Series Node server modules must be running and healthy.
- When you upload an update file, the file overwrites any previously uploaded update file.

- After uploading an update file, you cannot apply the update while the internal logs are being downloaded or a maintenance procedure is in progress.

- Before the HCP S Series software can be updated (with either a software upgrade or a hotfix), the firmware must be up to date as of the last time the HCP S Series OS was installed. If the firmware is not up to date, the software update fails. If this happens, update the firmware and then apply the software update again.

- While the software is being updated, you can make changes to the S Series Node configuration. However, most configuration changes don't take effect until the software update is complete.

- Software updates occur on one server module at a time. While the software is being updated on one server module, all S Series Node processing occurs on the other server module.

- When a software update finishes on the first server module, that server module is automatically rebooted. When the reboot is complete, the update 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 software update 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 the software on a server module is being updated, you cannot access that module by physical IP address.

- If you accessed the HCP S Series Management Console by using the physical IP address of the second server module while the software on the first server module was being updated, when failover occurs, you lose your connection to the S Series Node. At that point, you need to log in again, this time using the S Series Node domain name, a viritual IP address, or the physical IP address of the first server module to access the Management Console.

- If an error occurs during the apply step of an update, you can try restarting the update. If an error occurs again, do not try to restart the update a second time. Instead, contact your authorized service provider for help.

- Depending on the component firmware being updated, a firmware update can take from approximately ten minutes to approximately three hours. You should schedule this step for a time when the load on the S Series Node is light.

HCP S10 Node hardware maintenance

For certain S10 Node hardware maintenance procedures, you start the procedure either in the HCP S Series Management Console or by using the HCP S Series management API. These procedures are:

- Adding hard disk drives to an S Series Node
- Removing hard disk drives from an S Series Node
- Replacing hard disk drives in an S Series Node
- Replacing an S Series Node enclosure

Additionally, you use either the Management Console or management API to:

- Powering off an S Series Node
- Shut down, restart, or power an individual server module
- Turn beaconing on or off for the S Series Node enclosure, server modules, and power and cooling modules

To add, remove, or replace hard disk drives, you need to open the enclosure cover. When you open the cover, an alarm sounds intermittently. If the cover remains open for more than five minutes, the alarm becomes continuous. The alarm stops when the cover is closed.

Important: Do not perform multiple S10 Node maintenance procedures at the same time (for example, replacing the enclosure while adding hard disk drives). Doing so can have unpredictable results.

Maintaining hard disk drives

If you have the service role, you can add, remove, and replace hard disk drives in an HCP S10 Node. You can perform each operation on one or more drives at a time. However, you cannot combine different operations in a

single procedure.

For each operation you want to perform, you need to both start and end the procedure in the HCP S Series Management Console or by using the HCP S Series management API. If you don't start and end the procedure in this way, the S10 Node does not properly recognize that the operation has occurred. For example, if you add a drive without going through the Console or management API procedure, the S10 Node won't use that drive.

After you start a procedure, you can cancel it at any time. After you select the target slots for a procedure, if you don't end the procedure within four hours, the S10 Node automatically cancels it.

While a procedure is in progress, it's listed in the **Maintenance Procedure in Progress** section on the Management Console **Hardware Maintenance** page. When a procedure ends or is canceled, it moves to the **Maintenance History** list on that page. To view details about a listed procedure, click on the gear icon () for it.

Similarly, while a procedure is in progress, you can use the management API to retrieve information about it. You can also use management API to retrieve the maintenance procedure history list.

To add, remove, or replace hard disk drives, you need to open the enclosure cover. When you open the cover, an alarm sounds intermittently. If the cover remains open for more than five minutes, the alarm becomes continuous. The alarm stops when the cover is closed.

Replacing the enclosure

If the enclosure for an S10 Node is damaged or not functioning properly and you have the service role, you can replace the enclosure. You need to both start and end the replacement procedure in the HCP S Series Management Console or by using the HCP S Series management API.

If you don't start and end the enclosure replacement procedure in the Management Console or by using the management API, when you power the S10 Node back on, it will treat all hard disk drives, including the database drives, as failed. If this happens, contact your authorized service provider for help.

After you start an enclosure replacement procedure, you can cancel it at any time. After you select the enclosure for the procedure, if you don't end the procedure within four hours, the S10 Node automatically cancels it.

Note: If an enclosure replacement procedure is canceled after the enclosure is physically replaced, the procedure ends unsuccessfully. To recover from this situation, you need to contact your authorized service provider.

When replacing the enclosure for an S10 Node, you need to replace the righthand cover of the new enclosure with the righthand cover of the old enclosure. This is because the serial number for the S10 Node is on a label attached to that cover.

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 hard disk 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.

Hitachi Data Systems provides an SDK for the management API. You can download this SDK from https://sourceforge.net/projects/rhinomapisdk. This web site also gives you access to code samples and documentation on using the SDK.

This chapter:

- Describes what you can do with the management API
- Explains who can use the management API
- Contains an introduction to resources and properties
- Lists the HTTP methods supported by the management API
- Describes the input and output format supported by the management API
- Discusses query parameters, which are used with resource requests

- Describes the management API error response body
- Describes selected management API request and response headers

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 HCP S Series HS3 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 a report on 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 front-end, management, and back-end 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
- **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 licenses** — You can retrieve information about the current license for the S Series Node.
- **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 HCP S Series HS3 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:
  - Upgrade the HCP S Series software, apply a hotfix, upload a new S Series Node license, and set up a firmware update
  - Retrieve the status of an in-progress update operation
  - Retrieve a list of previous update operations
- **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, power and cooling module, or server 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 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 1.0.1 version of the management API to process the request:

```
curl -k -H "X-HCPS-API-VERSION: 1.0.1"
  -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/1.0.2.6
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 181
```

HTTP Server response header with the management API

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/1.0.2.6
```

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 on S Series Node domain names, see "HCP S Series Node identification" on page 19

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: 2
```

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 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 the 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 on 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 on management API configuration, see "HCP S Series management API configuration" on page 21. For information about SSL server certificates, see "SSL server certificates" on page 24.

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 on 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.

4

Management API resources

The main types of HCP S Series management API resources are:

- Alerts
- Beaconing
- Buckets
- Certificates
- Console
- DNS
- Events
- Hardware
- Identification
- Irreparables
- License
- Log
- Maintenance
- Management API
- Metrics
- Miscellaneous settings
- Networks

- Power
- Protocols
- Security
- Status
- Syslog
- Time
- Update
- User accounts
- Versions

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

A resource identifier is the portion of a resource URL that follows the mapi interface identifier. For more information on resource URLs, see "URLs for S Series Node access through the management API" on page 48.

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

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 on alerts, see "HCP S Series Node event log" on page 26.

Beaconing resources

The S10 Node enclosure, power and cooling modules, and server modules have LEDs that you can cause to blink. A blinking LED serves as a beacon so that the applicable component can be easily identified in your data center.

Beaconing resources let you turn beaconing on and off for enclosures, power and cooling modules, and server 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 on the query parameters used for turning beaconing on and off, see "/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id query parameters" on page 269.

(Continued)

Method	Use	Roles	Notes
/hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-module-id*			
POST	Turn beaconing on or off for a power and cooling module	Administrator Service	For information on the query parameters used for turning beaconing on and off, see "/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id query parameters" on page 269 "/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id query parameters" on page 269.
/hardware/beacon/server_module/*server-module-number*			
POST	Turn beaconing on or off for a server module	Administrator Service	For information on the query parameters used for turning beaconing on and off, see "/hardware/beacon/server_module/server-module-number query parameters" on page 270.

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 on the query parameters used to limit the bucket list, see "Managing resource lists" on page 73.

(Continued)

Method	Use	Roles	Notes
/buckets/*bucket-name*			
GET	Retrieve information about an existing bucket	Administrator Monitor	
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 on buckets, see "Buckets" on page 9 and "Considerations for working with buckets" on page 11.

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 on SSL server certificates, see "SSL server certificates" on page 24.

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 on Management Console configuration, see "HCP S Series Management Console configuration" on page 20.

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 on DNS server configuration, see "DNS servers and time servers" on page 25.

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. These messages are not returned when the request is made by a user without the security role. For information on the required and optional query parameters used to limit the event message list, see "/events query parameters" on page 152.

For more information on events, see "HCP S Series Node event log" on page 26.

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 more information on S10 Node hardware, see "HCP S10 Node hardware components" on page 2.

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 on S Series Node identification, see "HCP S Series Node identification" on page 19.

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 on the query parameters used to limit the irreparable object list, see "Managing resource lists" on page 73.
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.

(Continued)

Method	Use	Roles	Notes
/system/irreparables			
GET	Retrieve a list of irreparable objects stored on the S Series Node	Administrator Monitor Service	For information on the query parameters used to limit the irreparable object list, see "Managing resource lists" on page 73.
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 information about the current S Series Node license. The table below provides information about this resource.

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

For more information on S Series Node licenses, see "HCP S Series Node licenses" on page 20. For information on the resources used to upload a new S Series Node license, see "Update resources" on page 71.

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	
/system/logs/mark			
POST	Insert a comment into the internal logs	Administrator Monitor Security Service	For information on the query parameter used to specify the message to be inserted, see "/system/logs/mark query parameter" on page 326.
/system/logs/prepare			
POST	Prepare the internal logs for download	Administrator Service	This **POST** request starts a log download operation. For information on the query parameters used to specify the time period for the logs to be downloaded, see "/system/logs/prepare query parameters" on page 327.
/system/logs/status			
GET	Retrieve the status of the current log download operation	Administrator Service	

For more information on the S Series Node internal logs, see "HCP S Series Node internal logs" on page 29. For instructions on using the management API to download the internal logs, see "Downloading the internal logs" on page 376.

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.

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	
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 on hardware maintenance procedures, see "HCP S10 Node hardware maintenance" on page 33"Performing a hardware maintenance procedure" on page 378.

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 on management API configuration, see "HCP S Series management API configuration" on page 21.

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 on the query parameters used to limit the bucket list, see "Managing resource lists" on page 73.
metrics/gateways			
GET	Retrieve statistics about data access protocol (HS3) usage	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 on the query parameter used to enable and disable management network monitoring, see "/system/misc/settings/network/management/monitor query parameter" on page 334.

For more information on management network monitoring, see "Management network" on page 15.

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/*network-name*			
GET	Retrieve information about a network	Administrator Monitor Service	
POST	Modify a network	Administrator Service	

For more information about networks, see "HCP S Series Node networks" on page 11.

Power resources

Power resources let you power on and off and restart the S10 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 on the query parameters used to specify a power option, see "/hardware/power/node query parameters" on page 301"/hardware/power/node query parameters" on page 301.

(Continued)

Method	Use	Roles	Notes
/hardware/power/*server-module-number*			
POST	Power on or off or restart a single server module	Administrator Service	For information on the query parameters used to specify power options, see "/hardware/power/server-module-number query parameters" on page 302.

Protocol resources

Protocol resources let you retrieve a list of supported data access protocols and retrieve and modify the configuration of the HS3 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 HS3 API configuration	Administrator Monitor Service	
POST	Modify the HS3 API configuration	Administrator Service	

For more information on data access protocols, see "HCP S Series data access protocol configuration" on page 22.

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 on S Series Node security settings, see "HCP S Series Node security" on page 24.

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 on syslog logging, see "Syslog logging" on page 27.

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 on time settings, see "DNS servers and time servers" on page 25.

Update resources

Update resources let you update the HCP S Series software, firmware, and license. The table below provides information about these resources.

Method	Use	Roles	Notes
/system/update/update/apply			
POST	Apply an update	Service	
/system/update/history			
GET	Retrieve a list of all update procedures that have been performed on the S Series Node since the HCP S Series software was last installed	Administrator Service	
/system/update/manifest			
GET	Retrieve information about the currently uploaded update file	Service	
/system/update/progress			
GET	Retrieve information about the progress of a software upgrade or hotfix application	Service	
/system/update/restart			
POST	Restart a failed update operation	Service	
/system/update/status			
GET	Retrieve information about the status of an update operation on the S Series Node	Service	
/system/update/upload/license			
PUT	Upload a license file	Service	
/system/update/upload/software			
PUT	Upload and unpack a software upgrade file or hotfix file	Service	

For more information on software, firmware, and license updates, see "HCP S Series software and firmware maintenance" on page 30. For instructions on using the management API to perform updates, see "Performing an update" on page 408.

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	
GET	Retrieve a list of existing user accounts	Administrator Security	The user accounts are listed in alphabetical order by username. For information on 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	

(Continued)

Method	Use	Roles	Notes
/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 on user accounts, see "User accounts" on page 3 and "Considerations for working with user accounts" on page 8.

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 on the query parameter used to check management API version support, see "/versions POST query parameter and properties" on page 371.

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 and irreparable objects: **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 on those parameters, see "/alerts query parameters" on page 86 and "/events query parameters" on page 152.

For more information on query parameters, see "Management API query parameters" on page 43.

Management API 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, irreparables, and bucket metrics.

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

```
curl -k -H "X-HCPS-API-VERSION: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://admin.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 no value.

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"
  ]
}
```

Management API 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). 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, and irreparable objects.

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, you specify the case-sensitive name of the last resource in the previously returned list.
- In a request for irreparable objects, you 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 either 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, or irreparable objects 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.

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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://admin.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"
  ]
}
```

Management API 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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://admin.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"
  ]
}
```

Management API 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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://admin.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 on resource identifiers, see Chapter 4: "Management API resources" on page 53.

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 on resource properties, see "Resources and properties" on page 41.

/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 on the /alerts resource, see "Alerts resource" on page 55.

/alerts properties

The table below describes the properties in /alerts response bodies. For information on the query parameters mentioned in the table, see "/alerts query parameters" on page 86.

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.	

(Continued)

Property name	Data type	Description	Notes
message	String	Specifies the full text of the alert.	
pcode	String	Specifies the HDS 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** — Hard disk drives • **ENCLOSURE** — Enclosures • **FS** — Storage usage • **MAINT** — Maintenance procedures • **SECURITY** — Configuration that requires the security role; failed logins	

(Continued)

Property name	Data type	Description	Notes
		• **SERVER** — Server modules • **SYSTEM** — Configuration that does not require the security role; successful logins; system-initiated events • **UPGRADE** — Software and license updates	
scopeRef	Integer	For a scope of DRIVE, specifies the number of the enclosure that contains the hard disk 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 hard disk 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 on the possible values of this property for the ENCLOSURE and SERVER scopes, see "scopes, scopeRefs, and scopeSubRefs query parameters" on page 155.	
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 Hi-Track® Monitor should open a ticket for the condition to which the alert applies (assuming Hi-Track Monitor is configured to monitor the S Series Node). Hi-Track Monitor is an HDS 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 on these parameters, see "scopes, scopeRefs, and scopeSubRefs query parameters" on page 155.

