Quantum

Scalar i6000

RESTful Web Services API

6-68185-01 Rev B

Quantum Scalar i6000 RESTful Web Services API, 6-68185-01 Rev B, December 2015, Product of USA.

Scalar i6000 RESTful Web Services Application Programming Interface

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 General

The Scalar i6000 Web Services application interface uses the REpresentational State Transfer (REST) architectural principles. REST is a software architecture style that builds distributed systems consisting of clients and server components. Clients initiate requests and servers process these requests and return responses, where the response can be a status or a representation of the resources being requested.

REST is protocol independent. The Scalar i6000 uses the Hypertext Transfer Protocol (HTTP) as the application protocol. For more information regarding REST, visit http://en.wikipedia.org/wiki/Representational_state_transfer.

The Scalar i6000 REST architecture principles are defined as follows:

1.2 Addressable Resources

Information and data is represented as resources which are addressed by a Uniform Resource Identifier (URI). The format of a URI is as follows:

protocol://host:port/resource?query=action&filter=10

The Scalar i6000 communication protocol is either http or https. The host is the library domain name or library IP address, and the port defines the port number on which the web service request is requested, 80 for http and 443 for https.

While the host and port represent the Scalar i6000 network location, the resource defines the web service resource request which consists of text separated by "/" characters to further define unique resource requests.

Optional query parameters are identified by the "?" character which separates the resource from the query strings. Query parameters are name value pairs delimited by the "&" character to allow multiple query string definitions.

1.3 Representation-Oriented Resources

Each resource can be represented in different formats. Different platforms require different formats; browsers use Hypertext Markup Language (HTML), JavaScript uses JavaScript Object Notation (JSON) and JAVA may require Extensible Markup Language (XML).

The HTTP communication protocol defines the representation in the message body of the request or response. The message body can return data in any format and the Content-Type header of the HTTP request and response informs the client or server of the message body format.

The Scalar i6000 typically uses text/plain, application/xml and application/json. Depending on the request, it also supports application/x-tar, application/octet-stream

and multipart/form-data for transferring binary or text data.

1.4 Uniformed, Constrained Request Interfaces

A small set of well-defined methods/operations manipulate library resources. Create, Read, Update, and Delete (CRUD) operations are performed via http methods POST, GET, PUT, and DELETE:

- POST is used to create a new resource, either permanently, or temporarily, such as requesting a robot operation. The request message body contains the details for the new resource to be created.
- GET is used to retrieve resource information.
- PUT is used to update/modify a resource. The request message body contains the information needed to update the resource.
- DELETE is used to remove an existing resource.

2. Request and Response Interface Description

2.1 General

As described earlier, clients use HTTP or HTTPS as the application protocol to make requests to the Scalar i6000 Web Services (WS) server. A client request identifies the resource URI, and any query parameters, and the Scalar i6000 responds with status and/or request data.

The Scalar i6000 defines a base URI which, by itself, provides library identification information, and also serves as the starting resource URI for all additional Web Services URI requests. The base URI is defined as follows:

HTTP(S)://LIBRARY-NAME-or-IP/aml

HTTP and HTTPS define unsecure or secure communication requests, respectively. The base URI identifies the library network location where library resources can be accessed. URIs are defined with intuitive naming conventions using a directory style structure.

2.2 URI Examples

To illustrate the use of Scalar i6000 URIs, consider the following URI to interact with logical library partition resources for a tape library with domain name *myLibrary*:

https://myLibrary/aml/partitions

The above URI references all configured partition resources. A particular partition resource is referenced with a specific partition name:

https://myLibrary/aml/partition/{name}

The “{name}” URI path template represents the name of the partition resource being requested.

To limit partition resource information, query strings are added to the URI. In the following example, you can get a list of storage segments in frame 2 that belong to the partition with the given ‘{name}’ LL1.

https://myLibrary/aml/partition/LL1/segments?type=storage&frame=2

2.3 Web Services Request Examples

The requesting client accesses and manipulates library resources with the HTTP methods POST, GET, PUT and DELETE. These methods are included in the HTTP request header.

To illustrate, a *GET http://mylibrary/aml/partitions* request results in the following HTTP header data:

GET http://mylibrary/aml/partitions HTTP/1.1
Host: user@company.com
User-Agent: Mozilla/5.0

Content-type: text/plain
Accept: application/xml, application/json

The library web server response message is similar and contains the HTTP version being used, a response code, a short message that explains the response code, a variable set of optional headers and an optional message body.

```
HTTP/1.1 200 OK
Content-Type: application/xml
Server: Jetty(7.9.2 v20120308)
Content-Length: 870

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:partitionList xmlns:ns2="http://automatedMediaLibrary/">
   <partition>
      <name>Test Partition</name>
      <type>1</type>
      <driveDomainType>6</driveDomainType>
      <storageSlotCount>300</storageSlotCount>
      <driveCount>2</driveCount>
      <ieSlotCount>12</ieSlotCount>
      <xieSlotCount>0</xieSlotCount>
      <ampExtensionsCount>0</ampExtensionsCount>
      <mediaCount>209</mediaCount>
      <barcodeReporting>4</barcodeReporting>
      <vendorId>1</vendorId>
      <productId>3</productId>
   </partition>
</ns2:partitionList>
```

The response body in the above example contains a list *(“<ns2:partitionList>” root element)* of partition resources represented as XML data. The partition resources are defined as data objects which represent the library’s logical library partition configuration. The example shows an XML response, but as explained earlier, the data format of the message body could have been JSON and in this case, due to the HTTP request header containing the statement **Accept: application/xml, application/json** the library will report the data as requested either in plain text or XML.

The response message identifies the response type in the response message body via the “Content Type” or “Media Type” statement as to what data format is reported. In this particular case the requested resource supports XML (default) as well as a JSON format, however per request and default response, the XML format is returned, identified via the statement **Content-Type: application/xml.**

The Scalar i6000 supports the following formats:

- text/plain,
- application/xml,
- application/json,
- application/octet-stream,
- multipart/form-data, and
- application/x-tar

2.4 Web Services Response Code Objects

Typically, requests that perform creation or deletion operations (POST, DELETE methods) will just receive a response code object as defined below:

```
<ns2:WSResultCode xmlns:ns2="http://automatedMediaLibrary/">
   <code>200</code>
   <description>OK</description>
   <summary>Operation Completed Successfully</summary>
   <action>Logout</action>
   <customCode>0</customCode>
</ns2:WSResultCode>
```

The Scalar i6000 supports the above WSResultCode object which contains the HTTP result code in the HTTP response. In the above case, the code "*200"* is reporting a successful request, a brief description of this code, such as "*OK"*, a summary of the result, such as "*Operation Completed Successfully*", the action that was requested, such as "*Logout*" and an optional custom code that can further describe the result in certain contexts.

While resource object requests may report a single object or a list of objects, some requests may return plain text that are not resource object representations but report single scalar values. An example would be a request to get the current date configured on the library:

GET HTTP://mylibrary/aml/system/dateTime/date

The following is returned in the response body:

"2012-08-23"

Typically a PUT request will return the modified resource.

2.4.1 Response Codes

The response header contains an HTTP status code that reports the status of the request. These status codes are part of the HTTP standard. The Scalar i6000 supports the following status codes:

200 OK

The request has succeeded. The information returned with the response is dependent on the method used in the request, for example:

GET an entity corresponding to the requested resource is sent in the response.

PUT returns the modified entity as if a GET had requested it.

POST an entity describing or containing the result of the action.

201 Created

The request has been fulfilled and resulted in a new resource being created.

Example: A request has been initiated, a new object has been created and the URI to the new resource can be found in the response header Location value.

202 Accepted

The request has been accepted, this is an asynchronous operation.

Example: A request has been initiated, but has not completed yet. Status polling will allow status testing to determine progress and eventual result codes.

400 Bad Request

The request could not be understood by the server due to malformed syntax.

Example: the URI and method are correct, but a request parameter is incorrect, or the JSON or XML is invalid.

401 Unauthorized

The request requires a valid user login session.

Example: The user session expired; a login is required again.

403 Forbidden

The request requires a valid login and the user needs to have access to the requested resource.

Example: Try to create a resource that already exists, or the request is not allowed for the current user role.

404 Not Found

The server has not found anything matching the Request-URI.

Example: The URI and media type are correct, but an identified resource is not found or no longer valid. The following URI “aml/partition/{name}” uses a path template “name”, which should represent a valid configured partition name.

405 Method Not Allowed

The method specified in the Request-Line is not allowed for the resource identified by the Request-URI.

Example: The URI is supported, but the method requested (GET, POST, PUT, or DELETE) is not supported.

415 Unsupported Media Type

The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

Example: The Content-Type (Media Type) header is not correct in the request message. That is, the WS request is expecting application/xml or application/json.

500 Internal Server Error

The server encountered an unexpected condition which prevented it from fulfilling the request.

Example: The HTTP request was determined to be valid, but the requested operation failed.

501 Not Implemented

The server does not recognize the request method, lacks the ability to fulfill the request or does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.

Example: The HTTP request is valid, but functionality has not been configured or implemented.

503 Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay MAY be indicated in a Retry-After header. If no Retry-After is given, the client SHOULD handle the response as it would for a 500 response.

Example: The request is not allowed due to a service user being logged in to perform maintenance.

2.5 Cookies

A successful login to the WS interface responds with a session cookie in the response header data. This cookie will need to be used on each successive http/https request for request authentication purposes after the initial login.

2.6 Cache Controls

All responses have cache controls turned off. No caching mechanisms are provided through the WS interface.

3. Web Services Application Programming Interface

3.1 Resource Tables

All library resource URIs are defined within tables in the following sections. A description precedes each table and summarizes the resource and/or what the interface provides. It may also describe some differences between supported library models and whether the resource is supported by a particular library model. Table entries provide information as follows:

- the URI to a particular resource,
- the supported CRUD methods, GET, POST, PUT and DELETE,
- request data objects (XML or JSON), used with most PUT and POST methods,
- response data objects, resources, primarily represented in XML and JSON,
- the Content-type (Media Types) supported for that resource,
- any Location URIs for newly created objects,
- any parameter query strings, and
- the possible response codes.

3.1.1 Table Entry Explanations

3.1.1.1 URI

URI	**aml/drives**

The first row of a table is the URI. The URI is a unique identifier to a resource or a list of resources. The above *aml/drives* URI points to configured drive resources. When a client makes a request, it will use this partial URI to build the full URI that is needed to complete the request. For example, if a client wanted to GET a list of drives configured in a library whose domain name is *TEST*, the request URI would be:
http://TEST/aml/drives

3.1.1.2 Method

Method	**POST, GET, PUT or DELETE**

The Method row lists the available request method(s) for the URI. Each URI may support all or a subset of the defined CRUD methods.

3.1.1.3 Request/Response Header

Request Header	**Content-Type: text/plain, application/xml, application/json, multipart/form-data**
Response Header	**Content-Type: application/xml, application/json, application/octet-stream, application/x-tar**

The Request Header and Response Header fields describe what data formats are expected or reported.

3.1.1.3.1 Accept

The Accept header is used in an HTTP request to inform the server what Content-Type it expects in the response body. The library web server must support the requested Content-Type, of course, and the 'Header Response' table row specifies the supported content types. The majority of the requests support the Content-types 'application/xml' and 'application/json' where the default is 'application/xml'.

3.1.1.3.2 Content-Type/Media Type

The Content-Type defines the content format contained in the body of the HTTP request or response. The Media Type for **Request**s only apply to POST and PUT Methods and the following types are supported:

- **text/plain** – The client has inserted some plain text in the body of the request. This text is user to indicate some change that needs to be made to a resource. An example of this would be to change a partitions mode online/offline. If the client wanted to take a partition offline, they would insert a "2" in the body of the request (see Table 148: GET aml/partition/{name}/mode).
- **application/xml** – The content of the request body is Extensible Markup Language (XML) and this XML describes a resource as an object. This is used to update an existing resource or create a new resource.
- **application/json** – The content of the request is JavaScript Object Notation (JSON). The JSON describes the resource that needs to be update or created. When a Content-Type Request is defined as "application/xml, application/json" this means that the Web Service server is expecting XML since it is reported first, but it can also accept JSON. If the client sends the resource representation as JSON they need to tell the server to expect JSON. This is done by adding the 'Content-Type: application/json' header to the request. Most of the URI interfaces support both XML and JSON.
- **multipart/form-data** – This Content-Type is used to send file binary data to the Web Server. Form data is made of a key, value pair, where the key is an identifier and the value is the file data. For example, this type is currently used to upload library and drive firmware to the library (see Table 326: GET aml/system/software).

The Content-Type for **Response**s apply to all request methods and the following types are supported:

- **text/plain** – The client HTTP response body will contain text. This is typically used to report a specific property of a resource. For example, to find the mode of a partition the response body will contain a "1" or "2" (Online/Offline) (see Table 148: GET aml/partition/{name}/mode).
- **application/xml** – The content of the response body is XML and this XML represents the resource(s) as an object or the WSResultCode.
- **application/json** – The content of the response is JavaScript Object Notation (JSON). The requested resource is represented as a JSON object. When a Content-

Type Response is defined as “application/xml, application/json” this means that the Web Service response will be XML by default, since it is listed first. If the client wants the response to be represented as a JSON object then the client must add ‘Accept: application/json’ to the header of the HTTP request.

- **application/octet-stream** – The response body contains byte data, this could be readable text data or binary date.
- **application/x-tar** – The response body contains a compressed tar file.

3.1.1.3.3 Location

This response header field is used to convey the URI of a newly created resource and is typically used in conjunction with a HTTP response status code of 201. For example, if a user creates a new partition the response header would contain a location reference like “Location: http://library/partition/LL1” where ‘LL1’ is the name of the new partition.

3.1.1.3.4 Content-Disposition

This response header field is used to notify the client, specifically a browser client, that the response will contain an attachment. This response header will trigger the browser to use its default mechanism to save any file attachments/downloads. This header is typically used when the client browser needs to save library data. This field is mainly listed when save query parameters are specified. The format of the header field is “Content-Disposition: attachment; filename=the file name to apply for the save operation”. If the ‘save=name’ name/value pair is not specified, a default file name will be applied.

3.1.1.4 Parameters

Parameters	

Parameters are primarily user to filter the response from a GET request. The client sends supported query parameters as part of the URI request.

For example, to filter the aml/drives URI to return available unassigned drives only (drives currently not owned by a partition), the following request applies:

GET http://TEST/aml/drive**?status=available**

Multiple query parameters can be used if the URI interface supports them. In the case of the aml/media URI interface, to filter media for a particular partition, such as LL1, and only report media that are in drives, the following request applies:

http://TEST/aml/media**?partition=LL1&location=drive**

3.1.1.5 Response Code

Response Code	Supported HTTP Status codes

The Response Code identifies all supported command status codes for the request.

Note: Response codes *400, 401, 405, 415 or 500 are not listed separately here since any interface request can encounter such HTTP status code.*

3.1.1.6 Request Data

Request Data	

The Request Data reports the resource object or singleton data expected by the library web server in the body of the HTTP request.

3.1.1.7 Response Data

Response Data	

The Response Data details the data returned in the body of the HTTP response.

4. Library Resources

4.1 Overview

Typically, library command requests require a valid user login. Operations are supported based on login user role, such that any user can perform read operations, with few exception for user ID retrieval, and admin and service users will be able to perform all read, as well as create, update and delete operations.

To allow library discovery and frequent status monitoring, a read of the “aml” resource (see Table 1) allows library detection without valid login sessions (no authentication needed).

4.2 Web Services API Request Summary

Table 1: GET aml/

Description: Retrieve the library ping resource. The ping can be requested without logging in to the web services server (no authentication is needed). The ping can be used to discover what libraries are on you network.

URI	***aml/***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 128: ping

4.2.1 Access Groups, host, drive and partition access.

The following interfaces under, **”aml/access”,** provide the ability to control what hosts can access which drives and partitions configured in the library. By default all drives and partitions can be accessed by all hosts connected to the SAN. However, Quantum-branded libraries support a *Path Failover/Native Storage Networking* (SNW) license which once applied to a drive, prevents device access and requires access configurations to define which hosts may access the drive or partition (control path).

To grant host access to a device (drive or partition) an access group resource must first be created (see Table 3: GET aml/access/groups). Once groups are defined, hosts musts be added to an access group so that access can then be assigned to devices (see Table 17: GET aml/access/hosts and Table 12: POST aml/access/group/{name}/hosts).

NOTE: A host can only belong to a single access group.

Drive devices and partitions (hosted by a control path drive) are added to the access groups to define the host access to the drive or partition (see Table 2: GET aml/access/devices and Table 7: POST aml/access/group/{name}/device).

Table 2: GET aml/access/devices

Description: Retrieve a list of configured Access Device resources that could be added to an Access Group.

An access device is a drive or partition.

For a drive access device, the drive must be HP LTO 5 or greater, belong to a partition, have an SNW license applied, drive must be in P2P/Fabric mode and be connected to an Ethernet Expansion Blade.

For a partition access device, the partition must have a control path drive and the drive must have the above prerequisites.

URI	***aml/access/devices***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 1: accessDeviceList

Table 3: GET aml/access/groups

Description: Retrieve a list of Access Group resources.

