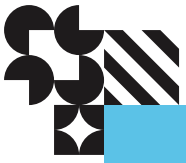

API Guide

HGST Active Archive System SA-7000

September 2015

1ET0033

Revision 1.1

Long Live Data ™ | www.hgst.com

Copyright

HGST, a Western Digital company
3403 Yerba Buena Road
San Jose, CA 95135
Produced in the United States

Contents

1 About this Guide

Topics:

The HGST Active Archive System provides an S3-compatible API. S3 is a storage infrastructure in which files are stored as *objects* that are organized into *buckets* and identified by *keys* owned by *users*. The Active Archive System supports creating, modifying, listing, and deleting objects, buckets, keys, and users. Most of these actions can be done through its S3 API. However, a few actions must be done through its OSMI or Q-Shell interface.

This guide provides a reference for the Active Archive System S3 API as well as instructions for completing all other storage management tasks. It also highlights the differences between Active Archive System and Amazon S3. This guide is not a standalone document; it is meant to be a supplement to the Amazon Simple Storage Service API Reference.

1.1 Conventions

Element	Sample Notation
OS shell or Q-Shell commands (user input)	`rm -rf /tmp`
OS shell or Q-Shell system output	`Installation successful!`
Commands longer than one line are split with "\"	`q.dss.manage.setPermissions('/manage', \ [....])`
User-supplied values	*`ManagementNodeVirtualIPAddress`* or `<ManagementNodeVirtualIPAddress>`
File and directory names	The file `aFile.txt` is stored in `/home/user`.
Any graphical user interface label	Click **OK**.
Keyboard keys and sequences	To cancel the operation, press `Ctrl+c`.
Menu navigation in a GUI	Navigate to **Dashboard** > **Administration** > **Hardware** > **Servers**.

1.2 Storage Notations

Convention	Prefix	Size (bytes)
KB	kilobyte	1,000
KiB	kibibyte	1,024
MB	megabyte	1,000,000
MiB	mebibyte	1,048,567
GB	gigabyte	1,00,000,000
GiB	gibibyte	1,073,741,824
TB	terabyte	1,000,000,000,000
TiB	tibibyte	1,099,511,627,776

- Sizes of disks are expressed with *SI prefixes* (kilo, mega, tera, peta, exa)
- Space, size of partitions and file systems are expressed with the *binary prefixes* (kibi, mebi, tebi, pebi, exbi)

- A comma (",") is used for digit grouping, for example 1,000 is 1 thousand.
- A period (".") is used as decimal mark, for example 12.5 %.

1.3 Admonitions

Type	Usage
Note:	Indicates extra information that has no specific hazardous or damaging consequences.
Tip:	Indicates a faster or more efficient way to do something.
Caution:	Indicates an action that, if taken or avoided, may result in hazardous or damaging consequences.
Warning:	Indicates an action that, if taken or avoided, may result in data loss or unavailability.

1.4 Related Documents

For more information about the Active Archive System, please consult the following documents:

- The *HGST Active Archive System Administration Guide* explains how to use the Active Archive System interfaces for executing system management, monitoring, and analytics tasks.
- The *HGST Active Archive System API Guide* provides a reference for the Active Archive System S3 API.
- The *HGST Active Archive System FRU Replacement Guide* provides procedures for replacing hardware components of the Active Archive System.
- The *HGST Active Archive System Installation Guide* provides instructions for the installation of the Active Archive System in the data center, and its initial bringup.
- The *HGST Active Archive System Release Notes* provide important information about changes, new features, and known limitations.
- The *HGST Active Archive System Site Requirements Document* contains data center requirements for the Active Archive System.
- The *HGST Active Archive System Troubleshooting Guide* provides help for issues you might encounter.
- The *HGST Active Archive System Upgrade Guide* provides instructions for software and firmware updates, and system expansion.

For the latest or online version of any of these documents, visit http://www.hgst.com/support.

2 Overview

Topics:

Active Archive System S3 API supports creating, modifying, listing, and deleting objects, buckets, and keys. This chapter gives an overview of Active Archive System S3 API.

2.1 Request Styles

Active Archive System S3 supports accessing buckets through both the *virtual hosted style* and the older *path style* requests. For more information about these request styles, see http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html.

2.1.1 Virtual Host Style Requests

In this request style, the bucket name is part of the domain name in the URL or the HTTP `Host` header.

2.1.1 Examples

PUT Object

```
"PUT http://mybucket.s3.hgst.com/myfile HTTP/1.1"
```

```
"PUT /myfile HTTP/1.1"
"Host: mybucket.s3.hgst.com"
...
```

DELETE Object

```
"DELETE http://mybucket.s3.hgst.com/myfile HTTP/1.1"
```

```
"DELETE /myfile HTTP/1.1"
"Host: mybucket.s3.hgst.com"
...
```

2.1.1 See Also

http://docs.aws.amazon.com/AmazonS3/latest/dev/VirtualHosting.html

2.1.2 Path Style Requests

In this request style, the bucket name is not part of the domain name; instead, it is the endpoint of the URL or the HTTP `Host` header.

2.1.2 Examples

PUT Object

```
"PUT /mybucket/myfile HTTP/1.1"
"Host: s3.hgst.com"
...
```

DELETE Object

```
"DELETE /mybucket/myfile HTTP/1.1"
"Host: s3.hgst.com"
...
```

2.1.2 See Also

http://docs.aws.amazon.com/AmazonS3/latest/dev/VirtualHosting.html

2.2 Bucket Names

Use DNS compliant, globally unique bucket names that comply with following rules:

- The bucket name must be between 3 and 63 characters long.
- The bucket name must be a series of one or more labels separated by a period (.), where each label:
  - Must start with a lowercase letter or a number
  - Must end with a lowercase letter or a number
  - Can contain lowercase letters, numbers, or dashes
- The bucket name must not be formatted as an IP address, for example `192.168.5.4`.

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/dev/BucketRestrictions.html.

2.3 Query Parameter Processing

Active Archive System S3 processes the following 7 query parameters:

- With the GET Bucket request:
  - `prefix`
  - `delimiter`
  - `marker`
  - `max-keys`
- With the GET Object request:
  - `Expires`
  - `AWSAccessKeyId`
  - `Signature`

All other query parameters are not implemented and throw an error to the client. To avoid getting a client error, add the following line to the `[s3]` section of the client daemon configuration file:

```
[s3]
query_params_to_ignore=ignore_this
```

2.4 Anonymous GET and LIST

Anonymous GET Access

If a bucket has READ access enabled for the special `everyone` user, unauthenticated clients (including web browsers) can read objects within that bucket.

By default, the `everyone` user does not have any access enabled to buckets, but that access can be enabled with the Q-Shell.

When anonymous GET is enabled, regular HTTP clients can fetch objects using URLs, such as `http://s3.hgst.com/`*`bucketname`*`/`*`objectname`*.

Anonymous LIST Permission

If a bucket has LIST access enabled for the `everyone` user, unauthenticated clients can list the contents of that bucket by fetching a URL, such as `http://s3.hgst.com/`*`bucketname`*.

The listing is an XML document in the format specified by S3.

2.5 S3 Permissions

An authenticated user can create up to 100 buckets. The user creating the bucket is the *owner* of the bucket.

The bucket owner can do all bucket operations that the Active Archive System has enabled, as described in Operations on Buckets on page 25. Users must have certain permissions, and authentication, to perform operations on buckets owned by other users.

Caution: Name spaces created using Active Archive System REST API are not visible using S3 API, and vice versa.

---

3 Authentication

Topics:

The Active Archive System implements a security model in which an identified and authenticated user can read, create, update, delete, and list objects. There is no anonymous user; every request must carry authentication information along with it.

> **Note:** The Active Archive System supports authentication and authorization at the bucket level, not at the object level.

3.1 Introduction

Using the Active Archive System command line interface, the storage administrator can:

- Create users with their credentials (password)
- Assign these users and their respective rights to a specific name space

Security rights are granted on a bucket basis, not on the objects that belong to this bucket.

The Active Archive System implements authentication by means of HTTP Digest authentication, as specified in RFC2617, as follows:

- The Active Archive System inspects the request URL and identifies the corresponding bucket for the request. The bucket metadata is inspected to find the relevant credentials. Special care is taken to also map the "upload-form" to the correct bucket secrets.
- A 401 response is sent back.
- The S3 client resends the request with an HTTP `Authentication` header.
- The Active Archive System checks the content of that header against the relevant credentials.

3.2 Getting Started

3.2 Out of the Box Authentication

When your environment is freshly installed, the following settings are active:

- No users are created, only the "everyone" account is present.
- Unauthenticated users can LIST on `/`.
- Unauthenticated users have no rights on `/manage`.
- Unauthenticated users have all (update, read, create, list, delete) rights on `/namespace`.

3.2 Before You Start

Remove unauthenticated users' rights on all name spaces.

```
q.dss.manage.setPermissions('/namespace','everyone',[])
```

3.2 Allowing an Administrator User to Create Name Spaces and Policies

1. Create a user to administer name spaces and policies.

```
q.dss.manage.addUser('admin','pass4admin')
```

2. Give the administrator rights on `/manage`.

```
q.dss.manage.setPermissions("/manage","admin",
['CREATE','LIST','READ','UPDATE','DELETE'])
```

   The administrator has no rights on the `/namespace` path and as such no rights on name spaces underneath.
3. Add a name space. You can do this using the Q-Shell.

   For the remainder of this section, assume that you have created a name space called `mynamespace`.
4. Create a user that will have all rights on this name space.

```
q.dss.manage.addUser('namespaceadmin','pass4namespaceadmin')
```

5. Set the permissions for the name space administrator.

```
q.dss.manage.setPermissions("/namespace/mynamespace","namespaceadmin",\\
 ['CREATE','LIST','READ','DELETE','UPDATE'])
```

3.2 Allowing Another User to Only Read Data from a Specific Name Space

- Create a user that only has read rights on this name space

```
q.dss.manage.addUser('namespacereader','pass4namespacereader')
```

- Set the permissions for the name space reader user

```
q.dss.manage.setPermissions("/namespace/mynamespace","namespacereader",['READ'])
```

3.2 Inheriting Permissions

Permissions can only be set at specific levels. Those levels are:

- `/`
- `/namespace`
- `/manage`
- `/namespace/mynamespace`, with `mynamespace` as the self-created name space.

The inherit flag dictates that the level where it is set "inherits" all permissions that are set at the level above it. For example:

- You have three users, one of which is the "everyone" user.
- The users have permissions at `/namespace` .
- You set the inherit flag at `/namespace/mynamespace` .

In this case, the permissions set at `/namespace` will also apply to `/namespace/mynamespace` .

```
In[7]:q.dss.manage.showPermissionSettings('/namespace/mynamespace')Out[7]:{u'Flags':
[u'INHERIT'],u'User-Permissions':{}}
```

3.2 Tips and Tricks

- If you are using a DSSTestSuite set up a running environment, you can use the following code to bring your environment into the state where the test name space is restricted.

```
q.dss.manage.setPermissions('/namespace','everyone',[])
q.dss.manage.addUser('admin','pass4admin')
q.dss.manage.setPermissions("/manage","admin",['CREATE','LIST', \\
                  'READ','DELETE','UPDATE'])
q.dss.manage.addUser('namespaceadmin','pass4namespaceadmin')
q.dss.manage.setPermissions("/namespace/test","namespaceadmin",\\
                  ['CREATE','LIST','READ','DELETE','UPDATE'])
q.X.manage.addUser('namespacereader','pass4namespacereader')
q.dss.manage.setPermissions("/namespace/test","namespacereader",['READ'])
```

- If you are using a browser to validate authentication, you can switch between users (and force re-authentication) by specifying the user name in the URL.

  Example: `http://admin@s3_domain_name:7070/manage/namespace/`
- The logfile of the client daemon is your first resource for troubleshooting authentication issues.
- The value of both the user name and the password of that user can be up to 256 complex characters (A-Z, a-z, 0-9 and/or special characters like #, @, !, ...).

3.2 Limitations

- At most 5,000,000 users can be created (as many as the number of supported name spaces)
- The maximum size of the user name and credentials is 256 bytes.
- It is recommended to limit the number of users that have rights on a single name space to 64.

3.2.1 WWW-Authenticate Header

3.2.1 Description

The Active Archive System sends the `WWW-Authenticate` header back to the client in the 401 response. For example,

```
WWW-Authenticate: Digest
  realm="Amplidata-AmpliStor/unknown-59b0f5a6e4709f566a5e8464153401ff76964cb9",
  qop=auth, nonce="246a5bf426a77dc28c195ce0373a9d2c",\\
  opaque="7e81b3abf2a84373b087cec73aca62c1"
```

The `WWW-Authenticate` header contains the following parameters:

realm

A unique string identifying the system.

qop

An indication to the client that the system wants to digest authentication.

nonce

A hexadecimal string that is calculated as a hash (see the RFC).

opaque

A hex string chosen at random at the start of the session.

3.2.2 Authorization Header

3.2.2 Description

The client sends the `Authorization` header in the subsequent resend of the request. For example,

```
Authorization: Digest
  username="user", realm="Amplidata-AmpliStor/unknown-\\
  59b0f5a6e4709f566a5e8464153401ff76964cb9",
  nonce="246a5bf426a77dc28c195ce0373a9d2c", uri="/namespace/VOL/file1", qop=auth,
  nc=00000001, cnonce="abababab", response="9ff88e8ecb9534c434f1e2a4292b2dd6",
  opaque="7e81b3abf2a84373b087cec73aca62c1"
```

The `Authorization` header contains the following parameters:

username

The user name, used for validating the response.

realm

Must be identical to the value used in `WWW-Authenticate`.

nonce

Must be identical to the value used in `WWW-Authenticate`.

uri

The request URI.

qop

Must be "`auth`".

nc

An 8-digit hex string counting up for each new request within the authorization session. This must be "1" in the first request and incremented each time.

cnonce

A random 8-byte string chosen by the client.

response

The client calculates the response by means of a formula based on request-method, URI, nonce, user name, secret realm, qop, nc, and cnonce. As all these fields are also available, the server can afterwards do the same calculation and see if it matches.

opaque

Must be identical as in `WWW-Authenticate`.