For more information on query parameters, see "Management API query parameters" on page 43.

/alert 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the query parameters used to limit the bucket list returned by a **GET** request, see "Managing resource lists" on page 73.

For more information on the /buckets resource, see "Bucket resources" on page 56.

/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 "/user_accounts/username properties" on page 364

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 "Management API count query parameter" on page 74.	
isTruncated	Boolean	Specifies whether the returned list of buckets is complete. Possible values are: • **true** — The bucket list is complete. • **false** — The bucket list is incomplete. For more information, see "Management API count query parameter" on page 74.	

(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 "Management API marker query parameter" on page 75.	
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 "Management API owner query parameter" on page 77.	
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 "Management API prefix query parameter" on page 76.	

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /buckets/*bucket-name* resource, see "Bucket resources" on page 56.

/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 10.	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: 2015-01-20 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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": "2015-01-26 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 on the query parameters used to limit the list of irreparable objects returned by a **GET** request, see "Managing resource lists" on page 73.

For more information on the /buckets/*bucket-name*/irreparables resource, see "Irreparables resources" on page 60.

/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 "Management API count query parameter" on page 74.	
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 complete. • **false** — The irreparable object list is incomplete. For more information, see "Management API count query parameter" on page 74.	
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 "Management API marker query parameter" on page 75.	
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: 2015-01-20 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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":"2015-02-08 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/certificates/system resource, see "Certificate resources" on page 57.

/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 Jan 28 14:51:57 UTC 2015	
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: Tue Jan 28 14:51:57 UTC 2020	
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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 Jan 28 14:51:57 UTC 2015",
  "expires": "Tue Jan 28 14:51:57 UTC 2020"
}
```

/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 on the /configuration/certificates/system/generate resource, see "Certificate resources" on page 57.

/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 Jan 28 14:51:57 UTC 2015	
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: Tue Jan 28 14:51:57 UTC 2020	
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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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": "Thu Jan 29 08:10:07 UTC 2015",
  "expires": "Wed Jan 29 08:10:07 UTC 2020"
}
```

/configuration/console

With the /configuration/console resource:

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

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

For more information on the /configuration/console resource, see "Console resource" on page 58.

/configuration/console properties

The table below describes the properties in /configuration/console 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 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, managementNetworkHttpEnabled, and managementNetworkHttpsEnabled properties. Support for Management Console URLs that use 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, in the URL.

(Continued)

Property name	Data type	Description	Notes
accessNetworkHttpsEnabled	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, managementNetworkHttpEnabled, and managementNetworkHttpsEnabled properties. Either this property or the managementNetworkHttpsEnabled 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 on allow and deny list handling, see "Allow and deny lists" on page 22.	

(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 managementNetworkHttpsEnabled 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 managementNetworkHttpsEnabled properties. Support for Management Console URLs that use 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, in the URL.

(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 managementNetworkHttpEnabled 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/dns resource, see "DNS resource" on page 58.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/ident resource, see "Identification resource" on page 60.

/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. The only possible value is **S10**.	This property is not valid on a **POST** request.
serialNumber	String	Specifies the five-digit 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 121
```

Response body

```
{
  "serialNumber": "12345",
  "domainName": "s-node-1.example.com",
  "softwareVersion": "1.0.2.6",
  "model": "S10"
}
```

/configuration/mapi

With the /configuration/mapi resource:

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

For more information on the /configuration/console resource, see "Management API resource" on page 65.

/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, managementNetworkHttpEnabled, and managementNetworkHttpsEnabled properties. Although S Series Nodes can support resource URLs that use HTTP without SSL security, for security reasons, client requests for access through the management API should always use HTTPS, not HTTP, in the URL.

(Continued)

Property name	Data type	Description	Notes
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, managementNetworkHttpEnabled, and managementNetworkHttpsEnabled properties. Either this property or the managementNetworkHttpsEnabled 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 on allow and deny list handling, see "Allow and deny lists" on page 22.	

(Continued)

Property name	Data type	Description	Notes
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.
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.

(Continued)

Property name	Data type	Description	Notes
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 managementNetworkHttpsEnabled 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 managementNetworkHttpsEnabled properties. Although S Series Nodes can support resource URLs that use HTTP without SSL security, for security reasons, client requests for access through the management API should always use HTTPS, not HTTP, in the URL.
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 managementNetworkHttpEnabled 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/networks/builtin resource, see "Network resources" on page 67.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 77
```

Response body

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

/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 on the /configuration/networks/builtin/*network-name* resource, see "Network resources" on page 67.

/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	Specifies the bonding mode for the network. Valid values are **active-backup** and **802.3ad**.	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.
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.

(Continued)

Property name	Data type	Description	Notes
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.
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 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 server interconnect network. This property 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 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.
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 not returned by a **GET** request for the management or server interconnect network.

(Continued)

Property name	Data type	Description	Notes
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.	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.	
subnet2	String	Specifies the secondary IPv6 subnet for the network. 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 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 for the server interconnect network and is not returned by a **GET** request for that 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 on network properties, see:

- "Access network" on page 12
- "Management network" on page 15

- "Server interconnect network" on page 18

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/protocols resource, see "Protocol resources" on page 68.

/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 HS3, 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/protocols/hs3 resource, see "Protocol resources" on page 68.

/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
accessNetworkHttp Enabled	Boolean	Specifies whether HTTP without SSL security can be used for access to the S Series Node through the HS3 API. 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 accessNetwork HttpsEnabled property is also **true**. If the request body for a **POST** request includes this property, the request body must also include the accessNetworkHttpsEnabled property.
accessNetworkHttps Enabled	Boolean	Specifies whether HTTPS can be used for access to the S Series Node through the HS3 API. 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 HS3 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 on allow and deny list handling, see "Allow and deny lists" on page 22.	
allowList	Array	Specifies a comma-separated list of IP addresses that are allowed access to the S Series Node through the HS3 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 HS3 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 HCP S Series HS3 API.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/console resource, see "Security resource" on page 69.

/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 eight through 256. The default is eight.	

(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 into 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 default is **true**. 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/syslog resource, see "Syslog resources" on page 70.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /configuration/syslog/test resource, see "Syslog resources" on page 70.

/configuration/time

With the /configuration/time resource:

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

For more information on the /configuration/time resource, see "Time resource" on page 70.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 91
```

Response body

```
{
  "timeServers": [
    "10.0.201.65"
  ],
  "network": "[access]",
  "networkIndex": 1
}
```

/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 on the /events resource, see "Events resource" on page 59.

/events properties

The table below describes the properties in /events response bodies. For information on the query parameters mentioned in the table, see "/events query parameters" on page 152.

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.
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 complete. • **false** — The event message list is incomplete.	

(Continued)

Property name	Data type	Description	Notes
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**.	
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
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.	

(Continued)

Property name	Data type	Description	Notes
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.	
scope	String	Specifies the type of component or activity to which the event applies. Possible values are: • **CERT** — SSL server certificates • **DRIVE** — Hard disk 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 and license updates	

(Continued)

Property name	Data type	Description	Notes
scopeRef	Integer	For a scope of DRIVE, specifies the number of the enclosure that contains the hard disk 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 hard disk 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 on the possible values of this property for the ENCLOSURE and SERVER scopes, see "scopes, scopeRefs, and scopeSubRefs query parameters" on page 155.	
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: 2015-01-20 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 zero.	
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 on query parameters, see "Management API query parameters" on page 43.

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:

  2015-02-04 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 in 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 hard disk 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 and license updates

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 hard disk 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

(Continued)

ID	Subcomponent
1003	eth1
1004	eth2
1005	eth3
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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/events?maxEvents=2
  &eventsAfter=2015-02-05+03:00:00+UTC&scope=SERVER&scopeRefs=1
  &prettyprint"
```

Request headers

```
GET /mapi/configuration/mapi/events?maxEvents=2&eventsAfter=2015-02-
05+03:00:00+UTC&scope=SERVER&scopeRefs=1&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 1424
```

Response body


```
{
  "major": false,
  "severity": "NOTICE",
  "scopes": [
    "SERVER"
  ],
  "scopeRefs": [
    1
  ],
  "eventsAfter": "2015-02-05 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": "2015-02-05 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": "2015-02-05 15:17:53 UTC",
        "timestampSuffix": "1423149473890.890410000.27"
      }
    ]
}
```


/hardware

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

For more information on the /hardware resource, see "Hardware resource" on page 59. For information on S Series Node hardware, see "HCP S10 Node hardware components" on page 2.

/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 the next page.	

(Continued)

Property name	Data type	Description	Notes
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 228.	

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 171.	In an S10 Node, 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 176.	

(Continued)

Property name	Data type	Description	Notes
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 179.	The NDS-4600 enclosure used for S10 Nodes 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 183.	In an S10 Node, an enclosure service component is a server module.
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.	

(Continued)

Property name	Data type	Description	Notes
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 187.	
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
ledStates	Array	Specifies a comma-separated list of four integers, the first three of which represent the state of these LEDs on the front of the enclosure, in order: • Software OK (amber LED third down on the right) • Software fault (green LED second down on the right) • Software install (blue LED bottom right) The fourth integer is always **0**. For information on the meaning of the integers, see the next table.	
lockdownReason	String	If the enclosure is in lockdown mode, specifies the reason why the enclosure is in lockdown mode. Possible values are: • **LARKSPUR_ LOCKDOWN_REASON_ BAD_MIDPLANE_VPD** • **LARKSPUR_ LOCKDOWN_REASON_ BAD_OTHER_ LARKSPUR_VPD** • **LARKSPUR_ LOCKDOWN_REASON_ FAILED_TO_READ_ICID**	If the value of this property is not **UNDEFINED**, contact your authorized service provider for help.

(Continued)

Property name	Data type	Description	Notes
		• **LARKSPUR_ LOCKDOWN_REASON_ INCOMPATIBLE_ ENCLOSURE** • **LARKSPUR_ LOCKDOWN_REASON_ INCOMPATIBLE_T10_ VENDOR** • **LARKSPUR_ LOCKDOWN_REASON_ INCOMPATIBLE_ VENDOR_BYTES** • **LARKSPUR_ LOCKDOWN_REASON_ LOCAL_LARKSPUR_ VPD_READ_FAILED** • **LARKSPUR_ LOCKDOWN_REASON_ PAIRING_FAILURE** • **LARKSPUR_ LOCKDOWN_REASON_ PRIMARY_OVERRIDE** • **LARKSPUR_ LOCKDOWN_REASON_ SEC_1_OVERRIDE** • **LARKSPUR_ LOCKDOWN_REASON_ SEC_2_OVERRIDE** • **LARKSPUR_ LOCKDOWN_REASON_ SXP_CONFIG_READ_ FAILED** • **LARKSPUR_ LOCKDOWN_REASON_ UNDEFINED**	

(Continued)

Property name	Data type	Description	Notes
		• **LARKSPUR_ LOCKDOWN_REASON_ UNKNOWN_LOCAL_ LARKSPUR_PART_ NUMBER** If the enclosure is not in lockdown mode, the value of this property is **UNDEFINED**.	
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 HDS 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 191.	
product	String	Specifies the enclosure product name.	

(Continued)

Property name	Data type	Description	Notes
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 198.	
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 201	
sbbPowerOnState	String	Specifies the power-on state of the storage bridge bay. Possible values are: • **LOCK_DOWN** • **SHARED_MODE** • **SINGLE_MODE**	
serial	String	Specifies the enclosure serial number.	

(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 204.	
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** • **UNAVAILABLE** • **UNKNOWN**	
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 218.	
uptime	Long	Specifies the number of seconds that have elapsed since the S Series Node was last powered on.	
vendor	String	Specifies the name of the enclosure vendor.	

(Continued)

Property name	Data type	Description	Notes
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 224.	
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.	

The table below describes the integer values for the ledStates property.

Value	Description
1	Off
2	Solid
3	Blinking 2,000ms on, 2,000ms off
4	Blinking 2,000ms off, 2,000ms on
5	Blinking 500ms on, 1,500ms off

(Continued)

Value	Description
6	Blinking 1,000ms on, 1,000ms off
7	Blinking 1,000ms off, 1,000ms on
8	Blinking 1,500ms on, 500ms off
9	Blinking 750ms on, 750ms off
10	Blinking 250ms on, 750ms off
11	Blinking 500ms on, 500ms off
12	Blinking 500ms off, 500ms on
13	Blinking 750ms on, 250ms off
14	Blinking 125ms on, 375ms off
15	Blinking 250ms on, 250ms off
16	Blinking 250ms off, 250ms on
17	Blinking 375ms on, 125ms off

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 S10 Node, only the enclosure itself has an audible alarm.

Property name	Data type	Description	Notes
code	Integer	Specifies the status code for the component with the alarm. 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.	
error	Boolean	Specifies whether the component with the alarm has an error-level condition, as indicated by the code or urgency 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.	

(Continued)

Property name	Data type	Description	Notes
fail	Boolean	Specifies whether the component with the alarm is marked failed. Possible values are: • **true** — The component with the alarm is marked failed. • **false**— The component with the alarm is not marked failed.	
id	Integer	Specifies the identifier for the component with the alarm.	
ident	Boolean	Unused. The value of this property is always **false**.	
location	String	Specifies the name of the alarm.	
muted	Boolean	Specifies whether the alarm is muted. Possible values are: • **true** — The alarm is muted. • **false** — The alarm is not muted.	
remind	Boolean	Specifies whether the alarm is in the remind state. Possible values are: • **true** — The alarm is in the remind state. • **false** — The alarm is not in the remind state.	

(Continued)

Property name	Data type	Description	Notes
swap	Boolean	Specifies whether the component with the alarm has been removed and then reused or replaced. Possible values are: • **true** — The component has been removed and then reused or replaced. This is a transient value. • **false**— Either of these: ◦ The component has never been removed and then reused or replaced. ◦ The component was removed and then reused or replaced, and the S Series Node subsequently reset the value of this property to **false**.	
urgency	Array	Specifies a comma-separated list of the levels of urgency that currently apply to the alarm. The levels of urgency are: • **INFO** • **NON_CRITICAL** • **CRITICAL** • **UN_RECOVERABLE**	The tone associated with the highest level of urgency is the one that's used to sound the alarm.

(Continued)

Property name	Data type	Description	Notes
warning	Boolean	Specifies whether the component with the alarm has a warning-level condition, as indicated by the code or urgency 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 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 product part number of the enclosure midplane.	
baseboardSerial	String	Specifies the serial number of 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	Specifies whether the enclosure cover is open. Possible values are: • **true** — The cover is open. • **false** — The cover is closed.	
enclConfigRev	String	Specifies the current revision of the enclosure configuration.	

