IBM Cloud Object Storage System™
Version 3.14.3

Simple Object HTTP API 2.5 Developer Guide

IBM

This edition applies to IBM Cloud Object Storage System and is valid until replaced by new editions.

Contents

Document Information

Intended Purpose and Audience

This reference describes the methods, requests and responses for the IBM Cloud Object Storage System™ Dispersed Storage Network Simple Object HTTP API.

Chapter 1. Introduction

Summary

The Simple Object HTTP application programming interface (SOH API) enables application developers to develop REST-based applications to access object vaults on an IBM Cloud Object Storage System™. It is a REST API available as part of the Dispersed Storage Object (DSO) API Family.

Table 1. Dispersed Object API List

REST API	Write Mode	Listings	Store User Metadata	Compatibility
Simple Object HTTP	Write-by-ID	Recovery Only		
Cloud Storage Object	Write-by-Name	Prefix-based Name Search and Recovery Listing	Yes	Amazon S3 v2006-03-01
OpenStack Object Storage	Write-by-Name	Prefix-based Name Search and Recovery Listing	Yes	OpenStack v1.0

Like any REST API in the Dispersed Storage Object API Family, the SOH API is automatically deployed on any IBM Cloud Object Storage Accesser® appliance. Any object vault that is deployed to an Accesser appliance can be access through any API in the API Family.

What's new in the 2.5 API

All HTTP responses now include a header that is called **X-Clv-Request-Id**. The header includes an identifier that is unique for each response and can be used by a Cleversafe Support Engineer for diagnostics and troubleshooting purposes.

Deployment and Client Configuration

When an object vault is deployed to an IBM Cloud Object Storage Accesser®appliance, it is automatically made available through each of the REST APIs in the Dispersed Storage Object API Family.

See the *Manager Administration Guide* for more information on deploying an Object Vault.

Clients can always access the SOH API via the following URL scheme (new in version 2.5):

```
http://{accesser-ip}/soh/{vault-name}/{object-id}
```

Where:

Table 2. URL scheme

URL Part	Description
{accesser-ip}	The IP address or host name of the Accesser appliance. The actual address can differ for client applications if a proxy or load balancer is being used.
soh	The literal that indicates the SOH API.
{vault-name}	The vault name that is being accessed. When omitted, a list of vaults is returned.
{object-id}	The identifier of the object that was written, and is specified for operations on existing objects. It is omitted when new objects are written.

If the SOH API is configured as the Default API for an Accesser appliance, clients can also access the API via the following URL scheme:

```
http://{accesser-ip}/{vault-name}/{object-name}
```

In ClevOS 2.8.0, the default API is initially set to SOH API. When ClevOS is upgraded, existing client applications that support the SOH API do not need to be modified unless an administrator changes the default API on a given Accesser appliance.

Access through HTTPS is also supported.

Note: See the *Manager Administration Guide* for more information about SSL and PKI access through an Accesser appliance.

Chapter 2. API reference

API REST URI format

Details of the SOH API commands.

An overview of the command set is shown in the table.

Table 3. SOH API REST URI Formats

Operation	URL Form	Method	Headers	Output
Vault Listing	/	GET		JSON Document
Vault Status	/vault-name	GET		JSON Document
Write Object	/vault-name	PUT	X-Digest	Object Identifier
Read Object	/vault-name/object-id	GET		Source Data
Delete Object	/vault-name/object-id	DELETE		
Object Status	/vault-name/object-id	HEAD		
List	/vault-name	GET	X-Operation, X-Start-Id, X-List-Length-Limit	Object Identifiers

The URL forms that are documented in the table are relative to the SOH API base URL. These commands are always relative to `http://{accesser-ip}/soh/`. When the default API is configured to SOH API, they are also available relative to `http://{accesser-ip}/`.

Note: For more information, see "Deployment and Client Configuration" on page 1.

API version

The SOH API version can be programmatically determined in two ways.