URI	***aml/access/groups***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 3: accessGroupList

Table 4: GET aml/access/group/{name}

Description: Retrieve the Access Group resources whose name is given by the URI path template “name”.

URI	*aml/access/group/{name}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 5: POST aml/access/group/{name}

Description: Create a new Access Group with the name given by URI path template "name". To create a new access group with name "AG3", use the following: aml/access/group/AG3.

URI	*aml/access/group/{name}*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 6: DELETE aml/access/group/{name}

Description: Delete Access Group with the name given by URI path template "name".

URI	*aml/access/group/{name}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 7: POST aml/access/group/{name}/device

Description: Add a new access device to the access group given by URI path template "name". The <access> element must be set to true, otherwise access will not be

granted on that port.

A partition accessDevice (type=2) is really a drive that is configured as a control path drive (see Table 141: GET aml/partition/{name}/controlPath).

The following example would allow access on port 1 for the drive with serial number F001396025 to all hosts that belong to the given access group.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accessDevice xmlns:ns2="http://automatedMediaLibrary/">
  <serialNumber>F001396025</serialNumber>
  <type>1</type>
  <port>
    <id>1<//id>
    <access>true</access>
  </port>
</ns2:accessDevice>
```

URI	***aml/access/group/{name}/device***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 2: accessDevice
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 8: PUT aml/access/group/{name}/device

Description: Update access device that belongs to the access group given by URI path template “name”.

The example below modifies access so the first port on the device will no longer be seen and the second port will be seen by all hosts in the access group.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accessDevice xmlns:ns2="http://automatedMediaLibrary/">
  <serialNumber>F001396025</serialNumber>
  <type>1</type>
  <port>
    <id>1</id>
    <access>false</access>
  </port>
  <port>
    <id>2</id>
    <access>true</access>
  </port>
</ns2:accessDevice>
```

URI	*aml/access/group/{name}/device*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 2: accessDevice
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 9: DELETE aml/access/group/{name}/device/{serialNumber}

Description: Remove the access device whose serial number is given by URI path template “serialNumber” from the access group given by URI path template “name”. This access device (drive or partition) will no longer be seen by hosts belonging to this access group.

URI	*aml/access/group/{name}/device/{serialNumber}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 10: GET aml/access/group/{name}/devices

Description: Retrieve the access devices configured for the access group whose name is given by URI path template “name”.

URI	*aml/access/group/{name}/devices*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 1: accessDeviceList

Table 11: POST aml/access/group/{name}/devices

Description: Add the access device list to the access group whose name is given by URI path template “name”.

None of the access devices must already belong to the access group.

Note: Remember that access is on a port basis. So the access element of each port must be set to true to grant access to each port on the device. Not all devices have multiple ports.

URI	***aml/access/group/{name}/devices***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 1: accessDeviceList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 12: POST aml/access/group/{name}/hosts

Description: Add a host to the Access Group resource whose name is given by the URI path template “name”.

The only element required in the host object is WWPN all other elements will be ignored.

URI	***aml/access/group/{name}/hosts***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 87: host
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 13: DELETE aml/access/group/{name}/host/{WWPN}

Description: Delete a host whose WWPN is given by the URI path template “WWPN” from the Access Group resource whose name is given by the URI path template “name”.

To delete host with WWPN “1234ABCD:1234ABCD” from Access Group “AG2” use the following URI:

aml/access/group/AG2/host/1234abcd:1234abcd

URI	*aml/access/group/{name}/host/{WWPN}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 4: accessGroup

Table 14: GET aml/access/host/{WWPN}

Description: Retrieve the host resource whose WWPN is given by URI path template "WWPN". To get the host with WWPN "1234ABCD:1234ABCD" use the following:

aml/access/host/1234ABCD:1234ABCD

URI	*aml/access/host/{WWPN}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 87: host

Table 15: PUT aml/access/host/{WWPN}

Description: Update the host resource whose WWPN is given by URI path template "WWPN". This would be primarily user to change the host name and host type. The WWPN element is required to find the host that needs to be updated.

URI	*aml/access/host/{WWPN}*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 87: host
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 87: host

Table 16: DELETE aml/access/host/{WWPN}

Description: Delete the host whose WWPN is given by the URI path template “WWPN”. This will remove the host from the libraries host cached data. You cannot delete a host that is online (mode = 1).

URI	***aml/access/host/{WWPN}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 17: GET aml/access/hosts

Description: Retrieve a list of host resources that were seen on the SAN from configure tape drives that are connected to the SAN.

URI	***aml/access/hosts***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 86: hostList

Table 18: POST aml/access/hosts

Description: Create a new host resource.

Note: This functionality is used primarily for testing purposes.

URI	***aml/access/hosts***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 87: host
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json

Response Data	See Figure 87: host

Table 19: GET aml/access/licenses

Description: Get a list of drives that have an SNW license or are eligible to have an SNW license applied.

An eligible drive must have the following settings:

Be owned by a partition

Note: Most features that use an SNW license, require the drive to have the following:

1. Be attached to an Ethernet Expansion Blade (EEB).
2. Be an HP or IBM LTO5 drive type generation or greater.
3. The drive ports must also be configured in point to point mode.

A drive that has an SNW license applied to it will have the drive.settings.license element with a value of 11.

Note: This interface applies to Quantum branded libraries only.

URI	***aml/access/licenses***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 26: driveList

Table 20: PUT aml/access/licenses

Description: Modify the SNW licenses for the drives in the requested driveList. The only drive elements that are required are the physicalSerialNumber or logicalSerialNumber and the settings.license.

To apply a license set the license element value to 11 and to remove a license, set the license element to an empty string.

The example below applies a license to drive with serial number F001396043 and removes a license from the drive with serial number F00139603D.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList   xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F001396043</logicalSerialNumber>
```