The Active Archive System changes the nonce at times during the session, because a nonce has a limited lifetime. It sends a STALE flag to the client to indicate that the nonce is stale.

3.2.3 Query Parameters

You can specify authentication information for certain types of requests by passing the required information in query parameters instead of in the HTTP `Authorization` header. For example, the Active Archive System authenticates your GET Object request when you specify the `Expires`, `AWSAccessKeyId`, and `Signature` query parameters. These parameters are described in the table below.

Parameter Name	Description	Example value
`Expires`	The time when the signature expires, specified as the number of seconds since the epoch (00:00:00 UTC on January 1, 1970). A request received after this time (according to the server), will be rejected.	`1141889120`
`AWSAccessKeyId`	Your AWS Secret Access Key Id. The value is alphanumeric (A-Z, a-z and/or 0-9) and up to 256 characters. This specifies the AWS Secret Access Key used to sign the request, and (indirectly) the identity of the developer making the request.	`AKIAIOSFODNN7-EXAMPLE`
`Signature`	The URL encoding of the Base64 encoding of the HMAC-SHA1 of `StringToSign`.	`vjbyPxybdZaNmGa%2-ByT272YEAiv4%3D`

Using these three query parameters instead of the HTTP `Authorization` header, you can enable direct third-party browser access to your private S3 data without proxying the request. The idea is to construct a *pre-signed request* and encode it as a *pre-signed URL* that the end-user's browser can retrieve. Additionally, you can limit a pre-signed request by specifying an expiration time.

For more information, see:

- http://docs.amazonwebservices.com/AmazonS3/latest/dev/RESTAuthentication.html#RESTAuthenticationQueryStringAuth
- http://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html
- http://docs.aws.amazon.com/AmazonS3/latest/dev/PresignedUrlUploadObject.html

3.3 User Identification

Users are identified by their unique user ID. This int64 value is automatically assigned when creating a new user and the value cannot be changed. Each user ID is linked to a user name (which must also be unique, deployment-wide), a password and a status, which can be `ACTIVE`, `INACTIVE`, or `DELETED`.

User ID 0 corresponds to a default user with user name `everyone`, who has no password. It corresponds to the unauthenticated user. Setting permissions on a path for this user allows unauthenticated users to perform operations.

3.4 Paths and Flags

The above mentioned permissions can be assigned, for each individual user, to a number of specific paths:

- root: `/`
- manage: `/manage`
- namespace root: `/namespace`
- a namespace: `/namespace/`*`namespace_name`*

In addition to user permissions, the security system allows to enable a global (not user specific) flag on each of these paths, namely *INHERIT*. Enabling the inherit flag on a path means that the permissions defined on this path are combined with the permissions of its parent path.

For example:

If a user has permission READ on `/namespace/VOL` and permission CREATE on `/namespace`, this user will, if the INHERIT flag on path `/namespace/VOL` is not set, not be allowed to create new objects in `/namespace/VOL`. If the INHERIT flag is enabled on `/namespace/VOL`, this user will obtain effective rights of READ, CREATE on `/namespace/VOL` .

INHERIT also works in cascade, for example if INHERIT is enabled on `/`, `/namespace` and `/namespace/VOL`, then the effective permissions on `/namespace/VOL` are those combined over all three paths. The cascade works top-down and stops at the first level where the INHERIT flag is not set.

3.5 Q-Shell Interface for Assignment of Rights to a Specific Path

3.5.1 Set Permission Flags for a Specific Path

To set the permission flags for a specified path, use the following code:

```
q.dss.manage.setPermissionFlags(self,\
path,\
permissionFlags,\
nodeIP='127.0.0.1',\
port=23510)
```

Parameter	Value	Explanation
`path`	string	The path for which to show the permissions: `/`, `/manage`, `/namespace` , or `/namespace/`*`namespace_name`* .

Parameter	Value	Explanation
`permissionFlags`	[string]	A list of permission flags. Valid values: `Inherit`.
`nodeIP`	string	The IP address of the storage daemon to contact.
`port`	int	The port number of the storage daemon to contact.

3.5.2 Get Permission Flags for a Specific Path

To get the permission flags for a specified path, use the following code:

```
q.dss.manage.showPermissionFlags(self,\
path,\
nodeIP='127.0.0.1',\
port=23510)
```

Parameter	Value	Explanation
`path`	string	The path for which to show the permissions: `/`, `/manage`, `/namespace` , or `/namespace/`*`namespace_name`* .
`nodeIP`	string	The IP of the storage daemon to contact.
`port`	int	The port of the storage daemon to contact.

3.5.3 Get Permission Flags for a Specific User

To show a user's access permissions for a specified path, use the following code:

```
q.dss.manage.showPermissions(self,\
path,\
userName, \
nodeIP='127.0.0.1', \
port=23510)
```

Parameter	Value	Explanation
`path`	string	The path for which to show the permissions: `/`, `/manage`, `/namespace` , or `/namespace/`*`namespace_name`* .
`userName`	string	The user name of the user, or `everyone` for the permissions for special user `everyone`.
`nodeIP`	string	The IP address of the storage daemon to contact.
`port`	int	The port number of the storage daemon to contact.

3.5.4 Set Permission Flags for a Specific User

To set the access permissions of a specific user on a specified path, use the following code:

```
q.dss.manage.setPermissions(self,\
path,\
userName,\
permissions,\
nodeIP='127.0.0.1', \
port=23510)
```

Parameter	Value	Explanation
`path`	string	The path for which to show the permissions: `/`, `/manage`, `/namespace` , or `/namespace/`*`namespace_name`* .
`userName`	string	The user name of the user, or `everyone` for the permissions for special user `everyone`.
`permissions`	[string]	A list of permissions. Valid values: `Read`, `Create`, `Delete`, `List`, `Update`.
`nodeIP`	string	The IP address of the storage daemon to contact.
`port`	int	The port number of the storage daemon to contact.

3.5.5 Show All Permissions and Flags for a Path

To show all permissions and flags for a specified path, use the following code:

```
q.dss.manage.showPermissionSettings(self,\
path, \
nodeIP='127.0.0.1', \
port=23510)
```

Parameter	Value	Explanation
`path`	string	The path for which to show the permissions: `/`, `/manage`, `/namespace` , or `/namespace/`*`namespace_name`* .
`nodeIP`	string	The IP address of the storage daemon to contact.
`port`	int	The port number of the storage daemon to contact.

4 API Reference

Topics:

4.1 Differences Between Active Archive System S3 and Amazon S3

4.1 Differences in Operations on the Service

Situation	Amazon AWS S3	Active Archive System S3
GET Service requests with query parameters when using a webdrive client		The Active Archive System does not handle several query parameters when using a webdrive client. Be sure to specify a bucket name in the request.

4.1 Differences in Operations on Buckets

Situation	Amazon AWS S3	Active Archive System S3
PUT Bucket with an existing bucket name	Returns `HTTP 200 OK`	Returns `HTTP 409 Conflict`
PUT Bucket with a bucket name containing special characters (`{, }, <, >, [, ] | `, ^, "`)	Returns `HTTP 400 Bad Request`	Returns `HTTP 501 Not Implemented`

Creating too many buckets for the same user		Missing elements:
		`CurrentNumberOfBuckets`
		`AllowdNumberOfBuckets`
		`HostId`
		Extra element:
		`Resource`

DELETE Bucket for a nonempty bucket		Extra period (".") mark
GET Bucket (List Objects) with `max-keys` greater than `1000`	Works without problems	Does not allow a `max-keys` value greater than `1000`, due to possible stack overflow.

GET Bucket versioning, dummy response		Returns extra headers:
		`Content-Type`
		`Pragma`

Situation	Amazon AWS S3	Active Archive System S3

		`Cache-Control`
		`Expires`
		`Date` (*omitted*)

HEAD Bucket		Returns extra `;charset=UTF-8` in `Content-Type` in header
HEAD Bucket for a bucket or key that does not exist	Returns `HTTP 404 Page Not Found`	Returns `HTTP 404 Page Not Found` together with the reason in the body: `NoSuchBucket`: The specified bucket does not exist. `NoSuchKey`: The specified key does not exist.
HEAD Bucket without sufficient permissions to access that bucket	Returns `HTTP 403 Forbidden`	Returns `HTTP 403 Forbidden`, together with the reason in the body: `AccessDenied`: Access denied
HEAD Bucket versioning	Returns `HTTP 405 Method Not Allowed`	Returns `HTTP 501 Not Implemented`

4.1 Differences in Operations on Objects

Situation	Amazon AWS S3	Active Archive System S3
Specifying an object key name	Maximum of 1024 characters	Allows more than 1024 characters
Specifying an object name with [,], \, and ^, using `s3cmd` (unverified with other clients)	Returns `HTTP 200 OK`	Returns `HTTP 501 Not Implemented`
Adding custom metadata, per object	Maximum of 2 KiB	Allows more than 2 KiB (see limitations)
DELETE Object for nonexistent objects	Returns `HTTP 200 OK`	Returns errors
Delete multiple objects	Returns `HTTP 200 OK`	Returns `HTTP 405 Method Not Allowed`

GET Object or HEAD Object with one of the following response-header query parameters:
`response-content-type`
`response-content-language`
`response-expires`
`response-cache-control`
`response-content-disposition`
`response-content-encoding`
		Sends an incorrect response

PUT Object for an object larger than 5GB	Not supported; you must use the multipart upload API	Supported, but may send errors

Situation	Amazon AWS S3	Active Archive System S3

PUT Object for a nonexistent bucket		Returns missing elements:
		`BucketName`
		`HostId`

PUT Object with the following headers:
`x-amz-storage-class`
`x-amz-server-side-encryption`
`x-amz-website-redirect-location`
`x-amz-grant-read`
`x-amz-grant-write`
`x-amz-grant-read-acp`
`x-amz-grant-write-acp`
`x-amz-grant-full-control`
		May fail this request

Canceling a multipart upload and then trying to upload another part	Socket timeout	Returns `HTTP 404 Not Found`, together with the reason in the body: `NoSuchUpload`: Specified multipart upload does not exist.

4.1 Miscellaneous Differences

Situation	Amazon AWS S3	Active Archive System S3
Sending dummy headers	Returns `HTTP 200 OK`	Returns errors
Sending a request with `Expires` parameter	Returns `HTTP 200 OK`	Returns `HTTP 501 Not Implemented`
Sending a request with `TE` parameter	Returns `HTTP 200 OK`	Returns `HTTP 501 Not Implemented`
Sending a request with timezone `CET`*`randomstring`*	This string is interpreted as `CET` by Amazon S3	This string is interpreted as `CET`*`ran`* by the Active Archive System, which is an unknown timezone.
Sending a request with an unsupported timezone format	Returns `HTTP 403 Forbidden`	Returns `HTTP 500 Internal Server Error`
Sending a request without the HTTP `Date` header	Returns `HTTP 403 Forbidden`	Returns `HTTP 500 Internal Server Error`
Sending a request with an unsupported HTTP version	Returns `HTTP 505 HTTP Version Not Supported`	Returns `HTTP 500 Internal Server Error`
Sending a request with bad URL encoding	Returns `HTTP 400 Invalid URI: EOF`	Returns `HTTP 501 Not Implemented`
Sending a request with an invalid path	Returns `HTTP 400 Invalid URI`	Returns `HTTP 500 Internal Server Error`
Sending a request with a different `Host` value than the one configured in the Active Archive System	Returns `HTTP 301 Moved Permanently`	Returns `HTTP 400 Bad Request`

Situation	Amazon AWS S3	Active Archive System S3
Specifying an incorrect md5sum value	Returns the error `The Content-MD5 you specified did not match what we received`	Returns the error `The Content-MD5 you specified was an invalid`
Sending a request with `Accept-Encoding: Identity`	Transfer-Encoding is chunked	The Active Archive System ignores header and Transfer-Encoding is not chunked
Sending a GET request with `If-Range` in header	Returns `HTTP 200 OK`	Returns `HTTP 501 Not Implemented`

- The following common response headers are not implemented:
  - `x-amz-id-2`
  - `x-amz-request-id`
- The `x-amz-metadata-directive` header is not implemented.
- When trying to to do a server-side copy, the following `x-amz-copy-source-if-*` headers are not implemented:
  - `x-amz-copy-source-if-match`
  - `x-amz-copy-source-if-none-match`
  - `x-amz-copy-source-if-unmodified-since`
  - `x-amz-copy-source-if-modified-since`

4.2 Headers

4.2 Custom Headers You Want to Upload with the Object

If you want to upload custom headers, add the following line to the `[s3]` section of the client daemon configuration file:

```
[s3]
custom_headers_to_upload=uploadme
```

Now the header `uploadme`, together with its value, is stored with the object metadata.

Caution: This is only supported for PUT object.

4.2 Headers You Want the Server to Ignore

If you want the server to ignore some headers, you can do the following:

```
[s3]
headers_to_ignore=Origin
```

This way the server does not respond with an error if you send the `Origin` header; it ignores the header.

4.2 x-amz-meta-* Headers You Want to Upload with the Object

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/dev/UsingMetadata.html.