- The JSON document that is returned by the vault listing or vault status query indicates the protocol name and version.
- Each response that is returned by the server includes the header **`X-SOH-Version`**. The API version is set as the value of this header.

The API version was exposed starting with SOH API 2.3. The version header is absent for the SOH API versions 2.2 or earlier. If a caller is parsing the API version to check for capability, use the following procedure:

- Parse the version number into a tuple of X.Y form.
- If X is not equal to the expected major version number, assume no compatibility and cancel.
- If Y is less than the expected version minor version number, assume that the feature is not present and either cancel or fall back to alternative behavior.

Request ID

All HTTP responses include a header that is called **`X-Clv-Request-Id`**, which includes an identifier that is unique for each response. It can be used by a Cleversafe Support Engineer for diagnostics and troubleshooting purposes. This request ID is logged internally on Accesser appliances on a per-request basis. The purpose of the value is to correlate failures that are presented to a client application with a request that was processed on the server. If client applications encounter a failure during an operation, they can log or output the request ID for use by users.

Authentication

The SOH API allows authentication by using a user name and password or an X.509 certificate and private key pair.

Basic access authentication (RFC 1945) must be used for user name and password authentication. This password is transmitted to the access server, which in turn sends the password to authenticate on behalf of the user to each slice server. If SSL is used, the plain text password is not exposed over the wire. A user who has an X.509 certificate and matching private key can perform Public Key Infrastructure (PKI) authentication if the certificate is assigned to a user account on the storage network. The System Manager can be used to assign certificates to user accounts. The certificate and private key can then be used to perform SSL client certificate authentication (RFC 2246).

Vault listing

The **GET** method is used to list all of the vaults that are deployed to a specific Accesser appliance. When a vault name is not specified as part of the URL, a JSON document that contains basic protocol name and version information and a vault listing is returned. The **HEAD** method can also be used to obtain API version information without returning a vault listing.

Sample request

```
GET / HTTP/1.1
Host: access.example.com
```

Requests protocol and vault listing.

Sample response

```
HTTP/1.1 200 OK
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: 40c5252a-716b-4a96-ac51-af0a5aa0893d
Content-Length: 316
```

```
{
 "interface_name": "hydroxide",
 "interface_version": "2.5",
 "vaults": [
 {
 "vault_identifier": "7bb87f5b-72ce-4ec5-a487-363b10000716",
 "vault_name": "vault-name"
 },
 {
 "vault_identifier": "06a2beb3-8d3c-4cc0-9a28-acaca84d7f2c",
 "vault_name": "other-vault-name"
 }
 ]
}
```

The response indicates that it is SOH API v2.5 and that two vaults exist, one named vault-name and another named other-vault-name.

Status codes

Table 4. Vault listing - status codes

Code	Status	Description
200	OK	Vault listing query was successful.
400	Bad Request	Request could not be understood by the access server due to malformed syntax.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to write to the indicated vault.
403	Forbidden	Client has insufficient privileges to perform the operation.
500	Internal Server Error	Internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

Response fields

Table 5. Vault listing - response fields

Field	Description
interface_name	Interface name. Callers can use this value to ensure that they are connecting to the correct location.
interface_version	The interface version. Callers can use this value to ensure that the server is exposing the appropriate level of functions.
vaults/vault_identifier	The universally unique identifier (UUID) of the vault this listing describes.
vaults/vault_name	The name of the vault this listing describes.

Vault status query

The **GET** method is used to query overall vault status and obtain API version information. When an object identifier is not specified as part of the URL, a JSON document that contains basic protocol name and version information and vault capacity statistics is returned. The **HEAD** method can also be used to obtain API version information without returning vault status information.

Sample request

```
GET /vault-name HTTP/1.1
Host: access.example.com
```

Requests protocol and vault status information for the vault name is vault-name.