(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, as indicated by the code or fail property or by the coverOpen property if the cover was opened outside of an add, remove, or replace drives maintenance procedure. 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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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	Specifies the name of the enclosure. The value of this property is always **Enclosure**.	
predictedFailure	Boolean	Specifies whether the fault LED on the front of the enclosure is in the PRDFAIL state, as indicated by the faultLed property for the enclosure service component. Possible values are: • **true** — The fault LED is in the PRDFAIL state. • **false**— The fault LED is not in the PRDFAIL state.	
scpFwRev	String	Specifies the currently installed revision of the Signal Conditioning Processor firmware.	
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 service properties

The table below describes the properties used to provide information about an enclosure service component in /hardware resource response bodies.

In an S10 Node, an enclosure service component corresponds to a server 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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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	Unused. This property is always returned with no value.	
fruSerial	String	Unused. This property is always returned with no value.	
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.	
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.	

(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 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.	

(Continued)

Property name	Data type	Description	Notes
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 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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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**.	
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.	

(Continued)

Property name	Data type	Description	Notes
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 HDS part number to use when ordering a replacement power supply.	
product	String	Specifies the power supply product name to use when requesting service.	
rev	String	Specifies the power supply hardware revision to use when requesting service.	
serial	String	Specifies the power supply serial number to use when requesting service.	
supplierProduct	String	Specifies the original-vendor product name 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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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. For an S10 Node, the value of this property is always **1**, 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 ids of the slots associated with the SAS expander.	
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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.	
nvStoreRevNum	String	Unused. The value of this property is always **N/A**.	
sasAddr	String	Specifies the SAS address for the SAS expander.	

(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**.	
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 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**.	
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	Specifies that the slot LED is blinking with a 50% duty cycle in a 1,000ms period.	
deviceOff	Boolean	Unused. The value of this property is always **false**.	
doNotRemove	Boolean	Specifies that the slot LED is blinking with a 50% duty cycle in a 4,000ms period.	

(Continued)

Property name	Data type	Description	Notes
drive	Object	Specifies a set of properties that provide information about the drive in the slot. For descriptions of these properties, see "Hardware: hard disk drive properties" on page 210.	This property is not returned if the slot is empty.
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	Specifies that the slot LED is blinking with a 50% duty cycle in a 1,500ms period.	

(Continued)

Property name	Data type	Description	Notes
id	Integer	Specifies the component identifier for the slot within the enclosure. The value of this property is one less than the numeric value in the name of the slot. For example, the value of this property for the slot named SLOT 7 is **6**.	
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	Specifies that the slot LED is blinking with a 50% duty cycle in a 1,500ms period.	
inFailedArray	Boolean	Specifies that the slot LED is blinking with a 75% duty cycle in a 1,000ms period.	
location	String	Specifies the name of the slot, in this format: SLOT *n* *n* is the position of the slot in the enclosure. For example, the value of this property for the slot in the seventh position in the enclosure is **SLOT 7**. The value of *n* is always one greater than the value of the id property for the slot.	

(Continued)

Property name	Data type	Description	Notes
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.	
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	Specifies that the slot LED is blinking with a 50% duty cycle in a 1,000ms period.	
readyToInsert	Boolean	Specifies that the slot LED is blinking with a 50% duty cycle in a 1,000ms period.	
rebuildRemap	Boolean	Specifies that the slot LED is blinking with a 75% duty cycle in a 1,000ms period.	
rebuildRemapAbort	Boolean	Specifies that the slot LED is blinking with a 50% duty cycle in a 4,000ms period.	

(Continued)

Property name	Data type	Description	Notes
reservedDevice	Boolean	Specifies that the slot LED is blinking with a 50% duty cycle in a 1,000ms period.	
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**.	
sasExpIndex	Integer	If the slot contains a drive, specifies the component identifier of the SAS expander to which the slot is connected. If the slot is empty, the value of this property is **-1**.	
slotNumber	Integer	Specifies the slot number. For example, the value of this property for the slot in the seventh position in the enclosure is **7**.	
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.	

(Continued)

Property name	Data type	Description	Notes

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**.	

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: hard disk drive properties

The table below describes the properties used to provide detailed information about a hard disk drive in a slot in an enclosure in /hardware resource response bodies.

Property name	Data type	Description	Notes
ataVersion	String	For a SATA drive, specifies the drive ATA version. For a SAS drive, this property is returned with 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: 2015-03-04 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: 2015-03-04 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. • **MAINT_CANCEL** — The drive was a target component for a canceled maintenance procedure.	

(Continued)

Property name	Data type	Description	Notes
		• **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. • **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. • **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.	

(Continued)

Property name	Data type	Description	Notes
		• **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.	
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.	

(Continued)

Property name	Data type	Description	Notes
pCode	String	Specifies the HDS part number to use when ordering a replacement drive.	
product	String	Specifies the drive product name.	
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**.	
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	For a SATA drive, specifies the drive SATA speed, in Gbps. For a SAS drive, this property is returned with no value.	

(Continued)

Property name	Data type	Description	Notes
sataVersion	String	For a SATA drive, specifies the drive SATA version. For a SAS drive, this property is returned with no value.	
sectorSize	Integer	Specifies the drive sector size, in bytes.	
serial	String	Specifies the drive serial number.	
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. • **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.	

(Continued)

Property name	Data type	Description	Notes
		• **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. • **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	

(Continued)

Property name	Data type	Description	Notes
vendor	String	Specifies the name of the drive vendor.	
wwid	String	Specifies the drive WWID.	

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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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 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 232.	
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 235.	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 238.	
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 241.	

(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 245.	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 247.	
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 253.	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: 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: 11/13/2014	
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	Specifies a comma-separated list of the IP addresses of the DNS servers with which the server module cannot currently communicate.	
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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.	
pageSwapOut	Long	Specifies the number of pages currently swapped out of memory.	
pCode	String	Specifies the HDS part number to use when ordering a replacement server module.	
product	String	Specifies the server module product name.	
rev	String	Specifies the server module hardware revision.	
serial	String	Specifies the serial number of the server module.	

(Continued)

Property name	Data type	Description	Notes
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.	
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.	
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 on mirror set states, see "Hardware: server module mirror set properties" on page 245.	
fwRef	String	Specifies the revision of the firmware currently installed on the disk.	
id	Integer	Specifies the disk ID. For SSD0, the value of this property is always **1007**. For SSD1, the value of this property is always **1008**.	This property is not returned for a USB flash drive.

(Continued)

Property name	Data type	Description	Notes
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 HDS part number to use when ordering a replacement disk.	
product	String	Specifies the disk product name.	
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 disk serial number.	
vendor	String	Specifies the name of the disk vendor.	

(Continued)

Property name	Data type	Description	Notes
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 on mirror set states, see "Hardware: server module mirror set properties" on page 245.	
wwid	String	Specifies the disk WWID.	

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.	

(Continued)

Property name	Data type	Description	Notes
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.	
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.	

(Continued)

Property name	Data type	Description	Notes
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 243.	

(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 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 the next page.	

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.	
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.	
percentComplete	String	If the mirror set is being resyncrhonized or recovered, specifies how much of the resynchronization or recovery process has been completed, as a percent from zero to 100.	This property is not returned if the mirror set is not being resyncrhonized or recovered.

(Continued)

Property name	Data type	Description	Notes
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.	
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 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" below.	
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 250.	

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. Possible values are **eth0** and **eth2**.	

(Continued)

Property name	Data type	Description	Notes
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.	
mode	String	Specifies the actual bonding mode being used on the network connected to the interface. Possible values are active-backup and 802.3ad.	
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**.	

(Continued)

Property name	Data type	Description	Notes
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 .	
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 the next page.	
type	String	Specifies the type of network that's using the interface. The only possible value is **ACCESS**.	

(Continued)

Property name	Data type	Description	Notes
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 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**.	

(Continued)

Property name	Data type	Description	Notes
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 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" below.	

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.	
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 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
```

Response body

```
{
  "enclosureInfo": [
    {
      "error": false,
      "warning": false,
      "notice": "false",
      "id": 1,
      "product": "NDS-4600-JD",
      "vendor": "NEWISYS",
      "fwRev": "0114-104",
      "fwRevs": [
        "0114-104"
      ],
      "wwid": "3500093d00179b000",
      "serial": "MXE340003ATRB0BB",
      "state1": "ADDED",
```

```
"state2": "ADDED",
"status": "AVAILABLE",
"uptime": 337159,
"enclosures": [
  {
    "scpFwRev": "000b",
    "enclConfigRev": "0000",
    "baseboardProduct": "TCA-00341-01-C-A",
    "coverOpen": false,
    "baseboardSerial": "MXE340003ATRB0BB",
    "warning": false,
    "error": false,
    "fail": false,
    "ident": false,
    "predictedFailure": false,
    "swap": false,
    "id": 60,
    "code": 1,
    "location": "Enclosure"
  }
],
"alarms": [
  {
    "error": false,
    "warning": false,
    "muted": false,
    "remind": false,
    "urgency": [
      "INFO"
    ],
    "id": 73,
    "code": 1,
    "location": "BaseBrd Alarm",
    "swap": false,
    "ident": false,
    "fail": false
  }
],
"powerSupplies": [
  {
    "error": false,
    "warning": false,
    "dcUnderVoltage": false,
    "vendor": "Newisys",
    "rev": "A6000000",
    "off": false,
    "supplierProduct": "TDPS-1865AB A000",
```

```
      "dcFail": false,
      "fwRev": "0227",
      "supplierSerial": "AAFT135003246600",
      "serial": "THDEL0003C9VF0AB",
      "acFail": false,
      "dcOverVoltage": false,
      "product": "PWR-00028-02-A00",
      "overTempWarn": false,
      "supplierRev": "05F00000",
      "overTempFail": false,
      "dcOverCurrent": false,
      "id": 61,
      "code": 1,
      "location": "PCM A",
      "swap": false,
      "ident": false,
      "fail": false,
      "pCode": "NDS4600-PS.X"
    },
    {
      .
      .
      .
    }
  ],
  "voltages": [
    {
      "error": false,
      "warning": false,
      "warnOver": false,
      "voltage": 3.3,
      "critUnder": false,
      "critOver": false,
      "warnUnder": false,
      "id": 88,
      "code": 1,
      "location": "PCM A 3.3v",
      "swap": false,
      "ident": false,
      "fail": false,
      "critOverThresh": 20.0,
      "warnOverThresh": 10.0,
      "warnUnderThresh": 10.0,
      "critUnderThresh": 20.0
    },
    {
      .
```

```
      .
      .
    }
  ]
  "currents": [
    {
      "current": 0.0,
      "error": false,
      "warning": false,
      "warnOver": false,
      "critOver": false,
      "id": 94,
      "code": 1,
      "location": "PCM A 3.3vi",
      "swap": false,
      "ident": false,
      "fail": false
    },
    {
      .
      .
      .
    }
  ],
  "sasConnectors": [
    {
      "error": false,
      "warning": false,
      "connectorPhyLink": 255,
      "connectorType": 1,
      "id": 80,
      "code": 5,
      "location": "IOM A MS1",
      "swap": false,
      "ident": false,
      "fail": false
    },
    {
      .
      .
      .
    }
  ],
  "enclosureServices": [
    {
      "error": false,
      "warning": false,
```

```
      "report": false,
      "fruProduct": "",
      "fruSerial": "",
      "fwRev": "0114-104",
      "id": 74,
      "code": 1,
      "location": "IOM A",
      "swap": false,
      "ident": false,
      "fail": false
    },
    {
      .
      .
      .
    }
  ],
  "fans": [
    {
      "actualFanSpeed": 9670,
      "error": false,
      "warning": false,
      "off": false,
      "actualSpeedCode": "HIGHEST_2ND 6",
      "id": 63,
      "code": 1,
      "location": "PCM A Fan 1",
      "swap": false,
      "ident": false,
      "fail": false
    },
    {
      .
      .
      .
    }
  ],
  "sasExpanders": [
    {
      "error": false,
      "warning": false,
      "sasAddr": "0x500093d00179b000",
      "fwRev": "0114-104",
      "nvStoreRevNum": "N/A",
      "arrayIndex": [],
      "id": 76,
      "code": 1,
```

```
      "location": "IOM A Pri Exp",
      "swap": false,
      "ident": false,
      "fail": false
    },
    {
      .
      .
      .
    },
    {
      "error": false,
      "warning": false,
      "sasAddr": "0x500093d00179b080",
      "fwRev": "0114-104",
      "nvStoreRevNum": "N/A",
      "arrayIndex": [
        0,
        1,
        2,
        3,
        4,
        5,
        48,
        49,
        50,
        51,
        52,
        53,
        24,
        25,
        28,
        29
      ],
      "id": 78,
      "code": 1,
      "location": "IOM B Pri Exp",
      "swap": false,
      "ident": false,
      "fail": false
    },
    {
      .
      .
      .
    }
  ],
```

```
"temperatures": [
  {
    "error": false,
    "warning": false,
    "critOver": false,
    "critUnder": false,
    "warnUnder": false,
    "warnOver": false,
    "temperature": 23,
    "id": 67,
    "code": 1,
    "location": "PCM A Temp 1",
    "swap": false,
    "ident": false,
    "fail": false,
    "critOverThresh": 75.0,
    "warnOverThresh": 70.0,
    "warnUnderThresh": 10.0,
    "critUnderThresh": 5.0
  },
  {
    .
    .
    .
  }
],
"slots": [
  {
    "warning": false,
    "error": false,
    "notice": "false",
    "drive": {
      "type": "DATA",
      "failCode": "NONE",
      "errorsDetected": false,
      "error": "false",
      "notice": "false",
       "vendor": "ATA",
      "product": "ST4000DM000-1F21",
      "fwRev": "CC54",
      "fwRevs": [
        "CC54"
      ],
      "serial": "W300DQ2V",
      "wwid": "35000c50069940ff7",
      "capacity": 4000787030016,
      "sectorSize": 512,
```

```
            "rotationRate": 5900,
            "sasAddr": "0x5000c50069940ff6",
            "ataVersion": "ACS-2",
            "sataVersion": "SATA 3.1",
            "sataSpeed": "6 Gbps",
            "formFactor": "",
            "protocol": "SATA",
            "state1": "ADDED",
            "state2": "ADDED",
            "changeTime1": "2015-03-05 10:28:07.91633",
            "changeTime2": "2015-03-04 14:39:01.098144",
            "evacuate": false,
            "reinsert": false,
            "pCode": "ASM-01664-01-A.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": "0x500093d00179b080",
           "sasExpIndex": 78,
          "sasAddr": "0x500093d00179b080",
          "id": 0,
          "code": 1,
          "location": "SLOT 1 ",
          "swap": false,
          "ident": false,
          "fail": false,
          "slotNumber": 1,
          "wwid": "35000c50069940ff7"
        },
        {
          .
          .
          .
        }
      ],
      "ledStates": [
```