```
      <settings>
        <license>11</license>
      </settings>
    </drive>
    <drive>
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
        <license></license>
      </settings>
    </drive>
</ns2:driveList>
```

When you apply an SNW license to a drive then that drive will no longer be seen on the SAN. You must configure that drive in an Access Group before hosts can see the drive.

URI	***aml/access/licenses***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 26: driveList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 26: driveList

Table 21: GET aml/access/license/{serialNumber}

Description: Retrieve the drive with serial number given by the URI path template "serialNumber". This interface would be used to determine if the drive has an SNW license. Unlike the interface 'Table 19: GET aml/access/licenses' the drive returned may not be eligible for an SNW license.

URI	***aml/access/license/{serialNumber}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 27: drive

Table 22: POST aml/access/license/{serialNumber}

Description: Apply an SNW license to the drive with the serial number given by the URI path template "serialNumber".

URI	*aml/access/license/{serialNumber}*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/access/license/{serialNumber}
Response Data	See Figure 172: WSResultCode

Table 23: DELETE aml/access/license/{serialNumber}

Description: Remove an SNW license from the drive with the serial number given by the URI path template “serialNumber”.

You cannot remove an SNW license from a drive if it is currently configured for Data Path Failover or Control Path Failover.

URI	*aml/access/license/{serialNumber}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 24: GET aml/devices/blades/ethernet

Description: Retrieve a list of ethernetExpansionBlade (Ethernet Expansion Blades, or EEBs) resources. EEBs provide the option for Ethernet connectivity to each LTO-5 or LTO-6 drive (for MCB-to-drive communication purposes only). The EEB provides a control path to the drive for commands as well as facilitates taking drive logs and downloading drive firmware.

URI	*aml/devices/blades/ethernet*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json

Response Data	See Figure 64: ethernetExpansionBladeList

Table 25: GET aml/devices/blade/ethernet/{serialNumber}

Description: Retrieve an ethernetExpansionBlade resource whose serial number is given by the URI path template “serialNumber”.

URI	***aml/devices/blade/ethernet/{serialNumber}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 65: ethernetExpansionBlade

Table 26: GET aml/devices/blade/ethernet/{serialNumber}/operations/identify

Description: Retrieve the EEB identify task whose serial number is given by URI path template “serialNumber”.

URI	***aml/devices/blade/ethernet/{serialNumber}/identify***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 27: POST aml/devices/blade/ethernet/{serialNumber}/operations/identify

Description: Start the EEB identify task whose serial number is given by URI path template “serialNumber”. This starts the status LED on the EEB to flash rapidly.

URI	***aml/devices/blade/ethernet/{serialNumber}/identify***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404

Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 28: DELETE aml/devices/blade/ethernet/{serialNumber}/operations/identify

Description: Stop the EEB identify task whose serial number is given by URI path template “serialNumber”. This stops the status LED on the EEB from flashing rapidly.

URI	***aml/devices/blade/ethernet/{serialNumber}/identify***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 29: GET aml/devices/blades/fibreChannel

Description: Retrieve a list of fcBlade (FC IO Blades) resources. There is one Fibre Channel (FC) I/O blade type supported: 7404 that auto-negotiates up to 4 Gbps. The 7404 FC I/O blade has an embedded controller that provides connectivity and features that enhance the performance and reliability of tape operations. It also provides two host communication ports and four connection ports to drives.

URI	***aml/devices/blades/fibreChannel***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 71: fcBladeList

Table 30: GET aml/devices/blade/fibreChannel/{serialNumber}

Description: Retrieve the FC IO Blade whose serial number is given by URI path template “serialNumber”.

URI	***aml/devices/blade/fibreChannel/{serialNumber}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A

Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 72: fcBlade

Table 31: GET aml/devices/blade/fibreChannel/{serialNumber}/operations/identify

Description: Retrieve the FC IO Blade identify task whose serial number is given by URI path template “serialNumber”.

URI	***aml/devices/blade/fibreChannel/{serialNumber}/identify***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 32: POST aml/devices/blade/fibreChannel/{serialNumber}/operations/identify

Description: Start the FC IO Blade identify task whose serial number is given by URI path template “serialNumber”. This starts the status LED on the IO Blade to flash rapidly.

URI	***aml/devices/blade/fibreChannel/{serialNumber}/identify***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/devices/blade/fibreChannel/{serialNumber}/operations/identify
Response Data	See Figure 161: task

Table 33: DELETE aml/devices/blade/fibreChannel/{serialNumber}/operations/identify

Description: Stop the FC IO Blade identify task whose serial number is given by URI path template “serialNumber”. This stops the status LED on the IO Blade from flashing rapidly.

URI	*aml/devices/blade/fibreChannel/{serialNumber}/identify*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 34: GET aml/devices/blade/fibreChannel/{serialNumber}/operations/powerCycle

Description: Retrieve the FC IO Blade power cycle task whose serial number is given by URI path template “serialNumber”.

URI	*aml/devices/blade/fibreChannel/{serialNumber}/powerCycle*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 35: POST aml/devices/blade/fibreChannel/{serialNumber}/operations/powerCycle

Description: Start the FC IO Blade power cycle task whose serial number is given by URI path template “serialNumber”. This operation will power cycle the IO Blade.

URI	*aml/devices/blade/fibreChannel/{serialNumber}/powerCycle*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/devices/blade/fibreChannel/{serialNumber}/operations/powerCycle
Response Data	See Figure 161: task

Table 36: GET aml/devices/blade/fibreChannel/{serialNumber}/operations/reboot

Description: Retrieve the FC IO Blade reboot task whose serial number is given by URI path template “serialNumber”.

URI	***aml/devices/blade/fibreChannel/{serialNumber}/reboot***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 37: POST aml/devices/blade/fibreChannel/{serialNumber}/operations/reboot

Description: Start the FC IO Blade reboot task whose serial number is given by URI path template “serialNumber”. This operation will reboot the IO Blade.

URI	***aml/devices/blade/fibreChannel/{serialNumber}/reboot***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/devices/blade/fibreChannel/{serialNumber}/operations/reboot
Response Data	See Figure 161: task

Table 38: GET aml/devices/blades/library

Description: Get the library controller blade (MCB) list resource.

URI	***aml/devices/blades/library***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200,
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 93: libraryControlBladeList

Table 39: GET aml/devices/blade/library/{serialNumber}

Description: Get the library controller blade resource whose serial number is given by the URI path template “serialNumber”.

URI	*aml/devices/blade/library/{serialNumber}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 94: libraryControlBlade

Table 40: GET aml/devices/ieStations

Description: Retrieve a list of ieStation resources. I/E stations enable you to import and export cartridges without interrupting normal library operation. There are two types of I/E stations on Scalar i6k libraries: 24-slot I/E stations and 72-slot I/E stations.

URI	*aml/devices/ieStations*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 88: ieStationList

Table 41: GET aml/devices/ieStation/{number}

Description: Retrieve the ieStation resource whose number is given by URI path template “number”.

URI	*aml/devices/ieStation/{number}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json

Response Data	See Figure 88: ieStation

Table 42: GET aml/devices/ieStation/{number}/lock

Description: Retrieve the ieStation resource locked status whose number is given by the URI path template “number”.

The valid status are 1 (locked) or 2 (Unlocked).

URI	***aml/devices/ieStation/{number}/lock***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	1 or 2

Table 43: PUT aml/devices/ieStation/{number}/lock

Description: Update the ieStation resource locked status whose number is given by the URI path template “number”.

The valid lock values are 1 (locked) or 2 (Unlocked).

URI	***aml/devices/ieStation/{number}/lock***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 (lock) or 2 (unlock)
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1 or 2

Table 44: GET aml/devices/robots

Description: Get a list of robot object resources.

URI	***aml/devices/robots***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 147: robotList

Table 45: GET aml/devices/robot/{name}

Description: Get the robot resource whose name is given by the URI path template "name".

URI	***aml/devices/robot/{name}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 148: robot

Table 46: GET aml/devices/robot/{name}/state

Description: Get the robots state whose name is given by the URI path template "name".

URI	***aml/devices/robot/{name}/state***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1 (varied On) or 2 (Varied Off)

Table 47: PUT aml/devices/robot/{name}/state

Description: Update the robots state whose name is given by the URI path template "name".

URI	***aml/devices/robot/{name}/state***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 (varied On) or 2 (Varied Off)
Response Codes	200, 403, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1 (varied On) or 2 (Varied Off)

Table 48: GET aml/devices/towers

Description: Retrieve the list of tower resources. Towers are high-density expansion modules (HDEM). These modules have larger storage capacities making them ideal for libraries where space is an issue.

URI	***aml/devices/towers***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 165: towerList

Table 49: GET aml/devices/tower/{id}

Description: Retrieve the tower resources whose id is given by the URI path template "id".

URI	***aml/devices/tower/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 166: tower

Table 50: POST aml/devices/tower/{id}/identify

Description: Identify the tower resource whose id is given by the URI path template "id". The status LED on the rear tower door will flash rapidly for about one minute.

URI	***aml/devices/tower/{id}/identify***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 51: GET aml/devices/tower/{id}/mode

Description: Retrieve the tower resources mode whose id is given by the URI path template “id”.

URI	***aml/devices/tower/{id}/mode***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1 (Online) or 2 (Offline)

Table 52: PUT aml/devices/tower/{id}/mode

Description: Update the tower resources mode whose id is given by the URI path template “id”.

URI	***aml/devices/tower/{id}/mode***
Method	**GET**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 (Online) or 2 (Offline)
Response Codes	200, 403, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1 (Online) or 2 (Offline)

Table 53: POST aml/devices/tower/{id}/reset

Description: Reset the towers controller board whose id is given by the URI path template “id”.

URI	***aml/devices/tower/{id}/reset***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 54: GET aml/devices/tower/{id}/state

Description: Retrieve the towers state whose id is given by the URI path template “id”.

URI	***aml/devices/tower/{id}/state***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1(Varied Off) or 2(Varied On)

Table 55: PUT aml/devices/tower/{id}/state

Description: Retrieve the towers state whose id is given by the URI path template “id”.

URI	***aml/devices/tower/{id}/state***
Method	**GET**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1(Varied Off) or 2(Varied On)

Table 56: GET aml/drives

Description: Retrieve all tape drive resources instances. To filter the number of drives you want to receive use the query parameters described below.

URI	***aml/drives***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are “partition” and “status”. The valid value for the named parameter are as follows: • partition = “partition name” • status = “available” or “used” To retrieve all the drive in the library use “aml/drives”. To retrieve all drives not belonging to a partition use “aml/drives?status=available” To retrieve all drives belonging to a partition named Sales use “aml/drives?partition=Sales”
Request Header	N/A

Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 26: driveList

Table 57: DELETE aml/drives/firmware/image/{name}

Description: Deletes the specified drive firmware file given by the 'name' path parameter.

URI	***aml/drives/firmware/image/{name}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 58: GET aml/drives/firmware/images

Description: Retrieve the list of currently installed firmware images.

URI	***aml/ drives/firmware/images***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are vendor and type, with the following values • vendor=IBM or HP • type=LTO2, LTO3, LTO4, LTO5, LTO6 and LTO7
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 76: firmwareFileList

Table 59: POST aml/drives/firmware/images

Description: Upload a drive firmware image using form data with key = 'file' and value = filename of firmware. So to upload a firmware file named HP_FH_FC_I69Z.E you would have the following:

file = HP_FH_FC_I69Z.E

file = the location of the file (/tmp/ HP_FH_FC_I69Z.E)

Note: The firmware file name must have one of the following extensions, .drv .fmr .fmrz .img .ro .e or .frm, the character case (upper/lower) is ignored.

URI	***aml/* drives/firmware/images**
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: multipart/form-data
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 60: GET aml/drives/firmware/operations/update

Description: Retrieve the progress of the current drive(s) update operations.

This returns a firmwareStatusList object. The updateState element does not apply to this interface and should be ignored.

If a 404 is returned then the firmware update has finished and there is no further status to report.

URI	***aml/drives/firmware/operations/update***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 80: firmwareStatusList

Table 61: POST aml/drives/firmware/operations/update

Description: Upgrade the selected drives with the specified firmware image files. To get the progress status of the update you need to do the above GET, this only applies to i6k libraries.

Note: The Scalar i6k supports an asynchronous request. Non-Scalar i6k tape libraries will support synchronous requests and block until the update has finished.

URI	***aml/drives/firmware/operations/update***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A

Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 77: firmwareUpdateList
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 62: GET aml/drives/logs

Description: Get a list of drive log resources that were generated by the library because of certain tape alerts generated by a drive.

URI	***aml/drives/logs***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 32: driveLogList

Table 63: GET aml/drives/log/{name}

Description: Retrieve the drive log resource instances whose name is given by the URI path template “name”.

URI	***aml/drives/log/{name}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are “save” with the following valid values: • save=the default name you want the browser to save the contents of the file too. If no name is given a default name will be supplied by the Web Server. The purpose of the save parameter is to tell the Web Browser that this is an attachment. If the client is not a Web Browser then the ‘Accept: application/octet-stream’ can be used to retrieve the file data.
Request Header	Accept: application/octet-stream (download the file content) Accept: application/xml or application/json (retrieve the driveLog resource object)
Request Data	N/A

Response Codes	200, 404
Response Header	Content-Type: application/octet-stream, application/xml or application/json Content-Disposition: attachment; filename=”the name of the file” (This will only happen if the save query parameter is requested) On success Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	The file content, or driveLog object, see Figure 33: driveLog

Table 64: DELETE aml/drives/log/{name}

Description: Delete the drive log resource instances whose name is given by the URI path template “name”.

URI	***aml/drives/logs***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 65: POST aml/drives/log/{name}/email

Description: Email the drive log resource instances whose name is given by the URI path template “name”.

URI	***aml/drives/log/{name}/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 66: GET aml/drives/ports

Description: Retrieve all drive port resource instances.

URI	***aml/drives/ports***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are “configuration” with the following valid values: • configuration = “actual” or “requested” Note: On the i6k only requested is applicable
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 34: drivePortList

Table 67: POST aml/drives/powerCycle

Description: Power Cycle one or more drives. The drive serial numbers can be physical or logical.

URI	***aml/drives/powerCycle***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 36: driveSerialNumberList
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 68: GET aml/drives/reports/activity

Description: Retrieve the library drive(s) activity for the last 24 hours. A list of 24 driveActivityStatistics objects will be returned. Each object will represent an hour of activity in the last 24 hours and will report the read and writes in MB for all drives in the library including the total mount counts and the hour of the day (library time UTC). The first entry in the list will represent the activity 23 hours ago, while the last entry in the list will represent activity for the current hour of the day.

URI	***aml/drives/reports/activity***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	The following query parameters are supported, partition and drive with the following values:

	• partition= the name of a specific partition • driveSerialNumber= The physical serial number of the drive If no query parameters are used the request will return data for all drives in the library. If the partition parameter is specified the data for the drives belonging to that partition will be reported. When the drive query parameter is specified then the data for that drive is reported.
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 28: driveActivityStatistics

Table 69: GET aml/drives/reports/activity/details

Description: Retrieve the library drive(s) activity for the last 24 hours. A list of 24 driveActivityStatistics objects will be returned for each drive installed in the library, see Figure 28: driveActivityStatistics for details. The detailedDriveActivityStatistics.drive object will also report the logical and physical serial number of the drive and the partition who own's the drive.

URI	***aml/drives/reports/activity/details***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 24: detailedDriveActivityStatistics

Table 70: GET aml/drives/reports/utilization

Description: Retrieve the list of drive utilization records. The Drive Utilization records provides information as to when tape drives were mounted and un-mounted, capturing how much data the drives read and wrote during such mount sessions. The information provides all details to allow further analysis as to which drives may be under- or over-utilized for tape cartridge load operations while also providing read/write performance data to assess tape cartridge residency needs.

If the Media Type text/plain is selected (Accept http header) the data will be returned in

CSV format.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/drives/reports/utilization***
Method	**GET**
User Role Access	I6k - Admin, Service, User
Parameters	The following query parameters are supported, start, length, period, date, partition, driveSerialNumber, barcode, save with the following values: • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last month, you would specify 30. • Date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. • Partition= the name of a specific partition • Barcode=The media barcode • Save=”name” where name is a file name to use to save the drive utilization information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser. A default “name” is provided if none is given and has the following format: driveUtilization_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv If no query parameters are used the request will return all the drive utilization data. If the partition parameter is specified the data for the drives belonging to that partition will be reported.

Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json, text/plain, application/octet-stream On success Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 37: driveUtilizationList

Table 71: POST aml/drives/reports/utilization/email

Description: Email the list of drive utilization records. The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 68: GET aml/drives/reports/utilization.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/drives/reports/utilization/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 72: GET aml/drive/{serialNumber}

Description: Retrieve the drive resources instances whose serial number is given by the URI path template “serialNumber”. The serialNumber can be the drives logical or physical serial number.

URI	***aml/drive/{serialNumber}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 27: drive

4.3 Drive Data Path

There are two types of data path failover: basic and advanced.

Basic Data Path Failover (BDPF): Basic Data Path Failover is a feature provided as part of the Path Failover/Native Storage Networking license and applies to HP LTO-5 and LTO-6 Fibre Channel tape drives only. It provides an alternate data path when a preferred data path fails.

The HP LTO-5 and higher Fibre Channel tape drives have two Fibre Channel ports. If you enable Data Path Failover on the tape drive, one port is used as the “active port” for data transmission, and the other port stands by for use if the active port fails. If the tape drive loses its Fibre Channel link with the active port, it will automatically “fail over” and use the standby port to continue drive operations.

Advanced Data Path Failover (ADPF): Advanced Data Path Failover is a feature provided as part of the Path Failover/Native Storage Networking license and applies only to HP LTO-6 Fibre Channel tape drives. It provides an alternate data path when a preferred data path fails.

The HP LTO-6 Fibre Channel tape drives have two Fibre Channel ports. If you enable Advanced Data Path Failover on the tape drive, both ports are available but the driver chooses one port to be “active” and used for data transmission, while the other port stands by for use if the “active” port fails. If the tape drive loses its Fibre Channel link with the “active” port, it will fail over and use the standby port to continue drive operations. The library utilizes a driver installed on the host to manage path failover. It uses Port 1 for data path transmission unless a failover occurs. Once failover occurs, the driver uses Port 2 until failover occurs again or the library is rebooted.

Table 73: GET aml/drive/{serialNumber}/dataPath

Description: Get the data path drive or capable data path drive whose serial number is given by the URI path template “serialNumber”. The serialNumber can be the drives logical or physical serial number.

URI	***aml/drive/{serialNumber}/dataPath***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 27: drive

Table 74: PUT aml/drive/{serialNumber}/dataPath

Description: Update the data path drive whose serial number is given by the URI path template “serialNumber”. The drive element field that needs to be up updated is the settings.dataPath field. The valid values are: 1 (None), 2 (Standard) and 3 (Advanced).

The example below turns on standard data path failover. The serialNumber can be the drives logical or physical serial number.

URI	***aml/drive/{serialNumber}/dataPath***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 27: drive The required fields of the drive object are as follows: <ns2:drive xmlns:ns2="http://automatedMediaLibrary/"> <logicalSerialNumber>F0012</logicalSerialNumber> OR <physicalSerialNumber>ABCD</physicalSerialNumber> <settings> <dataPath>2</dataPath> </settings> </ns2:drive>
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 27: drive

Table 75: GET aml/drive/{serialNumber}/operations/clean

Description: Get a list of clean drive tasks that we requested on the drive whose serial number is provided by the URI path template “'serialNumber”. The serialNumber must be the drives logical serial number.

URI	***aml/ drive/{serialNumber}/operations/clean***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 76: POST aml/drive/{serialNumber}/operations/clean

Description: Start a drive cleaning on the drive using the cleanDriveTask object. The

serialNumber must be the drive's logical serial number.

Before you can issue a drive cleaning operation, the drive mode must be changed to offline.

To determine what cleaning media to use to clean the drive you can first use the URI Table 183: GET aml/physicalLibrary/elements to find the configured cleaning slot elements.

You can also use the URI Table 98: GET aml/media to find the cleaning media configured in the library.

Note: This is an asynchronous request. The new task object URI that was created will be included in the 'Location' header of the response (see Table 77: GET aml/drive/{serialNumber}/operations/clean/{id}).

To determine if a drive cleaning task has completed, check the state element of the task object, when complete the state should be 5 (Completed) (see Figure 161: task).

URI	***aml/drive/{serialNumber}/operations/clean***
Method	**POST**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 12: cleanDriveTask
Response Codes	202, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/drive/{serialNumber}/operations/clean/{taskId}
Response Data	See Figure 172: WSResultCode

Table 77: GET aml/drive/{serialNumber}/operations/clean/{id}

Description: Retrieve the task object with the id given by URI path template "id" and the componentId given by URI path template "serialNumber". The serialNumber must be the drives logical serial number.

URI	***aml/drive/{serialNumber}/operations/clean/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 78: POST aml/drive/{serialNumber}/operations/eject

Description: Eject media from the drive whose serial number is provided by the URI

path template “'serialNumber”. The serialNumber can be either the drives physical or logical serial number.

URI	*aml/drive/{serialNumber}/operations/eject*
Method	**POST**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 79: GET aml/drive/{serialNumber}/operations/identify

Description: Retrieve drive identify task on the drive whose serial number is given by the URI path template “serialNumber”. The serialNumber must be the drives logical serial number.

URI	*aml/drive/{serialNumber}/operations/identify*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 80: POST aml/drive/{serialNumber}/operations/identify

Description: Initiate a drive identify on the drive whose serial number is given by the URI path template “serialNumber”. The serialNumber must be the drives logical serial number. Only one identify task can be initiated at a time. If an identify task is currently running you must stop it (DELETE) before starting another one.

URI	*aml/drive/{serialNumber}/operations/identify*
Method	**POST**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 81: DELETE aml/drive/{serialNumber}/operations/identify

Description: Stop a drive identify task for the drive whose serial number is given by the URI path template “serialNumber”. The serialNumber must be the drives logical serial number.

URI	***aml/drive/{serialNumber}/operations/identify***
Method	**DELETE**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 82: POST aml/drive/{serialNumber}/operations/powerCycle

Description: Power cycle the drive whose serial number is given by the URI path template “serialNumber”. The serialNumber can be the drives logical or physical serial number.

URI	***aml/drive/{serialNumber}/operations/powerCycle***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 83: GET aml/drive/{identifier}/operations/state

Description: Retrieve the current state (Varied On, Varied Off or Pending) of the drive whose identifier is provided by the URI path template “'identifier”. The identifier can be the physical or logical serial number of the drive or the drive coordinate is the following format:

“frame,rack,section,column,row” (aml/drive/2,1,4,1,1/operations/state)

The coordinate should be used when you Vary On a drive, because the drive serial numbers are not reported when the drive is in the Pending state.

A single string value will be returned and the possible values are:

1 (Varied On), 2 (Varied Off) and 3 (Pending)

URI	*aml/drive/{serialNumber}/operations/state*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1, 2 or 3

Table 84: PUT aml/drive/{identifier}/operations/state

Description: Change the drive state, 1 (Vary On) or 2 (Vary Off). This option should be used when you want to replace a bad drive.

The identifier can be the physical or logical serial number of the drive or the drive coordinate is the following format:

“frame,rack,section,column,row” (aml/drive/2,1,4,1,1/operations/state)

The coordinate should be used when you Vary On a drive, because the drive serial numbers are not reported when the drive is in the Pending state.

URI	*aml/drive/{serialNumber}/operations/state*
Method	**PUT**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 or 2
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 85: POST aml/drive/{serialNumber}/operations/test

Description: Run a drive self test on the drive whose serial number is provided by the URI path template “'serialNumber”. The serialNumber can be the drive’s logical or physical serial number.

URI	*aml/drive/{serialNumber}/operations/test*
Method	**POST**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 172: WSResultCode

Table 86: GET aml/drive/{serialNumber}/ports

Description: Retrieve the drive ports resources instances whose serial number is given by the URI path template “serialNumber”. The serialNumber can be the drives logical or physical serial number.

This will report both actual and requested topology and speed values for each port.

URI	***aml/drive/{serialNumber}/ports***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are “configuration”. Valid values for configuration are “actual” and “requested”
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 35: drivePorts

Table 87: PUT aml/drive/{serialNumber}/ports

Description: Modify the port settings for the drive whose serial number is given by the URI path template “serialNumber”. The serialNumber can be the drives logical or physical serial number.

The example below changes the requested topology to loop (3) and request speed to 2Gb (2) and loop id to 10.

Note: On the Scalar i6k the ‘requested’ setting can only be modified, the ‘actual’ can NOT be modified. Also only port id 1 can be modified.

URI	***aml/drive/{serialNumber}/ports***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 35: drivePorts The required fields are: <ns2:drivePorts xmlns:ns2="http://automatedMediaLibrary/"> <serialNumber>logical serial number</serialNumber> OR <physicalSerialNumber>physical serial

	number</physicalSerialNumber> <ports> <port> <id>1</id> <topology> <requested>3</requested> </topology> <loopId>10</loopId> <speed> <requested>2</requested> </speed> </port> </ports> </ns2:drivePorts>
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 35: drivePorts

Table 88: GET aml/drive/{serialNumber}/log

Description: Retrieve the drive log whose serial number is given by the URI path template “serialNumber”. The “serialNumber” can be either the physical or logical serial number of the drive. For example to retrieve a drive log whose serial number is DR1002 use the following URI:

aml/drive/DR1002/log

The drive must be connected to an IO Blade or Ethernet Expansion Blade for this feature to work..

URI	***aml/drive/{serialNumber}/log***
Method	**GET**
User Role Access	Admin, Service
Parameters	Optional query parameter is save, with the following valid values save=”name” where name is a file name to use to save the log data to. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. If the “name” is not given a default will be supplied.
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type: application/octet-stream, application/xml, application/json Cookie name= FileDownloadingProgressCookie

Response Data	Byte Stream if request completes successfully, otherwise See Figure 172: WSResultCode

Table 89: POST aml/drive/{serialNumber}/log/email

Description: Email drive log report whose serial number is given by the URI path template “serialNumber”. The “serialNumber” can be either the physical or logical serial number of the drive.

The information will be in an email attachment and the file format will be LTD.

The drive must be connected to an IO Blade or Ethernet Expansion Blade for this feature to work.

URI	***aml/drive/{serialNumber}/log/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml, application/json
Request Data	See Figure 58: email
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml, application/json
Response Data	See Figure 172: WSResultCode

Table 90: GET aml/drive/{serialNumber}/mode

Description: Retrieve the current mode (Online/Offline) of the drive whose serial number is provided by the URI path template “'serialNumber”. The “serialNumber” can be either the physical or logical serial number of the drive.

A single string value will be returned and the possible values are:

1 (online) or 2 (offline)

URI	***aml/drive/{serialNumber}/mode***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: test/plain, application/json
Response Data	1 or 2

Table 91: PUT aml/drive/{serialNumber}/mode

Description: Change the drive mode of the drive whose serial number is provided by the URI path template “'serialNumber”. The “serialNumber” can be either the physical or

logical serial number of the drive.

The mode can be, 1 (Online) or 2 (Offline). This option should be used when you want to block hosts from moving media to a drive (mode offline). When offline a Read Element Status will report an ASC 0x83 ASCQ 0x05 and move are rejected with an ASC 0x83 ASCQ 0x05.

A situation where the drive should be taken offline is when you are updating firmware to the drive.

URI	***aml/drive/{serialNumber}/mode***
Method	**PUT**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	Content-Type: text/plain, application/json
Request Data	1 or 2
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml, application/json
Response Data	See Figure 172: WSResultCode

Table 92: POST aml/drive/{serialNumber}/removeITNexus

Description: Remove the IT Nexus from the drive with the serial number given by URI path template “serialNumber”. The “serialNumber” can be either the physical or logical serial number of the drive.

The drive will clear all SCSI-host defined Initiator-Target (IT) reservations and media removal settings. This only applies to HP LTO5 and greater drives.

Note: This feature only applies to Scalar i6k libraries.

URI	***aml/drive/{serialNumber}/removeITNexus***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 93: GET aml/drive/{serialNumber}/reservation

Description: Request whether the drive whose serial number is provided by the URI path template “'serialNumber” is reserved for library (Web Service) operations only. The “serialNumber” can be either the physical or logical serial number of the drive. A connected host will be prevented from doing any operations (move media) on the drive if the request returns “true”. Valid return values are “true” or “false”.

URI	*aml/drive/{serialNumber}/reservation*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	“true” or “false”

Table 94: PUT aml/drive/{serialNumber}/reservation

Description: Change the reservation on the drive whose serial number is provided by the URI path template “'serialNumber”. The “serialNumber” can be either the physical or logical serial number of the drive. To block host operations, use “true” in the request body.

URI	*aml/drive/{serialNumber}/reservation*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	true or false
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 95: GET aml/enum

Description: Retrieve all the enum resource instances. The enum resources describe the name value pairs for certain XML object elements. For instance the RASGroup object has a property group and a value of 1. To find what 1 means you could request all enums and find in the componentList a component with the name “RASGroup” and the element with name “group”. You would then lookup the entry with a key=1 and see what its value is, in this case it is “Connectivity”, see example XML RESPONSE DATA below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<componentList    xmlns:ns2="http://automatedMediaLibrary/">
   <component name="RASGroup">
      <element name="group">
         <entry key="1" value="Connectivity"/>
         <entry key="2" value="Control"/>
         <entry ke4.4y="3" value="Media"/>
         <entry key="4" value="Drives"/>
         <entry key="5" value="Power"/>
         <entry key="6" value="Robotics"/>
```

```
    </element>
  </component>
  <component name="RASGroupStatus">
    <element name="group">
      <entry key="1" value="Connectivity"/>
      <entry key="2" value="Control"/>
      <entry key="3" value="Media"/>
      <entry key="4" value="Drives"/>
      <entry key="5" value="Power"/>
      <entry key="6" value="Robotics"/>
    </element>
    <element name="status">
      <entry key="1" value="Good"/>
      <entry key="2" value="Failed"/>
      <entry key="3" value="Degraded"/>
      <entry key="4" value="Warning"/>
      <entry key="5" value="Informational"/>
      <entry key="6" value="Unknown"/>
      <entry key="7" value="Invalid"/>
    </element>
  </component>
  </component>
 …..
 …..
</componentList>
```

