



Hitachi Content Platform

HCP Management API Reference

© 2010–2015 Hitachi Data Systems Corporation. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for any purpose without the express written permission of Hitachi Data Systems Corporation (hereinafter referred to as “Hitachi Data Systems”).

Hitachi Data Systems reserves the right to make changes to this document at any time without notice and assumes no responsibility for its use. This document contains the most current information available at the time of publication. When new and/or revised information becomes available, this entire document will be updated and distributed to all registered users.

Some of the features described in this document may not be currently available. Refer to the most recent product announcement or contact Hitachi Data Systems for information about feature and product availability.

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

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

- a) Acquiring the relevant consents as may be required under local privacy laws or otherwise from employees and other individuals to access relevant data; and
- b) Ensuring that data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Hitachi is a registered trademark of Hitachi, Ltd., in the United States and other countries. Hitachi Data Systems is a registered trademark and service mark of Hitachi, Ltd., in the United States and other countries.

Archivas, Essential NAS Platform, HiCommand, Hi-Track, ShadowImage, Tagmaserve, Tagmasoft, Tagmasolve, Tagmastore, TrueCopy, Universal Star Network, and Universal Storage Platform are registered trademarks of Hitachi Data Systems Corporation.

AIX, AS/400, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, ESCON, FICON, FlashCopy, IBM, Lotus, MVS, OS/390, RS6000, S/390, System z9, System z10, Tivoli, VM/ESA, z/OS, z9, z10, zSeries, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

All other trademarks, service marks, and company names in this document or web site are properties of their respective owners.

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

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

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



Contents

Preface	xi
Intended audiencexi
Product versionxi
Syntax notation	xii
Related documents.	xii
Getting help.	xv
Comments	xv
1 Introduction to the HCP management API	1
What you can do with the management API	2
Who can use the management API	4
Resources and properties	6
Supported methods	7
Input and output formats	8
HTTP Content-Type and Accept headers.	8
XML	9
JSON.	11
CSV.	12
WADL	13
Query parameters	15
verbose	15
prettyprint	16
Request-specific query parameters.	16
HCP-specific response headers	18
Enabling the management API in the Management Consoles	19
2 Access and authentication	21
URLs for HCP access through the management API	22
Using an IP address in a URL.	23

Using a hosts file	24
URL considerations	26
Authentication	28
Authentication token	28
Authorization header	29
3 Resources	31
Tenant resources	33
Example: Creating an HCP tenant	39
Example: Setting Tenant Management Console security for a tenant	41
Example: Generating a chargeback report for a tenant	43
Namespace resources	44
Example: Creating an HCP namespace	52
Example: Changing the compliance settings for an HCP namespace	54
Example: Configuring the HTTP protocol for an HCP namespace	55
Network resources	57
Example: Enabling advanced downstream DNS configuration	57
Example: Checking the advanced DNS configuration	58
Retention class resources	60
Example: Creating a retention class	62
Example: Retrieving a list of retention classes	63
Content class resources	64
Example: Creating a content class	66
Example: Retrieving a list of content classes	68
User account resources	70
Example: Creating a user account	72
Example: Changing the roles associated with a user account	73
Example: Changing the data access permissions associated with a user account	75
Example: Resetting the security user passwords for a tenant	76
Group account resources	77
Example: Creating a group account	79
Example: Retrieving group accounts	81
Replication resources	82
Example: Creating a replication link	88
Example: Retrieving a list of eligible local candidates	90
Example: Adding an HCP tenant to a replication link	91
Example: Setting the schedule for a replication link	92
Example: Failing over a replication link	94
Licenses resources	95
Example: Retrieving an active storage license list	95
Example: Uploading a new license	97

Paging, sorting, and filtering	98
Paging through the resource lists	98
Sorting resource lists	100
Filtering resource lists	101
4 Data types	103
Common property values	104
Passwords	104
Descriptions	105
Boolean values	105
Permission lists	105
availableServicePlan	106
availableServicePlan data type properties	106
availableServicePlan data type example	107
chargebackData	107
chargebackData data type properties	108
chargebackData data type example	111
chargebackReport	112
About chargeback reports	112
chargebackReport data type property	113
chargebackReport data type example	114
Query parameters for generating chargeback reports	114
Sample chargeback report	117
cifsProtocol	119
cifsProtocol data type properties	119
cifsProtocol data type example	120
complianceSettings	121
complianceSettings data type properties	121
complianceSettings data type example	123
connection	123
connection data type properties	124
connection data type example	127
consoleSecurity	127
consoleSecurity data type properties	128
consoleSecurity data type example	129
contactInfo	130
contactInfo data type properties	130
contactInfo data type example	131
content	132
content data type properties	132
content data type example	133
contentClass	133

contentClass data type properties	133
contentClass data type example	134
contentProperty	136
contentProperty data type properties	136
contentProperty data type example	140
contentProperties	140
contentProperties data type property	140
contentProperties data type example	140
customMetadataIndexingSettings	142
customMetadataIndexingSettings data type properties	142
customMetadataIndexingSettings data type example	143
dataAccessPermissions	144
dataAccessPermissions data type property	144
dataAccessPermissions data type example	144
emailNotification	145
emailNotification data type properties	145
emailNotification data type example	146
emailTemplate	147
emailTemplate data type properties	147
Email template variables	148
Email template defaults	149
emailTemplate data type example	149
failoverSettings	150
failoverSettings data type properties	150
failoverSettings data type example	150
groupAccount	151
groupAccount data type properties	151
groupAccount data type example	153
Query parameter for resetting the security group	153
httpProtocol	154
httpProtocol data type properties	155
httpProtocol data type example	160
ipSettings	160
ipSettings data type properties	161
ipSettings data type example	163
License	163
License data type properties	164
License data type example	164
Licenses	164
Licenses data type properties	165
Licenses data type example	165
Query parameter for retrieving a list of licenses	165
link	166
link data type properties	166

link data type example	171
Query parameters for replication link actions.	172
local (data type for replication link failoverSettings local property)	173
Replication link failoverSettings local data type properties	174
Replication link failoverSettings local data type example.	176
local (data type for replication link schedule local property)	176
Replication link schedule local data type properties	177
Replication link schedule local data type example	178
namespace	178
namespace data type properties	179
namespace data type example.	197
Query parameter for restarting indexing	198
namespaceDefaults	198
namespaceDefaults data type properties.	199
namespaceDefaults data type example.	202
namespacePermission.	203
namespacePermission data type properties	203
namespacePermission data type example	204
networkSettings	205
networkSettings properties	205
networkSettings example	205
nfsProtocol	206
nfsProtocol data type properties.	206
nfsProtocol data type example.	206
protocols	207
protocols data type properties	207
protocols data type example	208
recipient	208
recipient data type properties	209
recipient data type example.	211
recipients	211
recipients data type properties.	212
recipients data type example.	213
remote (data type for replication link failoverSettings remote property)	213
Replication link failoverSettings remote data type properties.	214
Replication link failoverSettings remote data type example	215
remote (data type for replication link schedule remote property)	216
Replication link schedule remote data type properties	217
Replication link schedule remote data type example.	219
replicationCollisionSettings	219
replicationCollisionSettings data type properties	220
replicationCollisionSettings data type example.	221
replicationService	221

replicationService data type properties	222
replicationService data type example	224
Query parameters for replication service actions	224
retentionClass	225
retentionClass data type properties	225
retentionClass data type example.	226
schedule	227
schedule data type properties	227
schedule data type example	227
searchSecurity	228
searchSecurity data type property	228
searchSecurity data type example	229
smtpProtocol	229
smtpProtocol data type properties	229
smtpProtocol data type example	231
statistics (data type for replication link statistics property)	231
Replication link statistics data type properties	232
Replication link statistics data type example	240
statistics (data type for tenant and namespace statistics resources)	240
Tenant and namespace statistics data type properties	240
Tenant and namespace statistics data type example	242
tenant (data type for replication link content tenant resource)	242
Replication link content tenant data type properties.	243
Replication link content tenant data type example	246
Query parameters for replication link content tenant actions.	247
tenant (data type for tenant resource)	247
tenant data type properties	248
tenant data type example	262
Query parameters for creating tenants	263
HCP tenant query parameters	263
Default tenant query parameters	265
transition.	265
transition data type properties	265
transition data type example	268
userAccount.	268
userAccount data type properties.	269
userAccount data type example	273
Query parameter for setting user account passwords.	273
Query parameter for resetting security user passwords	273
versioningSettings	274
versioningSettings data type properties	274

versioningSettings data type example.	277
Specifying retention values	277
Specifying a special value	277
Specifying an offset	277
Specifying a retention class	278
Specifying a fixed date	279
5 Usage considerations	281
Choosing an access method	282
Generating templates for resource creation.	283
Modifying resources	283
A HTTP return codes	285
B Sample Java application	289
What the application does.	290
Assumptions	290
Required libraries	291
Input JSON files	291
JSON file for creating the HCP tenant	292
JSON file for modifying the initial user account	292
JSON file for creating the user account with the compliance and monitor roles	293
JSON file for configuring the Tenant Management Console	293
JSON file for modifying the tenant	294
JSON files for creating the namespaces	294
JSON for modifying a namespace.	295
JSON file for configuring the HTTP protocol	296
JSON file for creating the user account with no roles	296
JSON file for granting data access permissions to the user account	297
JSON file for creating the retention class.	297
JSON file for creating the replication link.	298
Java application	299
C Management API XML schema	311
Glossary	313
Index	329



Preface

This book contains the information you need to use the **Hitachi Content Platform (HCP)** management API. This RESTful HTTP API enables you to programmatically create and manage tenants and namespaces and configure, monitor, and manage replication. The book explains how to use the API to access an HCP system, specify resources, and update and retrieve resource properties.



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

Intended audience

This book is intended for people who want to programmatically create and manage tenants and namespaces and configure, monitor, and manage replication. This audience includes:

- HCP system administrators and monitors
- HCP tenant administrators and monitors
- HCP application developers, when taking on administrative responsibilities or developing administrative utilities

This book assumes you are familiar with HCP concepts and functionality. It also assumes you're familiar with the HTTP protocol.

Product version

This book applies to release 7.1.1 of HCP.

Syntax notation

The table below describes the conventions used for the syntax of commands, expressions, URLs, and object names 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: password=password You enter: password=start123
<i>italics</i>	Replace with a value of the indicated type	
	Vertical bar — Choose one of the elements on either side of the bar, but not both	This book shows: enterpriseMode=(true false) You enter: enterpriseMode=true or: enterpriseMode=false
()	Parentheses — Include exactly one of the elements between the parentheses	

Related documents

The following documents contain additional information about Hitachi Content Platform:

- *Administering HCP* — This book explains how to use an HCP system to monitor and manage a digital object repository. It discusses the capabilities of the system, as well as its hardware and software components. The book presents both the concepts and instructions you need to configure the system, including creating the tenants that administer access to the repository. It also covers the processes that maintain the integrity and security of the repository contents.
- *Managing a Tenant and Its Namespaces* — This book contains complete information for managing the HCP tenants and namespaces created in an HCP system. It provides instructions for creating namespaces, setting up user accounts, configuring the protocols that allow access to namespaces, managing search and indexing, and downloading installation files for HCP Data Migrator. It also explains how to work with retention classes and the privileged delete functionality.

- *Managing the Default Tenant and Namespace* — This book contains complete information for managing the default tenant and namespace in an HCP system. It provides instructions for changing tenant and namespace settings, configuring the protocols that allow access to the namespace, managing search and indexing, and downloading installation files for HCP Data Migrator. It also explains how to work with retention classes and the privileged delete functionality.
- *Replicating Tenants and Namespaces* — This book covers all aspects of tenant and namespace replication. Replication is the process of keeping selected tenants and namespaces in two or more HCP systems in sync with each other to ensure data availability and enable disaster recovery. The book describes how replication works, contains instructions for working with replication links, and explains how to manage and monitor the replication process.
- *Using a Namespace* — This book describes the properties of objects in HCP namespaces. It provides instructions for accessing namespaces by using the HTTP, WebDAV, CIFS, and NFS protocols for the purpose of storing, retrieving, and deleting objects, as well as changing object metadata such as retention and shred settings. It also explains how to manage namespace content and view namespace information in the Namespace Browser.
- *Using the HCP HS3 API* — This book contains the information you need to use the HCP HS3 API. This S3™-compatible, RESTful, HTTP-based API enables you to work with buckets and objects in HCP. The book introduces the HCP concepts you need to understand in order to use HS3 effectively and contains instructions and examples for each of the bucket and object operations you can perform with HS3.
- *Using the HCP OpenStack Swift API* — This book contains the information you need to use the HCP HSwift API. This OpenStack Swift, RESTful, HTTP-based API enables you to work with containers and objects in HCP. The book introduces the HCP concepts you need to understand in order to use HSwift effectively and contains instructions and examples for each of the container and object operations you can perform with HSwift.

- *Using the Default Namespace* — This book describes the file system HCP uses to present the contents of the default namespace. It provides instructions for accessing the namespace by using the HCP-supported protocols for the purpose of storing, retrieving, and deleting objects, as well as changing object metadata such as retention and shred settings.
- *HCP Metadata Query API Reference* — This book describes the HCP metadata query API. This RESTful HTTP API enables you to query namespaces for objects that satisfy criteria you specify. The book explains how to construct and perform queries and describes query results. It also contains several examples, which you can use as models for your own queries.
- *Searching Namespaces* — This book describes the HCP Search Console (also called the Metadata Query Engine Console). It explains how to use the Console to search namespaces for objects that satisfy criteria you specify. It also explains how to manage and manipulate queries and search results. The book contains many examples, which you can use as models for your own searches.
- *Using HCP Data Migrator* — This book contains the information you need to install and use HCP Data Migrator (HCP-DM), a utility that works with HCP. This utility enables you to copy data between local file systems, namespaces in HCP, and earlier HCAP archives. It also supports bulk delete operations and bulk operations to change object metadata. Additionally, it supports associating custom metadata and ACLs with individual objects. The book describes both the interactive window-based interface and the set of command-line tools included in HCP-DM.
- *Installing an HCP System* — This book provides the information you need to install the software for a new HCP system. It explains what you need to know to successfully configure the system and contains step-by-step instructions for the installation procedure.
- *Deploying an HCP-VM System* — This book contains all the information you need to install and configure an HCP-VM system. The book also includes requirements and guidelines for configuring the VMWare® environment in which the system is installed.
- *Third-Party Licenses and Copyrights* — This book contains copyright and license information for third-party software distributed with or embedded in HCP.

- *HCP-DM Third-Party Licenses and Copyrights* — This book contains copyright and license information for third-party software distributed with or embedded in HCP Data Migrator.
- *Installing an HCP SAIN System — Final On-site Setup* — This book contains instructions for deploying an assembled and configured single-rack HCP SAIN system at a customer site. It explains how to make the necessary physical connections and reconfigure the system for the customer computing environment. It also contains instructions for configuring Hi-Track® Monitor to monitor the nodes in an HCP system.
- *Installing an HCP RAIN System — Final On-site Setup* — This book contains instructions for deploying an assembled and configured HCP RAIN system at a customer site. It explains how to make the necessary physical connections and reconfigure the system for the customer computing environment. The book also provides instructions for assembling the components of an HCP RAIN system that was ordered without a rack and for configuring Hi-Track Monitor to monitor the nodes in an HCP system.

Getting help

The Hitachi Data Systems® customer support staff is available 24 hours a day, seven days a week. If you need technical support, call:

- United States: (800) 446-0744
- Outside the United States: (858) 547-4526



Note: If you purchased HCP 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, number, and revision, and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Data Systems.

Thank you!

Introduction to the HCP management API

The **Hitachi Content Platform (HCP)** management API is a RESTful HTTP interface to a subset of the administrative functions of an HCP system. Using this API, you can manage tenants, namespaces, retention classes, content classes, tenant-level user and group accounts, and replication of HCP tenants and namespaces and default-namespace directories.

Each entity you can manage is referred to as a resource. Each resource has properties that provide information about it.

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 formats supported by the management API
- Introduces query parameters that can be used with resource requests
- Describes HCP-specific headers that are returned in response to resource requests
- Provides instructions for enabling the management API

To learn about tenants, namespaces, retention classes, content classes, user and group accounts, and replication, see *Administering HCP*, *Managing a Tenant and Its Namespaces*, *Managing the Default Tenant and Namespaces*, and *Replicating Tenants and Namespaces*.



Notes:

- Most of the examples in this book use cURL and Python with PycURL, a Python interface that uses the libcurl library. cURL and PycURL are both freely available open-source software. You can download them from <http://curl.haxx.se>.
 - In version 7.12.1 of PycURL, the **PUT** method was deprecated and replaced with **UPLOAD**. The Python examples in this book show **UPLOAD** but work equally well with **PUT**.
-

What you can do with the management API

The HCP management API lets you work with tenants, namespaces, retention classes, content classes, tenant-level user and group accounts, and replication.

For tenants, you can:

- Create or modify HCP tenants or the default tenant
- Delete HCP tenants
- Retrieve information about a single tenant
- Retrieve a list of all tenants
- Set default properties for namespaces created for a tenant
- Retrieve statistics about the content of the namespaces owned by a tenant
- Generate a chargeback report for a tenant

For namespaces, you can:

- Create or modify HCP namespaces or the default namespace
- Delete HCP namespaces

- Retrieve information about a single namespace
- Retrieve a list of all the namespaces owned by a tenant
- Retrieve statistics about the content of a namespace
- Generate a chargeback report for a namespace
- Reset the indexing checkpoint for a namespace

For retention classes, you can:

- Create, modify, and delete retention classes for a namespace
- Retrieve information about a single retention class
- Retrieve a list of all the retention classes defined for a namespace

For content classes, you can:

- Create, modify, and delete content classes for a tenant
- Retrieve information about a single content class
- Retrieve a list of all the content classes defined for a tenant

For tenant-level user accounts, you can:

- Create, modify, and delete user accounts
- Retrieve information about a single user account
- Retrieve a list of all the user accounts defined for a tenant
- Reset the passwords of all locally authenticated user accounts with the security role

For tenant-level group accounts, you can:

- Create, modify, and delete group accounts
- Retrieve information about a single group account
- Retrieve a list of all the group accounts defined for a tenant

- Create a new group account with the security role or give the security role to an existing group account

For replication, you can:

- Create, modify, and delete replication links
- Specify link content
- Retrieve a list of all links in which the HCP participates
- Monitor replication progress at the link level and the tenant level
- Schedule, suspend, and resume activity on links
- Pause and resume replication of individual tenants
- Shut down and reestablish all links
- Fail over, restore, and failback links
- Set the replication network
- Enable DNS failover
- Enable automatic sharing of domains and SSL server certificates
- Specify whether tenant and namespace replication status is displayed in the Tenant Management Console for HCP tenants

For details on the activities supported by the management API, see [Chapter 3, “Resources,”](#) on page 31.

Who can use the management API

To use the HCP management API, you need either a system-level or tenant-level user account that's defined in HCP. If HCP is configured to support Windows® Active Directory® (AD), applications can also use recognized AD user accounts to access HCP through the management API. **A recognized AD user account** is an AD user account for a user that belongs to one or more AD groups for which corresponding group accounts are defined in HCP.

What you can do with the API depends on:

- The level of account you're using
- The roles associated with the account (or applicable group accounts)
- For tenant-level accounts, whether the account (or applicable group accounts) has the allow namespace management property

The permissions granted by each role have the same effect with the management API as they do in the System Management and Tenant Management Consoles. For example, with a system-level user account that includes the administrator role, you can create, modify, and delete tenants and configure the default tenant and namespace. With a system-level user account that includes only the monitor role, you can only retrieve information about these entities.

Similarly, with a tenant-level user account that includes the administrator role, you can create, modify, and delete namespaces. With a tenant-level user account that includes only the monitor role, you can only retrieve information about these entities.

An HCP tenant can grant system-level users administrative access to itself. This enables users with system-level user accounts to perform the activities allowed by the tenant-level roles that correspond to their system-level roles.

If you have only the allow namespace management property and no roles, the activities you can perform with the HCP management API are limited to creating namespaces, listing and deleting namespaces you own, and viewing and modifying the versioning status of namespaces you own.

For you to use the management API with a system-level user account, the API must be enabled in the System Management Console. For you to use the API with a tenant-level user account, the API must be enabled in both the System Management Console and the applicable Tenant Management Console.

For more information on:

- System-level user accounts, see *Administering HCP*
- Tenant-level user accounts, see *Managing a Tenant and Its Namespaces*

- Enabling the management API, see [“Enabling the management API in the Management Consoles”](#) on page 19

For details on the user accounts required for management API activities, see [Chapter 3, “Resources,”](#) on page 31.

Resources and properties

Each entity that you can manage independently in the HCP management API is called a **resource**. Examples of resources are tenants and namespaces.

Resources have properties. The values of these properties define the resource. For example, tenant properties include the tenant name, description, hard quota and soft quota.

Some properties are treated as resources in their own right. For example, the Tenant Management Console configuration is a property of a tenant, but it is treated as a resource.

To identify a resource, you use a URL. For example, this URL identifies the tenant resource named Finance in the HCP system named hcp.example.com:

```
https://admin.hcp.example.com:9090/mapi/tenants/finance
```

You also use URLs to identify lists of resources. For example, this URL identifies the list of namespaces owned by the Finance tenant:

```
https://admin.hcp.example.com:9090/mapi/tenants/finance/namespaces
```

Each URL that identifies a resource or list of resources has a data type. For a list of resources, the data type is list. For an individual resource, the data type is a named unordered set of properties. For example, the data type for the retention class resource is retentionClass. The properties included in this data type are name, value, description, and allowDisposition.

Properties also have data types. The data type of a property can be string, integer, long, Boolean, or list, or it can be another named unordered set of properties. For example, the name property for a tenant resource has a data type of string. The ipSettings property for a Tenant Management Console configuration resource has a data type of ipSettings.

For more information on:

- HCP management API resources, see [Chapter 3, "Resources,"](#) on page 31
- Properties and data types, see [Chapter 4, "Data types,"](#) on page 103
- Resource URLs, see ["URLs for HCP access through the management API"](#) on page 22

Supported methods

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

Method	Description
PUT	Creates a resource. When creating a resource, you need to supply values for all properties that do not have default values. If properties with default values are omitted from the request body, those values are applied to the resource.
GET	Retrieves information about an individual resource or retrieves a list of resources of a given type.
HEAD	Performs a GET but does not return the response body. You can use a HEAD request to check whether a particular resource exists.
POST	Modifies a resource. When modifying a resource, you need to supply values only for the properties whose values you want to change. If properties are omitted from the request body, they remain unchanged.
DELETE	Deletes a resource.
OPTIONS	Describes the methods supported by a given resource.

Each request you submit to the management API can work on only one resource. So, for example, you cannot use a single **PUT** request to create two tenants.

Input and output formats

When you create or modify a resource through the HCP management API, you use XML or JSON to specify the resource properties. When you request information about resources, you can ask for the response to be returned in XML format or in JSON format. For one resource, `chargebackReport`, you can also ask for the response to be returned in CSV format.

The response to an **OPTIONS** request is always returned as Web Application Description Language (WADL). WADL is an XML-based description language for HTTP-based web applications.

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.

HTTP Content-Type and Accept headers

With a **PUT** or **POST** request, you use the HTTP **Content-Type** request header to specify the format in which you're providing the request body.

With a **GET** request, you can use the HTTP **Accept** request header to specify the format for the response body. If you omit this header, the API returns the response body in XML format.

In a Content-Type or Accept header, you specify the input or output format as an Internet media type:

- For XML, the Internet media type is **application/xml**.
- For JSON, the Internet media type is **application/json**.
- For JSON with callback, the Internet media type is **application/javascript**.
- For CSV, the Internet media type is **text/csv**.

You don't need to specify a Internet media type in an **OPTIONS** request. If you do specify one, it is ignored.

With `cURL`, you use the `-H` option to specify an HTTP header. So, for example, to specify that a request body uses XML, you would include this in the **curl** command:

```
-H "Content-Type: application/xml"
```


In Python with PycURL, you do this with the **HTTPHEADER** option. For example:

```
curl_setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml"])
```

HTTP headers and Internet media types are not case sensitive.

XML

In an XML request or response body:

- Property names are element names. Property values are element values. For example, the element that corresponds to the `softQuota` property with a value of 85 is:

```
<softQuota>85</softQuota>
```

- The name of the root element for a request that involves a single resource is the data type of that resource. For example, for this URL, which identifies a single tenant named Finance, the root element is **tenant**:

```
https://admin.hcp.example.com:9090/mapi/tenants/finance
```

- The name of the root element for a request for a list of resources is the term used to identify those resources in the URL. For example, for this URL, the root element is **tenants**:

```
https://admin.hcp.example.com:9090/mapi/tenants
```

- In a list of resources, each resource is the value of an element whose name is the name of the property used to identify the resource. For example, the response body for a request for the tenants defined in an HCP system might include this:

```
<tenants>
  <name>Default</name>
  <name>Finance</name>
  <name>HR</name>
  <name>Sales</name>
  <name>Marketing</name>
</tenants>
```

Here's a request for complete information about the Finance tenant to be returned in XML format:

```
curl -k -i -H "Accept: application/xml"
-H "Authorization: HCP m9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants/finance?verbose=true
&prettyprint"
```

Assuming that the tenant has granted system-level users administrative access to itself, here's the XML response body you get when you make the request using a system-level user account that includes the administrator role:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tenant>
  <administrationAllowed>true</administrationAllowed>
  <authenticationTypes>
    <authenticationType>LOCAL</authenticationType>
    <authenticationType>EXTERNAL</authenticationType>
  </authenticationTypes>
  <complianceConfigurationEnabled>true</complianceConfigurationEnabled>
  <creationTime>2012-02-09T09:11:17-0500</creationTime>
  <dataNetwork>net127</dataNetwork>
  <fullyQualifiedName>finance.hcp.example.com</fullyQualifiedName>
  <hardQuota>100 GB</hardQuota>
  <managementNetwork>net004</managementNetwork>
  <maxNamespacesPerUser>5</maxNamespacesPerUser>
  <name>Finance</name>
  <namespaceQuota>5</namespaceQuota>
  <replicationConfigurationEnabled>true</replicationConfigurationEnabled>
  <snmpLoggingEnabled>>false</snmpLoggingEnabled>
  <searchConfigurationEnabled>true</searchConfigurationEnabled>
  <servicePlan>Short-Term-Activity</servicePlan>
  <servicePlanSelectionEnabled>>false</servicePlanSelectionEnabled>
  <softQuota>90</softQuota>
  <syslogLoggingEnabled>true</syslogLoggingEnabled>
  <systemVisibleDescription>Created for the Finance department at Example
    Company by P.D. Grey on 2/9/2012.</systemVisibleDescription>
  <tags>
    <tag>Example Company</tag>
    <tag>pdgrey</tag>
  </tags>
  <tenantVisibleDescription>Please see Lee Green for any questions about this
    tenant and its namespaces.</tenantVisibleDescription>
  <id>4420f62f-3f63-43ab-a3cd-0bcf1f399daf</id>
  <versioningConfigurationEnabled>true</versioningConfigurationEnabled>
</tenant>
```

For the XML schema used by the management API, see [Appendix C, "Management API XML schema."](#) on page 311.

JSON

In a JSON request or response body:

- Properties are name/value pairs. For example, the name/value pair that corresponds to the softQuota property with a value of 85 is:

```
"softQuota": "85"
```

- A list of resources is represented by a name/value pair, where the name is the name of the property used to identify each resource and the value is a comma-separated list of the resource identifiers. For example, the response body for a request to list the tenants defined in an HCP system might look like this:

```
{
  "name" : [ "Default", "Finance", "HR", "Sales", "Marketing" ]
}
```

Here's a request for complete information about the Finance tenant to be returned in JSON format:

```
curl -k -i -H "Accept: application/json"
-H "Authorization: HCP m9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants/finance?verbose=true
&prettyprint"
```

Assuming that the tenant has granted system-level users administrative access to itself, here's the JSON response body you get when you make the request using a system-level user account that includes the administrator role:

```
{
  "administrationAllowed" : true,
  "authenticationTypes" : {
    "authenticationType" : [ "LOCAL", "EXTERNAL" ]
  },
  "complianceConfigurationEnabled" : true,
  "creationTime" : "2012-02-09T09:11:17-0500",
  "dataNetwork" : "net127",
  "fullyQualifiedName" : "finance.hcp.example.com",
  "hardQuota" : "100.00 GB",
  "managementNetwork" : "net004",
  "maxNamespacesPerUser" : 5,
```

```

"name" : "Finance",
"namespaceQuota" : "5",
"replicationConfigurationEnabled" : true,
"snmpLoggingEnabled" : false,
"searchConfigurationEnabled" : true,
"servicePlan" : "Short-Term-Activity",
"servicePlanSelectionEnabled" : false,
"softQuota" : 90,
"syslogLoggingEnabled" : true,
"systemVisibleDescription" : "Created for the Finance department at Example
    Company by P.D. Grey on 2/9/2012.",
"tags" : {
    "tag" : [ "Example Company", "pdgrey" ]
},
"tenantVisibleDescription" : "Please see Lee Green for any questions about this
    tenant and its namespaces.",
"id" : "4420f62f-3f63-43ab-a3cd-0bcf1f399daf",
"versioningConfigurationEnabled" : true
}

```

CSV

In a CSV response body (only for a **GET** of a `chargebackReport` resource), the name of each reported property for the resource is a field in the first line. Property values are fields in the subsequent lines.

Here's a request for the `chargebackReport` resource for the Accounts-
Receivable namespace to be returned in CSV format:

```

curl -k -i -H "Accept: text/csv"
-H "Authorization: HCP VVNFUK5BTUU=:319f4d26e3c536b5dd871bb2c52e3178"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces/
accounts-receivable/chargebackReport?start=2012-02-18T13:00:00-0500
&end=2012-02-18T13:59:59-0500&granularity=hour"

```

Here's the CSV response body:

```

systemName,tenantName,namespaceName,startTime,endTime,objectCount,
ingestedVolume,storageCapacityUsed,bytesIn,bytesOut,reads,writes,deletes,
deleted,valid
hcp.example.com,Finance,Accounts-Receivable,2012-02-18T13:00:00-0500,1240,
2012-02-18T13:59:59-0500,1240,692224,716720,2905,561,1,5,0,false,true

```

WADL

The response body for an **OPTIONS** request is always returned as WADL. The HTTP response headers include **Allow**, which lists the supported methods for the resource.

Here's a request for the methods you can use with the user accounts resource:

```
curl -k -iX OPTIONS
-H "Authorization: HCP m9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants/finance/userAccounts
?prettyprint"
```

Here are the response headers:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.sun.wadl+xml
Allow: OPTIONS,HEAD,POST,GET,PUT
X-HCP-SoftwareVersion: 6.0.1.64
Content-Length: 3575
```

Here's the WADL response body:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application xmlns="http://research.sun.com/wadl/2006/10">
  <doc xmlns:jersey="http://jersey.dev.java.net/" jersey:generatedBy="Jersey: 1.1.5
01/20/2010 04:04 PM"/>
  <resources base="https://admin.hcp.example.com:9090/mapi/">
    <resource path="tenants/finance/userAccounts">
      <method name="PUT" id="createUserAccount">
        <request>
          <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="password"/>
          <representation mediaType="application/xml"/>
          <representation mediaType="application/json"/>
        </request>
        <response>
          <representation mediaType="*/"/>
        </response>
      </method>
      <method name="HEAD" id="getUserAccountsHead">
        <request>
          <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="offset"/>
          <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="count"/>
          <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

        type="xs:string" style="query" name="filterType"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="filterString"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="sortType"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="sortOrder"/>
    </request>
    <response>
        <representation mediaType="application/xml"/>
        <representation mediaType="application/json"/>
        <representation mediaType="application/javascript"/>
    </response>
</method>
<method name="GET" id="getUserAccounts">
    <request>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="offset"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="count"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="filterType"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="filterString"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="sortType"/>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="sortOrder"/>
    </request>
    <response>
        <representation mediaType="application/xml"/>
        <representation mediaType="application/json"/>
        <representation mediaType="application/javascript"/>
    </response>
</method>
<method name="POST" id="resetPasswords">
    <request>
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema"
            type="xs:string" style="query" name="resetPasswords"/>
    </request>
    <response>
        <representation mediaType="application/xml"/>
        <representation mediaType="application/json"/>
    </response>
</method>
</resource>
</resources>
</application>

```

Query parameters

Some HCP 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 query parameters:

- If you specify an invalid value for a query parameter that requires a Boolean value (**true** or **false**), HCP interprets the value as **false**.
- If you specify an invalid value for any other required or optional query parameter, HCP returns a status code of 400 (Bad Request).
- If you omit a required query parameter, HCP returns a status code of 400 (Bad Request).
- If you specify a query parameter that's not valid for the request, HCP ignores it.

Query parameter names are case sensitive.

verbose

The **verbose** query parameter tells HCP how much information to return in response to a **GET** request for information about a resource. Valid values for this parameter are **true** and **false**.

In most cases, with **verbose=false**, a request for information about a resource returns only the properties whose values you can modify. For example, you cannot change the type of authentication for a user account. So, when you use **GET** with **verbose=false** to retrieve information about a user account, the `localAuthentication` property is omitted from the response body.

To retrieve all the properties for a resource, you need to append **verbose=true** to the resource URL. If you omit the **verbose** parameter, HCP uses the default value **false**.

prettyprint

The **prettyprint** query parameter causes the XML or JSON returned in response to a **GET** request to be formatted for readability. For example, with the **prettyprint** parameter, the returned XML for a list of tenants looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<tenants>
  <name>Default</name>
  <name>Finance</name>
  <name>HR</name>
  <name>Sales</name>
  <name>Marketing</name>
</tenants>
```

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

```
<?xml version="1.0" encoding="UTF-8"?><tenants><name>Default
</name><name>Finance</name><name>HR</name><name>Sales
</name><name>Marketing</name></tenants>
```

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

Request-specific query parameters

Some requests take query parameters that provide additional information to HCP about the operation you want to perform or that request a particular operation. The table below lists these requests and the query parameters they take.

Request	Parameters	More information
PUT to create an HCP tenant	username= <i>username</i> password= <i>password</i> forcePasswordChange=(true false) initialSecurityGroup= <i>group-name</i>	"HCP tenant query parameters" on page 263
PUT to create the default tenant and namespace	enterpriseMode=(true false) hashScheme= <i>hash-algorithm</i> searchEnabled=(true false) servicePlan= <i>service-plan-name</i>	"Default tenant query parameters" on page 265
PUT or POST to create or modify a user account	password= <i>password</i>	"Query parameter for setting user account passwords" on page 273

(Continued)

Request	Parameters	More information
POST to reset the security user passwords for a tenant	resetPasswords= <i>password</i>	"Query parameter for resetting security user passwords" on page 273
POST to reset the security group for a tenant	resetSecurityGroup= <i>group-name</i>	"Query parameter for resetting the security group" on page 153
POST to reset the indexing checkpoint for a namespace	resetMQECheckpoint= (<i>mm/dd/yyyy 0</i>)	"Query parameter for restarting indexing" on page 198
POST to perform an action on all replication links	shutDownAllLinks= <i>reason</i> reestablishAllLinks	"Query parameters for replication service actions" on page 224
POST to perform an action on a replication link	suspend resume failOver failBack restore beginRecovery completeRecovery	"Query parameters for replication link actions" on page 172
POST to perform an action on a replicating tenant	pause resume	"Query parameters for replication link content tenant actions" on page 247
GET to retrieve a limited number of tenants, namespaces, user accounts, or namespaces for which a given user or group account has any data access permissions	offset= <i>offset</i> count= <i>count</i>	"Paging through the resource lists" on page 98
GET to retrieve a sorted list of tenants, namespaces, user accounts, or namespaces for which a given user or group account has any data access permissions	sortType= <i>property-name</i> sortOrder=(ascending descending)	"Sorting resource lists" on page 100
GET to retrieve a filtered list of tenants, namespaces, user accounts, or namespaces for which a given user or group account has any data access permissions	filterType= <i>filter-type</i> filterString= <i>text-string</i>	"Filtering resource lists" on page 101
GET for tenant and namespace chargeback reports	start= <i>start-time</i> end= <i>end-time</i> granularity= <i>reporting-interval</i>	"Query parameters for generating chargeback reports" on page 114

When the only action you're requesting in a **POST** request for a resource other than a replication resource is specified by a query parameter, you need to provide an empty request body. With cURL, you specify this body as the argument for the **-d** option in the request:

- With a content type of XML, the argument is an empty root element for the resource being modified, enclosed in double quotation marks, like this:

```
-d "<root-element/>"
```

For example, here's a request to change only the password for the user account with the username *mwhite*:

```
curl -k -i -d "<userAccount/>" -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/
userAccounts/mwhite?password=p4ssw0rd"
```

- With a content type of JSON, the argument is an empty pair of curly braces enclosed in double quotation marks, like this:

```
-d "{}"
```

POST requests to perform actions on replication resources do not take a request body, empty or otherwise.

HCP-specific response headers

For an HCP management API request, the HTTP response headers always include the HCP-specific **X-HCP-SoftwareVersion** header. The value of this header is the version number of the currently installed HCP software; for example:

```
X-HCP-SoftwareVersion: 6.0.1.64
```

If a management API request results in an error and additional information about the error is available, the HTTP response headers include the HCP-specific **X-HCP-ErrorMessage** header; for example:

```
X-HCP-ErrorMessage: 'password' parameter is required.
```

Enabling the management API in the Management Consoles

To enable the HCP management API in the System and Tenant Management Consoles:

1. Log into the applicable Console using a user account with the security role.
2. In the top-level menu in the Console, mouse over **Security** to display a secondary menu.
3. In the secondary menu, click on **MAPI**.
4. In the **Management API Setting** section on the **Management API** page, select the **Enable the HCP management API** option.
5. Click on the **Update Settings** button.

Access and authentication

With the HCP 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 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 HCP access through the management API

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

- A format that starts with *admin*:

```
https://admin.hcp-domain-name:9090/mapi/resource-identifier
```

For example:

```
https://admin.hcp.example.com:9090/mapi/tenants
```

In this format, *hcp-domain-name* is the name of the domain associated with the [hcp_system] network.

To use this format, you need a system-level user account. Additionally, the management API must be configured at the system level to allow access from your client IP address.

- A format that starts by identifying a tenant:

```
https://tenant-name.hcp-domain-name:9090/mapi/  
resource-identifier
```

For example:

```
https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces
```

In this format, *hcp-domain-name* is the name of the domain associated with the tenant management network.

To use this format, you need a tenant-level user account or, if the tenant has granted system-level users administrative access to itself or is the default tenant, a system-level user account. Additionally, for an HCP tenant, the management API must be configured at the tenant level to allow access from your client IP address.

The following considerations apply to these URLs:

- The management API requires the use of SSL security with HTTP (HTTPS).



Note: If the HCP system uses a self signed SSL server certificate, the program submitting the management API request must include instructions either to trust the SSL certificate or 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. In Python with PycURL, you do this by setting the **SSL_VERIFYPEER** option to **false** or **0** (zero)

- The URL must always include the port number 9090. This is the port on which HCP listens for management API requests.
- The URL must always include the *mapi* interface identifier. This identifier is case sensitive.

For information on resource identifiers, see [Chapter 3, “Resources,”](#) on page 31.

Using an IP address in a URL

Normally, you let HCP choose the node on which to process a management API request. You can, however, use an IP address in the URL to access the system on a specific node. To do this, you replace the fully qualified domain name (FQDN) in the URL with a valid IP address for the node you want:

- To access HCP at the system level, you need to use an IP address defined for the node on the [hcp_system] network.
- To access HCP at the tenant level, you need to use an IP address defined for the node on the management network for the tenant.

If the node has both an IPv4 address and an IPv6 address in the applicable network, you can use either address. For example, to access the tenants resource at the system level on a node that has both the IPv4 address 192.168.210.16 and the IPv6 address 2001:0db8::101 defined for the [hcp_system] network, you can use either of these URLs:

`https://192.168.210.16:9090/mapi/tenants`

`https://[2001:0db8::101]:9090/mapi/tenants`

If you're accessing HCP at the system level, this is all you need to do. However, if you want to access a specific tenant, you also need to provide the fully qualified domain name of the tenant in an HTTP Host header. With cURL, you do this with the `-H` option. For example:

```
-H "Host: finance.hcp.example.com"
```

In Python with PycURL, you do this with the `HTTPHEADER` option. For example:

```
curl_setopt(pycurl.HTTPHEADER, ["Host: finance.hcp.example.com"])
```

For information on when to use an IP address for access to the HCP system, see ["Choosing an access method"](#) on page 282.



Note for tenant-level users: If you don't know the IP addresses for the HCP system, contact your HCP system administrator.

Using a hosts file

Typically, HCP is included as a subdomain in your DNS. If this is not the case:

- For system-level access, you can use an IP address in the URL, or you can use the [hcp_system] network domain name in the URL and use a `hosts` file to define mappings of one or more node IP addresses to that domain name.
- For tenant-level access, you need to use the tenant management network domain name in the URL and use a `hosts` file to define mappings of one or more node IP addresses to that domain name.

The location of the `hosts` file depends on the client operating system:

- On Windows, by default: `c:\windows\system32\drivers\etc\hosts`
- On Unix: `/etc/hosts`
- On Mac OS® X: `/private/etc/hosts`



Note: Every network is associated with a domain. If HCP is not included in your DNS, this is a dummy domain with a name that follows the conventions for well-formed DNS names.

Hostname mappings

Each entry in a `hosts` file is a mapping of an IP address for a system to an FQDN for that system.

Each `hosts` file entry that you create for access to HCP must include:

- A valid network IP address defined for an HCP node on the [hcp_system] network or tenant management network, as applicable
- The FQDN of the domain associated with the applicable network

For example, if the [hcp_system] network domain name is `hcp.example.com` and one of the HCP nodes has the IPv4 address `192.168.210.16` and the IPv6 address `2001:0db8::101` defined for that network, you could add either or both of these lines to the `hosts` file on the client:

```
192.168.210.16    admin.hcp.example.com
```

```
2001:0db8::101   admin.hcp.example.com
```

You can include comments in a `hosts` file either on separate lines or following a mapping on the same line. Each comment must start with a number sign (`#`). Blank lines are ignored.

Hostname mapping considerations

In the `hosts` file, you can map IP addresses for any number of nodes to a single domain name. The way a client uses multiple IP address mappings for a single domain name depends on the client platform. For information on how your client handles `hosts` file entries that define multiple IP address mappings for a single domain name, see your client documentation.

If any of the HCP nodes listed in the `hosts` file are unavailable, timeouts may occur when you use a `hosts` file to access the system through the management API.

Sample hosts file

Here's a sample hosts file that contains mappings at the system and tenant levels for both IPv4 and IPv6 addresses:

```
# HCP system-level mappings
192.168.210.16 admin.hcp.example.com
192.168.210.17 admin.hcp.example.com
192.168.210.18 admin.hcp.example.com
192.168.210.19 admin.hcp.example.com
2001:0db8::101 admin.hcp.example.com
2001:0db8::102 admin.hcp.example.com
2001:0db8::103 admin.hcp.example.com
2001:0db8::104 admin.hcp.example.com

# tenant-level mappings
192.168.210.16 finance.hcp.example.com
192.168.210.17 finance.hcp.example.com
192.168.210.18 finance.hcp.example.com
192.168.210.19 finance.hcp.example.com
2001:0db8::101 finance.hcp.example.com
2001:0db8::102 finance.hcp.example.com
2001:0db8::103 finance.hcp.example.com
2001:0db8::104 finance.hcp.example.com

192.168.210.16 hr.hcp.example.com
192.168.210.17 hr.hcp.example.com
192.168.210.18 hr.hcp.example.com
192.168.210.19 hr.hcp.example.com
2001:0db8::101 hr.hcp.example.com
2001:0db8::102 hr.hcp.example.com
2001:0db8::103 hr.hcp.example.com
2001:0db8::104 hr.hcp.example.com
```

URL considerations

The following considerations apply to URLs in management API requests.

URL length

The portion of a 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, HCP 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.



Note: In the values of query parameters appended to URLs, percent-encode question marks (?) and ampersands (&) but not other special characters.

Quotation marks with URLs in command lines

When using the HCP 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.

Authentication

To access the HCP system through the management API, you need to provide credentials in the form of a username and password. These credentials can be for a system-level user account or a tenant-level user account.



Note: To use a user account that was created in an HCP release earlier than 4.0, you need to log into the System or Tenant Management Console, as applicable, with that account at least once before you can use the account with the management API.

You need to provide credentials with every management API request. If you do not provide credentials or provide invalid credentials, HCP responds with a 403 (Forbidden) error message.

To provide credentials in a management API request, you specify an **authentication token** in an HTTP Authorization request header.

HCP also accepts credentials provided in an **hcp-api-auth** cookie. However, this method of providing credentials has been deprecated and should not be used in new applications.



Note: To use a recognized Active Directory user account for access to HCP through the management API, applications must use the SPNEGO protocol to negotiate the AD user authentication themselves. For more information on SPNEGO, see <http://tools.ietf.org/html/rfc4559>.

Authentication token

An authentication token consists of a username in Base64-encoded format and a password that's hashed using the MD5 hash algorithm, separated by a colon, like this:

base64-encoded-username:md5-hashed-password

For example, here's the token for the Base64-encoded username *lgreen* and the MD5-hashed password *p4ssw0rd*:

bGdyZWVu:2a9d119df47ff993b662a8ef36f9ea20

The GNU Core Utilities include the **base64** and **md5sum** commands, which convert text to Base64-encoded and MD5-hashed values, respectively. With these commands, a line such as this creates the required token:

```
echo `echo -n username | base64`:`echo -n password | md5sum` |  
awk '{print $1}'
```

The character before *echo*, before and after the colon, and following *md5sum* is a backtick (or grave accent). The **-n** option in the **echo** command prevents the command from appending a new line to the output. This is required to ensure correct Base64 and MD5 values.

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

Other tools that generate Base64-encoded and MD5-hashed values are available for download on the web. For security reasons, do not use interactive public web-based tools to generate these values.

Authorization header

You use the HTTP Authorization request header to provide the authentication token for an HCP management API request. The value of this header is **HCP** followed by the authentication token, in this format:

```
Authorization: HCP authentication-token
```

For example, here's the Authorization header for a user named *lgreen* and password *p4ssw0rd*:

```
Authorization: HCP bGdyZWVu:2a9d119df47ff993b662a8ef36f9ea20
```

Specifying the Authorization header with cURL

With cURL, you use the **-H** option to specify a header. So, for example, a request to list the tenants for the HCP system named *hcp.example.com* might look like this:

```
curl -k -i -H "Authorization: HCP bGdyZWVu:2a9d119df47ff993b662a8ef36f9ea20"  
-H "Accept: application/xml"  
"https://admin.hcp.example.com:9090/mapi/tenants"
```

Specifying the authentication header in Python with PycURL

In Python with PycURL, you use the **HTTPHEADER** option to specify a header, as in this example:

```
curl.setopt(pycurl.HTTPHEADER, ["Authorization: HCP  
bGdyZWVu:2a9d119df47ff993b662a8ef36f9ea20"])
```


Resources

The main types of HCP management API resources are tenants, namespaces, retention classes, content classes, user accounts, group accounts, and replication. Each main type of resource is associated with a set of resource identifiers, each of which identifies one of these:

- A list of resources of that type
- An instance of that type of resource
- A property of that type of resource that's treated as a resource in its own right

A resource identifier is the portion of a resource URL that follows *mapi*. For information on resource URLs, see [“URLs for HCP access through the management API”](#) on page 22.