```
        1,
        2,
        1,
        0
      ],
      "sbbPowerOnState": "SINGLE_MODE",
      "lockdownReason": "UNDEFINED",
      "pCode": "NDS4600-SHF-01.X"
    }
  ],
  "serverModuleInfo": [
    {
      "warning": false,
      "error": false,
      "id": 1,
      "status": "AVAILABLE",
      "powerOn": true,
      "bmcOnline": true,
      "is_dc": false,
      "peerState": {
        "peers": [
          {
            "ipAddress": "10.1.1.4",
            "bmcOnline": true,
            "powerOn": true
          }
        ]
      },
      "mirrorState": {
        "sets": [
          {
            "status": "OK",
            "warning": false,
            "error": false,
            "devNum": 12
          },
          {
            .
            .
            .
          }
        ]
      },
      "coreHardware": {
        "warning": false,
        "lastUpdate": 1425664801757,
        "upTime": 95623000,
```

```
    "bootTime": 1425569178757,
    "minuteLoad": "0.16",
    "fiveMinuteLoad": "0.11",
    "fifteenMinuteLoad": "0.10",
    "totalMemory": 33704763392,
    "usedMemory": 5135597568,
    "totalSwap": 3221221376,
    "freeSwap": 3221221376,
    "pageSwapIn": 0,
    "pageSwapOut": 0,
    "vendor": "Newisys",
    "product": "NDS-SB1EA",
    "rev": "TCA-00477-01-D",
    "serial": "MXE340003B7RN104",
    "ntpServer": "10.0.201.65",
    "cpus": [
      "Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz"
    ],
    "failedDnsServerConnections": [],
    "loadAvgError": false,
    "swapError": false,
    "error": false,
    "pCode": "ASM-01666-01-A.X",
    "biosVendor": "Newisys",
    "biosFwRev": "HDS 9.00",
    "biosDate": "11/13/2014"
  },
  "networkInterfaces": [
    {
      "eth": {
        "type": "MANAGEMENT",
        "maxSpeed": "1000F",
        "supportedSpeedDuplex": [
          "10H",
          "10F",
          "100H",
          "100F",
          "1000F"
        ],
        "name": "eth1",
        "ok": true,
        "mtu": 1500,
        "error": false,
        "warning": false,
        "speed": 1000,
        "duplex": "full"
      }
```

```
    },
    {
      .
      .
      .
    },
    {
      "bond": {
        "type": "ACCESS",
        "name": "bond0",
        "ok": true,
        "mtu": 1500,
        "error": false,
        "warning": false,
        "activeSlave": "eth0",
        "mode": "active-backup",
        "slaves": [
          {
            "type": "ACCESS",
            "maxSpeed": "10000F",
            "supportedSpeedDuplex": [
              "10000F"
            ],
            "name": "eth0",
            "ok": true,
            "mtu": 1500,
            "error": false,
            "degraded": false,
            "speed": 10000,
            "duplex": "full"
          },
          {
            .
            .
            .
          }
        ]
      }
    }
  ],
  "fileSystems": [
    {
      "mountPoint": "/",
      "availableSpace": 3655524352,
      "usedSpace": 1205493760,
      "totalSpace": 5146017792,
      "usedSpacePercentage": 24,
```

```
      "totalInodes": 327680,
      "usedInodes": 40388,
      "cutoff": false,
      "error": false,
      "warning": false
    },
    {
      .
      .
      .
    }
  ],
  "ipmi": [
    {
      "sensorType": "TEMPERATURE",
      "sensors": [
        {
          "name": "Temp_1",
          "detailedStatus": "33.0C (91.4F); (range 8.0-77.0C)",
          "error": false,
          "warning": false
        },
        {
          .
          .
          .
        }
      ],
      "error": false,
      "warning": false
    },
    {
      .
      .
      .
    }
  ],
  "disks": [
    {
      "operatingSystem": true,
      "removable": false,
      "warning": false,
      "error": false,
      "id": 1007,
      "vendor": "ATA",
      "product": "StorFly VSF202CC",
      "fwRev": "0828-000",
```

```
            "serial": "P1T13003454211120054",
            "wwid": "1ATA     StorFly VSF202CC100G-NEW     P1T13003454211120054",
            "capacity": 120034123776,
            "pCode": "ASM-01667-01-A.X"
          },
          {
            .
            .
            .
          }
        ]
      },
      {
        .
        .
        .
      }
    ]
}
```


/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 on the /hardware/beacon/enclosure/*enclosure-id* resource, see "Beaconing resources" on page 55.

/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 on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/beacon/enclosure/1?on"
```

Request headers

```
POST /mapi/configuration/mapi/hardware/beacon/enclosure/1?on HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-module-id*

With the /hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-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 on the /hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-module-id* resource, see "Beaconing resources" on page 55.

/hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-module-id* query parameters

To turn power and cooling module beaconing on and off, you use query parameters with a **POST** request for the /hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-module-id* resource. The query parameters you use are:

- **on** — Turns beaconing on for the specified power and cooling module
- **off** — Turns beaconing off for the specified power and cooling module

For more information on query parameters, see "Management API query parameters" on page 43.

/hardware/beacon/enclosure/*enclosure-number*/power_supply/*power-and-cooling-module-id* example

Here's a sample **POST** request that turns beaconing on for power and cooling module 74 in enclosure 1.

Request with curl command line

```
curl -k -X POST -H "X-HCPS-API-VERSION: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/beacon/enclosure
  /1/power_supply/74?on"
```

Request headers

```
POST /mapi/configuration/mapi/hardware/beacon/enclosure/1/power_supply/74?on
HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /hardware/beacon/server_module/*server-module-number* resource, see "Beaconing resources" on page 55.

/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 on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/beacon/server_module/1
  ?on"
```

Request headers

```
POST /mapi/configuration/mapi/hardware/beacon/server_module/1?on HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /hardware/maintenance resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 hard disk drives to the S Series Node. • **REMOVE_DRIVE** — Remove one or more hard disk drives from the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
		• **REPLACE_DRIVE** — Replace one or more hard disk drives in the S Series Node. • **REPLACE_ENCLOSURE** — Replace the S Series Node enclosure. 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" on page 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 177
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2015-02-27 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 on the /hardware/maintenance/active resource, see "Maintenance resources" on page 63.

/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 385.	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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 618
```

Response body


```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "slotNumber": 7
        },
        "drive": {}
      }
    ]
  }
}
```


/hardware/maintenance/history

With the /hardware/maintenance/history resource, a **GET** request returns a response body.

For more information on the /hardware/maintenance/history resource, see "Maintenance resources" on page 63.

/hardware/maintenance/history property

The table below describes the top-level property in /hardware/maintenance/history resource response bodies.

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 385.	If no maintenance procedures have been performed, the value of this property is an empty list.

/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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/history
  ?prettyprint"
```

Request headers

```
GET /mapi/hardware/maintenance/history?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 2083
```

Response body


```
{
  "maintProcedures": [
    {
      "id": 10,
      "maintType": "ADD_DRIVE",
      "state": "COMPLETED",
      "startTime": "2015-02-27 09:21:52 UTC",
      "startTsExtra": 77,
      "endTime": "2015-02-27 09:46:22 UTC",
      "notes": "Reinserting incorrectly removed drive.",
      "selections": {
        "maintSelections": [
          {
            "state": "ADDED",
            "code": "NONE",
            "codeString": "None",
            "enclosure": {
              "wwid": "3500093d00179b000",
              "id": 1,
              "product": "NDS-4600-JD",
              "serial": "MXE340003ATRB0BB",
              "slotNumber": 7
            },
            "drive": {
              "wwid": "35000c5006990b1d1",
              "vendor": "ATA",
              "product": "ST4000DM000-1F21",
              "serial": "W300DFDK",
              "capacity": 4000787030016,
              "state": "ADDED",
              "failCode": "NONE"
            }
          }
        ]
```

```
      }
    },
    {
      "id": 9,
      "maintType": "REMOVE_DRIVE",
      "state": "COMPLETED",
      "startTime": "2015-02-27 09:47:41 UTC",
      "startTsExtra": 95,
      "endTime": "2015-02-27 09:50:23 UTC",
      "selections": {
        "maintSelections": [
          {
            "state": "REMOVED",
            "code": "NONE",
            "codeString": "None",
            "enclosure": {
              "wwid": "3500093d00179b000",
              "id": 1,
              "product": "NDS-4600-JD",
              "serial": "MXE340003ATRB0BB",
              "slotNumber": 7
            },
            "drive": {
              "wwid": "35000c5006990b1d1",
              "vendor": "ATA",
              "product": "ST4000DM000-1F21",
              "serial": "W300DFDK",
              "capacity": 4000787030016,
              "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 on the /hardware/maintenance/*procedure-id* resource, see "Maintenance resources" on page 63.

/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 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 747
```

Response body

```
{
  "id": 9,
  "maintType": "REMOVE_DRIVE",
  "state": "PERFORMING",
  "startTime": "2015-02-27 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": {
            "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 on the /hardware/maintenance/*procedure-id*/cancel resource, see "Maintenance resources" on page 63.

/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 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 878
```

Response body

```
{
  "id": 11,
  "maintType": "REMOVE_DRIVE",
  "state": "CANCELED",
  "startTime": "2015-02-27 15:49:25 UTC",
  "startTsExtra": 128,
  "endTime": "2015-02-27 15:51:21 UTC",
  "selections": {
    "maintSelections": [
      {
        "state": "FAILED",
        "code": "CANCELED",
        "codeString": "Marked failed due to maintenance procedure cancelation",
        "codeSetInState": "PERFORMING",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "slotNumber": 7
```

```
        },
        "drive": {
          "wwid": "35000c5006990b1d1",
          "vendor": "ATA",
          "product": "ST4000DM000-1F21",
          "serial": "W300DFDK",
          "capacity": 4000787030016,
          "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 on the /hardware/maintenance/*procedure-id*/candidates resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/candidates resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 390.	

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 313
```

Response body

```
{
  "maintSelections": [
    {
      "state": "NONE",
      "code": "NONE",
      "codeString": "None",
      "enclosure": {
        "wwid": "3500093d00179b000",
        "id": 1,
        "product": "NDS-4600-JD",
        "serial": "MXE340003ATRB0BB",
        "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 on the /hardware/maintenance/*procedure-id*/complete resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/complete resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 829
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "COMPLETED",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "endTime": "2015-02-27 09:46:22 UTC",
  "notes": "Reinserting incorrectly removed drive.",
```

```
"selections": {
  "maintSelections": [
    {
      "state": "ADDED",
      "code": "NONE",
      "codeString": "None",
      "enclosure": {
      "wwid": "3500093d00179b000",
        "id": 1,
        "product": "NDS-4600-JD",
        "serial": "MXE340003ATRB0BB",
        "slotNumber": 7
      },
        "drive": {
        "wwid": "35000c5006990b1d1",
        "vendor": "ATA",
        "product": "ST4000DM000-1F21",
        "serial": "W300DFDK",
        "capacity": 4000787030016,
        "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 on the /hardware/maintenance/*procedure-id*/confirm resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/confirm resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 385.

/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": "3500093d00179b000"
      },
      "confirmAction": "true"
    }
  ]
}
```

Request with curl command line

```
curl -k -X POST -d @no_format_drive.json -H "X-HCPS-API-VERSION: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 791
```

Response body


```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "VERIFIED",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "slotNumber": 7
        },
        "drive": {
          "wwid": "35000c5006990b1d1",
          "vendor": "ATA",
          "product": "ST4000DM000-1F21",
          "serial": "W300DFDK",
          "capacity": 4000787030016,
          "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 on the /hardware/maintenance/*procedure-id*/perform resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/perform resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 494
```

Response body


```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "PERFORMING",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "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 on the /hardware/maintenance/*procedure-id*/select resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/select resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 "Hard disk 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.	

Hard disk 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 385.

/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": "3500093d00179b000"
      }
    }
  ]
}
```

Request with curl command line

```
curl -k -X POST -d @slot_selection.json -H "X-HCPS-API-VERSION: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 490
```

Response body


```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "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 on the /hardware/maintenance/*procedure-id*/update resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/update resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 543
```

Response body

```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "STARTED",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "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 on the /hardware/maintenance/*procedure-id*/verify resource, see "Maintenance resources" on page 63. For an example of using the /hardware/maintenance*procedure-id*/verify resource in a maintenance procedure, see "Replacing a hard disk drive" on page 379.

/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, see "Maintenance procedure properties" on page 385.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 799
```

Response body


```
{
  "id": 10,
  "maintType": "ADD_DRIVE",
  "state": "ACTION",
  "startTime": "2015-02-27 09:21:52 UTC",
  "startTsExtra": 77,
  "notes": "Reinserting incorrectly removed drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ACTION_NATIVE",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "slotNumber": 7
        },
        "drive": {
          "wwid": "35000c5006990b1d1",
          "vendor": "ATA",
          "product": "ST4000DM000-1F21",
          "serial": "W300DFDK",
          "capacity": 4000787030016,
          "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 on the /hardware/power/node resource, see "Power resources" on page 67.

/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 from one 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 on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /hardware/power/*server-module-number* resource, see "Power resources" on page 67.

/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 from one 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 on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on 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 73.

For more information on the /metrics/buckets resource, see "Metrics resources" on page 65.

/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 "Management API count query parameter" on page 74.	
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 complete. • **false** — The bucket list is incomplete. For more information, see "Management API count query parameter" on page 74.	

(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 "Management API marker query parameter" on page 75.	

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 size of the data stored 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 241
```

Response body


```
{
  "buckets": [
    {
      "name": "hcpsrv-hcp-ma",
      "objectcount": 47950,
      "logicalUsedBytes": 51485920460800,
      "physicalUsedBytes": 66931696599040
    }
  ],
  "marker": "",
  "bucketCount": 1000,
  "isTruncated": false
}
```


/metrics/gateways

With the /metrics/gateways resource, a **GET** request returns a response body.

For more information on the /metrics/gateways resource, see "Metrics resources" on page 65.

/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 HS3 protocol. These properties are described in the next table.	