URI	***aml/enum***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 17: componentList

Table 96: GET aml/enum/{component}

Description: Retrieve the enum resource instances whose is given by the URI path template “component”. This is similar to the aml/enum resource except this URI should only return the elements for the specific path component template parameter. The example output below shows what the response would be if the user used the component named “RASGroup”. So the URI “aml/enum/RASGroup” would return the response data seen below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<elementList   xmlns:ns2="http://automatedMediaLibrary/">
  <component name="RASGroup">
    <element name="group">
      <entry key="1" value="Connectivity"/>
```

```
            <entry key="2" value="Control"/>
            <entry key="3" value="Media"/>
            <entry key="4" value="Drives"/>
            <entry key="5" value="Power"/>
            <entry key="6" value="Robotics"/>
        </element>
    </component>
</elementList>
```

URI	*aml/enum/{component}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 17: componentList (ENUM)

Table 97: GET aml/enum/{component}/{element}

Description: Retrieve the enum component and element resource instance given by the URI path template “component” and “element”. So the URI “aml/enum/partition/interfaceType” would return the response data seen below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<entryList   xmlns:ns2="http://automatedMediaLibrary/">
    <component name="partition">
        <element name="interfaceType">
            <entry key="1" value="SCSI"/>
            <entry key="10" value="Mixed"/>
            <entry key="2" value="Fibre"/>
            <entry key="3" value="SAS"/>
        </element>
    </component>
</entryList>
```

URI	*aml/enum/{component}/{element}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 63: entryList (ENUM)

Table 98: GET aml/media

Description: Retrieve the media resource instances.

URI	***aml/media***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters are partition, location, start, length, frame, rack, type, status, and save, and have the following values: • partition=name of partition • location=”storage”, “drive”, “ie” or “xie” Range parameters • start=0-n • length=1-n or -1 for all media. • frame=0 – maximum number of frames • rack = 1 or 2 • type= 3(LTO2), 4(LTO3), 5(LTO4), 6(LTO5), 7(LTO6), 8(LTO7), 20(Cleaning) • status=”used” or “available” • save=”name” where name is a file name to use to save the media information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: mediaInventory_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv The sharedIE query parameter is an indication of whether or not shared IE media (unassigned media in an IE slot while the library is in auto assignment mode) should specifically be included or excluded from the media list. If true, shared IE media will be included. If false, shared IE media are excluded. For example /media?status=available, by default, includes shared IE media. Adding sharedIE=false (/media?used=available&sharedIE=false) will exclude the shared IE media from the list. So to retrieve all media in one call, “aml/media?start=0&length=-1” which is the default if no range is specified. To retrieve all media belonging to partition named TEST use “aml/media?partition=TEST”.

	To retrieve all media in IE stations use “aml/media?location=ie”. To retrieve the first 50 media belonging to partition TEST use “aml/media?partition=TEST&start=0&length=50”. **Note:** When using the range parameters, the media are sorted by their barcodes.
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 103: mediaList

Table 99: GET aml/media/cleaning

Description: Retrieve a list of cleaning media resources. These are specific media used for drive cleaning.

Note: The cleaningMedia.state element is not supported on the i6k.

URI	***aml/media/cleaning***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 13: cleaningMediaList

4.5 Extended Data Life Management

The EDLM feature requires an Extended Data Lifecycle Management license. One license covers the entire library. EDLM expands upon and replaces the Media Data Integrity Analysis (MeDIA) feature previously used on the library. The MeDIA feature only included manual media scans.

One library managed partition is required for the media scans. This library managed partition is accessible only by a library administrator. It is not presented to any other applications. The library managed partition is assigned its own dedicated resources and EDLM scanning is executed in the background with no impact to normal tape operations. Cartridges are moved into the EDLM library managed partition and scanned using EDLM-scanning drives residing in the EDLM library managed partition. After being

scanned, cartridges are returned to their original locations. Automatic media scanning policies are configured by partition. Each partition can have its own unique set of media scanning and action policies. You may optionally use StorNext Storage Manager to trigger media scans and automatically copy data off of suspect or failed tapes. To use StorNext you must separately install an API client plug-in.

Table 100: GET aml/media/edlm

Description: Get a list of EDLM media resources. These resources describe all the media in the library and whether they have been tested by the EDLM feature. The list can also be used to determine which media to start a new test on.

URI	***aml/media/edlm***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	The following query parameters are supported, partition, barcode, start, length, period, state, type and date with the following values: partition=name (request only media belonging to this partition barcode=the media barcode to filter on start=0-n (range parameter) length=1-n or -1 for all records period=the last number of days to include in the data reported. So if you want to report for the last 3 months, you would specify 90. Date=At what date (the lastTested field of the edlmMedia object is used) you want to start your query. The data returned will include all records that are equal or older than the date specified and applied against the lastTested field of the edlmMedia object. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. state=the current/last test state: 1(pending), 2(In progress), 3(Complete), 4(Stopped), 5(Paused) or 6(Resume) type=the test type of the current/last test type: 1(Quick Scan), 2(Normal Scan) or 3(Full Scan).
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 39: edlmMediaList

Table 101: POST aml/media/edlm

Start a new EDLM test session using the edlmMediaObject list. The required fields of the edlmMedia object are:

edlmMedia.barcode and edlmMedia.testType.

You can also set the edlmMedia.testPriority field.

This call is asynchronous, and the location headed in the response will report the new session id URI that can be used to determine the test state.

URI	***aml/media/edlm***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml or application/json
Request Data	See Figure 39: edlmMediaList
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml or application/json Location: aml/media/edlm/session/#
Response Data	See Figure 172: WSResultCode

Table 102: GET aml/media/edlm/results

Description: Get a list of EDLM media result resources. These resources report the results all media that have been tested by the EDLM feature. The same media may have more than one result, depending on how many times it has been tested.

URI	***aml/media/edlm/results***
Method	GET
User Role Access	Admin, Service, User
Parameters	The following query parameters are supported: partition, barcode, start, length, period, state, type and date, with the following values: • partition=name (request only results for media belonging to this partition • barcode=the media barcode to filter on • start=0-n (range parameter) • length=1-n or -1 for all records • period=the last number of days to include in the data reported. So if you want to report for the last 3 months, you would specify 90. • date=At what date (the lastRunDate of

	the edlmMediaResult object field is used) you want to start your query. The data returned will include all records that are equal or older than the date specified and applied against the lastRunDate field of the edlmMediaResult object. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is "yyyy-MM-dd HH:mm:ss" or "yyyy-MM-dd HH:mm:ss Z", the Z (time zone) will be ignored. • state=the current/last test state: 1(pending), 2(In progress), 3(Complete), 4(Stopped), 5(Paused) or 6(Resume) • type=the test type of the current/last test type: 1(Quick Scan), 2(Normal Scan) or 3(Full Scan).
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 41: edlmMediaResultList

Table 103: GET aml/media/edlm/result/{id}

Description: Retrieve the EDLM media result resource instances whose id is given by the URI path template "id".

URI	***aml/media/edlm/result/{id}***
Method	GET
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 42: edlmMediaResult

Table 104: DELETE aml/media/edlm/result/{id}

Description: Delete the EDLM media result resource instances whose id is given by

the URI path template “id”.

URI	*aml/media/edlm/result/{id}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 42: edlmMediaResult

Table 105: GET aml/media/edlm/sessions

Description: Get a list of EDLM session resources. These session resources report the status of a collection of media that have been tested or are currently been tested. When a user initiates EDLM test, see Table 101: POST aml/media/edlm, a new session is created and it aggregates the results of the media been tested.

URI	*aml/media/edlm/sessions*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 45: edlmSessionList

Table 106: GET aml/media/edlm/session/{id}

Description: Retrieve the EDLM session resource instances whose id is given by the URI path template “id”.

URI	*aml/media/edlm*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 46: edlmSession

Table 107: PUT aml/media/edlm/session/{id}/operations/pause

Description: Pause the EDLM media tests that were started by the session resource instances whose id is given by the URI path template “id”. The tests that will be paused have to have a current test state of pending.

URI	***aml/media/edlm***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 108: PUT aml/media/edlm/session/{id}/operations/resume

Description: Resume the EDLM media tests that were started by the session resource instances whose id is given by the URI path template “id”. The tests that will be resumed have to have a current test state of paused.

URI	***aml/media/edlm***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 109: PUT aml/media/edlm/session/{id}/operations/stop

Description: Stop the EDLM media tests that were started by the session resource instances whose id is given by the URI path template “id”. The tests that will be stopped have to have a current test state of paused, pending or in progress.

URI	***aml/media/edlm***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 172: WSResultCode

Table 110: POST aml/media/operations/moveMedium

Description: Move a media from its source coordinate to the destination element coordinate which must be empty and can be any element type storage, drive, ie or extended ie.

If you want the Standard partition(s) involved in the move operation taken offline before the move and turned back online after the move (if it was previously online) then set the mode element to 2. It is not necessary to take the partition offline and then back online since the host will be notified that there was an inventory change when a media is moved to/from a partition.

When moving media from a drive (unload) use the mediaHomeCoordinate element of the drive object (see Figure 27: drive) to determine the slot coordinate where the media came from (this will be the destinationCoordinate).

URI	***aml/media/operations/moveMedium***
Method	**POST**
User Role Access	Admin, Service, User (Partition Access Required)
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 110: moveMedium
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 111: GET aml/media/reports/crossPartitionMoves

Description: Get the cross partition moves report. This report contains a list of media that have been moved from one partition to another partition.

URI	***aml/media/reports/crossPartitionMoves***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	The following query parameters are supported, start, length, period, date, save with the following values: Range parameters • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last 3 months, you would

	specify 90. • date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. • save=”name” where name is a file name to use to save the cross partition media move information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: crossPartitionMoves_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv If no query parameters are used the request will return all the cross partition move data.
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json On success Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 21: crossPartitionMovesList

Table 112: POST aml/media/reports/crossPartitionMoves/email

Description: Email the list of cross partition media move records.

The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 111: GET aml/media/reports/crossPartitionMoves.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/crossPartitionMoves/email***
Method	**POST**
User Role Access	Admin, Service

Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 113: GET aml/media/reports/inventory

Description: Retrieve the media inventory resource instances.

URI	***aml/media/reports/inventory***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters are partition, location, start, length, frame, rack and have the following values: • partition=name of partition • location=”storage”, “drive” or “ie” Range parameters • start=0-n • length=1-n or -1 for all media. • Frame=0 – maximum number of frames • rack = 1 or 2 • save=”name” where name is a file name to use to save the media inventory information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: mediaInventory_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv **Note:** When using the range parameters, the media are sorted by their barcodes.
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json, text/plain
Response Data	See Figure 103: mediaList

Table 114: POST aml/media/reports/inventory/email

Description: Email the list of media records.

The information will be in an email attachment and the file format will be CSV.

The reportCriter does not support all of the query parameter supported by Table 111: GET aml/media/reports/inventory.

URI	***aml/media/reports/inventory/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 115: GET aml/media/reports/securityEvents

Description: Retrieve the list of media security records.

The Media Security records provide information as to which tape cartridge was removed either expectedly (properly exported) or removed unexpectedly (removed during open door condition or while library was powered off) to assist administrators with tape cartridge inventory verifications.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/securityEvents***
Method	**GET**
User Role Access	i6k - Admin, Service, User
Parameters	The following query parameters are supported, start, length, period, date, barcode, save with the following values: • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last month, you would specify 30. • date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records

	that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. • barcode=The media barcode • save=”name” where name is a file name to use to save the security event information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: mediaSecurityEvents_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv If no query parameters are used the request will return all the media security event data.
Request Header	N/A
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 105: mediaSecurityEventList

Table 116: POST aml/media/reports/securityEvents/email

Description: Email the list of media security records.

The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 115: GET aml/media/reports/securityEvents.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/securityEvents/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 172: WSResultCode

Table 117: GET aml/media/reports/tapeAlerts

Description: Retrieve the list of tape alert records.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/tapeAlerts***
Method	**GET**
User Role Access	i6k - Admin, Service, User
Parameters	The following query parameters are supported, start, length, period, date, driveSerialNumber, barcode, save with the following values: • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last week, you would specify 7. • date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. • driveSerialNumber= The physical serial number of the drive • barcode=The media barcode • save=”name” where name is a file name to use to save the tape alert information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: tapeAlerts_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv If no query parameters are used the request will return all the tape alert data.

Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 158: tapeAlertList

Table 118: POST aml/media/reports/tapeAlerts/email

Description: Email the list of drive tape alert records.

The tape alert records provide information as which drive reported which Tape Alert while being loaded with which tape cartridge. The report information provides details to allow further analysis as to which media or which drive may be a cause for media or drive failures.

The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 117: GET aml/media/reports/tapeAlerts.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/tapeAlerts/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 119: GET aml/media/reports/usage

Description: Retrieve the list of media usage records.

The Media Usage Report provides media information capturing tape cartridge identification and media capacity and usage information as well as unrecovered and recovered read/write error count statistics.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/usage***
Method	**GET**
User Role Access	I6k - Admin, Service, User

Parameters	The following query parameters are supported, start, length, period, date, barcode, save with the following values: • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last month, you would specify 30. • date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. • barcode=The media barcode • save=”name” where name is a file name to use to save the media usage information to. The file format will be CSV. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: mediaUsage_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv If no query parameters are used the request will return all the media usage data.
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 108: mediaUsageList

Table 120: POST aml/media/reports/usage/email

Description: Email the list of media usage records.

The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 119: GET aml/media/reports/usage.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/media/reports/usage/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 121: GET aml/medium/{barcode}

Description: Retrieve the media whose barcode is given by the URI path template "barcode". To retrieve the media object whose barcode is "100000L6" use:

"aml/medium/100000L6".

A list is returned in the case where we have duplicate barcodes, multiple media may have the same barcode.

URI	***aml/medium/{barcode}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 103: mediaList

4.6 Partition configuration and operations

The following URI's provide the ability to change a partitions configuration and perform certain operations on the partition. To create a new partition (basic mode) you must use the following interface: Table 122: GET aml/partitions. This interface does not allow the user to select specific storage or IE slots or drives; the user can basically only select the number of slots or drives they want to belong to the new partition. The interface also does not provide for drive cleaning or auto drive leveling. So if a user wants to create a partition (advanced mode) using specific slots and drives, configure drive cleaning and

other policies, such as drive leveling, control path, etc.., you will need to make multiple Web Service interface requests.

For example, to create a Standard partition with name “Sales Partition”, use the interface: Table 122: GET aml/partitions; interface and only specify those elements in the XML request. After this request has completed successfully, use the interface: Table 179: GET aml/partition/{name}/segments; interface to add specific slots, IE and drives to the partition, see the interface for more details. To set up a drive-cleaning policy for the partition, use the following interface: Table 168: GET aml/partition/{name}/policy/driveCleaning; this interface also provides the capability to delete a drive cleaning policy from a partition. The partition drive-leveling policy is provided using the interface: Table 172: GET aml/partition/{name}/policy/driveLeveling; this interface can also be used to remove selective firmware files from partition drive leveling (see interface for more information).

To modify the partition name, barcode reporting, vendor ID and product ID in one request you could use the following interface: Table 134: GET aml/partition/{name}. We also provide individual interfaces for each of these properties. There are also interfaces for changing the AMP extensions, control path and various policies on a partition (see the interfaces below for more details).