Sample response

```
HTTP/1.1 200 OK
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: 18161859-a867-4779-9399-87cfce838496
Content-Length: 348
```

```
{
 "interface_name": "hydroxide",
 "interface_version": "2.5",
 "vault_identifier": "8646e9d4-5ee7-4345-a908-0bb8989fd22f",
 "vault_name": "vault-name",
 "vault_usage": {
 "total_size": "239681228800",
 "quota_size": "214748364800",
 "used_size": "210517893120",
 "free_size": "28901191680",
 "usable_size": "4230471680"
 }
}
```

The response indicates that it is SOH API version 2.5 and that the vault named vault-name. The vault contains about 196 GiB of data and 27 GiB of free space. The vault has a hard quota of exactly 200 GiB, which means that about 4 GiB of the total available free space can be used.

Status codes

Table 6. Vault status query - status codes

Code	Status	Description
200	OK	The vault status query was successful.
400	Bad Request	The request could not be understood by the access server due to malformed syntax.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to write to the indicated vault.
403	Forbidden	The client has insufficient privileges to perform the operation.
500	Internal Server Error	An internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

Response fields

Table 7. Vault status query - response fields

Field	Description
interface_name	Interface name. Callers can use this value to ensure that they are connecting to the correct location.
interface_version	Interface version. Callers can use this value to ensure that the server is exposing the appropriate level of functions.
vault_identifier	Universally unique identifier (UUID) of the vault accessible at this URL.
vault_name	Name of the vault accessible at this URL.
vault_usage/total_size	Total number of bytes that could possibly be allocated to this vault.
vault_usage/quota_size	Total number of bytes that can be allocated to this vault when accounting for any configured hard quota.
vault_usage/used_size	Estimated total number of bytes stored in this vault.
vault_usage/free_size	Number of bytes that could be added to this vault.
vault_usage/usable_size	Number of bytes that can be added to this vault when accounting for any configured hard quota.

Write object

The **PUT** method is used to add an object to a vault. An object identifier is generated before the object is stored and returned after the operation is complete. The application should treat the object identifier as an opaque string. An object cannot be overwritten after it is created. Instead, an application must write any new data to a new object and delete the old object. Data that is written to a vault is visible after a response is returned from the write operation. Read and delete operations do not succeed until a write returns.

Sample request

```
PUT /vault-name HTTP/1.1
Content-Length: 43
Host: access.example.com
Content-Type: Text/Plain
The quick brown fox jumps over the lazy dog
```

An object that contains the text **The quick brown fox jumps over the lazy dog** is written to a vault named vault-name by using the access server `access.example.com`.

Sample response

```
HTTP/1.1 201 Created
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: 0f136d74-2ed6-4b6b-9f84-91735574aa5b
Content-Length: 37
d6133cd067ad4ea38cdcd1faa1e4e9040000
```

The response contains the object identifier of the data that was written to the Dispersed Storage network.

Status codes

Table 8. Write object - status codes

Code	Status	Description
201	Created	The object was successfully written to the vault.
400	Bad Request	The request could not be understood by the access server due to malformed syntax.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to write to the indicated vault.
403	Forbidden	The client has insufficient privileges to perform the operation.
500	Internal Server Error	An internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

Content digest

When an object is written, the client can request that the access server calculate a digest on the written content with the **X-Digest** request header. This digest is returned as a hex-encoded value with the **X-Content-Digest** response header. The purpose of content digest calculation is to allow clients to verify that the correct information was written across the wire. The calculated digest is not stored with the data. Data at rest integrity is ensured by the integrity codecs that are configured for the vault. Currently, three supported digest algorithms are available. Digest algorithm specifications are not case-sensitive.