The table below describes the properties used to provide information about the use of the HS3 protocol 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: 2015-01-20 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: 2015-01-20 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 HS3 API.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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": "2015-02-04 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": "2015-02-04 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": "2015-02-04 14:50:35 EST"
  }
}
```


/metrics/protection

With the /metrics/protection resource, a **GET** request returns a response body.

For more information on the /metrics/protection resource, see "Metrics resources" on page 65.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /metrics/system resource, see "Metrics resources" on page 65.

/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.	
capacity	Long	Specifies the total number of bytes that can be used for storing additional data and protecting data on the S Series Node (that is, the usable capacity). This is equal to the licensed capacity minus the space required for system overhead.	
capacityAvailable	Long	Specifies the total number of bytes currently available for storing additional data and protecting data. This is equal to the value of the capacity property minus the value of the capacityUsed property.	

(Continued)

Property name	Data type	Description	Notes
capacityUsed	Long	Specifies the total number of bytes currently used for storing and protecting data.	
objectCount	Long	Specifies the total number of objects currently stored on the S Series Node.	
percentUsed	Float	Specifies the percent, from zero to 100, of the total storage capacity that's currently used for storing and protecting data.	
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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 192
```

Response body

```
{
  "bucketCount": 1,
  "userCount": 7,
  "objectCount": 47970,
  "capacity": 105991847175783,
  "capacityAvailable": 39031977858663,
  "capacityUsed": 66959869317120,
  "percentUsed": 63.17
}
```

/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 on the query parameters used to limit the list of irreparable objects returned by a **GET** request, see "Managing resource lists" on page 73.

For more information on the /system/irreparables resource, see "Irreparables resources" on page 60.

/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 "Management API count query parameter" on page 74.	
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 complete. • **false** — The irreparable object list is incomplete. For more information, see "Management API count query parameter" on page 74.	

(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 "Management API marker query parameter" on page 75.	
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.	

(Continued)

Property name	Data type	Description	Notes
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: 2015-01-20 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.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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":"2015-02-08 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /system/license resource, see "License resource" on page 61.

/system/license properties

The table below describes the properties in /system/license resource response bodies.

Property name	Data type	Description	Notes
capacity	Long	Specifies the total storage capacity that can be installed in the S Series Node without violating the license agreement, in bytes.	

(Continued)

Property name	Data type	Description	Notes
expirationDate	Timestamp	Specifies the license expiration date, in this format. *yyyy-MM-dd hh:mm:ss* UTC For example: 2015-01-20 18:28:57 UTC If the license has no expiration date, the value of this property looks like this: 292278994-08-17 02:12:55 UTC. The time varies depending on the time the license was generated.	
quoteNumber	String	Specifies the seven-digit quote number for the license.	
serialNumber	String	Specifies the five-digit serial number for the S Series Node.	

(Continued)

Property name	Data type	Description	Notes
status	String	Specifies the status of the license. Possible values are: • **VALID** — The S Series Node has a valid license. • **EXCEEDED** — The storage installed on the S Series Node exceeds the licensed capacity. • **EXPIRED** — The S Series Node license is expired. • **NONE** — The S Series Node does not have an installed license. • **UNKNOWN** — The S Series Node cannot determine the status of its license.	
type	String	Specifies the product the license applies to. For an S Series Node, the value of this property is always **HCPS**.	

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 175
```

Response body

```
{
  "type": "HCPS",
  "status": "VALID",
  "serialNumber": "12345",
  "quoteNumber": "98765432",
  "capacity": 112000000000000,
  "expirationDate": "2018-09-09 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 on the /system/logs/cancel resource, see:

- "HCP S Series Node internal logs" on page 29
- "Log resources" on page 62
- "Downloading the internal logs" on page 376

/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 on the /system/logs/download resource, see:

- "HCP S Series Node internal logs" on page 29

- "Log resources" on page 62
- "Downloading the internal logs" on page 376

/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 on the /system/logs/mark resource, see "Log resources" on page 62.

/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 from one through 1,024 characters long and can contain any valid UTF-8 characters, including percent-encoded white space.

For more information on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /system/logs/prepare resource, see:

- "HCP S Series Node internal logs" on page 29
- "Log resources" on page 62
- "Downloading the internal logs" on page 376

/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:

02/11/2015

For more information on the /system/logs/prepare resource, see:

- "HCP S Series Node internal logs" on page 29
- "Log resources" on page 62
- "Downloading the internal logs" on page 376

/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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/prepare?startDate=02/10/2015
  &endDate=02/11/2015"
```

Request headers

```
POST /mapi/configuration/mapi/system/logs/prepare?startDate=02/10/2015
&endDate=02/11/2015 HTTP/1.1
Host: 10.0.0.3:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

For more information on the /system/logs/prepare resource, see:

- "HCP S Series Node internal logs" on page 29
- "Log resources" on page 62
- "Downloading the internal logs" on page 376

/system/logs/status

With the /system/logs/status resource, a **GET** request returns a response body.

For more information on the /system/logs/status resource, see:

- "HCP S Series Node internal logs" on page 29
- "Log resources" on page 62
- "Downloading the internal logs" on page 376

/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 376.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 376.

/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 on the /system/misc/setting/network/management/monitor resource, see "Miscellaneous settings resource" on page 66.

/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** — Disable management network monitoring.
- **false** — Enable management network monitoring.

These values are not case sensitive.

For more information on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /system/full health resource, see "Status resources" on page 69.

/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 81.	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 304.	
gatewayMetrics	Object	Specifies a set of properties that provide information about use of the HS3 protocol. For descriptions of these properties, see "/metrics/gateways properties" on page 307.	

(Continued)

Property name	Data type	Description	Notes
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 160.	
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 314.	
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 339.	
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 113.	
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 315.	

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 114.
    },
    "systemHealth": {
      See the response body in "/system/status/health example" on page 340.
    },
    "alertInfo": {
      See the response body in "/alert example" on page 87.
    },
    "systemMetrics": {
      See the response body in "/metrics/system example" on page 316.
    },
    "bucketMetrics": {
```

```
      See the response body in "/metrics/buckets example" on page 305.
    },
    "gatewayMetrics": {
      See the response body in "/metrics/gateways example" on page 312.
    },
    "protectionMetrics": {
      See the response body in "/metrics/protection example" on page 314.
    },
    "hardware": {
      See the response body in "/hardware example" on page 255.
    }
  }
}
```

/system/status/health

With the /system/status/health resource, a **GET** request returns a response body.

For more information on the /system/status health resource, see "Status resources" on page 69.

/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.	

(Continued)

Property name	Data type	Description	Notes
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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /system/update/apply resource, see:

- "HCP S Series software and firmware maintenance" on page 30
- "Update resources" on page 71
- "Performing an update" on page 408

/system/update/history

With the /system/update/history resource, a **GET** request returns a response body.

For more information on the /system/update/history resource, see "HCP S Series software and firmware maintenance" on page 30 and "Update resources" on page 71.

/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 and license 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.	
timestamp	Timestamp	Specifies the time at which the update finished, in this format: *yyyy-MM-dd hh:mm:ss* UTC For example: 2015-01-20 18:28:57 UTC	
title	String	Specifies the title of the update from the update file For a license update, the value of this property is always **HCP S Series Node License**.	

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	For a software upgrade file or hotfix file, specifies the version of the HCP S Series software in the file. For a license file, specifies the version of the tool used to create the file.	

/system/update/history example

Here's a sample **GET** request that retrieves the update history for the HCP S Series.

Request with curl command line

```
curl -k -H "X-HCPS-API-VERSION: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 787
```

Response body

```
{
  "historyList": [
    {
      "timestamp": "2015-07-13 15:02:40 UTC",
      "title": "HCP S Series Node License",
      "manifest": {
        "title": "HCPS License",
        "description": "HCPSLic_SN12345_Q9876543_C336TB_07-10-
2020.plk\n\nThis license file increases the total storage capacity.",
        "impact": "LOW",
        "impactDescription": "This package will cause no disruption of service.",
        "version": "1.0.0"
      }
    },
    {
      "timestamp": "2015-01-12 16:06:28 UTC",
      "title": "HCP S Series 1.0.1.12",
      "manifest": {
        "title": "HCP S Series 1.0.1.12",
        "description": "System installed to version 1.0.1.12",
        "impact": "NONE",
        "impactDescription": "No impact.",
        "version": "1.0.1.12"
      }
    }
  ]
}
```

/system/update/manifest

With the /system/update/manifest resource, a **GET** request returns a response body.

For more information on the /system/update/manifest resource, see "Update resources" on page 71.

/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 making the update will have on the S Series Node. Possible values are: • **NONE** • **LOW** • **MEDIUM** • **HIGH**	
impactDescription	String	Specifies a description of the impact making the update will have on the S Series Node.	
title	String	Specifies the title of the update from the update file manifest.	
version	String	For a software upgrade file or hotfix file, specifies the version of the HCP S Series software in the file. For a license file, specifies the version of the tool used to create the 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: 1.0.1"
  -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: 1.0.1
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: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 680
```

Response body


```
{
  "title": "HCP S Series 1.0.1.14",
  "description": "The HCP S Series Node version 1.0.1 update includes general
availability of the RESTful management API, whole-object deduplication, and the
resolution of several known issues.\n",
  "impact": "MEDIUM",
  "impactDescription": "\u003cp\u003eVersion restrictions: You can upgrade to release
1.0.1 only from release 1.0.\u003c/p\u003e\n\u003cp\u003eDuring an upgrade: Server
modules are upgraded one at a time. While a module is being upgraded, it is unavailable.
\u003cbr\u003eYou can make configuration changes, but certain changes won’t take
effect until the upgrade is complete.\u003c/p\u003e\n",
  "version": "1.0.1.14"
}
```

/system/update/progress

With the /system/update/progress resource, a **GET** request returns a response body.

For more information on the /system/update/progress resource, see:

- "HCP S Series software and firmware maintenance" on page 30
- "Update resources" on page 71
- "Performing an update" on page 408

/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" below.	
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.	

(Continued)

Property name	Data type	Description	Notes
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.	For a software upgrade or hotfix application, the server module is automatically rebooted after the update shows as one-hundred percent complete on it.
serverModuleNumber	Integer	Specifies the server module number.	

For an example of using the /system/update/progress resource, see "Performing an update" on page 408.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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
      }
    ]
  }
}
```

For an example of using the /system/update/progress resource in an update operation, see "Performing an update" on page 408.

/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 on the /system/update/restart resource, see:

- "HCP S Series software and firmware maintenance" on page 30
- "Update resources" on page 71
- "Performing an update" on page 408

/system/update/status

With the /system/update/status resource, a **GET** request returns a response body.

For more information on the /system/update/status resource, see:

- "HCP S Series software and firmware maintenance" on page 30
- "Update resources" on page 71
- "Performing an update" on page 408

/system/update/status property

The table below describes the property in /system/update/manifest 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:	

(Continued)

Property name	Data type	Description	Notes
		• **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. • **PREUPGRADE** — For a software upgrade or hotfix application, the S Series Node is performing prechecks to ensure that it is ready for a software upgrade or hotfix application. This is part of the apply step of an update operation. • **UPGRADING** — For a software upgrade or hotfix application, the S Series Node is in the process of updating the HCP S Series OS or software.	

(Continued)

Property name	Data type	Description	Notes
		• **COMPLETE** — A software upgrade or hotfix application 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**.	

For an example of using the /system/update/status resource, see "Performing an update" on page 408.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 27
```

Response body

```
{
  "state": "EXTRACTED"
}
```

For an example of using the /system/update/status resource in an update operation, see "Performing an update" on page 408.

/system/update/upload/license

With the /system/update/upload/license resource, a **PUT** request requires a license file as input and returns a response body.

For more information on the /system/update/upload/license resource, see:

- "HCP S Series software and firmware maintenance" on page 30
- "HCP S Series Node update files" on page 31
- "Update resources" on page 71
- "Performing an update" on page 408

/system/update/upload/license properties

The table below describes the properties in /system/update/upload/license resource response bodies.

Property name	Data type	Description	Notes
description	String	Specifies a description of the update.	
impact	String	Specifies the level of impact making the update will have on the S Series Node. Possible values are: • **NONE** • **LOW** • **MEDIUM** • **HIGH**	
impactDescription	String	Specifies a description of the impact making the update will have on the S Series Node.	
title	String	Specifies the title of the update from the update file manifest.	
version	String	For a software upgrade file or hotfix file, specifies the version of the HCP S Series software in the file. For a license file, specifies the version of the tool used to create the file.	

For an example of using the /system/update/upload/license resource, see "Performing an update" on page 408.

/system/update/upload/license example

Here's a sample **PUT** request that uploads a license file.

Request with curl command line

```
curl -k -T HCPSLic_SN12345_Q9876543_C336TB_07-10-2020.plk
  -H "X-HCPS-API-VERSION: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/upload/license
  ?prettyprint"
```

Request headers

```
PUT /system/update/upload/license?prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
Content-Length: 14016
```

Response headers

```
HTTP/1.1 201 Created
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 272
```

Response body

```
{
  "title": "HCPS License",
  "description": "HCPSLic_SN12345_Q9876543_C336TB_07-10-2020.plk\n\nThis license
file increases the total storage capacity.",
  "impact": "LOW",
  "impactDescription": "This package will cause no disruption of service.",
  "version": "1.0.0"
}
```

For an example of using the /system/update/upload/license resource, see "Performing an update" on page 408.

/system/update/upload/software

With the /system/update/upload/software resource, a **PUT** request requires a software upgrade file as input and returns a response body.

For more information on the /system/update/upload/software resource, see:

- "HCP S Series software and firmware maintenance" on page 30
- "HCP S Series Node update files" on page 31
- "Update resources" on page 71
- "Performing an update" on page 408

/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 making the update will have on the S Series Node. Possible values are: • **NONE** • **LOW** • **MEDIUM** • **HIGH**	
impactDescription	String	Specifies a description of the impact making the update will have on the S Series Node.	
title	String	Specifies the title of the update from the update file manifest.	

(Continued)

Property name	Data type	Description	Notes
version	String	For a software upgrade file or hotfix file, specifies the version of the HCP S Series software in the file. For a license file, specifies the version of the tool used to create the file.	

For an example of using the /system/update/upload/software resource, see "Performing an update" on page 408.

/system/update/upload/software example

Here's a sample **PUT** request that uploads a software upgrade file.

Request with curl command line

```
curl -k -T HS437_1.0.1.14.bin -H "X-HCPS-API-VERSION: 1.0.1"
  -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: 1.0.1
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: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 680
```

Response body


```
{
  "title": "HCP S Series 1.0.1.14",
  "description": "The HCP S Series Node version 1.0.1 update includes general
availability of the RESTful management API, whole-object deduplication, and the
resolution of several known issues.\n",
  "impact": "MEDIUM",
  "impactDescription": "\u003cp\u003eVersion restrictions: You can upgrade to release
1.0.1 only from release 1.0.\u003c/p\u003e\n\u003cp\u003eDuring an upgrade: Server
modules are upgraded one at a time. While a module is being upgraded, it is unavailable.
\u003cbr\u003eYou can make configuration changes, but certain changes won't take
effect until the upgrade is complete.\u003c/p\u003e\n",
  "version": "1.0.1.14"
}
```


For an example of using the /system/update/upload/software resource, see "Performing an update" on page 408.