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

- The data type of the resource when you use the resource in conjunction with a specific method. For information on data types that are sets of properties, see [Chapter 4. “Data types.”](#) on page 103.
- The methods supported by the resource. Because all resources support the **OPTIONS** method, this method is not included in the tables.
- The use for each supported method.

- The type of account required for access to the resource with each supported method.



Note: Where the tables in this chapter specify that a tenant-level user account is required, you can also use a system-level user account if the tenant has granted system-level users administrative access to itself. Additionally, in this case, when you use a system-level user account to retrieve information about the tenant, you get both the system-level and tenant-level tenant properties.

- Any additional notes about the resource.

This chapter also contains examples of using the management API to manipulate resources. Each example consists of:

- The **curl** command for the sample request
- The Python code for the sample request
- The HTTP request headers
- The HTTP response headers
- The request or response body, as applicable, in XML



Note: Each example in this chapter stands on its own. The requirements for executing each example are outlined in its initial description.

All the examples assume an HCP system that supports Active Directory, includes the replication feature, and has virtual network management enabled.

Additionally, this chapter contains instructions for paging through, sorting, and filtering the results of **GET** requests for tenants, namespaces, user accounts, and data access permissions.

Tenant resources

Tenant resources let you create, retrieve information about, modify, and delete tenants. The table below provides information about these resources.



Note: The management API is automatically enabled for tenants you create through the management API.

Data type	Method	Use	Access	Notes
.../tenants				
tenant	PUT	Create an HCP tenant or the default tenant and namespace.	System-level user account with the administrator role	<p>You can create the default tenant and namespace only if this is allowed by the HCP system configuration.</p> <p>For information on required and optional query parameters for creating tenants, see “Query parameters for creating tenants” on page 263.</p>
List	GET	Retrieve a list of the tenants defined in an HCP system	System-level user account with the monitor or administrator role	<p>The listed tenants are identified by tenant name.</p> <p>In XML, each listed tenant is the value of an element named name. In JSON, the name in the name/value pair that lists the tenants is name.</p>
.../tenants/tenant-name				
tenant	GET	Retrieve information about a tenant	<ul style="list-style-type: none"> For an HCP tenant, system-level user account with the monitor or administrator role or tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the monitor or administrator role 	For an HCP tenant, the information returned depends on whether you're using a system-level or tenant-level user account.

(Continued)

Data type	Method	Use	Access	Notes
N/A	HEAD	Check for the existence of a tenant	<ul style="list-style-type: none"> For an HCP tenant, system-level user account with the monitor or administrator role or tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the monitor or administrator role 	
tenant	POST	Modify a tenant	<ul style="list-style-type: none"> For an HCP tenant, system-level user account with the administrator role or tenant-level user account with the administrator role For the default tenant, system-level user account with the administrator role 	For an HCP tenant, the information you can modify depends on whether you're using a system-level or tenant-level user account.
N/A	DELETE	Delete a tenant	System-level user account with the administrator role	<p>The tenant cannot own any namespaces.</p> <p>Not valid for the default tenant.</p>
.../tenants/<i>tenant-name</i>/availableServicePlans				
List	GET	Retrieve a list of the service plans that are available for the tenant to assign to its namespaces	Tenant-level user account with the monitor or administrator role	<p>Valid only if the tenant is configured to allow service plan selection.</p> <p>The listed service plans are identified by service plan name.</p> <p>In XML, each listed service plan is the value of an element named name. In JSON, the name in the name/value pair that lists the service plans is name.</p>

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/availableServicePlans/<i>service-plan-name</i>				
available ServicePlan	GET	Retrieve information about a service plan that's available for the tenant to assign to its namespaces	Tenant-level user account with the monitor or administrator role	Valid only if the tenant is configured to allow service plan selection.
.../tenants/<i>tenant-name</i>/chargebackReport				
chargeback Report	GET	Generate a chargeback report for a tenant	Tenant-level user account with the monitor or administrator role	<p>Not valid for the default tenant.</p> <p>Supported output formats are XML, JSON, and CSV.</p> <p>For information on optional query parameters for generating chargeback reports, see "Query parameters for generating chargeback reports" on page 114.</p> <p>For information on chargeback reports, see "About chargeback reports" on page 112.</p>
.../tenants/<i>tenant-name</i>/consoleSecurity				
consoleSecurity	GET	Retrieve the Tenant Management Console configuration for a tenant	Tenant-level user account with the security role	Not valid for the default tenant.
	POST	Modify the Tenant Management Console configuration for a tenant	Tenant-level user account with the security role	

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/contactInfo				
contactInfo	GET	Retrieve the contact information for a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the monitor or administrator role 	
	POST	Modify the contact information for a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the administrator role For the default tenant, system-level user account with the administrator role 	
.../tenants/<i>tenant-name</i>/emailNotification				
email Notification	GET	Retrieve the email notification configuration for a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the with the monitor or administrator role 	
	POST	Modify the email notification configuration for a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the administrator role For the default tenant, system-level user account with the administrator role 	

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/tenant-name/namespaceDefaults				
namespace Defaults	GET	Retrieve the default settings for namespace creation for a tenant	Tenant-level user account with the monitor or administrator role	Not valid for the default tenant.
	POST	Modify the default settings for namespace creation for a tenant	Tenant-level user account with the administrator role	
.../tenants/tenant-name/permissions				
List	GET	Retrieve the list of permissions in the data access permission mask for a tenant	<ul style="list-style-type: none">For an HCP tenant, tenant-level user account with the monitor, administrator, or compliance roleFor the default tenant, system-level user account with the monitor, administrator, or compliance role	For the format of the permission list, see "Permission lists" on page 105.

(Continued)

Data type	Method	Use	Access	Notes
List	POST	Modify the list of permissions in the data access permission mask for a tenant	Tenant-level user account with the administrator role	<p>Valid values for permissions are:</p> <ul style="list-style-type: none"> • DELETE • PRIVILEGED • PURGE • READ • SEARCH • WRITE <p>These values are case sensitive.</p> <p>For the format of the permission list, see "Permission lists" on page 105.</p> <p>The set of permissions specified in the request body replaces the set of permissions currently included in the data access permission mask for the tenant. To remove all permissions, specify an empty list.</p> <p>If the set of permissions includes PURGE, delete permission is enabled automatically. If the set of permissions includes SEARCH, read permission is enabled automatically.</p> <p>By default, when you create a tenant, its data access permission mask includes all permissions.</p>

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/ <i>tenant-name</i> /searchSecurity				
searchSecurity	GET	Retrieve the Search Console configuration for a tenant	Tenant-level user account with the security role	Not valid for the default tenant.
	POST	Modify the Search Console configuration for a tenant	Tenant-level user account with the security role	
.../tenants/ <i>tenant-name</i> /statistics				
statistics	GET	Retrieve statistics about the content of the namespaces owned by a tenant	<ul style="list-style-type: none">For an HCP tenant, tenant-level user account with the monitor or administrator roleFor the default tenant, system-level user account with the monitor or administrator role	

Example: Creating an HCP tenant

Here's a sample **PUT** request that creates a tenant named Finance in the HCP system named hcp.example.com. The tenant definition is in an XML file named `FinanceTenant.xml`. The initial user account for the tenant has a username of *lgreen* and a password of *start123*. These are specified by query parameters. The request is made using a system-level user account that includes the administrator role.

This example assumes the existence of a service plan named Short-Term-Activity.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tenant>
  <name>Finance</name>
  <systemVisibleDescription>Created for the Finance department at Example
    Company by P.D. Grey on 2/9/2012.</systemVisibleDescription>
  <hardQuota>100 GB</hardQuota>
  <softQuota>90</softQuota>
  <namespaceQuota>5</namespaceQuota>
</tenant>
```

```
<authenticationTypes>
  <authenticationType>LOCAL</authenticationType>
  <authenticationType>EXTERNAL</authenticationType>
</authenticationTypes>
<complianceConfigurationEnabled>true</complianceConfigurationEnabled>
<versioningConfigurationEnabled>true</versioningConfigurationEnabled>
<searchConfigurationEnabled>true</searchConfigurationEnabled>
<replicationConfigurationEnabled>true</replicationConfigurationEnabled>
<tags>
  <tag>Example Company</tag>
  <tag>pdgrey</tag>
</tags>
<servicePlanSelectionEnabled>false</servicePlanSelectionEnabled>
<servicePlan>Short-Term-Activity</servicePlan>
<dataNetwork>net127</dataNetwork>
<managementNetwork>net004</managementNetwork>
</tenant>
```

Request with cURL command line

```
curl -k -iT FinanceTenant.xml -H "Content-Type: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants?username=lgreen
&password=start123&forcePasswordChange=false"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("FinanceTenant.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
  "Authorization: HCP \
  YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
  "https://admin.hcp.example.com:9090/mapi/tenants?" +
  "username=lgreen&password=start123&forcePasswordChange=false")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("FinanceTenant.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```


Request headers

```

PUT /mapi/tenants?username=lgreen&password=start123
    &forcePasswordChange=false HTTP/1.1
Host: admin.hcp.example.com:9090
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Content-Type: application/xml
Content-Length: 1016

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Setting Tenant Management Console security for a tenant

Here's a sample **POST** request that modifies the Tenant Management Console configuration for the Finance tenant. The Console configuration information is in an XML file named `FinanceMgmtConsole.xml`. The request is made using a tenant-level user account that includes the security role.

Request body in XML

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<consoleSecurity>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.103.18</ipAddress>
      <ipAddress>192.168.103.24</ipAddress>
      <ipAddress>192.168.103.25</ipAddress>
    </allowAddresses>
    <denyAddresses/>
    <allowIfInBothLists>false</allowIfInBothLists>
  </ipSettings>
  <minimumPasswordLength>6</minimumPasswordLength>
  <forcePasswordChangeDays>45</forcePasswordChangeDays>
  <disableAfterAttempts>3</disableAfterAttempts>
  <disableAfterInactiveDays>30</disableAfterInactiveDays>
  <logoutOnInactive>20</logoutOnInactive>
  <loginMessage/>
</consoleSecurity>

```

Request with cURL command line

```
curl -k -i -d @FinanceMgmtConsole.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/consoleSecurity"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("FinanceMgmtConsole.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "consoleSecurity")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.setopt(pycurl.INFILESIZE,
    os.path.getsize("FinanceMgmtConsole.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
POST /mapi/tenants/finance/consoleSecurity HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 620
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```

Example: Generating a chargeback report for a tenant

Here's a sample **GET** request that generates a chargeback report for the tenant named Finance, which has two namespaces, Accounts-Payable and Accounts-Receiveable. The report covers the one hour starting from 1:00 p.m. on March 26, 2014. The report is written in CSV format to a file named `FinanceStats-2014-03-26-1300.txt`. The request is made using a tenant-level user account that includes the administrator role.

Request with cURL command line

```
curl -k -i -H "Accept: text/csv"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/
chargebackReport?start=2014-03-26T13:00:00-0400
&end=2014-03-26T13:59:59-0400&granularity=hour" >
> FinanceStats-2014-03-26-1300.txt
```

Request in Python using PycURL

```
import pycurl
filehandle = open("FinanceStats-2014-13-26-1300.txt", 'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Accept: text/csv",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "chargebackReport?start=2014-03-26T13:00:00-0400" +
    "&end=2014-03-26T13:59:59-0400&granularity=hour")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.WRITEFUNCTION, filehandle.write)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
GET /mapi/tenants/finance/chargebackReport?start=2014-03-26T13:00:00-
0400&end=2014-03-26T13:59:59-0400&granularity=hour HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Accept: text/csv
```

Response headers

HTTP/1.1 200 OK
 Content-Type: text/csv
 X-HCP-SoftwareVersion: 7.0.0.16
 Content-Length: 565

Response body

systemName,tenantName,namespaceName,startTime,endTime,objectCount,
 ingestedVolume,storageCapacityUsed,bytesIn,bytesOut,reads,writes,deletes,deleted,
 valid
 hcp.example.com,Finance,Accounts-Payable,2014-03-26T13:11:13-0400,
 2014-03-26T13:59:59-0400,349,427316,885932,0,2531,2,0,0,false,false
 hcp.example.com,Finance,Accounts-Receiveable,2014-03-26T13:53:15-0400,
 2014-03-26T13:59:59-0400,2575,34173401,73624501,26652,67241,3,1,0,false,true
 hcp.example.com,Finance,,2014-03-26T13:08:10-0400,2014-03-26T13:59:59-0400,
 2924,34600717,74510433,26652,69772,5,1,0,false,true

Namespace resources

Namespace resources let you create, retrieve information about, modify, and delete namespaces. The table below provides information about these resources.

Data type	Method	Use	Access	Notes
.../tenants/ <i>tenant-name</i> /namespaces				
namespace	PUT	Create an HCP namespace	Tenant-level user account with the administrator role or allow namespace management property	

(Continued)

Data type	Method	Use	Access	Notes
List	GET	Retrieve a list of the namespaces owned by a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the monitor, administrator, or compliance role or allow namespace management property For the default tenant, system-level user account with the monitor or administrator role 	<p>The listed namespaces are identified by namespace name.</p> <p>In XML, each listed namespace is the value of an element named name. In JSON, the name in the name/value pair that lists the namespaces is name.</p> <p>For a user with the allow namespace management property and no roles, the retrieved list includes only the namespaces the user owns.</p>
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>				
namespace	GET	Retrieve information about a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor or administrator role or allow namespace management property For the default namespace, system-level user account with the monitor or administrator role 	A user with the allow namespace management property and no roles can retrieve information only about the namespaces the user owns. In this case, the retrieved information includes only the namespace name and owner.
N/A	HEAD	Check for the existence of a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor, administrator, or compliance role or allow namespace management property For the default namespace, system-level user account with the monitor or administrator role 	

(Continued)

Data type	Method	Use	Access	Notes
namespace	POST	Modify a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the administrator role For the default namespace, system-level user account with the administrator role 	For information on query parameters for resetting indexing checkpoints, see “Query parameter for restarting indexing” on page 198.
N/A	DELETE	Delete an HCP namespace	Tenant-level user account with the administrator role or allow namespace management property	<p>A user with the allow namespace management property and no roles can delete only the namespaces the user owns.</p> <p>The namespace cannot contain any objects.</p> <p>Not valid for the default namespace.</p>
String	GET	Retrieve information about whether a namespace is optimized for all access protocols or for optimized for cloud protocols only.	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the administrator role 	The default setting is to optimized for all protocols unless a user with the administrator role changed the setting on the System Management Console.

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/chargebackReport				
chargeback Report	GET	Generate a chargeback report for a namespace	Tenant-level user account with the monitor or administrator role	<p>Not valid for the default namespace.</p> <p>Supported output formats are XML, JSON, and CSV.</p> <p>For information on optional query parameters for generating chargeback reports, see “Query parameters for generating chargeback reports” on page 114.</p> <p>For information on chargeback reports, see “About chargeback reports” on page 112.</p>
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/complianceSettings				
compliance Settings	GET	Retrieve the default retention, shred, custom metadata handling, and disposition settings for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor, administrator, or compliance role For the default namespace, system-level user account with the monitor, administrator, or compliance role 	Default retention and shred settings do not apply to the default namespace.
	POST	Modify the default retention, shred, custom metadata handling, or disposition setting for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the compliance role For the default namespace, system-level user account with the compliance role 	

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/tenant-name/namespaces/namespace-name/customMetadataIndexingSettings				
custom Metadata Indexing Settings	GET	Retrieve settings specific to metadata query engine indexing of custom metadata for a search-enabled namespace	<ul style="list-style-type: none">For an HCP namespace, tenant-level user account with the monitor or administrator roleFor the default namespace, system-level user account with the monitor or administrator role	Not valid for namespaces that do not have search enabled.
	POST	Modify settings specific to metadata query engine indexing of custom metadata for a search-enabled namespace	<ul style="list-style-type: none">For an HCP namespace, tenant-level user account with the administrator roleFor the default namespace, system-level user account with the administrator role	
.../tenants/tenant-name/namespaces/namespace-name/permissions				
List	GET	Retrieve the list of permissions in the data access permission mask for a namespace	<ul style="list-style-type: none">For an HCP namespace, tenant-level user account with the monitor or administrator roleFor the default namespace, system-level user account with the monitor or administrator role	For the format of the permission list, see "Permission lists" on page 105.

(Continued)

Data type	Method	Use	Access	Notes
List	POST	Modify the list of permissions in the data access permission mask for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the administrator role For the default namespace, system-level user account with the administrator role 	<p>Valid values for permissions are:</p> <ul style="list-style-type: none"> DELETE PRIVILEGED PURGE READ SEARCH WRITE <p>These values are case sensitive.</p> <p>For the format of the permission list, see "Permission lists" on page 105.</p> <p>The set of permissions specified in the request body replaces the set of permissions currently included in the data access permission mask for the namespace. To remove all permissions, specify an empty list.</p> <p>If the set of permissions includes PURGE, delete permission is enabled automatically. If the set of permissions includes SEARCH, read permission is enabled automatically.</p> <p>By default, when you create a namespace, its data access permission mask includes all permissions.</p>

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/tenant-name/namespaces/namespace-name/protocols				
protocols	GET	Retrieve a subset of the HTTP namespace access protocol settings for the default namespace	System-level user account with the monitor or administrator role	Not valid for HCP namespaces. For HCP namespaces, this resource has been superseded by the .../protocols/http resource.
	POST	Modify a subset of the HTTP namespace access protocol settings for the default namespace	System-level user account with the administrator role	
.../tenants/tenant-name/namespaces/namespace-name/protocols/protocol-name				
Determined by protocol-name. Possible data types are: cifsProtocol httpProtocol nfsProtocol smtpProtocol	GET	Retrieve the applicable namespace access protocol settings for a namespace	Tenant-level user account with the monitor or administrator role	Not valid for the default namespace. Valid values for protocol-name are: <ul style="list-style-type: none">• cifs• http• nfs• smtp These values are case sensitive. The httpProtocol data type includes properties for both the HTTP and WebDAV protocols.
	POST	Modify the applicable namespace access protocol settings for a namespace	Tenant-level user account with the administrator role	

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/replicationCollisionSettings				
replication Collision Settings	GET	Retrieve the replication collision handling settings for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor or administrator role For the default namespace, system-level user account with the monitor or administrator role 	
	POST	Modify the replication collision handling settings for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the administrator role For the default namespace, system-level user account with the administrator role 	
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/statistics				
statistics	GET	Retrieve information about the content of a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor or administrator role For the default namespace, system-level user account with the monitor or administrator role 	

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/versioningSettings				
versioning Settings	GET	Retrieve the versioning settings for a namespace	Tenant-level user account with the monitor or administrator role or the allow namespace management property	A user with the allow namespace management property and no roles can retrieve and modify only the versioning enabled property and that property only for the namespaces the user owns. Not valid for the default namespace.
	POST	Modify the versioning settings for a namespace	Tenant-level user account with the administrator role or the allow namespace management property	

Example: Creating an HCP namespace

Here's a sample **PUT** request that creates an HCP namespace named Accounts-Receiveable for the Finance tenant. The namespace definition is in an XML file named `AccountsRecNS.xml`. The request is made using a tenant-level user account that includes the administrator role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<namespace>
  <name>Accounts-Receiveable</name>
  <description>Created for the Finance department at Example Company by Lee
    Green on 2/9/2012.</description>
  <hashScheme>SHA-256</hashScheme>
  <enterpriseMode>true</enterpriseMode>
  <hardQuota>50 GB</hardQuota>
  <softQuota>75</softQuota>
  <optimizedFor>ALL</optimizedFor>
  <versioningSettings>
    <enabled>true</enabled>
    <prune>true</prune>
    <pruneDays>10</pruneDays>
  </versioningSettings>
  <searchEnabled>true</searchEnabled>
  <indexingEnabled>true</indexingEnabled>
  <customMetadataIndexingEnabled>true</customMetadataIndexingEnabled>
  <replicationEnabled>true</replicationEnabled>
  <readFromReplica>true</readFromReplica>
</namespace>
```

```

<serviceRemoteSystemRequests>true</serviceRemoteSystemRequests>
<tags>
  <tag>Billing</tag>
  <tag>lgreen</tag>
</tags>
</namespace>

```

Request with cURL command line

```

curl -k -iT AccountsRecNS.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces"

```

Request in Python using PycURL

```

import pycurl
import os
filehandle = open("AccountsRecNS.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "namespaces")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("AccountsRecNS.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

PUT /mapi/tenants/finance/namespaces HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 835

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Changing the compliance settings for an HCP namespace

Here's a sample **POST** request that changes the compliance settings for the Accounts-Receiveable namespace. The new settings are in an XML file named `AR-compliance.xml`. The request is made using a tenant-level user account that includes the compliance role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<complianceSettings>
  <retentionDefault>A+7y</retentionDefault>
  <shreddingDefault>false</shreddingDefault>
  <customMetadataChanges>all</customMetadataChanges>
  <dispositionEnabled>true</dispositionEnabled>
</complianceSettings>
```

Request with cURL command line

```
curl -k -i -d @AR-compliance.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bXdoaXRl:ad49ce36d0cec9634ef63b24151be0fb"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces/
accounts-receivable/complianceSettings"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("AR-compliance.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bXdoaXRl:ad49ce36d0cec9634ef63b24151be0fb"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "namespaces/accounts-receivable/complianceSettings")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.setopt(pycurl.INFILESIZE,
    os.path.getsize("AR-compliance.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
POST /mapi/tenants/finance/namespaces/accounts-receivable/complianceSettings
HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bXdoaXRI:ad49ce36d0cec9634ef63b24151be0fb
Content-Type: application/xml
Content-Length: 285
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```

Example: Configuring the HTTP protocol for an HCP namespace

Here's a sample **POST** request that configures the HTTP protocol for the Accounts-Receiveable namespace. The new settings are in an XML file named `AR-http.xml`. The request is made using a tenant-level user account that includes the administrator role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<httpProtocol>
  <httpsEnabled>true</httpsEnabled>
  <httpEnabled>>false</httpEnabled>
  <restEnabled>true</restEnabled>
  <restRequiresAuthentication>true</restRequiresAuthentication>
  <httpActiveDirectorySSOEnabled>true</httpActiveDirectorySSOEnabled>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.140.10</ipAddress>
      <ipAddress>192.168.140.14</ipAddress>
      <ipAddress>192.168.140.15</ipAddress>
      <ipAddress>192.168.149.0/24</ipAddress>
    </allowAddresses>
    <allowIfInBothLists>>false</allowIfInBothLists>
    <denyAddresses>
      <ipAddress>192.168.149.5</ipAddress>
    </denyAddresses>
  </ipSettings>
</httpProtocol>
```

Request with cURL command line

```
curl -k -i -d @AR-http.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces/
accounts-receivable/protocols/http"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("AR-http.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "namespaces/accounts-receivable/protocols/http")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.setopt(pycurl.INFILESIZE,
    os.path.getsize("AR-http.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
POST /mapi/tenants/finance/namespaces/accounts-receivable/protocols/http
HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 285
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```


Network resources

Network resources let you activate advanced DNS configuration mode and view the current DNS configuration setting. The table below provides information about these resources.

Data type	Method	Use	Access	Notes
.../network				
Network settings	GET	Retrieves the current downstream DNS mode.	System-level user account with the monitor or administrator role	
	POST	Modifies the current downstream DNS mode setting for the given network.	System-level user account with the administrator role	Valid values for this setting are: <ul style="list-style-type: none"> ADVANCED BASIC

Example: Enabling advanced downstream DNS configuration

Here's a sample **POST** request that sets the downstream DNS mode to advanced. The DNS mode is specified in an XML file named `network.xml`. The request is made using a system-level user account that has the administrator role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<networkSettings>
  <downstreamDNSMode>ADVANCED</downstreamDNSMode>
</networkSettings>
```

Request with cURL command line

```
curl -k -i -d @network.xml -H "Content-Type: application/xml"
-H "Authorization: HCP YWRtaW4=:e00cf25ad42683b3df678c61f42c6bda"
"https://admin.hcp.example.com:9090/mapi/network"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("network.xml", 'rb')
curl = pycurl.Curl()
```

```

curl_setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml", /
"Authorization: HCP YWRtaW4=:e00cf25ad42683b3df678c61f42c6bda"])
curl_setopt(pycurl.URL,
    "https://admin.hcp.example.com:9090/mapi/network"
curl_setopt(pycurl.SSL_VERIFYPEER, 0)
curl_setopt(pycurl.SSL_VERIFYHOST, 0)
curl_setopt(pycurl.CUSTOMREQUEST, "POST")
curl_setopt(pycurl.UPLOAD, 1)
curl_setopt(pycurl.INFILESIZE, os.path.getsize("network.xml"))
curl_setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

POST /admin/network?downstreamDNSMode=ADVANCED HTTP/1.1
Host: admin.hcp.example.com:9090
Authorization: HCP YWRtaW4=:e00cf25ad42683b3df678c61f42c6bda
Content-Type: application/xml
Content-Length: 141

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.1.105
Content-Length: 0

```

Example: Checking the advanced DNS configuration

Here's a sample **GET** request that lists whether the advanced DNS configuration mode is enabled or disabled. The request is made using a system-level user account that has the administrator role.

Request with cURL command line

```

curl -k -H "Content-Type: application/xml"
-H "Authorization: HCP YWRtaW4=:e00cf25ad42683b3df678c61f42c6bda"
"https://admin.hcp.example.com:9090/mapi/network"

```

Request in Python using PycURL

```

import pycurl
import os
filehandle = open("network.xml", 'rb')
curl = pycurl.Curl()
curl_setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml", /
"Authorization: HCP YWRtaW4=:e00cf25ad42683b3df678c61f42c6bda"])

```

```

curl_setopt(pycurl.URL,
    "https://admin.hcp.example.com:9090/mapi/network"
curl_setopt(pycurl.SSL_VERIFYPEER, 0)
curl_setopt(pycurl.SSL_VERIFYHOST, 0)
curl_setopt(pycurl.CUSTOMREQUEST, "GET")
curl_setopt(pycurl.UPLOAD, 1)
curl_setopt(pycurl.INFILESIZE, os.path.getsize("network.xml"))
curl_setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

GET /admin/network HTTP/1.1
Host: admin.hcp.example.com:9090
Authorization: HCP YWRtaW4=:e00cf25ad42683b3df678c61f42c6bda
Content-Type: application/xml
Content-Length: 152

```

Response headers

```

HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.1.0.353
Content-Length: 0

```

Response body in XML

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<networkSettings>
  <downstreamDNSMode>BASIC</downstreamDNSMode>
</networkSettings>

```

Retention class resources

Retention class resources let you create, retrieve information about, modify, and delete retention classes. The table below provides information about these resources.

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/retentionClasses				
retentionClass	PUT	Create a retention class for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the compliance role For the default namespace, system-level user account with the compliance role 	
List	GET	Retrieve a list of the retention classes defined for a namespace	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor, administrator, or compliance role For the default namespace, system-level user account with the monitor, administrator, or compliance role 	<p>The listed retention classes are identified by retention class name.</p> <p>In XML, each listed retention class is the value of an element named name. In JSON, the name in the name/value pair that lists the retention classes is name.</p>
.../tenants/<i>tenant-name</i>/namespaces/<i>namespace-name</i>/retentionClasses/<i>retention-class-name</i>				
retentionClass	GET	Retrieve information about a retention class	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor, administrator, or compliance role For the default namespace, system-level user account with the monitor, administrator, or compliance role 	

(Continued)

Data type	Method	Use	Access	Notes
N/A	HEAD	Check for the existence of a retention class	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the monitor, administrator, or compliance role For the default namespace, system-level user account with the monitor, administrator, or compliance role 	
retentionClass	POST	Modify a retention class	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the compliance role For the default namespace, system-level user account with the compliance role 	
N/A	DELETE	Delete a retention class	<ul style="list-style-type: none"> For an HCP namespace, tenant-level user account with the compliance role For the default namespace, system-level user account with the compliance role 	You can delete a retention class only if the namespace is in enterprise mode.

Example: Creating a retention class

Here's a sample **PUT** request that creates a retention class named FN-Std-42 for the Accounts-Receivable namespace. The retention class is defined in an XML file named RC-FN-Std-42.xml. The request is made using a tenant-level user account that includes the compliance role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<retentionClass>
  <name>FN-Std-42</name>
  <description>Implements Finance department standard #42 - keep for 10
    years.</description>
  <value>A+10y</value>
  <allowDisposition>true</allowDisposition>
</retentionClass>
```

Request with cURL command line

```
curl -k -iT RC-FN-Std-42.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bXdoaXRl:ad49ce36d0cec9634ef63b24151be0fb"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces/
accounts-receivable/retentionClasses"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("RC-FN-Std-42.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
  "Authorization: HCP \
  bXdoaXRl:ad49ce36d0cec9634ef63b24151be0fb"])
curl.setopt(pycurl.URL,
  "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
  "namespaces/accounts-receivable/retentionClasses")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("RC-FN-Std-42.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```

PUT /mapi/tenants/finance/namespaces/accounts-receivable/retentionClasses
HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bXdoaXRI:ad49ce36d0cec9634ef63b24151be0fb
Content-Type: application/xml
Content-Length: 275

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Retrieving a list of retention classes

Here's a sample **GET** request that retrieves a list of the retention classes defined for the Accounts-Receiveable namespace. The request writes the list of retention classes to a file named `AR-retclasses.xml`. The request is made using a tenant-level user account that includes the compliance role.

Request with cURL command line

```

curl -k -H "Accept: application/xml"
-H "Authorization: HCP bXdoaXRI:ad49ce36d0cec9634ef63b24151be0fb"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces/
accounts-receivable/retentionClasses?prettyprint" > AR-retclasses.xml

```

Request in Python using PycURL

```

import pycurl
filehandle = open("AR-retclasses.xml", 'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Accept: application/xml",
    "Authorization: HCP \
    bXdoaXRI:ad49ce36d0cec9634ef63b24151be0fb"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "namespaces/accounts-receivable/retentionClasses?prettyprint")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.WRITEFUNCTION, filehandle.write)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```
GET /mapi/tenants/finance/namespaces/accounts-receivable/retentionClasses
?prettyprint HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bXdoaXRI:ad49ce36d0cec9634ef63b24151be0fb
Accept: application/xml
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 136
```

Response body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<retentionClasses>
  <name>FN-Std-37</name>
  <name>FN-Std-42</name>
</retentionClasses>
```

Content class resources

Content class resources let you create, retrieve information about, modify, and delete content classes. These resources are not available for tenants that do not have search configuration enabled.

The table below provides information about content class resources.

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/contentClasses				
contentClass	PUT	Create a content class for a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the administrator role For the default tenant, system-level user account with the administrator role 	
List	GET	Retrieve a list of the content classes for a tenant	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the monitor or administrator role 	<p>The listed content classes are identified by content class name.</p> <p>In XML, each listed content class is the value of an element named name. In JSON, the name in the name/value pair that lists the content classes is name.</p>
.../tenants/<i>tenant-name</i>/contentClasses/<i>content-class-name</i>				
contentClass	GET	Retrieve information about a content class	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the monitor or administrator role 	
N/A	HEAD	Check for the existence of a content class	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the monitor or administrator role For the default tenant, system-level user account with the monitor or administrator role 	

(Continued)

Data type	Method	Use	Access	Notes
contentClass	POST	Modify a content class	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the administrator role For the default tenant, system-level user account with the administrator role 	
N/A	DELETE	Delete a content class	<ul style="list-style-type: none"> For an HCP tenant, tenant-level user account with the administrator role For the default tenant, system-level user account with the administrator role 	The content class cannot contain any content properties.

Example: Creating a content class

Here's a sample **PUT** request that creates a content class named DICOM and associates it with the Medical-Records namespace, which is search enabled. The content class is defined in an XML file named `dicom.xml`. The request is made using a tenant-level user account that includes the administrator role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<contentClass>
  <name>DICOM</name>
  <contentProperties>
    <contentProperty>
      <name>Doctor_Name</name>
      <expression>/dicom_image/doctor/name</expression>
      <type>STRING</type>
      <multivalued>>false</multivalued>
      <format></format>
    </contentProperty>
    <contentProperty>
      <name>Doctor_Specialty</name>
      <expression>/dicom_image/doctor/specialties/specialty</expression>
      <type>STRING</type>
      <multivalued>>true</multivalued>
      <format></format>
    </contentProperty>
  </contentProperties>
</contentClass>
```

```

</contentProperty>
<contentProperty>
  <name>Followup_Needed</name>
  <expression>/dicom_image/followup_needed</expression>
  <type>BOOLEAN</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
<contentProperty>
  <name>Image_Date</name>
  <expression>/dicom_image/image/date</expression>
  <type>DATE</type>
  <multivalued>>false</multivalued>
  <format>MM/dd/yyyy</format>
</contentProperty>
<contentProperty>
  <name>Image_Type</name>
  <expression>/dicom_image/image/@type</expression>
  <type>STRING</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
<contentProperty>
  <name>Patient_ID</name>
  <expression>/dicom_image/patient/id</expression>
  <type>INTEGER</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
<contentProperty>
  <name>Patient_Name</name>
  <expression>/dicom_image/patient/name</expression>
  <type>STRING</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
</contentProperties>
<namespaces>
  <name>Medical-Records</name>
</namespaces>
</contentClass>

```

Request with cURL command line

```

curl -k -iT dicom.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://anytown-general-hospital.hcp.example.com:9090/mapi/tenants/
anytown-general-hospital/contentClasses"

```

Request in Python using PycURL

```

import pycurl
import os
filehandle = open("dicom.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://anytown-general-hospital.hcp.example.com:9090/mapi/" +
    "tenants/anytown-general-hospital/contentClasses")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("dicom.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

PUT /mapi/tenants/anytown-general-hospital/contentClasses HTTP/1.1
Host: anytown-general-hospital.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 2702

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Retrieving a list of content classes

Here's a sample **GET** request that retrieves a list of the content classes defined for the Anytown-General-Hospital tenant. The request writes the list of content classes to a file named `AGH-cc.xml`. The request is made using a tenant-level user account that includes the administrator role.

Request with cURL command line

```

curl -k -i -H "Accept: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://anytown-general-hospital.hcp.example.com:9090/mapi/tenants/
anytown-general-hospital/contentClasses?prettyprint" > AGH-cc.xml

```

Request in Python using PycURL

```

import pycurl
filehandle = open("AGH-cc.xml", 'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Accept: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://anytown-general-hospital.hcp.example.com:9090/mapi/" +
    "tenants/anytown-general-hospital/contentClasses")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.WRITEFUNCTION, filehandle.write)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

GET /mapi/tenants/anytown-general-hospital/contentClasses?prettyprint HTTP/1.1
Host: anytown-general-hospital.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Accept: application/xml

```

Response headers

```

HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 143

```

Response body in XML

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contentClasses>
  <name>Appointment</name>
  <name>DICOM</name>
</contentClasses>

```

User account resources

User account resources let you create, retrieve information about, modify, and delete user accounts. The table below provides information about these resources.

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/userAccounts				
userAccount	PUT	Create a user account for a tenant	Tenant-level user account with the security role	For information on the required query parameter for creating a user account, see “Query parameter for setting user account passwords” on page 273.
List	GET	Retrieve a list of the user accounts defined for a tenant	Tenant-level user account with the monitor, administrator, or security role	The listed user accounts are identified by the account username. In XML, each listed user account is the value of an element named username . In JSON, the name in the name/value pair that lists the user accounts is username .
N/A	POST	Reset the passwords of all locally authenticated user accounts with the security role	System-level user account with the administrator role	For information on the required query parameter for resetting security user passwords, see “Query parameter for resetting security user passwords” on page 273.
.../tenants/<i>tenant-name</i>/userAccounts/<i>username</i>				
userAccount	GET	Retrieve information about a user account	Tenant-level user account with the monitor, administrator, or security role	The information returned depends on the roles associated with the user making the request.
N/A	HEAD	Check for the existence of a user account	Tenant-level user account with the monitor, administrator, or security role	

(Continued)

Data type	Method	Use	Access	Notes
userAccount	POST	Modify a user account	Tenant-level user account with the administrator or security role	<p>A user with only the administrator role can modify only the allow-<code>NamespaceManagement</code> property. A user with only the security role cannot modify that property.</p> <p>For information on the query parameter for changing the password for a user account, see “Query parameter for setting user account passwords” on page 273.</p>
N/A	DELETE	Delete a user account	Tenant-level user account with the security role	
.../tenants/tenant-name/userAccounts/username/dataAccessPermissions				
dataAccess Permissions	GET	Retrieve information about the data access permissions associated with a user account	Tenant-level user account with the administrator or security role	
	POST	Modify the data access permissions associated with a user account	Tenant-level user account with the administrator role	<p>The request body must contain all permissions granted for each included namespace. If a namespace is not included, its permissions are not changed by the POST request.</p> <p>By default, when you create a user account, it does not include any data access permissions.</p>

Example: Creating a user account

Here's a sample **PUT** request that creates a user account for the Finance tenant. The account is defined in an XML file named `mwhite-UA.xml`. The username for the account is specified in the XML file. The password is specified by the **password** query parameter. The request is made using a tenant-level user account that includes the security role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<userAccount>
  <username>mwhite</username>
  <fullName>Morgan White</fullName>
  <description>Compliance officer.</description>
  <localAuthentication>true</localAuthentication>
  <forcePasswordChange>true</forcePasswordChange>
  <enabled>true</enabled>
  <roles>
    <role>COMPLIANCE</role>
    <role>MONITOR</role>
  </roles>
  <allowNamespaceManagement>false</allowNamespaceManagement>
</userAccount>
```

Request with cURL command line

```
curl -k -iT mwhite-UA.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/userAccounts
?password=start123"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("mwhite-UA.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "userAccounts?password=start123")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("mwhite-UA.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
```



```

curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

PUT /mapi/tenants/finance/userAccounts?password=start123 HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 365

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Changing the roles associated with a user account

Here's a sample **POST** request that changes the roles associated with the user account with the username *mwhite*. The new set of roles is specified in an XML file named *mwhite-UAroles.xml*. The request is made using a tenant-level user account that includes the security role.

Request body in XML

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<userAccount>
  <roles>
    <role>COMPLIANCE</role>
    <role>MONITOR</role>
    <role>ADMINISTRATOR</role>
  </roles>
</userAccount>

```

Request with cURL command line

```

curl -k -i -d @mwhite-UAroles.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/
userAccounts/mwhite"

```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("mwhite-UAroles.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "userAccounts/mwhite")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.setopt(pycurl.INFILESIZE,
    os.path.getsize("mwhite-UAroles.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
POST /mapi/tenants/finance/userAccounts/mwhite HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 120
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```

Example: Changing the data access permissions associated with a user account

Here's a sample **POST** request that changes the data access permissions associated with the user account with the username *pblack*. The new set of permissions is specified in an XML file named *pblack-UAperms.xml*. The file includes permissions for both the Accounts-Receiveable and Accounts-Payable namespaces. The request is made using a tenant-level user account that includes the administrator role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataAccessPermissions>
  <namespacePermission>
    <namespaceName>Accounts-Receiveable</namespaceName>
    <permissions>
      <permission>READ</permission>
      <permission>BROWSE</permission>
      <permission>WRITE</permission>
      <permission>DELETE</permission>
      <permission>PURGE</permission>
      <permission>SEARCH</permission>
    </permissions>
  </namespacePermission>
  <namespacePermission>
    <namespaceName>Accounts-Payable</namespaceName>
    <permissions>
      <permission>READ</permission>
    </permissions>
  </namespacePermission>
</dataAccessPermissions>
```

Request with cURL command line

```
curl -k -i -d @pblack-UAperms.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/userAccounts/
pblack/dataAccessPermissions"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("pblack-UAperms.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
"Authorization: HCP \
```

```

bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"]})
curl_setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "userAccounts/pblack/dataAccessPermissions")
curl_setopt(pycurl.SSL_VERIFYPEER, 0)
curl_setopt(pycurl.SSL_VERIFYHOST, 0)
curl_setopt(pycurl.UPLOAD, 1)
curl_setopt(pycurl.CUSTOMREQUEST, "POST")
curl_setopt(pycurl.INFILESIZE,
    os.path.getsize("pblack-UAperms.xml"))
curl_setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

POST /mapi/tenants/finance/userAccounts/pblack/dataAccessPermissions
HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 572

```

Response headers

```

HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Resetting the security user passwords for a tenant

Here's a sample **POST** request that resets the passwords for the tenant named Finance to *start123*. The empty request body has a content type of JSON. The request is made using a system-level user account that includes the administrator role.

Request with cURL command line

```

curl -k -i -d "{}" -H "Content-Type: application/json"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants/finance/userAccounts
?resetPasswords=start123"

```

Request in Python using PycURL

```

import pycurl
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Authorization: HCP \
    YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
    "https://admin.hcp-ca.example.com:9090/mapi/tenants/finance/" +
    "userAccounts?resetPasswords=start123")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()

```

Request headers

```

POST /mapi/tenants/finance/userAccounts?resetPasswords=start123 HTTP/1.1
Host: admin.hcp.example.com:9090
Accept: */*
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Content-Type: application/json
Content-Length: 2

```

Response headers

```

HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Group account resources

Group account resources let you create, retrieve information about, modify, and delete group accounts. The table below provides information about these resources.

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/groupAccounts				
groupAccount	PUT	Create a group account for a tenant	Tenant-level user account with the security role	You can create a group account only if HCP is configured to support AD.

(Continued)

Data type	Method	Use	Access	Notes
List	GET	Retrieve a list of the group accounts defined for a tenant	Tenant-level user account with the monitor, administrator, or security role	The listed group accounts are identified by the group name. In XML, each listed group account is the value of an element named groupname . In JSON, the name in the name/value pair that lists the group accounts is groupname .
N/A	POST	Add the security role to an existing group account or create a new group account with the security role	System-level user account with the administrator role	For information on the required query parameter for resetting the security group, see “Query parameter for resetting the security group” on page 153.
.../tenants/tenant-name/groupAccounts/group-name				
groupAccount	GET	Retrieve information about a group account	Tenant-level user account with the monitor, administrator, or security role	The information returned depends on the roles associated with the user making the request.
N/A	HEAD	Check for the existence of a group account	Tenant-level user account with the monitor, administrator, or security role	
groupAccount	POST	Modify a group account	Tenant-level user account with the administrator or security role	A user with only the administrator role can modify only the allow- <code>NamespaceManagement</code> property. A user with only the security role cannot modify that property.
N/A	DELETE	Delete a group account	Tenant-level user account with the security role	

(Continued)

Data type	Method	Use	Access	Notes
.../tenants/<i>tenant-name</i>/groupAccounts/<i>group-name</i>/dataAccessPermissions				
dataAccess Permissions	GET	Retrieve information about the data access permissions associated with a group account	Tenant-level user account with the administrator or security role	
	POST	Modify the data access permissions associated with a group account	Tenant-level user account with the administrator role	<p>The request body must contain all permissions granted for each included namespace. If a namespace is not included, its permissions are not changed by the POST request.</p> <p>By default, when you create a group account, it does not include any data access permissions.</p>

Example: Creating a group account

Here's a sample **PUT** request that creates a group account for the Finance tenant. The account is defined in an XML file named `admin-GA.xml`. The name of the group account is specified in the XML file. For the request to be successful, a group with this name must already exist in AD. The request is made using a tenant-level user account that includes the security role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<groupAccount>
  <groupname>hcp-admin@ad.example.com</groupname>
  <roles>
    <role>MONITOR</role>
    <role>ADMINISTRATOR</role>
  </roles>
</groupAccount>
```

Request with cURL command line

```
curl -k -iT admin-GA.xml -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/groupAccounts"
```

Request in Python using PycURL

```
import pycurl
import os
filehandle = open("admin-GA.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "groupAccounts")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("admin-GA.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
PUT /mapi/tenants/finance/groupAccounts HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Content-Type: application/xml
Content-Length: 365
```

Response headers

```
HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```


Example: Retrieving group accounts

Here's a sample **GET** request that retrieves all the group accounts defined for the Finance tenant. The request writes the list of group accounts to a file named `finance-groups.xml`. The request is made using a tenant-level user account that includes the security role.

Request with cURL command line

```
curl -k -H "Accept: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/groupAccounts
?prettyprint" > finance-groups.xml
```

Request in Python using PycURL

```
import pycurl
filehandle = open("finance-groups.xml", 'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Accept: application/xml",
    "Authorization: HCP \
    bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"])
curl.setopt(pycurl.URL,
    "https://finance.hcp.example.com:9090/mapi/tenants/finance/" +
    "groupAccounts?prettyprint")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.WRITEFUNCTION, filehandle.write)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
GET /mapi/tenants/groupAccounts?prettyprint HTTP/1.1
Host: finance.hcp.example.com:9090
Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6
Accept: application/xml
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 307
```

Response body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<groupAccounts>
  <groupname>hcp-admin@ad.example.com</groupname>
  <groupname>hcp-security@ad.example.com</groupname>
  <groupname>hcp-compliance@ad.example.com</groupname>
  <groupname>AR-read@ad.example.com</groupname>
  <groupname>AR-full-access@ad.example.com</groupname>
  <groupname>AP-read@ad.example.com</groupname>
  <groupname>AP-full-access@ad.example.com</groupname>
</groupAccounts>
```

Replication resources

Replication resources let you configure, monitor, and manage replication. The table below provides information about these resources.

Data type	Method	Use	Access	Notes
.../services/replication				
replication Service	GET	Retrieve replication service settings	System-level user account with the monitor or administrator role	
	POST	Modify replication service settings or perform an action on the replication service	System-level user account with the administrator role	For information on query parameters for performing actions on the replication service, see “Query parameters for replication service actions” on page 224.
.../services/replication/links				
link	PUT	Create a replication link	System-level user account with the administrator role	
List	GET	Retrieve a list of the replication links in which the HCP system being queried participates	System-level user account with the monitor or administrator role	The listed links are identified by link name. In XML, each listed link is the value of an element named name . In JSON, the name in the name/value pair that lists the links is name .