4.3 Summary

4.3 Operations on the Service

Method	Supported by Active Archive System?
GET Service on page 24	Yes

4.3 Operations on Buckets

Method	Supported by Active Archive System?
DELETE Bucket on page 25	Yes
DELETE Bucket lifecycle	No
DELETE Bucket policy	No
DELETE Bucket website	No
GET Bucket (List Objects) on page 26	Yes
GET Bucket acl	Yes (dummy response)
GET Bucket lifecycle	No
GET Bucket policy	No
GET Bucket location	Yes (dummy response)
GET Bucket logging	Yes (dummy response)
GET Bucket notification	No
GET Bucket Object versions	No
GET Bucket requestPayment	No
GET Bucket versioning	Yes (dummy response)
GET Bucket website	No
HEAD Bucket on page 28	Yes
List Multipart Uploads on page 29	Yes
PUT Bucket on page 29	Yes
PUT Bucket acl	No
PUT Bucket lifecycle	No
PUT Bucket policy	No
PUT Bucket logging	No
PUT Bucket notification	No
PUT Bucket requestPayment	No
PUT Bucket versioning	No
PUT Bucket website	No

4.3 Operations on Objects

Method	Supported by Active Archive System?
DELETE Object on page 31	Yes
Delete Multiple Objects	No
GET Object on page 32	Yes(*)
GET Object acl	No
GET Object torrent	No
HEAD Object on page 32	Yes(*)
POST Object	No

Method	Supported by Active Archive System?
PUT Object on page 33	Yes
PUT Object acl	No
PUT Object - Copy on page 34	Yes
Initiate Multipart Upload on page 37	Yes
Upload Part on page 39	Yes
Upload Part - Copy on page 40	Yes
Complete Multipart Upload on page 36	Yes
Abort Multipart Upload on page 35	Yes
List Parts on page 38	Yes

(*) The GET Object and HEAD Object requests support `if-*` headers. For more information, see Request Headers for GET and HEAD Object on page 40.

4.4 Compatibility Calls

To enable compatibility with some applications, the Active Archive System returns a dummy response, rather than an exception, for the following calls. It does not implement these calls.

4.4 GET Bucket location

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketGETlocation.html

`s3cmd` call:

```
s3cmd info s3://bucketname
```

Request:

```
GET /?location HTTP/1.1
Date: Mon, 26 Nov 2012 20:31:36 GMT
Authorization: AWS admin:W8QyXloRQVXXGeLkMzDE223iYMg=
User-Agent: jclouds/1.5.0-beta.6 java/1.7.0_09
Host: testfolder2.s3.amazonaws.com:7070
Accept: text/html, image/gif, image/jpeg,
*; q=.2, */*; q=.2
Connection: keep-alive
""
```

Response:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<LocationConstraint xmlns=\"http://s3.amazonaws.com/doc/2006-03-01/\"/>
```

4.4 GET Bucket logging

For more information, see Amazon S3 GET logging.

`s3cmd` call:

```
s3cmd accesslogs3://bucketname
```

Request:

```
GET /bucketname?logging HTTP/1.1
```

```
Host: 127.0.0.1:7070
Accept: */*
x-amz-acl:authenticate-read
x-amz-date:Tue, 27 Nov 2012 08:43:34 GMT
Authorization: AWS admin:bAGHIFf98WMPu1FGpAmTAEHeaJM=
""
```

Response:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<BucketLoggingStatus xmlns=\"http://s3.amazonaws.com/doc/2006-03-01/\">
<LoggingEnabled>
   <TargetBucket>mybucketlogs</TargetBucket>
   <TargetPrefix>mybucket-access_log</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

4.4 GET Bucket versioning

For more information, see Amazon S3 GET versioning Status

`s3cmd` call:

No `s3cmd` call exists for GET /?versioning.

Request:

```
GET /bucketname?versioning HTTP/1.1
Host: 127.0.0.1:7070
Accept: */*
x-amz-acl:authenticate-read
x-amz-date:Tue, 27 Nov 2012 08:43:34 GMT
Authorization: AWS admin:iC1TR8CvqFYxM+QOoCG9PY0r6cg=
""
```

Response:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<VersioningConfiguration xmlns=\"http://s3.amazonaws.com/doc/2006-03-01/\"/>
```

4.4 GET Bucket acl

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketGETacl.html.

`s3cmd` call:

```
s3cmd info s3://bucketname
```

Request:

```
GET /?acl HTTP/1.1
Date: Mon, 26 Nov 2012 21:22:36 GMT
Authorization: AWS admin:05MWHGy46KX0gW6mL5DN1v6FfC8=
User-Agent: jclouds/1.5.0-beta.6 java/1.7.0_09
Host: testfolder5.s3.amazonaws.com:7070
Accept: text/html, image/gif, image/jpeg,
*; q=.2, */*; q=.2
Connection: keep-alive
""
```

Response:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
```

```
<AccessControlPolicyxmlns=\"http://s3.amazonaws.com/doc/2006-03-01/\"/>
```

4.5 Operations on the Service

4.5.1 GET Service

4.5.1 Description

GET Service lists all the buckets owned by the authenticated sender of the request. For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTServiceGET.html.

4.5.1 Requests

Syntax

```
"GET / HTTP/1.1"
"Host: s3_domain_name"
"Accept-Encoding: identity"
"content-length: 0"
"Authorization: authorization_string"
"x-amz-date: Tue, 04 Sep 2012 08:03:16 +0000"
""
```

4.5.1 Responses

For a description of response elements, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTServiceGET.html.

4.5.1 Examples

Sample s3cmd

```
s3cmd ls
```

Sample Request

```
"GET / HTTP/1.1"
"Host: s3.hgst.com"
"Accept-Encoding: identity"
"content-length: 0"
"Authorization: authorization_string"
"x-amz-date: Tue, 04 Sep 2012 08:03:16 +0000"
""
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3_domain_name/doc/2006-03-01/">
        <Owner>
                <ID>32</ID>
                <DisplayName>admin</DisplayName>
        </Owner>
        <Buckets>
                <Bucket>
                        <Name>mybucketname1</Name>
                        <CreationDate>2013-06-13T14:26:06.000Z</CreationDate>
                </Bucket>
                <Bucket>
                        <Name>mybucketname2</Name>
                        <CreationDate>2013-06-13T14:26:13.000Z</CreationDate>
```

```
                </Bucket>
                <Bucket>
                        <Name>mybucketname3</Name>
                        <CreationDate>2013-06-13T11:56:02.000Z</CreationDate>
                </Bucket>
        </Buckets>
</ListAllMyBucketsResult>
```

4.6 Operations on Buckets

4.6.1 DELETE Bucket

4.6.1 Description

DELETE Bucket deletes the empty bucket specified in the request. To make this request, you must be the owner of the bucket. For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketDELETE.html.

4.6.1 Requests

Syntax

```
"DELETE /bucketname HTTP/1.1"
"Host: s3_domain_name"
"Accept-Encoding: identity"
"content-length: 0"
"Authorization: authorization_string"
"x-amz-date: Tue, 04 Sep 2012 08:03:16 +0000"
""
```

4.6.1 Responses

This operation uses only common response headers. For a description of common response headers, see http://docs.aws.amazon.com/AmazonS3/latest/API/RESTCommonRequestHeaders.html.

4.6.1 Examples

Sample s3cmd

```
s3cmd rb s3://bucketname
```

Sample Request

```
"DELETE /mybucketname1 HTTP/1.1"
"Host: s3_domain_name"
"Accept-Encoding: identity"
"content-length: 0"
"Authorization: authorization_string"
"x-amz-date: Tue, 04 Sep 2012 08:03:16 +0000"
""
```

Sample Response

```
HTTP/1.1 204 No Content
DAV: 1,3
```

```
Date: Tue, 04 Sep 2012 08:03:16 GMT
Server: SystemID
Content-Length: 0
```

4.6.2 GET Bucket (List Objects)

4.6.2 Description

GET Bucket (List Object) lists the objects in the specified bucket. To make this request, you must be the bucket owner or have list permissions on the bucket path. For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketGET.html.

4.6.2 Requests

Syntax

```
"GET /bucketname HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:46:14 GMT"
"Authorization: authorization_string"
""
```

4.6.2 Responses

This operation uses only common response headers. For a description of common response headers, see http://docs.aws.amazon.com/AmazonS3/latest/API/RESTCommonRequestHeaders.html.

4.6.2 Examples

Sample s3cmd

```
s3cmd ls s3://bucketname
```

Sample Request and Response with Full Listing and No Prefix

```
"GET /mybucket HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:46:14 GMT"
"Authorization: authorization_string"
""
```

```
HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 13:45:35 GMT
Server: SystemID
Content-Type: application/xml;charset=UTF-8
Content-Length: 5903
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
```

The XML response body (in the case where files `x` and `y` are present as keys in *`bucketname`*) looks like this:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<ListBucketResult xmlns=\"http://s3_domain_name/doc/2006-03-01/\">
        <Name>bucketname</Name>
        <Prefix></Prefix>
```

```
        <Marker></Marker>
        <MaxKeys>1000</MaxKeys>
        <IsTruncated>false</IsTruncated>
        <Contents>
                <Key>x</Key>
                <LastModified>2012-09-04T11:15:12.000Z</LastModified>
                <ETag>c10be79e50b74fff8c643b9760a33b3e</ETag>
                <Size>55</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                      <ID></ID>
                      <DisplayName></DisplayName>
                </Owner>
        </Contents>
        <Contents>
                <Key>y</Key>
                <LastModified>2012-09-04T11:15:07.000Z</LastModified>
                <ETag>30cbeaa5e43c4443bc60d24aa6fd8d92</ETag>
                <Size>55</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID></ID>
                        <DisplayName></DisplayName>
                </Owner>
        </Contents>
</ListBucketResult>
```

Sample Request and Response with Full Listing and Prefix

```
"GET /bucketname?prefix=x HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:46:14 GMT"
"Authorization: authorization_string"
""
```

```
HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 13:45:35 GMT
Server: SystemID
Content-Type: application/xml;charset=UTF-8
Content-Length: 5903
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
```

The body (in case file x and y are present as keys in bucket "bucketname") looks like:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<ListBucketResult xmlns=\"http://s3_domain_name/doc/2006-03-01/\">
        <Name>bucketname</Name>
        <Prefix></Prefix>
        <Marker></Marker>
        <MaxKeys>1000</MaxKeys>
        <IsTruncated>false</IsTruncated>
        <Contents>
                <Key>x</Key>
                <LastModified>2012-09-04T11:15:12.000Z</LastModified>
                <ETag>c10be79e50b74fff8c643b9760a33b3e</ETag>
```

```
                <Size>55</Size>
                <StorageClass>STANDARD</StorageClass>
                <Owner>
                        <ID></ID>
                        <DisplayName></DisplayName>
                </Owner>
        </Contents>
</ListBucketResult>
```

Note how the key with name `y` is not present in the XML since it did not match the mentioned prefix.

4.6.3 HEAD Bucket

4.6.3 Description

HEAD Bucket checks whether a specified bucket exists, and whether you have permission to access it. For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketHEAD.html.

4.6.3 Requests

Syntax

```
"HEAD /bucketname HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 03 Sep 2012 14:54:17 GMT"
"Authorization: authorization_string"
""
```

4.6.3 Responses

This operation uses only common response headers. For a description of common response headers, see http://docs.aws.amazon.com/AmazonS3/latest/API/RESTCommonRequestHeaders.html.

4.6.3 Examples

Sample s3cmd

No s3cmd call exists for HEAD Bucket.

Sample Request

```
"HEAD /mybucketname1 HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 03 Sep 2012 14:54:17 GMT"
"Authorization: authorization_string"
""
```

Sample Response

```
HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 13:45:35 GMT
Server: SystemID
Content-Type: application/xml;charset=UTF-8
Content-Length: 5903
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
```

4.6.4 PUT Bucket

4.6.4 Description

PUT Bucket creates a new bucket. To make this request, you must be an authenticated user. For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketPUT.html.

4.6.4 Requests

Syntax

```
"PUT /bucketname HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"Content-Length: 0"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 14:51:57 GMT"
"Authorization: authorization_string"
"Expect: 100-continue"
""
```

4.6.4 Responses

This operation uses only common response headers. For a description of common response headers, see http://docs.aws.amazon.com/AmazonS3/latest/API/RESTCommonRequestHeaders.html.

4.6.4 Examples

Sample s3cmd

```
s3cmd mb s3://bucketname
```

Sample Request

```
"PUT /mybucketname1 HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"Content-Length: 0"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 14:51:57 GMT"
"Authorization: authorization_string"
"Expect: 100-continue"
""
```

Sample Response

```
HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 14:51:19 GMT
Server: SystemID
Content-Type: text/plain;charset=UTF-8
Content-Length: 0
```

4.6.5 List Multipart Uploads

4.6.5 Description

This operation lists all multipart uploads which are in progress. An in progress multipart upload is a multipart upload that has been initiated, using the Initiate Multipart Upload on page 37 request, but has not yet been completed or aborted. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadListMPUpload.html.

4.6.5 Requests

Syntax

```
GET /?uploads HTTP/1.1
Host: 192.168.12.148:7070
Date: Mon, 1 Nov 2013 20:34:56 GMT
Authorization: authorization_string
```

4.6.5 Responses

```
HTTP/1.1 204 No Content
DAV: 1,3
Date: Tue, 04 Sep 2012 08:03:16 GMT
Server: SystemID
Content-Length: 0
```

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2013 20:34:56 GMT
Content-Length: 1330
Connection: keep-alive
Server: SystemID
```

4.6.5 Examples

Sample Response Body When There Are In-Progress Multipart Uploads

```
<?xml version="1.0" encoding="UTF-8"?>
<ListMultipartUploadsResult xmlns="s3_domain_name">
        <Bucket>testbucket</Bucket>
        <KeyMarker />
        <UploadIdMarker />
        <NextKeyMarker />
        <NextUploadIdMarker />
        <IsTruncated>false</IsTruncated>
        <Upload>
                <Key>MyMusic.mp4</Key>
                <UploadId>XMgbGlrZSBlbHZpbmcncyBub3QgaGF2aW5nIG11Y2ggbHVjaw</UploadId>
                <Initiator>
                        <ID>arn:aws:iam::111122223333:user/user1-11111a31-\\
                                        17b5-4fb7-9df5-b111111f13de
                        </ID>
                        <DisplayName>user1-11111a31-17b5-4fb7-9df5-b111111f13de</
DisplayName>
                </Initiator>
                <Owner>

 <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
                        <DisplayName>OwnerDisplayName</DisplayName>
                </Owner>
                <StorageClass>STANDARD</StorageClass>
                <Initiated>2010-11-10T20:48:33.000Z</Initiated>
        </Upload>
</ListMultipartUploadsResult>
```

4.7 Operations on Objects

4.7.1 DELETE Object

4.7.1 Description

DELETE Object deletes the permissions on the object path.