Table 122: GET aml/partitions

Description: Retrieve all partition resource instances configure on the library.

URI	***aml/partitions***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	The following query parameters are supported, fields, with the following values • fields=A comma separate list of partition object elements you want to get values for. The partition elements id, name, type, mode and serialNumber will always be included and if the user specifies ‘fields=’, these are the elements that will have real values. The elements that are supported are driveDomainType , storageSlotCount, emptyStorageSlotCount, driveCount, emptyDriveCount, ieSlotCount, emptyIeSlotCount, xieSlotCount, emptyXieSlotCount, ampExtensionsCount, mediaCount, barcodeReporting, vendorId, productId, controlPathProvider, policySettings.
Request Header	N/A
Request Data	N/A
Response Codes	200

Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 118: partitionList

Table 123: POST aml/partitions

Description: Create a new partition resource. When using the partition XML object to create a partition, the following valid values apply:

type: 1 (standard), 2 (EDLM), 3 (AMP), 4 (Active Vault), 5(CVTL)

driveDomainType: 0 (Unknown), 3 (LTO2), 4 (LTO3), 5 (LTO4), 6 (LTO5), 7 (LTO6), 8 (LTO7)

barcodeReporting: 1 (Prefix), 2 (Suffix), 3 (Disable), 4 (Pass Through) and 2 more for non-Scalari6k products 5 (Standard 6), 6 (Plus 6)

vendorId: 0 (ADIC), 1 (Quantum), – default 1

productId: 0 (N/A), 1 (Scalar 24), 2 (Scalar 100), 3 (Scalar 1000), 4 (Scalar 10k), 5 (Scalar i500), 6 (Scalar i2000), 7 (Scalar i6000)

The productId's 5, 6 and 7 can only be used if the Quantum vendorId is selected. For vendorId 1 all productId except 0 can be used.

The productId 5, 6 and 7 can only be used if the Quantum vendorId is selected. For vendorId 1 all productId except 0 can be used.

For HP-branded libraries productId and vendorId do not apply and will always be set to vendorId of 2 and productId of 0.

The example below creates a partition named "Test Partition" with 2 LTO-5 drives and 300 slots. The vendor id will be "Quantum", the product id will be "Scalar i6000" and the barcodeReporting policy will be "Pass Through".

The required fields for creating a partition are "name" and "type".

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:partition  xmlns:ns2="http://automatedMediaLibrary/">
   <name>Test </name>
   <type>1</type>
   <driveDomainType>6</driveDomainType>
   <storageSlotCount>300</storageSlotCount>
   <driveCount>2</driveCount>
   <ieSlotCount>12</ieSlotCount>
   <xieSlotCount>0</xieSlotCount>
  <ampExtensionsCount>0</ampExtensionsCount>
   <barcodeReporting>4</barcodeReporting>
   <vendorId>1</vendorId>
   <productId>7</productId>
</ns2:partition>
```

A partition name can only contain the following characters

Quantum: A-Z a-z 0-9 _ and spaces

HP: A-Z a-z 0-9 . _ @ - and spaces

The maximum number of character allowed is 64.

URI	*aml/partitions*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 119: partition
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 119: partition

Table 124: GET aml/partitions/policy/activeVault

Description: Get a list of activeVaultPolicy resources configured on the library. Active Vault policies are configured on standard partitions. The policies intercept host commands that export tape cartridges to library managed Active Vault partitions. The Active Vault feature requires an Active Vault license.

URI	*aml/partitions/policy/activeVault*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 5: activeVaultPolicyList

Table 125: POST aml/partitions/policy/activeVault

Description: Create a new active vault policy for partition named LL2. You can either choose an externalDefinedExport policy or a vaultDefinedExport policy but not both.

URI	*aml/partitions/policy/activeVault*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 6: activeVaultPolicy The following example would move any media barcode that contains the characters ‘00LTO5’ to

	the active vault ‘AV Partition’ when the media is exported (moved to the IE station) from the partition ‘LL2’. <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:activeVaultPolicy xmlns:ns2="http://automatedMediaLibrary/"> <partitionName>LL2</partitionName> <vaultDefinedExport> <activeVaultName>AV Partition</activeVaultName> <mediaFilter>*00LTO5</mediaFilter> </vaultDefinedExport> </ns2:activeVaultPolicy> The other option is to choose the external application server(s) configured on the library, see Table 296: GET aml/system/policy/externalApplicationServers. You would used the externalApplicationServers.name field.
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 126: GET aml/partitions/policy/autoImport

Description: Retrieve the list of Auto Import Policy resources. The Auto Import feature allows the import of media from an AMP partition to a Standard partition based on an Auto Import Policy configured for each standard partition. The policy defines a range of media barcodes to use to determine which media get imported into which partition.

URI	***aml/partitions/policy/autoImport***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 9: autoImportPolicyList

Table 127: GET aml/partitions/policy/autoExport

Description: Retrieve the list of Auto Export Policy resources. The Auto Export feature reroutes the export of media (media moves by a host to IE stations) from a Standard

partition to an AMP partition. This feature can only be configured if there is an AMP partition configured.

URI	*aml/partitions/policy/autoExport*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 7: autoExportPolicyList

Table 128: GET aml/partitions/policy/driveCleaning

Description: Get a list of drive cleaning policy resources configured on the library.

URI	*aml/partitions/policy/driveCleaning*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 29: driveCleaningPolicyList

Table 129: GET aml/partitions/policy/edlm

Description: Get a list of edlmPolicy resources configured on the library.

URI	*aml/partitions/policy/edlm*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 43: edlmPolicyList

Table 130: POST aml/partitions/policy/edlm

Description: Create a new edlm policy. The example below creates a policy for partition

LL1.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:edlmPolicy xmlns:ns2="http://automatedMediaLibrary/">
   <partitionName>LL1</partitionName>
   <tapeAlert>
      <scanType>1</scanType>
      <count>3</count>
   </tapeAlert>
   <timeInterval>
      <quickScan>0</quickScan>
      <normalScan>365</normalScan>
      <fullScan>1095</fullScan>
   </timeInterval>
   <onImport>1</onImport>
   <scanPriotity>2</scanPriotity>
   <concurrentScans>0</concurrentScans>
   <continueOnError>true</continueOnError>
   <disableRasTicketGeneration>true</disableRasTicketGeneration>
   <externalPolicies>
      <externalApplicationServersName>The name of the external applications servers configuration
</externalApplicationServersName>
      <mediaCopyPolicy>3</mediaCopyPolicy>
      <suspectCountScanType>1</suspectCountScanType>
   </externalPolicies>
</ns2:edlmPolicy>
```

URI	***aml/partitions/policy/edlm***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 44: edlmPolicy
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/partition/{name}/policy/edlm
Response Data	See Figure 172: WSResultCode

Table 131: GET aml/partitions/policy/ekm

Description: Get a list of partition encryption policy resources. Encryption key management systems generate, protect, store, and manage encryption keys. These keys are used by their respective tape drives to encrypt information being written to tape, and decrypt information being read from tape media. Encryption Key Management (EKM) is a licensable feature. Encryption on the Scalar i6000 tape library is enabled by partition only.

If a configured partition does not support Encryption Key Management it will not be included in this list.

URI	***aml/partitions/policy/ekm***
Method	**GET**

User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Header	Content-Type:application/xml or application/json
Response Codes	200
Response Data	See Figure 120: partitionEncryptionPolicyList

Table 132: GET aml/partitions/reports/utilization

Description: Retrieve the partition utilization file resource. The output below is a sample of what the report looks like.

The Partition Utilization Report provides information for library partition usage determination, capturing high watermark counts for drive, slot and media usage. The report information provides all necessary details to evaluate library and partition usage for proper sizing and/or usage determination.

Note: This is a licensed feature (Advanced Reporting).

URI	***aml/partitions/reports/utilization***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Optional query parameter is save, with the following valid values • save=”name” where name is a file name to use to save the partition utilization information to. The file format will be in text. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: partitionUtilization_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: application/octet-stream On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	Sample Data Partition Utilization Report Library: dvt4 SN: 273190049 Date: Tue Jun 03 10:21:27 MDT 2014

	---------- June : 2014 ---------- HIGH WATER MARKS: DWM:2, SWM:12, MWM:2, PN:LL5 DWM:0, SWM:24, MWM:3, PN:AMP Part DWM:1, SWM:24, MWM:3, PN:MEDIA Partition DWM:3, SWM:12, MWM:6, PN:LL1 DWM:1, SWM:12, MWM:1, PN:edlm DWM:0, SWM:108, MWM:17, PN:AV DWM:1, SWM:18, MWM:0, PN:LL3 DWM:1, SWM:120, MWM:41, PN:LL2 DWM:1, SWM:120, MWM:2, PN:EDLM DWM:0, SWM:12, MWM:1, PN:AMP DWM:0, SWM:12, MWM:1, PN:amp partition DWM:0, SWM:0, MWM:0, PN:JUnit Empty Partition DWM:0, SWM:120, MWM:2, PN:AMPPY DWM:2, SWM:660, MWM:126, PN:JUnit Standard LL One DWM:0, SWM:330, MWM:55, PN:JUnit Active Vault Partition DWM:2, SWM:660, MWM:126, PN:JUnit Standard Partition DWM:1, SWM:24, MWM:2, PN:Sales Partition One DWM:0, SWM:330, MWM:55, PN:JUnit AMP Partition DWM:0, SWM:24, MWM:3, PN:AV Part DWM:0, SWM:12, MWM:1, PN:amp DWM:1, SWM:102, MWM:2, PN:LL4 IBM PARTITION ACTIVITY: Mon Jun 02 10:52:15 MDT 2014,LL1,0,0,0,Delete Thu May 15 16:06:44 MDT 2014,Sales Partition One,1,24,2,Rename Sales Partition Sales Partition One Fri May 09 16:24:29 MDT 2014,Sales Partition One,1,24,2,Create first Tue May 27 14:59:38 MDT 2014,EDLM,0,0,0,Delete Tue May 06 14:10:14 MDT 2014,EDLM,1,120,2,Create first

Table 133: POST aml/partitions/reports/utilization/email

Description: Email the Partition Utilization activity report. The information will be in an email attachment and the file format will be text.

URI	*aml/partitions/reports/utilization/email*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 134: GET aml/partition/{name}

Description: Retrieve the partition whose name is given by the URI path template "name". To retrieve the partition name "Test Partition", the following URI would be requested:

"aml/partition/Test Partition"

URI	*aml/partition/{name}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 119: partition

Table 135: PUT aml/partition/{name}

Description: Modify the partition whose name is given by the URI path template "name".

The following URI "aml/partition/LL2" and XML request data will modify the partitions, name, ampExtensionsCount, barcodeReporting, vendorId and productId. These values are the only ones that can be modified through this URI. The serialNumber field must be specified as this is used to lookup the partition you want to modify.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:partition xmlns:ns2="http://automatedMediaLibrary/">
   <name>Partition2</name>
   <serialNumber>273190048_LL3</serialNumber>
   <barcodeReporting>1</barcodeReporting>
   <vendorId>1</vendorId>
   <productId>7</productId>
</ns2:partition>
```

To change other partition attributes see the following:

Table 137: GET aml/partition/{name}/ampExtensionsCount

Table 139: GET aml/partition/{name}/barcodeReporting

Table 148: GET aml/partition/{name}/mode

Table 150: GET aml/partition/{name}/name

Table 179: GET aml/partition/{name}/segments

Table 205: GET aml/physicalLibrary/segments

URI	***aml/partition/{name}***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 119: partition
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 119: partition

Table 136: DELETE aml/partition/{name}

Description: Delete the partition whose name is given by the URI path template “name”.

URI	***aml/partition/{name}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 137: GET aml/partition/{name}/ampExtensionsCount

Description: Retrieve the current amp extensions count for the partition whose name is provided by the URI path template “name”.

A single string value will be returned that represents the current configured extension count.

Note: The count represents the number of segments/magazines.

URI	*aml/partition/{name}/ampExtensionsCount*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	The number of AMP extensions

Table 138: PUT aml/partition/{name}/ampExtensionsCount

Description: Change the partitions amp extensions count (segments), the number that is requested will be added or subtracted from the current extension count. If you want to add 5 segments to the current count, request 5. If you want to remove 4 request -4.

This option only applies to standard partitions.

Note: The partition mode must change before a connected host will see these changes. To change the partitions mode, see: Table 148: GET aml/partition/{name}/mode.

URI	*aml/partition/{name}/ampExtensionsCount*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	The number of extensions to add or subtract from the current count.
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 139: GET aml/partition/{name}/barcodeReporting

Description: Retrieve the current barcodeReporting methodology (Prefix, Suffix, Disabled or Pass Through) of the partition whose name is given by the URI path template “name”.

A single string value will be returned and the possible values are:

1 (Prefix), 2 (Suffix), 3 (Disabled) or 4 (Pass Through).

URI	*aml/partition/{name}/barcodeReporting*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404

Response Header	Content-Type:text/plain or application/json
Response Data	1,2,3 or 4

Table 140: PUT aml/partition/{name}/barcodeReporting

Description: Change the partitions barcode reporting methodology, 1 (Prefix), 2 (Suffix), 3 (Disabled) or 4 (Pass Through).

This option only applies to standard partitions.

Note: The partition mode must change before a connected host will see these changes. To change the partitions mode, see: Table 148: GET aml/partition/{name}/mode.

URI	***aml/partition/{name}/barcodeReporting***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1,2,3 or 4
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

4.7 Control Path

You must define a control path for each standard partition. The control path is used to connect a partition to a host application. The i6000 does not automatically assign a control path when you create a partition. Each partition control path can occur through one of several different physical connection points depending on the hardware configuration of your library.

You can configure IBM and HP LTO-5 or LTO-6 FC drives as the control path for a partition. The drive must not be connected to an FC I/O blade, but it must be connected to an Ethernet Expansion blade.

You do not need an SNW license to configure a drive for control path. However, you do need an SNW license to configure control path failover.

There are five types of control path configurations:

Control Path: This configuration options does not require an SNW license and can use any EEB attached IBM or HP LTO drive.

Multi Control Path: This configuration option requires an SNW license and allows you to assign multiple IBM and/or HP LTO-5 or higher drives as possible control paths.

Basic Control Path Failover (BCPF): This configuration provides support for only HP LTO-5 and LTO-6 path failover licensed drives for basic control path failover. When BCPF is used, one drive is assigned as the primary control path and another drive as

the control path failover (secondary) drive. The control path failover drive is used whenever the primary control path drive fails, becomes inoperable, or loses connectivity.

Advanced Control Path Failover (ACPF): This configuration provides support for onlyHP LTO-6, path failover licensed drives for advanced control path failover. When ACPF is used, one drive is configured as the primary control path and one or more drives are selected as failover drives. This configuration requires an Advanced Path Failover (APF) device driver installed on an attached host.

Advanced Control Path: This configuration provides support for only IBM LTO-5 and LTO-6 drives and requires an SNW license. When Advanced Control Path is used, multiple IBM drives can be designated as control path drives with a device driver installed on an attached host. This driver will determine which drive is used as the primary control path and will choose a new control path drive in the event the primary control path fails.

Table 141: GET aml/partition/{name}/controlPath

Description: Retrieve the list of control path capable drives for the partition whose name is given as the URI path template “name”.

URI	***aml/partition/{name}/controlPath***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	See Figure 26: driveList

Table 142: PUT aml/partition/{name}/controlPath

Description: Update the control path settings on the requested drives.

See the XML REQUEST DATA examples below for more details:

Example 1: Configure a Standard CP drive, no CPF. The controlPath type must be 2 and primary must be set to true.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>2</type>
         </controlPath>
      </settings>
   </drive>
```

```
</ns2:driveList>
```

Example 2: Remove a CP drive, the controlPath type must be set to 1. This is also used if you want to disable Control Path in the case where there are CPF drives configured. You only need to specify the CP drive to disable.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>1</type>
         </controlPath>
      </settings>
   </drive>
</ns2:driveList>
```

Example 3: Configure Basic CP/CPF pair. The controlPath type must be set to 3 and one of the drives must have primary set to true.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F001396043</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>3</type>
         </controlPath>
      </settings>
   </drive>
      <drive>
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>false</primary>
            <type>3</type>
         </controlPath>
      </settings>
   </drive>
</ns2:driveList>
```

Example 4: Set the CP/CPF pair to advanced mode. The controlPath type must be set to 4.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F001396043</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
```

```
            <type>4</type>
         </controlPath>
      </settings>
   </drive>
      <drive>
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>false</primary>
            <type>4</type>
         </controlPath>
      </settings>
   </drive>
</ns2:driveList>
```

Example 5: Remove a CPF drive from a CP/CPF pair. The CPF drive controlPath type must be set to 1 and the CP drive controlPath type must be set to 2.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F001396043</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>2</type>
         </controlPath>
      </settings>
   </drive>
      <drive>
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>false</primary>
            <type>1</type>
         </controlPath>
      </settings>
   </drive>
</ns2:driveList>
```

Example 6: Add a basic CPF drive to an existing CP drive. Both drives must have their controlPath type set to 3.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F001396043</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>3</type>
         </controlPath>
      </settings>
   </drive>
      <drive>
```