(Continued)

Data type	Method	Use	Access	Notes
.../services/replication/links/link-name				
link	GET	Retrieve information about a replication link	System-level user account with the monitor or administrator role	
N/A	HEAD	Check for the existence of a replication link	System-level user account with the monitor or administrator role	
link	POST	Modify or perform an action on a replication link	System-level user account with the administrator role	For information on query parameters for performing actions on replication links, see “Query parameters for replication link actions” on page 172.
N/A	DELETE	Delete a replication link	System-level user account with the administrator role	
.../services/replication/links/link-name/content				
content	GET	Retrieve a list of the items included in a replication link	System-level user account with the monitor or administrator role	
.../services/replication/links/link-name/content/defaultNamespaceDirectories				
List	GET	Retrieve a list of the default-namespace directories included in a replication link	System-level user account with the monitor or administrator role	<p>This resource is available only if the default tenant exists.</p> <p>The listed directories are identified by directory name.</p> <p>In XML, each listed directory is the value of an element named name. In JSON, the name in the name/value pair that lists the directories is name.</p>

(Continued)

Data type	Method	Use	Access	Notes
.../services/replication/links/<i>link-name</i>/content/defaultNamespaceDirectories/<i>directory-name</i>				
N/A	PUT	Add a default-namespace directory to a replication link	System-level user account with the administrator role	
	DELETE	Remove a default-namespace directory from a replication link	System-level user account with the administrator role	
.../services/replication/links/<i>link-name</i>/content/chainedLinks				
List	GET	Retrieve a list of the chained links included in an active/passive link	System-level user account with the monitor or administrator role	<p>Not valid for active/active links.</p> <p>The listed chained links are identified by link name.</p> <p>In XML, each listed chained link is the value of an element named name. In JSON, the name in the name/value pair that lists the chained links is name.</p> <p>This resource is not valid for active/active links.</p>
.../services/replication/links/<i>link-name</i>/content/chainedLinks/<i>link-name</i>				
N/A	PUT	Add a chained link to an active/passive link	System-level user account with the administrator role	You cannot add a chained link to an active/active link.
	DELETE	Remove a chained link from an active/passive link	System-level user account with the administrator role	

(Continued)

Data type	Method	Use	Access	Notes
.../services/replication/links/link-name/content/tenants				
List	GET	Retrieve a list of the HCP tenants included in a replication link	System-level user account with the monitor or administrator role	The listed tenants are identified by tenant name. In XML, each listed tenant is the value of an element named name . In JSON, the name in the name/value pair that lists the tenants is name .
.../services/replication/links/link-name/content/tenants/tenant-name				
N/A	PUT	Add an HCP tenant to a replication link	System-level user account with the administrator role	
tenant (data type for replication link content tenant)	GET	Retrieve replication status information for a tenant included in a replication link	System-level user account with the monitor or administrator role	
N/A	POST	Perform an action on a tenant included in a replication link	System-level user account with the administrator role	For information on query parameters for performing actions on tenants included in replication links, see “Query parameters for replication link content tenant actions” on page 247.
	DELETE	Remove an HCP tenant from a replication link	System-level user account with the administrator role	
.../services/replication/links/link-name/localCandidates				
content	GET	Retrieve a list of the items on the system being queried that are eligible to be included in a replication link	System-level user account with the monitor or administrator role	

(Continued)

Data type	Method	Use	Access	Notes
.../services/replication/links/<i>link-name</i>/localCandidates/defaultNamespaceDirectories				
List	GET	Retrieve a list of the default-namespace directories on the system being queried that are eligible to be included in a replication link	System-level user account with the monitor or administrator role	<p>This resource is available only if the default tenant exists.</p> <p>The listed directories are identified by directory name.</p> <p>In XML, each listed directory is the value of an element named name. In JSON, the name in the name/value pair that lists the directories is name.</p>
.../services/replication/links/<i>link-name</i>/localCandidates/chainedLinks				
List	GET	Retrieve a list of the inbound links on the system being queried that are eligible to be included in an active/passive link	System-level user account with the monitor or administrator role	<p>Not valid for active/active links.</p> <p>The listed chained links are identified by link name.</p> <p>In XML, each listed chained link is the value of an element named name. In JSON, the name in the name/value pair that lists the chained links is name.</p> <p>This resource is not valid for active/active links.</p>
.../services/replication/links/<i>link-name</i>/localCandidates/tenants				
List	GET	Retrieve a list of the HCP tenants on the system being queried that are eligible to be included in a replication link	System-level user account with the monitor or administrator role	<p>The listed tenants are identified by tenant name.</p> <p>In XML, each listed tenant is the value of an element named name. In JSON, the name in the name/value pair that lists the tenants is name.</p>

(Continued)

Data type	Method	Use	Access	Notes
.../services/replication/links/<i>link-name</i>/remoteCandidates				
content	GET	Retrieve a list of the items on the other system involved in a replication link that are eligible to be included in the link	System-level user account with the monitor or administrator role	
.../services/replication/links/<i>link-name</i>/remoteCandidates/defaultNamespaceDirectories				
List	GET	Retrieve a list of the default-namespace directories on the other system involved in a replication link that are eligible to be included in the link	System-level user account with the monitor or administrator role	<p>This resource is available only if the default tenant exists.</p> <p>The listed directories are identified by directory name.</p> <p>In XML, each listed directory is the value of an element named name. In JSON, the name in the name/value pair that lists the directories is name.</p>
.../services/replication/links/<i>link-name</i>/remoteCandidates/chainedLinks				
List	GET	Retrieve a list of the inbound links on the other system involved in an active/passive link that are eligible to be included in the link	System-level user account with the monitor or administrator role	<p>Not valid for active/active links.</p> <p>The listed chained links are identified by link name.</p> <p>In XML, each listed chained link is the value of an element named name. In JSON, the name in the name/value pair that lists the chained links is name.</p> <p>This resource is not valid for active/active links.</p>

(Continued)

Data type	Method	Use	Access	Notes
.../services/replication/links/<i>link-name</i>/remoteCandidates/tenants				
List	GET	Retrieve a list of the HCP tenants on the other system involved in a replication link that are eligible to be included in the link	System-level user account with the monitor or administrator role	The listed tenants are identified by directory name. In XML, each listed tenant is the value of an element named name . In JSON, the name in the name/value pair that lists the tenants is name .
.../services/replication/links/<i>link-name</i>/schedule				
schedule	GET	Retrieve the schedules for an active/active link or the schedule for an active/passive link	System-level user account with the monitor or administrator role	
	POST	Modify the schedules for an active/active link or the schedule for an active/passive link	System-level user account with the administrator role	

Example: Creating a replication link

Here's a sample **PUT** request that creates an active/active link named MA-CA between the local HCP system, hcp-ma.example.com, and a remote HCP system named hcp-ca.example.com. The link definition is in an XML file named MA-CA.xml. The request is made using a system-level user account that includes the administrator role.

Request body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<link>
  <name>MA-CA</name>
  <description>Active/active link between systems in MA and CA</description>
  <type>ACTIVE_ACTIVE</type>
  <connection>
    <remoteHost>replication.admin.hcp-ca.example.com</remoteHost>
  </connection>
</link>
```



```

    <compression>false</compression>
    <encryption>false</encryption>
    <priority>OLDEST_FIRST</priority>
    <failoverSettings>
      <local>
        <autoFailover>true</autoFailover>
        <autoFailoverMinutes>60</autoFailoverMinutes>
      </local>
      <remote>
        <autoFailover>true</autoFailover>
        <autoFailoverMinutes>60</autoFailoverMinutes>
      </remote>
    </failoverSettings>
  </link>

```

Request with cURL command line

```

curl -k -iT MA-CA.xml -H "Content-Type: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication/links"

```

Request in Python using PycURL

```

import pycurl
import os
filehandle = open("MA-CA.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
    "https://admin.hcp-ma.example.com:9090/mapi/services/" +
    "replication/links")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("MA-CA.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```

PUT /mapi/services/replication/links HTTP/1.1
Host: admin.hcp-ma.example.com:9090
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Content-Type: application/xml
Content-Length: 648

```

Response headers

```
HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```

Example: Retrieving a list of eligible local candidates

Here's a sample **GET** request that retrieves a list of the HCP tenants and default-namespace directories on the local system, hcp-ma.example.com, that are eligible to be included in the active/active replication link named MA-CA. The request writes the list of local candidates to a file named local-candidates.xml. The request is made using a system-level user account that includes the administrator role.

Request with cURL command line

```
curl -k -H "Accept: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication/links/
MA-CA/localCandidates?prettyprint" > local-candidates.xml
```

Request in Python using PycURL

```
import pycurl
filehandle = open("local-candidates.xml", 'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Accept: application/xml",
    "Authorization: HCP \
    YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
    "https://admin.hcp-ma.example.com:9090/mapi/services/" +
    "replication/links/MA-CA/localCandidates?prettyprint")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.WRITEFUNCTION, filehandle.write)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
GET /mapi/services/replication/links/MA-CA/localCandidates?prettyprint HTTP/1.1
Host: admin.hcp-ma.example.com:9090
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Accept: application/xml
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 330
```

Response body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content>
  <defaultNamespaceDirectories>
    <name>brochures_2012</name>
    <name>brochures_2013</name>
    <name>brochures_2014</name>
  </defaultNamespaceDirectories>
  <tenants>
    <name>Finance</name>
    <name>HR</name>
    <name>Sales-Mktg</name>
  </tenants>
</content>
```

Example: Adding an HCP tenant to a replication link

Here's a sample **PUT** request that adds the tenant named Finance to the replication link named MA-CA. The request is made using a system-level user account that includes the administrator role.

Request with cURL command line

```
curl -k -iX PUT
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication/links/
MA-CA/content/tenants/Finance"
```

Request in Python using PycURL

```
import pycurl
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Authorization: HCP \
```

```

YWxscm9sZXNM=:04EC9F614D89FF5C7126D32ACB448382"] )
curl_setopt(pycurl.URL,
    "https://admin.cluster-doc4.lab.archivas.com:9090/mapi/" +
    "services/replication/links/4-3/content/tenants/LisaTenant-2")
curl_setopt(pycurl.SSL_VERIFYPEER, 0)
curl_setopt(pycurl.SSL_VERIFYHOST, 0)
curl_setopt(pycurl.CUSTOMREQUEST, "PUT")
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()

```

Request headers

```

PUT /mapi/services/replication/links/MA-CA/content/tenants/Finance HTTP/1.1
User-Agent: curl/7.27.0
Host: admin.hcap-ma.example.com:9090
Authorization: HCP YWxscm9sZXNM=:04EC9F614D89FF5C7126D32ACB448382
Accept: */*

```

Response headers

```

HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0

```

Example: Setting the schedule for a replication link

Here's the a sample **POST** request that sets the local and remote schedules for an active/active link named MA-CA. The new schedule is in an XML file named `schedule_MA-CA.xml`. The request is made using a system-level user account that includes the administrator role.

Request body in XML

```

<schedule>
  <local>
    <scheduleOverride>NONE</scheduleOverride>
    <transition>
      <time>Sun:00</time>
      <performanceLevel>HIGH</performanceLevel>
    </transition>
    <transition>
      <time>Sun:06</time>
      <performanceLevel>MEDIUM</performanceLevel>
    </transition>
    <transition>
      <time>Sat:00</time>
      <performanceLevel>HIGH</performanceLevel>
    </transition>
  </local>
</schedule>

```

```

    <transition>
      <time>Sat:06</time>
      <performanceLevel>MEDIUM</performanceLevel>
    </transition>
  </local>
  <remote>
    <scheduleOverride>NONE</scheduleOverride>
    <transition>
      <time>Sun:00</time>
      <performanceLevel>HIGH</performanceLevel>
    </transition>
    <transition>
      <time>Mon:00</time>
      <performanceLevel>MEDIUM</performanceLevel>
    </transition>
  </remote>
</schedule>

```

Request with cURL command line

```

curl -k -i -d @schedule_MA-CA.xml
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication/links/
MA-CA/schedule"

```

Request in Python using PycURL

```

import pycurl
import os
filehandle = open("schedule_MA-CA.xml", 'rb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
    "https://admin.hcp-ma.example.com:9090/mapi/services/" +
    "replication/links/MA-CA/schedule")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.setopt(pycurl.INFILESIZE,
    os.path.getsize("schedule_MA-CA.xml"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request headers

```
POST /mapi/services/replication/links/MA-CA/schedule HTTP/1.1
User-Agent: curl/7.27.0
Host: admin.hcp-ma.example.com:9090
Accept: */*
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Content-Type: application/xml
Content-Length: 807
```

Response headers

```
HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```

Example: Failing over a replication link

Here's a sample POST request that fails over the link named MA-CA to the local system, hcp-ca.example.com. The request is made using a system-level user account that includes the administrator role.

Request with cURL command line

```
curl -k -iX POST
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ca.example.com:9090/mapi/services/replication/links/
MA-CA?failOver"
```

Request in Python using PycURL

```
import pycurl
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Authorization: HCP \
    YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
    "https://admin.hcp-ca.example.com:9090/mapi/services/" +
    "replication/links/MA-CA?failOver")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.CUSTOMREQUEST, "POST")
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
```

Request headers

```
POST /mapi/services/replication/links/MA-CA?failOver HTTP/1.1
User-Agent: curl/7.27.0
Host: admin.hcp-ca.example.com:9090
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Accept: */*
```

Response headers

```
HTTP/1.1 200 OK
X-HCP-SoftwareVersion: 7.0.0.16
Content-Length: 0
```

Licenses resources

Licensing resources let you upload a license key and retrieve both current license and past license lists.

Data type	Method	Use	Access	Notes
.../storage/licenses				
Licenses	GET	Retrieve either the current storage license or a list of the current and past storage licenses	System-level user account with the monitor or administrator role	HCP does not return information about licensing of economy storage. To display the list of current and past storage licenses, use the verbose query parameter. For more information on this parameter, see Chapter 4, "Query parameter for retrieving a list of licenses." on page 165.
N/A	PUT	Upload a new storage license	System-level user account with the administrator role	The input to a PUT request is the .plk file containing the license to be uploaded.

Example: Retrieving an active storage license list

Here's a sample **GET** request that retrieves the current active storage license information. The request is made using a system-level user account that includes the administrator role.

Request with cURL command line

```
curl -k -i -H "Accept: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/storage/licenses"
```

Request in Python using PycURL

```
import pycurl
filehandle = open("HCPLic_SN12345_Q0123456789_A10TB_01-01-2018.plk",
'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Accept: application/xml",
"Authorization: HCP \
YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
"https://admin.hcp.example.com:9090/mapi/storage/licenses")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.WRITEFUNCTION, filehandle.write)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()
```

Request headers

```
GET /mapi/storage/licenses?prettyprint HTTP/1.1
Host: admin.hcp.example.com:9090
Accept: application/xml
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
```

Response headers

```
HTTP/1.1 200 OK
Content-Type: application/xml
X-HCP-SoftwareVersion: 7.1.0.251
Content-Length: 356
```

Response body in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<licenses>
  <license>
    <activeCapacity>10000000000000</activeCapacity>
    <expirationDate>Jan 1, 2018</expirationDate>
    <extendedCapacity>0</extendedCapacity>
    <quoteNumber>0123456789</quoteNumber>
    <serialNumber>12345</serialNumber>
```



```

        <uploadDate>Aug 14, 2013</uploadDate>
    </license>
</licenses>

```

Example: Uploading a new license

Here's a sample **PUT** request that uploads a storage license. The request is made using a system-level user account that includes the administrator role.

Request with cURL command line

```

curl -i -k -T HCPLic_SN12345_Q0123456789_A10TB_01-01-2018.plk
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
https://admin.hcp.example.com:9090/mapi/storage/licenses

```

Request in Python using PycURL

```

import pycurl
import os
filehandle = open("HCPLic_SN12345_Q0123456789_A10TB_01-01-2018.plk",
'wb')
curl = pycurl.Curl()
curl.setopt(pycurl.HTTPHEADER, ["Content-Type: application/xml",
    "Authorization: HCP \
    YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"])
curl.setopt(pycurl.URL,
    "https://admin.hcp.example.com:9090/mapi/storage/licenses")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE,
    os.path.getsize("HCPLic_SN12345_Q0123456789_A10TB_01-01-2018.plk"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

```

Request header

```

PUT /mapi/services/replication/links HTTP/1.1
Host: admin.hcp.example.com:9090
Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382
Content-Type: application/xml
Content-Length: 356

```

Response header

```
HTTP/1.1 200 OK
Content-Type: application/xml
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-HCP-SoftwareVersion: 7.1.0.251
Content-Length: 0
```

Paging, sorting, and filtering

By default, in response to a **GET** request for:

- The tenants resource, HCP returns a list of all the tenants defined in the system
- The namespaces resource, HCP returns a list of all the namespaces owned by the applicable tenant
- The userAccounts resource, HCP returns a list of all the user accounts defined for the applicable tenant
- The dataAccessPermissions resource, HCP returns a list of all the namespaces for which the applicable user account or group account has any permissions, along with the permissions granted for each of those namespaces

With very large numbers of these items, such requests can overload or reduce the efficiency of the client. Additionally, if you're interested in only a small number of the listed items, finding the information you want can be difficult.

To manage the results of **GET** requests, you can use query parameters to page through, sort, and filter these resource lists.

Paging through the resource lists

You can limit the number of items HCP returns in response to an individual **GET** request by specifying an offset into the applicable list and a count of the items to return. By issuing multiple such requests, you can retrieve the entire list, one manageable number of items at a time. This is called **paging**.

To specify the offset and count, you use these query parameters in the **GET** request:

- **offset=offset** specifies the offset of the first item to include in the returned tenant or namespace list.
- **count=count** specifies the number of items to return.

These considerations apply to paging through resource lists:

- The **offset** and **count** parameters are valid only with the tenants, namespaces, userAccounts, and dataAccessPermissions resources.
- The first item in the complete list is at offset zero. So, for example, to retrieve the first five items, you would specify **offset=0** and **count=5**. To retrieve the sixth through tenth items, you would specify **offset=5** and **count=5**.
- If you omit the **offset** parameter, the returned list starts with the item at offset zero.
- If you specify an offset that is greater than or equal to the number of items in the complete list, no items are returned.
- If you omit the **count** parameter, the returned list includes all items starting from the offset and continuing through the end of the complete list.
- If you specify a count that is greater than the number of items remaining in the complete list after the offset, the returned list includes all items starting from the offset and continuing through the end of the complete list.
- If an item is deleted while you are paging through the applicable list, the full list of items returned may be incomplete. For example, suppose the system has six tenants (A, B, C, D, E, and F):
 1. You issue a **GET** request for the tenants resource with these query parameters: **offset=0&count=3**

HCP returns A, B, and C.
 2. Tenant B is deleted without your knowledge.

The remaining tenants are A, C, D, E, and F.

3. You issue a second **GET** request for the tenants resource with these query parameters: **offset=3&count=3**

HCP returns E and F. It does not return D because D is now at offset two.

- You can page, sort, and filter resource lists in the same request. For information sorting, see [“Sorting resource lists”](#) below. For information on filtering, see [“Filtering resource lists”](#) on page 101.



Tip: By default, resource lists are returned in an arbitrary order, and the order may not be the same if the request is issued more than once. To ensure that you get all the expected items when paging, specify a sort order in each request.

Sorting resource lists

You can retrieve a sorted list of items of a particular resource type by specifying the property you want to sort by and the sort order in the **GET** request. You can sort tenants and namespaces by name or by hard quota. You can sort user accounts by username.

To specify a sort property and sort order, you use these query parameters in the **GET** request:

- **sortType=property-name** specifies the property you want to sort by. Valid values are:
 - For the tenants, namespaces, and dataAccessPermissions resources, **name** and **hardQuota**
 - For the userAccounts resource, **username**

This parameter is optional. For the tenants, namespaces, and dataAccessPermissions resources, the default is **name**.

- **sortOrder=(ascending|descending)** specifies the order in which to sort the listed items.

This parameter is optional. The default is **ascending**.

For example, this **GET** request sorts the list of tenants defined in the HCP system in descending order by hard quota:

```
curl -k -i -H "Accept: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants?sortType=hardQuota
&sortOrder=descending&prettyprint"
```

These considerations apply to sorting resource lists:

- The **sortType** and **sortOrder** parameters are valid only with the tenants, namespaces, userAccounts, and dataAccessPermissions resources.
- You can page, sort, and filter resource lists in the same request. For information on paging, see ["Paging through the resource lists"](#) above. For information on filtering, see ["Filtering resource lists"](#) below.

Filtering resource lists

You can retrieve a subset of items of a particular resource type by specifying a filter in the **GET** request. To apply a filter, you specify the property you want to filter by and the text string to use as the filter. The filtered list includes only those items for which the value of the specified property begins with or is the same as the specified text string.

You can filter tenants and namespaces by name or by tag. You can filter user accounts by username.

To specify a filter, you use these query parameters in the **GET** request:

- **filterType=property-name** specifies the property by which to filter the resource list. Valid values for *property-name* are:
 - For the tenants, namespaces, and dataAccessPermissions resources, **name** and **tag**
 - For the userAccounts resource, **username**

The **filterType** parameter is optional. For the tenants, namespaces, and dataAccessPermissions resources, the default is **name**.

- **filterString=text-string** specifies the text string to use as the filter. This string is not case sensitive.

Text strings can be at most 64 characters long and can contain any valid UTF-8 characters except commas (,). White space is allowed and must be percent encoded.

For example, this **GET** request filters the list of namespaces for the Finance tenant by names beginning with the string *accounts*:

```
curl -k -i -H "Accept: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces
?filterType=name&filterString=accounts&prettyprint"
```

These considerations apply to filtering resource lists:

- The **filterType** and **filterString** parameters are valid only with the tenants, namespaces, userAccounts, and dataAccessPermissions resources.
- If a **GET** request includes the **filterType** parameter but does not include the **filterString** parameter, HCP returns the complete list of items of the applicable type.
- You can page, sort, and filter resource lists in the same request. For information on paging, see [“Paging through the resource lists”](#) on page 98. For information sorting, see [“Sorting resource lists”](#) above.

Data types

Many of the data types that describe HCP management API resources and properties are named, unordered sets of properties. When you create a resource, some properties for the applicable data type are required and some are optional. You need to specify a value for each required property. If you omit an optional property, HCP supplies a default value for it.

When you modify a resource, all properties for the applicable data type are optional. If you omit a property, the current value of the property remains unchanged.

When you create or modify a resource, HCP 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 whose values cannot be set with the user account used for the request



Note: If a tenant has granted system-level users administrative access to itself, actions that can be performed with a tenant-level user account can also be performed with a system-level user account that includes the applicable roles.

Some resources also have required or optional query parameters. For information on specifying query parameters, see [“Query parameters”](#) on page 15.

Common property values

These types of values, among others, are common to multiple HCP management API resources or properties:

- Passwords
- Descriptions
- Boolean values
- Permission lists

Passwords

User accounts have a password property. When creating or modifying a user account, the password is specified by the **password** query parameter.

When you create a tenant, you have the option of creating an initial user account for that tenant. In this case, you use query parameters, including the **password** parameter, to define that account.

To reset the passwords for all locally authenticated user accounts with the security role, you specify the new password in the **resetPasswords** query parameter.

When you enable WebDAV basic authentication, you need to specify a username and password. In this case, you use the `webdavBasicAuthPassword` property of the `httpProtocol` data type to specify the password.

Passwords can be up to 64 characters long, are case sensitive, and can contain any valid UTF-8 characters, including white space. To be valid, a password must include at least one character from two of these three groups: alphabetic, numeric, and other.

The minimum password length is configurable. To set the minimum password length, you use the `minimumPasswordLength` property of the `consoleSecurity` data type. For more information on that property, see ["consoleSecurity"](#) on page 127.

Descriptions

Namespaces, namespace defaults, retention classes, and user accounts all have a description property. Tenants have two properties for descriptions — `systemVisibleDescription` and `tenantVisibleDescription`.

All descriptions can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space.

All description properties are optional. If you omit the property when you create a tenant, retention class, or user account, the description is blank. If you omit the property when you create a namespace, the description defaults to the description specified by the namespace defaults for the tenant.

Boolean values

For properties that take a Boolean value, valid values are:

- **true**, **t**, or **1** (one)
- **false**, **f**, or **0** (zero)

These values are not case sensitive.

Invalid values are interpreted as false.

Permission lists

These resources and properties have values that are lists of permissions:

- The permissions resource for tenants (specifies the permissions in the data access permission mask for a tenant)
- The permissions resource for namespaces (specifies the permissions in the data access permission mask for a namespace)
- The permissions property of the namespacePermission data type (used in the specification of the data access permissions for a user or group account)
- The `authMinimumPermissions` and `authAndAnonymousMinimumPermissions` properties of the namespaces data type

The permissions that can be included in a list differ for the various resources and properties. However, in each case:

- In XML, each listed permission is the value of an element named **permission**. For example:

```
<permissions>
  <permission>READ</permission>
  <permission>WRITE</permission>
  <permission>DELETE</permission>
  <permission>PURGE</permission>
  <permission>SEARCH</permission>
</permissions>
```

- In JSON, the name in the name/value pair that lists the permissions is **permission**. For example:

```
"permissions" : {
  "permission" : [ "READ", "WRITE", "DELETE", "PURGE", "SEARCH" ]
}
```

availableServicePlan

The availableServicePlan data type describes the availableServicePlans resource for tenants.

availableServicePlan data type properties

The table below describes the properties included in the availableServicePlan data type.

Property name	Data type	Description	Notes
description	String	Specifies the description of the service plan.	
name	String	Specifies the name of the service plan.	

availableServicePlan data type example

Here's an XML example of the availableServicePlan data type:

```
<availableServicePlan>
  <name>Platinum</name>
  <description>Most highly available, best performance -- suitable for data that will
    be frequently accessed or modified throughout its life</description>
</availableServicePlan>
```

chargebackData

The chargebackData data type describes the chargebackData property of the chargebackReport data type. The chargebackReport data type is used to generate chargeback reports.

Most of the properties in the chargebackData data type represent statistics that describe the usage of a given namespace or of all the namespaces owned by a given tenant. In a chargeback report, each set of these statistics applies to one namespace or tenant during a given reporting interval (for example, one hour or one day).

Chargeback statistics either reflect a point in time or are dynamic. Point-in-time statistics are measurements taken at the end of a reporting interval, such as the used storage capacity for a namespace at the end of an hour. Dynamic statistics are measurements, such as the number of reads or writes to a namespace, that are accumulated over time.

HCP accumulates dynamic statistics on an hourly basis, starting at the beginning of each hour. So, for example, if the reporting interval is an hour, one statistic might represent the number of successful writes to a namespace that occurred between 11:00:00 and 11:59:59. If the reporting interval is a day, each reported dynamic statistic is the sum of the hourly values for the day, where the day starts at 00:00:00 and ends at 23:59:59.

For more information on chargeback reports, see ["chargebackReport"](#) on page 112.

chargebackData data type properties

The table below describes the properties included in the chargebackData data type. When the output format for a chargebackReport resource is CSV, the properties included in the chargebackData data type are ordered. This table lists the properties in the order in which they appear in a CSV response body.



Note: For information on additional properties that are included in the chargebackData data type when the request for the chargebackReport resource is made using a system-level user account, see *Administering HCP*.

Property name	Data type	Description	Notes
systemName	String	One of: <ul style="list-style-type: none"> The name of the domain associated with the [hcp_system] network for the HCP system to which the set of aggregated statistics in the line applies The name of the domain associated with the data access network for the tenant to which the set of aggregated statistics in the line applies The name of the domain associated with the data access network for the tenant that owns the namespace to which the set of statistics in the line applies 	
tenantName	String	Either: <ul style="list-style-type: none"> The name of the tenant to which the set of statistics applies The name of the tenant that owns the namespace to which the set of statistics applies 	

(Continued)

Property name	Data type	Description	Notes
namespaceName	String	The name of the namespace to which the set of statistics applies.	This property is included only in chargeback reports generated for namespaces. It is not included in chargeback reports generated for tenants.
startTime	String	<p>The start time of the reporting interval to which the set of statistics applies, in this ISO 8601 format:</p> <p><i>yyyy-MM-ddThh:mm:ssZ</i></p> <p>Z represents the offset from UTC, in this format:</p> <p><i>(+ -)hhmm</i></p> <p>For example:</p> <p>2012-02-18T14:00:00-0500</p>	
endTime	String	The end time of the reporting interval to which the set of statistics applies, in the same ISO 8601 format as is used for the startTime property.	
objectCount	Long	The number of objects in the given namespace or in all the namespaces owned by the given tenant.	This is a point-in-time statistic.
ingestedVolume	Long	The total size of the stored data and custom metadata, in bytes, before it was added to the given namespace or to all the namespaces owned by the given tenant.	This is a point-in-time statistic.
storageCapacityUsed	Long	The total number of bytes occupied by stored data in the given namespace or in all the namespaces owned by the given tenant. This includes object data, metadata, and any redundant data required to satisfy the applicable service plan.	This is a point-in-time statistic.

(Continued)

Property name	Data type	Description	Notes
bytesIn	Long	<p>The total number of bytes successfully written to the given namespace or to all the namespaces owned by the given tenant during the reporting interval.</p> <p>If data was compressed before being transmitted, this is the number of bytes before compression.</p>	This is a dynamic statistic.
bytesOut	Long	<p>The total number of bytes read from the given namespace or from all the namespaces owned by the given tenant during the reporting interval.</p> <p>If data (including XML for directory listings) was compressed before being transmitted, this is the number of bytes before compression.</p>	This is a dynamic statistic.
reads	Long	The total number of read operations performed in the given namespace or in all the namespaces owned by the given tenant during the reporting interval.	This is a dynamic statistic.
writes	Long	The total number of write operations successfully performed in the given namespace or in all the namespaces owned by the given tenant during the reporting interval.	This is a dynamic statistic.
deletes	Long	The total number of delete and purge operations performed in the given namespace or in all the namespaces owned by the given tenant during the reporting interval.	This is a dynamic statistic.

(Continued)

Property name	Data type	Description	Notes
deleted	String	<p>One of:</p> <ul style="list-style-type: none"> false — The namespace or tenant currently exists. included — For a tenant only, the statistics in the set include values for one or more namespaces that were subsequently deleted. 	
valid	Boolean	<p>The status of the set of statistics in the line. Possible values are:</p> <ul style="list-style-type: none"> true — HCP successfully collected all statistics in the set. false — The statistics in the set do not reflect all the activity that occurred during the reporting interval. This may be due, for example, to one or more nodes being unavailable during that time, to a network failure, or to other hardware issues. 	

chargebackData data type example

Here's an XML example of the chargebackData data type:

```

<chargebackData>
  <bytesIn>2905</bytesIn>
  <bytesOut>561</bytesOut>
  <deleted>>false</deleted>
  <deletes>0</deletes>
  <endTime>2012-02-18T13:59:59-0500</endTime>
  <ingestedVolume>692224</ingestedVolume>
  <namespaceName>Accounts-Receivable</namespaceName>
  <objectCount>1240</objectCount>
  <reads>1</reads>
  <startTime>2012-02-18T13:00:00-0500</startTime>

```

```

<storageCapacityUsed>716720</storageCapacityUsed>
<systemName>hcp.example.com</systemName>
<tenantName>Finance</tenantName>
<valid>true</valid>
<writes>5</writes>
</chargebackData>

```

chargebackReport

The chargebackReport data type describes the chargebackReport resource for tenants and namespaces. You use this resource to generate chargeback reports for tenants and namespaces.

About chargeback reports

A **chargeback report** contains historical statistics about a tenant or namespace, broken out either by hour or by day. You can also generate chargeback reports that contain a single set of statistics for a given time period, such as a specific month.

Chargeback reports can serve as input to billing systems that need to determine charges for capacity and bandwidth usage at the tenant or namespace level. Because a chargeback report can cover a specified time period, you can create applications that generate these reports at regular intervals and feed those reports to your billing system.



Tip: After a tenant or namespace is deleted, you can no longer generate chargeback reports for it. Therefore, to ensure that you don't lose usage statistics, you should take this fact into consideration when setting the regular interval at which to generate these reports.

Chargeback reports are also a good source of information for system analysis, enabling you to adjust storage and bandwidth allocations based on usage patterns.

Chargeback reports are available only for HCP tenants and namespaces. You cannot generate a chargeback report for the default tenant or namespace.

A chargeback report for a namespace contains statistics only for that namespace. A chargeback report for a tenant contains aggregated namespace statistics. For example, the number of read operations for a tenant during a given reporting interval is the total of the numbers of read operations that occurred in all the namespaces owned by that tenant during that reporting interval. For information on how chargeback data is collected, see ["chargebackData"](#) on page 107.

You can use a system-level user account to request a tenant chargeback report regardless of whether the tenant has granted system-level users administrative access to itself. To generate a namespace chargeback report using a system-level user account, system-level users must have administrative access to the owning tenant.

When generating a chargeback report, you use query parameters on the resource URL in the **GET** request to specify the reporting interval and the time period you want the report to cover. HCP keeps chargeback statistics for 180 days. As a result, chargeback reports cannot report statistics from more than 180 days in the past. For information on the chargeback query parameters, see ["Query parameters for generating chargeback reports"](#) on page 114.



Note: If you upgraded HCP less than 180 days ago from a release that does not support chargeback reports, the earliest available statistics are from the time the upgrade was completed.

The response to a chargebackReport **GET** request can be formatted as XML, JSON, or CSV. For an example of the response to a request for a chargeback report in XML format, see ["chargebackReport data type example"](#) below. For an example of the response to a request for a chargeback report in CSV format, see ["Sample chargeback report"](#) on page 117.

chargebackReport data type property

The table below describes the property included in the chargebackReport data type.

Property name	Data type	Description	Notes
chargebackData	chargeback Data	Specifies statistics that describe the usage of a given namespace or of all the namespaces owned by a given tenant. For details, see "chargebackData" on page 107.	The response body includes one instance of this property for each reporting interval in the time period specified by the query parameters in the GET request for the chargebackReport resource.

chargebackReport data type example

Here's an XML example of the chargebackReport data type:

```
<chargebackReport>
  <chargebackData>
    <bytesIn>2905</bytesIn>
    <bytesOut>561</bytesOut>
    <deleted>false</deleted>
    <deletes>0</deletes>
    <endTime>2012-02-18T13:59:59-0500</endTime>
    <ingestedVolume>692224</ingestedVolume>
    <namespaceName>Accounts-Receivable</namespaceName>
    <objectCount>1240</objectCount>
    <reads>1</reads>
    <startTime>2012-02-18T13:00:00-0500</startTime>
    <storageCapacityUsed>716720</storageCapacityUsed>
    <systemName>hcp.example.com</systemName>
    <tenantName>Finance</tenantName>
    <valid>true</valid>
    <writes>5</writes>
  </chargebackData>
  <chargebackData>
    <bytesIn>3927</bytesIn>
    <bytesOut>0</bytesOut>
    <deleted>false</deleted>
    <deletes>0</deletes>
    <endTime>2012-02-18T14:59:59-0500</endTime>
    <ingestedVolume>705357</ingestedVolume>
    <namespaceName>Accounts-Receivable</namespaceName>
    <objectCount>1247</objectCount>
    <reads>0</reads>
    <startTime>2012-02-18T14:00:00-0500</startTime>
    <storageCapacityUsed>720896</storageCapacityUsed>
    <systemName>hcp.example.com</systemName>
    <tenantName>Finance</tenantName>
    <valid>true</valid>
    <writes>7</writes>
  </chargebackData>
</chargebackReport>
```

Query parameters for generating chargeback reports

You use query parameters appended to the request URL to specify the time period and reporting interval for a chargeback report. These parameters are all optional. Default values are used for any you omit.

Time period

To specify the time period you want the report to cover, use these query parameters:

- **start=***start-time* specifies the beginning of the time period. These considerations apply:
 - With a reporting interval of **hour** or **total**, if you specify a start time that is not on an hour break, the first reporting interval in the report is the hour that includes the specified start time. For example, if you specify a start time of 9:45:00, the first reporting interval in the report starts at 9:00:00.
 - With a reporting interval of **day**, if you specify a start time that is not on a day break, the first reporting interval in the report is the day that includes the specified start time. For example, if you specify a start time of 9:45:00 on October 6th, the first reporting interval in the report starts at 00:00:00 on October 6th.
 - If you specify a start time that is earlier than the time of the earliest available chargeback statistics or if you omit the **start** parameter, the first reporting interval in the report is the interval that includes the earliest available statistics.
- **end=***end-time* specifies the end of the time period. These considerations apply:
 - With a reporting interval of **hour** or **total**, the last reporting interval in the report is the hour that includes the specified end time. For example, if you specify an end time of 9:00:00, the last reporting interval in the report ends at 9:59:59.
 - With a reporting interval of **day**, the last reporting interval in the report is the day that includes the specified end time. For example, if you specify an end time of 00:00:00 on October 6th, the last reporting interval in the report ends at 23:59:59 on October 6th.
 - If you specify an end time that is later than the current time or if you omit the **end** parameter, the last reporting interval in the report is the interval that includes the current time. The point-in-time statistics reported for this interval are the statistics at the current time. The dynamic statistics are the statistics accumulated so far for the interval.

If you specify both a start time and an end time, the start time must be earlier than the end time.

For **start-time** and **end-time**, use the ISO 8601 format, which is described in [“Specifying retention values”](#) on page 277.

Reporting interval

To specify the reporting interval for a chargeback report, use the **granularity** query parameter. Valid values for this parameter are:

- **hour** — The reporting interval is one hour.
- **day** — The reporting interval is one day.
- **total** — The reporting interval is the time period defined by the start and end times for the report. In this case, the report contains a single set of chargeback statistics.

The default is **total**.

These values are not case sensitive.



Note: The statistics reported for the current reporting interval, if included in the chargeback report, may not reflect some reads and writes that have already occurred during the current hour. After the hour is past, however, the statistics for that hour are complete.

Query parameter examples

Here are some examples of combining query parameters to get specific chargeback reports:

- To get hourly statistics for the entire day of February 18, 2012:

```
start=2012-02-18T00:00:00-0500&end=2012-02-18T23:59:59-0500
&granularity=hour
```

- To get daily statistics for the week starting February 19, 2012:

```
start=2012-02-19T00:00:00-0500&end=2012-02-25T23:59:59-0500
&granularity=day
```

- To get a single set of statistics for the entire month of February 2012:

```
start=2012-02-01T00:00:00-0500&end=2012-02-29T23:59:59-0500
&granularity=total
```

- To get hourly statistics for the current day from 8:00 a.m. up to the current time, where the current day is February 22, 2012:

`start=2012-02-22T08:00:00-0500&granularity=hour`

Sample chargeback report

The next page shows an example of a chargeback report for the Accounts-Receiveable namespace owned by the Finance tenant in the HCP system named `hcp.example.com`. The report covers two days, February 7, 2013, and February 8, 2013, and has a reporting interval of a day. The report is shown as it would appear in a spreadsheet.

systemName	tenantName	namespaceName	startTime	endTime	objectCount	ingestedVolume	storageCapacityUsed	bytesIn	bytesOut	reads	writes	deletes	deleted	valid
hcp.example.com	Finance	Accounts-Receivable	2013-02-07T00:00:00-0500	2013-02-07T23:59:59-0500	1616	858071	884656	3399	11090	12	7	0	FALSE	TRUE
hcp.example.com	Finance	Accounts-Receivable	2013-02-08T00:00:00-0500	2013-02-08T23:59:59-0500	1623	868998	897224	10927	45247	17	9	0	FALSE	TRUE

cifsProtocol

The cifsProtocol data type describes the cifs resource for HCP namespaces.

cifsProtocol data type properties

The table below describes the properties included in the cifsProtocol data type.

Property name	Data type	Description	Notes
caseForcing	Boolean	<p>Specifies whether the CIFS protocol is case forcing for the namespace. Valid values are:</p> <ul style="list-style-type: none"> UPPER — The protocol changes object names to all uppercase in requests it passes to HCP. LOWER — The protocol changes object names to all lowercase in requests it passes to HCP. DISABLED — The protocol is not case forcing. <p>The default is DISABLED.</p> <p>These values are not case sensitive.</p>	
caseSensitive	Boolean	<p>Specifies whether the CIFS protocol is case sensitive for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — The protocol is case sensitive. false — The protocol is not case sensitive. <p>The default is true.</p>	

(Continued)

Property name	Data type	Description	Notes
enabled	Boolean	<p>Specifies whether the CIFS protocol is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — CIFS is enabled. false — CIFS is disabled. <p>The default is false.</p>	
ipSettings	ipSettings	<p>Specifies which IP addresses can and cannot access the namespace through the CIFS protocol.</p> <p>For details, see “ipSettings” on page 160.</p>	
requiresAuthentication	Boolean	<p>Specifies whether user authentication is required or allowed for access to the namespace through the CIFS protocol. Valid values are:</p> <ul style="list-style-type: none"> true — User authentication is required. false — User authentication is allowed. <p>The default is true.</p>	This property can be set to true only if the owning tenant supports AD authentication.

cifsProtocol data type example

Here's an XML example of the cifsProtocol data type:

```

<cifsProtocol>
  <caseForcing>DISABLED</caseForcing/>
  <caseSensitive>true</caseSensitive>
  <enabled>true</enabled>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.140.10</ipAddress>
      <ipAddress>192.168.140.14</ipAddress>
      <ipAddress>192.168.140.15</ipAddress>
      <ipAddress>192.168.149.0/24</ipAddress>
    
```



```

    </allowAddresses>
    <denyAddresses>
      <ipAddress>192.168.149.5</ipAddress>
    </denyAddresses>
  </ipSettings>
  <requiresAuthentication>true</requiresAuthentication>
</cifsProtocol>

```

complianceSettings

The complianceSettings data type describes the complianceSettings resource for namespaces.

complianceSettings data type properties

The table below describes the properties included in the complianceSettings data type.

Property name	Data type	Description	Notes
customMetadataChanges	String	<p>Specifies which operations are allowed with custom metadata for objects under retention. Valid values are:</p> <ul style="list-style-type: none"> ADD — Allows custom metadata to be added for objects under retention but not replaced or deleted ALL — Allows custom metadata to be added, replaced, and deleted for objects under retention NONE — Disallows all custom metadata operations for objects under retention <p>The default is ADD.</p> <p>These values are not case sensitive.</p>	

(Continued)

Property name	Data type	Description	Notes
dispositionEnabled	Boolean	<p>Specifies whether objects with expired retention periods should automatically be deleted from the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Disposition is enabled. false — Disposition is not enabled. <p>The default is false.</p>	For disposition to take effect at the namespace level, it must also be enabled in the HCP system configuration.
retentionDefault	String	<p>For an HCP namespace, specifies the default retention setting for objects added to the namespace. Valid values are special values, offsets, retention classes, and fixed dates. For information on specifying these values, see “Specifying retention values” on page 277.</p> <p>The default is 0 (zero), which is equivalent to Deletion Allowed.</p>	This property is not valid for the default namespace.
shreddingDefault	Boolean	<p>For an HCP namespace, specifies the default shred setting for objects added to the namespaces. Valid values are:</p> <ul style="list-style-type: none"> true — New objects are marked for shredding. false — New objects are not marked for shredding. <p>The default is false.</p>	<p>This property is not valid for the default namespace.</p> <p>Once an object is marked for shredding, its shred setting cannot be changed.</p>

complianceSettings data type example

Here's an XML example of the complianceSettings data type:

```
<complianceSettings>  
  <customMetadataChanges>ALL</customMetadataChanges>  
  <retentionDefault>A+7Y</retentionDefault>  
  <dispositionEnabled>true</dispositionEnabled>  
  <shreddingDefault>false</shreddingDefault>  
</complianceSettings>
```

connection

The connection data type describes the connection property of the link data type.

connection data type properties

The table below describes the properties included in the connection data type.

Property name	Data type	Description	Notes
localhost	String	<p>Identifies the local system for the replication link. Valid values are:</p> <ul style="list-style-type: none"> The domain name for the remote system for the link to use for communication with the local system (as that name is known to the remote system), in either of these formats: <code>replication.hcp-domain-name</code> <code>replication.admin.hcp-domain-name</code> One or more comma-separated IP addresses of storage nodes in the local system (as those addresses are known to the remote system) <p>For more information on these values, see the description of the remoteHost property.</p> <p>Typically, you specify this property on a PUT request to create a link only if the system on which you're creating the link uses network address translation (NAT) for communication with the other system.</p>	This property is optional on a PUT request to create a replication link.

(Continued)

Property name	Data type	Description	Notes
localPort	Integer	<p>Specifies the port on which the local system for the replication link listens for data from the remote system. The default is 5748.</p> <p>Typically, you specify a different port only if other port usage makes it necessary.</p>	This property is optional on a PUT request to create a replication link.

(Continued)

Property name	Data type	Description	Notes
remoteHost	String	<p>Identifies the remote system for the replication link. Valid values are:</p> <ul style="list-style-type: none"> The domain name of the remote system, in either of these formats: <code>replication.hcp-domain-name</code> <code>replication.admin.hcp-domain-name</code> <code>hcp-domain-name</code> must be the name of the domain associated with the network that's selected for replication on the remote system. The second format is required if the domain for the replication network is shared with other networks. One or more comma-separated IP addresses of storage nodes in the remote system. These must be the node IP addresses in the network that's selected for replication on the remote system. <p>The local system for the link transmits data only to the nodes identified by the domain name or IP addresses you specify. Therefore, you should specify IP addresses only if you have a compelling reason to do so (for example, HCP is not using DNS, or you need to reduce the processing load on some number of nodes).</p>	This property is required on a PUT request to create a replication link.

(Continued)

Property name	Data type	Description	Notes
remotePort	Integer	<p>Specifies the port on which the remote system for the replication link listens for data from the local system. The default is 5748.</p> <p>Typically, you specify a different port only if other port usage makes it necessary.</p>	This property is optional on a PUT request to create a replication link.

connection data type example

Here's an XML example of the connection data type:

```
<connection>
  <localHost>
    192.168.210.16, 192.168.210.17, 192.168.210.18, 192.168.210.19
  </localHost>
  <localPort>5748</localPort>
  <remoteHost>replication.admin.hcp-ca.example.com</remoteHost>
  <remotePort>5748</remotePort>
</connection>
```

consoleSecurity

The consoleSecurity data type describes the consoleSecurity resource for tenants.

consoleSecurity data type properties

The table below describes the properties included in the consoleSecurity data type.

Property name	Data type	Description	Notes
disableAfterAttempts	Integer	Specifies the number of times a locally authenticated or RADIUS-authenticated user can enter an incorrect password before the user account is automatically disabled. Valid values are integers in the range zero through 999. The default is five. A value of zero means accounts are never disabled due to failed login attempts.	If the last locally authenticated user account with the security role is disabled due to failed login attempts and no group accounts have the security role, the user account is reenabled automatically after one hour.
disableAfterInactiveDays	Integer	Specifies the number of days an HCP user account can remain inactive before it's automatically disabled. Valid values are integers in the range zero through 999. The default is 180 days. A value of zero means accounts are never automatically disabled due to inactivity.	If no group accounts have the security role, the last locally authenticated user account with the security role is not automatically disabled due to inactivity.
forcePasswordChange Days	Integer	Specifies the number of days after which locally authenticated users are automatically forced to change their passwords. Valid values are integers in the range zero through 999,999. The default is 180 days. A value of zero means users are never automatically forced to change their passwords.	