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectDELETE.html.

4.7.1 Requests

Syntax

```
"DELETE /bucketname/a/b/c/d/e HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:42:42 GMT"
"Authorization: authorization_string"
""
```

4.7.1 Responses

4.7.1 Examples

Sample s3cmd

```
s3cmd del s3://bucketname/x
```

Sample Request

```
"DELETE /mybucketname1/a/b/c/d/e HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:42:42 GMT"
"Authorization: authorization_string"
""
```

Sample Response

```
HTTP/1.1 204 No Content
DAV: 1,3
Date: Mon, 27 Aug 2012 13:45:35 GMT
Server: SystemID
Content-Type: text/plain
Content-Length: 0
```

4.7.2 GET Object

4.7.2 Description

GET Object reads the permissions on the object path.

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectGET.html.

4.7.2 Requests

Syntax

```
"GET /bucketname/a/b HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:41:44 GMT"
"Authorization: authorization_string"
""
```

4.7.2 Responses

4.7.2 Examples

Sample s3cmd

```
s3cmd get s3://mybucketname1/x/tmp/x.res
```

Sample Request

```
"GET /bucketname/a/b HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:41:44 GMT"
"Authorization: authorization_string"
""
```

Sample Response

```
HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 13:42:03 GMT
Server: SystemID
Content-Length: 0
Accept-Ranges: bytes
Last-Modified: Mon, 27 Aug 2012 13:41:06 GMT
ETag: \"f0a602cd8cd2d4e01c811daa4ba1b451\"
```

4.7.3 HEAD Object

4.7.3 Description

HEAD Object returns the specified object's metadata. For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectHEAD.html.

4.7.3 Requests

Syntax

```
"HEAD /bucketname/a/b/c/d/e HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
```

```
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:42:42 GMT"
"Authorization: authorization_string"
""
```

4.7.3 Responses

4.7.3 Examples

Sample s3cmd

No `s3cmd` call exists for HEAD Object.

Sample Request

```
"HEAD /mybucketname1/a/b/c/d/e HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:42:42 GMT"
"Authorization: authorization_string"
""
```

Sample Response

```
HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 13:42:03 GMT
Server: SystemID
Content-Length: 1048576
Accept-Ranges: bytes
Last-Modified: Mon, 27 Aug 2012 13:42:03 GMT
ETag: \"fdf3a80e1ce052e7cd021563d98cb53b\"
```

4.7.4 PUT Object

4.7.4 Description

PUT Object does one of the following:

- Updates permissions if the object status exists.
- Creates permissions if the object does not exist on the object path.

For more information, see http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectPUT.html.

4.7.4 Requests

Syntax

```
"PUT /bucketname/a/b HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"Content-Length: 0"
"Content-Type: binary/octet-stream"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:41:44 GMT"
"Authorization: authorization_string"
"Expect: 100-continue"
""
```

4.7.4 Responses

4.7.4 Examples

Sample s3cmd

```
s3cmd put /tmp/x s3://bucketname/x
```

Sample Request

```
"PUT /mybucketname1/a/b HTTP/1.1"
"User-Agent: Mozilla/4.0 (Compatible; AMZS3; CVRF 1.0; Windows Server 2008 R2)"
"Host: s3_domain_name"
"Accept: */*"
"Content-Length: 0"
"Content-Type: binary/octet-stream"
"x-amz-acl: authenticated-read"
"x-amz-date: Mon, 27 Aug 2012 13:41:44 GMT"
"Authorization: authorization_string"
"Expect: 100-continue"
""
```

Sample Response

```
response: HTTP/1.1 200 OK
DAV: 1,3
Date: Mon, 27 Aug 2012 13:41:06 GMT
Server: SystemID
Content-Type: text/plain;charset=UTF-8
Content-Length: 0
Accept-Ranges: bytes
Last-Modified: Mon, 27 Aug 2012 13:41:06 GMT
ETag: \"3c46e4cdef1a6f620271991fbb84e7b6\"
```

4.7.5 PUT Object - Copy

4.7.5 Description

PUT Object - Copy creates a copy of an object that is already stored in the Active Archive System. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectCOPY.html.

4.7.5 Requests

Syntax

```
"PUT target_bucketname/target_objectname HTTP/1.1"
'content-length': '0'
'Authorization': 'authorization_string'
'x-amz-date': 'Thu, 13 Jun 2013 14:35:25 +0000'
'x-amz-copy-source': '/source_bucketname/source_objectname'
'x-amz-metadata-directive': 'COPY'
```

4.7.5 Responses

This operation returns the `CopyObjectResult` response element.

4.7.5 Examples

Sample s3cmd

```
s3cmd cp s3://bucketname/xs3://bucketname/y
```

Sample Request

```
"PUT mybucketname1/favoriteobj1 HTTP/1.1"
```

```
'content-length': '0'
'Authorization': 'authorization_string'
'x-amz-date': 'Thu, 13 Jun 2013 14:35:25 +0000'
'x-amz-copy-source': '/mybucketname1/favoriteobj2'
'x-amz-metadata-directive': 'COPY'
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 13 Jun 2013 14:35:25 GMT
Dav: 1,3
content-type: application/xml
content-length: 239
server: SystemID
```

The body of this response is in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
   <CopyObjectResult xmlns="http://s3_domain_name/doc/2006-03-01/">
      <LastModified>Thu, 13 Jun 2013 14:35:25 GMT</LastModified>
      <ETag>&quot;7c909b3e2820c8b47ed418753698a6da&quot;</ETag>
   </CopyObjectResult>
```

4.7.6 Abort Multipart Upload

4.7.6 Description

Abort Multipart Upload stops an in progress multipart upload and frees any parts that have already been uploaded. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadAbort.html.

Note:

Most S3 clients, such as `s3cmd` and `Cyberduck`, support multipart object operations by default. However, you must configure your S3 client to be compatible with the following Active Archive System S3 restrictions on multipart object operations:

- Maximum object size is 5 TB.
- Minimum object size is 5 MB.
- Maximum number of parts is 10,000.

For more information on changing the settings in the client daemon configuration file, see *Managing Storage* in the *HGST Active Archive System Administration Guide*.

4.7.6 Examples

Sample Request

```
DELETE /example-object?uploadId=BbyOjajJe30TSjDP7Bq4Z.rmTu_\\
pDwVOFBFi7tqHWwe5ZgUFMKGwZje2TQl4Zz_i4mBhjLD1vXxj8_\\
c7rQC_BLwGQplJh_bcOftTq0b.U0o4_F8lyRkYJ8qBKFFmNwhS HTTP/1.1
Host: s3_domain_name
Date:  Fri, 13 Dec 2013 08:19:22 GMT
Authorization: authorization_string
```

Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: Weag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 996c76696e6727732072657175657374
Date:  Fri, 13 Dec 2013 08:19:22 GMT
Content-Length: 0
```

```
Connection: keep-alive
Server: SystemID
```

4.7.7 Complete Multipart Upload

4.7.7 Description

Complete Multipart Upload completes a multipart upload by assembling all of the uploaded parts. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadComplete.html.

Note:

Most S3 clients, such as `s3cmd` and `Cyberduck`, support multipart object operations by default. However, you must configure your S3 client to be compatible with the following Active Archive System S3 restrictions on multipart object operations:

- Maximum object size is 5 TB.
- Minimum object size is 5 MB.
- Maximum number of parts is 10,000.

For more information on changing the settings in the client daemon configuration file, see *Managing Storage* in the *HGST Active Archive System Administration Guide*.

4.7.7 Examples

Sample Request

```
POST /c2c4dc85-38ac-4f89-aa29-16a19bbc3468?uploadId=BbyOjajJe30TSjDP7Bq4Z.rmTu_\\
pDwVOFBFi7tqHWwe5ZgUFMKGwZje2TQl4Zz_i4mBhjLD1vXxj8_c7rQC_BLwGQplJh_\\
bcOftTq0b.U0o4_F8lyRkYJ8qBKFFmNwhS HTTP/1.1
Content-Length: 243
Accept-Encoding: identity
Accept: */*
Host: s3_domain_name
x-amz-date: Fri, 13 Dec 2013 08:19:30 +0000
Content-Type: application/octet-stream
Authorization: authorization_string

<CompleteMultipartUpload><Part>
   <PartNumber>1</PartNumber>
   <ETag>a75ed575cda7730c7c14b40352da1555</ETag>
</Part>
<Part>
   <PartNumber>2</PartNumber>
   <ETag>1f778a8d73d57b0e162667b635631173</ETag>
</Part>
</CompleteMultipartUpload>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: vjq6W420mzRUGJcTPHSDTCDpYMbjba7RLXdhb0MSJzxaz5/cuLq6cHGFhAgPVfs/
x-amz-request-id: CCC7CBBC674808C5
Date: Fri, 13 Dec 2013 08:19:17 GMT
Content-Type: application/xml
Transfer-Encoding: chunked
Server: SystemID
```

Sample Response Body

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<CompleteMultipartUploadResult xmlns="http://s3_domain_name1/doc/2006-03-01/">
   <Location>http://s3_domain_name2/c2c4dc85-38ac-4f89-aa29-\\
        16a19bbc3468
   </Location>
   <Bucket>bucket_name</Bucket>
   <Key>c2c4dc85-38ac-4f89-aa29-16a19bbc3468</Key>
   <ETag>&quot;329454b9b991ccd9b34e0ba4ae0b5004-2&quot;</ETag>
</CompleteMultipartUploadResult>
```

4.7.8 Initiate Multipart Upload

4.7.8 Description

Initiate Multipart Upload starts a multipart upload and returns an upload ID and a storage object ID. The upload ID is used to associate all the parts in the specific multipart upload. A multipart upload request is stored with a key in the format: `mpoI_bucketID_objectName_uploadID`. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadInitiate.html.

Note:

Most S3 clients, such as `s3cmd` and `Cyberduck`, support multipart object operations by default. However, you must configure your S3 client to be compatible with the following Active Archive System S3 restrictions on multipart object operations:

- Maximum object size is 5 TB.
- Minimum object size is 5 MB.
- Maximum number of parts is 10,000.

For more information on changing the settings in the client daemon configuration file, see *Managing Storage* in the *HGST Active Archive System Administration Guide*.

4.7.8 Examples

Sample Request

```
POST http://bucket_name/example-object.iso?uploads HTTP/1.1
Host: s3_domain_name
Accept-Encoding: identity
content-length: 0
content-type: application/x-iso9660-image
Authorization: authorization_string
x-amz-date: Fri, 13 Dec 2013 08:19:16 +0000
content-encoding: UTF-8
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: JBP5qj+DsZcfrawymY/4UDTf2JKdYD0B/rmL+USGtVwg5AZKuu/icNAAiwqfEgX9
x-amz-request-id: 7F200D15AB4013C8
Date: Fri, 13 Dec 2013 08:19:03 GMT
Transfer-Encoding: chunked
Server: SystemID
```

Sample Response Body

```
<?xml version="1.0" encoding="UTF-8"?>
   <InitiateMultipartUploadResult xmlns="http://s3_domain_name/doc/2006-03-01/">
      <Bucket>bucket_name</Bucket>
      <Key>c2c4dc85-38ac-4f89-aa29-16a19bbc3468</Key>
      <UploadId>BbyOjajJe30TSjDP7Bq4Z.rmTu_pDwVOFBFi7tqHWwe5ZgUFMKGwZje2TQl4Zz_\\
           i4mBhjLD1vXxj8_c7rQC_BLwGQplJh_bcOftTq0b.U0o4_F8lyRkYJ8qBKFFmNwhS
```

```
      </UploadId>
   </InitiateMultipartUploadResult>
```

4.7.9 List Parts

4.7.9 Description

List Parts lists the parts that have been uploaded for a specific multipart upload. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadListParts.html.

4.7.9 Examples

Sample Request

The following code lists relevant multipart part entries.