```
      <logicalSerialNumber>F00139603D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>false</primary>
            <type>3</type>
         </controlPath>
      </settings>
   </drive>
</ns2:driveList>
```

Example 7: Add an advanced CPF drive to an existing ACPF cluster.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
      <logicalSerialNumber>F001396025</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>false</primary>
            <type>4</type>
         </controlPath>
      </settings>
   </drive>
</ns2:driveList>
```

Example 8: Configure three multi control path drives. This would be the same for Advanced IBM control path except the type field would be set to 6 instead of 5.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList xmlns:ns2="http://automatedMediaLibrary/">
   <drive>
     <!-- <physicalSerialNumber>GB120401FD</physicalSerialNumber> -->
      <logicalSerialNumber>F00139709D</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>5</type>
         </controlPath>
      </settings>
    </drive>
    <drive>
     <!-- <physicalSerialNumber>10WT017350</physicalSerialNumber> -->
      <logicalSerialNumber>F001397091</logicalSerialNumber>
      <settings>
         <controlPath>
            <primary>true</primary>
            <type>5</type>
         </controlPath>
      </settings>
    </drive>
   <drive>
     <!-- <physicalSerialNumber>10WT017242</physicalSerialNumber> -->
      <logicalSerialNumber>F001397097</logicalSerialNumber>
      <settings>
         <controlPath>
```

```
            <primary>true</primary>
            <type>5</type>
          </controlPath>
        </settings>
    </drive>
</ns2:driveList>
```

To remove one or all of the Multi control paths supply the drive list you want to remove and set the type field to 1.

When configuring Multi and IBM Advance control path the primary field must be set to true.

URI	***aml/partition/{name}/controlPath***
Method	**PUT**
User Role Access	Admin, Service
Parameters	The query parameters are takePartitionOffline. The valid values for the query parameters are as follows • takePartitionOffline = true or false (this parameter allows you to take the partition offline before the operation is performed. The partition will be taken online when the operation has completed. If the partition was already offline before the operation was performed, it will be left in that state.)
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 26: driveList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 26: driveList

Table 143: GET aml/partition/{name}/dataPath

Description: Get a list of drive resources that are data path failover capable who belong to the partition whose name is provided by the URI path template “'name”. A drive is data path failover capable if it is connected to an EEB, is LTO5 or greater and has an SNW license applied to it.

URI	***aml/partition/{name}/dataPath***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 26: driveList

Table 144: PUT aml/partition/{name}/dataPath

Description: Update the drive resources data path failover settings who belong to the partition whose name is provided by the URI path template “'name”.

URI	*aml/partition/{name}/dataPath*
Method	**GET**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml or application/json
Request Data	See Figure 26: driveList
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 26: driveList

Table 145: GET aml/partition/{name}/driveSerialNumbers

Description: Get a list of drive serial numbers for the drives that are configured in the partition whose name is provided by the URI path template “'name”.

URI	*aml/partition/{name}/driveSerialNumbers*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 36: driveSerialNumberList

Table 146: POST aml/partition/{name}/driveSerialNumbers

Description: Add the drives with the serial numbers given in the driveSerialNumberList to the partition whose name is provided by the URI path template “'name”.

The example below adds the drives with logical serial number, F001396031 and F001396007 to the partition whose name is provided by the URI path template “'name”.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveSerialNumberList xmlns:ns2="http://automatedMediaLibrary/">
   <serialNumber>F001396031</serialNumber>
   <serialNumber>F001396007</serialNumber>
</ns2:driveSerialNumberList>
```

URI	*aml/partition/{name}/driveSerialNumbers*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 36: driveSerialNumberList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 36: driveSerialNumberList

Table 147: DELETE aml/partition/{name}/driveSerialNumbers

Description: Remove the drives with the serial numbers given in the driveSerialNumberList from the partition whose name is provided by the URI path template “'name”.

The example below removes the drive with logical serial number, F001396007 from the partition whose name is provided by the URI path template “'name”.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveSerialNumberList xmlns:ns2="http://automatedMediaLibrary/">
   <serialNumber>F001396007</serialNumber>
</ns2:driveSerialNumberList>
```

URI	*aml/partition/{name}/driveSerialNumbers*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 36: driveSerialNumberList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 36: driveSerialNumberList

Table 148: GET aml/partition/{name}/mode

Description: Retrieve the current mode (Online/Offline) of the partition whose name is provided by the URI path template “'name”.

A single string value will be returned and the possible values are:

1 (online) or 2 (offline)

URI	*aml/partition/{name}/mode*
Method	**GET**
User Role Access	Admin, Service, User

Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	1 or 2

Table 149: PUT aml/partition/{name}/mode

Description: Change the partition mode, 1 (Online) or 2 (Offline). This option only applies to standard partitions.

URI	***aml/partition/{name}/mode***
Method	**PUT**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 or 2
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 150: GET aml/partition/{name}/name

Description: Retrieve the current name of the partition whose name is provided by the URI path template 'name'. The return name should be the same as the 'name' template.

The only purpose of the method is for completeness/consistency of the Web Service interface.

URI	***aml/partition/{name}/name***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:text/plain or application/json
Response Data	The name of the partition

Table 151: PUT aml/partition/{name}/name

Description: Change the partition's name . A partition name can only contain the following characters:

Quantum: A-Z a-z 0-9 _ and spaces

HP: A-Z a-z 0-9 . _ @ - and spaces

The maximum number of character allowed is 64.

Note: The partition mode must change before a connected host will see these changes. To change the partitions mode, see: Table 148: GET aml/partition/{name}/mode.

URI	***aml/partition/{name}/mode***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	The new name of the partition
Response Codes	200, 403, 404
Response Header	Content-Type:text/plain or application/json
Response Data	See Figure 172: WSResultCode

Table 152: GET aml/partition/{name}/operations

Description: Get the list of tasks resources that were started/requested by the partition whose name is given by the URI path template “name”.

URI	***aml/partition/{name}/operations***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 153: GET aml/partition/{name}/operations/autoImport

Description: Get a list of auto import media tasks that were requested on the AMP partition whose name is provided by the URI path template “'name”.

URI	***aml/partition/{name}/operations/autoImport***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 154: POST aml/partition/{name}/operations/autoImport

Description: Start an Auto Import on the partition whose name is provided by the URI path template “'name”.

The partition must be an AMP partition type.

Note: This is an asynchronous request. The new task object URI that was created will be included in the ‘Location’ header of the response, see Figure 161: task.

To determine if an auto import task has completed, check the state element of the task object, when complete the state should be 5 (Completed), see Figure 161: task.

URI	***aml/partition/{name}/operations/autoImport***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 155: GET aml/partition/{name}/operations/autoImport/{id}

Description: Retrieve the task object with the id given by URI path template “id” and the componentId (partition name) given by URI path template “name”.

URI	***aml/partition/{name}/operations/autoImport/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 156: DELETE aml/partition/{name}/operations/autoImport/{id}

Description: Delete the task object with the id given by URI path template “id” and the componentId given by URI path template “name” .

Note: This does not stop the specific auto import request. It is just a way to clean up operation tasks from the libraries database.

URI	*aml/partition/{name}/operations/autoImport/{id}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 157: GET aml/partition/{name}/operations/inventory

Description: Get the list of inventory tasks resources that were started/requested by the partition whose name is given by the URI path template “name”.

URI	*aml/partition/{name}/operations/inventory*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 158: POST aml/partition/{name}/operations/inventory

Description: Request library to do an inventory for the partition whose name is given by the URI path template “name”.

The startElement must be a valid element address, if the elementCount is greater that the number of elements, the inventory will ignore the extra elements. To inventory the whole partition, set the startElement to 0 and the elementCount to 65000.

If the partition is not taken offline the inventory will fail.

The example below requests to do an inventory of the first 10 slots in the partition.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:inventoryTask xmlns:ns2="http://automatedMediaLibrary/">
  <offline>true</offline>
  <startElement>4096</startElement>
  <elementCount>10</elementCount>
</ns2:inventoryTask>
```

Note: This is an asynchronous request. The new task resource URI that was created will be included in the ‘Location’ header of the response. See Table 158: POST aml/partition/{name}/operations/inventory.

URI	*aml/partition/{name}/operations/inventory*
Method	**POST**
User Role Access	i6k - Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 90: inventoryTask
Response Codes	202, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/partition/{name}/operations/inventory
Response Data	See Figure 172: WSResultCode

Table 159: GET aml/partition/{name}/operations/inventory/{taskId}

Description: Get the inventory tasks resources that were started/requested by the partition whose name is given by the URI path template “name” and whose task id is given by the URI path template “taskId”.

To determine if an inventory task has completed, check the state element of the task object, when complete the state should be 5 (Completed). See Figure 161: task, for more details.

URI	*aml/partition/{name}/operations/inventory/{taskId}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 160: DELETE aml/partition/{name}/operations/inventory/{taskId}

Description: Delete the inventory task resources that were started/requested by the partition whose name is given by the URI path template “name” and whose task id is given by the URI path template “taskId”.

The task must be completed (See Figure 161: task) before it can be deleted.

URI	*aml/partition/{name}/operations/inventory/{taskId}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404

Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 161: GET aml/partition/{name}/policy/activeVault

Description: Get the active vault policy resources for the partition whose name is given by the URI path template “name”.

URI	***aml/partition/{name}/policy/activeVault***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 6: activeVaultPolicy

Table 162: PUT aml/partition/{name}/policy/activeVault

Description: Modify the active vault policy resources for the partition whose name is given by the URI path template “name”.

To create a new active vault policy, see Table 125: POST aml/partitions/policy/activeVault.

The example below modifies the active vault policy for partition LL2. Instead of using StorNext to decide what to do on an export, move the media directly to the AV Partition vault.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:activeVaultPolicy xmlns:ns2="http://automatedMediaLibrary/">
   <partitionName>LL2</partitionName>
   <vaultDefinedExport>
     <activeVaultName>AV Partition</activeVaultName>
   </vaultDefinedExport>
</ns2:activeVaultPolicy>
```

URI	***aml/partition/{name}/policy/activeVault***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 6: activeVaultPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json

Response Data	See Figure 6: activeVaultPolicy

Table 163: DELETE aml/partition/{name}/policy/activeVault

Description: Delete the active vault policy resources for the partition whose name is given by the URI path template “name”.

URI	***aml/partition/{name}/policy/activeVault***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 164: GET aml/partition/{name}/policy/autoImport

Description: Retrieve Auto Import Policy resource for the partition whose name is given by the URI path template “name”.

The Auto Import feature allows the import of media from an AMP partition to a Standard partition based on an Auto Import Policy configured for each standard partition. The policy defines a range of media barcodes to use to determine which media gets imported into which partition.

URI	***aml/partition/{name}/policy/autoImport***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 10: autoImportPolicy

Table 165: PUT aml/partition/{name}/policy/autoImport

Description: Update the Auto Import Policy resource for the partition whose name is given by the URI path template “name”.

The required autoImportPolicy fields required are partitionName, which should be a valid partition name and match the URI path name.

If the mediaBarcodeFilter is not given or is an empty string then Auto Import will be

turned off for this partition.

The mediaBarcodeFilter is constrained to the following regex "([a-zA-Z0-9]{5,15}-[a-zA-Z0-9]{5,15};?\\s*)*". Examples of valid ranges are as follows:

000100-000200
00500-00550; 00000700-00000900
100400900-100500000

Examples of invalid ranges:

000200-000100
000100-000400; 000200-000300

URI	*aml/partition/{name}/policy/autoImport*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 10: autoImportPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 10: autoImportPolicy

Table 166: GET aml/partition/{name}/policy/autoExport

Description: Retrieve Auto Export policy resource for the partition whose name is given by the URI path template “name”.

The Auto Export feature reroutes media that have been moved from a standard partition storage slot to an IE station by a host application to an AMP partition.

URI	*aml/partition/{name}/policy/autoExport*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 8: autoExportPolicy

Table 167: PUT aml/partition/{name}/policy/autoExport

Description: Update the Auto Export Policy resource for the partition whose name is given by the URI path template “name”.

To configure a Auto Export policy for partition LL1 and have the media rerouted to the

AMP partition Amp1 you would use the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:autoExportPolicy
   xmlns:ns2="http://automatedMediaLibrary/">
   <partitionName>LL2</partitionName>
   <destinationAmpPartitionName>amp</destinationAmpPartitionName>
</ns2:autoExportPolicy>
```

To un-configure Auto Export for partition LL2 you would do the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:autoExportPolicy
   xmlns:ns2="http://automatedMediaLibrary/">
   <partitionName>LL2</partitionName>
</ns2:autoExportPolicy>
```

An Auto Export policy can only be configured for a Standard partition. You cannot configure an Auto Export policy if no AMP partition exist. If a partition has an Active Vault policy configure then you cannot configure an Auto Export policy on it.

URI	***aml/partition/{name}/policy/autoExport***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 8: autoExportPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 8: autoExportPolicy

4.8 Drive Cleaning Policy

The following interface allows you to configure drive cleaning policies on a partition basis. If drive-cleaning functionality is enabled on the host application, do not enable drive cleaning for any partitions in the library. The basic policy is to enable cleaning when the drive requests it, because of a hard read or write error. You can also set a policy to clean drives in a partition after a certain amount of motion hours; this policy only applies to HP LTO5 or higher drives. For EDLM partitions you can also select a mount count to trigger a drive cleaning.

Note: As of release i12.1 it is no longer necessary to use the POST or DELETE method of the interface to configure or remove a drive cleaning policy for a partition. An enabled element has been added to the driveCleaningPolicy object that determines if a policy is configured or not for a particular partition. It is recommended that the PUT method be used instead of the POST and DELETE. There are three supported methods on the i6k, by just selecting the enable drive cleaning, driveCleaningPolicy.enable (drive requested to be cleaned), by configuring driveCleaningPolicy.driveCleaning.motionTime (This applies only to HP LTO5 drives or greater) and

driveCleaningPolicy.driveCleaning.mountCount (only supported for EDLM drives).

Table 168: GET aml/partition/{name}/policy/driveCleaning

Description: Get the drive cleaning policy resource for the partition whose name is given by the URI path template “name”.

URI	*aml/partition/{name}/policy/driveCleaning*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 30: driveCleaningPolicy

Table 169: POST aml/partition/{name}/policy/driveCleaning

Description: DEPRECATED. Create a new drive cleaning policy for the partition whose name is given by the URI path template “name”. The mountCount element should only be used for EDLM partitions.

Note: As of i12.1 it is recommended you use the PUT method instead of this POST method.

URI	*aml/partition/{name}/policy/driveCleaning*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 30: driveCleaningPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 30: driveCleaningPolicy

Table 170: PUT aml/partition/{name}/policy/driveCleaning

Description: Update the drive cleaning policy resource for the partition whose name is given by the URI path template “name”.

Note: As of i12.1 the enabled element has been added to the driveCleaningPolicy object. This must be used to determine if a cleaning policy be applied to the partition.

URI	*aml/partition/{name}/policy/driveCleaning*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 30: driveCleaningPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 30: driveCleaningPolicy

Table 171: DELETE aml/partition/{name}/policy/driveCleaning

Description: DEPRECATED. Delete the drive cleaning policy resources for the partition whose name is given by the URI path template “name”.

Note: As of i12.1 it is recommended you use the PUT method instead of this DELETE method.

URI	*aml/partition/{name}/policy/driveCleaning*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 172: GET aml/partition/{name}/policy/driveLeveling

Description: Get the drive leveling policy resource for the partition whose name is given by the URI path template “name”. The drive leveling policy uses specified drive firmware images(s) to be used on the different drive generations configured in the partition. To see what firmware images are currently available/installed on the library, see the following interface: Table 58: GET aml/drives/firmware/images.

The type element provides three options, 0 (No drive leveling), 1 (Auto leveling) and 2 (Selective leveling).

Note: The current interface for i6k only supports options 0 and 2.

URI	*aml/partition/{name}/policy/driveLeveling*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A

Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 31: driveLevelingPolicy

Table 173: PUT aml/partition/{name}/policy/driveLeveling

Description: Update the drive leveling policy resource for the partition whose name is given by the URI path template “name”. See examples below for details:

Example 1: Add another firmware file to be used for leveling drives in a partition. The firmware file(s) selected must be compatible with the drive(s) configured in the library. In this example we are adding firmware file “LTO6_D7Y0.fcp_fh.fmrz”. For every PUT request you must specify all the firmware files you want to apply to the partition. So in this example we need to include “LTO6FH_FC_J3KZ.E” even though it is already configured.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveLevelingPolicy
   xmlns:ns2="http://automatedMediaLibrary/">
   <partition>LL1</partition>
   <type>2</type>
   <firmwareFile>
      <name>LTO6FH_FC_J3KZ.E</name>
      <version>J3KZ</version>
      <vendor>HP</vendor>
      <type>LTO6</type>
   </firmwareFile>
   <firmwareFile>
      <name>LTO6_D7Y0.fcp_fh.fmrz</name>
      <version>D7Y0</version>
      <vendor>IBM</vendor>
      <type>LTO6</type>
   </firmwareFile>
</ns2:driveLevelingPolicy>
```

Example 2: Remove firmware file “LTO6FH_FC_J3KZ.E” from the current configuration. Basically this firmware file object is not included in the request. Again, when you want to configure selective drive leveling you must always include all the leveling firmware files in the request.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveLevelingPolicy
   xmlns:ns2="http://automatedMediaLibrary/">
   <partition>LL1</partition>
   <type>2</type>
   <firmwareFile>
      <name>LTO6_D7Y0.fcp_fh.fmrz</name>
      <version>D7Y0</version>
      <vendor>IBM</vendor>
      <type>LTO6</type>
```

```
    </firmwareFile>
</ns2:driveLevelingPolicy>
```

Note: driveLevelingPolicy.type 1 (Auto Leveling) is not supported on i6k, only type 0 (None) or 2 (Selective Leveleing) is supported.

URI	*aml/partition/{name}/policy/driveLeveling*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 31: driveLevelingPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 31: driveLevelingPolicy

Table 174: GET aml/partition/{name}/policy/edlm

Description: Get the edlm policy resources for the partition whose name is given by the URI path template “name”.

URI	*aml/partition/{name}/policy/edlm*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 44: edlmPolicy

Table 175: PUT aml/partition/{name}/policy/edlm

Description: Modify the edlm policy resources for the partition whose name is given by the URI path template “name”.

The example below modifies the policy so RAS ticket generation and notification is disabled on bad or suspect media and scan policy is scheduled every 30 days for normal scans and 60 days for full scans.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:edlmPolicy xmlns:ns2="http://automatedMediaLibrary/">
   <partitionName>LL1</partitionName>
   <disableRasTicketGeneration>true</disableRasTicketGeneration>
   <timeInterval>
      <quickScan>0</quickScan>
      <normalScan>30</normalScan>
```

```
        <fullScan>60</fullScan>
    </timeInterval>
</ns2:edlmPolicy>
```

URI	*aml/partition/{name}/policy/edlm*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 44: edlmPolicy
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 44: edlmPolicy

Table 176: DELETE aml/partition/{name}/policy/edlm

Description: Delete the edlm policy resources for the partition whose name is given by the URI path template “name”.