Supported digest algorithms

- md5
- sha1
- sha512

Sample request

```
PUT /vault-name HTTP/1.1
X-SOH-Version: 2.5
Content-Length: 43
Host: access.example.com
Content-Type: Text/Plain
X-Digest: md5
The quick brown fox jumps over the lazy dog
```

Sample response

```
HTTP/1.1 201 Created
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: 3d1fbf9b-1ec0-4e8b-b284-34d89d530763
X-Content-Digest: 9e107d9d372bb6826bd81d3542a419d6
Content-Length: 37
d6133cd067ad4ea38cdcd1faa1e4e9040000
```

Read object

The **GET** method retrieves a previously stored object by using the object identifier. The URL consists of the vault name followed by the object identifier. An object is not available until it is written. Partially completed objects cannot be accessed with the read operation.

Sample request

```
GET /vault-name/d6133cd067ad4ea38cdcd1faa1e4e9040000 HTTP/1.1
```

An object is retrieved from a vault that is named vault-name by using the access server `access.example.com`.

Sample response

```
HTTP/1.1 200 OK
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: dcaf63bf-2379-4934-b986-349ade203257
ETag: d6133cd067ad4ea38cdcd1faa1e4e9040000
Content-Length: 43
The quick brown fox jumps over the lazy dog
```

The response contains the source data of the retrieved object. The **ETag** is equal to the object ID returned in the write request.

Status codes

Table 9. Read object - status codes

Code	Status	Description
200	OK	Object was read successfully from the vault.
206	Partial Content	Object was read successfully from the vault and a portion was returned.
400	Bad Request	URL contained an invalid object identifier or was otherwise malformed.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to read from the indicated vault.
403	Forbidden	Client has insufficient privileges to perform the operation.
404	Not Found	No such object exists in the vault.
416	Requested Range Not Satisfiable	Requested range is greater than available from the object.
500	Internal Server Error	Internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

Range requests

HTTP Range function is defined in sections 14.16, 14.35, and 19.2 of Hypertext Transfer Protocol -- HTTP/1.1 [RFC 2616]. A certain portion of an object can be requested by using the Range header. When an access server receives a request with a Range header, it replies with a proper **`206 Partial Content`** or **`416 Requested Range Not Satisfiable`** statuses, depending on whether the requested range is satisfiable. If an access server receives a syntactically invalid range request, the Range header is ignored and an ordinary full-entity response is returned. It conforms to Hypertext Transfer Protocol -- HTTP/1.1 [RFC 2616 section 14.35.1]. Multi-range requests, as described in Hypertext Transfer Protocol -- HTTP/1.1 [RFC 2616 section 19.2], are not supported. When a multi-range request is received, the server treats the request as a syntactically invalid range request and returns an ordinary full-entity response.

Delete object

The **`DELETE`** method removes a previously stored object by using the object identifier. The URL consists of the vault name followed by the object identifier. This operation reports success if the object was deleted from the system.

If the object was already deleted or never existed, an error is returned. The HTTP API does not allow a delete operation to be performed on an object that is still being written. It is because an object identifier is not provided to the caller until the write operation is finished. If a delete operation is performed on an object that is being read, the read might be interrupted. While the HTTP protocol does not allow a specific error message to indicate when it occurs, a second attempt to read the same object results in a **`404 Not Found`** error.

Sample request

```
DELETE /vault-name/d6133cd067ad4ea38cdcd1faa1e4e9040000 HTTP/1.1
Host: access.example.com
```

An object is removed from a vault that is named vault-name by using the access server `access.example.com`.

Sample response

```
HTTP/1.1 204 No Content
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: a3378bc8-ab5b-4603-be35-751a7331fd31
```

The response indicates that the object is removed.

Status codes

Table 10. Delete object - status codes

Code	Status	Description
204	No Content	The object was successfully removed from the vault.
400	Bad Request	The request could not be understood by the access server due to malformed syntax.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to read from the indicated vault.
403	Forbidden	The client has insufficient privileges to perform the operation.
404	Not Found	No such object exists in the vault.
500	Internal Server Error	An internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

Object status

The **HEAD** method returns the status of a previously stored object by using the object identifier. The URL consists of the vault name followed by the object identifier. An object is not available until it is written. Partially completed objects cannot be accessed with the read operation.