(Continued)

Property name	Data type	Description	Notes
ipSettings	ipSettings	Specifies which IP addresses can and cannot access the Tenant Management Console. For details, see “ipSettings” on page 160.	
loginMessage	String	Specifies message text to appear on the login page of the Tenant Management and Search Consoles. 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.	
logoutOnInactive	Integer	Specifies the number of minutes a Tenant Management or Search Console session started with an explicit login can be inactive before it times out. Valid values are integers in the range zero through 999. The default is ten.	
minimumPasswordLength	Integer	Specifies the minimum number of characters for user account passwords. Valid values are integers in the range two through 64. The default is six.	

consoleSecurity data type example

Here’s an XML example of the consoleSecurity data type:

```

<consoleSecurity>
  <disableAfterAttempts>3</disableAfterAttempts>
  <disableAfterInactiveDays>30</disableAfterInactiveDays>
  <forcePasswordChangeDays>45</forcePasswordChangeDays>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.103.18</ipAddress>
      <ipAddress>192.168.103.24</ipAddress>
      <ipAddress>192.168.103.25</ipAddress>
    </allowAddresses>
    <allowIfInBothLists>false</allowIfInBothLists>
  </ipSettings>
</consoleSecurity>

```

contactInfo

```
<denyAddresses/>
</ipSettings>
<loginMessage></loginMessage>
<logoutOnInactive>20</logoutOnInactive>
<minimumPasswordLength>6</minimumPasswordLength>
</consoleSecurity>
```

contactInfo

The contactInfo data type describes the contactInfo resource for tenants.

contactInfo data type properties

The table below describes the properties included in the contactInfo data type. All of these properties are optional. The property values can contain any valid UTF-8 characters, including white space.

Property name	Data type	Description	Notes
address1	String	Specifies the first line of an address for the tenant contact	
address2	String	Specifies the second line of an address for the tenant contact	
city	String	Specifies the city for the tenant contact	
countryOrRegion	String	Specifies the country for the tenant contact	
emailAddress	String	Specifies a valid email address for the tenant contact	
extension	String	Specifies a telephone number extension for the tenant contact	
firstName	String	Specifies the first name of the tenant contact	
lastName	String	Specifies the last name of the tenant contact	
primaryPhone	String	Specifies a telephone number for the tenant contact	Do not include a telephone number extension. Use the extension property for that.

(Continued)

Property name	Data type	Description	Notes
state	String	Specifies the state or province for the tenant contact	
zipOrPostalCode	String	Specifies the postal code for the tenant contact	

contactInfo data type example

Here's an XML example of the contactInfo data type:

```
<contactinfo>
  <address1>Exmple Co., Finance Dept.</address1>
  <address2>10 Main St.</address2>
  <city>Anytown</city>
  <countryOrRegion>USA</countryOrRegion>
  <emailAddress>lgreen@example.com</emailAddress>
  <extension>123</extension>
  <firstName>Lee</firstName>
  <lastName>Green</lastName>
  <zipOrPostalCode>12345</zipOrPostalCode>
  <primaryPhone>555-555-5555</primaryPhone>
  <state>MA</state>
</contactinfo>
```

content

The content data type describes the content, localCandidates, and remoteCandidates resources for replication links.

content data type properties

The table below describes the properties included in the content data type.

Property name	Data type	Description	Notes
defaultNamespace Directories	List	Lists the default-namespace directories that are candidates for or included in the replication link, as applicable.	<p>This property is returned by a GET request only if the default tenant exists.</p> <p>The listed directories are identified by directory name.</p> <p>In XML, each listed directory is the value of an element named name. In JSON, the name in the name/value pair that lists the directories is name.</p>
chainedLinks	List	For an active/passive link, lists the chained links that are candidates for or included in the replication link, as applicable.	<p>This property is not returned by any GET request for the candidates for or content of an active/active link.</p> <p>The listed chained links are identified by link name.</p> <p>In XML, each listed chained link is the value of an element named name. In JSON, the name in the name/value pair that lists the chained links is name.</p>
tenants	List	Lists the HCP tenants that are candidates for or included in the replication link, as applicable.	<p>The listed tenants are identified by tenant name.</p> <p>In XML, each listed tenant is the value of an element named name. In JSON, the name in the name/value pair that lists the tenants is name.</p>

content data type example

Here's an XML example of the content data type:

```
<content>
  <defaultNamespaceDirectories>
    <name>brochures_2014</name>
  </defaultNamespaceDirectories>
  <chainedLinks/>
  <tenants>
    <name>Finance</name>
    <name>HR</name>
    <name>Sales-Mktg</name>
  </tenants>
</content>
```

contentClass

The contentClass data type describes the contentClasses resource for tenants.

contentClass data type properties

The table below describes the properties included in the contentClass data type.

Property name	Data type	Description	Notes
contentProperties	content Properties	Specifies the set of content properties in the content class. For details, see "contentProperties" on page 140.	This property is optional on a PUT request. The set of content properties specified in the request body replaces the set of content properties currently in the content class. To remove all content properties, specify an empty set.
name	String	Specifies the name of the content class. Content class names must be from one through 64 characters long, can contain any valid UTF-8 characters, including white space, and are not case sensitive.	This property is required on a PUT request.

(Continued)

Property name	Data type	Description	Notes
namespaces	List	Associates zero, one, or more namespaces with the content class. Valid values are the names of existing search-enabled namespaces.	<p>This property is optional on a PUT request.</p> <p>The set of namespaces specified in the request body replaces the set of namespaces currently associated with the content class. To dissociate all namespaces from the content class, specify an empty set.</p> <p>In XML, each listed namespace is the value of an element named name. In JSON, the name in the name/value pair that lists the namespaces is name.</p>

contentClass data type example

Here's an XML example of the contentClass data type:

```

<contentClass>
  <name>DICOM</name>
  <contentProperties>
    <contentProperty>
      <name>Doctor_Name</name>
      <expression>/dicom_image/doctor/name</expression>
      <type>STRING</type>
      <multivalued>>false</multivalued>
      <format></format>
    </contentProperty>
    <contentProperty>
      <name>Doctor_Specialty</name>
      <expression>/dicom_image/doctor/specialties/specialty</expression>
      <type>STRING</type>
      <multivalued>>true</multivalued>
      <format></format>
    </contentProperty>
    <contentProperty>
      <name>Followup_Needed</name>
      <expression>/dicom_image/followup_needed</expression>
      <type>BOOLEAN</type>
      <multivalued>>false</multivalued>
      <format></format>
    </contentProperty>
  </contentProperties>
</contentClass>

```

```

</contentProperty>
<contentProperty>
  <name>Image_Date</name>
  <expression>/dicom_image/image/date</expression>
  <type>DATE</type>
  <multivalued>>false</multivalued>
  <format>MM/dd/yyyy</format>
</contentProperty>
<contentProperty>
  <name>Image_Type</name>
  <expression>/dicom_image/image/@type</expression>
  <type>STRING</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
<contentProperty>
  <name>Patient_ID</name>
  <expression>/dicom_image/patient/id</expression>
  <type>INTEGER</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
<contentProperty>
  <name>Patient_Name</name>
  <expression>/dicom_image/patient/name</expression>
  <type>STRING</type>
  <multivalued>>false</multivalued>
  <format></format>
</contentProperty>
</contentProperties>
<namespaces>
  <name>Medical-Records</name>
</namespaces>
</contentClass>

```

contentProperty

The contentProperties data type describes the contentProperties property of the contentClass data type.

contentProperty data type properties

The table below describes the properties included in the contentProperty data type.

Property name	Data type	Description	Notes
expression	String	<p>Specifies the expression for the content property. Valid values are valid XPath expressions, optionally prefixed with an annotation name, in this format:</p> <p><i>@annot-name:xpath-expression</i></p> <p>For more information on content property expressions, see <i>Managing a Tenant and Its Namespaces</i> or <i>Managing the Default Tenant and Namespace</i>.</p>	This property is required.

(Continued)

Property name	Data type	Description	Notes
format	String	<p>Specifies the format for the content property. Valid values are specific to each data type for which the format property is valid:</p> <ul style="list-style-type: none"> For DATE, the value must be a valid datetime format. If you don't specify a format, the metadata query engine indexes only values that match patterns such as MM/dd/yyyy, MM-dd-yyyy, yyyy-MM-dd, or yyyy-MM-dd'T'HH:mm:ssZ. For FLOAT, the value must be a valid number format that can map content property values to decimal numbers. If you don't specify a format, the metadata query engine indexes only sequences of digits that optionally include one decimal point. For INTEGER, the value must be a valid number format that can map content property values to integers. If you don't specify a format, the metadata query engine indexes only sequences of digits with no special characters. <p>For more information on content property formats, see <i>Managing a Tenant and Its Namespaces</i> or <i>Managing the Default Tenant and Namespace</i>.</p>	This property is optional and can have a value only when the value of the type property is DATE , FLOAT , or INTEGER .

(Continued)

Property name	Data type	Description	Notes
multivalued	Boolean	<p>Specifies whether the content property is single-valued or multivalued. Valid values are:</p> <ul style="list-style-type: none">• true — The content property can have multiple values for any given object.• false — The content property can have only one value for any given object. <p>The default is false.</p>	
name	String	<p>Specifies the name of the content property. Content property names must be from one through 25 characters long, can contain only alphanumeric characters and underscores (_), and are case sensitive. White space is not allowed.</p>	This property is required.

(Continued)

Property name	Data type	Description	Notes
type	String	<p>Specifies the data type of the content property. Valid values are:</p> <ul style="list-style-type: none"> • BOOLEAN — The metadata query engine indexes the value as true or false. • DATE — The metadata query engine indexes the value as a date and time. • FLOAT — The metadata query engine indexes the value as a decimal number with or without an exponent, depending on the value. • FULLTEXT — The metadata query engine indexes the value as a text string after breaking it into tokens. • INTEGER — The metadata query engine indexes the value as an integer. • STRING — The metadata query engine indexes the content property values as text strings. <p>These values are not case sensitive.</p> <p>For more information on content property data types, see <i>Managing a Tenant and Its Namespaces</i> or <i>Managing the Default Tenant and Namespace</i>.</p>	This property is required.

contentProperty data type example

Here’s an XML example of the contentProperty data type:

```
<contentProperty>
  <name>Image_Date</name>
  <expression>/dicom_image/image/date</expression>
  <type>DATE</type>
  <multivalued>>false</multivalued>
  <format>MM/dd/yyyy</format>
</contentProperty>
```

contentProperties

The contentProperties data type describes the contentProperties property of the contentClass data type.

contentProperties data type property

The table below describes the property included in the contentProperties data type.

Property name	Data type	Description	Notes
contentProperty	content Property	Specifies a content property. For details, see “contentProperty” on page 136.	

contentProperties data type example

Here’s an XML example of the contentProperties data type:

```
<contentProperties>
  <contentProperty>
    <name>Doctor_Name</name>
    <expression>/dicom_image/doctor/name</expression>
    <type>STRING</type>
    <multivalued>>false</multivalued>
    <format></format>
  </contentProperty>
  <contentProperty>
    <name>Doctor_Specialty</name>
    <expression>/dicom_image/doctor/specialties/specialty</expression>
    <type>STRING</type>
```

```

        <multivalued>true</multivalued>
        <format></format>
    </contentProperty>
    <contentProperty>
        <name>Followup_Needed</name>
        <expression>/dicom_image/followup_needed</expression>
        <type>BOOLEAN</type>
        <multivalued>false</multivalued>
        <format></format>
    </contentProperty>
    <contentProperty>
        <name>Image_Date</name>
        <expression>/dicom_image/image/date</expression>
        <type>DATE</type>
        <multivalued>false</multivalued>
        <format>MM/dd/yyyy</format>
    </contentProperty>
    <contentProperty>
        <name>Image_Type</name>
        <expression>/dicom_image/image/@type</expression>
        <type>STRING</type>
        <multivalued>false</multivalued>
        <format></format>
    </contentProperty>
    <contentProperty>
        <name>Patient_ID</name>
        <expression>/dicom_image/patient/id</expression>
        <type>INTEGER</type>
        <multivalued>false</multivalued>
        <format></format>
    </contentProperty>
    <contentProperty>
        <name>Patient_Name</name>
        <expression>/dicom_image/patient/name</expression>
        <type>STRING</type>
        <multivalued>false</multivalued>
        <format></format>
    </contentProperty>
</contentProperties>

```

customMetadataIndexingSettings

The customMetadataIndexingSettings data type describes the customMetadataIndexingSettings resource for namespaces.

customMetadataIndexingSettings data type properties

The table below describes the properties included in the customMetadataIndexingSettings data type.

Property name	Data type	Description	Notes
contentClasses	List	<p>Associates zero, one, or more content classes with the namespace. Valid values are the names of existing content classes.</p> <p>Content class names are case sensitive.</p>	<p>The set of content classes specified in the request body replaces the set of content classes currently associated with the namespace. To dissociate all content classes from the namespace, specify an empty set.</p> <p>In XML, each listed content class is the value of an element named name. In JSON, the name in the name/value pair that lists the content classes is name.</p>
fullIndexingEnabled	Boolean	<p>Specifies whether the metadata query engine indexes the full text of custom metadata. Valid values are:</p> <ul style="list-style-type: none"> true — The metadata query engine indexes the full text of custom metadata. false — The metadata query engine does not index the full text of custom metadata. <p>The default is false.</p>	<p>You can set this property to true only while custom metadata indexing is enabled for the namespace.</p>

(Continued)

Property name	Data type	Description	Notes
excludedAnnotations	String	<p>Specifies a comma-separated list of the names of annotations to be excluded from indexing by the metadata query engine.</p> <p>Instead of explicit names, you can use patterns. The wildcard character for pattern matching is the asterisk (*), which matches any number of characters of any type, including none. The asterisk can occur anywhere in the pattern.</p> <p>Annotation names are case sensitive.</p>	<p>The list of annotation names you specify for this property replaces the current list of annotation names. To remove all annotation names from the list, specify this property with no value.</p> <p>You can set a value for this property only while custom metadata indexing is enabled for the namespace.</p> <p>Disabling custom metadata indexing for the namespace automatically deletes the list of excluded annotation for the namespace.</p>

customMetadataIndexingSettings data type example

Here's an XML example of the customMetadataIndexingSettings data type:

```
<customMetadataIndexingSettings>
  <contentClasses>
    <name>DICOM</name>
    <name>Appointment</name>
  </contentClasses>
  <excludedAnnotations>misc*, email</excludedAnnotations>
  <fullIndexingEnabled>>false</fullIndexingEnabled>
</customMetadataIndexingSettings>
```

dataAccessPermissions

The dataAccessPermissions data type describes the dataAccessPermissions resource for group accounts and user accounts.

dataAccessPermissions data type property

The table below describes the property included in the dataAccessPermissions data type.

Property name	Data type	Description	Notes
namespacePermission	namespacePermission	Specifies the permissions the user or group account has for a namespace. For details, see "namespacePermission" on page 203.	Include one instance of this property for each namespace for which you want to change the permissions. If you omit a namespace, its permissions are not changed.

dataAccessPermissions data type example

Here's an XML example of the dataAccessPermissions data type:

```
<dataAccessPermissions>
  <namespacePermission>
    <namespaceName>Accounts-Payable</namespaceName>
    <permissions>
      <permission>BROWSE</permission>
      <permission>READ</permission>
    </permissions>
  </namespacePermission>
  <namespacePermission>
    <namespaceName>Accounts-Receivable</namespaceName>
    <permissions>
      <permission>BROWSE</permission>
      <permission>SEARCH</permission>
      <permission>PURGE</permission>
      <permission>DELETE</permission>
      <permission>READ</permission>
      <permission>WRITE</permission>
    </permissions>
  </namespacePermission>
</dataAccessPermissions>
```


emailNotification

The emailNotification data type describes the emailNotification resource for tenants.

emailNotification data type properties

The table below describes the properties included in the emailNotification data type.

Property name	Data type	Description	Notes
emailTemplate	email Template	Specifies the template for email messages HCP sends to the specified recipients. For details, see "emailTemplate" on page 147.	The email template specified in the request body replaces the current email template. To restore the default email template, specify the emailTemplate property with no value. For information on the default email template, see "Email template defaults" on page 149.
enabled	Boolean	Specifies whether email notification is enabled for the tenant. Valid values are: <ul style="list-style-type: none">true — Email notification is enabled.false — Email notification is disabled. The default is false .	Email notification at the tenant level is supported only if HCP is configured to support it at the system level.
recipients	recipients	Specifies the set of recipients for email notification. For details, see "recipients" on page 211.	The set of recipients specified in the request body replaces the set of recipients currently configured for email notification. To remove all email recipients, specify an empty set. Recipients are added to the Bcc list for each email.

emailNotification data type example

Here's an XML example of the emailNotification data type:

```
<emailNotification>
  <enabled>true</enabled>
  <emailTemplate>
    <from>log@$location</from>
    <subject>$severity - $shortText</subject>
    <body>A message was written to the tenant log on $date.\n\n$reason\n\n
      $action</body>
  </emailTemplate>
  <recipients>
    <recipient>
      <address>lgreen@example.com, sgold@example.com</address>
      <importance>MAJOR</importance>
      <severity>ERROR</severity>
      <type>GENERAL,SECURITY</type>
    </recipient>
    <recipient>
      <address>mwhite@example.com</address>
      <importance>ALL</importance>
      <severity>WARNING</severity>
      <type>COMPLIANCE</type>
    </recipient>
  </recipients>
</emailNotification>
```

emailTemplate

The emailTemplate data type describes the emailTemplate property of the emailNotification data type.

emailTemplate data type properties

The table below describes the properties included in the emailTemplate data type.

Property name	Data type	Description	Notes
body	String	<p>Specifies the format of the body of the email messages HCP sends when email notification is enabled. Valid values include any combination of plain text and email template variables.</p> <p>Plain text can include spaces and line breaks but not tabs. The character sequence consisting of a backslash (\) followed by a lowercase n creates a line break.</p> <p>For a description of email template variables, see “Email template variables” below.</p>	To change the body in the email template to blank, include the from and subject properties in the request body and omit the body property.
from	String	<p>Specifies the content of the email From line. Valid values include any combination of plain text and email template variables and must have the form of a valid email address.</p> <p>Some email servers require that the value in the From line be an email address that is already known to the server.</p> <p>For a description of email template variables, see “Email template variables” below.</p>	If this property is included in the request body, the subject property must also be included.

(Continued)

Property name	Data type	Description	Notes
subject	String	<p>Specifies the content of the email Subject line. Valid values include any combination of plain text and email template variables.</p> <p>Plain text can include spaces but not line breaks or tabs.</p> <p>For a description of email template variables, see "Email template variables" below.</p>	If this property is included in the request body, the from property must also be included.

Email template variables

The values you specify for the body, from, and subject properties of the emailTemplate data type can include variables that correspond to the information available for each log message (for example, the severity of the event that triggered the message or the short description of the event). When sending email, HCP replaces the variables in the email message with the applicable information.

To include a variable in the email template, you specify the variable name preceded by the dollar sign (\$). A dollar sign followed by anything other than a variable name is displayed as a dollar sign in the email HCP sends.

The table below lists the variables you can use in the email template.

Variable	Description
\$action	The action to take in response to the message
\$date	The date and time at which the event occurred (for example, Wed Feb 8 2012 3:15:57 PM EST)
\$fullText	The full text of the message
\$id	The message ID
\$location	The fully qualified name of the HCP system on which the event occurred (for example, hcp-ma.example.com)
\$origin	For user-initiated events, the IP address from which the event request was sent and the port through which HCP received the event request, separated by a colon (for example, 192.168.152.181:8000)
\$reason	The reason why HCP issued the message

(Continued)

Variable	Description
\$scope	
\$severity	The severity of the event that triggered the message
\$shortText	A brief description of the event that triggered the message
\$type	The type of message (General, Security, or Compliance), preceded by Important and a comma if the message is important (for example, Important, Security)
\$user	The user ID and username of the event initiator (for example, 105ff38f-4770-4f98-b5b3-8371ab0af359 lgreen)

Email template defaults

The table below shows the format of the default email template.

Field	Default value
From	log@\$location
Subject	[\$severity] \$shortText
Body	<p>The following event occurred on \$date: \$fullText</p> <p>Reason: \$reason</p> <p>Action: \$action</p> <p>Details: User: \$user Origin: \$origin</p>

emailTemplate data type example

Here's an XML example of the emailTemplate data type:

```

<emailTemplate>
  <from>log@$location</from>
  <body>A message was written to the tenant log on $date.\n\n$reason\n\n
    $action</body>
  <subject>$severity - $shortText</subject>
</emailTemplate>

```

failoverSettings

The failoverSettings data type describes the failoverSettings property of the link data type.

failoverSettings data type properties

The table below describes the properties included in the failoverSettings data type.

Property name	Data type	Description	Notes
local	local	Specifies the automatic failover and failback settings for the local system for the replication link. For details, see “local (data type for replication link failoverSettings local property)” on page 173.	This property is optional on a PUT request to create a replication link.
remote	remote	Specifies the automatic failover and failback settings for the remote system for the replication link. For details, see “remote (data type for replication link failoverSettings remote property)” on page 213.	This property is optional on a PUT request to create a replication link.

failoverSettings data type example

Here’s an XML example of the failoverSettings data type; the properties shown are those that are returned by a **GET** request sent to the primary system for an active/passive link:

```
<failoverSettings>
  <remote>
    <autoFailover>true</autoFailover>
    <autoFailoverMinutes>120</autoFailoverMinutes>
    <autoCompleteRecovery>false</autoCompleteRecovery>
    <autoCompleteRecoveryMinutes>60</autoCompleteRecoveryMinutes>
  </remote>
</failoverSettings>
```

groupAccount

The groupAccount data type describes the groupAccounts resource.

groupAccount data type properties

The table below describes the properties included in the groupAccount data type.

Property name	Data type	Description	Notes
allowNamespaceManagement	Boolean	<p>Specifies whether the group account has the allow namespace management property. Valid values are:</p> <ul style="list-style-type: none"> true — The group account has the allow namespace property. false — The group account does not have the allow namespace management property. <p>On a PUT request, the default is true if the roles property includes ADMINISTRATOR in the same request; otherwise, the default is false.</p> <p>On a POST request, adding ADMINISTRATOR to the roles for the group account automatically enables the allow namespace management property for the account.</p> <p>Users in groups with the allow namespace management property can use the HCP management and HS3 APIs to:</p> <ul style="list-style-type: none"> Create namespaces List, view and change the versioning status of, and delete namespaces they own 	<p>This property is not valid on a PUT request. It is valid on a POST request only if the user making the request has the administrator role.</p>

(Continued)

Property name	Data type	Description	Notes
externalGroupID	String	Specifies the security identifier (SID) of the AD group that corresponds to the HCP group account. For a PUT request, valid values are the SIDs of AD groups defined in the AD forest supported by HCP.	<p>Either this property or the groupname property is required on a PUT request. If you include both properties in the request body, they must identify the same AD group.</p> <p>This property is not valid on a POST request. It is returned only by a verbose GET request and only when the user making the request has the security role.</p>
groupname	String	<p>Specifies the name of the HCP group account. For a PUT request, valid values are the names of AD groups defined in the AD forest supported by HCP, in either of these formats:</p> <p><i>group-name</i></p> <p><i>group-name@ad-domain-name</i></p> <p>If you omit the domain name, HCP uses the AD domain specified in the system configuration.</p> <p>Be sure to use the second format if a group with the specified name exists in more than one domain in the AD forest or if the group name looks like a SID.</p>	<p>Either this property or the externalGroupID property is required on a PUT request. If you include both properties in the request body, they must identify the same AD group.</p> <p>This property is not valid on a POST request.</p>

(Continued)

Property name	Data type	Description	Notes
roles	List	<p>Associates zero, one, or more roles with the group account. Valid values for roles are:</p> <ul style="list-style-type: none"> ADMINISTRATOR COMPLIANCE MONITOR SECURITY <p>These values are not case sensitive.</p> <p>The default is no roles.</p>	<p>This property is valid on a POST request and returned by a GET request only when the user making the request has the security role.</p> <p>For an existing group account, the set of roles specified in the request body replaces the set of roles currently associated with the group account. To remove all roles, specify an empty set.</p> <p>In XML, each listed role is the value of an element named role. In JSON, the name in the name/value pair that lists the roles is role.</p>

groupAccount data type example

Here's an XML example of the groupAccount data type:

```
<groupAccount>
  <allowNamespaceManagement>false</allowNamespaceManagement>
  <externalGroupID>S-1-5-21-1522923621-2272695913-102089983-3621
</externalGroupID>
  <groupname>hcp-admin@ad.example.com</groupname>
  <roles>
    <role>MONITOR</role>
    <role>ADMINISTRATOR</role>
  </roles>
</groupAccount>
```

Query parameter for resetting the security group

A tenant must at all times have at least one user that can manage user and group accounts. This means that the tenant must have at least one user or group account with the security role. If the tenant does not have such a group account and the passwords for all user accounts with the security role have been lost, the tenant is in a state in which no users can manage user and group accounts.

To recover from this state, you can use the **resetSecurityGroup** query parameter to do either of these:

- Assign the security role to an existing HCP group account. In this case, the value of the **resetSecurityGroup** parameter must be the name or external group ID of an existing HCP group account.
- Create a new HCP group account with only the security role. In this case, the value of the **resetSecurityGroup** parameter must be the name or SID of an AD group defined in the AD forest supported by HCP. You can specify the name in either of these formats:

group-name

group-name@ad-domain-name

If you omit the domain name, HCP uses the AD domain specified in the system configuration.

Be sure to use the second format if a group with the specified name exists in more than one domain in the AD forest or if the group name looks like a SID.

To reset the security group for a tenant, you need a system-level user account with the administrator role.

You use the **resetSecurityGroup** query parameter with a **POST** request against the groupAccounts resource, as in this example:

```
curl -k -i -d "<groupAccounts/>" -H "Content-Type: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/groupAccounts
?resetSecurityGroup=hcp-admin@ad.example.com"
```

As an alternative to resetting the security group, you can reset the passwords of all users with the security role. For information on doing this, see ["Query parameter for resetting security user passwords"](#) on page 273.

httpProtocol

The httpProtocol data type describes the http resource for HCP namespaces. This data type includes properties for the HTTP, HS3, and WebDAV protocols.

httpProtocol data type properties

The table below describes the properties included in the httpProtocol data type.

Property name	Data type	Description	Notes
hs3Enabled	Boolean	<p>Specifies whether the HS3 API is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — The HS3 API is enabled. false — The HS3 API is disabled. <p>The default is false.</p>	This property can be set to true only if ACLs are enabled for the namespace.
hs3Requires Authentication	Boolean	<p>Specifies whether user authentication is required or optional for access to the namespace through the HS3 API. Valid values are:</p> <ul style="list-style-type: none"> true — User authentication is required. false — User authentication is optional. <p>The default is true.</p>	
hswiftEnabled	Boolean	<p>Specifies whether the HSwift API is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — The HSwift API is enabled. false — The HSwift API is disabled. <p>The default is false.</p>	

(Continued)

Property name	Data type	Description	Notes
hswiftRequiresAuthentication	Boolean	<p>Specifies whether user authentication is required or optional for access to the namespace through the HSwift API. Valid values are:</p> <ul style="list-style-type: none"> true — User authentication is required. false — User authentication is optional. <p>The default is true.</p>	
httpActiveDirectorySSOEnabled	Boolean	<p>Specifies whether HCP supports AD single sign-on for HTTP and HS3 access to the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — HCP supports AD single sign-on for HTTP and HS3 access to the namespace. false — HCP does not support AD single sign-on for HTTP and HS3 access to the namespace. <p>The default is true.</p>	<p>This property is valid on a POST request only if owning the tenant supports AD authentication.</p> <p>You can set the value of this property to true only while HTTP or HTTPS is enabled. Disabling both HTTP and HTTPS automatically disables this property.</p>
httpEnabled	Boolean	<p>Specifies whether the HTTP port is open for HTTP and WebDAV access to the namespace without SSL security. Valid values are:</p> <ul style="list-style-type: none"> true — The HTTP port is open. false — The HTTP port is closed. <p>The default is false.</p>	

(Continued)

Property name	Data type	Description	Notes
httpsEnabled	Boolean	<p>Specifies whether the HTTPS port is open for HTTP and WebDAV access to the namespace with SSL security. Valid values are:</p> <ul style="list-style-type: none"> true — The HTTPS port is open. false — The HTTPS port is closed. <p>The default is true.</p>	
ipSettings	ipSettings	<p>Specifies which IP addresses can and cannot access the namespace through the HTTP and WebDAV protocols.</p> <p>For details, see “ipSettings” on page 160.</p>	
restEnabled	Boolean	<p>Specifies whether the RESTful implementation of the HTTP protocol is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — The RESTful implementation of the HTTP protocol is enabled. false — The RESTful implementation of the HTTP protocol is disabled. <p>The default is true.</p>	

(Continued)

Property name	Data type	Description	Notes
restRequiresAuthentication	Boolean	<p>Specifies whether user authentication is required or optional for access to the namespace through the HTTP protocol. Valid values are:</p> <ul style="list-style-type: none"> true — User authentication is required. false — User authentication is optional. <p>The default is true.</p>	
webdavBasicAuthEnabled	Boolean	<p>Specifies whether the WebDAV protocol requires basic authentication for access to the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — WebDAV requires basic authentication. false — WebDAV does not require basic authentication. <p>The default is false.</p>	<p>You can set the value of this property to true only if a WebDAV username and password already exist or are specified by the webdav-BasicAuthUsername and webdavBasicAuthPassword properties in the same request.</p>
webdavBasicAuthPassword	String	<p>Specifies the password to use for basic authentication with the WebDAV protocol. For the rules for passwords, see "Passwords" on page 104.</p>	<p>You can specify a value for this property only if a basic authentication username already exists or is specified by the webdavBasicAuth-Username property in the same request.</p> <p>To remove the basic authentication password, specify the webdavBasic-AuthPassword property with no value. You can remove the password only if you remove the basic authentication username in the same request.</p>

(Continued)

Property name	Data type	Description	Notes
webdavBasicAuth Username	String	<p>Specifies the username to use for basic authentication with the WebDAV protocol.</p> <p>Username must be from one through 64 characters long and can contain any valid UTF-8 characters but cannot start with an opening square bracket ([). White space is allowed.</p> <p>Username are not case sensitive.</p>	<p>You can specify a value for this property only if a basic authentication password already exists or is specified by the webdavBasicAuth-Password property in the same request.</p> <p>To remove the basic authentication username, specify the webdavBasicAuthUsername property with no value. You can remove the username only if you remove the basic authentication password in the same request.</p>
webdavCustomMetadata	Boolean	<p>Specifies whether WebDAV dead properties can be stored as custom metadata. Valid values are:</p> <ul style="list-style-type: none"> true — Dead properties can be stored as custom metadata. false — Dead properties cannot be stored as custom metadata. <p>The default is false.</p>	
webdavEnabled	Boolean	<p>Specifies whether the WebDAV protocol is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — The WebDAV protocol is enabled. false — The WebDAV protocol is disabled. <p>The default is false.</p>	

httpProtocol data type example

Here's an XML example of the httpProtocol data type:

```
<httpProtocol>
  <hswiftEnabled>false</hswiftEnabled>
  <hswiftRequiresAuthentication>false</hswiftRequiresAuthentication>
  <hs3Enabled>false</hs3Enabled>
  <hs3RequiresAuthentication>false</hs3RequiresAuthentication>
  <httpActiveDirectorySSOEnabled>true</httpActiveDirectorySSOEnabled>
  <httpEnabled>false</httpEnabled>
  <httpsEnabled>true</httpsEnabled>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.140.10</ipAddress>
      <ipAddress>192.168.140.14</ipAddress>
      <ipAddress>192.168.140.15</ipAddress>
      <ipAddress>192.168.149.0/24</ipAddress>
    </allowAddresses>
    <allowIfInBothLists>false</allowIfInBothLists>
    <denyAddresses>
      <ipAddress>192.168.149.5</ipAddress>
    </denyAddresses>
  </ipSettings>
  <restEnabled>true</restEnabled>
  <restRequiresAuthentication>true</restRequiresAuthentication>
  <webdavBasicAuthEnabled>false</webdavBasicAuthEnabled>
  <webdavBasicAuthPassword></webdavBasicAuthPassword>
  <webdavBasicAuthUsername></webdavBasicAuthUsername>
  <webdavCustomMetadata>false</webdavCustomMetadata>
  <webdavEnabled>false</webdavEnabled>
</httpProtocol>
```

ipSettings

The ipSettings data type describes the ipSettings property of these data types:

- cifsProtocol
- consoleSecurity
- httpProtocol
- nfsProtocol

- protocols
- searchSecurity
- smtpProtocol

ipSettings data type properties

The table below describes the properties included in the ipSettings data type.

Property name	Data type	Description	Notes
allowAddresses	allow Addresses	<p>Specifies the IP addresses to be allowed access to HCP through the given interface. By default, the set of IP addresses includes only 0.0.0.0/0.</p> <p>For information on specifying the IP addresses for this property, see “allowAddresses and denyAddresses list entries” below.</p>	<p>The set of IP addresses specified in the request body replaces the set of IP addresses currently allowed access to HCP through the given interface. To remove all IP addresses, specify an empty set.</p> <p>In XML, each IP address specification is the value of an element named ipAddress. In JSON, the name in the name/value pair that specifies the IP addresses is ipAddress.</p>
allowIfInBothLists	Boolean	<p>Specifies how HCP should handle IP addresses that are either both allowed and denied or neither allowed nor denied access to HCP through the given interface. Valid values are true and false. The default is true.</p> <p>For the effects of specifying true or false, see “allowIfInBothLists property effects” on page 163.</p>	<p>This property is not valid for the cifsProtocol, nfsProtocol, and smtpProtocol data types.</p> <p>For details on the effects of specifying true or false for this property, see “allowIfInBothLists property effects” on page 163.</p>

(Continued)

Property name	Data type	Description	Notes
denyAddresses	deny Addresses	<p>Specifies the IP addresses to be denied access to HCP through the given interface. By default, the set of IP addresses is empty.</p> <p>For information on specifying the IP addresses for this property, see “allowAddresses and denyAddresses list entries” below.</p>	<p>This property is not valid for the nfsProtocol data type.</p> <p>The set of IP addresses specified in the request body replaces the set of IP addresses currently denied access to HCP through the given interface. To remove all IP addresses, specify an empty set.</p> <p>In XML, each IP address specification is the value of an element named ipAddress. In JSON, the name in the name/value pair that specifies the IP addresses is ipAddress.</p>

allowAddresses and denyAddresses list entries

Each ipAddress entry within an allowAddresses or denyAddresses property can have a value of:

- An individual IP address.
- A comma-separated list of IP addresses.
- A range of IP addresses specified as *ip-address/subnet-mask* (for example, 192.168.100.197/255.255.255.0).
- A range of IP addresses specified in CIDR format (for example, 192.168.100.0/24). The CIDR entry that matches all IP addresses is 0.0.0.0/0.

allowIfInBothLists property effects

The table below describes the effects of specifying **true** or **false** for the allowIfInBothLists property.

Listed IP addresses	true	false
allowAddresses: none denyAddresses: none	All IP addresses can access HCP through the given interface.	No IP addresses can access HCP through the given interface.
allowAddresses: at least one denyAddresses: none	All IP addresses can access HCP through the given interface.	Only IP addresses in the allowAddresses list can access HCP through the given interface.
allowAddresses: none denyAddresses: at least one	All IP addresses not in the denyAddresses list can access HCP through the given interface. IP addresses in the denyAddresses list cannot.	No IP addresses can access HCP through the given interface.
allowAddresses: at least one denyAddresses: at least one	IP addresses appearing in both or neither of the lists can access HCP through the given interface.	IP addresses appearing in both or neither of the lists cannot access HCP through the given interface.

ipSettings data type example

Here's an XML example of the ipSettings data type:

```
<ipSettings>
  <allowAddresses>
    <ipAddress>192.168.140.10</ipAddress>
    <ipAddress>192.168.140.14</ipAddress>
    <ipAddress>192.168.140.15</ipAddress>
    <ipAddress>192.168.149.0/24</ipAddress>
  </allowAddresses>
  <allowIfInBothLists>false</allowIfInBothLists>
  <denyAddresses>
    <ipAddress>192.168.149.5</ipAddress>
  </denyAddresses>
</ipSettings>
```

license

The license data type retrieves information about current and past storage licenses.

license data type properties

The table below describes the properties included in the license data type.

Property name	Data type	Description	Notes
activeCapacity	Long	Specifies the active storage capacity in terabytes.	Value is returned in bytes.
expirationDate	String	Specifies the storage license expiration date. If there is no expiration date, enter None .	
extendedCapacity	Long	Specifies the extended storage capacity in terabytes.	Value is returned in bytes.
quoteNumber	String	Specifies the storage license quote number.	
serialNumber	String	Specifies the serial number of the HCP system the storage license is intended for.	
uploadDate	String	Specifies the date which the license was uploaded.	This property is returned only by a verbose GET request.

license data type example

Here's an XML example of the license data type:

```
<license>
  <activeCapacity>10000000000000</activeCapacity>
  <expirationDate>Jan 1, 2018</expirationDate>
  <extendedCapacity>0</extendedCapacity>
  <quoteNumber>0123456789</quoteNumber>
  <serialNumber>12345</serialNumber>
  <uploadDate>Aug 14, 2013</uploadDate>
</license>
```

licenses

The licenses data type describes the storage licenses resource.

licenses data type properties

The table below describes the property included in the licenses data type.

Property name	Data type	Description	Notes
license	license	Specifies information about individual storage license. For more information, see Chapter 4, "license." on page 163.	The response to a GET for the licenses resource returns either a single instance of the license property or multiple instances depending on whether the request includes the verbose query parameter. For more information, see Chapter 4, "license." on page 163.

licenses data type example

Here's an XML example of the licenses data type:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<licenses>
  <license>
    <activeCapacity>10000000000000</activeCapacity>
    <expirationDate>Jan 1, 2018</expirationDate>
    <extendedCapacity>0</extendedCapacity>
    <quoteNumber>0123456789</quoteNumber>
    <serialNumber>12345</serialNumber>
    <uploadDate>Aug 14, 2013</uploadDate>
  </license>
</licenses>
```

Query parameter for retrieving a list of licenses

To retrieve a list of all the storage licenses that have ever been uploaded to the HCP system, you use this query parameter:

```
verbose=true
```

Here's a sample **GET** request that retrieves both the current and past storage licenses for an HCP system:

```
curl -k -b -H "Accept:application/xml"
-H "Authorization: HCP YWxscm9sZXNM=:04EC9F614D89FF5C7126D32ACB448382"
https://admin.archivas.com:9090/mapi/storage/licenses?verbose=true
```

link

The link data type describes the links resource.

link data type properties

The table below describes the properties included in the link data type.

Property name	Data type	Description	Notes
compression	Boolean	<p>Specifies whether HCP should compress data before transmitting it to the other system involved in the replication link. For an active/active link, this setting applies to data being replicated in both directions on the link.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> true — HCP should compress data. false — HCP should not compress data. <p>The default is false.</p>	This property is optional on a PUT request.
connection	connection	<p>Identifies the remote system for the replication link and specifies how the two systems should communicate with each other.</p> <p>For details, see “connection” on page 123.</p>	This property is required on a PUT request.
description	String	<p>Specifies a description of the replication link. This description is optional. The default is no description.</p> <p>To remove a description from an existing link, specify the description property with no value.</p> <p>For the rules for descriptions, see “Common property values” on page 104.</p>	This property is optional on a PUT request.

(Continued)

Property name	Data type	Description	Notes
encryption	Boolean	<p>Specifies whether HCP should encrypt data before transmitting it to the other system involved in the replication link. For an active/active link, this setting applies to data being replicated in both directions on the link.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> true — HCP should encrypt data. false — HCP should not encrypt data. <p>The default is false.</p>	This property is optional on a PUT request.
failoverSettings	failover Settings	<p>Specifies the automatic failover and failback settings for the replication link.</p> <p>For details, see “failoverSettings” on page 150.</p>	This property is optional on a PUT request.
id	String	Specifies the system-supplied unique ID for the replication link. HCP generates this ID automatically when you create a link.	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.
name	String	<p>Specifies the name of the replication link. Link names must be from one through 64 characters long and can contain any valid UTF-8 characters, including white space. Link names are not case sensitive.</p> <p>Link names must be unique within a replication topology.</p>	This property is required on a PUT request.

(Continued)

Property name	Data type	Description	Notes
priority	String	<p>Specifies whether priority should be given to objects with the oldest changes, regardless of namespace, or processing should be balanced across namespaces. For an active/active link, this setting applies to both systems involved in the link.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • FAIR — Give priority to objects with the oldest changes. • OLDEST_FIRST — Balance processing across namespaces. <p>The default is OLDEST_FIRST.</p> <p>These values are not case sensitive.</p>	This property is optional on a PUT request.
statistics	statistics	<p>Specifies information about activity on the replication link.</p> <p>For details, see “statistics (data type for replication link statistics property)” on page 231.</p>	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.

(Continued)

Property name	Data type	Description	Notes
status	String	<p>Specifies the general state of the replication link. Possible values are:</p> <ul style="list-style-type: none">• GOOD — The link is healthy.• WARNING — The link is healthy, but normal replication is not occurring on the link.• BAD — The link is unhealthy.	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.</p>

link

(Continued)

Property name	Data type	Description	Notes
statusMessage	String	<p>Specifies text describing the current state of the replication link. Possible values are:</p> <ul style="list-style-type: none">• Synchronizing data• Sending data• Receiving data• Recovering data• Completing recovery• Scheduled off period• Suspended by user• Pending remote reply• Pending• Failed over• Remote storage full, link suspended• Local storage full, link suspended• High error rate• Stalled link• Broken link <p>For details on these status messages, see <i>Replicating Tenants and Namespaces</i>.</p>	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.</p>

(Continued)

Property name	Data type	Description	Notes
suspended	Boolean	Specifies whether the replication link is currently suspended. Possible values are: <ul style="list-style-type: none"> true — The link is suspended. false — The link is not suspended. 	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.
type	String	Specifies the replication link type. Valid values are: <ul style="list-style-type: none"> ACTIVE_ACTIVE — The link is an active/active link. OUTBOUND — The link is an outbound active/passive link. INBOUND — The link is an inbound active/passive link. These values are not case sensitive.	This property is required on a PUT request. You can change the value of this property from OUTBOUND or INBOUND to ACTIVE_ACTIVE . You cannot change the value from: <ul style="list-style-type: none"> ACTIVE_ACTIVE to OUTBOUND or INBOUND OUTBOUND to INBOUND INBOUND to OUTBOUND

link data type example

Here's an XML example of the link data type; the properties shown are those that are returned by a verbose **GET** request for an active/active link:

```
<link>
  <compression>false</compression>
  <connection>
    <localHost>
      192.168.210.16, 192.168.210.17, 192.168.210.18, 192.168.210.19
    </localHost>
    <localPort>5748</localPort>
    <remoteHost>replication.admin.hcp-ca.example.com</remoteHost>
    <remotePort>5748</remotePort>
  </connection>
  <description>Active/active link between systems in MA and CA</description>
  <encryption>false</encryption>
  <failoverSettings>
```

```

<local>
  <autoFailover>true</autoFailover>
  <autoFailoverMinutes>60</autoFailoverMinutes>
</local>
<remote>
  <autoFailover>true</autoFailover>
  <autoFailoverMinutes>60</autoFailoverMinutes>
</remote>
</failoverSettings>
<id>2de89eec-0ec0-4f98-b852-0778dfeee792</id>
<name>MA-CA</name>
<priority>OLDEST_FIRST</priority>
<statistics>
  <upToDateAsOfString>2014-03-18T10:47:59-0400
  </upToDateAsOfString>
  <upToDateAsOfMillis>1395154079640</upToDateAsOfMillis>
  <bytesPending>189027593061</bytesPending>
  <bytesPendingRemote>319740</bytesPendingRemote>
  <bytesReplicated>72254119306967</bytesReplicated>
  <bytesPerSecond>56215390</bytesPerSecond>
  <objectsPending>534</objectsPending>
  <objectsPendingRemote>2</objectsPendingRemote>
  <objectsReplicated>295661</objectsReplicated>
  <operationsPerSecond>119.1</operationsPerSecond>
  <errors>0</errors>
  <errorsPerSecond>0.0</errorsPerSecond>
</statistics>
<status>GOOD</status>
<statusMessage>Synchronizing data</statusMessage>
<suspended>>false</suspended>
<type>ACTIVE_ACTIVE</type>
</link>

```

Query parameters for replication link actions

To perform actions on replication links, you use these query parameters:

- **suspend** — Suspend activity on the link.
- **resume** — Resume activity on a suspended link.
- **failOver** — Fail over the link to the remote system.
- **failBack** — For an active/active link, fail back the link.
- **beginRecovery** — For an active/passive link, begin data recovery.

- **completeRecovery** — For an active/passive link, complete data recovery.
- **restore** — Restore the link to the remote system.

You use these query parameters with a **POST** request against the link resource. You cannot include a request body with this request.

Here's a sample **POST** request that suspends activity on the replication link named MA-CA:

```
curl -k -iX POST
-H "Authorization: HCP YWxscm9sZXNM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication/links/MA-CA
?suspend"
```

local (data type for replication link failoverSettings local property)

The local data type covered in this section describes the local property of the failoverSettings data type that describes the failoverSettings property of the replication link resource.

local (data type for replication link failoverSettings local property)

Replication link failoverSettings local data type properties

The table below describes the properties included in the local data type that describes the local property of the failoverSettings data type.

Property name	Data type	Description	Notes
autoCompleteRecovery	Boolean	<p>For an active/passive link, specifies whether the complete recovery phase starts automatically while data is being recovered from the local system to the remote system for the replication link. Valid values are:</p> <ul style="list-style-type: none">• true — The complete recovery phase starts automatically.• false — The complete recovery phase does not start automatically. <p>The default is false.</p>	<p>This property is optional on a PUT request to create an active/passive link. It is not valid on a PUT or POST request for an active/active link and is not returned by any GET request for an active/active link.</p>

(Continued)

Property name	Data type	Description	Notes
autoCompleteRecovery Minutes	Integer	<p>For an active/passive link, specifies the number of minutes the up-to-date-as-of time for the replication link must be less than before HCP automatically starts the complete recovery phase while data is being recovered from the local system to the remote system for the link. Valid values are integers in the range one through 9,999. The default is 120.</p> <p>The up-to-date-as-of time is the difference between:</p> <ul style="list-style-type: none"> The date and time before which configuration changes and changes to namespace content are guaranteed to have been sent to the remote system The current date and time 	This property is optional on a PUT request to create an active/passive link. It is not valid on a PUT or POST request for an active/active link and is not returned by any GET request for an active/active link.
autoFailover	Boolean	<p>Specifies whether the replication link automatically fails over to the local system after a loss of connectivity to the remote system for the link. Valid values are:</p> <ul style="list-style-type: none"> true — The link automatically fails over. false — The link does not fail over automatically. <p>The default is false.</p>	This property is optional on a PUT request to create a replication link.

local (data type for replication link schedule local property)

(Continued)

Property name	Data type	Description	Notes
autoFailoverMinutes	Integer	Specifies the number of minutes HCP waits before automatically failing over the replication link to the local system after a loss of connectivity to the remote system for the link. Valid values are integers in the range seven through 9,999. The default is 120.	This property is optional on a PUT request to create a replication link.

Replication link failoverSettings local data type example

Here's an XML example of the local data type that describes the local property of the failoverSettings data type; the properties shown are those that are returned by a **GET** request sent to the replica for an active/passive link:

```
<local>
  <autoFailover>true</autoFailover>
  <autoFailoverMinutes>120</autoFailoverMinutes>
  <autoCompleteRecovery>false</autoCompleteRecovery>
  <autoCompleteRecoveryMinutes>60</autoCompleteRecoveryMinutes>
</local>
```

local (data type for replication link schedule local property)

The local data type covered in this section describes the local property of the schedule data type that describes the schedule resource for replication links.

Replication link schedule local data type properties

The table below describes the properties included in the local data type that describes the local property of the schedule data type.

Property name	Data type	Description	Notes
scheduleOverride	String	<p>Specifies an override for the local schedule for the replication link. Valid values are:</p> <ul style="list-style-type: none"> • LOW — The performance level is low for the entire week. • MEDIUM — The performance level is medium for the entire week. • HIGH — The performance level is high for the entire week. • CUSTOM — The performance level is the custom setting for the entire week. For information on setting the custom performance level, see <i>Replicating Tenants and Namespaces</i>. • NONE — The schedule for the link is not overridden. <p>These values are not case sensitive.</p> <p>To remove an existing override, specify NONE as the value for this property. If you don't explicitly remove an existing override when changing the local schedule, the override remains in effect.</p>	

(Continued)

Property name	Data type	Description	Notes
transition	transition	Specifies a scheduled change of performance level for the replication link on the local system. For details, see “transition” on page 265.	

Replication link schedule local data type example

Here's an XML example of the local data type that describes that describes the local property of the schedule data type:

```
<local>
  <scheduleOverride>NONE</scheduleOverride>
  <transition>
    <time>Sun:00</time>
    <performanceLevel>HIGH</performanceLevel>
  </transition>
  <transition>
    <time>Sun:06</time>
    <performanceLevel>MEDIUM</performanceLevel>
  </transition>
  <transition>
    <time>Sat:00</time>
    <performanceLevel>HIGH</performanceLevel>
  </transition>
  <transition>
    <time>Sat:06</time>
    <performanceLevel>MEDIUM</performanceLevel>
  </transition>
</local>
```

namespace

The namespace data type describes the namespaces resource.

namespace data type properties

The table below describes the properties included in the namespace data type.

Property name	Data type	Description	Notes
acIsUsage	String	<p>Specifies whether the namespace supports access control lists (ACLs) and, if so, whether HCP honors ACLs in the namespace. Valid values are:</p> <ul style="list-style-type: none"> NOT_ENABLED — The namespace does not support ACLs. ENFORCED — The namespace supports ACLs, and HCP honors ACLs in the namespace. NOT_ENFORCED — The namespace supports ACLs, but HCP does not honor ACLs in the namespace. <p>The default is NOT_ENABLED.</p> <p>These values are not case sensitive.</p>	<p>This property is optional on a PUT request.</p> <p>You can change the value of this property from NOT_ENABLED to ENFORCED or NOT_ENFORCED but not from ENFORCED or NOT_ENFORCED to NOT_ENABLED.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
allowPermissionAndOwnershipChanges	Boolean	<p>Specifies whether changes to POSIX UIDs, GIDs, and permissions and to object ownership are allowed for objects under retention in the namespace. Valid value are:</p> <ul style="list-style-type: none"> true — Changes to POSIX UIDs, GIDs, and permissions and to object ownership are allowed for objects under retention. false — Changes to POSIX UIDs, GIDs, and permissions and to object ownership are not allowed for objects under retention. <p>The default is false.</p>	<p>This property is optional on a PUT request.</p> <p>This property is not valid for the default namespace.</p>
appendEnabled	Boolean	<p>Specifies whether the namespace supports appendable objects. Valid values are:</p> <ul style="list-style-type: none"> true — The namespace supports appendable objects. false — The namespace does not support appendable objects. <p>The default is false.</p>	<p>This property is optional on a PUT request.</p> <p>You cannot enable both appendable objects and versioning for a namespace.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
atimeSynchronization Enabled	Boolean	<p>Specifies whether atime synchronization with retention settings is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Atime synchronization is enabled. false — Atime synchronization is not enabled. <p>The default is false.</p>	<p>This property is optional on a PUT request.</p> <p>This property is not valid for the default namespace.</p>
authAndAnonymous MinimumPermissions	List	<p>Lists the minimum data access permissions for all unauthenticated users of the namespace. Authenticated users also have these permissions when the value of the authUsersAlways-GrantedAllPermissions property is true.</p> <p>Valid values for permissions are:</p> <ul style="list-style-type: none"> BROWSE DELETE PURGE READ READ_ACL WRITE WRITE_ACL <p>The default is no permissions.</p> <p>These values are not case sensitive.</p>	<p>This property is optional on a PUT request.</p> <p>For the format of the permission list, see "Permission lists" on page 105.</p> <p>The set of permissions specified in the request body replaces the current set of minimum permissions for all users. To remove all the permissions for all users, specify an empty list.</p> <p>If the set of permissions includes PURGE, delete permission is enabled automatically. If the set of permissions includes READ, browse permission is enabled automatically.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
authMinimumPermissions	List	<p>Lists the minimum data access permissions for authenticated users of the namespace. Valid values for permissions are:</p> <ul style="list-style-type: none"> • BROWSE • DELETE • PURGE • READ • READ_ACL • WRITE • WRITE_ACL <p>The default is no permissions.</p> <p>These values are not case sensitive.</p>	<p>This property is optional on a PUT request.</p> <p>For the format of the permission list, see "Permission lists" on page 105.</p> <p>The set of permissions specified in the request body replaces the current set of permissions for authenticated users. To remove all the permissions for authenticated users, specify an empty list.</p> <p>If the set of permissions includes PURGE, delete permission is enabled automatically. If the set of permissions includes READ, browse permission is enabled automatically.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
authUsersAlwaysGranted AllPermissions	Boolean	<p>Specifies whether users have the minimum data access permissions specified by the <code>authAndAnonymous-MinimumPermissions</code> property in addition to the minimum data access permissions specified by the <code>authMinimumPermissions</code> property when using a protocol that requires authentication. Valid values are:</p> <ul style="list-style-type: none"> true — Users have the minimum data access permissions specified by both properties when using a protocol that requires authentication. false — Users have only the minimum data access permissions specified by the <code>authMinimum-Permissions</code> property when using a protocol that requires authentication. <p>The default is true.</p>	<p>This property is optional on a PUT request.</p> <p>This property is not valid for the default namespace.</p>
creationTime	String	<p>Specifies the date and time at which the namespace was created, in this ISO 8601 format:</p> <p><code>yyyy-MM-ddThh:mm:ssZ</code></p> <p>Z represents the offset from UTC, in this format:</p> <p><code>(+ -)hhmm</code></p> <p>For example:</p> <p>2012-02-09T19:44:03-0500</p>	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.</p>

(Continued)

Property name	Data type	Description	Notes
customMetadataIndexing Enabled	Boolean	<p>Specifies whether custom metadata indexing is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Custom metadata indexing is enabled. false — Custom metadata indexing is disabled. <p>The default is false.</p>	<p>This property is optional on a PUT request.</p> <p>You can set this property to true only while indexing is enabled for the namespace.</p> <p>Disabling custom metadata indexing for the namespace automatically disables full custom metadata indexing.</p>
customMetadata ValidationEnabled	Boolean	<p>Specifies whether custom metadata XML checking is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — When custom metadata is added to an object in the namespace, HCP checks whether the custom metadata is well-formed XML. false — When custom metadata is added to an object in the namespace, HCP does not check whether the custom metadata is well-formed XML. <p>The default is false.</p>	<p>This property is optional on a PUT request.</p>

(Continued)

Property name	Data type	Description	Notes
description	String	<p>Specifies a description of the namespace. This description is optional. The default is the description specified by the namespace defaults for the tenant.</p> <p>To remove a description from an existing namespace, specify the description property with no value.</p> <p>For the rules for descriptions, see “Common property values” on page 104.</p>	This property is optional on a PUT request.
dpl	String		<p>Deprecated. Namespace DPL is now a service plan function.</p> <p>This property is ignored on a PUT or POST request. In response to a GET request, the value of this property is always Dynamic.</p>
enterpriseMode	Boolean	<p>Specifies the namespace retention mode. Valid values are:</p> <ul style="list-style-type: none"> true — The namespace is in enterprise mode. false — The namespace is in compliance mode. <p>The default is the retention mode specified by the namespace defaults for the tenant.</p>	<p>This property is optional on a PUT request.</p> <p>You can set the value of this property to false only if the owning tenant is allowed to set the retention mode for its namespaces.</p> <p>In enterprise mode, privileged deletes are allowed, retention class durations can be shortened, and retention classes can be deleted. In compliance mode, these activities are not allowed.</p> <p>The value of this property can be changed from true to false but not from false to true.</p>

(Continued)

Property name	Data type	Description	Notes
fullyQualifiedName	String	Specifies the fully qualified hostname of the namespace.	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.
hardQuota	String	<p>For an HCP namespace, specifies the total amount of space allocated to the namespace. This is the space available for storing objects, including object data, metadata, and the redundant data required to satisfy the namespace service plan.</p> <p>Valid values are decimal numbers with up to two places after the period, followed by a space, followed by MB, GB, or TB (for example, 1.25 TB). The minimum value is 1 (one) for GB and .01 for TB. The maximum value is the amount of space the tenant has available to allocate to its namespaces. The default is the hard quota specified by the namespace defaults for the tenant.</p>	<p>This property is optional on a PUT request.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
hashScheme	String	<p>Specifies the cryptographic hash algorithm for the namespace. Valid values are:</p> <p>MD5 SHA-1 SHA-256 SHA-384 SHA-512 RIPEMD-160</p> <p>These values are case sensitive.</p> <p>The default is the cryptographic hash algorithm specified by the namespace defaults for the tenant.</p>	This property is optional on a PUT request. It is not valid on a POST request and is returned only by a verbose GET request.
id	String	<p>Specifies the system-supplied unique ID for the namespace. HCP generates this ID automatically when you create a namespace.</p>	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.
indexingDefault	Boolean	<p>For an HCP namespace, specifies the default index setting for objects added to the namespaces. Valid values are:</p> <ul style="list-style-type: none"> true — New objects are marked for indexing. false — New objects are not marked for indexing. <p>The default is true.</p>	<p>This property is optional on a PUT request.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
indexingEnabled	Boolean	<p>Specifies whether indexing is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Indexing is enabled. false — Indexing is disabled. <p>On a PUT request, the default is true if searchEnabled is set to true in the same request; otherwise, the default is false.</p>	<p>This property is optional on a PUT request.</p> <p>You can set this property to true only while search is enabled for the namespace.</p> <p>Disabling indexing for the namespace automatically disables custom metadata indexing.</p>
isDplDynamic	Boolean		<p>Deprecated. Namespace DPL is now a service plan function.</p> <p>This property is not valid on a PUT or POST request. In response to a GET request, the value of this property is always true.</p>
mqeIndexingTimestamp	String	<p>Specifies the date and time before which objects are guaranteed to have been indexed by the metadata query engine, in this ISO 8601 format:</p> <p><i>yyyy-MM-ddThh:mm:ssZ</i></p> <p>Z represents the offset from UTC, in this format:</p> <p><i>(+ -)hhmm</i></p> <p>For example:</p> <p>2012-02-09T19:44:03-0500</p>	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.</p>

(Continued)

Property name	Data type	Description	Notes
name	String	<p>Specifies the name of the namespace. HCP derives the hostname for the namespace from this name. The hostname is used in URLs for access to the namespace.</p> <p>In English, the name you specify for a namespace must be from one through 63 characters long and can contain only alphanumeric characters and hyphens (-) but cannot start or end with a hyphen. In other languages, because the derived hostname cannot be more than 63 characters long, the name you specify may be limited to fewer than 63 characters.</p> <p>Namespace names cannot contain special characters other than hyphens and are not case sensitive. White space is not allowed.</p> <p>Namespace names cannot start with <i>xn--</i> (that is, the characters <i>x</i> and <i>n</i> followed by two hyphens).</p> <p>Namespace names must be unique for the owning tenant. Different tenants can have namespaces with the same name.</p> <p>You can reuse namespace names that are not currently in use. So, for example, if you delete a namespace, you can give a new namespace the same name as the one the deleted namespace had.</p> <p>The name of the default namespace is always <i>Default</i>.</p>	<p>This property is required on a PUT request.</p> <p>The namespace name is used in the URL for access to the namespace.</p> <p>You can change the namespace name any time after you create the namespace, except while the HS3, CIFS, or NFS protocol is enabled for the namespace. However, when you change the name, the URL for the namespace may change as well.</p>

(Continued)

Property name	Data type	Description	Notes
owner	String	<p>For an HCP namespace, specifies the namespace owner. Valid values are:</p> <ul style="list-style-type: none"> If the owner has an HCP user account, the username for that account. If the owner has an AD user account, the account username along with the name of the AD domain in which the account is defined, in this format: <i>username@ad-domain-name</i> <p>If the namespace doesn't have an owner, this property has no value.</p>	<p>This property is optional on a PUT request.</p> <p>If this property is included on a PUT or POST request with a value that identifies an AD user account, the request must also include the <code>ownerType</code> property with a value of EXTERNAL.</p> <p>In response to a GET request, if the owner is an AD user and HCP cannot communicate with AD or cannot find the user account in AD, the value of this property is the security ID (SID) for the AD user account.</p> <p>This property is not valid for the default namespace.</p>
ownerType	String	<p>For an HCP namespace, specifies the type of the user account that identifies the object owner. Valid values are:</p> <ul style="list-style-type: none"> LOCAL — The user account is defined in HCP. EXTERNAL — The user account is defined in AD. <p>The default is LOCAL.</p> <p>These values are not case sensitive.</p> <p>If the namespace doesn't have an owner, this property has no value.</p>	<p>This property is optional on a PUT request.</p> <p>You can specify a value for this property on a PUT or POST request only if you specify a value for the <code>owner</code> property in the same request.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
optimizedFor	String	<p>Specifies whether the namespace is optimized for cloud or optimized for all access protocols. The valid values are:</p> <ul style="list-style-type: none"> • CLOUD - Enables cloud optimization • ALL - Enables all access protocols <p>The default value is whatever value was configured on the system administrator set on the System Management Console.</p>	<p>This property is optional on a PUT and POST request.</p> <p>This property is not valid for the default namespace</p>

(Continued)

Property name	Data type	Description	Notes
readFromReplica	Boolean	<p>Specifies whether read from replica is enabled for the namespace.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> true — Read from replica is enabled. false — Read from replica is disabled. <p>On a PUT request to create an HCP tenant, the default is true if replicationEnabled is set to true in the same request; otherwise, the default is false. On a PUT request to create the default tenant and namespace, the default is true if the HCP system supports replication; otherwise, the default is false.</p>	<p>This property is optional on a PUT request. You can set the value of this property to true on a PUT request only if, for an HCP namespace, the replicationEnabled property is set to true in the same request or, for the default namespace, the HCP system supports replication.</p> <p>On a POST request, you can set the value for this property to true for an HCP namespace only if the namespace is already selected for replication or the replicationEnabled property is set to true in the same request. You can set the value for this property to true for the default namespace only if the HCP system supports replication.</p> <p>This property is not returned by any GET request for an HCP namespace that is not selected for replication. This property is not returned by any GET request for the default namespace if the HCP system does not support replication,</p>

(Continued)

Property name	Data type	Description	Notes
replicationEnabled	Boolean	<p>For an HCP namespace, specifies whether the namespace is selected for replication. Valid values are:</p> <ul style="list-style-type: none"> true — The namespace is selected for replication. false — The namespace is not selected for replication. <p>The default is the replication setting specified by the namespace defaults for the tenant.</p>	<p>This property is optional on a PUT request.</p> <p>You can set the value of this property to true only if the owning tenant has replication configuration enabled.</p> <p>If the HCP system does not support replication, this property is not returned by any GET request.</p> <p>This property is not valid for the default namespace. If the HCP system supports replication, replication is automatically enabled for that namespace.</p>
replicationTimestamp	String	<p>Specifies the oldest change time for objects and configuration changes not yet replicated or recovered from the namespace, as applicable, in this format:</p> <p><i>yyyy-MM-ddThh:mm:ssZ</i></p> <p>Z represents the offset from UTC, in this format:</p> <p><i>(+ -)hhmm</i></p> <p>For example:</p> <p>2012-02-27T14:32:19-0500</p>	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request and only when these conditions are true:</p> <ul style="list-style-type: none"> The namespace is selected for replication. The tenant that owns the namespace is included in a replication link.

(Continued)

Property name	Data type	Description	Notes
searchEnabled	Boolean	<p>Specifies whether search is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Search is enabled. false — Search is disabled. <p>The default is the search setting specified by the namespace defaults for the tenant.</p>	<p>This property is optional on a PUT request.</p> <p>You can set the value of this property to true only if the owning tenant is allowed to enable search for its namespaces.</p> <p>Disabling search after it has been enabled:</p> <ul style="list-style-type: none"> Automatically disables indexing for the namespace. Removes objects in the namespace from the metadata query engine index. If you subsequently reenables search for the namespace, the namespace must be reindexed from scratch. Deletes the list of excluded annotations for the namespace. Deletes all associations the namespace has with any content classes.
servicePlan	String	<p>Specifies the service plan associated with the namespace. Valid values are names of existing service plans. The default is the service plan specified by the namespace defaults for the tenant.</p>	<p>This property is valid on a PUT or POST request and returned by a GET request only if the owning tenant is allowed to select service plans for its namespaces. If valid, this property is optional on a PUT request.</p>

(Continued)

Property name	Data type	Description	Notes
serviceRemoteSystem Requests	Boolean	<p>Specifies whether HCP can service HTTP requests against the namespace that are redirected from other HCP systems. Valid values are:</p> <ul style="list-style-type: none"> true — HCP can service requests against the namespace that are redirected from other systems. false — HCP cannot service requests against the namespace that are redirected from other systems. <p>The default is true.</p>	This property is optional on a PUT request.
softQuota	Integer	<p>For an HCP namespace, specifies the percentage point at which HCP notifies the tenant that free storage space for the namespace is running low. Valid values are integers in the range ten through 95. The default is the soft quota specified by the namespace defaults for the tenant.</p>	<p>This property is optional on a PUT request.</p> <p>This property is not valid for the default namespace.</p>

(Continued)

Property name	Data type	Description	Notes
tags	List	<p>Associates zero, one, or more tags with the namespace. Each tag can be from one through 64 characters long and can contain any valid UTF-8 characters except commas (,). White space is allowed.</p> <p>Tags are not case sensitive.</p>	<p>This property is optional on a PUT request.</p> <p>For an existing namespace, the set of tags specified in the request body replaces the set of tags currently associated with the namespace. To remove all tags, specify an empty set.</p> <p>In XML, each listed tag is the value of an element named tag. In JSON, the name in the name/value pair that lists the tags is tag.</p> <p>This property is not valid for the default namespace.</p>
versioningSettings	versioning Settings	<p>For an HCP namespace, specifies the versioning settings for the namespace.</p> <p>If this property is omitted on a PUT request, the default is the versioning settings specified by the namespace defaults for the tenant.</p> <p>For details, see "versioningSettings" on page 274.</p>	<p>This property is optional on a PUT request. If this property is included on a PUT request and the owning tenant is not allowed to set versioning for its namespaces, the value of the enabled property for versioningSettings, if specified, must be false.</p> <p>This property is not valid on a POST request and is not returned by a GET request. To modify or retrieve the versioning settings for a namespace, use the namespace versioning-Settings resource.</p> <p>This property is not valid for the default namespace.</p> <p>You cannot enable versioning for a namespace while the CIFS, NFS, WebDAV, or SMTP protocol or appendable objects are enabled.</p>

namespace data type example

Here's an XML example of the namespace data type; the properties shown are those that are returned in response to a verbose **GET** request for an HCP namespace on an HCP system that includes the replication feature:

```
<namespace>
  <aclsUsage>ENFORCED</aclsUsage>
  <authUsersAlwaysGrantedAllPermissions>true
</authUsersAlwaysGrantedAllPermissions>
  <allowPermissionAndOwnershipChanges>true
</allowPermissionAndOwnershipChanges>
  <appendEnabled>false</appendEnabled>
  <atimeSynchronizationEnabled>false</atimeSynchronizationEnabled>
  <authMinimumPermissions>
    <permission>BROWSE</permission>
    <permission>READ</permission>
    <permission>WRITE</permission>
  </authMinimumPermissions>
  <creationTime>2012-02-09T15:42:36-0500</creationTime>
  <customMetadataIndexingEnabled>true</customMetadataIndexingEnabled>
  <customMetadataValidationEnabled>true</customMetadataValidationEnabled>
  <description>Created for the Finance department at Example Company by Lee
    Green on 2/9/2012.</description>
  <dpl>Dynamic</dpl>
  <enterpriseMode>true</enterpriseMode>
  <fullyQualifiedName>accounts-receivable.finance.hcp.example.com
</fullyQualifiedName>
  <hardQuota>50 GB</hardQuota>
  <hashScheme>SHA-256</hashScheme>
  <indexingDefault>true</indexingDefault>
  <indexingEnabled>true</indexingEnabled>
  <isDplDynamic>true</isDplDynamic>
  <mqeIndexingTimestamp>2014-03-26T18:11:13-0400
</mqeIndexingTimestamp>
  <name>Accounts-Receivable</name>
  <owner>pblack</owner>
  <ownerType>LOCAL</ownerType>
  <optimizedFor>ALL</optimizedFor>
  <readFromReplica>true</readFromReplica>
  <replicationEnabled>true</replicationEnabled>
  <replicationTimestamp>2012-02-27T06:45:52-0500</replicationTimestamp>
```

```

<searchEnabled>true</searchEnabled>
<servicePlan>Short-Term-Activity</servicePlan>
<serviceRemoteSystemRequests>true</serviceRemoteSystemRequests>
<softQuota>75</softQuota>
<tags>
  <tag>Billing</tag>
  <tag>lgreen</tag>
</tags>
<id>0e774b8d-8936-4df4-a352-b68766b5c287</id>
<authAndAnonymousMinimumPermissions>
  <permission>BROWSE</permission>
  <permission>READ</permission>
</authAndAnonymousMinimumPermissions>
</namespace>

```

Query parameter for restarting indexing

While search is enabled for a namespace, you can restart metadata query engine indexing of the namespace from the time the namespace was created or from a specified date. To do this, you use this query parameter:

```
resetMQECheckpoint=(mm/dd/yyyy|0)
```

The value **0** (zero) restarts indexing from the time the namespace was created.

You use the **resetMQECheckpoint** query parameter with a **POST** request against the namespace resource.

Here's a sample **POST** request that restarts metadata query engine indexing of the accounts-receivable namespace from the time the namespace was created:

```

curl -k -i -d "<namespace/>" -H "Content-Type: application/xml"
-H "Authorization: HCP bGdyZWVu:a3b9c163f6c520407ff34cfdb83ca5c6"
"https://finance.hcp.example.com:9090/mapi/tenants/finance/namespaces/
accounts-receivable?resetMQECheckpoint=0"

```

namespaceDefaults

The namespaceDefaults data type describes the namespaceDefaults resource.

namespaceDefaults data type properties

The table below describes the properties included in the namespaceDefaults data type.

Property name	Data type	Description	Notes
description	String	<p>Specifies the default description for new HCP namespaces. This description is optional. The default is no description.</p> <p>To remove a description from the namespace defaults, specify the description property with no value.</p> <p>For the rules for descriptions, see "Common property values" on page 104.</p>	
dpl	String		<p>Deprecated. Namespace DPL is now a service plan function.</p> <p>This property is ignored on a POST request. In response to a GET request, the value of this property is always Dynamic.</p>
effectiveDpl	String		<p>Deprecated. Namespace DPL is now a service plan function.</p> <p>This property is not valid on a POST request and is returned only by a verbose GET request. In response to a verbose GET request, the value of this property is always Dynamic.</p>

(Continued)

Property name	Data type	Description	Notes
enterpriseMode	Boolean	<p>Specifies the default retention mode for new HCP namespaces. Valid values are:</p> <ul style="list-style-type: none"> true — The namespace is in enterprise mode. false — The namespace is in compliance mode. <p>The default is true.</p>	<p>In enterprise mode, privileged deletes are allowed, retention class durations can be shortened, and retention classes can be deleted. In compliance mode, these activities are not allowed.</p> <p>This property is valid on a POST request and returned by a GET request only if the tenant for which you're specifying namespace defaults is allowed to set the retention mode for its namespaces.</p>
hardQuota	String	<p>Specifies the default hard quota for new HCP namespaces.</p> <p>The hard quota is the total amount of space allocated to the namespace. This is the space available for storing objects, including object data, metadata, and the redundant data required to satisfy the namespace service plan.</p> <p>Valid values are decimal numbers with up to two places after the period, followed by a space, followed by MB, GB, or TB (for example, 1.25 TB). The minimum value is 1 (one) for GB and .01 for TB. The maximum value is equal to the hard quota for the tenant. The default is 50 GB.</p>	

(Continued)

Property name	Data type	Description	Notes
hashScheme	String	<p>Specifies the default cryptographic hash algorithm for new HCP namespaces. Valid values are:</p> <p>MD5 SHA-1 SHA-256 SHA-384 SHA-512 RIPEMD-160</p> <p>These values are case sensitive.</p> <p>The default is SHA-256.</p>	
replicationEnabled	Boolean	<p>Specifies the default replication setting for new HCP namespaces. Valid values are:</p> <ul style="list-style-type: none"> true — The namespace is selected for replication. false — The namespace is not selected for replication. <p>The default is false.</p>	<p>This property is valid on a POST request and returned by a GET request only if the tenant for which you're specifying namespace defaults has replication configuration enabled.</p>

(Continued)

Property name	Data type	Description	Notes
searchEnabled	Boolean	Specifies the default search setting for new HCP namespaces. Valid values are: <ul style="list-style-type: none"> true — Search is enabled. false — Search is disabled. The default is false .	This property is valid on a POST request and returned by a GET request only if the owning tenant is allowed to enable search for its namespaces.
servicePlan	String	Specifies the default service plan for new HCP namespaces. Valid values are names of existing service plans. The default is the Default service plan.	This property is valid on a POST request and returned by a GET request only if the tenant for which you're specifying namespace defaults is allowed to select service plans for its namespaces.
softQuota	Integer	Specifies the default soft quota for new HCP namespaces. Valid values are integers in the range ten through 95. The default is 85.	
versioningSettings	versioning Settings	Specifies the default versioning settings for new HCP namespaces. The default is no versioning. For details, see "versioningSettings" on page 274.	This property is valid on a POST request and returned by a GET request only if the tenant for which you're specifying namespace defaults has versioning configuration enabled.

namespaceDefaults data type example

Here's an XML example of the namespaceDefaults data type:

```
<namespaceDefaults>
  <description></description>
  <dpl>Dynamic</dpl>
  <effectiveDpl>Dynamic</effectiveDpl>
  <enterpriseMode>true</enterpriseMode>
  <hardQuota>75 GB</hardQuota>
  <hashScheme>SHA-256</hashScheme>
```

```

<replicationEnabled>true</replicationEnabled>
<searchEnabled>true</searchEnabled>
<servicePlan>Short-Term-Activity</servicePlan>
<softQuota>75</softQuota>
<versioningSettings>
  <enabled>true</enabled>
  <prune>true</prune>
  <pruneDays>10</pruneDays>
</versioningSettings>
</namespaceDefaults>

```

namespacePermission

The namespacePermission data type describes the namespacePermission property of the dataAccessPermissions data type.

namespacePermission data type properties

The table below describes the properties included in the namespacePermission data type.

Property name	Data type	Description	Notes
namespaceName	String	Specifies the name of the namespace to which the permissions listed in the permission property provide access.	

(Continued)

Property name	Data type	Description	Notes
permissions	List	<p>Lists the permissions associated with the namespace identified by the namespaceName property. Valid values for permissions are:</p> <ul style="list-style-type: none"> • BROWSE • CHOWN • DELETE • PRIVILEGED • PURGE • READ • READ_ACL • SEARCH • WRITE_ACL • WRITE <p>These values are not case sensitive.</p>	<p>For the format of the permission list, see "Permission lists" on page 105.</p> <p>The set of permissions specified in the request body replaces the set of permissions currently associated with the namespace identified by the namespaceName property. To remove all the permissions for a namespace, specify an empty list.</p> <p>If the set of permissions includes PURGE, delete permission is enabled automatically. If the set of permissions includes READ, browse permission is enabled automatically. If the set of permissions includes SEARCH, browse and read permissions are enabled automatically.</p>

namespacePermission data type example

Here's an XML example of the namespacePermission data type:

```

<namespacePermission>
  <namespaceName>Accounts-Receivable</namespaceName>
  <permissions>
    <permission>BROWSE</permission>
  
```

```

    <permission>CHOWN</permission>
    <permission>SEARCH</permission>
    <permission>PURGE</permission>
    <permission>WRITE_ACL</permission>
    <permission>DELETE</permission>
    <permission>PRIVILEGED</permission>
    <permission>READ</permission>
    <permission>READ_ACL</permission>
    <permission>WRITE</permission>
  </permissions>
</namespacePermission>

```

networkSettings

The network data type sets the downstream DNS to either basic or advanced mode.

networkSettings data type properties

The table below describes the properties included in the Network data type.

Property name	Data type	Description	Notes
downstreamDNSMode	String	<p>Specifies whether downstream DNS is set to basic or advanced mode. Valid values are:</p> <ul style="list-style-type: none"> BASIC ADVANCED <p>The default is basic.</p>	

networkSettings data type example

Here's an XML example of the Network data type:

```

<networkSettings>
  <downstreamDNSMode>ADVANCED</downstreamDNSMode>
</networkSettings>

```

nfsProtocol

The nfsProtocol data type describes the nfs resource for HCP namespaces.

nfsProtocol data type properties

The table below describes the properties included in the nfsProtocol data type.

Property name	Data type	Description	Notes
enabled	Boolean	Specifies whether the NFS protocol is enabled for the namespace. Valid values are: <ul style="list-style-type: none"> true — NFS is enabled. false — NFS is disabled. The default is false .	
gid	Integer	Specifies the default POSIX GID for objects that don't have an explicit POSIX GID. Valid values are integers greater than or equal to zero. The default is zero.	
ipSettings	ipSettings	Specifies which IP addresses can access the namespace through the NFS protocol. For details, see "ipSettings" on page 160.	
uid	Integer	Specifies the default POSIX UID for objects that don't have an explicit POSIX UID. Valid values are integers greater than or equal to zero. The default is zero.	

nfsProtocol data type example

Here's an XML example of the nfsProtocol data type:

```
<nfsProtocol>
  <enabled>true</enabled>
  <gid>0</gid>
```

```

<ipSettings>
  <allowAddresses>
    <ipAddress>192.168.140.10</ipAddress>
    <ipAddress>192.168.140.14</ipAddress>
    <ipAddress>192.168.140.15</ipAddress>
    <ipAddress>192.168.149.0/24</ipAddress>
  </allowAddresses>
</ipSettings>
<uid>0</uid>
</nfsProtocol>

```

protocols

The protocols data type describes the protocols resource for default namespaces.



Note: For HCP namespaces, the protocols resource is superseded by the .../protocols/http resource, which has a data type of httpProtocol. For information on that data type, see ["httpProtocol"](#) on page 154.

protocols data type properties

The table below describes the properties included in the protocols data type.

Property name	Data type	Description	Notes
httpEnabled	Boolean	<p>Specifies whether the HTTP protocol is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — HTTP is enabled. false — HTTP is disabled. <p>The default is false.</p>	

recipient

(Continued)

Property name	Data type	Description	Notes
httpsEnabled	Boolean	Specifies whether the HTTPS protocol is enabled for the namespace. Valid values are: <ul style="list-style-type: none">true — HTTPS is enabled.false — HTTPS is disabled. The default is true .	
ipSettings	ipSettings	Specifies which IP addresses can and cannot access the namespace through the HTTP and HTTPS protocols. For details, see “ipSettings” on page 160.	

protocols data type example

Here’s an XML example of the protocols data type:

```
<protocols>
  <httpEnabled>>false</httpEnabled>
  <httpsEnabled>>true</httpsEnabled>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.140.10</ipAddress>
      <ipAddress>192.168.140.14</ipAddress>
      <ipAddress>192.168.140.15</ipAddress>
      <ipAddress>192.168.149.0/24</ipAddress>
    </allowAddresses>
    <allowIfInBothLists>>false</allowIfInBothLists>
    <denyAddresses>
      <ipAddress>192.168.149.5</ipAddress>
    </denyAddresses>
  </ipSettings>
</protocols>
```

recipient

The recipient data type describes the recipient property of the recipients data type.

recipient data type properties

The table below describes the properties included in the recipient data type.

Property name	Data type	Description	Notes
address	String	Specifies a comma-separated list of well-formed email addresses for notification about messages added to the tenant log.	This property is required on a POST request.
importance	String	<p>Specifies whether HCP should send email only about important log messages to the specified email addresses or send email about all log messages. Valid values are:</p> <ul style="list-style-type: none">• ALL — Send email about all log messages.• MAJOR — Send email only about important log messages. <p>The default is MAJOR.</p> <p>These values are not case sensitive.</p>	

recipient

(Continued)

Property name	Data type	Description	Notes
severity	String	<p>Specifies the severity level of the log messages about which to send email. Valid values are:</p> <ul style="list-style-type: none">• NOTICE — Send email about log messages with a severity level of notice, warning, or error.• WARNING — Send email about log messages with a severity level of warning or error.• ERROR — Send email only about log messages with a severity level of error. <p>The default is ERROR.</p> <p>These values are not case sensitive.</p>	

(Continued)

Property name	Data type	Description	Notes
type	String	<p>Specifies the types of log messages about which to send email. Valid values are comma-separated lists of one or more of:</p> <ul style="list-style-type: none"> • GENERAL — Send email about log messages that do not have a type of security or compliance. • SECURITY — Send email about log messages with a type of security. • COMPLIANCE — Send email about log messages with a type of compliance. <p>The default is GENERAL.</p> <p>These values are not case sensitive.</p>	

recipient data type example

Here's an XML example of the recipient data type:

```
<recipient>
  <address>lgreen@example.com, sgold@example.com</address>
  <importance>MAJOR</importance>
  <severity>ERROR</severity>
  <type>GENERAL,SECURITY</type>
</recipient>
```

recipients

The recipients data type describes the recipients property of the emailNotification data type.

recipients data type properties

The table below describes the property included in the recipients data type.

Property name	Data type	Description	Notes
recipient	recipient	<p>Specifies:</p> <ul style="list-style-type: none">• One or more email addresses to which HCP sends email about log messages• The types of log messages about which HCP sends email to those addresses <p>For details, see "recipient" on page 208.</p>	<p>Include one instance of this property (up to 25) for each set of email addresses you want in the recipients list for email notification.</p> <p>The set of recipients specified in the request body replaces the set of recipients currently configured for email notification.</p> <p>Recipients are added to the Bcc list for each email.</p> <p>Because each instance of the recipient property can specify multiple email addresses, you can specify a total of more than 25 addresses across all instances of this property. However, if you specify more than 25, HCP sends each email only to an arbitrary 25 of them.</p>

recipients data type example

Here's an XML example of the recipients data type property:

```
<recipients>
  <recipient>
    <address>lgreen@example.com, sgold@example.com</address>
    <importance>MAJOR</importance>
    <severity>ERROR</severity>
    <type>GENERAL,SECURITY</type>
  </recipient>
  <recipient>
    <address>mwhite@example.com</address>
    <importance>ALL</importance>
    <severity>WARNING</severity>
    <type>COMPLIANCE</type>
  </recipient>
</recipients>
```

remote (data type for replication link failoverSettings remote property)

The local data type covered in this section describes the remote property of the failoverSettings data type.

Replication link failoverSettings remote data type properties

The table below describes the properties included in the remote data type that describes the remote property of the failoverSettings data type.

Property name	Data type	Description	Notes
autoCompleteRecovery	Boolean	<p>For an active/passive link, specifies whether the complete recovery phase starts automatically while data is being recovered from the remote system for the replication link to the local system. Valid values are:</p> <ul style="list-style-type: none"> true — The complete recovery phase starts automatically. false — The complete recovery phase does not start automatically. <p>The default is false.</p>	<p>This property is optional on a PUT request to create an active/passive link. It is not valid on a PUT or POST request for an active/active link and is not returned by any GET request for an active/active link.</p>
autoCompleteRecovery Minutes	Integer	<p>For an active/passive link, specifies the number of minutes the up-to-date-as-of time for the replication link must be less than before HCP automatically starts the complete recovery phase while data is being recovered from the remote system for the link to the local system. Valid values are integers in the range one through 9,999. The default is 120.</p> <p>The up-to-date-as-of time is the difference between:</p> <ul style="list-style-type: none"> The date and time before which configuration changes and changes to namespace content are guaranteed to have been sent to the local system The current date and time 	<p>This property is optional on a PUT request to create an active/passive link. It is not valid on a PUT or POST request for an active/active link and is not returned by any GET request for an active/active link.</p>

(Continued)

Property name	Data type	Description	Notes
autoFailover	Boolean	<p>Specifies whether the replication link automatically fails over to the remote system for the link after a loss of connectivity to the local system. Valid values are:</p> <ul style="list-style-type: none"> true — The link automatically fails over. false — The link does not fail over automatically. <p>The default is false.</p>	This property is optional on a PUT request to create a replication link.
autoFailoverMinutes	Integer	<p>Specifies the number of minutes HCP waits before automatically failing over the replication link to the remote system for the link after a loss of connectivity to the local. Valid values are integers in the range seven through 9,999. The default is 120.</p>	This property is optional on a PUT request to create a replication link.

Replication link failoverSettings remote data type example

Here's an XML example of the remote data type that describes the remote property of the failoverSettings data type; the properties shown are those that are returned by a **GET** request sent to the primary system for an active/passive link:

```
<remote>
  <autoFailover>true</autoFailover>
  <autoFailoverMinutes>120</autoFailoverMinutes>
  <autoCompleteRecovery>false</autoCompleteRecovery>
  <autoCompleteRecoveryMinutes>60</autoCompleteRecoveryMinutes>
</remote>
```

remote (data type for replication link schedule remote property)

remote (data type for replication link schedule remote property)

The remote data type covered in this section describes the remote property of the schedule data type that describes the schedule resource for replication links.

Replication link schedule remote data type properties

The table below describes the properties included in the remote data type that describes the remote property of the schedule data type.

remote (data type for replication link schedule remote property)

Property name	Data type	Description	Notes
scheduleOverride	String	<p>Specifies an override for the local schedule for the replication link. Valid values are:</p> <ul style="list-style-type: none">• LOW — The performance level is low for the entire week.• MEDIUM — The performance level is medium for the entire week.• HIGH — The performance level is high for the entire week.• CUSTOM — The performance level is the custom setting for the entire week. For information on setting the custom performance level, see <i>Replicating Tenants and Namespaces</i>.• NONE — The schedule for the link is not overridden. <p>These values are not case sensitive.</p> <p>To remove an existing override, specify NONE as the value for this property. If you don't explicitly remove an existing override when changing the local schedule, the override remains in effect.</p>	
transition	transition	<p>Specifies a scheduled change of performance level for the replication link on the local system.</p> <p>For details, see “transition” on page 265.</p>	

Replication link schedule remote data type example

Here's an XML example of the local data type that describes that describes the local property of the schedule data type:

```
<remote>
  <scheduleOverride>NONE</scheduleOverride>
  <transition>
    <time>Sun:00</time>
    <performanceLevel>HIGH</performanceLevel>
  </transition>
  <transition>
    <time>Mon:00</time>
    <performanceLevel>MEDIUM</performanceLevel>
  </transition>
</remote>
```

replicationCollisionSettings

The replicationCollisionSettings data type describes the replicationCollisionSettings resource for namespaces.

replicationCollisionSettings data type properties

The table below describes the properties included in the replicationCollisionSettings data type.

Property name	Data type	Description	Notes
action	String	<p>Specifies how HCP should handle objects flagged as replication collisions. Valid values are:</p> <ul style="list-style-type: none"> MOVE — Move the object to the <code>.lost+found</code> directory in the same namespace. RENAME — Rename the object in place. <p>When renaming an object, HCP changes the object name to <code>object-name.collusion</code>.</p> <p>These values are not case sensitive.</p> <p>The default is MOVE.</p>	
deleteDays	Integer	<p>Specifies the number of days objects flagged as replication collisions must remain in the namespace before they are automatically deleted. Valid values are integers in the range zero through 36,500 (that is, 100 years). A value of zero means delete immediately.</p>	<p>This property is required on a POST request when the value of the deleteEnabled property is true. It is invalid on a POST request when the value of the deleteEnabled property is false.</p>

(Continued)

Property name	Data type	Description	Notes
deleteEnabled	Boolean	<p>Specifies whether HCP should automatically delete objects flagged as replication collisions. Valid values are:</p> <ul style="list-style-type: none"> true — Automatically delete objects flagged as replication collisions after the number of days specified by the deleteDays property. false — Do not automatically delete objects flagged as replication collisions. 	

replicationCollisionSettings data type example

Here's an XML example of the replicationCollisionSettings data type property:

```
<replicationCollisionSettings>
  <action>MOVE</action>
  <deleteDays>10</deleteDays>
  <deleteEnabled>true</deleteEnabled>
</replicationCollisionSettings>
```

replicationService

The replicationService data type describes the replication resource.

replicationService data type properties

The table below describes the properties included in the replicationService data type.

Property name	Data type	Description	Notes
allowTenantsToMonitorNamespaces	Boolean	<p>Specifies whether the Tenant Management Console for HCP tenants displays the status of replication of the tenant and its namespaces. Valid values are:</p> <ul style="list-style-type: none"> true — The Tenant Management Console displays replication status information for all HCP tenants. false — The Tenant Management Console hide replication status information for all HCP tenants. <p>The default is false.</p>	
enabledDNSFailover	Boolean	<p>Specifies whether DNS failover is enabled for the HCP system. Valid values are:</p> <ul style="list-style-type: none"> true — DNS failover is enabled for the system. false — DNS failover is disabled for the system. <p>The default is false.</p>	

(Continued)

Property name	Data type	Description	Notes
enableDomainAndCertificateSynchronization	Boolean	<p>Specifies whether HCP periodically sends its domains and SSL server certificates to each other HCP system with which it participates as a sending system in a replication link. Valid values are:</p> <ul style="list-style-type: none"> true — HCP periodically sends its domains and SSL server certificates to each other system with which it participates as a sending system in a replication link. false — HCP does not send its domains and SSL server certificates to other systems. <p>The default is false.</p>	
network	String	<p>Specifies the replication network for the HCP system. Valid values are any network defined in the HCP system except [hcp_backend]. The default is [hcp_system].</p> <p>Network names are not case sensitive.</p>	

(Continued)

Property name	Data type	Description	Notes
status	String	<p>Specifies whether all activity on all replication links in which the HCP system participates is currently stopped. Possible values are:</p> <ul style="list-style-type: none"> • ENABLED — Activity on each replication link in which the system participates is occurring according the individual link status. • SHUTDOWN — All activity on all replication links in which the system participates is currently stopped. 	This property is not valid on a POST request. It is returned only by a verbose GET request.

replicationService data type example

Here's an XML example of the replicationService data type:

```
<replicationService>
  <allowTenantsToMonitorNamespaces>false
</allowTenantsToMonitorNamespaces>
  <enableDNSFailover>true</enableDNSFailover>
  <enableDomainAndCertificateSynchronization>true
</enableDomainAndCertificateSynchronization>
  <network>[hcp_system]</network>
  <status>ENABLED</status>
</replicationService>
```

Query parameters for replication service actions

To shut down all replication links in which the HCP system participates, you use this query parameter:

shutDownAllLinks=*reason*

reason is a text string that specifies the reason why you're shutting down all links. This string can be up to 1,024 characters long and can contain any valid UTF-8 characters, including white space. The string you specify must be percent encoded, as described in ["Percent-encoding for special characters"](#) on page 27.

To reestablish all replication links in which the HCP system participates after they have been shut down, you use this query parameter:

`reestablishAllLinks`

You use the **shutDownAllLinks** and **reestablishAllLinks** query parameters with a **POST** request against the replication resource. You cannot include a request body with this request.

Here's a sample **POST** request that shuts down all replication links:

```
curl -k -iX POST
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication
?shutDownAllLinks=More%20bandwidth%20for%20app%20XYZ"
```

retentionClass

The retentionClass data type describes the retentionClasses resource.

retentionClass data type properties

The table below describes the properties included in the retentionClass data type.

Property name	Data type	Description	Notes
allowDisposition	Boolean	<p>Specifies whether HCP automatically deletes expired objects in the retention class. Valid values are:</p> <ul style="list-style-type: none"> true — Automatically delete expired objects. false — Do not automatically delete expired objects. 	<p>This property is required on a PUT request when the value of the retention class is an offset. It is ignored if the value is not an offset.</p> <p>This property is required on a POST request when the retention class value is being changed to an offset from another type of value.</p>

(Continued)

Property name	Data type	Description	Notes
description	String	<p>Specifies a description of the retention class. This description is optional. The default is no description.</p> <p>To remove a description from an existing retention class, specify the description property with no value.</p> <p>For the rules for descriptions, see “Common property values” on page 104.</p>	
name	String	<p>Specifies the name of the retention class. Retention class names must be from one through 64 characters long, can contain only alphanumeric characters, hyphens (-), and underscores (_), and are not case sensitive.</p> <p>The retention class name must be unique for the namespace. Different namespaces can have retention classes with the same name.</p>	This property is required on a PUT request. It is not valid on a POST request and is returned only by a verbose GET request.
value	String	<p>Specifies the retention class value. Valid values are special values and offsets. For information on specifying these values, see “Specifying retention values” on page 277.</p>	This property is required on a PUT request.

retentionClass data type example

Here's an XML example of the retentionClass data type:

```
<retentionClass>
  <description>Implements Finance department standard #42 - keep for 10
    years.</description>
  <value>A+10y</value>
```

```

    <allowDisposition>true</allowDisposition>
    <name>FN-Std-42</name>
  </retentionClass>

```

schedule

The schedule data type describes the schedule resource for replication links.

schedule data type properties

The table below describes the properties included in the schedule data type.

Property name	Data type	Description	Notes
local	local	Specifies the schedule for the replication link on the local system. For details, see “local (data type for replication link schedule local property)” on page 176.	This property is not valid on a POST request for an inbound active/passive link.
remote	remote	Specifies the schedule for the replication link on the remote system. For details, see “remote (data type for replication link schedule remote property)” on page 216.	This property is not valid on a POST request for an outbound active/passive link.

schedule data type example

Here’s an XML example of the schedule data type:

```

<schedule>
  <local>
    <scheduleOverride>NONE</scheduleOverride>
    <transition>
      <time>Sun:00</time>
      <performanceLevel>HIGH</performanceLevel>
    </transition>
    <transition>
      <time>Sun:06</time>
      <performanceLevel>MEDIUM</performanceLevel>
    </transition>
  </local>
</schedule>

```

```

</transition>
<transition>
  <time>Sat:00</time>
  <performanceLevel>HIGH</performanceLevel>
</transition>
<transition>
  <time>Sat:06</time>
  <performanceLevel>MEDIUM</performanceLevel>
</transition>
</local>
<remote>
  <scheduleOverride>NONE</scheduleOverride>
  <transition>
    <time>Sun:00</time>
    <performanceLevel>HIGH</performanceLevel>
  </transition>
  <transition>
    <time>Mon:00</time>
    <performanceLevel>MEDIUM</performanceLevel>
  </transition>
</remote>
</schedule>

```

searchSecurity

The searchSecurity data type describes the searchSecurity resource for tenants.

searchSecurity data type property

The table below describes the property included in the searchSecurity data type.

Property name	Data type	Description	Notes
ipSettings	ipSettings	Specifies which IP addresses can and cannot access the Search Console for the tenant. For details, see "ipSettings" on page 160.	

searchSecurity data type example

Here's an XML example of the searchSecurity data type:

```
<searchSecurity>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.140.36</ipAddress>
      <ipAddress>192.168.140.109</ipAddress>
    </allowAddresses>
    <allowIfInBothLists>false</allowIfInBothLists>
    <denyAddresses/>
  </ipSettings>
</searchSecurity>
```

smtpProtocol

The smtpProtocol data type describes the smtp resource for HCP namespaces.

smtpProtocol data type properties

The table below describes the properties included in the smtpProtocol data type.

Property name	Data type	Description	Notes
emailFormat	String	<p>Specifies the format for email objects added through the SMTP protocol. Valid values are:</p> <ul style="list-style-type: none"> • .eml • .mbox <p>The default is .eml.</p>	

(Continued)

Property name	Data type	Description	Notes
emailLocation	String	<p>Specifies the path for the directory in which to store email objects added through the SMTP protocol. This is the path relative to the root of the namespace (that is, <code>rest</code>, <code>data</code>, or <code>fcfs_data</code>). Start and end the path with a forward slash (/), like this:</p> <p style="text-align: center;">/email/company_all/</p> <p>The default is <code>/email/</code>.</p> <p>Directory names are case sensitive.</p>	If the specified directory does not exist, HCP creates it.
enabled	Boolean	<p>Specifies whether the SMTP protocol is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — SMTP is enabled. false — SMTP is disabled. <p>The default is false.</p>	
ipSettings	ipSettings	<p>Specifies which IP addresses can and cannot access the namespace through the SMTP protocol.</p> <p>For details, see “ipSettings” on page 160.</p>	
separateAttachments	Boolean	<p>Specifies whether HCP should store attachments separately from the email they accompany. Valid values are:</p> <ul style="list-style-type: none"> true — Store attachments separately. false — Do not store attachments separately. <p>The default is false.</p>	

smtpProtocol data type example

Here's an XML example of the smtpProtocol data type:

```
<smtpProtocol>
  <emailFormat>.eml</emailFormat>
  <emailLocation>/finance/email/</emailLocation>
  <enabled>true</enabled>
  <ipSettings>
    <allowAddresses>
      <ipAddress>192.168.45.213</ipAddress>
    </allowAddresses>
    <denyAddresses/>
  </ipSettings>
  <separateAttachments>>false</separateAttachments>
</smtpProtocol>
```

statistics (data type for replication link statistics property)

The statistics data type covered in this section describes the statistics property of the link data type.

statistics (data type for replication link statistics property)

Replication link statistics data type properties

The table below describes the properties included in the statistics data type that describes the statistics property of the link data type.

Property name	Data type	Description	Notes
bytesPending	Long	<p>Specifies the approximate amount of data currently waiting to be replicated from the local system to the remote system for the replication link, in bytes. This value is the sum of the amounts of data waiting to be sent in each HCP namespace included in the link. This value does not include data in the default namespace.</p> <p>If the local system is the replica for an active/passive link, the value of this property during replication is zero. If the local system is the primary system for an active/passive link, the value of this property during data recovery is zero.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.
bytesPendingRemote	Long	<p>Specifies the approximate amount of data currently waiting to be replicated from the remote system for the replication link to the local system, in bytes. This value is the sum of the amounts of data waiting to be sent in each HCP namespace included in the link. This value does not include data in the default namespace.</p> <p>For an active/passive link, the value of this property is always zero.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.

statistics (data type for replication link statistics property)

(Continued)

Property name	Data type	Description	Notes
bytesPerSecond	Double	<p>Specifies the current rate of data transmission on the replication link, in bytes per second.</p> <p>For an active/active link, the value of this property is the rate of data transmission from the local system to the remote system. For an active/passive link, the value of this property is the rate of data transmission during replication or recovery, whichever is happening at the time. In any case, the data transmission rate is cumulative for all the HCP namespaces and default-namespace directories included in the link.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.
bytesReplicated	Long	Specifies the total number of bytes of data replicated from the local system to the remote system for the replication link since the link was created.	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.
errors	Long	Specifies the total number of errors that have occurred during replication or recovery from the local system to the remote system for the replication link since the link was created.	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.

(Continued)

Property name	Data type	Description	Notes
errorsPerSecond	Double	<p>Specifies the current rate of errors on the replication link per second.</p> <p>For an active/active link, the value of this property is the error rate for replication from the local system to the remote system for the link. For an active/passive link, the value of this property is the error rate during replication or recovery, whichever is happening at the time.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.
objectsPending	Long	<p>Specifies the approximate number of objects currently waiting to be replicated from the local system to the remote system for the replication link. This value is the sum of the numbers of objects waiting to be sent in each HCP namespace included in the link. This value does not include objects in the default namespace.</p> <p>If the local system is the replica for an active/passive link, the value of this property during replication is zero. If the local system is the primary system for an active/passive link, the value of this property during data recovery is zero.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.

statistics (data type for replication link statistics property)

(Continued)

Property name	Data type	Description	Notes
objectsPendingRemote	Long	<p>Specifies the approximate number of objects currently waiting to be replicated from the remote system for the replication link to the local system. This value is the sum of the numbers of objects waiting to be sent in each HCP namespace included in the link. This value does not include objects in the default namespace.</p> <p>For an active/passive link, the value of this property is always zero.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.
objectsReplicated	Long	Specifies the total number of objects replicated from the local system to the remote system for the replication link since the link was created.	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.

(Continued)

Property name	Data type	Description	Notes
operationsPerSecond	Double	<p>Specifies the current rate of operations on the replication link per second. An operation is the replication of any of these:</p> <ul style="list-style-type: none"> An object, directory, symbolic link, metadata change, or object deletion An HCP tenant or HCP namespace or the modification or deletion of an HCP tenant or HCP namespace For HCP tenants only, the creation, modification, or deletion of a user account The creation, modification, or deletion of a retention class The creation, modification, or deletion of a content class A tenant log message <p>For an active/active link, the value of this property is the operation rate for replication from the local system to the remote system for the link. For an active/passive link, the value of this property is the operation rate during replication or recovery, whichever is happening at the time. In any case, the operation rate is cumulative for all the tenants being replicated or recovered on the link.</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.

statistics (data type for replication link statistics property)

(Continued)

Property name	Data type	Description	Notes
upToDateAsOfMillis	Long	<p>Specifies, in milliseconds since January 1, 1970, at 00:00:00 UTC:</p> <ul style="list-style-type: none">• For an active/active link, the date and time before which configuration changes and changes to namespace content are guaranteed to be synchronized in both directions between the two systems involved in the link• For an active/passive link, the date and time before which configuration changes and changes to namespace content are guaranteed to have been replicated or recovered on the link, as applicable	<p>This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.</p>

(Continued)

Property name	Data type	Description	Notes
upToDateAsOfString	String	<p>Specifies, as a datetime string:</p> <ul style="list-style-type: none"> For an active/active link, the date and time before which configuration changes and changes to namespace content are guaranteed to be synchronized in both directions between the two systems involved in the link For an active/passive link, the date and time before which configuration changes and changes to namespace content are guaranteed to have been replicated or recovered on the link, as applicable <p>The datetime string has this format:</p> <p><i>yyyy-MM-ddT hh:mm:ssZ</i></p> <p>In this format, <i>hh</i> is the hour on a 24-hour clock, and <i>Z</i> represents the offset from UTC and is specified as:</p> <p><i>(+ -)hhmm</i></p> <p>For example:</p> <p>2014-03-18T10:47:59-0400</p>	This property is not valid on a PUT or POST request for a replication link. It is returned only by a verbose GET request.

statistics (data type for tenant and namespace statistics resources)

Replication link statistics data type example

Here's an XML example of the statistics data type that describes the statistics property of the link data type; the properties shown are those that are returned by a verbose **GET** request for an active/active link:

```
<statistics>
  <upToDateAsOfString>2014-03-18T10:47:59-0400</upToDateAsOfString>
  <upToDateAsOfMillis>1395154079640</upToDateAsOfMillis>
  <bytesPending>189027593061</bytesPending>
  <bytesPendingRemote>319740</bytesPendingRemote>
  <bytesReplicated>72254119306967</bytesReplicated>
  <bytesPerSecond>56215390</bytesPerSecond>
  <objectsPending>534</objectsPending>
  <objectsPendingRemote>2</objectsPendingRemote>
  <objectsReplicated>295661</objectsReplicated>
  <operationsPerSecond>119.1</operationsPerSecond>
  <errors>0</errors>
  <errorsPerSecond>0.0</errorsPerSecond>
</statistics>
```

statistics (data type for tenant and namespace statistics resources)

The statistics data type covered in this section describes the statistics resource for tenants and namespaces.

Tenant and namespace statistics data type properties

The table below describes the properties included in the statistics data type that describes the statistics resource for tenants and namespaces.

Property name	Data type	Description	Notes
compressedCount	Long	Specifies the number of compressed objects in a given namespace or in all the namespaces owned by a given tenant.	This property is included in the response body only when the request is made using a system-level user account.
compressedSavedSize	Long	Specifies the total number of bytes of storage freed by compression in a given namespace or in all the namespaces owned by a given tenant.	This property is included in the response body only when the request is made using a system-level user account.

(Continued)

Property name	Data type	Description	Notes
customMetadataCount	Long	Specifies the number of custom metadata files stored in a given namespace or in all the namespaces owned by a given tenant.	
customMetadataSize	Long	Specifies the total number of bytes of custom metadata stored in a given namespace or in all the namespaces owned by a given tenant.	
ingestedVolume	Long	Specifies the total size of the stored data and custom metadata, in bytes, before it was added to a given namespace or to all the namespaces owned by a given tenant.	
objectCount	Long	Specifies the number of objects in a given namespace or in all the namespaces owned by a given tenant.	
shredCount	Long	Specifies the number of objects waiting to be shredded in a given namespace or in all the namespaces owned by a given tenant.	
shredSize	Long	Specifies the total number of bytes of object data and metadata waiting to be shredded in a given namespace or in all the namespaces owned by a given tenant.	
storageCapacityUsed	Long	Specifies the total number of bytes occupied by stored data in the given namespace or in all the namespaces owned by the given tenant. This includes object data, metadata, and any redundant data required to satisfy the applicable service plans.	

tenant (data type for replication link content tenant resource)

Tenant and namespace statistics data type example

Here's an XML example of the statistics data type that describes the statistics resource for tenants and namespaces; the properties shown are those that are returned in response to a **GET** request for tenant statistics where the request is made using a system-level user account:

```
<statistics>
  <compressedCount>854</compressedCount>
  <compressedSavedSize>413696</compressedSavedSize>
  <customMetadataCount>5</customMetadataCount>
  <customMetadataSize>3276</customMetadataSize>
  <objectCount>1616</objectCount>
  <shredCount>0</shredCount>
  <shredSize>0</shredSize>
  <storageCapacityUsed>884656</storageCapacityUsed>
</statistics>
```

tenant (data type for replication link content tenant resource)

The tenant data type covered in this section describes the *tenant-name* resource for replication link content.

Replication link content tenant data type properties

The table below describes the properties included in the tenant data type that describes the *tenant-name* resource for replication link content.

Property name	Data type	Description	Notes
bytesPending	Long	<p>For an active/active link, an outbound active/passive link during replication, or an inbound active/passive link during data recovery, specifies the approximate amount of data that is in the namespaces owned by the tenant and that is currently waiting to be replicated from the local system to the remote system for the replication link, in bytes.</p> <p>For an outbound link during data recovery or for an inbound link during replication, the value of this property is always zero. This value is also always zero for the default tenant.</p>	
bytesPendingRemote	Long	<p>Specifies the approximate amount of data that is in the namespaces owned by the tenant and that is currently waiting to be replicated from the remote system for the replication link to the local system, in bytes.</p> <p>For an active/passive link, the value of this property is always zero. This value is also always zero for the default tenant.</p>	

tenant (data type for replication link content tenant resource)

(Continued)

Property name	Data type	Description	Notes
objectsPending	Long	<p>For an active/active link, an outbound active/passive link during replication, or an inbound active/passive link during data recovery, specifies the approximate number of objects that are in the namespaces owned by the tenant and that are currently waiting to be replicated from the local system to the remote system for the replication link.</p> <p>For an outbound link during data recovery or for an inbound link during replication, the value of this property is always zero. This value is also always zero for the default tenant.</p>	
objectsPendingRemote	Long	<p>Specifies the approximate number of objects that are in the namespaces owned by the tenant and that are currently waiting to be replicated from the remote system for the replication link to the local system.</p> <p>For an active/passive link, the value of this property is always zero. This value is also always zero for the default tenant.</p>	

(Continued)

Property name	Data type	Description	Notes
status	String	<p>Specifies the status of activity for the tenant on the replication link. Possible values are:</p> <ul style="list-style-type: none"> • AUTO_PAUSED — HCP automatically paused replication or recovery of the tenant. • PAUSED — A user paused replication or recovery of the tenant. • REPLICATING — Replication or recovery of the tenant is proceeding normally. 	
upToDateAsOfMillis	Long	<p>Specifies, in milliseconds since January 1, 1970, at 00:00:00 UTC, the date and time before which configuration changes and changes to namespace content for all the replicating namespaces owned by the tenant are guaranteed to have been replicated from the local system to the remote system for the replication link.</p>	

tenant (data type for replication link content tenant resource)

(Continued)

Property name	Data type	Description	Notes
upToDateAsString	String	<p>Specifies, as a datetime string, the date and time before which configuration changes and changes to namespace content for all the replicating namespaces owned by the tenant are guaranteed to have been replicated from the local system to the remote system for the replication link.</p> <p>The datetime string has this format:</p> <p><i>yyyy-MM-ddT hh:mm:ssZ</i></p> <p>In this format, <i>hh</i> is the hour on a 24-hour clock, and <i>Z</i> represents the offset from UTC and is specified as:</p> <p><i>(+ -)hhmm</i></p> <p>For example:</p> <p>2014-03-18T10:47:59-0400</p>	

Replication link content tenant data type example

Here's an XML example of the tenant data type that describes the *tenant-name* resource for replication link content; the properties shown are those that are returned by a **GET** request for a tenant in an active/active link:

```
<tenant>
  <status>REPLICATING</status>
  <upToDateAsString>2014-03-19T17:45:15-0400</upToDateAsString>
  <upToDateAsOfMillis>1395265515303</upToDateAsOfMillis>
  <objectsPending>196</objectsPending>
  <bytesPending>46347338966</bytesPending>
  <objectsPendingRemote>14</objectsPendingRemote>
  <bytesPendingRemote>735856</bytesPendingRemote>
</tenant>
```

Query parameters for replication link content tenant actions

To perform actions on tenants in replication links, you use these query parameters:

- **pause** — Pauses replication or recovery for the tenant on the replication link, as applicable.
- **resume** — Resumes replication or recovery for the tenant on the replication link, as applicable.

You use these query parameters with a **POST** request against the replication link content tenant resource. You cannot include a request body with this request.

Here's a sample **POST** request that pauses replication activity for the Finance tenant on the replication link named MA-CA:

```
curl -k -iX POST
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp-ma.example.com:9090/mapi/services/replication/links/
MA-CA/content/tenants/Finance?pause"
```

tenant (data type for tenant resource)

The tenant data type covered in this section describes the tenants resource.

When you create an HCP tenant, you need to specify query parameters that define the initial user account for the tenant. When you create the default tenant, you need to specify query parameters that further define the default namespace. For information on these query parameters, see [“Query parameters for creating tenants”](#) on page 263.

tenant (data type for tenant resource)

tenant data type properties

The table below describes the properties included in the tenant data type that describes the tenants resource.

Property name	Data type	Description	Notes
administrationAllowed	Boolean	<p>For an HCP tenant, specifies whether system-level users have administrative access to the tenant. Valid values are:</p> <ul style="list-style-type: none">• true — System-level users have administrative access to the tenant.• false — System-level users do not have administrative access to the tenant. <p>The default is false.</p>	<p>This property is not valid on a PUT request. It is valid on a POST request and returned by a GET request only when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>

(Continued)

Property name	Data type	Description	Notes
authenticationTypes	String	<p>For an HCP tenant, lists the types of user authentication supported by the tenant. Valid authentication types are:</p> <ul style="list-style-type: none"> • LOCAL — The tenant supports local authentication by HCP. • RADIUS — The tenant supports remote authentication by RADIUS servers. • EXTERNAL — The tenant supports remote authentication by AD. <p>These values are not case sensitive.</p> <p>If you omit this property on a PUT request, the default is LOCAL and RADIUS.</p> <p>You can specify any combination of these values except RADIUS alone.</p>	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>For an existing tenant, the list of authentication types specified in the request body replaces the list of authentication types currently supported by the tenant. A tenant must always have at least one authentication type enabled.</p> <p>In XML, each listed authentication type is the value of an element named authenticationType. In JSON, the name in the name/value pair that lists the authentication types is authenticationType.</p> <p>This property is not valid for the default tenant.</p>

tenant (data type for tenant resource)

(Continued)

Property name	Data type	Description	Notes
complianceConfiguration Enabled	Boolean	<p>For an HCP tenant, specifies whether the tenant can set the retention mode of the namespaces it owns.</p> <p>Valid values are:</p> <ul style="list-style-type: none">true — The tenant can set the retention mode.false — The tenant cannot set the retention mode.	<p>This property is required on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p> <p>In enterprise mode, privileged deletes are allowed, retention class durations can be shortened, and retention classes can be deleted. In compliance mode, these activities are not allowed.</p> <p>You can change the value of this property from false to true but not from true to false.</p>
creationTime	String	<p>Specifies the date and time at which the tenant was created, in this ISO 8601 format:</p> <p><code>yyyy-MM-ddThh:mm:ssZ</code></p> <p>Z represents the offset from UTC, in this format:</p> <p><code>(+ -)hhmm</code></p> <p>For example:</p> <p>2012-02-09T19:26:32-0500</p>	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.</p>

(Continued)

Property name	Data type	Description	Notes
dataNetwork	String	Specifies the data access network for the tenant. Valid values are any network defined in the HCP system except [hcp_backend]. The default is [hcp_system].	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is not returned by any GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>
fullyQualifiedName	String	Specifies the fully qualified hostname of the tenant.	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.

tenant (data type for tenant resource)

(Continued)

Property name	Data type	Description	Notes
hardQuota	String	<p>For an HCP tenant, specifies the total amount of space available to the tenant for allocation to its namespaces.</p> <p>Valid values are decimal numbers with up to two places after the period, followed by a space, followed by MB, GB, or TB (for example, 10.25 TB). The minimum value is 1 (one) for GB and .01 for TB.</p>	<p>This property is required on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>
id	String	<p>Specifies the system-supplied unique ID for the tenant. HCP generates this ID automatically when you create a tenant.</p>	<p>This property is not valid on a PUT or POST request. It is returned only by a verbose GET request.</p>
managementNetwork	String	<p>Specifies the management network for the tenant. Valid values are any network defined in the HCP system except [hcp_backend]. The default is [hcp_system].</p>	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is not returned by any GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>

(Continued)

Property name	Data type	Description	Notes
maxNamespacesPerUser	integer	For an HCP tenant, specifies the maximum number of namespaces that can be owned at one time by any given user. Valid values are zero through 10,000. The default is 100.	<p>This property is not valid on a PUT request. It is valid on a POST request and returned by a GET request only when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>

(Continued)

Property name	Data type	Description	Notes
name	String	<p>Specifies the name of the tenant. HCP derives the hostname for the tenant from this name. The hostname is used in URLs for access to the tenant and its namespaces.</p> <p>In English, the name you specify for a tenant must be from one through 63 characters long, can contain only alphanumeric characters and hyphens (-), and cannot start or end with a hyphen. In other languages, because the derived English hostname cannot be more than 63 characters long, the name that you specify can be limited to fewer than 63 characters.</p> <p>Tenant names cannot contain special characters other than hyphens and are not case sensitive. White space is not allowed.</p> <p>Tenant names cannot start with <i>xn--</i> (that is, the characters <i>x</i> and <i>n</i> followed by two hyphens).</p> <p>Tenant names must be unique for the HCP system. Different systems can have tenants with the same name. However, a tenant cannot be replicated to another system that already has a tenant with the same name.</p> <p>You can reuse tenant names that are not currently in use. So, for example, if you delete a tenant, you can give a new tenant the same name as the one the deleted tenant had.</p> <p>The name of the default tenant is always <i>Default</i>.</p>	<p>This property is required on a PUT request. It is valid on a POST request for an HCP tenant only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request for an HCP tenant when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not returned by any GET request for the default tenant.</p> <p>The tenant name is used in the URL for access to the tenant.</p> <p>You can change the name of an HCP tenant any time after you create the tenant, except while the CIFS or NFS protocol is enabled for any namespaces owned by the tenant. However, when you change the name, the URL for the tenant may change as well.</p>

(Continued)

Property name	Data type	Description	Notes
namespaceQuota	String	<p>For an HCP tenant, specifies the number of namespaces HCP reserves for the tenant out of the total number of namespaces the system can have. This is also maximum number of namespaces the tenant can own at any given time.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> Integers in the range 1 (one) through the current number of namespaces available for allocation. The number of available namespaces is equal to 10,000 minus the number of namespaces currently allocated to HCP tenants, minus the number of namespaces currently owned by HCP tenants without a quota, minus one for the default namespace, if it exists. If any tenants are above quota, the number of excess namespaces is also subtracted from the number of available namespaces. None. <p>The default is None.</p>	<p>This property is optional on a PUT request.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>

tenant (data type for tenant resource)

(Continued)

Property name	Data type	Description	Notes
replicationConfiguration Enabled	Boolean	<p>For an HCP tenant, specifies whether the tenant is eligible for replication.</p> <p>Valid values are:</p> <ul style="list-style-type: none">• true — The tenant is eligible for replication.• false — The tenant is not eligible for replication.	<p>The property is required on a PUT request if the HCP system supports replication. If the HCP system does not support replication, this property is optional and, if included, must have a value of false.</p> <p>This property is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant. If the HCP system supports replication, directories in the default namespace are automatically eligible for replication.</p> <p>You can change the value of this property from false to true but not from true to false.</p>

(Continued)

Property name	Data type	Description	Notes
searchConfiguration Enabled	Boolean	<p>For an HCP tenant, specifies whether the tenant can enable and disable search for the namespaces it owns. Valid values are:</p> <ul style="list-style-type: none"> true — The tenant can enable and disable search for its namespaces. false — The tenant cannot enable or disable search for its namespaces. <p>The default is false.</p>	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p> <p>You can change the value of this property from false to true but not from true to false.</p>

tenant (data type for tenant resource)

(Continued)

Property name	Data type	Description	Notes
servicePlan	String	<p>For an HCP tenant, specifies the service plan associated with the tenant.</p> <p>Valid values are names of existing service plans. The default is Default.</p>	<p>This property is valid on a PUT or POST request only when the value of the servicePlanSelectionEnabled property is false.</p> <p>This property is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by a GET request only when the request is made using a system-level user account and only when the value of the servicePlanSelection-Enabled property is false.</p> <p>This property is not valid for the default tenant. To specify a service plan for the default tenant, use the servicePlan query parameter when creating the tenant. For information on this parameter, see "Default tenant query parameters" on page 265.</p>

(Continued)

Property name	Data type	Description	Notes
servicePlanSelection Enabled	Boolean	<p>For an HCP tenant, specifies whether the tenant can select service plans for the namespaces it owns. Valid values are:</p> <ul style="list-style-type: none"> true — The tenant can select service plans for its namespaces. false — The tenant cannot select service plans for its namespaces. <p>The default is false.</p>	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p> <p>You can change the value of this property from false to true but not from true to false.</p>
snmpLoggingEnabled	Boolean	<p>Specifies whether tenant log messages are sent to the SNMP managers specified at the HCP system level. Valid values are:</p> <ul style="list-style-type: none"> true — Tenant log messages are sent to SNMP managers. false — Tenant log messages are not sent to SNMP managers. <p>The default is false.</p>	<p>This property is not valid on a PUT request.</p> <p>For HCP tenants, this property is valid on a POST request and returned by a GET request only when the request is made using a tenant-level user account.</p>

(Continued)

Property name	Data type	Description	Notes
softQuota	Integer	For an HCP tenant, specifies the percentage point at which HCP notifies the tenant that free storage space is running low. Valid values are integers in the range zero through 100.	<p>This property is required on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p>
syslogLoggingEnabled	Boolean	<p>Specifies whether tenant log messages are sent to the syslog servers specified at the HCP system level. Valid values are:</p> <ul style="list-style-type: none"> true — Tenant log messages are sent to syslog servers. false — Tenant log messages are not sent to syslog servers. <p>The default is false.</p>	<p>This property is not valid on a PUT request.</p> <p>For HCP tenants, this property is valid on a POST request and returned by a GET request only when the request is made using a tenant-level user account.</p>
systemVisibleDescription	String	<p>Specifies the system-level description of the tenant. This description is optional. The default is no description.</p> <p>To remove a system-level description from an existing tenant, specify the systemVisibleDescription property with no value.</p> <p>For the rules for descriptions, see “Common property values” on page 104.</p>	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is not returned by any GET request when the request is made using a tenant-level user account.</p>

(Continued)

Property name	Data type	Description	Notes
tags	List	<p>Associates zero, one, or more tags with the tenant. Each tag can be from one through 64 characters long and can contain any valid UTF-8 characters except commas (,). White space is allowed.</p> <p>Tags are not case sensitive.</p>	<p>This property is optional on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is not returned by any GET request when the request is made using a tenant-level user account.</p> <p>For an existing tenant, the set of tags specified in the request body replaces the set of tags currently associated with the namespace. To remove all tags, specify an empty set.</p> <p>In XML, each listed tag is the value of an element named tag. In JSON, the name in the name/value pair that lists the tags is tag.</p> <p>This property is not valid for the default namespace.</p>
tenantVisibleDescription	String	<p>Specifies the tenant-level description of the tenant. This description is optional. The default is no description.</p> <p>To remove a tenant-level description from an existing tenant, specify the tenantVisibleDescription property with no value.</p> <p>For the rules for descriptions, see “Common property values” on page 104.</p>	<p>To retrieve or set the value of this property, you need a tenant-level user account.</p>

tenant (data type for tenant resource)

(Continued)

Property name	Data type	Description	Notes
versioningConfiguration Enabled	Boolean	<p>For an HCP tenant, specifies whether the namespaces owned by the tenant can have versioning enabled. Valid values are:</p> <ul style="list-style-type: none">true — Namespaces owned by the tenant can have versioning enabled.false — Namespaces owned by the tenant cannot have versioning enabled.	<p>This property is required on a PUT request. It is valid on a POST request only when the request is made using a system-level user account.</p> <p>This property is returned by any GET request when the request is made using a system-level user account. It is returned only by a verbose GET request when the request is made using a tenant-level user account.</p> <p>This property is not valid for the default tenant.</p> <p>You can change the value of this property from false to true but not from true to false.</p>

tenant data type example

Here's an XML example of the tenant data type that describes the tenants resource; the properties shown are those that are returned in response to a verbose **GET** request for an HCP tenant where the request is made using a system-level user account and the tenant has granted system-level users administrative access to itself:

```
<tenant>
  <administrationAllowed>true</administrationAllowed>
  <authenticationTypes>
    <authenticationType>LOCAL</authenticationType>
    <authenticationType>EXTERNAL</authenticationType>
  </authenticationTypes>
  <complianceConfigurationEnabled>true</complianceConfigurationEnabled>
  <creationTime>2012-02-09T09:11:17-0500</creationTime>
  <dataNetwork>net127</dataNetwork>
  <fullyQualifiedName>finance.hcp.example.com</fullyQualifiedName>
  <hardQuota>100 GB</hardQuota>
  <managementNetwork>net004</managementNetwork>
  <maxNamespacesPerUser>5</maxNamespacesPerUser>
  <name>Finance</name>
  <namespaceQuota>5</namespaceQuota>
```

```

<replicationConfigurationEnabled>true</replicationConfigurationEnabled>
<snmpLoggingEnabled>>false</snmpLoggingEnabled>
<searchConfigurationEnabled>true</searchConfigurationEnabled>
<servicePlan>Short-Term-Activity</servicePlan>
<servicePlanSelectionEnabled>>false</servicePlanSelectionEnabled>
<softQuota>90</softQuota>
<syslogLoggingEnabled>true</syslogLoggingEnabled>
<systemVisibleDescription>Created for the Finance department at Example
  Company by P.D. Grey on 2/9/2012.</systemVisibleDescription>
<tags>
  <tag>Example Company</tag>
  <tag>pdgrey</tag>
</tags>
<tenantVisibleDescription>Please see Lee Green for any questions about this
  tenant and its namespaces.</tenantVisibleDescription>
<id>4420f62f-3f63-43ab-a3cd-0bcf1f399daf</id>
<versioningConfigurationEnabled>true</versioningConfigurationEnabled>
</tenant>

```

Query parameters for creating tenants

When you create a tenant, you need to specify query parameters on the resource URL. These parameters differ for HCP tenants and the default tenant.

HCP tenant query parameters

When you create an HCP tenant, HCP automatically creates the initial user or group account for the tenant, depending on which query parameters you include in the **PUT** request.

Creating an initial user account

To create a tenant with an initial user account, you use these query parameters, which correspond to user account properties with the same name:

- **username** — This parameter is required when you create a tenant. The username you specify is also used as the full name for the user account.
- **password** — This parameter is required when you create a tenant.
- **forcePasswordChange** — This parameter is optional when you create a tenant. The default is **false**.

For information on values for these parameters, see [“userAccount data type properties”](#) on page 269.

tenant (data type for tenant resource)

The user account that's created:

- Is enabled
- Is locally authenticated
- Has only the security role
- Has no data access permissions
- Has no description

The **username**, **password**, and **forcePasswordChange** query parameters are valid only when you create an HCP tenant and only if you enable local authentication for the tenant in the same request. They are not valid on a request to modify a tenant.

For an example of a request that uses these query parameters, see ["Example: Creating an HCP tenant"](#) on page 39.

Creating an initial group account

To create the tenant with an initial group account, you use the **initialSecurityGroup** query parameter. The value of this parameter must be the name or SID of an AD group defined in the AD forest supported by HCP. You can specify the name in either of these formats:

group-name

group-name@ad-domain-name

If you omit the domain name, HCP uses the AD domain specified in the system configuration.

Be sure to use the second format if a group with the specified name exists in more than one domain in the AD forest or if the group name looks like a SID.

The group account that's created:

- Has only the security role
- Has no data access permissions

The **initialSecurityGroup** query parameter is valid only when you create an HCP tenant and only if you enable AD authentication for the tenant in the same request. It is not valid on a request to modify a tenant.

Default tenant query parameters

When you create the default tenant, HCP automatically creates the default namespace. To provide information about this namespace, you use these query parameters, which correspond to namespace properties with the same name:

- **enterpriseMode** — This parameter is required when you create the default tenant.
- **hashScheme** — This parameter is required when you create the default tenant.
- **searchEnabled** — This parameter is optional when you create the default tenant. The default is **false**. (By default, if you specify **searchEnabled=true**, search indexing is enabled. Otherwise, search indexing is disabled by default.)
- **servicePlan** — This parameter is optional when you create the default tenant. The default is **Default**.

For information on values for these parameters, see ["namespace data type properties"](#) on page 179.

These query parameters are valid only when you create the default tenant. They are not valid on requests to modify the tenant.

transition

The transition data type describes the transition property of the local and remote data types used to describe the local and remote properties of the replication link schedule resource.

transition data type properties

The table below describes the properties included in the transition data type.

transition

Property name	Data type	Description	Notes
performanceLevel	String	<p>Specifies the performance level that applies to the replication link on the local or remote system, as applicable, starting at the day and time specified by the time property. Valid values are:</p> <ul style="list-style-type: none"> • LOW — The performance level changes to low at the specified day and time. • MEDIUM — The performance level changes to medium at the specified day and time. • HIGH — The performance level changes to high at the specified day and time. • CUSTOM — The performance level changes to the custom setting at the specified day and time. For information on setting the custom performance level, see <i>Replicating Tenants and Namespaces</i>. • OFF — The local or remote system, as applicable, stops sending data on the link at the specified day and time. <p>These values are not case sensitive.</p> <p>You cannot set the schedule to OFF for the entire week. To stop replication on the link for the entire week, suspend the link.</p>	This property is required for within occurrence of the transition property.

(Continued)

Property name	Data type	Description	Notes
time	Date	<p>Specifies a day and time at which the performance level for the replication link changes to the level specified by the performanceLevel property. Valid values are datetime values in this format:</p> <p><i>EE:hh</i></p> <p>In this format:</p> <ul style="list-style-type: none"> <i>EE</i> is the three-letter day (for example, Mon). This value is case sensitive. <i>hh</i> is the hour on a 24-hour clock. For midnight, use 00. 	This property is required for within occurrence of the transition property.

transition data type example

Here's an XML example of the transition data type:

```
<transition>
  <time>Tue:18</time>
  <performanceLevel>HIGH</performanceLevel>
</transition>
```

userAccount

The userAccount data type describes the userAccounts resource.

When you create a user account, you use a query parameter to specify the password for the account. You use the same query parameter to change the password for a user account. For information on this query parameter, see ["Query parameter for setting user account passwords"](#) on page 273.

userAccount data type properties

The table below describes the properties included in the userAccount data type.

Property name	Data type	Description	Notes
allowNamespaceManagement	Boolean	<p>Specifies whether the user account has the allow namespace management property. Valid values are:</p> <ul style="list-style-type: none"> true — The user account has the allow namespace property. false — The user account does not have the allow namespace management property. <p>On a PUT request, the default is true if the roles property includes ADMINISTRATOR in the same request; otherwise, the default is false.</p> <p>On a POST request, adding ADMINISTRATOR to the roles for the user account automatically enables the allow namespace management property for the account.</p> <p>Users with the allow namespace management property can use the HCP management and HS3 APIs to:</p> <ul style="list-style-type: none"> Create namespaces List, view and change the versioning status of, and delete namespaces they own 	<p>This property is not valid on a PUT request. It is valid on a POST request only if the user making the request has the administrator role.</p>


(Continued)

Property name	Data type	Description	Notes
description	String	<p>Specifies the description of the user account. This description is optional. The default is no description.</p> <p>To remove a description from an existing user account, specify the description property with no value.</p> <p>For the rules for descriptions, see “Common property values” on page 104.</p>	This property is valid on a POST request only if the user making the request has the security role.
enabled	Boolean	<p>Specifies whether the user account is enabled. Valid values are:</p> <ul style="list-style-type: none"> true — The user account is enabled. false — The user account is disabled. 	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.
forcePasswordChange	Boolean	<p>Specifies whether the password for the user account must be changed the next time the account is used to log into the Tenant Management Console. Valid values are:</p> <ul style="list-style-type: none"> true — The password must be changed. false — The password does not need to be changed. 	This property is required on a PUT request. It is valid on a POST request and returned by a GET request only if the user making the request has the security role.
fullName	String	Specifies the full name of the user for whom you’re creating the account. This name must be from one through 64 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.

(Continued)

Property name	Data type	Description	Notes
localAuthentication	Boolean	<p>Specifies whether the user account is authenticated locally or by a RADIUS server specified at the HCP system level. Valid values are:</p> <ul style="list-style-type: none"> true — The user account is authenticated locally. false — The user account is authenticated by a RADIUS server. 	This property is required on a PUT request. It is not valid on a POST request and is returned only by a verbose GET request.
roles	List	<p>Associates zero, one, or more roles with the user account. Valid values for roles are:</p> <ul style="list-style-type: none"> ADMINISTRATOR COMPLIANCE MONITOR SECURITY <p>These values are not case sensitive.</p> <p>The default is no roles.</p>	<p>This property is valid on a POST request and returned by a GET request only when the user making the request has the security role.</p> <p>For an existing user account, the set of roles specified in the request body replaces the set of roles currently associated with the user account. To remove all roles, specify an empty set.</p> <p>In XML, each listed role is the value of an element named role. In JSON, the name in the name/value pair that lists the roles is role.</p>
userGUID	String	Specifies the system-supplied globally unique user ID for the user account. HCP generates this ID automatically when you create an account.	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request and only when the user making the request has the security role.

(Continued)

Property name	Data type	Description	Notes
userID	Integer	<p>Specifies the system-supplied local user ID for the user account. HCP generates this ID automatically when you create an account.</p> <p>Local user IDs are unique within an HCP system. These IDs are maintained for backward compatibility but are no longer used for user identification.</p>	This property is not valid on a PUT or POST request. It is returned only by a verbose GET request and only when the user making the request has the security role.
username	String	<p>Specifies the username for the user account. Usernames must be from one through 64 characters long and can contain any valid UTF-8 characters, including white space, but cannot start with an opening square bracket ([).</p> <p>Usernames are not case sensitive.</p> <p>The username for a user account must be unique for the tenant. Different tenants can have user accounts with the same username.</p> <p>You can reuse usernames that are not currently in use. So, for example, if you delete the account for a user and then create a new account for that user, you can give the user the same username as before.</p> <p> Tip: Consider using email addresses as usernames. This enables users to more easily remember their HCP usernames. It also gives you easy access to email addresses should you need to contact any users.</p>	This property is required on a PUT request. It is valid on a POST request only when the user making the request has the security role.

userAccount data type example

Here's an XML example of the userAccount data type:

```
<userAccount>
  <allowNamespaceManagement>false</allowNamespaceManagement>
  <description>Compliance officer.</description>
  <enabled>true</enabled>
  <forcePasswordChange>true</forcePasswordChange>
  <fullName>Morgan White</fullName>
  <localAuthentication>true</localAuthentication>
  <roles>
    <role>MONITOR</role>
    <role>COMPLIANCE</role>
  </roles>
  <userGUID>a8ae69dc-e2e3-44a9-aa64-9c142a38ed5d</userGUID>
  <userID>517</userID>
  <username>mwhite</username>
</userAccount>
```

Query parameter for setting user account passwords

You use the **password** query parameter to specify the password for a new user account and to change the password for an existing user account. The value of this parameter is the password you want. For the rules for passwords, see ["Passwords"](#) on page 104.

For an example of a request that uses this query parameter, see ["Example: Creating a user account"](#) on page 72.

Query parameter for resetting security user passwords

A tenant must at all times have at least one user that can manage user and group accounts. This means that the tenant must have at least one user or group account with the security role. If the tenant does not have such a group account and the passwords for all user accounts with the security role have been lost, the tenant is in a state in which no users can manage user and group accounts.

To recover from this state, you can use the **resetPasswords** query parameter to change the passwords of all user accounts with the security role to a password that you specify. The value of this parameter is the password you want. For the rules for passwords, see ["Passwords"](#) on page 104.

To reset security user passwords for a tenant, you need a system-level user account with the security role.

You use the **resetPasswords** query parameter with a **POST** request against the userAccounts resource, as in this example:

```
curl -k -i -d "<userAccounts/>" -H "Content-Type: application/xml"
-H "Authorization: HCP YWxscm9sZXM=:04EC9F614D89FF5C7126D32ACB448382"
"https://admin.hcp.example.com:9090/mapi/tenants/finance/userAccounts
?resetPasswords=start123"
```

As an alternative to resetting user account passwords, you can reset a security group for the tenant,. For information on doing this, see ["Query parameter for resetting the security group"](#) on page 153.

versioningSettings

The versioningSettings data type describes the versioningSettings resource for namespaces and the versioningSettings property of the namespace and namespaceDefaults data types.

versioningSettings data type properties

The table below describes the properties included in the versioningSettings data type.

Property name	Data type	Description	Notes
daysOnPrimary	Integer	Specifies the number of days old versions of objects must remain in the namespace before they are pruned. Valid values are integers in the range zero through 36,500 (that is, 100 years). A value of zero means prune immediately. The default is zero.	Deprecated; replaced by the pruneDays property. If specified on a PUT or POST request, this property has the same function as the pruneDays property. You cannot include both this property and the pruneDays property in the same request. This property is not returned by any GET request.
daysOnReplica	Integer		Deprecated. The pruneDays property applies to the namespace on all systems on which the namespace exists. This property is ignored on a PUT or POST request and is not returned by any GET request.

(Continued)

Property name	Data type	Description	Notes
enabled	Boolean	<p>Specifies whether versioning is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Versioning is enabled. false — Versioning is disabled. <p>The default is false.</p>	<p>This property is required on a PUT request to create a namespace and on a POST request to modify namespace defaults if the request includes the versioningSettings property.</p> <p>You cannot enable versioning for a namespace while the CIFS, NFS, WebDAV, or SMTP protocol or appendable objects are enabled.</p>
keepDeletionRecords	Boolean	<p>Specifies whether HCP should keep records of deletion operations (delete, purge, prune, disposition) that occur in the namespace if the namespace has ever had versioning enabled. Valid values are:</p> <ul style="list-style-type: none"> true — Keep records of deletion operations. false — Do not keep records of deletion operations. <p>The default is true.</p> <p>The amount of time for which HCP keeps deletion records is determined by the system configuration.</p>	<p>This property is not valid on a POST request to modify namespace defaults and is not returned by any GET request for namespace defaults.</p>

(Continued)

Property name	Data type	Description	Notes
prune	Boolean	<p>Specifies whether version pruning is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Version pruning is enabled for the namespace. false — Version pruning is disabled for the namespace. <p>The default is false.</p>	<p>This property is required on a PUT request to create a namespace and on a POST request to modify namespace defaults if the enabled property is set to true.</p> <p>You cannot include both this property and the <code>pruneOnPrimary</code> property in the same request.</p>
pruneDays	Integer	<p>Specifies the number of days old versions of objects must remain in the namespace before they are pruned. Valid values are integers in the range zero through 36,500 (that is, 100 years). A value of zero means prune immediately.</p> <p>The default is zero.</p>	<p>This property is required on a PUT request to create a namespace and on a POST request to modify namespace defaults if the <code>prune</code> property is set to true.</p> <p>You cannot include both this property and the <code>daysOnPrimary</code> property in the same request.</p>
pruneOnPrimary	Boolean	<p>Specifies whether version pruning is enabled for the namespace. Valid values are:</p> <ul style="list-style-type: none"> true — Version pruning is enabled for the namespace. false — Version pruning is disabled for the namespace. <p>The default is false.</p>	<p>Deprecated; replaced by the <code>prune</code> property.</p> <p>If specified on a PUT or POST request, this property has the same function as the <code>prune</code> property. You cannot include both this property and the <code>prune</code> property in the same request.</p> <p>This property is not returned by any GET request.</p>
pruneOnReplica	Boolean		<p>Deprecated. The <code>prune</code> property applies to the namespace on all systems on which the namespace exists.</p> <p>This property is ignored on a PUT or POST request and is not returned by any GET request.</p>

versioningSettings data type example

Here's an XML example of the versioningSettings data type:

```
<versioningSettings>
  <enabled>true</enabled>
  <keepDeletionRecords>true</keepDeletionRecords>
  <prune>true</prune>
  <pruneDays>10</pruneDays>
</versioningSettings>
```

Specifying retention values

These properties require you to specify a retention value:

- The retentionDefault property of the complianceSettings data type. The value of this property can be a special value, offset, retention class, or fixed date.
- The value property of the retentionClass data type. The value of this property can be a special value or an offset.

Specifying a special value

To specify a special value as a retention value, use one of these:

- **0 (zero) or Deletion Allowed** — Allows the object to be deleted at any time.
- **-1 or Deletion Prohibited** — Prevents the object from ever being deleted by means of a normal delete operation. If the namespace is in enterprise mode, however, the object can be deleted by means of a privileged delete operation.
- **-2 or Initial Unspecified** — Prevents the object from being deleted but allows its retention setting to be changed.

These values are not case sensitive.

Specifying an offset

To specify an offset as a retention value, use a standard expression that conforms to this syntax:

```
^A([+-]\d+y)?([+-]\d+M)?([+-]\d+w)?([+-]\d+d)?([+-]\d+h)?([+-]\d+m)?([+-]\d+s)?$
```

The table below explains this syntax.

Character	Description
^	Start of the expression
\$	End of the expression
()	Sequence of terms treated as a single term
?	Indicator that the preceding term is optional
[]	Group of alternatives, exactly one of which must be used
+	Plus
-	Minus
A	The time at which the object was added to the repository
\d+	An integer in the range 0 (zero) through 9,999
y	Years
M	Months
w	Weeks
d	Days
h	Hours
m	Minutes
s	Seconds

The time measurements included in an expression must go from the largest unit to the smallest (that is, in the order in which they appear in the syntax). These measurements are case sensitive. You can omit measurements that have value of zero.

Here are some offset examples:

A+100y

A+20d-5h

A+2y+1d

Specifying a retention class

To specify a retention class as a retention value, use this format:

C+*retention-class-name*

For example, if the name of the retention class is HlthReg-107, specify the property value as:

```
C+HlthReg-107
```

The retention class you specify must already exist.

Specifying a fixed date

To specify a fixed date as a retention value, you can use either of these formats:

- Time in seconds since January 1, 1970, at 00:00:00. For example:

```
1451520000
```

The calendar date that corresponds to 1451520000 is Thursday, December 31, 2015, at 00:00:00 EST.

- Date and time in this ISO 8601 format:

```
yyyy-MM-ddThh:mm:ssZ
```

z represents the offset from UTC and is specified as:

```
(+|-)hhmm
```

For example: 2015-12-31T00:00:00-0500

If you specify certain invalid dates, HCP automatically adjusts the value to make a real date. For example, if you specify a default retention setting of 2015-11-33T00:00:00-0500, which is three days past the end of November, objects added to the namespace get a retention setting of 2015-12-03T00:00:00-0500.

Usage considerations

This chapter contains usage considerations for the HCP management API.

Choosing an access method

You can access the HCP system through the management API by specifying either a hostname or an IP address in the resource URL. If the system uses DNS and you specify a hostname, HCP selects the IP address for you from the currently available nodes. HCP uses a round-robin method to ensure that it doesn't always select the same address.

When you specify IP addresses, your application must take responsibility for balancing the load among the nodes. In this case, you risk trying to connect (or reconnect) to a node that is not available. However, using explicit IP addresses to connect to specific nodes can sometimes have advantages over using hostnames.

These considerations apply when deciding which method to use:

- If the client uses a `hosts` file to map HCP hostnames to IP addresses, the client system has full responsibility for converting any hostnames to IP addresses. Therefore, HCP cannot spread the load or prevent attempts to connect to unavailable nodes.

For more information on using a `hosts` file, see ["Using a hosts file"](#) on page 24.

- If the client caches DNS information, connecting by hostname may result in the same node being used repeatedly.
- When you access the HCP system by hostname, HCP ensures that requests are distributed among the nodes, but it does not ensure that the resulting loads on the nodes are evenly balanced.
- When multiple applications access the HCP system by hostname concurrently, HCP is less likely to spread the load evenly across the nodes than with a single application.



Note: When using hostnames, you can ping the HCP system periodically to check whether you're getting connections to different nodes.

For more information on using an IP address to access the HCP system, see ["Using an IP address in a URL"](#) on page 23.

Generating templates for resource creation

When you use the HCP management API to create a resource, the best way to ensure that the request body includes the appropriate properties is to use a template. You can generate your own template by submitting a **GET** request for an existing resource of the same type. In the request, include the **verbose=false** query parameter on the resource URL (or omit the **verbose** parameter to accept the default of **false**).

With a nonverbose **GET** request HCP returns only properties whose values you can set. This enables you to use the response body as a template for the request body for creating additional resources of the same type.

In most cases, in response to a nonverbose **GET** request, HCP returns all the properties required for creating a resource of the same type and none of the properties that are invalid on a **PUT** request. The only exception is for namespaces, where the response body does not include the `versioningSettings` property. To complete the template for namespaces, you need to add this property.

To generate the template for creating tenants, you need to make the **GET** request using a system-level user account. The tenant you specify in the **GET** request must *not* have granted system-level users administrative access to itself.

Modifying resources

The best way to modify a resource through the HCP management API is to:

1. Submit a **GET** request for the resource you want to modify. In the request, include the **verbose=false** query parameter on the resource URL (or omit the **verbose** parameter to accept the default of **false**). This ensures that the response body includes only the properties whose values you can set.
2. In the response body from the **GET** request, change property values as needed.
3. Submit the **POST** request for the resource, using the entire modified response body as the request body.

By modifying a resource in this way, you ensure that all the property values are set as expected. A possible drawback to this method is that someone else can modify the resource in between your **GET** and **POST** requests. In this case, the values in your **POST** request will overwrite any modifications. However, this may, in fact, be the result you want.



HTTP return codes

This appendix contains a table that describes the possible return codes for HCP management API requests.

Code	Meaning	Methods	Description
200	OK	All	HCP successfully created, retrieved, checked the existence of, modified, or deleted a resource.
302	Found	HEAD	The resource identified by the URL exists, but the user account identified by the Authorization header or hcp-api-auth cookie doesn't have permission to access it.
400	Bad Request	All	<p>The request was not valid. These are some, but not all, of the possible reasons:</p> <ul style="list-style-type: none"> • The URL in the request is not well-formed. • The request is missing a required query parameter. • The request contains a required or optional non-Boolean query parameter with an invalid value. • For a PUT or POST request, the request body: <ul style="list-style-type: none"> ◦ Is missing a required property ◦ Includes a property that is invalid for the resource ◦ Has a property with an invalid value ◦ Contains XML or JSON that is not well-formed • The Content-Type or Accept header in the request specifies a nonexistent Internet media type. <p>If more information about the error is available, the response headers include the HCP-specific X-HCP-ErrorMessage header.</p>
401	Unauthorized	All	HCP was unable to handle the request. If this happens repeatedly, contact your authorized HCP service provider for help.

(Continued)

Code	Meaning	Methods	Description
403	Forbidden	All	<p>The requested operation is not allowed. These are some, but not all, of the possible reasons:</p> <ul style="list-style-type: none">• The URL in the request is missing the port number (9090).• The request does not include an Authorization header or hcp-ns-auth cookie.• The Authorization header or hcp-ns-auth cookie specifies invalid credentials.• The user account identified by the Authorization header or hcp-api-auth cookie is either a system-level account where a tenant-level account is required or a tenant-level account where a system-level account is required.• The user account identified by the Authorization header or hcp-api-auth cookie 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:<ul style="list-style-type: none">◦ For a tenant, HCP could not delete the tenant because it owns one or more namespaces.◦ For a namespace, HCP could not delete the namespace because it contains one or more objects.◦ For a user account, HCP could not delete the user account because it is the last locally authenticated, enabled user account with the security role, and no group accounts have the security role.• For a tenant-level request, the HCP management API is not enabled for the tenant. Use the Tenant Management Console for the applicable tenant to enable the API. <p>If more information about the error is available, the response headers include the HCP-specific X-HCP-ErrorMessage header.</p>

(Continued)

Code	Meaning	Methods	Description
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.
409	Conflict	PUT	<p>For a PUT request, HCP could not create the resource because it already exists.</p> <p>If more information about the error is available, the response headers include the HCP-specific X-HCP-ErrorMessage header.</p>
414	Request URI Too Large	All	The portion of the URL following <code>rest</code> is longer than 4,095 bytes.
415	Unsupported Media Type	All	The management API does not support the Internet media type specified by the Content-Type or Accept header.
500	Internal Server Error	All	An internal error occurred. If this happens repeatedly, contact your authorized HCP service provider for help.
503	Service Unavailable	All	<p>HCP is temporarily unable to handle the request. Possible reasons include:</p> <ul style="list-style-type: none">• HCP is currently unavailable due to system overload, maintenance, or upgrade. Try the request again in a little while.• The HCP system is currently being upgraded. <p>If more information about the error is available, the response headers include the HCP-specific X-HCP-ErrorMessage header.</p>

Sample Java application

This appendix contains a sample Java[®] application that uses the HCP management API to define various resources in an HCP system. It also shows the content of the JSON files that serve as input to the application.

What the application does

The sample Java application shown in this appendix uses the HCP management API to:

1. Create one HCP tenant
2. Give the initial tenant-level user account the administrator role
3. Create a tenant-level user account with the monitor and compliance roles
4. Configure the Tenant Management Console
5. Enable syslog logging for the tenant and system-level administrative access to the tenant
6. Create two namespaces
7. Enable disposition for one of the namespaces
8. Configure the HTTP protocol for both namespaces
9. Create a tenant-level user account with no roles
10. Grant data access permissions to the user account with no roles
11. Create one retention class
12. Create a replication link
13. Add a tenant to the replication link

Assumptions

The sample Java application assumes:

- That the system against which you're running the application is named `hcp-ma.example.com`
- The existence of a system-level HCP user account with the username *rsilver* and the password *p4ssw0rd*
- That all input files are located in the `/home/rsilver/MAPI` directory

- The existence of a service plan named Short-Term-Activity
- That HCP is configured to support Active Directory
- That the HCP system includes the replication feature
- The existence of another HCP system named hcp-ca.example.com
- That the system against which you're running the application and the system named hcp-ca.example.com have exchanged replication SSL server certificates with each other

If you want to run the sample application, you may need to modify the sample input JSON files to conform to your HCP system.

Required libraries

To run the sample application presented in this appendix, you need to have these Java libraries installed:

- HttpClient 4.0 (httpclient-4.0.jar)
- HttpCore 4.0 (httpcore-4.0.1.jar)
- Commons Logging 1.1 (commons-logging-1.1.jar)

You can download the first two of these libraries from:

<http://hc.apache.org/downloads.cgi>

You can download the third one from:

http://commons.apache.org/logging/download_logging.cgi

Input JSON files

The sample Java application uses the JSON files shown in the following sections.

JSON file for creating the HCP tenant

Here's the content of the JSON file that creates an HCP tenant named Finance. The name of the file is `FinanceTenant.json`. For the username and password of the initial user account for this tenant, the sample application specifies *lgreen* and *start123*, respectively.

```
{
  "name" : "Finance",
  "systemVisibleDescription" : "Created for the Finance department at Example
Company by Robin Silver",
  "hardQuota" : "100.00 GB",
  "softQuota" : 90,
  "namespaceQuota" : "5",
  "authenticationTypes" : {
    "authenticationType" : ["EXTERNAL", "LOCAL"]
  },
  "complianceConfigurationEnabled" : true,
  "versioningConfigurationEnabled" : true,
  "searchConfigurationEnabled" : true,
  "replicationConfigurationEnabled" : true,
  "tags" : {
    "tag" : [ "Example Company", "pdgrey" ]
  },
  "servicePlanSelectionEnabled" : false,
  "servicePlan" : "Short-Term-Activity"
}
```

JSON file for modifying the initial user account

Here's the content of the JSON file that adds the administrator role to the *lgreen* user account. The name of the file is `lgreen-UAroles.json`.

```
{
  "roles" : {
    "role" : [ "SECURITY", "ADMINISTRATOR" ]
  }
}
```

JSON file for creating the user account with the compliance and monitor roles

Here's the content of the JSON file that creates a user account with the monitor and compliance roles. The name of the file is `mwhite-UA.json`. For the username and password for this account, the sample application specifies *mwhite* and *start123*, respectively.

```
{
  "username" : "mwhite",
  "fullName" : "Morgan White",
  "description" : "Compliance officer.",
  "localAuthentication" : true,
  "forcePasswordChange" : false,
  "enabled" : true,
  "roles" : {
    "role" : [ "COMPLIANCE", "MONITOR" ]
  }
}
```

JSON file for configuring the Tenant Management Console

Here's the content of the JSON file that configures the Tenant Management Console for the Finance tenant. The name of the file is `FinanceMgmtConsole.json`.

```
{
  "ipSettings" : {
    "allowAddresses" : {
      "ipAddress" : [ "192.168.103.18", "192.168.103.24", "192.168.103.25" ]
    },
    "denyAddresses" : {
      "ipAddress" : [ ]
    },
    "allowIfInBothLists" : false
  },
  "minimumPasswordLength" : 6,
  "forcePasswordChangeDays" : 45,
  "disableAfterAttempts" : 3,
  "disableAfterInactiveDays" : 30,
  "logoutOnInactive" : 20,
  "loginMessage" : "Welcome to the Finance Tenant"
}
```

JSON file for modifying the tenant

Here's the content of the JSON file that enables syslog logging for the Finance tenant and also enables system-level administrative access to the tenant. The name of the file is `modFinance.json`.

```
{
  "administrationAllowed" : true,
  "maxNamespacesPerUser" : 5,
  "sysloggingEnabled" : true
}
```

JSON files for creating the namespaces

Here's the content of the JSON file that creates a namespace named Accounts-Receiveable. The name of the file is `AccountsRecNS.json`.

```
{
  "name" : "Accounts-Receiveable",
  "description" : "Created for the Finance department at Example Company by Lee Green",
  "hashScheme" : "SHA-256",
  "enterpriseMode" : true,
  "hardQuota" : "50.00 GB",
  "softQuota" : 75,
  "versioningSettings" : {
    "enabled" : true,
    "prune" : true,
    "pruneDays" : 10,
  },
  "aclsUsage" : "ENABLED",
  "searchEnabled" : true,
  "indexingEnabled" : true,
  "customMetadataIndexingEnabled" : true,
  "replicationEnabled" : true,
  "readFromReplica" : true,
  "serviceRemoteSystemRequests" : true,
  "tags" : {
    "tag" : [ "Billing", "lgreen" ]
  }
}
```

Here's the content of the JSON file that creates a namespace named Accounts-Payable. The name of the file is `AccountsPayNS.json`.

```
{
  "name" : "Accounts-Payable",
  "description" : "Created for the Finance department at Example Company by Lee Green",
  "hashScheme" : "SHA-256",
  "enterpriseMode" : true,
  "hardQuota" : "50.00 GB",
  "softQuota" : 75,
  "versioningSettings" : {
    "enabled" : true,
    "prune" : true,
    "pruneDays" : 10,
  },
  "aclsUsage" : "ENABLED",
  "searchEnabled" : true,
  "indexingEnabled" : true,
  "customMetadataIndexingEnabled" : true,
  "replicationEnabled" : true,
  "readFromReplica" : true,
  "serviceRemoteSystemRequests" : true,
  "tags" : {
    "tag" : [ "Billing", "lgreen" ]
  }
}
```

JSON for modifying a namespace

Here's the content of the JSON file that enables disposition for the Accounts-Receiveable namespace. The name of the file is `AcctsRecCompliance.json`.

```
{
  "dispositionEnabled" : true
}
```

JSON file for configuring the HTTP protocol

Here's the content of the JSON file that is used to configure the HTTP protocol for both namespaces. The name of the file is `http.json`.

```
{
  "httpsEnabled" : true,
  "httpEnabled" : false,
  "restEnabled" : "true",
  "restRequiresAuthentication" : true,
  "httpActiveDirectorySSOEnabled" : true,
  "ipSettings" : {
    "allowAddresses" : {
      "ipAddress" : [ "192.168.140.10", "192.168.140.15", "192.168.149.0/24" ]
    },
    "denyAddresses" : {
      "ipAddress" : [ "192.168.149.5" ]
    },
    "allowIfInBothLists" : false
  }
}
```

JSON file for creating the user account with no roles

Here's the content of the JSON file that creates a user account with the username *pblack*. The name of the file is `pblack-UA.json`. For the password for this account, the sample application specifies *start123*.

```
{
  "username" : "pblack",
  "fullName" : "Paris Black",
  "description" : "Data access user.",
  "localAuthentication" : true,
  "forcePasswordChange" : false,
  "enabled" : true
}
```


JSON file for granting data access permissions to the user account

Here's the content of the JSON file that grants data access permissions to the user account with the username *pblack*. The name of the file is `pblack-UAperms.json`.

```
{
  "namespacePermission" : [ {
    "namespaceName" : "Accounts-Receivable",
    "permissions" : {
      "permission" : [ "BROWSE", "READ", "SEARCH", "PURGE", "DELETE", "WRITE" ]
    }
  }, {
    "namespaceName" : "Accounts-Payable",
    "permissions" : {
      "permission" : [ "BROWSE", "READ" ]
    }
  }
]
```

JSON file for creating the retention class

Here's the content of the JSON file that creates a retention class named `Fn-Std-42` for the Accounts-Receivable namespace. The name of the file is `RC-FN-Std-42.json`.

```
{
  "name" : "FN-Std-42",
  "description" : "Implements Finance department standard #42 - keep for 10 years.",
  "value" : "A+10y",
  "allowDisposition" : true
}
```

JSON file for creating the replication link

Here's the content of the JSON file that creates a replication link named MA-CA. The name of the file is `LinkMA-CA.json`.

```
{
  "name" : "MA-CA",
  "description" : "Active/active link between systems in MA and CA",
  "type" : "ACTIVE_ACTIVE",
  "connection" : {
    "remoteHost" : "replication.admin.hcp-ca.example.com"
  },
  "compression" : false,
  "encryption" : false,
  "priority" : "OLDEST_FIRST",
  "failoverSettings" : {
    "remote" : {
      "autoFailover" : true,
      "autoFailoverMinutes" : 60
    },
    "local" : {
      "autoFailover" : true,
      "autoFailoverMinutes" : 60
    }
  }
}
```

Java application

Here is the sample Java application that uses the JSON files shown in the preceding sections.

```
import sun.misc.BASE64Encoder;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.KeyManagementException;
import java.io.*;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.List;
import java.util.ArrayList;

import org.apache.http.conn.scheme.SchemeRegistry;
import org.apache.http.conn.scheme.Scheme;
import org.apache.http.conn.ssl.SSLSocketFactory;
import org.apache.http.conn.ClientConnectionManager;
import org.apache.http.conn.params.ConnPerRouteBean;
import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.params.HttpParams;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.impl.conn.tsccm.ThreadSafeClientConnManager;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.impl.client.AbstractHttpClient;
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.HttpRequest;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.entity.FileEntity;
import org.apache.http.client.utils.URIUtils;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.client.methods.HttpPut;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.util.EntityUtils;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;
import java.security.cert.X509Certificate;
import java.security.cert.CertificateException;
import java.security.cert.KeyStore;
import java.security.cert.KeyStoreException;

/**
 * HCP Management API - Sample Java Application
 */
public class MAPISample {
```

```

private AbstractHttpClient httpClient = null;
private String protocol;
private int port;
private String uname64;
private String encodedPassword;

private String hcpSystemAddr;

private enum RequestType {
    PUT, POST;
}

public class HCPNotInitializedException extends Exception {
    public HCPNotInitializedException(String msg) {
        super("HTTP client could not be initialized in HCPAdapter for the " +
            "following reason: " + msg);
    }
}

public static void main(String [] args) {
    MAPISample adapter = null;
    try {
        adapter = new MAPISample();

        // Set up the username and password for the system-level user account
        // in the adapter.
        String hcpSystemAddr = "hcp-ma.example.com";
        String adminUser = "rsilver";
        String adminPassword = "p4ssw0rd";
        adapter.setUpSystemInfo(hcpSystemAddr, adminUser, adminPassword);

        // Create a tenant, using FinanceTenant.json as input.
        String tenantName = "finance";
        String tenantUser = "lgreen";
        String tenantPassword = "start123";
        File f = new File("/home/rsilver/MAPI/FinanceTenant.json");
        adapter.createTenant(f, tenantUser, tenantPassword);

        // Switch the adapter to the initial user account for the new tenant.
        adapter.setUpSystemInfo(hcpSystemAddr, tenantUser, tenantPassword);

        // Modify the initial user account, using lgreen-URoles.json as input.
        f = new File("/home/rsilver/MAPI/lgreen-URoles.json");
        adapter.modifyUserAccount(tenantName, tenantUser, f);

        // Create a user account for compliance, using mwhite-UA.json as input
        // and specifying start123 as the account password.
        f = new File("/home/rsilver/MAPI/mwhite-UA.json");
        adapter.createTenantUserAccount(tenantName, "start123", f);

        // Configure the Tenant Management Console, using FinanceMgmtConsole.json
        // as input.
        f = new File("/home/rsilver/MAPI/FinanceMgmtConsole.json");
    }
}

```

```

adapter.configureTenantSecurity(tenantName, f);

// Modify the tenant, using modFinance.json as input.
f = new File("/home/rsilver/MAPI/modFinance.json");
adapter.modifyTenant(tenantName, f);

// Create a namespace, using AccountsRecNS.json as input.
f = new File("/home/rsilver/MAPI/AccountsRecNS.json");
adapter.createNamespace(tenantName, f);

// Create a namespace, using AccountsPayNS.json as input.
f = new File("/home/rsilver/MAPI/AccountsPayNS.json");
adapter.createNamespace(tenantName, f);

// Modify the Accounts-Receiveable namespace, using
// AcctsRecCompliance.json as input.
String namespaceName = "Accounts-Receiveable";
f = new File("/home/rsilver/MAPI/AcctsRecCompliance.json");
adapter.modifyNamespace(namespaceName, tenantName, f);

// Configure HTTP for the Accounts-Receiveable namespace, using http.json
// as input.
String namespaceName = "Accounts-Receiveable";
f = new File("/home/rsilver/MAPI/http.json");
adapter.modifyNamespaceHTTP(namespaceName, tenantName, f);

// Configure HTTP for the Accounts-Payable namespace, using http.json
// as input.
String namespaceName = "Accounts-Payable";
f = new File("/home/rsilver/MAPI/http.json");
adapter.modifyNamespaceHTTP(namespaceName, tenantName, f);

// Create a user account with no roles, using pblack-UA.json as input
// and specifying start123 as the account password.
f = new File("/home/rsilver/MAPI/pblack-UA.json");
adapter.createTenantUserAccount(tenantName, "start123", f);

// Modify the user account, using pblack-UAperms.json as input.
String userAcctName = "pblack"
f = new File("/home/rsilver/MAPI/pblack-UAperms.json");
adapter.changeDataUserAccountPerms(userAcctName, tenantName, f);

// Create a retention class, using RC-FN-Std-42.json as input.
f = new File("/home/rsilver/MAPI/RC-FN-Std-42.json");
adapter.createRetentionClass(namespaceName, tenantName, f);

// Create a replication link, using LinkMA-CA.json as input
// and add HCP tenant Finance
adapter.setUpSystemInfo(hcpSystemAddr, adminUser, adminPassword);
f = new File("/home/rsilver/MAPI/LinkMA-CA.json");
adapter.createReplicationLink(f, tenantName);

} catch (HCPNotInitializedException e) {

```

```

        e.printStackTrace();
    } finally {
        if(adapter != null) {
            adapter.shutdownHttpClient();
        }
    }
}

/**
 * Constructor - initializes the HTTP client.
 */
public MAPISample() throws HCPNotInitializedException{
    initHttpClient();
}

/**
 * When done with this adapter, shut it down.
 */
public void shutdownHttpClient() {
    httpClient.getConnectionManager().shutdown();
}

/**
 * Initialize the HCP system access settings.
 * @param hcpSystemAddr
 * @param username
 * @param password
 */
public void setUpSystemInfo(String hcpSystemAddr, String username,
                           String password)
{
    // This is the root for management API commands. In general, these values
    // should be retrieved from configuration settings.
    this.hcpSystemAddr = hcpSystemAddr;

    // The management API requires HTTPS and port 9090.
    protocol = "https";
    port = 9090;

    // Calculate the authentication token for management API access to HCP.
    BASE64Encoder base64Encoder = new BASE64Encoder();
    uname64 = base64Encoder.encode(username.getBytes());
    encodedPassword = toMD5Digest(password);
}

public void createTenant(File jsonInputFile, String tenantInitialUserName,
                        String password) {
    String addr = "admin." + hcpSystemAddr;
    String path = "/mapi/tenants";

    List<NameValuePair> metadata = new ArrayList<NameValuePair>();
    metadata.add(new BasicNameValuePair("username", tenantInitialUserName));
    metadata.add(new BasicNameValuePair("password", password));
}

```

```

        metadata.add(new BasicNameValuePair("forcePasswordChange", "false"));
        String queryString = URLEncodedUtils.format(metadata, "UTF-8");

        this.executeRequest(RequestType.PUT, addr, path, queryString, jsonInputFile);
    }

    public void modifyUserAccount(String tenantName, String username,
                                  File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName +
            "/userAccounts/" + username;

        this.executeRequest(RequestType.POST, addr, path, null, jsonInputFile);
    }

    public void createTenantUserAccount(String tenantName, String password,
                                          File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/userAccounts";

        List<NameValuePair> metadata = new ArrayList<NameValuePair>();
        metadata.add(new BasicNameValuePair("password", password));
        String queryString = URLEncodedUtils.format(metadata, "UTF-8");

        this.executeRequest(RequestType.PUT, addr, path, queryString, jsonInputFile);
    }

    public void configureTenantSecurity(String tenantName, File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/consoleSecurity";

        this.executeRequest(RequestType.POST, addr, path, null, jsonInputFile);
    }

    public void modifyTenant(String tenantName, File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName;

        this.executeRequest(RequestType.POST, addr, path, null, jsonInputFile);
    }

    public void createNamespace(String tenantName, File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/namespaces";

        this.executeRequest(RequestType.PUT, addr, path, null, jsonInputFile);
    }

    public void modifyNamespace(String namespaceName, String tenantName,
                                 File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/namespaces/" +
            namespaceName + "/complianceSettings";
    }

```

```

        this.executeRequest(RequestType.POST, addr, path, null, jsonInputFile);
    }

    public void modifyNamespaceHTTP(String namespaceName, String tenantName,
                                    File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/namespaces/" +
            namespaceName + "/protocols/http";

        this.executeRequest(RequestType.POST, addr, path, null, jsonInputFile);
    }

    public void changeDataUserAccountPerms(String dataUserAcctName,
                                            String tenantName, File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/userAccounts/" +
            dataUserAcctName + "/dataAccessPermissions";

        List<NameValuePair> metadata = new ArrayList<NameValuePair>();
        metadata.add(new BasicNameValuePair("debug", "true"));
        String queryString = URLEncodedUtils.format(metadata, "UTF-8");

        this.executeRequest(RequestType.POST, addr, path, queryString, jsonInputFile);
    }

    public void createRetentionClass(String namespaceName, String tenantName,
                                    File jsonInputFile) {
        String addr = tenantName + "." + hcpSystemAddr;
        String path = "/mapi" + "/tenants/" + tenantName + "/namespaces/" +
            namespaceName + "/retentionClasses";

        this.executeRequest(RequestType.PUT, addr, path, null, jsonInputFile);
    }

    public void createReplicationLink(File jsonInputFile, String tenantName) {
        String addr = "admin." + hcpSystemAddr;
        String path = "/mapi/services/replication/links";

        this.executeRequest(RequestType.PUT, addr, path, null, jsonInputFile);

        path = "/mapi/services/replication/links/MA-CA/content/tenants/" +
            tenantName;

        this.executeRequest(RequestType.PUT, addr, path, null, null);
    }

    /**
     * Execute the HTTP request to perform the applicable management API operation.
     * @param requestType
     * @param addr
     * @param path
     * @param queryString

```



```

* @param jsonInputFile
*/
private void executeRequest(RequestType requestType, String addr, String path,
                           String queryString, File jsonInputFile) {

    boolean success = false;
    try {
        // Set up the HTTP host.
        HttpHost httpHost = new HttpHost(addr, port, protocol);

        URI uri = URIUtils.createURI(protocol, addr, port, path, queryString,
                                     null);

        // JSON file.
        FileEntity fileEntity = new FileEntity(
            jsonInputFile, "application/json; charset=\"UTF-8\"");

        HttpRequest request;
        if(requestType == RequestType.PUT) {
            request = new HttpPut(uri);
            ((HttpPut)request).setEntity(fileEntity);
        } else {
            request = new HttpPost(uri);
            ((HttpPost)request).setEntity(fileEntity);
        }

        // Set up the authentication header.
        String header = "HCP " + uname64 + ":" + encodedPassword;
        request.setHeader("Authorization", header);

        // You should retry the request if the execute throws an IOException or
        // if HCP returns a server error. You should put the number of retry
        // attempts in a configuration file that can be changed depending on
        // network conditions.
        int retries = 3;
        while(retries > 0)
        {
            --retries;
            HttpResponse response = null;
            try {
                response = httpClient.execute(httpHost, request);
                if (response != null)
                {
                    // Get back the status and log it.
                    int statusCode = response.getStatusLine().getStatusCode();
                    System.out.println("Status code for PUT = " + statusCode);

                    // PUT returns a 201 (Created) if it is successful.
                    if(statusCode == 201) {
                        success = true;
                    }
                }

                // Return codes below 500 are due to either a successful

```

```

        // PUT, an error by the client, or an authentication error.
        // Errors >= 500 are HCP server errors, so you should retry
        // on those errors.
        if(statusCode < 500) {
            retries = 0;
            // Notify the user about the error. For descriptions of
            // the management API return codes, see Appendix A.
        }
        else {
            if(retries == 0) {
                // Notify your HCP system administrator about the
                // error.
            }

            // Wait two minutes; then try the request again.
            Thread.sleep(2*60*1000);
        }
    }

    catch(IOException e) {
        // An IOException from the client means there was a transport
        // error and is likely a one-off I/O issue. Try the request
        // again.
        e.printStackTrace();

        if(retries == 0) {
            // Notify your network administrator.
        }
    }
    // Clean up after ourselves and release the HTTP connection to the
    // connection manager.
    EntityUtils.consume(httpResponse.getEntity());
}
} catch (URISyntaxException e) {
    e.printStackTrace();
} catch (InterruptedException e) {
    e.printStackTrace();    // Wait.
}
}

/**
 * Start the HTTP client.
 */
private void initHttpClient() throws HCPNotInitializedException
{
    // Register the HTTPS scheme.
    SchemeRegistry schemeRegistry = new SchemeRegistry();
    try {
        // The recommended protocol is TLS.
        SSLContext sslcontext = SSLContext.getInstance("TLS");

        // The trust manager used here was created for use with this sample

```

```

// application. For more information on creating trust managers, see
// http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/
// JSSERefGuide.html#TrustManager
MyX509TrustManager trustMgr = new MyX509TrustManager();

sslcontext.init(null, new TrustManager[] {trustMgr}, null);
SSLSocketFactory sslSocketFactory = new SSLSocketFactory(sslcontext);

// The hostname verifier verifies that the hostname matches the one
// stored in the X.509 certificate on the server (that is, the SSL
// server certificate used by the HCP system). You can use
// AllowAllHostnameVerifier, BrowserCompatHostnameVerifier, or
// StrictHostnameVerifier. This sample application allows all hostnames.
sslSocketFactory.setHostnameVerifier(
    SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);

// Register the HTTPS scheme.
Scheme https = new Scheme("https", sslSocketFactory, 9090);
schemeRegistry.register(https);

// Specify any HTTP parameters you want.
HttpParams params = new BasicHttpParams();
params.setIntParameter("http.connection.timeout", 60000);

// This manages a thread-safe pool of connections that are created on
// first request, then persisted and leased out to subsequent requests.
// By default, HCP closes a connection after ten minutes. To change
// this setting, contact your authorized HCP service provider.
ClientConnectionManager connMgr = new ThreadSafeClientConnManager(
    params, schemeRegistry);
ConnPerRouteBean connPerRoute = new ConnPerRouteBean(20);

// HCP recommended settings: max connections per node = 20;
// total max connections = 200
ConnManagerParams.setMaxConnectionsPerRoute(params, connPerRoute);
ConnManagerParams.setMaxTotalConnections(params, 200);

// Ensure that the connection manager does not block indefinitely in the
// connection request operation.
ConnManagerParams.setTimeout(params, 2000);    // milliseconds

// Create the HTTP client.
httpClient = new DefaultHttpClient(connMgr, params);

} catch (NoSuchAlgorithmException e1) {
    throw new HCPNotInitializedException(e1.getMessage());
} catch (KeyManagementException e1) {
    throw new HCPNotInitializedException(e1.getMessage());
}
}

private static final String HEX_DIGITS[] = {"0", "1", "2", "3", "4", "5", "6",
                                             "7", "8", "9", "A", "B", "C", "D",

```

```

        "E", "F"};
private static String encodeBytes(byte[] bytes) {
    if (bytes == null || bytes.length == 0) {
        return "";
    }

    StringBuffer out = new StringBuffer(bytes.length * 2);

    byte ch;
    for (int i = 0; i < bytes.length; i++) {
        ch = (byte) (bytes[i] & 0xF0);
        ch = (byte) (ch >>> 4);
        ch = (byte) (ch & 0x0F);
        out.append(HEX_DIGITS[(int) ch]);
        ch = (byte) (bytes[i] & 0x0F);
        out.append(HEX_DIGITS[(int) ch]);
    }

    return out.toString();
}

protected String toMD5Digest(String sInStr) {
    StringBuffer mOutDigest = new StringBuffer("");

    try {
        MessageDigest pMD = MessageDigest.getInstance("MD5");

        byte pDigest[] = pMD.digest(sInStr.getBytes());

        // Convert to string.
        for(int i=0; i < pDigest.length; i++) {
            mOutDigest.append(Integer.toHexString(0xFF & pDigest[i]));
        }
    }
    catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
        e.printStackTrace();
    }

    return mOutDigest.toString();
}

}

/* Simple trust manager implementation. */

class MyX509TrustManager implements X509TrustManager {
    private X509TrustManager standardTrustManager = null;

    public MyX509TrustManager() {
    }

    public MyX509TrustManager(KeyStore keystore)

```

```

        throws NoSuchAlgorithmException, KeyStoreException {
    super();
    TrustManagerFactory factory =
        TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
    factory.init(keystore);
    TrustManager[] trustmanagers = factory.getTrustManagers();
    if (trustmanagers.length == 0) {
        throw new NoSuchAlgorithmException("no trust manager found");
    }
    this.standardTrustManager = (X509TrustManager)trustmanagers[0];
}

    public void checkClientTrusted(X509Certificate[]
certificates,String authType) throws CertificateException {
    standardTrustManager.checkClientTrusted(certificates,authType);
}

    public void checkServerTrusted(X509Certificate[]
certificates,String authType) throws CertificateException {
    if ((certificates != null) && (certificates.length == 1)) {
        certificates[0].checkValidity();
    } else {
        standardTrustManager.checkServerTrusted(certificates,authType);
    }
}

    public X509Certificate[] getAcceptedIssuers() {
        return this.standardTrustManager.getAcceptedIssuers();
    }
}

```




Management API XML schema

HCP uses an XML schema to validate the XML in management API request bodies and to generate the XML in management API response bodies. To retrieve this schema from the HCP system, you use a URL with this format:

```
https://(admin|tenant-name).hcp-domain-name:9090/static/  
mapi-7_0_0.xsd
```

To retrieve the schema, you need a system-level or tenant-level user account with the administrator role.

Here's a sample **curl** command that retrieves the management API schema and writes it to a file named `mapi_schema.xsd`:

```
curl -k -i -H "Accept: application/xml"  
-H "Authorization: HCP YWxscm9sZXNM=:04EC9F614D89FF5C7126D32ACB448382"  
"https://admin.hcp.example.com:9090/static/mapi-7_0_0.xsd"  
> mapi_schema.xsd
```




Glossary

A

access control list (ACL)

Optional metadata consisting of a set of grants of permissions to perform various operations on an object. Permissions can be granted to individual users or to groups of users.

ACLs are provided by users or applications and are specified as either XML or JSON.

access protocol

See [namespace access protocol](#).

ACL

See [access control list \(ACL\)](#).

Active Directory (AD)

A Microsoft product that, among other features, provides user authentication services.

active/active link

A replication link on which data is replicated in both directions between the two HCP systems. The HCP tenants and namespaces and default-namespace directories included in the link are read-write on both systems.

active/passive link

A replication link on which data is replicated in one direction between the two HCP systems. The HCP tenants and namespaces and default-namespace directories included the link are read-write on only one system at a time.

AD

See [Active Directory \(AD\)](#).

allow list

A list of IP addresses that are allowed access to the HCP system when using a particular external interface (such as the System Management Console).

annotation

A discrete unit of custom metadata. Annotations are typically specified in XML format.

anonymous access

A method of access to a namespace wherein the user or application gains access without presenting any credentials. *See also* [authenticated access](#).

atime

In POSIX file systems, metadata that specifies the date and time a file was last accessed. In HCP, POSIX metadata that initially specifies the date and time at which an object was ingested. HCP does not automatically change the **atime** value when the object is accessed.

Users and applications can change this metadata, thereby causing it to no longer reflect the actual storage time. Additionally, HCP can be configured to synchronize **atime** values with retention settings.

authenticated access

A method of access to the HCP system or a namespace wherein the user or application presents credentials to gain access. *See also* [anonymous access](#).

authentication

See [user authentication](#).

C**chargeback report**

A report that contains historical statistics about tenant or namespace capacity and bandwidth usage, broken out either by hour or by day. You can also generate chargeback reports that contain a single set of statistics for each tenant or namespace covering the past 180 days (or as much of that time period for which the statistics are available).

CIFS

Common Internet File System. One of the namespace access protocols supported by HCP. CIFS lets Windows clients access files on a remote computer as if the files were part of the local file system.

comma-separated-values (CSV) file

A text file containing tabular data. Each line in a CSV file corresponds to a table row and contains a set of comma-separated values, each of which corresponds to a table column.

compliance mode

The retention mode in which objects under retention cannot be deleted through any mechanism. This is the more restrictive retention mode.

content class

A content class is a named construct that is used to characterize objects in one or more namespaces. Content classes use object metadata to impose structure on the unstructured namespace content. They do this through content properties.

cryptographic hash value

A system-generated metadata value calculated by a cryptographic hash algorithm from object data. This value is used to verify that the content of an object has not changed.

CSV file

See [comma-separated-values \(CSV\) file](#).

custom metadata

User-supplied information about an HCP object. Custom metadata is specified as one or more annotations, where each annotation is a discrete unit of information about the object. Users and applications can use custom metadata to understand and repurpose object content.

D**data access permission mask**

A set of permissions that determine which of these operations are allowed in a namespace: read (including read ACL), write (including write ACL and change owner), delete, purge, privileged operations, and search. Data access permission masks are defined at the system, tenant, and namespace level. The effective permissions for a namespace are those that are allowed at all three levels.

data protection level (DPL)

The number of copies of the data for an object HCP must maintain in the repository. The DPL for an object is determined by the service plan that applies to the namespace containing the object.

data recovery

For an active/passive replication link, the process of copying data from the replica back to the primary system after the link has been failed over.

default namespace

A namespace that supports only anonymous access through the HTTP protocol. An HCP system can have at most one default namespace. The default namespace is used mostly with applications that existed before release 3.0 of HCP.

default tenant

The tenant that manages the default namespace.

deny list

A list of IP addresses that are denied access to the HCP system when using a particular external interface (such as the System Management Console).

DNS

See [domain name system \(DNS\)](#).

domain

A group of computers and devices on a network that are administered as a unit.

domain name system (DNS)

A network service that resolves domain names into IP addresses for client access.

DPL

See [data protection level \(DPL\)](#).

dynamic DPL

A namespace data protection level that, at any given time, matches the system-level DPL setting.

E

enterprise mode

The retention mode in which these operations are allowed:

- ⑩ Privileged delete
- ⑩ Changing the retention class of an object to one with a shorter duration
- ⑩ Reducing retention class duration
- ⑩ Deleting retention classes

This is the less restrictive retention mode.

F

failback

The process that restarts replication on a link that has been failed over and returns the HCP systems involved in the link to normal operation. Typically, you fail back a link when an unavailable system becomes available again.

failover

The process that stops replication on a replication link. Typically, you fail over a link when one of the systems involved in the link becomes unavailable.

G

GID

POSIX group identifier.

group account

A representation of an Active Directory group in HCP. A group account enables Active Directory users in the Active Directory group to access one or more HCP interfaces.

H

hard quota

For an HCP tenant, the total amount of storage available to the tenant for allocation to its namespaces. For an HCP namespace, the total amount of storage available for storing objects in the namespace.

hash value

See [cryptographic hash value](#).

HCP

See [Hitachi Content Platform \(HCP\)](#).

HCP management API

A RESTful HTTP interface to a subset of the administrative functions of an HCP system. Using this API, you can manage tenants, namespaces, content classes, retention classes, and tenant-level user and group accounts.

HCP namespace

A namespace that supports user authentication for data access through the HTTP, HS3, and CIFS protocols. HCP namespaces also support storage usage quotas, access control lists, and versioning. An HCP system can have multiple HCP namespaces.

HCP node

See [node](#).

HCP tenant

A tenant created to manage HCP namespaces.

HDDS

See [Hitachi Data Discovery Suite \(HDDS\)](#).

HDDS search facility

One of the search facilities available for use with the HCP Search Console. This facility interacts with Hitachi Data Discovery Suite. To use this facility, you need to first install and configure HDDS, which is a separate product from HCP.

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.

Hitachi Data Discovery Suite (HDDS)

A Hitachi product that enables federated searches across multiple HCP systems and other supported systems.

HS3 API

One of the namespace access protocols supported by HCP. HS3 is a RESTful, HTTP-based API that is compatible with Amazon S3. Using HS3, users and applications can create and manage buckets and bucket contents.

HTTP

HyperText Transfer Protocol. One of the namespace access protocols supported by HCP. In the context of namespace access, the HTTP protocol is also called the REST API.

HCP also uses HTTP for client communication with the System Management, Tenant Management, and Search Consoles and for client access through the HCP management API

HTTPS

HTTP with SSL security. See [HTTP](#) and [SSL](#).

I**inbound link**

An active/passive replication link from the perspective of the replica for the link.

index

An index of the objects in namespaces that is used to support search operations.

index setting

The property of an object that determines whether the metadata query engine indexes the custom metadata associated with the object.

J

JSON

JavaScript Object Notation. A language-independent format for encoding data in the form of name/value pairs.

L

local authentication

Authentication wherein HCP internally checks the validity of the specified username and password.

M

management API

See [HCP management API](#).

metadata

System-generated and user-supplied information about an object. Metadata is stored as an integral part of the object it describes, thereby making the object self-describing.

metadata query API

A RESTful HTTP interface that lets you search HCP for objects that meet specified metadata-based or operation-based criteria. With this API, you can search not only for objects currently in the repository but also for information about objects that are no longer in the repository.

metadata query engine

One of the search facilities available for use with HCP. The metadata query engine works internally to perform searches and return results either through the metadata query API or to the HCP Metadata Query Engine Console (also known as the HCP Search Console).

Metadata Query Engine Console

The web application that provides interactive access to the HCP search functionality provided by the metadata query engine.

N

namespace

A logical partition of the objects stored in an HCP system. A namespace consists of a grouping of objects such that the objects in one namespace are not visible in any other namespace. Namespaces are configured independently of each other and, therefore, can have different properties.

namespace access protocol

A protocol that can be used to transfer data to and from namespaces in an HCP system. HCP supports the HTTP, HS3, WebDAV, CIFS, NFS, and SMTP protocols for access to HCP namespaces and the default namespace. HCP also supports the NDMP protocol for access to the default namespace.

NFS

Network File System. One of the namespace access protocols supported by HCP. NFS lets clients access files on a remote computer as if the files were part of the local file system.

node

A server running HCP software and networked with other such servers to form an HCP system.

O

object

An exact digital representation of data as it existed before it was ingested into HCP, together with the system and custom metadata that describes that data. Objects can also include ACLs that give users and groups permission to perform certain operations on the object.

An object is handled as a single unit by all transactions and internal processes, including shredding, indexing, versioning, and replication.

outbound link

An active/passive replication link from the perspective of the primary system for the link.

P

permission

One of these:

- In a data access permission mask, the condition of allowing a specific type of operation to be performed in a namespace.
- In a tenant-level user account, the granted ability to perform a specific type of operation in a given namespace.
- In an ACL associated with an object, the granted ability to perform a specific type of operation on the object.
- The granted ability to access the HCP System Management Console, Tenant Management, or HCP Search Console and to perform a specific activity or set of activities in that Console. Permissions of this type are granted by roles associated with the user account.

permission mask

See [data access permission mask](#).

primary system

For an active/passive replication link, the HCP system from which the replication service objects and other information to the replica during normal replication.

privileged delete

A delete operation that works on an object regardless of whether the object is under retention, except if the object is on hold. This operation is available only to users and applications with explicit permission to perform it.

Privileged delete operations work only in namespaces in enterprise mode.

privileged purge

A purge operation that works on an object regardless of whether the object is under retention, except if the object is on hold. This operation is available only to users and applications with explicit permission to perform it.

Privileged purge operations work only in namespaces in enterprise mode.

protocol

See [namespace access protocol](#).

pruning

See [version pruning](#).

purge

The operation that deletes all versions of an object.

R**RADIUS**

Remote Authentication Dial-In User Service. A protocol for authenticating credentials that authorize access to an IP network.

recognized Active Directory user account

An Active Directory user account for a user that belongs to one or more Active Directory groups for which corresponding group accounts are defined in HCP.

recovery

See [data recovery](#).

remote authentication

Authentication wherein HCP uses a remote service to check the validity of the specified username and password.

replica

For an active/passive link, the HCP system to which the replication service copies objects and other information from the primary system during normal replication.

replication

The process of keeping selected HCP tenants and namespaces and selected default-namespace directories in two HCP systems in sync with each other. Basically, this entails copying object creations, deletions, and metadata changes from each system to the other or from one system to the other. HCP also replicates tenant and namespace configuration, tenant-level user and group accounts, retention classes, content classes, and all HCP tenant log messages.

replication link

A configurable, secure trust relationship between two HCP systems that determines what is replicated between the systems and how data is transmitted between the systems.

repository

The aggregate of the namespaces defined for an HCP system.

resource

An entity that you can manage independently in the HCP management API. Examples of resources are tenants and namespaces.

REST

Representational State Transfer. A software architectural style that defines a set of rules (called constraints) for client/server communication. In a REST architecture:

- Ⓢ Resources (where a resource can be any coherent and meaningful concept) must be uniquely addressable.
- Ⓢ Representations of resources (for example, in XML format) are transferred between clients and servers. Each representation communicates the current or intended state of a resource.
- Ⓢ Clients communicate with servers through a uniform interface (that is, a set of methods that resources respond to) such as HTTP.

REST API

One of the namespace access protocols supported by HCP. The REST API is also called the HTTP protocol.

retention class

A named retention setting. The value of a retention class can be a duration, Deletion Allowed, Deletion Prohibited, or Initial Unspecified.

retention mode

A namespace property that affects which operations are allowed on objects under retention. A namespace can be in either of two retention modes: compliance or enterprise.

retention period

The period of time during which an object cannot be deleted (except by means of a privileged delete).

retention setting

The property that determines the retention period for an object.

role

A named collection of permissions that can be associated with an HCP user account, where each permission allows the user to perform some specific interaction or set of interactions with the HCP System Management Console, the Tenant Management Console, the HCP management API, the metadata query API, or, for default namespaces only, the HCP Search Console. Roles generally correspond to job functions.

S**Search Console**

The web application that provides interactive access to HCP search functionality. When the Search Console uses the HCP metadata query engine for search functionality, it is called the Metadata Query Engine Console.

search facility

An interface between the HCP Search Console and the search functionality provided by the metadata query engine or HDDS. Only one search facility can be selected for use with the Search Console at any given time.

service plan

A named specification of an HCP service behavior that determines how HCP manages objects in a namespace. Service plans enable you to tailor service activity to specific namespace usage patterns or properties.

shred setting

The property that determines whether an object will be shredded or simply removed when it's deleted from HCP.

shredding

The process of deleting an object and overwriting the locations where all its copies were stored in such a way that none of its data or metadata can be reconstructed. Also called **secure deletion**.

SMTP

Simple Mail Transfer Protocol. The namespace access protocol HCP uses to receive and store email data directly from email servers.

SNMP

Simple Network Management Protocol. A protocol HCP uses to facilitate monitoring and management of the system through an external interface.

soft quota

The percentage point at which HCP notifies a tenant that allocated storage space is being used up. For a tenant, the soft quota measures the space used in all the namespaces the tenant owns relative to the hard quota for that tenant. For a namespace, the soft quota measures the space used in only that namespace relative to the hard quota for that namespace.

SPNEGO

Simple and Protected GSSAPI Negotiation. A protocol used for client authentication against a remote server.

SSL

Secure Sockets Layer. A key-based Internet protocol for transmitting documents through an encrypted link.

SSL server certificate

A file containing cryptographic keys and signatures. When used with the HTTP protocol, an SSL server certificate helps verify that the web site holding the certificate is authentic. An SSL server certificate also helps protect data sent to or from that site.

storage node

An HCP node that manages the objects that are added to HCP and can be used for object storage. Each storage node runs the complete HCP software.

subdomain

A subset of the computers and devices in a domain.

syslog

A protocol used for forwarding log messages in an IP network. HCP uses syslog to facilitate system monitoring through an external interface.

System Management Console

The system-specific web application that lets you monitor and manage HCP.

system metadata

System-managed properties that describe the content of an object. System metadata includes policies, such as retention and data protection level, that influence how transactions and services affect the object.

T**tag**

An arbitrary text string associated with an HCP tenant or namespace. Tags can be used to group tenants or namespaces and to filter tenant or namespace lists.

tenant

An administrative entity created for the purpose of owning and managing namespaces. Tenants typically correspond to customers or business units.

Tenant Management Console

The tenant-specific web application that lets you monitor and manage tenants and namespaces.

U**UID**

POSIX user ID.

Unix

Any UNIX-like operating system (such as UNIX itself or Linux).

user account

A set of credentials that gives a user access to one or more of the System Management Console, the Tenant Management Console, the HCP management API, the HCP Search Console, namespace content through the namespace access protocols, and the metadata query API.

user authentication

The process of checking that the combination of a specified username and password is valid when a user tries to log into the System Management Console, the Tenant Management Console, or the HCP Search Console, to access the HCP system through the management API, or to access a namespace.

V

versioning

An optional namespace feature that enables the creation and management of multiple versions of an object.

version pruning

The automatic deletion of previous versions of objects that are older than a specified amount of time.

W

WADL

Web Application Description Language. An XML-based description language for HTTP-based web applications.

WebDAV

Web-based Distributed Authoring and Versioning. One of the namespace access protocols supported by HCP. WebDAV is an extension of HTTP.

X

XML

Extensible Markup Language. A standard for describing data content using structural tags called elements.

XPath

A language used to formulate expressions that navigate through and select elements and attributes in XML documents.

Index

A

- Accept request header 8–9
- access methods, choosing 282
- ACL usage options 179
- acIsUsage property 179
- action property 220
- actions
 - performing on replication links 83
 - performing on replication service 57, 82, 95
 - performing on tenants in links 85
- Active Directory user accounts 4, 28
- adding
 - chained links to links 84
 - directories to links 84
 - tenants to links 85, 91–92
- address property 209
- address1 property 130
- address2 property 130
- administrationAllowed property 248
- administrative access to tenants from system
 - level 5, 32, 103
- Allow response header 13
- allowAddresses property 161
- allowDisposition property 225
- allowIfInBothLists property
 - about 161
 - effects 163
- allowNamespaceManagement property
 - groupAccount data type 151
 - userAccount data type 269
- allowPermissionAndOwnershipChanges property 180
- allowTenantsToMonitorNamespaces property 222
- appendEnabled property 180
- application, sample 290–309
- application/javascript Internet media type 8
- application/json Internet media type 8
- application/xml Internet media type 8
- atimeSynchronizationEnabled property 181
- authAndAnonymousMinimumPermissions property 181
- authenticatedMinimumPermissions property 105–106
- authentication
 - about 28
 - Active Directory user accounts 28
 - Authorization header 29
 - hcp-api-auth cookie 28
- authentication token
 - about 28
 - generating 29
- authenticationType property 249
- authenticationTypes property 249
- authMinimumPermissions property 182
- Authorization header
 - about 29
 - specifying with cURL 29
- authUsersAlwaysGrantedAllPermissions property 183
- autoCompleteRecovery property
 - link failoverSettings local data type 174
 - link failoverSettings remote data type 214
- autoCompleteRecoveryMinutes property
 - link failoverSettings local data type 175
 - link failoverSettings remote data type 214
- autoFailover property
 - link failoverSettings local data type 175
 - link failoverSettings remote data type 215
- autoFailoverMinutes property
 - link failoverSettings local data type 176
 - link failoverSettings remote data type 215
- available service plans
 - listing in JSON 34
 - listing in XML 34
 - retrieving individual 35
 - retrieving lists of 34
- availableServicePlan data type 106–107
- availableServicePlans resource identifier 34

B

Base64 username encoding 28–29
 beginning recovery on links 172
 beginRecovery query parameter 172
 body property 147
 Boolean values 15, 105
 bytesPending property
 statistics data type for links 233
 tenant data type for link content 243
 bytesPendingRemote property
 statistics data type for links 233
 tenant data type for link content 243
 bytesPerSecond property 234
 bytesReplicated property 234

C

case sensitivity 179
 annotation names 143
 authentication types 249
 Boolean values 105
 case forcing options 119
 CIFS 119
 content class names 133, 142
 custom metadata changes options 121
 email location directories 230
 hashScheme values 187, 201
 HTTP headers 9
 Internet media types 9
 link names 167
 link priority 168
 link type 171
 message importance options 209
 message severity options 210
 message type options 211
 namespace names 189
 network names 223
 owner type options 190
 passwords 104
 percent-encoded values 27
 performance levels 177, 218, 267
 permissions 38, 49, 181, 182, 204
 query parameters 15
 replication collision action options 220
 retention class names 226
 roles 271
 special values for retention 277
 standard expression for retention offsets 278
 tags 196, 261
 tenant names 254
 user roles 153
 usernames 272

 usernames for WebDAV basic
 authentication 159
 caseForcing property 119
 caseSensitive property 119
 chainedLinks
 link content resource identifier 84
 link local candidates resource identifier 86
 link remote candidates resource identifier 87
 property 132
 changing
 See [modifying](#)
 chargeback reports
 See also [chargebackData](#); [chargebackReport](#)
 about 112–113
 generating (namespaces) 47
 generating (tenants) 35
 query parameters 114–116
 reporting interval 113
 sample 117–118
 statistics collection 107
 time period covered 113
 chargebackData
 See also [chargeback reports](#);
 [chargebackReport](#)
 data type 107–112
 property 113
 chargebackReport
 See also [chargeback reports](#);
 [chargebackData](#)
 data type 112, 113–118
 query parameters 114–116
 resource identifier (namespaces) 47
 resource identifier (tenants) 35
 checking existence
 content classes 65
 group accounts 78
 namespaces 45
 replication links 83
 retention classes 61
 tenants 34
 user accounts 70
 choosing access methods 282
 cifs resource identifier 50
 cifsProtocol data type 119–121
 city property 130
 common property values 104–106
 completeRecovery query parameter 173
 completing recovery on links 173
 compliance settings
 modifying 47, 54–55
 retrieving 47
 complianceConfigurationEnabled property 250

- complianceSettings
 - data type 121–123
 - resource identifier 47
 - compressedCount property 240
 - compressedSavedSize property 240
 - compression property 166
 - connection
 - data type 123–127
 - property 166
 - consoleSecurity
 - See also [Tenant Management Console configuration](#)
 - data type 127–130
 - resource identifier 35
 - contact information
 - modifying 36
 - retrieving 36
 - contactInfo
 - data type 130–131
 - resource identifier 36
 - content
 - data type 132–133
 - resource identifier 83
 - content classes
 - checking existence 65
 - creating 65, 66–68
 - deleting 66
 - listing in JSON 65
 - listing in XML 65
 - modifying 65, 66
 - resources 64–66
 - retrieving lists of 65, 68–69
 - contentClass data type 133–135
 - contentClasses
 - property 142
 - resource identifier 65
 - content-class-name* resource identifier 65
 - contentProperties
 - data type 140–141
 - property 133
 - contentProperty
 - data type 136–140
 - property 140
 - Content-Type request header 8–9
 - count query parameter 99–100
 - countryOrRegion property 130
 - creating
 - content classes 65, 66–68
 - group accounts 77, 79–80
 - group accounts, initial 264–265
 - HCP tenants 39–41
 - namespaces 44, 52–53
 - replication links 82, 88–90, 95–??
 - resources 7
 - retention classes 60, 62–63
 - tenants 33, 263–265
 - user accounts ??–58, 70, 72–73
 - user accounts, initial 263–264
 - creationTime property
 - namespace data type 183
 - tenant data type 250
 - CSV
 - Internet media type for 8
 - response body contents 12
 - cURL
 - adding a tenant to a link 91
 - creating a content class 67
 - creating a group account 80
 - creating a namespace 53
 - creating a replication link 89, 96
 - creating a retention class 62
 - creating a user account 57, 58, 72
 - creating an HCP tenant 40
 - failing over a link 94
 - H option 8, 24, 29
 - k option 23
 - modifying a user account 58, 73
 - modifying compliance settings 54
 - modifying the HTTP protocol 56
 - modifying the Tenant Management Console configuration 42
 - modifying user account permissions 75
 - retrieving a list of eligible local candidates 90
 - retrieving content classes 68
 - retrieving group accounts 81
 - retrieving retention classes 63
 - setting the schedule for a link 93
 - custom metadata handling setting
 - modifying 47
 - retrieving 47
 - custom metadata indexing settings for metadata
 - query engine
 - modifying 48
 - retrieving 48
 - customMetadataChanges property 121
 - customMetadataCount property 241
 - customMetadataIndexingSettings
 - data type 142–143
 - resource identifier 48
 - customMetadataSize property 241
 - customMetadataValidationEnabled property 184
- D**
- data access permission masks
 - modifying (namespaces) 49
 - modifying (tenants) 38

data access permission masks, retrieving (namespaces)

- retrieving (namespaces) 48
- retrieving (tenants) 37
- data access permissions
 - filtering 101–102
 - group account, modifying 79
 - group account, retrieving 79
 - paging 98–100
 - sorting 100–101
 - user account, modifying 71
 - user account, retrieving 71
- data types
 - about 6–7, 103
 - availableServicePlan 106–107
 - chargebackData 107–112
 - chargebackReport 112, 113–118
 - cifsProtocol 119–121
 - complianceSettings 121–123
 - connection 123–127
 - consoleSecurity 127–130
 - contactInfo 130–131
 - content 132–133
 - contentClass 133–135
 - contentProperties 140–141
 - contentProperty 136–140
 - customMetadataIndexingSettings 142–143
 - dataAccessPermissions 144
 - emailNotification 145–146
 - emailTemplate 147–149
 - failoverSettings 150
 - groupAccount 151–153
 - httpProtocol 154–160
 - ipSettings 160–163
 - link 163–??, 166–172
 - local (failoverSettings) 173–176
 - local (link schedule) 176–178
 - namespace 178–198
 - namespaceDefaults 198–203
 - namespacePermission 203–205
 - nfsProtocol 206–207
 - protocols 207–208
 - recipient 208–211
 - recipients 211–213
 - remote (failoverSettings) 213–215
 - remote (link schedule) 216–219
 - replicationCollisionSettings 219–221
 - replicationService 221–224
 - retentionClass 225–227
 - schedule 227–228
 - searchSecurity 228–229
 - smtpProtocol 229–231
 - statistics (links) 231–240
 - statistics (tenants and namespaces) 240–242
 - tenant (link content) 242–246
 - tenant (tenant resource) 247–263
 - userAccount 268–273
 - versioningSettings 274–277
- dataAccessPermissions
 - data type 144
 - resource identifier 71, 79
- dataNetwork property 251
- daysOnPrimary property 274
- daysOnReplica property 274
- default email message template 149
- default namespace
 - See [namespaces](#)
- default retention setting
 - modifying 47
 - retrieving 47
- default shred setting
 - modifying 47
 - retrieving 47
- default tenant
 - See also [tenants](#)
- default tenant, query parameters for
 - creating 265
- defaultNamespaceDirectories
 - property 132
 - resource identifier (link content) 83
 - resource identifier (link local candidates) 86
 - resource identifier (link remote candidates) 87
- DELETE
 - about 7
 - deleting content classes 66
 - deleting group accounts 78
 - deleting namespaces 46
 - deleting replication links 83
 - deleting retention classes 61
 - deleting tenants 34
 - deleting user accounts 71
 - removing chained links from links 84
 - removing directories from links 84
 - removing tenants from links 85
- deleted property 111
- deleteDays property 220
- deleteEnabled property 221
- deleting
 - See also [removing](#)
 - content classes 66
 - group accounts 78
 - namespaces 46
 - replication links 83
 - retention classes 61
 - tenants 34
 - user accounts 71

- Deletion Allowed 277
- Deletion Prohibited 277
- denyAddresses property 162
- description property
 - availableServicePlan data type 106
 - link data type 166
 - namespace data type 185
 - namespaceDefaults data type 199
 - retentionClass data type 226
 - userAccount data type 270
 - valid values 105
- directories
 - adding to links 84
 - removing from links 84
- directory-name* resource identifier 84
- disableAfterAttempts property 128
- disableAfterInactiveDays property 128
- disposition setting
 - modifying 47
 - retrieving 47
- dispositionEnabled property 122
- DNS name of HCP system 24
- dpl
 - property (namespace data type) 185
 - property (namespaceDefaults data type) 199
- dynamic statistics 107

E

- email message template
 - default 149
 - variables 148–149
- emailAddress property 130
- emailFormat property 229
- emailLocation property 230
- emailNotification
 - data type 145–146
 - resource identifier 36
- emailTemplate
 - data type 147–149
 - property 145
- enabled property
 - cifsProtocol data type 120
 - emailNotification data type 145
 - nfsProtocol data type 205, 206
 - smtpProtocol data type 230
 - userAccount data type 270
 - versioningSettings data type 275
- enableDNSFailover property 222
- enableDomainAndCertificateSynchronization
 - property 223
- enabling HCP management API 19
- encryption property 167
- end query parameter 115
- enterpriseMode
 - property (namespace data type) 185
 - property (namespaceDefaults data type) 200
 - query parameter 265
- error codes 286–288
- errors property 234
- errorsPerSecond property 235
- examples
 - adding a tenant to a link 91–92
 - availableServicePlan data type 107
 - chargebackReport response body (CSV) 12
 - cifsProtocol data type 120–121
 - complianceSettings data type 123
 - connection data type 127
 - consoleSecurity data type 129–130
 - contactInfo data type 131
 - content data type 133
 - contentClass data type 134–135
 - contentProperties data type 140–141
 - contentProperty data type 140
 - creating a content class 66–68
 - creating a group account 79–80
 - creating a namespace 52–53
 - creating a replication link 88–90, 95–??
 - creating a retention class 62–63
 - creating a user account ??–58, 72–73
 - creating an HCP tenant 39–41
 - customMetadataIndexingSettings data
 - type 143
 - dataAccessPermissions data type 144
 - emailNotification data type 146
 - emailTemplate data type 149
 - failing over a link 94–95
 - failoverSettings data type 150
 - filtering namespace lists 102
 - groupAccount data type 153
 - httpProtocol data type 160
 - ipSettings data type 163
 - Java application 299–309
 - link data type 171–172
 - local data type (link failoverSettings) 176
 - local data type (link schedule) 178
 - modifying a user account 58–59, 73–74
 - modifying compliance settings 54–55
 - modifying HTTP protocol 55–56
 - modifying the Tenant Management Console
 - configuration 41–42
 - modifying user account permissions 75–76
 - namespace data type 197–198
 - namespaceDefaults data type 202–203
 - namespacePermission data type 204–205

- nfsProtocol data type 206–207
- pausing tenant replication 247
- protocols data type 208
- recipient data type 211
- recipients data type 213
- remote data type (link failoverSettings) 215
- remote data type (link schedule) 219
- replicationCollisionSettings data type 221
- replicationService data type 224
- resetting security groups 154
- resetting security user passwords 274
- retentionClass data type 226–227
- retrieving a list of eligible local candidates 90–91
- retrieving content classes 68–69
- retrieving group accounts 81–82
- retrieving retention classes 63–64
- schedule data type 227–228
- searchSecurity data type 229
- setting the schedule for a link 92–94
- shutting down all replication links 225
- smtpProtocol data type 231
- sorting tenant lists 101
- statistics data type (link statistics) 240
- statistics data type (tenant and namespace statistics resource) 242
- suspending activity on a link 173
- tenant data type 262–263
- tenant data type (link content) 246
- tenant response body (JSON) 11–12
- tenant response body (XML) 10
- transition data type 268
- userAccount data type 273
- versioningSettings data type 277
- WADL response body 13–14
- excludedAnnotations property 143
- expression property 136
- extension property 130

F

- failBack query parameter 172
- failing back links 172
- failing over links 94–95, 172
- failOver query parameter 172
- failoverSettings
 - data type 150
 - property 167
- filtering 101–102
- filterString query parameter 101–102
- filterType query parameter 101–102
- firstName property 130
- fixed dates (retention settings) 279
- forcePasswordChange

- property 270
- query parameter 263
- forcePasswordChangeDays property 128
- format property 137
- formats
 - input 8–12
 - output 8–14
 - resource URLs 22–23
- from property 147
- fullIndexingEnabled property 142
- fullName property 270
- fullyQualifiedName property
 - namespace data type 186
 - tenant data type 251

G

- generating
 - chargeback reports (namespaces) 47
 - chargeback reports (tenants) 35
 - templates for resource creation 283
- GET
 - about 7
 - Accept request header 8
 - filtering 101–102
 - generating chargeback reports (namespaces) 47
 - generating chargeback reports (tenants) 35
 - modifying custom metadata indexing settings for metadata query engine 48
 - modifying link schedules 88
 - paging 98–100
 - prettyprint query parameter 16
 - retrieving available service plan lists 34
 - retrieving available service plans, individual 35
 - retrieving compliance settings 47
 - retrieving content class lists 65
 - retrieving content classes, individual 65
 - retrieving custom metadata handling setting 47
 - retrieving custom metadata indexing settings for metadata query engine 48
 - retrieving data access permission masks (namespaces) 48
 - retrieving data access permission masks (tenants) 37
 - retrieving default retention setting 47
 - retrieving default shred setting 47
 - retrieving disposition setting 47
 - retrieving group account lists 78
 - retrieving group account permissions 79
 - retrieving group accounts, individual 78
 - retrieving link content chained link lists 84

- retrieving link content directory lists 83
- retrieving link content lists 83
- retrieving link content tenant lists 85
- retrieving link local candidate chained link lists 86
- retrieving link local candidate directory lists 86
- retrieving link local candidate lists 85
- retrieving link local candidate tenant lists 86
- retrieving link remote candidate chained link lists 87
- retrieving link remote candidate directory lists 87
- retrieving link remote candidate lists 87
- retrieving link remote candidate tenant lists 88
- retrieving link schedules 88
- retrieving namespace defaults 37
- retrieving namespace lists 45
- retrieving namespaces, individual 45
- retrieving protocol settings 50
- retrieving replication collision handling settings 51
- retrieving replication link lists 82
- retrieving replication links, individual 83
- retrieving replication settings 57, 82, 95
- retrieving retention class lists 60
- retrieving retention classes, individual 60
- retrieving Search Console configuration 39
- retrieving statistics (namespaces) 51
- retrieving statistics (tenants) 36, 39
- retrieving tenant contact information 36
- retrieving tenant lists 33
- retrieving Tenant Management Console configuration 35
- retrieving tenant replication status 85
- retrieving tenants, individual 33
- retrieving user account lists 70
- retrieving user account permissions 71
- retrieving user accounts, individual 70
- retrieving versioning settings 52
- sorting 100–101
- verbose query parameter 15

gid property 206

granularity query parameter 116

See also [reporting interval for chargeback reports](#)

group accounts

- checking existence 78
- creating 77, 79–80
- creating initial 264–265
- deleting 78
- listing in JSON 78

- listing in XML 78
- modifying 78
- modifying data access permissions 79
- resetting security group 78, 153–154
- resources 77–79
- retrieving data access permissions 79
- retrieving individual 78
- retrieving lists of 78, 81–82

groupAccount data type 151–153

groupAccounts resource identifier 77

group-name resource identifier 78

H

-H option (cURL) 8, 24, 29

hardQuota property

- namespace data type 186
- namespaceDefaults data type 200
- sorting 100
- tenant data type 252

hashScheme

- property (namespace data type) 187
- property (namespaceDefaults data type) 201
- query parameter 265

HCP management API

- about 1–4
- credentials for 28–29
- enabling 19
- input formats 8–12
- output formats 8–14
- port number 23
- properties 6–7
- resource URLs 22–23
- resources 6–7
- SSL security 23
- supported methods 7
- with system-level administrative access to tenants 5, 32, 103
- user requirements 4–6

HCP namespaces

See [namespaces](#)

HCP system DNS name 24

HCP tenants

See also [tenants](#)

- creating 39–41
- query parameters for creating 263–265

hcp-api-auth cookie 28

HCP-specific response headers 18

HEAD

- about 7
- checking content class existence 65
- checking group account existence 78
- checking namespace existence 45
- checking replication link existence 83

HEAD, checking retention class existence

- checking retention class existence 61
- checking tenant existence 34
- checking user account existence 70
- Host header 24
- hostname mappings 25–26
- hosts files 24–26, 282
- hs3Enabled property 155
- hs3RequiresAuthentication property 155
- HTTP protocol, modifying 55–56
- HTTP request headers
 - Accept 8–9
 - Authorization 29
 - Content-Type 8–9
 - Host 24
 - specifying with cURL 8
 - specifying with Python with PycURL 9
- http resource identifier 50
- HTTP response headers
 - Allow 13
 - HCP specific 18
- HTTP return codes 286–288
- httpEnabled property 156, 207
- HTTPHEADER option (Python with PycURL) 9, 24, 29
- httpProtocol data type 154–160
- HTTPS 23
- httpsEnabled property 157, 208

I

- id property
 - link data type 167
 - namespace data type 187
 - tenant data type 252
- importance property 209
- indexing checkpoints, resetting 198
- indexing, restarting 198
- indexingDefault property 187
- indexingEnabled property 188
- ingestedVolume property 241
- initial
 - group accounts, creating 264–265
 - user accounts, creating 263–264
- Initial Unspecified 277
- initialSecurityGroup query parameter 264
- input files for sample application 291–298
- input formats 8–12
- Internet media types 8–9
- IP addresses
 - entries for allowAddresses/denyAddresses lists 162
 - listing 161, 162

- in resource URLs 23–24, 282
- ipAddress property 161, 162
- ipSettings
 - data type 160–163
 - property (cifsProtocol data type) 120
 - property (consoleSecurity data type) 129
 - property (httpProtocol data type) 157
 - property (nfsProtocol data type) 206
 - property (protocols data type) 208
 - property (searchSecurity data type) 228
 - property (smtpProtocol data type) 230
- isDplDynamic property 188

J

- Java
 - libraries for sample application 291
 - sample application 299–309
- JSON
 - allowAddresses list 161
 - authenticationType list 249
 - available service plan list 34
 - content class list 65, 142
 - denyAddresses list 162
 - empty request body 18
 - group account list 78
 - Internet media type for 8
 - link content chained link list 84, 132
 - link content directory list 83, 132
 - link content tenant list 85, 132
 - link list 82
 - link local candidate chained link list 86
 - link local candidate directory list 86
 - link local candidate tenant list 86
 - link remote candidate chained link list 87
 - link remote candidate directory list 87
 - link remote candidate tenant list 88
 - namespace list (contentClass data type) 134
 - namespace list (namespaces resource) 45
 - permission list 106
 - request body contents 11–12
 - response body contents 11–12
 - retention class list 60
 - role list 142, 153, 271
 - sample application input files 291–298
 - tag list 196, 261
 - tenant list 33
 - user account list 70

K

- k option (cURL) 23
- keepDeletionRecords property 275

L

- lastName property 130
- length of resource URLs 26
- libraries for sample application 291
- link data type 163–??, 166–172
- link resource identifier 82
- link-name*
 - resource identifier (link content) 84
 - resource identifier (links) 83
- links
 - See [replication links](#)
- local
 - data type (failoverSettings) 173–176
 - data type (link schedule) 176–178
 - property 150
- localAuthentication property 271
- localCandidates resource identifier 85
- localhost property 124
- localPort property 125
- loginMessage property 129
- logoutOnInactive property 129

M

- Mac OS X hosts file 24
- management API
 - See [HCP management API](#)
- managementNetwork property 252
- mapi interface identifier 23
- maxNamespacesPerUser property 253
- MD5 password hashing 28–29
- metadata query engine custom metadata
 - indexing settings
 - modifying 48
 - retrieving 48
- metadata query engine, restarting indexing 198
- methods, supported 7
- minimumPasswordLength property 129
- modifying
 - compliance settings 47, 54–55
 - content classes 66
 - custom metadata handling setting 47
 - custom metadata indexing settings for
 - metadata query engine 48
 - data access permission masks
 - (namespaces) 49
 - data access permission masks (tenants) 38
 - default retention setting 47
 - default shred setting 47
 - disposition setting 47
 - group account permissions 79
 - group accounts 78
 - HTTP protocol 55–56

- link schedules 88
- namespace defaults 37
- namespaces 46
- protocol settings 50
- replication collision handling settings 51
- replication links 83
- replication settings 57, 82, 95
- resources 7, 283
- retention classes 61
- Search Console configuration 39
- tenant contact information 36
- Tenant Management Console
 - configuration 35, 41–42
- tenants 34
- user account permissions 71, 75–76
- user accounts 58–59, 71, 73–74
- versioning settings 52
- multivalued property 138

N

- name property
 - availableServicePlan data type 106
 - contentClass data type 133
 - contentProperty data type 138
 - filtering 101
 - link data type 167
 - namespace data type 189
 - retentionClass data type 226
 - sorting 100
 - tenant data type 254
- namespace data type 178–198
- namespace defaults
 - modifying 37
 - retrieving 37
- namespaceDefaults
 - data type 198–203
 - resource identifier 37
- namespaceName property 203
- namespace-name* resource identifier 45
- namespacePermission
 - data type 203–205
 - property 144
- namespaceQuota property 255
- namespaces
 - checking existence 45
 - creating 44, 52–53
 - deleting 46
 - filtering 101–102
 - listing in JSON 45
 - listing in XML 45
 - modifying 46
 - optimizedFor 191
 - paging 98–100

namespaces, resource identifier

- resource identifier 44
- resources 44–52
- retrieving individual 45
- retrieving lists of 45
- sorting 100–101
- namespaces property 134
- network property 223
- nfs resource identifier 50
- nfsProtocol data type 206–207

O

- objectCount property 109, 241
- objectsPending property
 - statistics data type for links 235
 - tenant data type for link content 244
- objectsPendingRemote property
 - statistics data type for links 236
 - tenant data type for link content 244
- objectsReplicated property 236
- offset query parameter 99–100
- offsets, specifying 277–278
- operationsPerSecond property 237
- OPTIONS 7
- output formats 8–14
- owner property 190
- ownerType property 190

P

- paging 98–100
- password query parameter
 - tenant creation 104, 263
 - user account creation 104
 - user account modification 104
- passwords
 - MD5 hashed in credentials 28–29
 - resetting security user 70, 104
 - valid values 104
- pause query parameter 247
- pausing tenant replication 247
- percent-encoding special characters in resource URLs 27
- permission property 106
- permissions
 - See also* [data access permission masks](#)
 - listing in JSON 106
 - listing in XML 106
 - property 105–106, 204
 - resource identifier (namespaces) 48, 105–106
 - resource identifier (tenants) 37, 105–106
 - resource value 105–106
 - valid values 38, 49, 204

- point-in-time statistics 107
- port number 23
- POST
 - about 7
 - Content-Type request header 8
 - modifying compliance settings 47
 - modifying content classes 66
 - modifying custom metadata handling setting 47
 - modifying data access permission masks (namespaces) 49
 - modifying data access permission masks (tenants) 38
 - modifying default retention setting 47
 - modifying default shred setting 47
 - modifying disposition setting 47
 - modifying group account permissions 79
 - modifying group accounts 78
 - modifying namespace defaults 37
 - modifying namespaces 46
 - modifying protocol settings 50
 - modifying replication collision handling settings 51
 - modifying replication links 83
 - modifying replication settings 57, 82, 95
 - modifying retention classes 61
 - modifying Search Console configuration 39
 - modifying tenant contact information 36
 - modifying Tenant Management Console configuration 35
 - modifying tenants 34
 - modifying user account permissions 71
 - modifying user accounts 71
 - modifying versioning settings 52
 - performing actions on replication links 83
 - performing actions on replication service 57, 82, 95
 - performing actions on tenants in links 85
 - resetting indexing checkpoints 198
 - resetting security group 78, 154
 - resetting security user passwords 70, 274
- prettyprint query parameter 16
- primaryPhone property 130
- priority property 168
- properties
 - about 6–7, 103
 - availableServicePlan data type 106
 - chargebackReport data type 113
 - cifsProtocol data type 119–120
 - complianceSettings data type 121–129
 - connection data type 124–127
 - contactInfo data type 130–131
 - content data type 132

- contentClass data type 133–134
- contentProperties data type 140
- contentProperty data type 136–139
- customMetadataIndexingSettings data type 142–143
- dataAccessPermissions data type 144
- emailNotification data type 145
- emailTemplate data type 147–148
- failoverSettings data type 150
- groupAccount data type 151–153
- httpProtocol data type 155–159
- ipSettings data type 161–162
- link data type 164–??, 166–171
- local data type (link failoverSettings) 174–176
- local data type (link schedule) 177–178
- namespace data type 179–196
- namespaceDefaults data type 199–202
- namespacePermission data type 203–204
- nfsProtocol data type 206
- permissions 105–106
- protocols data type 207–208
- recipient data type 209–211
- recipients data type 212
- remote data type (link failoverSettings) 214–215
- remote data type (link schedule) 217–218
- replicationCollisionSettings data type 220–221
- replicationService data type 222–224
- retentionClass data type 225–226
- schedule data type 227
- searchSecurity data type 228
- smtpProtocol data type 229–230
- statistics data type (link statistics) 232–239
- statistics data type (tenant and namespace statistics resource) 240–241
- tenant data type 248–262
- tenant data type (link content) 243–246
- transition data type 265–268
- userAccount data type 269–272
- versioningSettings data type 274–276
- protocol settings
 - modifying 50
 - retrieving 50
- protocol-name* resource identifier 50
- protocols
 - data type 207–208
 - resource identifier 50
- prune property 276
- pruneDays property 276
- pruneOnPrimary property 276
- pruneOnReplica property 276

PUT

- about 7
- adding chained links to links 84
- adding directories to links 84
- adding tenants to links 85
- Content-Type request header 8
- creating content classes 65
- creating group accounts 77
- creating namespaces 44
- creating replication links 82
- creating retention classes 60
- creating tenants 33
- creating user accounts 70

Python with PycURL

- about 2
- adding a tenant to a link 91–92
- creating a content class 68
- creating a group account 80
- creating a namespace 53
- creating a replication link 89, ??–96, ??–98
- creating a retention class 62, ??–97, 98–??
- creating a user account ??–58, 58–59, 72–73
- creating an HCP tenant 40
- failing over a link 94
- HTTPHEADER option 9, 24, 29
- modifying a user account 58–??, 74
- modifying compliance settings 54
- modifying the HTTP protocol 56
- modifying the Tenant Management Console configuration 42
- modifying user account permissions 75–76
- retrieving a list of eligible local candidates 90
- retrieving content classes 69
- retrieving group accounts 81
- retrieving retention classes 63, 96
- setting the schedule for a link 93

Q

query parameters

- about 15
- beginRecovery 172
- chargeback report generation 114–116
- completeRecovery 173
- count 99–100
- default tenant creation 265
- end 115
- enterpriseMode 265
- failBack 172
- failOver 172
- filterString 101–102
- filterType 101–102
- forcePasswordChange 263

- granularity 116
- hashScheme 265
- HCP tenant creation 263–265
- initialSecurityGroup 264
- offset 99–100
- password (tenant creation) 104, 263
- password (user account creation) 104
- password (user account modification) 104
- pause 247
- prettyprint 16
- reestablishAllLinks 225
- request specific 16–18
- resetMQECheckpoint 198
- resetPasswords 104, 273–274
- resetSecurityGroup 154
- restarting indexing 198
- restore 173
- resume (links) 172
- resume (tenant replication) 247
- searchEnabled 265
- security group reset 154
- security user password reset 273–274
- servicePlan 265
- shutDownAllLinks 224–225
- sortOrder 100–101
- sortType 100–101
- start 115
- suspend 172
- username 263
- verbose 15, 283

quotation marks with resource URLs 27

R

- readFromReplica property 192
- recipient
 - data type 208–211
 - property 212
- recipients
 - data type 211–213
 - property 145
- reestablishAllLinks query parameter 225
- reestablishing all links 225
- remote
 - data type (failoverSettings) 213–215
 - data type (link schedule) 216–219
 - property 150
- remoteCandidates resource identifier 87
- remoteHost property 126
- remotePort property 127
- removing
 - See also* [deleting](#)
 - chained links from links 84
 - directories from links 84

- tenants from links 85
- replication
 - resource identifier 57, 82, 95
 - resources 82–88, 95–??
- replication collision handling settings
 - modifying 51
 - retrieving 51
- replication links
 - adding chained links to 84
 - adding directories to 84
 - adding tenants to 85, 91–92
 - beginning recovery 172
 - checking existence 83
 - completing recovery 173
 - creating 82, 88–90, 95–??
 - deleting 83
 - failing back 172
 - failing over 94–95, 172
 - listing in JSON 82
 - listing in XML 82
 - listing link content chained links in JSON 84
 - listing link content chained links in XML 84
 - listing link content directories in JSON 83
 - listing link content directories in XML 83
 - listing link content tenants in JSON 85, 132
 - listing link content tenants in XML 85, 132
 - listing link local candidate chained links in JSON 86
 - listing link local candidate chained links in XML 86
 - listing link local candidate directories in JSON 86
 - listing link local candidate directories in XML 86
 - listing link local candidate tenants in JSON 86
 - listing link local candidate tenants in XML 86
 - listing link remote candidate chained links in JSON 87
 - listing link remote candidate chained links in XML 87
 - listing link remote candidate directories in JSON 87
 - listing link remote candidate directories in XML 87
 - listing link remote candidate tenants in JSON 88
 - listing link remote candidate tenants in XML 88
 - modifying 83
 - modifying schedules 88
 - performing actions on 83
 - performing actions on tenants 85

- reestablishing all 225
- removing chained links from 84
- removing directories from 84
- removing tenants from 85
- restoring 173
- resuming 172
- retrieving content chained link lists 84
- retrieving content directory lists 83
- retrieving content lists 83
- retrieving content tenant lists 85
- retrieving eligible local candidates 90–91
- retrieving individual 83
- retrieving lists of 82
- retrieving local candidate chained link lists 86
- retrieving local candidate directory lists 86
- retrieving local candidate lists 85
- retrieving local candidate tenant lists 86
- retrieving remote candidate chained link lists 87
- retrieving remote candidate directory lists 87
- retrieving remote candidate lists 87
- retrieving remote candidate tenant lists 88
- retrieving schedules 88
- retrieving tenant replication status 85
- scheduling activity 92–94
- shutting down all 224–225
- suspending 172
- replication service, performing actions on 57, 82, 95
- replication settings
 - modifying 57, 82, 95
 - retrieving 57, 82, 95
- replicationCollisionSettings
 - data type 219–221
 - resource identifier 51
- replicationConfigurationEnabled property 256
- replicationEnabled property
 - namespace data type 193
 - namespaceDefaults data type 201
- replicationService data type 221–224
- replicationTimestamp property 193
- reporting interval for chargeback reports
 - about 113
 - query parameter 116
- reports, chargeback 112–118
- request body
 - empty with POST 18
 - formats 8–12
 - generating templates for 283
 - JSON 11–12
 - UTF-8 encoding 8
 - XML 9–11
- request headers
 - See [HTTP request headers](#)
- request-specific query parameters 16–18
- requiresAuthentication property 120
- resetMQECheckpoint query parameter 198
- resetPasswords query parameter 104, 273–274
- resetSecurityGroup query parameter 154
- resetting
 - indexing checkpoints 198
 - security group 78, 153–154
 - security user passwords 70, 104, 273–274
- resource identifiers
 - about 31
 - availableServicePlans 34
 - chainedLinks (link content) 84
 - chainedLinks (link local candidates) 86
 - chainedLinks (link remote candidates) 87
 - chargebackReport (namespaces) 47
 - chargebackReport (tenants) 35
 - cifs 50
 - complianceSettings 47
 - consoleSecurity 35
 - contactInfo 36
 - content 83
 - contentClasses 65
 - content-class-name* 65
 - customMetadataIndexingSettings 48
 - dataAccessPermissions 71, 79
 - defaultNamespaceDirectories (link content) 83
 - defaultNamespaceDirectories (link local candidates) 86
 - defaultNamespaceDirectories (link remote candidates) 87
 - directory-name* 84
 - emailNotification 36
 - groupAccounts 77
 - group-name* 78
 - http 50
 - link-name* (link content) 84
 - link-name* (links) 83
 - links 82
 - localCandidates 85
 - namespaceDefaults 37
 - namespace-name* 45
 - namespaces 44
 - nfs 50
 - permissions (namespaces) 48, 105–106
 - permissions (tenants) 37, 105–106
 - protocol-name* 50
 - protocols 50
 - remoteCandidates 87
 - replication 57, 82, 95

- replicationCollisionSettings 51
- retentionClasses 60
 - retention-class-name* 60
- schedule 88
- searchSecurity 39
- service-plan-name* 35
- smtp 50
- statistics (namespaces) 51
- statistics (tenants) 39
- tenant-name* (link content tenants) 85
- tenant-name* (tenants) 33
- tenants 33
- tenants (link content) 85
- tenants (link local candidates) 86
- tenants (link remote candidates) 88
- userAccounts 70
- username* 70
- versioningSettings 52
- resource URLs
 - about 6
 - formats 22–23
 - IP addresses in 23–24, 282
 - length 26
 - percent-encoding special characters 27
 - quotation marks with 27
 - user requirements 22
- resources
 - about 6–7
 - content class 64–66
 - group account 77–79
 - modifying 283
 - namespace 44–52
 - properties for creating 7
 - properties for modifying 7
 - replication 82–88, 95–??
 - retention class 60–61
 - tenant 33–39
 - user account 70–71
- response body
 - CSV 12
 - formats 8–14
 - JSON 11–12
 - UTF-8 encoding 8
 - WADL 13–14
 - XML 9–11
- response headers
 - See [HTTP response headers](#)
- restarting indexing 198
- restEnabled property 157
- restore query parameter 173
- restoring links 173
- restRequiresAuthentication property 158
- resume query parameter
 - links 172
 - tenant replication 247
- resuming
 - links 172
 - tenant replication 247
- retention classes
 - checking existence 61
 - creating 60, 62–63
 - deleting 61
 - listing in JSON 60
 - listing in XML 60
 - modifying 61
 - resources 60–61
 - as retention settings 278–279
 - retrieving individual 60
 - retrieving lists of 60, 63–64
- retention settings
 - fixed dates 279
 - offsets 277–278
 - retention classes 278–279
 - special values 277
- retentionClass data type 225–227
- retentionClasses resource identifier 60
- retention-class-name* resource identifier 60
- retentionDefault property
 - about 122
 - valid values 277–279
- retrieving
 - available service plan lists 34
 - available service plans, individual 35
 - compliance settings 47
 - content class lists 65, 68–69
 - content classes, individual 65
 - custom metadata handling setting 47
 - custom metadata indexing settings for
 - metadata query engine 48
 - data access permission masks
 - (namespaces) 48
 - data access permission masks (tenants) 37
 - default retention setting 47
 - default shred setting 47
 - disposition setting 47
 - eligible local candidates 90–91
 - group account lists 78, 81–82
 - group account permissions 79
 - group accounts, individual 78
 - link content chained link lists 84
 - link content directory lists 83
 - link content lists 83
 - link content tenant lists 85
 - link local candidate chained link lists 86
 - link local candidate directory lists 86
 - link local candidate lists 85

- link local candidate tenant lists 86
- link remote candidate chained link lists 87
- link remote candidate directory lists 87
- link remote candidate lists 87
- link remote candidate tenant lists 88
- link schedules 88
- namespace defaults 37
- namespace lists 45
- namespaces, individual 45
- protocol settings 50
- replication collision handling settings 51
- replication link lists 82
- replication links, individual 83
- replication settings 57, 82, 95
- retention class lists 60, 63–64
- retention classes, individual 60
- Search Console configuration 39
- statistics (namespaces) 51
- statistics (tenants) 36, 39
- tenant contact information 36
- tenant lists 33
- Tenant Management Console
 - configuration 35
- tenant replication status 85
- tenants, individual 33
- user account lists 70
- user account permissions 71
- user accounts, individual 70
- versioning settings 52
- return codes 286–288
- role property
 - group accounts 153
 - user accounts 271
- roles
 - changing for user accounts 58–59, 73–74
 - property (groupAccount data type) 153
 - property (userAccount data type) 271

S

- sample application
 - about 290
 - assumptions 290–291
 - code 299–309
 - JSON input files 291–298
 - required libraries 291
- schedule
 - data type 227–228
 - resource identifier 88
- scheduleOverride property
 - link local schedule 177
 - link remote schedule 218
- scheduling activity on links 92–94
- schema, XML 311

- Search Console configuration
 - modifying 39
 - retrieving 39
- searchConfigurationEnabled property 257
- searchEnabled
 - property (namespace data type) 194
 - property (namespaceDefaults data type) 202
 - query parameter 265
- searchSecurity
 - data type 228–229
 - resource identifier 39
- security group, resetting 78, 153–154
- security user passwords, resetting 70, 104, 273–274
- self-signed SSL certificates 23
- separateAttachments property 230
- servicePlan
 - property (namespace data type) 194
 - property (namespaceDefaults data type) 202
 - property (tenant data type) 258
 - query parameter 265
- service-plan-name* resource identifier 35
- servicePlanSelectionEnabled property 259
- serviceRemoteSystemRequests property 195
- severity property 210
- shredCount property 241
- shreddingDefault property 122
- shredSize property 241
- shutDownAllLinks query parameter 224–225
- shutting down all links 224–225
- smtp resource identifier 50
- smtpProtocol data type 229–231
- snmpLoggingEnabled property 259
- softQuota property
 - namespace data type 195
 - namespaceDefaults data type 202
 - tenant data type 260
- sorting 100–101
- sortOrder query parameter 100–101
- sortType query parameter 100–101
- special characters in resource URLs, percent-encoding 27
- special values (retention settings) 277
- specifying retention values 277–279
- SSL security 23
- SSL_VERIFYPEER option (Python with PycURL) 23
- start query parameter 115
- state property 131
- statistics
 - See also [chargeback reports](#)
 - data type (links) 231–240

statistics, data type (tenants and namespaces)

- data type (tenants and namespaces) 240–242
- dynamic 107
- point in time 107
- property 168
- resource identifier (namespaces) 51
- resource identifier (tenants) 39
- retrieving (namespaces) 51
- retrieving (tenants) 36, 39
- status codes 286–288
- status property
 - link data type 169
 - replicationService data type 224
 - tenant data type for link content 245
- statusMessage property 170
- storageCapacityUsed property 109, 241
- subject property 148
- suspend query parameter 172
- suspended property 171
- suspending links 172
- syslogLoggingEnabled property 260
- System Management Console, enabling the management API 19
- system-level administrative access to tenants 5, 32, 103, 248
- systemVisibleDescription property
 - about 260
 - valid values 105

T

- tag property
 - filtering 101
 - namespaces 196
 - tenants 261
- tags property
 - namespace data type 196
 - tenant data type 261
- templates, generating for resource creation 283
- tenant data type
 - link content tenant resource 242–246
 - tenant resource 247–263
- Tenant Management Console configuration
 - See also* [consoleSecurity](#)
 - modifying 35, 41–42
 - retrieving 35
- Tenant Management Console, enabling the management API 19
- tenant property 132
- tenant-name* resource identifier
 - link content tenants resource 85

- tenants resource 33
- tenants
 - adding to links 85
 - checking existence 34
 - creating 33, 263–265
 - deleting 34
 - filtering 101–102
 - listing in JSON 33
 - listing in XML 33
 - modifying 34
 - paging 98–100
 - pausing replication 247
 - removing from links 85
 - resource identifier 33
 - resources 33–39
 - resuming replication 247
 - retrieving individual 33
 - retrieving lists of 33
 - retrieving replication status 85
 - sorting 100–101
 - system-level administrative access to 5, 32, 103
- tenants resource identifier
 - link content 85
 - link local candidates 86
 - link remote candidates 88
- tenantVisibleDescription property
 - about 261
 - valid values 105
- text/csv Internet media type 8
- time period covered by chargeback reports
 - about 113
 - query parameters 115–116
- totalBytesRead property 110
- totalBytesWritten property 110
- totalDeletes property 110
- totalReads property 110
- totalWrites property 110
- transition property
 - link local schedule 178
 - link remote schedule 218
- type property
 - contentProperty data type 139
 - link data type 171
 - recipient data type 211

U

- uid property 206
- unauthenticatedMinimumPermissions
 - property 105–106

- Unix hosts file 24
- upToDateAsOfMillis property
 - statistics data type for links 238
 - tenant data type for link content 245
- upToDateAsOfString property
 - statistics data type for links 239
 - tenant data type for link content 246
- URLs
 - See [resource URLs](#)
- user accounts
 - checking existence 70
 - creating ??–58, 70, 72–73
 - creating initial 263–264
 - deleting 71
 - filtering 101–102
 - for HCP management API access 4
 - listing in JSON 70
 - listing in XML 70
 - modifying 58–59, 71, 73–74
 - modifying data access permissions 71, 75–76
 - paging 98–100
 - resetting security passwords 70
 - resetting security user passwords 273–274
 - resources 70–71
 - retrieving data access permissions 71
 - retrieving individual 70
 - retrieving lists of 70
 - sorting 100–101
- user requirements
 - management API 4–6
 - URL formats 22
- user roles
 - See [roles](#)
- userAccount data type 268–273
- userAccounts resource identifier 70
- userGUID property 271
- userID property 272
- username
 - property 272
 - property, filtering 101
 - property, sorting 100
 - query parameter 263
- username* resource identifier 70
- usernames, Base64 encoded in credentials 28–29
- UTF-8 encoding 8

V

- valid property 111
- value property
 - about 226
 - valid values 277–278

- variables, email message template 148–149
- verbose query parameter 15, 283
- versioning settings
 - modifying 52
 - retrieving 52
- versioningConfigurationEnabled property 262
- versioningSettings
 - data type 274–277
 - property (namespace data type) 196
 - resource identifier 52
- versioningSettings property 202

W

- WADL
 - about 8
 - response body contents 13–14
- WebDAV protocol 50
- webdavBasicAuthEnabled property 158
- webdavBasicAuthPassword property 104, 158
- webdavBasicAuthUsername property 159
- webdavCustomMetadata property 159
- webdavEnabled property 159
- Windows hosts file 24

X

- X-HCP-ErrorMessage response header 18
- X-HCP-SoftwareVersion response header 18
- XML

- allowAddresses list 161
- authenticationType list 249
- available service plan list 34
- content class list 65, 142
- denyAddresses list 162
- empty request body 18
- group account list 78
- Internet media type for 8
- link content chained link list 84, 132
- link content directory list 83, 132
- link content tenant list 85, 132
- link list 82
- link local candidate chained link list 86
- link local candidate directory list 86
- link local candidate tenant list 86
- link remote candidate chained link list 87
- link remote candidate directory list 87
- link remote candidate tenant list 88
- namespace list (contentClass data type) 134
- namespace list (namespaces resource) 45
- permission list 106
- request body contents 9–11
- response body contents 9–11
- retention class list 60

XML, role list

role list 142, 153, 271

schema 311

tag list 196, 261

tenant list 33

user account list 70

Z

zipOrPostalCode property 131

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-90ARC029-11