```
GET /example-object?uploadId=BbyOjajJe30TSjDP7Bq4Z.rmTu_\\
pDwVOFBFi7tqHWwe5ZgUFMKGwZje2TQl4Zz_i4mBhjLD1vXxj8_\\
c7rQC_BLwGQplJh_bcOftTq0b.U0o4_F8lyRkYJ8qBKFFmNwhS&max-parts=2\\
&part-number-marker=1 HTTP/1.1
Host: s3_domain_name
Date: Fri, 13 Dec 2013 08:19:32 GMT
Authorization: authorization_string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Fri, 13 Dec 2013 08:19:19 GMT
Content-Length: 985
Connection: keep-alive
Server: SystemID
```

Sample Response Body

```
<?xml version="1.0" encoding="UTF-8"?>
<ListPartsResult xmlns="http://s3_domain_name/doc/2006-03-01/">
   <Bucket>bucket_name</Bucket>
   <Key>example-object</Key>
   <UploadId>BbyOjajJe30TSjDP7Bq4Z.rmTu_pDwVOFBFi7tqHWwe5ZgUFMKGwZje2TQl4Zz_\\
        i4mBhjLD1vXxj8_c7rQC_BLwGQplJh_bcOftTq0b.U0o4_F8lyRkYJ8qBKFFmNwhS
   </UploadId>
   <Initiator>
      <ID>arn:aws:iam::111122223333:user/some-user-11116a31-17b5-\\
         4fb7-9df5-b288870f11xx</ID>
      <DisplayName>umat-user-11116a31-17b5-4fb7-9df5-b288870f11xx</DisplayName>
   </Initiator>
   <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>someName</DisplayName>
   </Owner>
   <StorageClass>STANDARD</StorageClass>
   <PartNumberMarker>1</PartNumberMarker>
   <NextPartNumberMarker>3</NextPartNumberMarker>
   <MaxParts>2</MaxParts>
   <IsTruncated>true</IsTruncated>
   <Part>
      <PartNumber>2</PartNumber>
      <LastModified>2010-11-10T20:48:34.000Z</LastModified>
      <ETag>"7778aef83f66abc1fa1e8477f296d394"</ETag>
      <Size>10485760</Size>
   </Part>
```

```
   <Part>
      <PartNumber>3</PartNumber>
      <LastModified>2010-11-10T20:48:33.000Z</LastModified>
      <ETag>"aaaa18db4cc2f85cedef654fccc4a4x8"</ETag>
      <Size>10485760</Size>
   </Part>
</ListPartsResult>
```

4.7.10 Upload Part

4.7.10 Description

Upload Part uploads a part in a multipart upload. A multipart part has a one-to-one mapping with a *part*, which can be considered a subobject. The part is first uploaded and created, and then linked to the specific multipart part. A possible previous "old" part is overwritten (no longer accessible) and deleted. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadUploadPart.html.

Note:

Most S3 clients, such as `s3cmd` and `Cyberduck`, support multipart object operations by default. However, you must configure your S3 client to be compatible with the following Active Archive System S3 restrictions on multipart object operations:

- Maximum object size is 5 TB.
- Minimum object size is 5 MB.
- Maximum number of parts is 10,000.

For more information on changing the settings in the client daemon configuration file, see *Managing Storage* in the *HGST Active Archive System Administration Guide*.

4.7.10 Examples

Sample Request

```
PUT /c2c4dc85-38ac-4f89-aa29-16a19bbc3468?partNumber=1\\
&uploadId=BbyOjajJe30TSjDP7Bq4Z.rmTu_pDwVOFBFi7tqHWwe5ZgUFMKGwZje2TQl4Zz_\\
i4mBhjLD1vXxj8_c7rQC_BLwGQplJh_bcOftTq0b.U0o4_F8lyRkYJ8qBKFFmNwhS HTTP/1.1
Content-Length: 5242880
Accept-Encoding: identity
Accept: */*
Content-MD5: p17Vdc2ncwx8FLQDUtoVVQ==
Host: s3_domain_name
x-amz-date: Fri, 13 Dec 2013 08:19:18 +0000
Content-Type: application/octet-stream
Authorization: authorization_string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: BD6QQJVqotgk5pzONQl8xQRLfY7XIUF24n0PZScTlnS3CP4fMSmwm5yf958KYDNp
x-amz-request-id: F28B7459125E4943
Date: Fri, 13 Dec 2013 08:19:05 GMT
ETag: "a75ed575cda7730c7c14b40352da1555"
Content-Length: 0
Server: SystemID
```

4.7.11 Upload Part - Copy

4.7.11 Description

Upload Part - Copy uploads a part in a multipart upload by creating a copy of an object that is already stored in the Active Archive System. For more information, see http://docs.aws.amazon.com/AmazonS3/latest/API/mpUploadUploadPartCopy.html.

Note:

Most S3 clients, such as `s3cmd` and `Cyberduck`, support multipart object operations by default. However, you must configure your S3 client to be compatible with the following Active Archive System S3 restrictions on multipart object operations:

- Maximum object size is 5 TB.
- Minimum object size is 5 MB.
- Maximum number of parts is 10,000.

For more information on changing the settings in the client daemon configuration file, see *Managing Storage* in the *HGST Active Archive System Administration Guide*.

4.7.11 Examples

Sample Request

```
PUT /newobject?partNumber=2&uploadId=VCVsb2FkIElEIGZvciBlbZZpbmcncyBteS1tb3ZpZS5\\
tMnRzIHVwbG9hZR HTTP/1.1
Host: s3_domain_name
Date:  Fri, 13 Dec 2013 08:25:19 GMT
x-amz-copy-source: /source_bucketname/source_objectname
x-amz-copy-source-range:bytes=500-6291456
Authorization: authorization_string
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date:  Fri, 13 Dec 2013 08:25:19 GMT
Server: SystemID
```

Sample Response Body

```
<CopyPartResult>
   <LastModified>2008-01-29T08:22:00</LastModified>
   <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>
```

4.7.12 Request Headers for GET and HEAD Object

The GET Object and HEAD Object requests support `if-*` headers. These include:

- `If-Match`: return the object if its entity tag (ETag) is the same as the one specified, or return HTTP 412 (precondition failed, XML response).
- `If-Modified-Since`: return the object if it has been modified since the specified time, or return HTTP 304 (object not modified).
- `If-None-Match`: return the object if its entity tag (ETag) is different from the one specified, or return HTTP 304 (object not modified).
- `If-Unmodified-Since`: return the object if it has not been modified since the specified time, or return HTTP 412 (precondition failed, XML response).

The timestamps used for `If-Modified-Since` and `If-Unmodified-Since` must comply with the specifications of RFC2616.

The preferred timestamp is the RFC-822 (updated by RFC-1123): *`Day`*`, `*`DD Mon YYYY HH`*`:`*`MM`*`:`*`SS`*` GMT`.

The timestamp is case sensitive. Valid values are:

- *`Day`*: `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, `Sat`, `Sun`
- *`DD`* (*): two digits for the day of the month
- *`Mon`*: `Jan`, `Feb`, `Mar`, `Apr`, `May`, `Jun`, `Jul`, `Aug`, `Sep`, `Oct`, `Nov`, `Dec`
- *`YYYY`*: four digits for the year
- *`HH`*: two digits for the hour
- *`MM`*: two digits for the minutes
- *`SS`*: two digits for the seconds
- `GMT`: timestamp must be represented in GMT

> **Note:** (*)If you do not use two digits for the day of the month with an `If-Modified-Since` or `If-Unmodified-Since` request, you should receive an HTTP 304 or HTTP 412 response, respectively; however in Active Archive System, you get an HTTP 200 but with warnings in the client daemon log.

The implementation of these headers is according the RFC2626. More information about these header options can be found in the http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html, section 14.24, 14.25, 14.26, and 14.27.

You can also combine the following requests:

- `If-Match` and `If-Unmodified-Since`

If-Match	If-Unmodified-Since	Result
True	True	HTTP 200, object returned
True	False	XML Error 412 (precondition failed)
False	True	XML Error 412 (precondition failed)
False	False	XML Error 412 (precondition failed)

- `If-None-Match` and `If-Modified-Since`

If-None-Match	If-Modified-Since	Result
True	True	HTTP 200, object returned
True	False	HTTP 200, object returned
False	True	HTTP 200, object returned
False	False	HTTP 304 (not modified)

4.7.12.1 Examples

For an object created/modified on `Mon Jan 20 09:00:00 2014` with ETag `dbb126075a1d0b4ca64c3e2ae0159bc7`:

4.7.12.1 If-Match

- Matching ETag (HTTP 200):

```
curl -X GET "http://monty.s3.amazonws.com:7070/python" -H "if-match:
 dbb126075a1d0b4ca64c3e2ae0159bc7"
```

```
HTTP/1.1 200 OK
x-amz-id-2: V+INEFf6K34RtoqgQGfhDBiWo2oUauc8oh9Jpm2aA/XahYK8s3lt5s+pNGztkJh4
x-amz-request-id: 8A0EA6052595BDB2
```

```
Date: Fri, 14 Feb 2014 12:34:01 GMT
Last-Modified: Mon, Jan 20 09:00:00 2014 GMT
ETag: "dbb126075a1d0b4ca64c3e2ae0159bc7"
Accept-Ranges: bytes
Content-Type: application/octet-stream
Content-Length: 4
Server: AmazonS3
```

- Different ETag (HTTP 412 Precondition Failed XML Response):

```
curl -X GET "http://monty.s3.amazonws.com:7070/python" -H "if-match:
 dbb126075a1d0b4ca64c3e2ae0159xv3"
```

```
<Error>
   <Code>PreconditionFailed</Code>
   <Message>At least one of the pre-conditions you specified did  not hold</Message>
   <Condition>If-Match</Condition>
   <RequestId>F58D72A68D9CF0C2</RequestId>
   <HostId>xdvNIe0dyKfGOPu9lfwWcMbKKReL3oyO+vqRHtWI591rtqD7dZxHax02uOmr/HWM</HostId>
</Error>
```

4.7.12.1 If-None-Match

- Matching ETag:

```
curl -X GET "http://monty.s3.amazonws.com:7070/python" -H "if-match:
 dbb126075a1d0b4ca64c3e2ae0159bc7"
```

–> returns HTTP Error 304

```
HTTP/1.1 304 Not Modified
x-amz-id-2: T7kHsSgFbRy+DLcO2Y+jNkC/ZKVJN5KkKsAolS6Lmu9f0siX47sAdUEQsq/gzAr9
x-amz-request-id: 58449835A39BB796
Date: Fri, 14 Feb 2014 10:55:57 GMT
Last-Modified: Mon, Jan 20 09:00:00 2014 GMT
ETag: "dbb126075a1d0b4ca64c3e2ae0159bc7"
Server: AmazonS3
```

- Different ETag:

```
curl -X GET "http://monty.s3.amazonws.com:7070/python" -H "if-match:
 dbb126075a1d0b4ca64c3e2ae0159xv3"
```

```
HTTP/1.1 200 OK
x-amz-id-2: V+INEFf6K34RtoqgQGfhDBiWo2oUauc8oh9Jpm2aA/XahYK8s3lt5s+pNGztkJh4
x-amz-request-id: 8A0EA6052595BF2C
Date: Fri, 14 Feb 2014 12:34:01 GMT
Last-Modified: Mon, Jan 20 09:00:00 2014 GMT
ETag: "dbb126075a1d0b4ca64c3e2ae0159bc7"
Accept-Ranges: bytes
Content-Type: application/octet-stream
Content-Length: 4
Server: AmazonS3
```

4.7.12.1 If-Modified-Since

- Object modified since given date:

```
curl -X GET "http://monty.s3.amazonaws.com:7070/python" -H "if-modified-since: Tue
 Feb 18 08:38:47 2011"
```

```
HTTP/1.1 200 OK
```

```
x-amz-id-2: V+INEFf6K34RtoqgQGfhDBiWo2oUauc8oh9Jpm2aA/XahYK8s3lt5s+pNGztkJh4
x-amz-request-id: 8A0EA605259D345A
Date: Fri, 14 Feb 2014 12:34:01 GMT
Last-Modified: Mon, Jan 20 09:00:00 2014 GMT
ETag: "dbb126075a1d0b4ca64c3e2ae0159bc7"
Accept-Ranges: bytes
Content-Type: application/octet-stream
Content-Length: 4
Server: AmazonS3
```

- Object not modified since given date:

```
curl -X GET "http://monty.s3.amazonaws.com:7070/python" -H "if-modified-since: Tue
 Feb 18 08:38:47 2015
```

```
HTTP/1.1 304 Not Modified
x-amz-id-2: T7kHsSgFbRy+DLcO2Y+jNkC/ZKVJN5KkKsAolS6Lmu9f0siX47sAdUEQsq/gzAr9
x-amz-request-id: 58449835A39BB796
Date: Fri, 14 Feb 2014 10:55:57 GMT
Last-Modified: Mon, Jan 20 09:00:00 2014 GMT
ETag: "dbb126075a1d0b4ca64c3e2ae0159bc7"
Server: AmazonS3
```

4.7.12.1 If-Unmodified-Since

- Object modified since given date:

```
curl -X GET "http://monty.s3.amazonaws.com:7070/python" -H "if-unmodified-since: Tue
 Feb 18 09:40:53 2011"
```

```
<Error>
   <Code>PreconditionFailed</Code>
   <Message>At least one of the pre-conditions you specified did not hold</Message>
   <Condition>If-Unmodified-Since</Condition>
   <RequestId>F58D72A68D9CFDE1</RequestId>
   <HostId>xdvNIe0dyKfGOPu9lfwWcMbKKReL3oyO+vqRHtWI591rtqD7dZxHax02uOmr/HWM</HostId>
</Error>
```

- Object not modified since given date:

```
curl -X GET "http://monty.s3.amazonaws.com:7070/python" -H "if-unmodified-since: Tue
 Feb 18 09:38:47 2015"
```