/user_accounts

With the /user_accounts resource:

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

For information on the query parameters used to limit the user account list returned by a **GET** request, see "Managing resource lists" on page 73.

For more information on the /user_accounts resource, see "User account resources" on page 72.

/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 364.

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 "Management API count query parameter" on page 74.	
isTruncated	Boolean	Specifies whether the returned list of user accounts is complete. Possible values are: • **true** — The user account list is complete. • **false** — The user account list is incomplete. For more information, see "Management API count query parameter" on page 74.	
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 "Management API marker query parameter" on page 75.	

(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 "Management API prefix query parameter" on page 76.	
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_acounts 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
Content-Length: 255
```

Response headers

```
HTTP/1.1 201 Created
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 0
```

/user_acounts 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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /user_accounts/*username* resource, see "User account resources" on page 72.

/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: 2015-01-20 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 from one 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: 2015-01-20 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: 2015-01-20 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 6.	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 5.	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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 445
```

Response body


```
{
  "passwordExpires": true,
  "lastPasswordChangeTime": "2015-03-17 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": "2015-01-23 13:48:28 UTC",
  "failedLoginAttempts": 0,
  "lastLoginTime": "2015-03-17 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 on the /user_accounts/*username*/access_key/generate resource, see "User account resources" on page 72.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
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 on the /versions resource, see "Versions resource" on page 73.

/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 on query parameters, see "Management API query parameters" on page 43.

/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: 1.0.1"
  -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: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 85
```

Response body

```
{
  "supportedVersions": [
    "1.0.0",
    "1.0.1"
  ],
  "latestVersion": "1.0.1"
}
```

/versions POST example

Here's a sample **POST** request that checks whether version 1.0.2 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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/versions?version=1.0.2&prettyprint"
```

Request headers

```
POST /mapi/versions?version=1.0.2&prettyprint HTTP/1.1
Host: mapi.s-node-1.example.com:9090
X-HCPS-API-VERSION: 1.0.1
Authorization: Basic YWRtaW46U3RhcnQxMjMh
```

Response headers

```
HTTP/1.1 200 OK
Server: HCP S Series/1.0.2.6
X-HCPS-Domain-Name: s-node-1.example.com
X-HCPS-Server-Module-Number: 2
X-HCPS-SUPPORTED-API-VERSIONS: [1.0.0, 1.0.1]
X-HCPS-API-VERSION: 1.0.1
Content-Type: application/json;charset=UTF-8
Content-Length: 46
```

Response body

```
{
  "version": "1.0.2",
  "supported": false
}
```

6

Management API procedures

Most management API resources are used to perform a standalone action, such as 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
- Adding, removing, or replacing hard disk drives or replacing the enclosure
- Upgrading the HCP S Series software, applying a hotfix, or installing a new license

This chapter contains instructions for using the management API to perform each of these types of procedures.

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: 1.0.1"
  -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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/prepare?startDate=02/10/2015
  &endDate=02/11/2015"
```

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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/download" > logs-2-11-15.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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://10.0.0.3:9090/mapi/system/logs/cancel"
```

For more information on the S Series Node internal logs, see "HCP S Series Node internal logs" on page 29. For more information on the resources used during a log download operation, see "Log resources" on page 62.

Performing a hardware maintenance procedure

A hardware maintenance procedure starts when you issues 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.

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). As part of this step for a replace enclosure procedure, the S Series Node automatically shuts down.

5. Physically add, remove, or replace the target components. For instructions on performing these physical tasks, see the S Series Node Help.

6. Verify that no errors have occurred during the procedure (/hardware/maintenance/*procedure-id*/verify).

7. For an add or replace drives 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 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.

8. Complete the procedure (/hardware/maintenance/*procedure-id*/complete). You need to perform this step even if the errors occurred during the procedure.

For more information about hardware maintenance procedures, see "HCP S10 Node hardware maintenance" on page 33.

Replacing a hard disk 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: 1.0.1"
  -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": "2015-03-02 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: 1.0.1"
  -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: 1.0.1"
  -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": "3500093d00179b000",
      "id": 1,
      "product": "NDS-4600-JD",
      "serial": "MXE340003ATRB0BB",
      "slotNumber": 2
    },
    "drive": {
      "wwid": "35000c500699065a4",
      "vendor": "ATA",
      "product": "ST4000DM000-1F21",
```

```
        "serial": "W300DSV4",
        "capacity": 4000787030016,
        "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: 1.0.1"
  -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": "3500093d00179b000"
      },
      "drive": {
        "wwid": "35000c500699065a4"
      }
    }
  ]
}
```

Note: 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: 1.0.1"
  -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": "2015-03-02 14:13:13 UTC",
  "startTsExtra": 189,
  "notes": "Replacing corrupt drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ADD",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "slotNumber": 2
        },
        "drive": {},
        "replacedDrive": {
          "wwid": "35000c500699065a4",
          "vendor": "ATA",
          "product": "ST4000DM000-1F21",
          "serial": "W300DSV4",
          "capacity": 4000787030016,
          "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 S Series Node Help.

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: 1.0.1"
  -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": "2015-03-02 14:13:13 UTC",
  "startTsExtra": 189,
  "notes": "Replacing corrupt drive.",
  "selections": {
    "maintSelections": [
      {
        "state": "ACTION_FOREIGN",
        "code": "NONE",
        "codeString": "None",
        "enclosure": {
          "wwid": "3500093d00179b000",
          "id": 1,
          "product": "NDS-4600-JD",
          "serial": "MXE340003ATRB0BB",
          "slotNumber": 2
        },
        "drive": {
          "wwid": "35000c5006cf531e1",
          "vendor": "ATA",
          "product": "ST4000DM000-1F21",
          "serial": "S30052HR",
          "capacity": 4000787030016,
          "state": "DISCOVERED",
          "failCode": "NONE"
```

```
        },
        "replacedDrive": {
          "wwid": "35000c500699065a4",
          "vendor": "ATA",
          "product": "ST4000DM000-1F21",
          "serial": "W300DSV4",
          "capacity": 4000787030016,
          "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: 1.0.1"
  -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": "3500093d00179b000"
      },
      "drive": {
        "wwid": "35000c5006cf531e1"
      },
      "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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/hardware/maintenance/12
  /complete?prettyprint"
```

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: 2015-01-20 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 hard disk drives to an enclosure. • **REMOVE_DRIVE** — Remove one or more hard disk drives from an enclosure. • **REPLACE_DRIVE** — Replace one or more hard disk drives in an enclosure. • **REPLACE_ENCLOSURE** — Replace an enclosure.	
notes	String	Specifies user-supplied text that is associated with the maintenance procedure.	This property is returned only if it has a value.

(Continued)

Property name	Data type	Description	Notes
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 389.	
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: 2015-01-20 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. • **CANCELED** — The maintenance procedure was canceled before it was completed.	

(Continued)

Property name	Data type	Description	Notes
		• **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. • **VERIFYING** — The S Series Node is checking whether the maintenance procedure was performed correctly.	

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" on the next page.	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:	This property is not returned until the maintenance procedure has been verified.

(Continued)

Property name	Data type	Description	Notes
		• **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. • **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.	

(Continued)

Property name	Data type	Description	Notes
		• **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 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. • **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.	

(Continued)

Property name	Data type	Description	Notes
		• **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 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: hard disk drive properties" on page 403. 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 402.	

(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: hard disk drive properties" on page 403.	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.
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.	

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
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.	
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 a replace enclosure procedure, the value of this property is **-1**.	
wwid	String	Specifies the enclosure WWID from the internal database.	

Maintenance procedure: hard disk drive properties

The table below describes the properties used to provide information about the hard disk 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 hard disk drive, in bytes.	
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. • **MAINT_CANCEL** — The drive was a target component for a canceled maintenance procedure.	

(Continued)

Property name	Data type	Description	Notes
		• **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. • **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.	

(Continued)

Property name	Data type	Description	Notes
		• **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.	
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 system. • **ADDED** — The drive is part of the system. • **ADDING** — The drive is in the process of being integrated into the system. • **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.	

(Continued)

Property name	Data type	Description	Notes
		• **MIRROR** — The drive is a database drive that is set to be integrated into the system. • **MIRRORED** — The drive is a database drive that is part of the system. • **MISSING** — The drive is unavailable. • **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 system. • **REMOVED** — The drive has been logically removed from the system. • **REMOVING** — The drive is in the process of being logically removed from the system.	
vendor	String	Specifies the name of the drive vendor from the internal database.	
wwid	String	Specifies the drive WWID from the internal database.	

Performing an update

An update operation can be a software upgrade, a hotfix application, or an installation of a new license. In all cases, the operation entails an upload step and an apply step, as explained in "HCP S Series software and firmware maintenance" on page 30.

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/license or /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 software, license, and firmware updates" on page 31.

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: 1.0.1"
  -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
  "https://mapi.s-node-1.example.com:9090/mapi/system/update/status
  ?prettyprint"
```

You can start an update operation if the value of the state property in the response body is **READY**:

```
{
  "state": "READY"
}
```

2. Do either of these:

   - To upload a license file, issue a **PUT** request for the /system/update/upload/license resource:

     ```
     curl -k -T HCPSLic_SN12345_Q9876543_C336TB_07-10-2020.plk
       -H "X-HCPS-API-VERSION: 1.0.1"
       -H "Authorization: Basic YWRtaW46U3RhcnQxMjMh"
       "https://mapi.s-node-
     1.example.com:9090/mapi/system/update/upload/license
       ?prettyprint"
     ```

   - To upload a software upgrade file or hotfix file, issue a **PUT** request for the /system/update/upload/software resource:

     ```
     curl -k -T HS437_1.1.0.17.bin -H "X-HCPS-API-VERSION: 1.0.1"
       -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 if 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: 1.0.1"
  -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: 1.0.1"
  -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: 1.0.1"
       -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

This appendix contains a table that describes the possible HTTP status codes returned in response to HCP S Series management API requests.

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.
400	Bad Request	All	The request was not valid. These are some, but not all, of the possible reasons: • The management API version specifies 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 • For a PUT request to upload a software upgrade file, the version of the HCP S Series software in the file is earlier than the currently installed version.
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.

Glossary

A

access network

The network used for external client access to the HCP S Series Node though the HCP S Series HS3 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.

B

bucket

A named logical container for objects. Buckets group objects such that the objects in one bucket are not visible in any other bucket.

D

data access protocol

A method of access to an HCP S Series Node for the purpose of reading and writing data. For data access protocols, S Series Nodes support only the HS3 API.

E

enclosure

In an HCP S10 Node, the container for the power and cooling modules, server modules, and hard disk drives.

F

foreign drive

A hard disk drive that was previously used in a different S Series Node or in the current S Series Node before the HCP S Series software on the S Series Node was upgraded or reinstalled.

H

HCP

See Hitachi Content Platform.

HCP S Series HS3 API

An Amazon S3-compatible API for writing, retrieving, and otherwise managing objects in an HCP S Series Node.

HCP S Series management API

A RESTful HTTP interface to the administrative functions of an HCP S Series Node. Using the management API, you can monitor and manage the S Series Node.

HCP S Series Management Console

A web-based interface to the administrative functions of an HCP S Series Node. Using the Management Console, you can monitor and manage the S Series Node.

HCP S Series Node

A highly efficient, highly available, cost-effective storage device that supports very large amounts of data. S Series Nodes can be integrated into HCP systems to serve as additional storage for tiering purposes. The use of commodity hardware in S Series Nodes ensures that the costs of growth and repair remain low.

HCP S10 Node

An HCP S Series Node model that consists of two cooperating server modules and multiple high-density disks in a single enclosure.

Hitachi Content Platform (HCP)

A distributed object-based storage system designed to support large, growing repositories of fixed-content data. HCP provides a single scalable environment that can be used for archiving, business continuity, content depots, disaster recovery, e-discovery, and other services. With its support for multitenancy, HCP securely segregates data among various constituents in a shared infrastructure. Clients can use a variety of industry-standard protocols and various HCP-specific interfaces to access and manipulate objects in an HCP repository.

HS3

See HCP S Series HS3 API.

M

management API

See HCP S Series management API.

Management Console

See HCP S Series Management Console.

management network

One of the networks that can be used for external client access to an HCP S Series Node through the HCP S Series Management Console or management API. The other network available for this purpose is the access network. The management network is not used for data access.

metadata

Information about an object. Metadata is an integral part of the object it describes, thereby making the object self-describing.

N

native drive

A hard disk drive that was previously used in and then removed from the current S Series Node, where the HCP S Series software on the S Series Node has not been upgraded or reinstalled since the drive was removed.

O

object

An exact digital reproduction of data as it existed before it was written to an HCP S Series Node, together with metadata about that data.

P

power and cooling module

An HCP S10 Node component that provides power and cooling functions within the S10 Node enclosure.

R

role

A named collection of permissions that can be associated with an HCP S Series Node user account, where each permission allows the user to perform some specific interaction or set of interactions with the HCP S Series Management Console, management API, or HS3 API.

S

server interconnect network

The network used by the HCP S Series Node server modules to communicate with each other. The two server modules are the only devices on this isolated network.

server module

An HCP S Series Node component that runs the HCP S Series software. The server modules in an S Series Node work cooperatively to handle data access, implement data protection, and generally monitor and manage the S Series Node.

