Working with system metadata

You can perform these operations on the system metadata associated with objects:

• Specifying metadata on object creation

• Modifying object metadata

• Checking the existence of an object or multiple versions of an object

This chapter describes how to specify and modify system metadata. For information about listing the metadata for an object, see Checking the existence of an object or multiple versions of an object. For more information about metadata, see Object properties.

Note: When converting legacy applications to use the RESTful HTTP interface to access an HCP namespace, be aware that the values used to specify certain metadata settings differ from those used for metadata settings in the default namespace and HCAP 2.x archives. For information about the default namespace, see Using the Default Namespace.

Specifying metadata on object creation

You use the HTTP PUT method to store objects in the namespace. When storing an object, you can override the namespace default values for these metadata properties:

• Hold

• Index setting

• Retention setting
• Shred setting

You can also store one of two predefined ACLs for an object when storing an object:

• **all_read** — Grants read permission for the object to all users

• **auth_read** — Grants read permission for the object to all authenticated users

To specify metadata when storing an object, you use URL query parameters.

**Request contents (PUT object and metadata)**

You are here:

The PUT request to store an object and override the default values for its metadata includes these elements:

• If you're accessing the namespaces as an authenticated user, an Authorization header

• A URL specifying the location in which to store the object

• A body containing the fixed-content data to be stored in the namespace

• Any combination of the query parameters described in the table below

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>One of:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>all_read</strong> — Grants read permission for the object to all users</td>
<td>Stores a predefined ACL for the object</td>
</tr>
<tr>
<td></td>
<td>• <strong>auth_read</strong> — Grants read permission for the object to all authenticated users</td>
<td></td>
</tr>
<tr>
<td>hold</td>
<td>true or false</td>
<td>Paces an object on hold or specifies that it's not on hold</td>
</tr>
<tr>
<td>index</td>
<td>true or false</td>
<td>Specifies whether the object should be indexed for search</td>
</tr>
<tr>
<td>Parameter</td>
<td>Valid Values</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>retention</td>
<td>Any valid retention expression, as described in Specifying retention settings</td>
<td>Specifies the retention setting for the object</td>
</tr>
<tr>
<td>shred</td>
<td>true or false</td>
<td>Specifies whether to shred the object after it is deleted</td>
</tr>
</tbody>
</table>

For more information about storing an object, including additional parameters that you can use, see Storing an object or version of an object

Trademarks and Legal Disclaimer

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.

Access permission (PUT object and metadata)

You are here:

To specify metadata in a request to store an object, you need write permission. Additionally, to specify the hold parameter, you also need privileged permission.

Trademarks and Legal Disclaimer

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.

Request-specific return code (PUT object and metadata)

You are here:

The table below describes the HTTP return code that has specific meaning for a PUT request that includes metadata query parameters. For descriptions of all possible return codes, see HTTP return codes.

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Bad Request</td>
<td>One of: •The URL in the request is not well-formed.</td>
</tr>
<tr>
<td>Code</td>
<td>Meaning</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The request contains an unsupported query parameter or an invalid value for a query parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The request is trying to store an object with a predefined ACL in a namespace that does not support ACLs.</td>
</tr>
</tbody>
</table>

**Request-specific response headers (PUT object and metadata)**

You are here:

The table below describes the request-specific HTTP response headers for this operation. For information about all HCP-specific response headers, see [HCP-specific HTTP response headers](https://knowledge.hitachivantara.com/Documents/Storage/Content_Platform/8.2/Developer_documentation/Using_a_namespace/HTTP/04_Working_with_system_metadata).

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
</table>
| ETag                          | The ETag of the stored object data, enclosed in double quotation marks (").
| X-HCP-CustomMetadataHash      | The cryptographic hash algorithm HCP uses and the cryptographic hash value of the stored custom metadata, in this format:  
X-HCP-CustomMetadataHash: hash-algorithm hash-value  
You can use the returned hash value to verify that the stored custom metadata is the same as the metadata you sent. To do this, compare this value with a hash value that you generate from the original custom metadata. |
### Header Description

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-HCP-Hash</td>
<td>The cryptographic hash algorithm HCP uses and the cryptographic hash value of the stored object, in this format: X-HCP-Hash: hash-algorithm\hash-value You can use the returned hash value to verify that the stored data is the same as the data you sent. To do so, compare this value with a hash value that you generate from the original data.</td>
</tr>
<tr>
<td>X-HCP-VersionId</td>
<td>The version ID of the object or version.</td>
</tr>
</tbody>
</table>

**Example: Setting metadata when storing an object**

You are here:

Here's a sample HTTP PUT request that stores a file named Q1_2012.ppt in the quarterly_rpts directory, sets the retention value for the object to the Reg-107 class, and sets the shred setting for the object to true.