```
HTTP/1.1 200 OK
x-amz-id-2: V+INEFf6K34RtoqgQGfhDBiWo2oUauc8oh9Jpm2aA/XahYK8s3lt5s+pNGztkJh4
x-amz-request-id: 8A0EA605259DCEF9
Date: Fri, 14 Feb 2014 13:34:01 GMT
Last-Modified: Mon, Jan 20 09:00:00 2014 GMT
ETag: "dbb126075a1d0b4ca64c3e2ae0159bc7"
Accept-Ranges: bytes
Content-Type: application/octet-stream
Content-Length: 4
Server: AmazonS3
```

4.7.12.1 Combined Header Requests

- `If-Match` combined with `If-Unmodified-Since`:

```
curl -X GET "http://monty.s3.amazonaws.com:7070/python" -H "if-unmodified-since: Tue
 Feb 18 09:40:53 2011" -H "if-match: dbb126075a1d0b4ca64c3e2ae0159bc6"
```

- `If-None-Match` combined with `If-Modified-Since`:

```
curl -X GET "http://monty.s3.amazonaws.com:7070/python" -H "if-modified-since: Tue
 Feb 18 09:40:53 2011" -H "if-none-match: dbb126075a1d0b4ca64c3e2ae0159bc6"
```

5 Capacity Reporting

Topics:

5.1 GET/HEAD

To retrieve a batched list of name space capacity reports, perform a GET on either:

- `/manage/capacity/namespace`
- `/s3manage/capacity/bucket`

To retrieve a batched list of syncstore IDs, perform a GET on either:

- `/manage/capacity/syncstore/`
- `/s3manage/capacity/syncstore/`

To retrieve the capacity report of a specific syncstore, perform a GET on either:

- `/manage/capacity/syncstore/<syncstore id>`
- `/s3manage/capacity/syncstore/<syncstore id>`

A HEAD request returns the same response as a GET request, but the server does not return a message-body in the response.

5.2 Get Syncstore Capacity Report

5.2 json
REQUEST `/manage/capacity/syncstore/<syncstore id>`

```
GET /manage/capacity/syncstore/<syncstore id> HTTP/1.1
Host: my.example.com
Accept: application/json
```

or:

```
GET /manage/capacity/syncstore/ <syncstore id>?meta=json HTTP/1.1
Host: my.example.com
```

RESPONSE `/manage/capacity/syncstore/<syncstore id>`

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision
Content-Type: application/json;charset=UTF-8
Content-Length: 58
{
"Id":"167da646596f4f83b6cb979cfd4c2afb",
"Count":5000
```

```
}
```

5.3 List Name Space Capacity Reports

By default, only the first 50 name space capacity reports are returned. This limit can be adjusted to X by appending the query parameter`limit=X` to the URL.

The maximum value for this limit is 1,023. Higher values will result in an HTTP 500 Error in at most 1000 entries for S3.

The following example URL can be used: `/manage/capacity/namespace/?limit=500`

In addition, the name space from which the list should start, can be specified using `marker=Y`. For example, the following URL could be used to list 4 name space capacity reports starting from name space `test`: `/manage/user?marker=test&limit=4` . An additional query parameter, `include_marker=false` can be added, which results in the marker being not included in the list.

5.3 json

REQUEST /manage/capacity/namespace/

```
GET /manage/capacity/namespace/ HTTP/1.1
Host: my.example.com
Date: _date_
Accept: application/json
```

or:

```
GET /manage/capacity/namespace/?list=json HTTP/1.1
Host: my.example.com
Date: _date_
```

RESPONSE /manage/capacity/namespace/

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: application/json;charset=UTF-8
Content-Length: 1201
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT

{"name":"NS1",
"start_date":1400154451.88,
"last_update":1400154452.01,
"current_policy_id_stats":[
    {"Id":"8e8b395a79d941e7b0d9a0e9476babeb",
        "Statistics":{
            "nr_objects_ok":4,
            "nr_superblocks_ok":4,
            "nr_objects_repair":0,
            "nr_superblocks_repair":0,
            "nr_objects_delete":0,
            "nr_superblocks_delete":0,
            "nr_objects_unverified":0,
            "nr_superblocks_unverified":0,
            "capacity_frontend_ok":30788,
            "capacity_frontend_repair":0,
            "capacity_frontend_delete":0,
            "capacity_frontend_objects_unverified":0,
```

```
            "capacity_frontend_superblocks_unverified":0,
            "disksafety_objects":[ {"disk safety":4, "count":4} ],
            "disksafety_superblocks":[ {"disk safety":4, "count":4} ],
            "disksafety_objects_offline":[ {"disk safety":4, "count":4} ],
            "disksafety_superblocks_offline":[ {"disk safety":4, "count":4} ],
            "disksafety_objects_decommissioned":[ {"disk safety":4, "count":4} ],
            "disksafety_superblocks_decommissioned":[ {"disk safety":4, "count":4} ],
            "policy_stats_hashtbl":{
                "nr" : {
                    "object" : {
                        "ok" : 4,
                        "all" : 4,
                        "repair" : 0,
                        "change policy": 0,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 4 },
                        "disk safety offline" : { "4" : 4 },
                        "disk safety decommissioned" : { "4" : 4 },
                        "needs conversion" : 0 },
                    "part" : {
                        "ok" : 4 ,
                        "all" : 4 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 4 },
                        "disk safety offline" : { "4" : 4 },
                        "disk safety decommissioned" : { "4" : 4 },
                        "needs conversion" : 0 },
                    "superblock" : {
                        "ok" : 4 ,
                        "all" : 4 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 4 },
                        "disk safety offline" : { "4" : 4 },
                        "disk safety decommissioned" : { "4" : 4 },
                        "needs conversion" : 0 } },
                    "capacity" : {
                        "object" : {
                        "ok" : 30788 ,
                        "all" : 30788 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 30788 },
                        "disk safety offline" : { "4" : 30788 },
                        "disk safety decommissioned" : { "4" : 30788 },
                        "needs conversion" : 0 },
                    "part" : {
                        "ok" : 30788 ,
                        "all" : 30788 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 30788 },
                        "disk safety offline" : { "4" : 30788 },
                        "disk safety decommissioned" : { "4" : 30788 },
                        "needs conversion" : 0 },
                    "superblock" : {
                        "ok" : 30788 ,
```

```
                        "all" : 30788 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 30788 },
                        "disk safety offline" : { "4" : 30788 },
                        "disk safety decommissioned" : { "4" : 30788 },
                        "needs conversion" : 0 }
                    }
                }
            }
        },
    { "Id" : "e10bf033967e46baa88909d521cdd963" ,
     "Statistics" :{
        "nr_objects_ok" : 5 ,
        "nr_superblocks_ok" : 201 ,
        "nr_objects_repair" : 0 ,
        "nr_superblocks_repair" : 0 ,
        "nr_objects_delete" : 0 ,
        "nr_superblocks_delete" : 0 ,
        "nr_objects_unverified" : 0 ,
        "nr_superblocks_unverified" : 0 ,
        "capacity_frontend_ok" : 13309621000 ,
        "capacity_frontend_repair" : 0 ,
        "capacity_frontend_delete" : 0 ,
        "capacity_frontend_objects_unverified" : 0 ,
        "capacity_frontend_superblocks_unverified" : 0 ,
        "disksafety_objects" :[ { "disk safety" : 4 ,"count" : 5 } ],
        "disksafety_superblocks" :[ {"disk safety" : 4 ,"count" : 201 } ],
        "disksafety_objects_offline" :[ {"disk safety" : 4 , "count" : 5 } ],
        "disksafety_superblocks_offline" :[ {"disk safety" : 4 , "count" : 201 } ],
        "disksafety_objects_decommissioned" :[ {"disk safety" : 4 , "count" : 5 } ],
        "disksafety_superblocks_decommissioned" :[ {"disk
safety" : 4 ,"count" : 201 } ],
        "policy_stats_hashtbl" :{
            "nr" : {
                "object" : {
                    "ok" : 5 ,
                    "all" : 5 ,
                    "repair" : 0 ,
                    "change policy" : 0 ,
                    "unverified" : 0 ,
                    "disk safety normal" : { "4" : 5 },
                    "disk safety offline" : { "4" : 5 },
                    "disk safety decommissioned" : { "4" : 5 },
                    "needs conversion" : 0 },
                "part" : {
                    "ok" : 5 ,
                    "all" : 5 ,
                    "repair" : 0 ,
                    "change policy" : 0 ,
                    "unverified" : 0 ,
                    "disk safety normal" : { "4" : 5 },
                    "disk safety offline" : { "4" : 5 },
                    "disk safety decommissioned" : { "4" : 5 },
                    "needs conversion" : 0 },
                "superblock" : {
                    "ok" : 201 ,
                    "all" : 201 ,
                    "repair" : 0 ,
                    "change policy" : 0 ,
                    "unverified" : 0 ,
```

```
                    "disk safety normal" : { "4" : 201 },
                    "disk safety offline" : { "4" : 201 },
                    "disk safety decommissioned" : { "4" : 201 },
                    "needs conversion" : 0 } },
                "capacity" : {
                    "object" : {
                        "ok" : 13309621000 ,
                        "all" : 13309621000 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 13309621000 },
                        "disk safety offline" : { "4" : 13309621000 },
                        "disk safety decommissioned" : { "4" : 13309621000 },
                        "needs conversion" : 0 },
                    "part" : {
                        "ok" : 13309621000 ,
                        "all" : 13309621000 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 13309621000 },
                        "disk safety offline" : { "4" : 13309621000 },
                        "disk safety decommissioned" : { "4" : 13309621000 },
                        "needs conversion" : 0 },
                    "superblock" : { "ok" : 13309621000 ,
                        "all" : 13309621000 ,
                        "repair" : 0 ,
                        "change policy" : 0 ,
                        "unverified" : 0 ,
                        "disk safety normal" : { "4" : 13309621000 },
                        "disk safety offline" : { "4" : 13309621000 },
                        "disk safety decommissioned" : { "4" : 13309621000 },

                        "needs conversion" : 0
                    }
                }
            }
        }
    } ],
    "old_policy_id_stats" :[ ],
    "object_name_length_stats" :{ "name" : "object_name_length_stats" ,
        "average" : 12.777778 ,
        "variance" : 18.839506 ,
        "count" : 9 ,
        "min" : 5.000000 ,
        "max" : 20.000000
    }
}
```


- `version`: version of the format in which the data is stored. This information is of no further use for the user.
- `name`: name of the monitored name space
- `start_date`: date and time when the last monitor crawl was started
- `last_update`: last update of the cached data
- `current_policy_id_stats`: dict with policy guid as key and value the following information:
  - `nr_objects_ok`: number of healthy objects
  - `nr_superblocks_ok`: number of healthy superblocks
  - `nr_objects_repair`: number of objects that have at least one superblock in repair status
  - `nr_superblocks_repair`: number of superblocks that need repair

- `nr_objects_delete`: number of objects that need to be deleted
- `nr_superblocks_delete`: number of superblocks that still need to be deleted
- `nr_objects_unverified`: number of unverified objects. Every object is verified each 180 days if it is still a healthy object. If this verification would not have taken place within these 180 days, the object is considered as unverified.
- `nr_superblocks_unverified`: number of unverified superblocks. This is similar as the unverified objects, but for superblocks.
- `capacity_frontend_ok`: the sum of the sizes of the objects put that are healthy, expressed in bytes
- `capacity_frontend_repair`: the sum of the sizes of the objects put that need to be repaired, expressed in bytes
- `capacity_frontend_delete`: the sum of the sizes of the objects put that still need to be deleted, expressed in bytes
- `capacity_frontend_objects_unverified`: the sum of the sizes of the objects put that is still unverified, expressed in bytes
- `capacity_frontend_superblocks_unverified`: the sum of the sizes of the superblocks put that is still unverified, expressed in bytes
- `disksafety_objects`: a dict with keys going from 'disksafety - spread width' to disksafety giving for all these values the number of objects that have that disk safety, taking into account the ABANDONED blockstores.
- `disksafety_superblocks`: a dict with keys going from 'disksafety - spread width' to disksafety giving for all these values the number of superblocks that have that disk safety, taking into account the ABANDONED blockstores
- `disksafety_objects_offline`: a dict with keys going from 'disksafety - spread width' to disksafety giving for all these values the number of objects that have that disk safety, taking into account the ABANDONED and OFFLINE blockstores
- `disksafety_superblocks_offline`: a dict with keys going from 'disksafety - spread width' to disksafety giving for all these values the number of superblocks that have that disk safety, taking into account the ABANDONED and OFFLINE blockstores
- `disksafety_objects_decommissioned`: a dict with keys going from 'disksafety - spread width' to disksafety giving for all these values the number of objects that have that disk safety, taking into account the ABANDONED, OFFLINE and DECOMMISSIONED blockstores
- `disksafety_superblocks_decommissioned`: a dict with keys going from 'disksafety - spread width' to disksafety giving for all these values the number of superblocks that have that disk safety, taking into account the ABANDONED,OFFLINE and DECOMMISSIONED blockstores
- `policy_stats_hashtbl`: overview with the statistics of all objects, parts, and superblocks with that policy as their target. This is a dict which contains two main sections, `nr` and `capacity`, respectively the number of items and the taken disk space of the items. These two main sections show the information by objects, parts, and superblocks.

  Note: The Active Archive System parts are not exactly the same as the S3 parts. In S3, there is a flat structure of the parts; in Active Archive System there is a tree structure, in which the main object (or storage object) consists of parts and each part can consist of child parts.

  — `nr`: gives you an overview of the number of items (`objects`, `parts`, `superblocks`)
  — `capacity`: gives you an overview of the taken disk space, split in `objects`, `parts`, and `superblocks`. The indicated disk spaces are expressed in bytes.
    - `ok`: all healthy items, which do not have either label REPAIR, CHANGE_POLICY, or UNVERIFIED
    - `all`: all items
    - `repair`: all items which need a repair action
    - `change policy`: items for which a new policy is selected, but which are not yet encoded with the new policy
    - `unverified`: items which are unverified. Every item is verified each 180 days if it is still a healthy object. If this verification would not have taken place within these 180 days, the item is considered as unverified.

  - `disk safety normal`: dict which takes the healthy blockstores into account. The disk safety is the key and the number of items/disk space is the value. This dict can contain multiple key/value pairs.
  - `disk safety offline`: dict which takes the offline blockstores into account.
  - `needs conversion`: remaining items/data volume that need conversion. This is a decreasing value and is ideally 0.
  - `disk safety decommissioned`: dict which takes the decommissioned blockstores into account.
- `old_policy_id_stats`: has been replaced by the parameter `change policy`. It was used to report on the statistics of objects in a name space which were not yet encoded with the target policy. This parameter is still in use for compatibility reasons.
- `object_name_length_stats`: statistics about the length of the object names in the name space.
  - `average`: average object name length
  - `count`: number of object to calculate the average length
  - `variance`: variance of the length
  - `max`: length of the longest object name
  - `min`: length of the shortest object name
  - `name`: name of the statistic

You can see that there are two policy ID statistics. This occurs when the name space uses a policy which has also an enabled small files policy.

5.4 List Syncstores

5.4 json
REQUEST /manage/capacity/syncstore/