Index

/

/alerts resource
- about 55
- example 87
- input/output 80
- properties 81
- query parameter 86

/buckets resource
- about 56
- example 90
- input/output 88
- properties 89, 92

/buckets/bucket-name resource
- about 57
- example 93
- input/output 91

/buckets/bucket-name/irreparables resource
- about 60
- GET example 96
- HEAD example 97
- input/output 93
- properties 94

/configuration/certificates/system resource
- about 57
- example 100
- input/output 98
- properties 99

/configuration/certificates/system/generate resource
- about 57
- example 103
- input/output 101
- properties 102

/configuration/console resource
- about 58
- example 109
- input/output 104
- properties 105

/configuration/dns resource
- about 58
- example 112
- input/output 110
- properties 111

/configuration/ident resource
- about 60
- example 114
- input/output 113
- properties 113

/configuration/mapi resource
- about 65
- example 121
- input/output 115
- properties 116

/configuration/networks/builtin resource
- about 67
- example 122
- input/output 122
- properties with GET request 122

/configuration/networks/builtin/network-name resource
- about 67
- example 128
- input/output 123
- network names 123
- properties 124

/configuration/protocols resource
- about 68
- example 130
- input/output 129
- property 129

/configuration/protocols/hs3 resource
- about 68
- example 133
- input/output 130
- properties 131

/configuration/security resource
- about 69

example 136
input/output 134
properties 135
/configuration/syslog resource
about 70
example 141
input/output 137
properties 138
/configuration/syslog/test resource
about 70
input/output 142
/configuration/time resource
about 70
example 144
input/output 143
properties 143
/events resource
about 59
example 158
input/output 145
properties 146
query parameters 152
/hardware resource
about 59
example 255
input/output 160
properties 160
/hardware/beacon/enclosure/enclosure-number resource
about 55
example 268
input/output 267
query parameters 267
/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-id resource
about 56
example 269
input/output 268
query parameters 269
/hardware/beacon/server_module/server-module-number resource
about 56
example 270
input/output 270
query parameters 270
/hardware/maintenance resource
about 63
example 272
input/output 271
request body property 271
response body properties 272
/hardware/maintenance/active resource
about 63
example 274
input/output 273
property 274
/hardware/maintenance/history resource
about 63
example 276
input/output 275
property 276
/hardware/maintenance/procedure-id resource
about 63
example 279
input/output 278
properties 279
/hardware/maintenance/procedure-id/cancel resource
about 63
example 280
input/output 280
properties 280
/hardware/maintenance/procedure-id/candidates resource
about 63
example 283
input/output 282
property 283
/hardware/maintenance/procedure-id/complete resource
about 63
example 285
input/output 284
properties 285
/hardware/maintenance/procedure-id/confirm resource
about 64
example 289
input/output 286
request body properties 287
response body properties 289
/hardware/maintenance/procedure-id/perform resource
about 64
example 291
input/output 291
properties 291
/hardware/maintenance/procedure-id/select resource
about 64
example 295

input/output 292
request body properties 293
response body properties 295
/hardware/maintenance/procedure-id/update resource
about 64
example 297
input/output 296
request body property 297
response body properties 297
/hardware/maintenance/procedure-id/verify resource
about 64
example 299
input/output 299
properties 299
/hardware/power/node resource
about 67
example 301
input/output 301
query parameters 301
/hardware/power/server-module-number resource
about 68
example 303
input/output 302
query parameters 302
/metrics/buckets resource
about 65
example 305
input/output 303
properties 304
/metrics/gateways resource
about 65
example 312
input/output 306
properties 307
/metrics/protection resource
about 66
example 314
input/output 313
properties 314
/metrics/system resource
about 66
example 316
input/output 315
properties 315
/system/irreparables resource
about 61
GET example 320
HEAD example 321
input/output 317
properties 318
/system/license resource
about 61
example 324
input/output 322
properties 322
/system/logs/cancel resource
about 62
input/output 325
/system/logs/download resource
about 62
input/output 325
/system/logs/mark resource
about 62
example 326
input/output 326
query parameters 326-327
/system/logs/prepare resource
about 62
example 328
input/output 327
/system/logs/status resource
about 62
example 332
input/output 329
properties 330
/system/misc/settings/network/management/monitor resource
about 66
GET example 334
input/output 333
POST example 335
property 333
query parameter 334
/system/status/full resource
about 69
input/output 336
properties 336
/system/status/health resource
about 69
example 338, 340
input/output 339
properties 339
/system/update/apply resource
about 71
input/output 341
/system/update/history resource
about 71
example 343
input/output 341
properties 342

/system/update/manifest resource
about 71
example 346
input/output 345
properties 345
/system/update/progress resource
about 71
example 349
input/output 347
properties 347
/system/update/restart resource
about 71
input/output 351
/system/update/status resource
about 71
example 353
input/output 351
properties 351
/system/update/upload/license resource
about 71
example 355
input/output 354
properties 354
/system/update/upload/software resource
about 71
example 358
input/output 356
properties 357
/user_accounts resource
about 72
GET example 362
input/output 359
properties 360
PUT example 361
/user_accounts/username resource
about 72
example 368
input/output 363
properties 364
/user_accounts/username/access_key/generate resource
about 73
example 370
input/output 369
properties 369
/versions resource
about 73
GET example 372
input/output 370
POST example 373
properties with GET request 371
properties with POST request 372
query parameter 371

A

access network
about 11-12
configuration, retrieving with management API 128
access/secret keys
about 3
generating with management API 73, 370
adding notes to maintenance procedures with management API 64
administrator role 7
alarms, properties with management API 171
alerts
about 27
properties with management API 82
resource 55
retrieving with management API 55, 87
allow lists 22
applying updates 71
authentication token, generating 51
authentication, management API 51
Authorization header, management API 51

B

beaconing
enclosure example with management API 268
power and cooling module example with management API 269
resources 55
server module example with management API 270
turning on anad off for enclosures with management API 55
turning on and off for power and cooling modules with management API 56
turning on and off for server modules with management API 56
bonded network interfaces, properties with management API 247
bonding mode 14
bonding, access network 12
Boolean values in query parameters, management API 43
bucket names 10
bucket owners
about 10
changing 10

deleting 9
buckets
about 9
checking existence with management API 57
considerations 11
creating with management API 56
deleting 57
maximum number of 10
maximum owned 10
modifying 57
resources 56
retrieving individual with management API 57
retrieving list of with management API 56, 90
retrieving usage statistics with management API 305
usage statistics, properties with management API 305
usage statistics, retrieving with management API 65
bytes under repair 314

C

canceling log download with management API 62
canceling maintenance procedures with management API 63
capacity
licensed 20
usage statistics, retrieving with management API 66
case sensitivity
percent-encoded values 50
query parameters, management API 43
changing
bucket owners 10
domain name 20
passwords 3, 8
usernames 8
checking existence
buckets with management API 57
user accounts with management API 72
checking management API version for support with management API 73
component firmware *See* firmware updates
configuring
HS3 API 22
management API 21
Management Console 20
confirming maintenance procedure actions with management API 64
Content-Type header with management API 43
core hardware properties with management API 232
count property, management API 74
count query parameter, management API 74
creating buckets with management API 56
creating user accounts
considerations 8
management API 72
currents, properties with management API 176

D

data access protocols*See also* HS3 API
resources 68
retrieving list of with management API 68, 130
data drives*See also* drives
data role 8
data types for resource properties 42
database drives*See also* drives
DELETE method for management API 42
deleting buckets
considerations 11
management API 57
deleting user accounts
considerations 8
management API 72
deny lists 22
disable query parameter, system/misc/settings/network/management/monitor resource 334
disabled user accounts 5
disabling management network monitoring with management API 66
disks *See* hard disk drives
DNS servers
about 25
configuration resource 58
configuration, modifying with management API 58
configuration, retrieving with management API 58, 112
domain names
management API resource URLs 48
modifying with management API 60
SSL server certificates 24

downloading internal logs
about 29
managment API 376
downloading internal logs with management API 62
drives*See also* hard disk drives

E

enabling management network monitoring with management API 66
enclosure service component properties with management API 183
enclosures
about 2
alarm properties with management API 171
current properties with management API 176
fan properties with management API 187
hard disk drive properties with management API 210
Larkspur properties with management API 179
power supply properties with management API 191
properties with management API 161, 402
replacing 34
SAS connector properties with management API 198
SAS expander properties with management API 201
slot properties with management API 204
temperature properties with management API 218
voltage properties with management API 224
ending maintenance procedures with management API 63
error response body, management API 44
Ethernet interfaces, properties with management API 250
event log
about 26
event properties with management API 148
resources 59
retrieving with management API 59, 158
security events 28
syslog logging 27
eventsAfter query parameter, /events resource 153
eventsBefore query parameter, /events resource 153
examples, management API
adding notes to maintenance procedure 297
canceling maintenance procedure 280
checking management API version 373
completing maintenance procedure 285
confirming drive action 289
creating user account 361
disabling management network monitoring 335
generating access/secret keys 370
generating SSL server certificate information 103
inserting comment into internal logs 326
powering off individual server module 303
preparing for maintenance procedure 291
replacing hard disk drive 379
restarting both server modules 301
retrieving active maintenance procedure list 274
retrieving alert list 87
retrieving bucket information 93
retrieving bucket irreparable object count 97
retrieving bucket irreparable object list 96
retrieving bucket list 90
retrieving bucket usage statistics 305
retrieving data repair statistics 314
retrieving DNS server configuration 112
retrieving event message list 158
retrieving HS3 API configuration 133
retrieving HS3 usage statistics 312
retrieving license information 324
retrieving log download operation status 332
retrieving maintenance history list 276
retrieving maintenance procedure candidates 283
retrieving management API configuration 121
retrieving Management Console configuration 109
retrieving management network monitoring setting 334
retrieving network information 128
retrieving predefined network list 122
retrieving protocol list 130

retrieving S Series Node brief status 338, 340
retrieving S Series Node hardware information 255
retrieving S Series Node identification 114
retrieving S Series Node irreparable object count 321
retrieving S Series Node irreparable object list 320
retrieving S Series Node usage statistics 316
retrieving security settings 136
retrieving specified maintenanc procedure 279
retrieving SSL server certificate information 100
retrieving supported management API versions 372
retrieving syslog configuration 141
retrieving time settings 144
retrieving update history 343
retrieving update manifest 346
retrieving update operation progress 349
retrieving update status 353
retrieving user account information 368
retrieving user account list 362
selecting target components for maintenance procedure 295
starting log preparation 328
starting maintenance procedure 272
turning beaconing on for enclosure 268
turning beaconing on for power and cooling module 269
turning beaconing on for server module 270
uploading license file 355
uploading software upgrade file 358
verifying maintenance procedure 299
expired passwords 4

F

fans, properties with management API 187
file systems, properties with management API 238
firmware updates
about 30
applying 71
considerations 31
files for 31
uploading with management API 71

G

gateways *See* data access protocols
generating access/secret keys with management API
about 73
example 370
GET method for management API 42

H

hard disk drives*See also* drives
maintaining 33
properties with management API 210, 403
hardware
components 2
maintenance procedures 33, 378
resource 59
retrieving information about with management API 59
S Series Node usage, retrieving information about with management API 255
HCP S Series Management Console *See* Management Console
HCP S Series Node*See also* HCP S10 Node
interfaces 20
security 24
serial number 20
HCP S Series Node identification
about 19
resource 60
retrieving with management API 60, 114
HCP S Series Node license *See* licenses
HCP S Series Node management API *See* management API
HCP S Series Node status
resources 69
retrieving brief with management API 69, 338, 340
retrieving complete with management API 69
HCP S Series operating system*See also* HCP S Series software
HCP S Series software
hotfixes 30
upgrading 30
HCP S10 Node*See also* HCP S Series Node
hardware components 2
powering off 67
restarting 67
HCP systems, S Series objects for 9
HEAD method for management API 42

history
maintenance procedures, retrieving with management API 63
update operations, retrieving with management API 71
Hitachi Content Platform S Series Node *See* HCP S Series Node
Hitachi Content Platform S10 Node *See* HCP S10 Node
hosts files
management API resource URLs 49
hotfixes
about 30
applying 71
considerations 31
files for 31
uploading uploading with management API 71
HS3 API 68
allow/deny lists 22
allowed/denied IP addresses 22
configuration 22
configuration, modifying with management API 68
configuration, retrieving with management API 68, 133
credentials 3
properties with management API 307
retrieving usage statistics with management API 312
usage statistics, retrieving with management API 65
HTTP status codes, management API 413

I

input format for management API 43
inserting comments into internal logs
about 29
management API 62
internal logs
about 29
adding comments to with management API 62
adding comments to, about 29
downloading with management API 62, 376
downloading, about 29
preparing for download with management API 62, 328
resetting with management API 62
resources 62
retrieving download status with management API 62, 332
IP addresses
default for access network 13
default for management network 16
management API resource URLs 48
virtual 12
IP modes
access network 12
management network 15
server interconnect network 18
IPMI
properties with management API 241
sensor properties with management API 235, 243
IPv4 mode
access network 12
management network 15
server interconnect network 18
IPv6 addresses, management API resource URLs 48
IPv6 mode
access network 12
management network 15
irreparable bytes count, retrieving with management API 66
irreparable objects
properties with management API 96, 319
resources 60
retrieving count of in bucket with management API 97
retrieving count of with management API 60-61, 321
retrieving list of in bucket with management API 96
retrieving list of with management API 320
retrieving lists of with management API 60-61
isTruncated property, management API 75

L

Larkspur I/O modules, properties with management API 179
licenses
about 20
about uploading 30
applying 71
considerations for uploading 31
files for 31
resource 61

retrieving information about with management API 61
retrieving with management API 324
uploading with management API 71
login page message;message on login page 21
logs *See* internal logs

M

maintaining
hard disk drives 33
HCP S Series software 30
maintenance procedures
adding notes to management API 297
adding notes to with management API 64
canceling with management API 63, 280
completing with management API 285
confirming actions with management API 64
confirming drive action with management API 289
enclosure properties with management API 402
ending with management API 63
hard disk drive properties with management API 403
hardware 33
IDs 272
listing active with management API 63
listing candidates with management API 63
listing history with management API 63
performing with management API 378
preparing for with management API 291
preparing S10 Node for with management API 64
properties with management API 385
replacing hard disk drives 379
resources 63
retrieving candidates with management API 283
retrieving individual with management API 63
retrieving list of active with management API 274
retrieving list of past with management API 276
retrieving specified with management API 279
selecting components with management API 64
selecting target components with management API 295
slot properties with management API 402
starting with management API 63, 272
target component list properties with management API 389
target component properties with management API 390
verifying with management API 64, 299
major query parameter, /events resource 155
management API
about 37
allow/deny lists 22
allowed/denied IP addresses 22
authentication 51
checking version for support with management API 73, 373
configuration 21
configuration resource 65
configuration, modifying with management API 65
configuration, retrieving with management API 65
error response body 44
HTTP status codes 413
procedures 375
request body format 43
response body format 43
retrieving configuration with management API 121
retrieving supported version with management API 372
supported methods 42
versions resource 73
versions, retrieving supported with managment API 73
what you can do 38
who can use it 41
Management Console
allow/deny lists 22
allowed/denied IP addresses 21
configuration resource 58
configuration, about 20
configuration, modifying with management API 58
configuration, retrieving with management API 58
enclosure replacement procedure 34
hard disk drive maintenance procedures 34
retrieving configuration with management API 109

security 24
session timeout 25
management network 11, 15
enabling/disabling monitoring with management API 66
monitoring resource 66
monitoring setting, retrieving with management API 66
retrieving monitoring setting with management API 334
managing alert lists, query parameters 86
managing event lists, query parameters 152
managing resource lists
count query parameter 74
marker query parameter 75
owner query parameter 77
prefix query parameter 76
query parameters 73
MAPI *See* management API
marker propery, management API 75
marker query parameter, management API 75
marking internal logs
about 29
management API 62
maxEvents query parameter, /events resource 153
maximum transmission unit *See* MTU
maximums
buckets 10
buckets owned 10
user accounts 5
metrics resources 65
modifying buckets
considerations 11
management API 57
modifying DNS server configuration, management API 58
modifying domain names, management API 60
modifying HS3 API configuration, management API 68
modifying management API configuration, management API 65
modifying Management Console configuration, management API 58
modifying networks
considerations 18
management API 67
modifying security settings, management API 69
modifying syslog logging configuration, management API 70
modifying time settings, management API 70
modifying user accounts
considerations 8
management API 72
monitor role 7
MTU
access network 14
management network 17