**Request with curl command line**

curl -k -iT Q1_2012.ppt
-H "Authorization: HCP bXl1c2Vy:3f3c6784e975317774380db177774ac8d"
"https://finance.europe.hcp.example.c...quarterly_rpts/
Q1_2012.ppt?retention=C+Reg-107&shred=true"

**Request in Python using PycURL**

```python
import pycurl
import os
filehandle = open("Q1_2012.ppt", 'rb')
curl = pycurl.Curl()
```
curl.setopt(pycurl.HTTPHEADER, ["Authorization: HCP bXl1c2Vy:3f3c6784e975317774380db177774ac8d")
curl.setopt(pycurl.URL, "https://finance.europe.hcp.example.com\rest/quarterly_rpts/Q1_2012.ppt?retention=C+Reg-107")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.UPLOAD, 1)
curl.setopt(pycurl.INFILESIZE, os.path.getsize("Q1_2012.ppt"))
curl.setopt(pycurl.READFUNCTION, filehandle.read)
curl.perform()
print curl.getinfo(pycurl.RESPONSE_CODE)
curl.close()
filehandle.close()

Request headers

PUT /rest/quarterly_rpts/Q1_2012.ppt?retention=C+Reg-107 HTTP/1.1
Host: finance.europe.hcp.example.com
Authorization: HCP bXl1c2Vy:3f3c6784e975317774380db177774ac8d
Content-Length: 678400

Response headers

HTTP/1.1 201 Created
X-HCP-ServicedBySystem: hcp.example.com
ETag: "9c604138ffb0f308a8552a3752e5a1be"
Location: /rest/quarterly_rpts/Q1_2012.ppt
X-HCP-VersionId: 80238663473089
X-HCP-Hash: SHA-256 E830B86212A66A792A79D58BB185EE63A4FADA76BB8A1C...
X-HCP-Time: 1334829227
Content-Length: 0

Modifying object metadata

You are here:

You use the HTTP POST method to change the metadata for existing objects in a namespace. The request changes only the metadata that it explicitly specifies. All other metadata remains unchanged.
You can modify metadata only for the current version of an object. You cannot modify metadata for an old version.

If the request to change metadata specifies a symbolic link, HCP changes the metadata on the object that’s the target of the link.

**Trademarks and Legal Disclaimer**

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.

---

### Request contents (PUT object and modify metadata)

You are here:

The POST request to modify object metadata has these elements:

- If you’re accessing the namespace as an authenticated user, an Authorization header
- A URL specifying the object for which you’re changing the metadata
- A body containing form-encoded metadata (application/x-www-form-urlencoded content type). The content can consist of any combination of the parameters described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hold</td>
<td>true or false</td>
<td>Places an object on hold or specifies that it’s not on hold.</td>
</tr>
<tr>
<td>index</td>
<td>true or false</td>
<td>Specifies whether the object should be indexed for search.</td>
</tr>
<tr>
<td>retention</td>
<td>Any valid retention expression, as described in <a href="https://knowledge.hitachivantara.com/Documents/Storage/Content_Platform/8.2/Developer_documentation/Using_a_namespace">Specifying retention settings</a></td>
<td>Specifies the retention setting for the object.</td>
</tr>
<tr>
<td>shred</td>
<td>true or false</td>
<td>Specifies whether to shred the object after it is deleted.</td>
</tr>
<tr>
<td>owner</td>
<td>One of:</td>
<td>Specifies the user that owns the object.</td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
</table>
|           | •The name of a user account that's defined in HCP  
           | •The name of an AD user account that HCP recognizes. This can be either the user principal name or the Security Accounts Manager (SAM) account name for the AD user account.  
           | •An empty string | If you specify an AD user account, you also need to specify the domain parameter.  
           |               | Specifying an empty string causes the object to have no owner. |
| domain    | An Active Directory domain | Specifies the Active Directory domain that contains the user account specified by the owner query parameter. |

**Trademarks and Legal Disclaimer**

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.