```
GET /manage/capacity/syncstore/ HTTP/1.1
Host: my.example.com
Date: _date_
Accept: application/json
```

or:

```
GET /manage/capacity/syncstore/?list=json HTTP/1.1
Host: my.example.com
Date: _date_
```

RESPONSE /manage/capacity/syncstore

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: application/json;charset=UTF-8
Content-Length: 78
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT


[
 "160dfb46b98c481f9d18b3d98041017d",
 "825a2f8627334b3bb8f480f63ef7f663"
]
```

6 User Management

Topics:

The Active Archive System provides a RESTful API for user management.

6.1 Prerequisites

In order to create, list, or delete users through the API, you must first create a user that is specifically for administration. To do this, proceed as follows:

1. Log into the Management Node through SSH.
2. Enter the Q-Shell.

```
/opt/qbase3/qshell
```

3. Create an administrator user using the following command and replacing *username* and *password* with the desired credentials.

```
q.dss.manage.addUser("username","password")
```

4. Enable user management operations for the administrator user using the following command and replacing *username* with the username you specified in the previous step.

```
q.dss.manage.setPermissions("/manage","username",
['CREATE','LIST','READ','UPDATE','DELETE'])
```

6.2 Adding a User

To add a new user, the following fields must be specified:

- `name`: The user name. This must be unique.
- `secret`: The password.

The following fields are optional:

- `default-policy`: The default policy to use when this user creates a bucket.
- `default-small-files-policy`: The default small files policy to use when this user creates a bucket.
- `default-small-files-threshold`: The default threshold to use to select either the regular or the small files policy, when this user creates a bucket.

> **Note:** `default-small-files-policy` and `default-small-files-threshold` need to be specified together: either both or not at all. If an optional field is not specified, it is set to `None`.

6.2 http / text

REQUEST /manage/user

```
POST /manage/user HTTP/1.1
Host: my.example.com
X-Ampli-Name: "my_user_name"
X-Ampli-Secret: "my_secret"
X-Ampli-Default-Policy: "cadb732d2fdd4486be9b05dbe4b51d14"
X-Ampli-Default-Small-Files-Policy: "66da5c8f321d4dd28e3da264c42d04b7"
X-Ampli-Default-Small-Files-Threshold: 25000
Accept:plain/text
```

or:

```
POST /manage/user?meta=http HTTP/1.1
Host: my.example.com
X-Ampli-Name: "my_user-name"
X-Ampli-Secret: "my_secret"
X-Ampli-Default-Policy: "cadb732d2fdd4486be9b05dbe4b51d14"
X-Ampli-Default-Small-Files-Policy: "66da5c8f321d4dd28e3da264c42d04b7"
X-Ampli-Default-Small-Files-Threshold: 25000
```

RESPONSE /manage/user

```
HTTP/1.1 201 Created
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: text/plain;charset=UTF-8
Content-Length: 0
X-Ampli-Name: "my_user_name"
X-Ampli-Id: "36"
X-Ampli-Status: "ACTIVE"
X-Ampli-Seq: 0
X-Ampli-Num-S3-Namespaces: 0
X-Ampli-Default-Policy: "cadb732d2fdd4486be9b05dbe4b51d14"
X-Ampli-Default-Small-Files-Policy: "66da5c8f321d4dd28e3da264c42d04b7"
X-Ampli-Default-Small-Files-Threshold: 25000
```

6.2 json

REQUEST /manage/user

```
POST / manage / user HTTP / 1.1
Host : my . example . com
Content-Type: application/json
Accept : application / json
Content - Length : 205

{
"Name" : "my_user_name" ,
"Secret" : "my_secret" ,
"Default-Policy" : "cadb732d2fdd4486be9b05dbe4b51d14" ,
"Default-Small-Files-Policy" : "66da5c8f321d4dd28e3da264c42d04b7" ,
"Default-Small-Files-Threshold" : 10000
}
```

or

```
POST /manage/user?meta=json HTTP/1.1
Host: my.example.com
Content-Length: 205
```

```
{
"Name":"my_user_name",
"Secret":"my_secret",
"Default-Policy":"cadb732d2fdd4486be9b05dbe4b51d14",
"Default-Small-Files-Policy":"66da5c8f321d4dd28e3da264c42d04b7",
"Default-Small-Files-Threshold":10000
}
```

RESPONSE /manage/user

```
HTTP / 1.1 201 Created
Date : _date_
Server : Amplidata - AmpliStor / _revision_
Content - Type : application / json ; charset = UTF - 8
Content - Length : 245
{
"Name" : "my_user_name" ,
"Id" : "41" ,
"Status" : "ACTIVE" , "Seq" : 0 ,
"Num-S3-Namespaces" : 0 ,
"Default-Policy" : "cadb732d2fdd4486be9b05dbe4b51d14" ,
"Default-Small-Files-Policy" : "66da5c8f321d4dd28e3da264c42d04b7" ,
"Default-Small-Files-Threshold" : 10000
}
```

6.2 xml

REQUEST /manage/user

```
POST /manage/user HTTP/1.1
Host: my.example.com
Content-Type: application/xml
Accept: application/xml
Content-Length: 410

<?xml version="1.0" encoding="UTF-8"?>
<User xmlns="http://my.example.com/oss/api/rest/1/oss.xsd">
   <Name>my_user_name</Name>
   <Secret>my_secret</Secret>
   <Default-Policy>cadb732d2fdd4486be9b05dbe4b51d14</Default-Policy>
   <Default-Small-Files-Policy>66da5c8f321d4dd28e3da264c42d04b7</Default-Small-Files-
Policy>
   <Default-Small-Files-Threshold>50000</Default-Small-Files-Threshold>
</User>
```

or

```
POST /manage/user?meta=xml HTTP/1.1
Host: my.example.com
Content-Length: 410

<?xml version="1.0" encoding="UTF-8"?>
<User xmlns="http://my.example.com/oss/api/rest/1/oss.xsd">
   <Name>my_user_name</Name>
   <Secret>my_secret</Secret>
   <Default-Policy>cadb732d2fdd4486be9b05dbe4b51d14</Default-Policy>
   <Default-Small-Files-Policy>66da5c8f321d4dd28e3da264c42d04b7</Default-Small-Files-
Policy>
   <Default-Small-Files-Threshold>50000</Default-Small-Files-Threshold>
</User>
```

RESPONSE /manage/user

```
HTTP/1.1 201 Created
Date: _date_
Server: Amplidata-AmpliStor/_revision-
Content-Type: application/xml;charset=UTF-8
Content-Length: 403

<?xml version="1.0" encoding="UTF-8"?>
<User>
   <Name>my_user_name</Name>
   <Id>47</Id>
   <Status>ACTIVE</Status>
   <Seq>0</Seq>
   <Num-S3-Namespaces>0</Num-S3-Namespaces>
   <Default-Policy>cadb732d2fdd4486be9b05dbe4b51d14</Default-Policy>
   <Default-Small-Files-Policy>66da5c8f321d4dd28e3da264c42d04b7</Default-Small-Files-
Policy>
   <Default-Small-Files-Threshold>50000</Default-Small-Files-Threshold>
</User>
```

6.3 GET User

A GET can be performed on:

- `/manage/user`: returns a list of all user names
- `/manage/user/<username>`: returns the details of the user, specified by the user name
- `/manage/user_by_id/<id>`: returns the details of the user, specified by the user ID

Listing by ID is not supported.

If a user has certain optional fields which are currently set to None, they will be absent from the http, json, and xml output.

A HEAD request yields the same response as a GET request, except that the server does not return a message-body in the response.

6.3 http / text

REQUEST /manage/user/<username>

```
GET /manage/user/my_user_name HTTP/1.1
Host: my.example.com
Accept: text/plain
```

or:

```
GET /manage/user/my_user_name?meta=http HTTP/1.1
Host: my.example.com
```

RESPONSE /manage/user/<username>

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: text/plain;charset=UTF-8
Content-Length: 0
X-Ampli-Name: "my_user_name"
X-Ampli-Id: "48"
X-Ampli-Status: "ACTIVE"
X-Ampli-Seq: 0
```

```
X-Ampli-Num-S3-Namespaces: 0
X-Ampli-Default-Policy: "cadb732d2fdd4486be9b05dbe4b51d14"
X-Ampli-Default-Small-Files-Policy: "66da5c8f321d4dd28e3da264c42d04b7"
X-Ampli-Default-Small-Files-Threshold: 50000
```

6.3 json

REQUEST /manage/user/<username>

```
GET / manage / user / my_user_name HTTP / 1.1
Host : my . example . com
Accept : application / json
```

or:

```
GET /manage/user/my_user_name?meta=json HTTP/1.1Host: my.example.com
```

RESPONSE /manage/user/<username>

```
HTTP / 1.1 200 OK
Date : _date_
Server : Amplidata - AmpliStor / _revision_
Content - Type : application / json ; charset = UTF - 8
Content - Length : 245

{
"Name" : "my_user_name" ,
"Id" : "48" ,
"Status" : "ACTIVE" ,
"Seq" : 0 ,
"Num-S3-Namespaces" : 0 ,
"Default-Policy" : "cadb732d2fdd4486be9b05dbe4b51d14" ,
"Default-Small-Files-Policy" : "66da5c8f321d4dd28e3da264c42d04b7" ,
"Default-Small-Files-Threshold":50000
}
```

6.3 xml

REQUEST /manage/user/<username>

```
GET /manage/user/my_user_name HTTP/1.1
Host: my.example.com
Accept: application/xml
```

or:

```
GET /manage/user/my_user_name?meta=xml HTTP/1.1
Host: my.example.com
```

RESPONSE /manage/user/<username>

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: application/xml;charset=UTF-8
Content-Length: 402

<?xml version="1.0" encoding="UTF-8"?>
<User>
   <Name>my_user_name</Name>
   <Id>48</Id>
   <Status>ACTIVE</Status>
```

```
   <Seq>0</Seq>
   <Num-S3-Namespaces>0</Num-S3-Namespaces>
   <Default-Policy>cadb732d2fdd4486be9b05dbe4b51d14</Default-Policy>
   <Default-Small-Files-Policy>66da5c8f321d4dd28e3da264c42d04b7</Default-Small-Files-
Policy>
   <Default-Small-Files-Threshold>50000</Default-Small-Files-Threshold>
</User>
```

6.4 List Users

By default, only the first 50 users are returned. This limit can be adjusted to X by appending the query parameter "limit=X" to the URL. The maximum value for this limit is 1,000, higher values will result in at most 1000 entries. For example, the following URL could be used: `/manage/user?limit=500` .

In addition, the user from which the list should start can be specified using `marker=Y` . For example, the following URL could be used to list four users starting from user "test": `/manage/user?marker=test&limit=4` .

An additional query parameter, `include_marker=false` can be added, which results in the marker being not included in the list.

6.4 http / text

REQUEST /manage/user/

```
GET /manage/user/ HTTP/1.1
Host: my.example.com
Date: _date_
Accept: text/plain
```

or:

```
GET /manage/user/?list=http HTTP/1.1
Host: my.example.com
Date: _date_
```

RESPONSE /manage/user/

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: text/plain;charset=UTF-8
Content-Length: _length_

"fred"
"eric"
```

6.4 json

REQUEST /manage/user/

```
GET / manage / user / HTTP / 1.1
Host : my . example . com
Date : _date_
Accept : application / json
```

or:

```
GET /manage/user/?list=json HTTP/1.1
Host: my.example.com
Date: _date_
```

RESPONSE /manage/user/

```
HTTP / 1.1 200 OK
Date : _date_
Server : Amplidata - AmpliStor / _revision_
Content - Type : application / json ; charset = UTF - 8
Content - Length : _length_

[
"fred",
"eric"
]
```

6.4 xml
REQUEST /manage/user/

```
GET /manage/user/ HTTP/1.1
Host: my.example.com
Date: _date_
Accept: application/xml
```

or:

```
GET /manage/user/?list=xml HTTP/1.1
Host: my.example.com
Date: _date_
```

RESPONSE /manage/user/

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Type: application/xml;charset=UTF-8
Content-Length: _length_


<Users xmlns="http://my.example.com/oss/api/rest/1/oss.xsd">
   <Name>fred</Name>
   <Name>eric</Name>
</Users>
```

6.5 Modify a User

The following fields of an existing user can be modified:

- `name`
- `secret`
- `default-policy`
- `default-small-files-policy`
- `default-small-files-threshold`

Modifying one or more fields is accomplished by including the field name and the new value in the PUT request. Field names that do not occur in the http headers, json, and xml do not change.