Sample request

```
HEAD /vault-name/d6133cd067ad4ea38cdcd1faa1e4e9040000 HTTP/1.1
Host: access.example.com
```

An object is retrieved from a vault that is named vault-name by using the access server **access.example.com**.

Sample response

```
HTTP/1.1 200 OK
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: b1a4edcf-20ce-4eae-b72c-0b8fd2a815b8
ETag: d6133cd067ad4ea38cdcd1faa1e4e9040000
Content-Length: 43
```

The response indicates that the object exists and how many bytes it contains.

Status codes

Table 11. Object status - status codes

Code	Status	Description
200	OK	Object status query was successful.
400	Bad Request	URL contained an invalid object identifier or was otherwise malformed.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to read from the indicated vault.
403	Forbidden	Client has insufficient privileges to perform the operation.
404	Not Found	No such object exists in the vault.
500	Internal Server Error	Internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

List

The **`GET`** method that is used with the **`X-Operation`** header set to list retrieves a full or partial listing of objects that are stored in the vault. The URL consists of the vault name.

This operation is inefficient and is recommended for use in recovery scenarios only. List results are sorted numerically. The amount of data that is returned can be large. If transmission is interrupted, a list operation can be resumed by using the **`X-Start-Id`** header to indicate the last returned object identifier. Listing will resume after the indicated start object identifier, which will not be returned at the beginning of the listing.

Sample request

```
GET /vault-name HTTP/1.1
Host: access.example.com
X-Operation: list
X-Start-Id: 0af2592884594185968ab68cb51b0f2a0000
```

The list of stored objects is retrieved from a vault that is named **`vault-name`** by using the access server `access.example.com`. The **`X-Start-Id`** header is optional. When omitted, a complete list is retrieved. When present, listing starts after the indicated object identifier.

Sample response

```
HTTP/1.1 200 OK
Server: Cleversafe/3.2.0.0
Accept-Ranges: bytes
Date: Wed, 25 Sep 2013 14:38:12 GMT
X-SOH-Version: 2.5
X-Clv-Request-Id: ba30f684-7fb9-49bc-928e-ee05aa832096
Content-Length: 222
196b9e5c94d64afbb23901d6806170bf0000
6f8c571646554dbbb6a3c9429b47f1340000
9d82186423354edc9fddd43dd969adfa0000
d6133cd067ad4ea38cdcd1faa1e4e9040000
dc559d4b6602476d870c65673756be010000
e5c4483d371a44cf991e5fa4cf36caf00000
```

The response contains a list of object identifiers that are delimited by the end-of-line character.

Status codes

Table 12. List - status codes

Code	Status	Description
200	OK	The object was successfully written to the vault.
400	Bad Request	The request could not be understood by the access server due to malformed syntax.
401	Unauthorized	User authentication is needed. The request either did not provide correct credentials or the user does not have permission to read from the indicated vault.
403	Forbidden	The client has insufficient privileges to perform the operation.
500	Internal Server Error	An internal error was encountered on the access server. It is an unrecoverable error. Details can be provided in message body.

Limiting list results

The **`X-List-Length-Limit`** header will cause listing to stop after the indicated number of object identifiers is returned. A client can also stop listing by closing the connection.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM® in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Accesser®, Cleversafe®, ClevOS™, Dispersed Storage®, dsNet®, IBM Cloud Object Storage Accesser®, IBM Cloud Object Storage Dedicated™, IBM Cloud Object Storage Insight™, IBM Cloud Object Storage Manager™, IBM Cloud Object Storage Slicestor®, IBM Cloud Object Storage Standard™, IBM Cloud Object Storage System™, IBM Cloud Object Storage Vault™, SecureSlice™, and Slicestor® are trademarks or registered trademarks of Cleversafe, an IBM Company and/or International Business Machines Corp.

Other product and service names might be trademarks of IBM or other companies.

Homologation statement

This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact an IBM representative or reseller for any questions.

Printed in USA