URI	*aml/partition/{name}/policy/edlm*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 177: GET aml/partition/{name}/policy/ekm

Description: Retrieve the encryption key management (EKM) policy for the partition with the name given by URI path template “name”.

URI	*aml/partition/{name}/policy/ekm*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 121: partitionEncryptionPolicy

Table 178: PUT aml/partition/{name}/policy/ekm

Description: Update the encryption key management (EKM) policy for the partition with the name given by URI path template “name”.

URI	***aml/partition/{name}/policy/ekm***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 121: partitionEncryptionPolicy The Request data will be a little different depending on the ekmServerType requested, see the following examples for details: *<ns2:partitionEncryptionPolicy xmlns:ns2="http://automatedMediaLibrary/"> <partitionName>LL2</partitionName> <ekmServerType>4</ekmServerType> <libraryManaged>true</libraryManaged> <fipsEnabled>false</fipsEnabled> <keyReuse>false</keyReuse> <keyType>2</keyType> </ns2:partitionEncryptionPolicy>* The element fipsEnabled does not apply to partitions with IBM drives or for ekmServerType 8 (QEKM) and 32 (TKLM). The ekmServerType 8 and 32 (QEKM/TKLM) only applies to partitions containing only IBM drives. **Note**: The partition must be taken offline before you can change the EKM policy. All drives in the partition must be unloaded before this request can be processed. The ekmServerType selected must be configured before you can set a policy to libraryManaged. If the server is currently configured by another partition to Key Per Library (keyType) then you must select the same keyType. If Key Per Partition is configure by another partition for this server type then only Key Per Partition or Key Per Media can be selected. If Key Per Media is configure by another partition for this server type then only Key Per Media or Key Per Partition can be selected.
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 121: partitionEncryptionPolicy

4.9 Partition resource assignment

The following interface provides the capability to add and delete drives, storage, Import/Export (IE) and extended IE (XIE) slots to and from a partition. Each of these resources are presented as a segment where storage, IE and XIE segments contain 6 slots and a drive segment contains 1 drive. To determine the location and type of a segment, use the “coordinate” element of segment object. If a segment is removed from a partition (DELETE method) it is reassigned back to the physical library so it is now available to any partition configured on the library. To find all segments in the physical library use the URI:

Table 205: GET aml/physicalLibrary/segments. This URI provides query parameters to filter out exactly what type of segments, the segment state, etc.. the user is looking for.

Table 179: GET aml/partition/{name}/segments

Description: Retrieve the segments resources for the partition given by URI path template “name”.

URI	***aml/partition/{name}/segments***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters names are type, state, start, length, frame and rack with the following valid values: • type=”storage”, “ie”, “drive” or “xie” • state=”full” or “empty” • start=0-n • length=1-n or -1 for all media. • frame=0 – maximum number of frames • rack = 1 or 2So to retrieve all segments in one call, “aml/partition/{name}/segments?start=0&length=-1” which is the default if no range is specified. To retrieve all drive segments “aml/partition/{name}/segments?type=drive”
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 149: segmentList

Table 180: POST aml/partition/{name}/segments

Description: Add the list of segments to the partition whose name is given by URI path template “name”. The added segments must not be currently owned by this partition or another partition, they must be available. To discover which segments are available, use the following URI: Table 205: GET aml/physicalLibrary/segments.

Example 1: To add a drive to a partition you would send the request shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segmentList  xmlns:ns2="http://automatedMediaLibrary/">
   <segment>
      <coordinate>
         <frame>1</frame>
         <rack>1</rack>
         <section>2</section>
         <column>1</column>
         <row>1</row>
         <type>4</type>
      </coordinate>
   </segment>
</ns2:segmentList>
```

Example 2: Add 2 storage segments and on IE segment.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segmentList   xmlns:ns2="http://automatedMediaLibrary/">
   <segment>
      <coordinate>
         <frame>0</frame>
         <rack>1</rack>
         <section>7</section>
         <column>3</column>
         <row>1</row>
         <type>2</type>
      </coordinate>
   </segment>
   <segment>
      <coordinate>
         <frame>0</frame>
         <rack>1</rack>
         <section>8</section>
         <column>3</column>
         <row>1</row>
         <type>2</type>
      </coordinate>
   </segment>
   <segment>
      <coordinate>
         <frame>1</frame>
         <rack>2</rack>
         <section>2</section>
         <column>3</column>
         <row>1</row>
         <type>3</type>
```

```
      </coordinate>
   </segment>
</ns2:segmentList>
```

Example 3: The above 2 requests combined into one single request.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segmentList   xmlns:ns2="http://automatedMediaLibrary/">
   <segment>
      <coordinate>
         <frame>1</frame>
         <rack>1</rack>
         <section>2</section>
         <column>1</column>
         <row>1</row>
         <type>4</type>
      </coordinate>
   </segment>
   <segment>
      <coordinate>
         <frame>0</frame>
         <rack>1</rack>
         <section>7</section>
         <column>3</column>
         <row>1</row>
         <type>2</type>
      </coordinate>
   </segment>
   <segment>
      <coordinate>
         <frame>0</frame>
         <rack>1</rack>
         <section>8</section>
         <column>3</column>
         <row>1</row>
         <type>2</type>
      </coordinate>
   </segment>
   <segment>
      <coordinate>
         <frame>1</frame>
         <rack>2</rack>
         <section>2</section>
         <column>3</column>
         <row>1</row>
         <type>3</type>
      </coordinate>
   </segment>
</ns2:segmentList>
```

The following URI can be user to find what segments are currently available (not part of an existing partition): Table 205: GET aml/physicalLibrary/segments**.**

URI	*aml/partition/{name}/segments*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 149: segmentList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 149: segmentList

Table 181: DELETE aml/partition/{name}/segments

Description: Delete the list of segments from the partition whose name is given by the URI path template “name”.

URI	*aml/partition/{name}/segments*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 149: segmentList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 149: segmentList

Table 182: PUT aml/partition/{name}/segments

Description: Reconfigure the partition whose name is given by the URI path template “name” with the new segmentList request data. The segmentList can contain storage, drive, ie and extended ie segments. The segmentList can contain segments that currently belong to the partition or available (non allocated) segments. You cannot assign segments that are currently owned by another partition.

The example below reconfigures a partition with one drive and 3 storage segments (18 slots).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segmentList xmlns:ns2="http://automatedMediaLibrary/">
  <segment>
    <coordinate>
      <frame>1</frame>
      <rack>1</rack>
      <section>9</section>
      <column>1</column>
      <row>1</row>
```

```
            <type>4</type>
        </coordinate>
        <size>1</size>
        <configuredType>0</configuredType>
    </segment>
    <segment>
        <coordinate>
            <frame>0</frame>
            <rack>1</rack>
            <section>1</section>
            <column>4</column>
            <row>1</row>
            <type>2</type>
        </coordinate>
        <size>6</size>
        <configuredType>0</configuredType>
    </segment>
    <segment>
        <coordinate>
            <frame>0</frame>
            <rack>1</rack>
            <section>2</section>
            <column>4</column>
            <row>1</row>
            <type>2</type>
        </coordinate>
        <size>6</size>
        <configuredType>0</configuredType>
    </segment>
    <segment>
        <coordinate>
            <frame>0</frame>
            <rack>1</rack>
            <section>3</section>
            <column>4</column>
            <row>1</row>
            <type>2</type>
        </coordinate>
        <size>6</size>
        <configuredType>0</configuredType>
    </segment>
</ns2:segmentList>
```

URI	***aml/partition/{name}/segments***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 149: segmentList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 149: segmentList

Table 183: GET aml/physicalLibrary

Description: Retrieve the physical library resource instances.

Note: The autoClean element will be deprecated for i12 release. The new drive cleaning methodology is cover by the following interface: Table 168: GET aml/partition/{name}/policy/driveCleaning.

The autoConfiguration and autoCalibration feature are no longer supported for i12.

URI	***aml/physicalLibrary***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 123: physicalLibrary

Table 184: PUT aml/physicalLibrary

Description: Update the physical library settings features. All of the setting that can be modified are under physicalLibrary.phySettings object, and include the following features:

1. driveSerialNumberSpoofing – enable or disable the ability for the library to present drives with logical serial numbers, rather than physical serial numbers, to a host.
2. autoInventory – enable/disable a library inventory ever time the library is rebooted or when the RCU goes from not ready to ready state.
3. autoCalibration – This feature is deprecated in i12 (680Q/680H).
4. autoConfiguration –This feature is deprecated in i12 (680Q/680H).
5. autoCleaning – This feature is deprecated in i12 (680Q/680H). Drive cleaning is now configured on a partition basis, see Table 168: GET aml/partition/{name}/policy/driveCleaning.
6. autoDriveUnload – enable/disable the ability of the drive to automatically eject the media from a drive after it has unloaded.
7. Ipv6 – enable/disable the IPv6 feature .

   Note: As of i12.3 release, this interface has been deprecated, you cannot disable IPv6. Calling this interface will do nothing.
8. extendedIe – enable/disable the Extended IE feature, which allows the configuration of storage slots to be used as IE slots.

9. sendUsageStatistics – set the interval the library will send the library usage and performance data to Quantum.
10. healthCheck – Set the interval for the library to automatically performs health checks on the robot(s), tower and the robotic power rails. If problems are found, a RAS ticket is generated. The intervals are in days (0-180). If 0 is selected the feature is turned off.
11. aisleLights – Set the time period the aisle lights will be turned on for. Valid values are 30 or 60 minutes or 0 for always turned off.
12. webCamera – Set the IP of the host that is running the application that is managing the camera mounted in the library.
13. icmpService – enable/disable the capability for device on the network to ping the library.
14. sshService – enable/disable the capability to establish a secure shell (ssh) connection to the library.
15. cliService – enable/disable the remote CLI interface.
16. xmlInterfaceService – enable/disable the Vision remote interface.
17. serviceLogin – enable/disable the ability of a Service user to login to the library from the Graphical User Interface (Local and Remote interfaces).
18. sessionTimeout – Set the session timeout a user will be automatically logged out (session will be terminated). The valid values are, 1 through 1440 minutes. The timeout is based on the length of inactivity of a user login session.
19. snmp – Configure the library SNMP settings.
20. Smis – Configure the library SMIS settings.

One or many settings can be applied on a single request. All you need to do is include the feature you want to update. So, for example if I want to update the ipv6 and sessionTimeout features you canuse the following XML:

```
 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:physicalLibrary xmlns:ns2="http://automatedMediaLibrary/">
  <phySettings>
    <ipv6>
      <enabled>true</enabled>
    </ipv6>
    <sessionTimeout>
      <minutes>15</minutes>
    </sessionTimeout>
  </phySettings>
</ns2:physicalLibrary>
```

It is rare, but sometimes a feature settings update may fail for some reason. If that feature was included with other feature update in a single request, the request will return a HTTP failure status, but that does not necessarily mean that all the features we not updated successfully. It is recommended that you do a GET request after a failure

has occurred to get the current setting of the individual features.

URI	*aml/physicalLibrary*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 123: physicalLibrary
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 123: physicalLibrary

Table 185: GET aml/physicalLibrary/elements

Description: Retrieve the elementList resource instances.

URI	*aml/physicalLibrary/elements*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters names are partition, state, type, start, length, frame, rack with the following valid values: • partition=name of partition • state=”full” or “empty”. The slot contains a media (full) or the slot is empty (empty) • type=”storage”, “ie”, “drive” “xie” or “cleaning” • start=0-n • length=1-n or -1 for all elements. • frame=0 – maximum number of frames (specify which frame you want) • rack = 1 or 2 So to retrieve all elements in one call, use “aml/physicalLibrary/elements?start=0&length=-1” which is the default if no range is specified. To retrieve all elements belonging to partition named TEST use “aml/physicalLibrary/elements?partition=TEST”. To retrieve all ie elements that contain media use “aml/physicalLibrary/elements?type=ie&state=full”.
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 56: elementList

Table 186: GET aml/physicalLibrary/mode

Description: Get the current library mode, 1 (online) or 2(offline).

URI	***aml/physicalLibrary/mode***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:text/plain or application/json
Response Data	1 or 2

Table 187: PUT aml/physicalLibrary/mode

Description: Change the library mode to 1 (online) or 2 (offline).

URI	***aml/physicalLibrary/mode***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 or 2
Response Codes	200, 403
Response Header	Content-Type:text/plain or application/json
Response Data	1 or 2

Table 188: GET aml/physicalLibrary/operations

Description: Get the list of tasks resources that were started/requested.

URI	***aml/*physicalLibrary/operations**
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters names are type with the following valid values: • type = 0 (All), 1 (inventory), 2 (library shutdown), 3 (library reboot), 4 (identify drive), 5 (drive clean), 6 (power cycle FC IO blade), 7 (reset FC IO blade), 8 (identify FC IO blade), 9 (Identify Ethernet Expansion Blade), 10 (Auto Import Media), 11 (Generate Command History Logs)
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 160: taskList

Table 189: GET aml/physicalLibrary/operations/inventory

Description: Get a list of inventory tasks that we requested on the physical library.

URI	***aml/physicalLibrary/operations/inventory***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 190: POST aml/physicalLibrary/operations/inventory

Description: Start an inventory of the physical library using the inventoryTask object. This object will specify the starting element address and ending element address (inventory range) and whether to take the physical library offline. If the physical library is not taken offline the inventory will fail.

Note: This is an asynchronous request. The new task object URI that was created will be included in the 'Location' header of the response, see Table 191: GET aml/physicalLibrary/operations/inventory/{id}.

To determine if an inventory task has completed, check the state element of the task object, when complete the state should be 5 (Completed), see Figure 161: task.

URI	***aml/physicalLibrary/operations/inventory***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 90: inventoryTask
Response Codes	202, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 191: GET aml/physicalLibrary/operations/inventory/{id}

Description: Retrieve the task object with the id given by URI path template "id".

To retrieve a task object with id of 86, you would use the following URI:

‘aml/physicalLibrary/operations/inventory/86’.

URI	*aml/physicalLibrary/operations/inventory/{id}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 192: DELETE aml/physicalLibrary/operations/inventory/{id}

Description: Delete the task object with the id given by URI path template “id”.

Note: This does not stop the specific inventory request. It is just a way to clean up operation tasks from the libraries database.

URI	***aml*/physicalLibrary/operations/inventory/{id}**
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 193: GET aml/physicalLibrary/operations/shutdown

Description: Get a list of shutdown tasks that we requested on the physical library.

URI	*aml/physicalLibrary/operations/shutdown*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 194: POST aml/physicalLibrary/operations/shutdown

Description: Shut down the library using the shutdownTask object. This object must specify a reboot element with a value of false, otherwise the library will reboot. To

complete the shutdown process you must physically push the power button on the library.

URI	*aml/physicalLibrary/operations/shutdown*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 155: shutdownTask
Response Codes	202, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 195: GET aml/physicalLibrary/operations/shutdown/{id}

Description: Retrieve the task object of type 2 (Shutdown) with the id given by URI path template “id”.

URI	*aml/physicalLibrary/operations/shutdown/{id}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 196: DELETE aml/physicalLibrary/operations/shutdown/{id}

Description: Delete the task object of type 2 (Shutdown) with the id given by URI path template “id”.

Note: This does not stop the shutdown.