The values for optional fields:

- `default-policy`
- `default-small-files-policy`

- `default-small-files-threshold`

can be removed from the user by setting them to None (http), null (json), or by specifying an empty XML element.

`Default-small-files-policy` and `default-small-files-threshold` need to be enabled or disabled together. If they are already enabled for a specific user, you can change the policy and the threshold separably.

The following examples change the user name, secret, default policy, and remove the optional `default-small-files-policy` and `default-small-files-threshold` fields.

The user to modify can be identified by user name (URL `/manage/user/`*`user_name`*) or by ID (URL `/manage/user_by_id/`*`id`*).

6.5 http / text

REQUEST `/manage/user/`*`username`*

```
PUT /manage/user/my_user_name HTTP/1.1
Host: my.example.com
X-Ampli-Name: "my_new_user_name"
X-Ampli-Secret: "my_new_secret"
X-Ampli-Default-Policy: "a618d783ba3e4723b5f6f3d5d7c3c2c1"
X-Ampli-Default-Small-Files-Policy: None
X-Ampli-Default-Small-Files-Threshold: None
Accept:plain/text
```

or

```
PUT /manage/user/my_user_name?meta=http HTTP/1.1
Host: my.example.com
X-Ampli-Name: "my_new_user_name"
X-Ampli-Secret: "my_new_secret"
X-Ampli-Default-Policy: "a618d783ba3e4723b5f6f3d5d7c3c2c1"
X-Ampli-Default-Small-Files-Policy: None
X-Ampli-Default-Small-Files-Threshold: None
```

RESPONSE `/manage/user/`*`username`*

```
HTTP/1.1 200 OK
Date: _date_
Server: Amplidata-AmpliStor/_release_
Content-Type: text/plain;charset=UTF-8
Content-Length: 0
X-Ampli-Name: "my_new_user_name"
X-Ampli-Id: "51"
X-Ampli-Status: "ACTIVE"
X-Ampli-Seq: 1
X-Ampli-Num-S3-Namespaces: 0
X-Ampli-Default-Policy: "a618d783ba3e4723b5f6f3d5d7c3c2c1"
```

6.5 json

REQUEST `/manage/user/`*`username`*

```
PUT / manage / user / my_user_name HTTP / 1.1
Host : my . example . com
Accept : application / json
Content - Type : application / json
Content - Length : 182

{
"Name" : "my_new_user_name" ,
"Secret" : "my_new_secret" ,
"Default-Policy" : "a618d783ba3e4723b5f6f3d5d7c3c2c1" ,
```

```
"Default-Small-Files-Policy" : null ,
"Default-Small-Files-Threshold" : null
}
```

or

```
PUT /manage/user/my_user_name?meta=json HTTP/1.1
Host: my.example.com
Content-Length: 182

{
"Name":"my_new_user_name",
"Secret":"my_new_secret",
"Default-Policy":"a618d783ba3e4723b5f6f3d5d7c3c2c1",
"Default-Small-Files-Policy":null,
"Default-Small-Files-Threshold":null
}
```

RESPONSE /manage/user/*username*

```
HTTP / 1.1 200 OK
Date : _date_
Server : Amplidata - AmpliStor / _release_
Content - Type : application / json ; charset = UTF - 8
Content - Length : 145

{
"Name" : "my_new_user_name" ,
"Id" : "52" ,
"Status" : "ACTIVE" ,
"Seq" : 1 ,
"Num-S3-Namespaces" : 0 ,
"Default-Policy" : "a618d783ba3e4723b5f6f3d5d7c3c2c1"
}
```

6.5 xml

REQUEST /manage/user/*username*

```
PUT /manage/user/my_user_name HTTP/1.1
Host: my.example.com
Content-Type: application/xml
Accept: application/xml
Content-Length: 324

<?xml version="1.0" encoding="UTF-8"?>
<User xmlns="http://my.example.com/oss/api/rest/1/oss.xsd">
   <Name>my_new_user_name </Name>
   <Secret>my_new_secret </Secret>
   <Default-Policy>a618d783ba3e4723b5f6f3d5d7c3c2c1</Default-Policy>
   <Default-Small-Files-Policy />
   <Default-Small-Files-Threshold />
</User>
```

or

```
PUT /manage/user/my_user_name?meta=xml HTTP/1.1
Host: my.example.com
Content-Length: 324

<?xml version="1.0" encoding="UTF-8"?>
<User xmlns="http://my.example.com/oss/api/rest/1/oss.xsd">
```

```
   <Name>my_new_user_name</Name>
   <Secret>my_new_secret</Secret>
   <Default-Policy>a618d783ba3e4723b5f6f3d5d7c3c2c1</Default-Policy>
   <Default-Small-Files-Policy />
   <Default-Small-Files-Threshold />
</User>
```

RESPONSE /manage/user/*username*

```
HTTP/1.1 200 OK
Date: Fri, 08 Mar 2013 13:24:02 GMT
Server: Amplidata-AmpliStor/_revision_
Content-Type: application/xml;charset=UTF-8
Content-Length: 246

<?xml version="1.0" encoding="UTF-8"?>
<User><Name>my_new_user_name</Name>
   <Id>53</Id>
   <Status>ACTIVE</Status>
   <Seq>1</Seq>
   <Num-S3-Namespaces>0</Num-S3-Namespaces>
   <Default-Policy>a618d783ba3e4723b5f6f3d5d7c3c2c1</Default-Policy>
</User>
```

6.6 Delete a User

The user to delete can be identified by:

- his user name (URL `/manage/user/<user name>`), or
- his ID (URL `/manage/user_by_id/<user id>`).

6.6 By User Name

Request /manage/user/<username>

```
DELETE /manage/user/my_user_name HTTP/1.1
Host: my.example.com
```

Response /manage/user/<username>

```
HTTP/1.1 204 No Content
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Length: 0
```

6.6 By ID

Request /manage/user/<username>

```
DELETE /manage/user_by_id/my_id HTTP/1.1
Host: my.example.com
```

Response /manage/user/<username>

```
HTTP/1.1 204 No Content
Date: _date_
Server: Amplidata-AmpliStor/_revision_
Content-Length: 0
```

6.7 Q-Shell Interface for User Management

6.7.1 Add a New User With User Name, Credentials, and Status

To add a new user, use the following code:

```
q.dss.manage.addUser(self,userName,credentials,\\
      status='ACTIVE',nodeIP='127.0.0.1',port=23510)
```

Parameter	Value	Explanation
userName	string	The user name of the user.
credentials	string	The secret keyword of the user.
status	string	The user status. Choose either ACTIVE (default) or INACTIVE.
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

6.7.2 Get User Information by Username

To get the user information for the user identified as *userName*, use the following code:

```
q.dss.manage.showUser(self,\\
      userName,showDeleted=False,nodeIP='127.0.0.1',\\
      port=23510)
```

Parameter	Value	Explanation
userName	string	The user name of the user.
showDeleted	bool	Show deleted users or not.
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

Note: Setting `showDeleted` to `True` also shows the user if it has status `DELETED`.

6.7.3 Getting User Information by ID

To get the user information by using its ID, use the following code:

```
q.dss.manage.showUserById(self, \\
      userId,showDeleted=False,nodeIP='127.0.0.1', \\
      port=23510)
```

Parameter	Value	Explanation
userId	int	The user ID of the user.
showDeleted	bool	Show deleted users or not.

Parameter	Value	Explanation
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

6.7.4 List All Users

To list all users, use the following code:

```
q.dss.manage.listUsers(self, \\
showDeleted=False,nodeIP='127.0.0.1', \\
port=23510)
```

Parameter	Value	Explanation
showDeleted	bool	Show deleted users (true) or not (false).
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

6.7.5 Modify User Information by Username

To modify a user by using its user name, use the following code:

```
q.dss.manage.changeUser(self, \\
      userName,credentials=None,status=None, \\
      nodeIP='127.0.0.1',port=23510)
```

Parameter	Value	Explanation
userName	string	The user name of the user.
credentials	string	The new secret of the user, None for no change.
status	string	The new status of the user, one of: ACTIVE, INACTIVE, None for no change.
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

6.7.6 Modify User Information by ID

To modify a user by using its ID, use the following code:

```
q.dss.manage.changeUserById(self,userId, \\
      userName=None,credentials=None, \\
```

```
      status=None,nodeIP='127.0.0.1',port=23510)
```

Parameter	Value	Explanation
userId	int	The user ID of the user.
userName	string	The new user name of the user, None for no change.
credentials	string	The new secret of the user, None for no change.
status	string	The new status of the user, one of: ACTIVE, INACTIVE, None for no change.
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

6.7.7 Delete a User

To delete a user, use the following code:

```
q.dss.manage.deleteUser(self,userName, \\
      nodeIP='127.0.0.1', \\
      port=23510)
```

Parameter	Value	Explanation
userName	string	The new user name of the user, None for no change.
nodeIP	string	The IP of the storage daemon to contact.
port	int	The port of the storage daemon to contact.

A S3 Error Codes and HTTP Status Codes

Topics:

A.1 S3 Error Codes

For more info about S3 error codes, see http://docs.amazonwebservices.com/AmazonS3/latest/API/ErrorResponses.html.

A.1 Error Responses

Error Code	Description	Http Status Code	DSS Exception
Invalid Bucket Name	The specified bucket is not valid.	400 Bad Request	Exc.Invalid_namespace _name
Metadata Too Large	Your metadata headers exceed the maximum allowed metadata size.	400 Bad Request	
Invalid Argument	Invalid Argument	400 Bad Request	
Too Many Buckets	You have attempted to create more buckets than allowed.	400 Bad Request	
Invalid Digest	The Content-MD5 you specified was an invalid.	400 Bad Request	
Entity Too Large	Your proposed upload exceeds the maximum allowed object size.	400 Bad Request	
Access Denied	Access Denied	403 Forbidden	None
Signature Does Not Match	The calculated request signature does not match the signature you provided. Check your AWS Secret Access Key and signing method.	403 Forbidden	
Invalid Access KeyId	The AWS Access Key Id you provided does not exist in our records.	403 Forbidden	
No Such Bucket	The specified bucket does not exist.	404 Not Found	Exc.Namespace_not _found
No Such Key	The specified key does not exist.	404 Not Found	Exc.Storage_object_not _found
Not Such Bucket Policy	The specified bucket does not have a bucket policy.	404 Not Found	
Method Not Allowed	The specified method is not allowed against this resource.	405 Method Not Allowed	

Error Code	Description	Http Status Code	DSS Exception
Bucket Already Exists	The requested bucket name is not available. The bucket name space is shared by all users of the system. Please select a different name and try again.	409 Conflict	Exc.Namespace_already_exists
Bucket Not Empty	The bucket you tried to delete is not empty.	409 Conflict	
Missing Content Length	You must provide the Content-Length HTTP header.	411 Length Required	
Internal Error	An internal error was encountered. Please try again.	500 Internal Server Error	
Not Implemented	A header you provided implies functionality that is not implemented.	501 Not Implemented	
Concurrent PUT Detected	Concurrent PUT request detected for identical file. First PUT request wins	XML error	Concurrent PUT on this resource detected

A.1 S3 Error Return Code Examples

If the bucket does not exist, the following error response is generated:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<Error>
   <Code>NoSuchBucket</Code>
   <Message>The specified bucket does not exist.</Message>
   <Resource></Resource>
   <RequestId></RequestId>
</Error>
```

If the key does not exist, the following error response is generated:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<Error>
   <Code>NoSuchKey</Code>
   <Message>The specified key does not exist.</Message>
   <Resource>fileX</Resource>
   <RequestId></RequestId>
</Error>
```

Note: A `requestId` is not returned.

If a concurrent PUT request is detected:

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n
<Error>
   <Code>ConcurrentPutDetected</Code>
   <Message>"Concurrent put on this resource detected."</Message><RequestId></
RequestId>
   <Resource>/bucket/object</Resource>
</Error>
```

A.2 HTTP Status Codes

The Active Archive System follows the standard HTTP/1.1 status code definitions. The following table describes a list of potential status codes:

Type	HTTP Description Message	HTTP/1.1 Status Code	Description
SUCCESS	Continue	100	The client SHOULD continue with its request.
SUCCESS	OK	200	The request was successfully executed.
SUCCESS	Object Created	201	The request was successfully executed and the object is created.
REDIRECTION	Not Modified	304	The precondition is not satisfied.
ERROR	Bad Request	400	The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.
ERROR	Unauthorized	401	The request failed because of lack of permissions.
ERROR	Forbidden	403	The Server understood the request, but refuses to fulfill it.
ERROR	Object Not Found	404	The server has not found anything matching the Request-URI.
ERROR	Conflict	409	Operation aborted because of a conflict.
ERROR	Precondition Failed	412	The precondition given in the request header field evaluated to false when tested on the server.
ERROR	Request Entity Too Large	413	The requested data (object, customer data, etc.) is larger than the server is able to process.
ERROR	Metastore <id> is full	500	No more free space on the given Metastore
ERROR	Internal Server Error	501	The server encountered an internal error.
FAILURE	HTTP Version Not Supported	505	The server does not support the HTTP protocol version

Type	HTTP Description Message	HTTP/1.1 Status Code	Description
			that was used in the request message.
ERROR	Insufficient Storage	507	No more free space on the blockstores

Index