**Request considerations (PUT object and modify metadata)**

You are here:

These considerations apply to requests to change object metadata:

•You cannot change the hold and retention settings in the same request.

•You cannot change the shred setting from true to false.

**Trademarks and Legal Disclaimer**

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.

**Access permission (PUT object and modify metadata)**

You are here:

To modify object metadata, you need write permission.
To specify a hold parameter, you also need privileged permission for the namespace.

To specify an owner parameter, you need change owner permission for the namespace.

Request-specific return codes (PUT object and modify metadata)

You are here:

The table below describes the HTTP return codes that have specific meaning for this request. For descriptions of all possible return codes, see [HTTP return codes](#).

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
<td>HCP successfully updated the metadata.</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
<td>One of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The URL in the request is not well-formed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The request is trying to change the retention setting from a retention class to an explicit setting, such as a datetime value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The request is trying to change the retention setting and the retention hold setting at the same time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The request is trying to change the retention setting for an object on hold.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The request is trying to change the shred setting from true to false.</td>
</tr>
<tr>
<td>Code</td>
<td>Meaning</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
<td>One of: &lt;br&gt;• The Authorization header or hcp-ns-auth cookie specifies invalid credentials. &lt;br&gt;• The namespace requires client authentication, and the request does not have an Authorization header or hcp-ns-auth cookie. &lt;br&gt;• The user doesn’t have write permission. &lt;br&gt;• For a request to hold or release an object, the user doesn’t have privileged permission.</td>
</tr>
</tbody>
</table>
### Code and Meaning

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>404</td>
<td>Not Found</td>
<td>HCP could not find the specified object, or the URL path contains a symbolic link to a directory anywhere other than in the last component.</td>
</tr>
</tbody>
</table>

For a request to change the owner of an object, the user doesn't have change owner permission.

The access method (HTTP or HTTPS) is disabled.

---

**Trademarks and Legal Disclaimer**

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.

---

**Request-specific response headers (PUT object and modify metadata)**

You are here:

This operation does not return any request-specific response headers. For information about all HCP-specific response headers, see [HCP-specific HTTP response headers](https://knowledge.hitachivantara.com/Documents/Storage/Content_Platform/8.2/Developer_documentation/Using_a_namespace/HTTP/04_Working_with_system_metadata).

---

**Example: Changing multiple metadata values for an existing object**

You are here:

Here’s a sample HTTP POST request that makes these changes to the metadata for the Q1_2012.ppt file:

- Increases retention by one year from the current value
- Turns on shredding on delete
- Turns off indexing

*Request with curl command line*
curl -k -i -H "Authorization: HCP bXl1c2Vy:3f3c6784e97531774380db177774ac8d"
-d "retention=R+1y" -d "shred=true" -d "index=false"
"https://finance.europe.hcp.example.com/rest/quarterly_rpts/Q1_2012.ppt"

Request in Python using PycURL

import pycurl
curl = pycurl.Curl()
theHeader = ["Transfer-Encoding: chunked"]
curl.setopt(pycurl.HTTPHEADER, ["Authorization: HCP
bXl1c2Vy:3f3c6784e97531774380db177774ac8d", theHeader])
curl.setopt(pycurl.URL, "https://finance.europe.hcp.example.com
/rest/quarterly_rpts/Q1_2012.ppt")
curl.setopt(pycurl.SSL_VERIFYPEER, 0)
curl.setopt(pycurl.SSL_VERIFYHOST, 0)
curl.setopt(pycurl.POST, 1)
theFields = "retention=R+1y&shred=true&index=false"
theHeader = ["Transfer-Encoding: chunked"]
curl.setopt(pycurl.POSTFIELDS, theFields)

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

Request headers

POST /rest/quarterly_rpts/Q1_2012.ppt HTTP/1.1
Host: finance.europe.hcp.example.com
Authorization: HCP bXl1c2Vy:3f3c6784e97531774380db177774ac8d
Content-Length: 37
Content-Type: application/x-www-form-urlencoded

Response headers

HTTP/1.1 200 OK
X-HCP-ServicedBySystem: hcp.example.com
Content-Length: 0

Trademarks and Legal Disclaimer

© 2015, 2019 Hitachi Vantara Corporation. All rights reserved.