N

network interfaces
bonded, properties with management API 247
Ethernet, properties with management API 250
properties with management API 247
networks
about 11
access network 11-12
configuration, retrieving with manageement API 67
considerations 18
management network 11, 15
modifying with management API 67
names with management API 123
resources 67
retrieving list of with management API 67
server interconnect network 11, 18

O

objects 9
one-foot CAT6 Ethernet cable *See* server interconnect cables
OS drives*See also* drives
output format for management API 43
owner property, management API 77
owner query parameter, management API 77

P

passwords
automatic expiration 24
changing 3, 8
expired 4
minimum length 24
rules for 6
percent-encoding special characters in resource URLs 50
ping 24
POST method for management API 42

power and cooling modules
about 2
power resources 67
power supplies, properties with management API 191
powering off
S10 Nodes with management API 67
server modules with management API 68
powering on server modules with management API 68
prefix property, management API 77
prefix query parameter, management API 76
preparing internal logs for download with management API 62
preparing S10 Node for maintenance procedures with management API 64
prettyprint query parameter, management API
about 44
error response body 44
properties *See* resource properties
protocols *See* HS3 API
PUT method for management API 42

Q

query parameters, management API
/alerts resource 86
/events resource 152
/hardware/beacon/enclosure/enclosure-number resource 267
/hardware/beacon/enclosure/enclosure-number/power_supply/power-and-cooling-module-ide resource 269
/hardware/beacon/server_module/server-module-number resource 270
/hardware/power/node resource 301
/hardware/power/server-module-number resource 302
/system/logs/mark resource 326-327
/system/misc/settings/network/management/monitor resource 334
/versions resource 371
about 43
count 74
managing resource lists 73
marker 75
owner 77
prefix 76
prettyprint 44
quotation marks with resource URLs 51

R

repair statistics 314
replacing enclosures, about 34
replacing hard disk drives, example 379
request body format, management API 43
request headers, management API
Content-Type 43
X-HCPS-API-VERSION 45
reserved usernames 5
resource identifiers
/alerts 55
/buckets 56
/buckets/bucket-name 57
/buckets/bucket-name/irreparables 60
/configuration/certificates/system 57
/configuration/certificates/system/generate 57
/configuration/console 58
/configuration/dns 58
/configuration/ident 60
/configuration/mapi 65
/configuration/networks/builtin 67
/configuration/networks/builtin/network-name 67
/configuration/protocols 68
/configuration/protocols/hs3 68
/configuration/security 69
/configuration/syslog 70
/configuration/syslog/test 70
/configuration/time 70
/events 59
/hardware 59
/hardware/beacon/enclosure/enclosure-id resource identifier 55
/hardware/beacon/enclosure/enclosure-id/power_supply/power-and-cooling-module-id 56
/hardware/beacon/server_module/server-module-id 56
/hardware/maintenance 63
/hardware/maintenance/active 63
/hardware/maintenance/history 63
/hardware/maintenance/procedure-id 63
/hardware/maintenance/procedure-id/cancel 63
/hardware/maintenance/procedure-id/candidates 63
/hardware/maintenance/procedure-id/complete 63

/hardware/maintenance/procedure-id/confirm 64
/hardware/maintenance/procedure-id/perform 64
/hardware/maintenance/procedure-id/select 64
/hardware/maintenance/procedure-id/update 64
/hardware/maintenance/procedure-id/verify 64
/hardware/power/node 67
/hardware/power/server-module-number 68
/metrics/buckets 65
/metrics/gateways 65
/metrics/protection 66
/metrics/system 66
/system/irreparables 61
/system/license 61
/system/logs/cancel 62
/system/logs/download 62
/system/logs/mark 62
/system/logs/prepare 62
/system/logs/status 62
/system/misc/settings/network/management/monitor 66
/system/status/full 69
/system/status/health 69
/system/update/apply 71
/system/update/history 71
/system/update/manifest 71
/system/update/progress 71
/system/update/restart 71
/system/update/status 71
/system/update/upload/license 71
/system/update/upload/software 71
/user_accounts 72-73
/user_accounts/username 72
/versions 73
about 54
resource properties
/alerts resource 81
/buckets resource 89, 92
/buckets/bucket-name/irreparables resource 94
/configuration/certificates/system resource 99
/configuration/certificates/system/generate resource 102
/configuration/console resource 105
/configuration/dns resource 111
/configuration/ident resource 113
/configuration/mapi resource 116
/configuration/networks/builtin resource 122
/configuration/networks/builtin/network-name resource 124
/configuration/protocols resource 129
/configuration/protocols/hs3 resource 131
/configuration/security resource 135
/configuration/syslog resource 138
/configuration/time resource 143
/events resource 146
/hardware resource 160
/hardware/maintenance resource request body 271
/hardware/maintenance resource response body 272
/hardware/maintenance/active resource 274
/hardware/maintenance/history resource 276
/hardware/maintenance/procedure-id resource 279
/hardware/maintenance/procedure-id/cancel resource 280
/hardware/maintenance/procedure-id/candidates resource 283
/hardware/maintenance/procedure-id/complete resource 285
/hardware/maintenance/procedure-id/confirm resource request bodies 287
/hardware/maintenance/procedure-id/confirm resource response bodies 289
/hardware/maintenance/procedure-id/perform resource 291
/hardware/maintenance/procedure-id/select resource request bodies 293
/hardware/maintenance/procedure-id/select resource response bodies 295
/hardware/maintenance/procedure-id/update resource request body 297
/hardware/maintenance/procedure-id/update resource response bodies 297

/hardware/maintenance/procedure-id/verify resource 299
/metrics/buckets resource 304
/metrics/gateways resource 307
/metrics/protection resource 314
/metrics/system resource 315
/system/irreparables resource 318
/system/license resource 322
/system/logs/status resource 330
/system/misc/settings/network/management/monitor resource 333
/system/status/full resource 336
/system/status/health resource 339
/system/update/history resource 342
/system/update/manifest resource 345
/system/update/progress resource 347
/system/update/status resource 351
/system/update/upload/license resource 354
/system/update/upload/software resource 357
/user_accounts resource 360
/user_accounts/username resource 364
/user_accounts/username/access_key/generate resource 369
/versions resource with GET request 371
/versions resource with POST request 372
about 42
alarms 171
alerts 82
bonded network interface 247
bucket usage statistics 305
currents 176
data types 42
enclosure service components 183
enclosures 161
Ethernet interface 250
events 148
fans 187
file system 238
hard disk drives 210
HS3 307
IPMI 241
IPMI sensors 235, 243
irreparable objects 96, 319
Larkspur I/O modules 179
maintenance procedures 385
mirror set list 244
network interface 247
power supplies 191
SAS connectors 198
SAS expanders 201
server module core hardware 232
server module mirror sets 245
server module peer list 253
server module peers 254
server module statistics 309
server modules 228
slots 204
temperatures 218
update manifests 343
updates 342
usage 80
voltages 224
resource URLs
about 48
case sensitivity 50
length 50
percent-encoding special characters 50
quotation marks with 51
resources
about 41
alerts 55
beaconing 55
buckets 56
data access protocols 68
DNS server configuration 58
event log 59
hardware 59
HCP S Series Node identification 60
internal logs 62
irreparable objects 60
licenses 61
maintenance procedures 63
Management Console configuration 58, 65
management network monitoriing 66
managment API versions 73
metrics 65
networks 67
power 67
S Series Node status 69
security 69
SSL server certificates 57
syslog logging 70
updates 71
user accounts 72
response body format, management API 43
response headers, management API
Server 46
X-HCPS-API-VERSION 45
X-HCPS-ErrorMessage 46
X-HCPS-Irreparable-Count 60-61
X-HCPS-Server-Module-Number 46

X-HCPS-SUPPORTED-API-VERSIONS 45
restarting
HCP S10 Nodes with management API 67
server modules with management API 68
update procedure with management API 71
retrieving
active maintenance procedure list with management API 63
alerts with management API 55
bucket list with management API 56
bucket usage statistics with management API 65
buckets, individual with management API 57
capacity usage statistics with management API 66
DNS server configuration with management API 58
event list with management API 59
hardware information with management API 59
HS3 API configuration with management API 68
HS3 API usage statistics with management API 65
irreparable byte count with management API 66
irreparable object count with management API 60-61
irreparable object lists with management API 60-61
license information with management API 61
log download status with management API 62
maintenance procedure candidate list with management API 63
maintenance procedure history list with management API 63
maintenance procedures, individual with management API 63
management API configuration with management API 65
Management Console configuration with management API 58
management network monitoring setting with management API 66
network configuration with management API 67
network list with management API 67
protocol list with management API 68
S Series Node identification information with management API 60
S Series Node status, brief with management API 69
S Series Node status, complete with management API 69
security settings with management API 69
supported management API versions with management API 73
syslog logging configuration with management API 70
time settings with management API 70
update history list with management API 71
update manifest with management API 71
update progress with management API 71
update status with management API 71
user account list with management API 72
user accounts, individual with management API 72
reusing usernames 6
roles 7

S

S Series Node *See* HCP S Series Node
S10 Node *See* HCP S10 Node
SAS connector properties with management API 198
SAS expander properties with management API 201
SAS hard disk drives 3, *See also* hard disk drives
SATA hard disk drives 3, *See also* hard disk drives
scopeRefs query parameter
alerts resource 87
events resource 156
scopes query parameter
alerts resource 87
events resource 155
scopeSubRefs query parameter
alerts resource 87
events resource 157
secret keys, about 3
security events 28
security role
about 7
last user account with 8

security, HCP S Series Node 24
configuration, retrieving with management API 136
resource 69
settings, modifying with management API 69
settings, retrieving with management API 69
selecting maintenance procedure components with management API 64
sending test message to syslog servers 142
serial number 20
server interconnect cables 3
server interconnect network 11, 18
server modules
about 2
bonded network interfaces with management API 247
core hardwarel properties with management API 232
Ethernet interfaces with management API 250
file systems with management API 238
IPMI information with management API 241
IPMI sensor information with management API 235, 243
mirror set list property with management API 244
mirror set properties with management API 245
network interfaces with management API 247
peer list property with management API 253
peer properties with management API 254
powering off individual with management API 303
powering off with management API 68
powering on with management API 68
properties with management API 228
restarting both with management API 301
restarting with management API 68
statistical properties with management API 309
update preparation progress, retrieving with management API 348
update progress, retrieving with management API 349
Server response header, management API 46
service role 8
severities query parameter, /alerts resource 86
severity query parameter, /events resource 154
slots
enclosure slot properties with management API 204
maintenance procedure slot properties with management API 402
software *See* HCP S Series software
software updates
about 30
applying 71
considerations 31
files for 31
uploading with management API 71
special characters in resource URLs, percent-encoding 50
speed/duplex setting
access network 14
management network 17
SSDs*See also* OS drives
SSH access to server modules 24
SSL server certificates
about 24
management API requests 49
resources 57
retrieving with management API 100
starting maintenance procedures with management API 63
statistics
bucket usage, retrieving with management API 65
capacity usage, retrieving with management API 66
HS3 API usage, retrieving with management API 65
irreparable byte count, retrieving with management API 66
S Series Node usage, retrieving with management API 316
subnets
access network 13
management network 15
server interconnect network 18-19
supported methods for management API 42
switch configuration
access network 12
management network 15
syslog logging 27
configuration, modifying with management API 70

configuration, retrieving with management API 70, 141
resources 70
sending test message 142
testing with management API 70

T

target component list properties 389
target component properties 390
temperature properties with management API 218
testing syslog logging with management API 70, 142
time servers 26
configuration, modifying with management API 70
configuration, retrieving with management API 70, 144
resource 70
turning beaconing on and off
enclosures with management API 55
power and cooling modules with management API 56
server modules with management API 56

U

update files
about 31
applying 71
uploading with management API 71
update manifests
properties with management API 343
retrieving with management API 346
updates
properties with management API 342
retrieving history with management API 343
retrieving progress of current with management API 349
retrieving status of with management API 353
uploading license files with management API 355
uploading software upgrade files with management API 358
updating software and firmware
about 30
apply updates 71
considerations 31
listing history with management API 71
resources 71
restarting with management API 71
retrieving manifest with management API 71
retrieving progress with management API 71
retrieving status with management API 71
update files 31
upload update files with management API 71
URLs for management API resources
about 42, 48
case sensitivity 50
length 50
percent-encoding special characters 50
quotation marks with 51
user accounts
about 3
access/secret keys 3
access/secret keys, about 3
automatically disabled 25
checking existence with management API 72
considerations 8
creating with management API 72, 361
deleting with management API 72
disabled 5
generating access/secret keys with management API 73
maximum number of 5
modifying with management API 72
passwords 3
properties 4
resources 72
retrieving individual with management API 72
retrieving list of with management API 72, 362
roles 7
usernames
changing 8
rules for 5

V

verifying maintenance procedures with management API 64
version query parameter, versions resource 371
virtual IP addresses 12
VLAN IDs
access network 14

management network 17
voltage properties with management API 224

X

X-HCPS-API-VERSION request header 45
X-HCPS-API-VERSION response header 45
X-HCPS-ErrorMessage response header 46
X-HCPS-Irreparable-Count response header 60-61
X-HCPS-Server-Module-Number response header, management API 46
X-HCPS-SUPPORTED-API-VERSIONS response header 45

Hitachi Data Systems

Corporate Headquarters
2845 Lafayette Street
Santa Clara, California 95050-2627
U.S.A.
www.hds.com

Regional Contact Information

Americas
+1 408 970 1000
info@hds.com

Europe, Middle East, and Africa
+44 (0) 1753 618000
info.emea@hds.com

Asia Pacific
+852 3189 7900
hds.marketing.apac@hds.com

MK-HCPS004-01