URI	*aml/physicalLibrary/operations/shutdown/{id}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 197: GET aml/physicalLibrary/operations/reboot

Description: Get a list of reboot tasks that were previously requested.

URI	*aml/physicalLibrary/operations/reboot*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 160: taskList

Table 198: POST aml/physicalLibrary/operations/reboot

Description: Get a list of reboot tasks that were previously requested.

URI	*aml/physicalLibrary/operations/reboot*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 155: shutdownTask
Response Codes	202, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 199: GET aml/physicalLibrary/operations/reboot/{id}

Description: Retrieve the task object of type 3 (Reboot) with the id given by URI path template “id”.

URI	*aml/physicalLibrary/operations/reboot/{id}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 161: task

Table 200: DELETE aml/physicalLibrary/operations/reboot/{id}

Description: Delete the task object of type 3 (Reboot) with the id given by URI path template “id”.

Note: This does not stop the reboot.

URI	*aml/physicalLibrary/operations/reboot/{id}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 201: GET aml/physicalLibrary/subset/configuration

Description: This is a subset of the physicalLibrary resource, which groups all configuration options of the library.

URI	*aml/physicalLibrary/subset/configuration*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 124: physicalLibraryConfigurationFigure 172: WSResultCode

Table 202: GET aml/physicalLibrary/subset/remoteAccess

Description: This is a subset of the physicalLibrary resource, which groups all remote network service access options of the library.

URI	*aml/physicalLibrary/subset/remoteAccess*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json

Response Data	See Figure 125: physicalLibraryRemoteAccessFigure 172: WSResultCode

Table 203: GET aml/physicalLibrary/subset/resources

Description: This is a subset of the physicalLibrary resource, which groups all library resources.

URI	***aml/physicalLibrary/subset/resources***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 126: physicalLibraryResourcesFigure 172: WSResultCode

Table 204: GET aml/physicalLibrary/subset/settings

Description: This is a subset of the physicalLibrary resource, which groups similar physical library settings.

URI	***aml/physicalLibrary/subset/settings***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 127: physicalLibrarySettingsFigure 172: WSResultCode

Table 205: GET aml/physicalLibrary/segments

Description: Get a list of segment resources. A segment on an Scalar i6k, can be one of the following types:

Storage, this is equivalent to a storage magazine, which consists of 6 slots.

Drive, a drive segment consists of one drive.

IE, is equivalent to an IE magazine which consistes on 6 slots. IE magazines are always

located in IE Stations.

Extended IE, an extended IE (XIE) segment is a storage magazine the is reported as an XIE. From a connected host, these elements will show up in the IE element address space.

Cleaning, a cleaning segment is a storage magazine reserved for cleaning media only.

URI	***aml/physicalLibrary/segments***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters names are partition, status, type, start, length, frame, rack with the following valid values: • partition=name of partition • status=”available” or “used” (owned by a partition or not) • type=”storage”, “ie”, “drive”, “xie” or “cleaning” • start=0-n • length=1-n or -1 for all media. • frame=0 – maximum number of frames • rack = 1 or 2 So to retrieve all segments in one call,use “aml/physicalLibrary/segments?start=0&length=-1” which is the default if no range is specified. To retrieve all segments belonging to partition named LL1 use “aml/physicalLibrary/segments?partition=LL1”. To retrieve all ie segments use “aml/physicalLibrary/segments?type=ie”.
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 149: segmentList

Table 206: GET aml/physicalLibrary/segments/amp

Description: Retrieve the storage segment resources belonging to Automated Media Pool (AMP) partitions.

URI	***aml/physicalLibrary/segments/amp***
Method	GET
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 149: segmentList

Table 207: PUT aml/physicalLibrary/segments/amp

Description: Change the owner of the segments in the segmentList. This interface is used to reassign *storage* segments from a Standard partition to an Automated Media Pool (AMP) partition and vise versa.

When reassigning storage segments from an AMP to a Standard partition the Standard partition should have enough available AMP extensions configured to satisfy the request (reassign segments <= AMP extensions).

The segment list in the request must belong to a single partition and must be reassigned to a single partition.

In the example below we have a segment that belongs to an AMP partition called AMP and we want to reassign it to a partition called LL1. The segmentList will contain a single segment that currently belongs to the AMP partition but we have changed its owner element to LL1 the partition we want to reassign the segment too.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segmentList xmlns:ns2="http://automatedMediaLibrary/">
   <segment>
      <coordinate>
         <frame>0</frame>
         <rack>1</rack>
         <section>4</section>
         <column>3</column>
         <row>1</row>
         <type>2</type>
      </coordinate>
      <size>6</size>
      <owner>LL1</owner>
      <configuredType>0</configuredType>
   </segment>
</ns2:segmentList>
```

To get a list of AMP storage segments use the GET request above. To get a list of Standard storage segments, use Table 179: GET aml/partition/{name}/segments with the appropriate query parameters.

URI	***aml/physicalLibrary/segments/amp***
Method	PUT
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 149: segmentList
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or

	application/json
Response Data	See Figure 172: WSResultCode

Table 208: GET aml/physicalLibrary/segments/cleaning

Description: Get a list of cleaning segment resources. These segments will contain cleaning media used for library initiated cleaning.

URI	***aml/physicalLibrary/segments/cleaning***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 149: segmentList

Table 209: POST aml/physicalLibrary/segments/cleaning

Description: Create a new cleaning segment resource. If no cleaning media are already present in the segment they can be imported into the segment, see Table 110: POST aml/media/operations/moveMedium.

URI	***aml/physicalLibrary/segments/cleaning***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 150: segment
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 210: DELETE aml/physicalLibrary/segments/cleaning

Description: Remove a cleaning segment resource. This will make the segment available, so it can be used by a partition.

URI	***aml/ physicalLibrary/segments/cleaning***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json

Request Data	See Figure 150: segment. This will be the current cleaning segment you want to delete.
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 172: WSResultCode

Table 211: GET aml/physicalLibrary/status

Description: Retrieve the library status information resource. This resource reports general state/status information about the library.

Note: This URI does not affect the users session timeout; all other URI are considered a user activity and reset the session timeout.

URI	***aml/physicalLibrary/status***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 95: libraryStatus

Table 212: GET aml/service/logs

Description: Get a list of service logs.

URI	***aml/service/logs***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 153: serviceLogList

Table 213: GET aml/service/log/{name}

Description: Get the service log whose name is given by the URI path template “name”. The available log names are provide by the interface

Table 212: GET aml/service/logs

URI	*aml/service/log/{name}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are “save” with the following valid values: • save=the default name you want the browser to save the contents of the file too. If no name is given a default name will be supplied by the Web Server. The purpose of the save parameter is to tell the Web Browser that this is an attachment.
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type: application/octet-stream, application/xml or application/json Content-Disposition: attachment; filename=”the name of the file” (This will only happen if the save query parameter is requested) On success Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	The log file content.

Table 214: POST aml/service/log/{name}/email

Description: Email the service log whose name is given by the URI path template “name”.

URI	*aml/service/log/{name}/email*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 215: POST aml/service/resetFactoryDefault

Description: Reset the libraries configuration to factory defaults.

URI	*aml/service/resetFactoryDefault*
Method	**POST**
User Role Access	Admin, Service

Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	202, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 216: GET aml/system/configuration/record

Description: Get a library configuration record.

URI	***aml/system/configuration/record***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Optional query parameter is save, with the following valid values • save=”name” where name is a file name to use to save the configuration record/report information to. The file format will be text. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file.
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type: text/plain or application/octet-stream On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	Byte Stream or text

Table 217: POST aml/system/configuration/record/email

Description: Email a library configuration record to recipients found in the provided email object.

URI	***aml/system/configuration/record/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json

Response Data	See Figure 173: WSResultCode

Table 218: GET aml/system/dateTime

Description: Retrieve the date and time resource. This reports the time on the library with timezone included.

URI	*aml/system/dateTime*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 23: dateTime

Table 219: PUT aml/system/dateTime

Description: Update the current date time resource on the library. The example XML object below shows the request body data used to update the time. This data could have been JSON but all the examples in this document are XML. To discover all the available timezone see Table 222: GET aml/system/dateTime/timeZoneIDs.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:dateTime  xmlns:ns2="http://automatedMediaLibrary/">
   <date>2013-05-22</date>
   <time>13:56:15</time>
   <timezone>American/Denver</timezone>
</ns2:dateTime>
```

URI	*aml/system/dateTime*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 23: dateTime
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 23: dateTime

Table 220: GET aml/system/dateTime/ntp

Description: Retrieve the Network Time Protocol (NTP) resource. This is used to synchronize the libraries clock with a number of NTP servers on the internet.

URI	*aml/system/dateTime/ntp*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 118: NTP

Table 221: PUT aml/system/dateTime/ntp

Description: Update the NTP resource. To disable NTP just send an empty NTP object, no server defined.

URI	*aml/system/dateTime/ntp*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 118: NTP
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 118: NTP

Table 222: GET aml/system/dateTime/timeZoneIDs

Description: Retrieve the available list of timezone ID's. When you are setting the timezone use the string value reported between the values reported in parentheses, for example, from the response data below: “<ID>(GMT+13:00) MIT (WSDT)</ID> “ use “MIT”.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:timeZoneIDs   xmlns:ns2="http://automatedMediaLibrary/">
   <ID>(GMT-12:00) Etc/GMT+12 (GMT-12:00)</ID>
   <ID>(GMT-11:00) Etc/GMT+11 (GMT-11:00)</ID>
   <ID>(GMT+13:00) MIT (WSDT)</ID>
   <ID>(GMT+13:00) Pacific/Apia (WSDT)</ID>
   <ID>(GMT-11:00) Pacific/Midway (SDT)</ID>
   <ID>(GMT-11:00) Pacific/Niue (NUST)</ID>
   <ID>(GMT-11:00) Pacific/Pago_Pago (SDT)</ID>
   <ID>(GMT-11:00) Pacific/Samoa (SDT)</ID>
   <ID>(GMT-11:00) US/Samoa (SDT)</ID>
   <ID>(GMT-10:00) America/Adak (HADT)</ID>
   ……..
```

</ns2:timeZoneIDs>

URI	*aml/system/dateTime/timeZoneIDs*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 165: timeZoneIDs

Table 223: GET aml/system/ekm/communicationCertificates

Description: Retrieve the list of communicationCertificate resource information installed on the library.

URI	*aml/system/ekm/communicationCertificates*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 15: communicationCertificateList

Table 224: POST aml/system/ekm/communicationCertificates

Description: Install the communication certificates to be used between the library and the EKM servers. The following form data key value pairs are required:

For all server types, the key “type” is required:
type=qekm, tklm, skm, rkm or kmip

Note: rkm is no longer supported

QEKM or TKLM
root=The root certificate file

SKM
quantum=The Quantum certificate bundle file
Or the following
root= The root certificate file
admin=The admin certificate file

adminpassword=The admin password
client=The client certificate file
clientpassword=The client password

KMIP
root= The root certificate file
client=The client certificate file
clientpassword=The client password

URI	*/aml/system/ekm/communicationCertificates*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: multipart/form-data
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 225: GET aml/system/ekm/logs

Description: Retrieve the SKM server logs.

URI	*/aml/system/ekm/logs*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are type and save with the following valid values: • save=the default name you want the browser to save the contents of the file to. If no name is given a default name will be supplied by the Web Server. The purpose of the save parameter is to tell the Web Browser that this is an attachment. If the client is not a Web Browser then the 'Accept: application/octet-stream' can be used to retrieve the file data. • type=0(Primary), 1(secondary) and 2(Import Warning) If no type is given, then 0 is the default.
Request Header	Accept: application/octet-stream (download the file content)
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: application/octet-stream, application/xml or application/json Content-Disposition: attachment;

	filename=”the name of the file” (This will only happen if the save query parameter is requested) Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	The file content, it is in tgz format.

Table 226: GET aml/system/ekm/reports/audit/mediaStatus

Description: Retrieve the list of EKM media statistics resources.

URI	***aml/system/ekm/reports/audit/mediaStatus***
Method	**GET**
User Role Access	i6k - Admin, Service, User
Parameters	Optional query parameter is start, length, period, date, partition and save, with the following valid values: • save=”name” where name is a file name to use to save the media status information to. The file format will be CSV. • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last month, you would specify 30. • date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored. • partition=The partition name. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: mediaStatus_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv
Request Header	N/A

Request Data	N/A
Response Codes	200
Response Header	Content-Type: application/xml or application/json On success Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 47: ekmMediaStatusList

Table 227: POST aml/system/ekm/reports/audit/mediaStatus/email

Description: Email the list of media encryption status records.

The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 226: GET aml/system/ekm/reports/audit/mediaStatus.

URI	***aml/system/ekm/reports/audit/mediaStatus/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 228: GET aml/system/ekm/reports/audit/partitionActivity

Description: Retrieve the list of EKM partition activity resources.

The EKM (Encryption Key Management) Report provides partition encryption method summaries, providing historic information as to when a library partition was enabled or disabled for library managed encryption.

URI	***aml/system/ekm/reports/audit/partitionActivity***
Method	**GET**
User Role Access	i6k - Admin, Service, User
Parameters	Optional query parameter is save, with the following valid values: • save=”name” where name is a file name to use to save the partition activity information to. The file format will be CSV. • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last month, you would specify 30. • partition=The partition name

	The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: partitionActivity_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type: application/xml or application/json On success Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 49: ekmPartitionActivityList

Table 229: POST aml/system/ekm/reports/audit/partitionActivity/email

Description: Email the list of partition encryption activity records.

The information will be in an email attachment and the file format will be CSV.

The reportCriteria object supports the same query parameters as Table 228: GET aml/system/ekm/reports/audit/partitionActivity.

URI	***aml/system/ekm/reports/audit/partitionActivity/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 230: GET aml/system/ekm/servers

Description: Get a list of EKM server resources. These resources report all configured EKM server configured on the library. These servers are used to managed the encryption keys that are used to encrypt/decrypt the data read and written to media in the library.

URI	***aml/system/ekm/servers***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A

Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 51: ekmServersList

Table 231: POST aml/system/ekm/servers

Description: Configure new EKM server resources.

URI	***aml/system/ekm/servers***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 52: ekmServers The Request data will be a little different depending on the EKM server types requested, see the following examples for details: QEKM and TKLM *<ns2:ekmServers* *xmlns:ns2="http://automatedMediaLibrary/">* *<ekmServerType>8</ekmServerType>* *<server>* *<hostName>192.168.10.80</hostName>* *<port>3801</port>* *</server>* *<ssl>false</ssl>* *</ns2:ekmServers>* The ekmServerType must be 8 for QEKM and 32 for TKLM. You must provide at least one server and a maximum of two servers can be configure. If you want to turn on Transport Layer Security (TLS) between the library and the server, set ssl to true. SKM *<ns2:ekmServers* *xmlns:ns2="http://automatedMediaLibrary/">* *<ekmServerType>16</ekmServerType>* *<server>* *<hostName>192.168.20.100</hostName>* *</server>* *<server>* *<hostName>192.168.20.101</hostName>* *</server>* *</ns2:ekmServers>* The ekmServerType must be set to 16. You are required to configure two servers. No port

	number is required since it is hard coded to 6000. KMIP *<ns2:ekmServers* *xmlns:ns2="http://automatedMediaLibrary/">* *<ekmServerType>4</ekmServerType>* *<server>* *<hostName>10.20.169.146</hostName>* *<port>5696</port>* *</server>* *<server>* *<hostName>10.20.169.147</hostName>* *<port>5696</port>* *</server>* *</ns2:ekmServers>* The ekmServerType must be set to 4. You must have at least two servers with no more that 10 configured. You cannot configure a server type if it is already configured. Each server requested must be unique. The ekmServers.ekmPathDiagnosticsInterval element determines if the path diagnostics test will be run and at what interval. The options are 0-60 minutes, where 0 means do not run these tests.
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json Location: aml/system/ekm/servers
Response Data	See Figure 173: WSResultCode

Table 232: GET aml/system/ekm/servers/{type}

Description: Retrieve the EKM servers with the type given by URI path template “type”. The type must be the server type you want to retrieve, see Figure 52: ekmServers, specifically the ekmServerType field.

URI	***aml/system/ekm/servers/{type}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 52: ekmServers

Table 233: PUT aml/system/ekm/servers/{type}

Description: Update the EKM servers with the type given by URI path template “type”. The type must be the server type you want to update, see Figure 52: ekmServers, the ekmServerType field.

You must supply all of the server(s) even the ones that may not have changed. Basically this interface reconfigures the EKM servers for a given server type.

URI	***aml/system/ekm/servers/{type}***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 52: ekmServers
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 52: ekmServers

Table 234: DELETE aml/system/ekm/servers/{type}

Description: Delete the EKM servers with the type given by URI path template “type”. The type must be the server type you want to delete (see Figure 52: ekmServers, the ekmServerType field).

Note: You cannot delete a server type if it is currently being used by a partition.

If you delete the servers then any media that was written too using keys from the servers, can no longer by read.

URI	***aml/system/ekm/servers/{type}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 235: POST aml/system/ekm/servers/{type}/test

Description: Test the EKM servers with the type given by URI path template “type”.

URI	***aml/system/ekm/servers/{type}/test***

Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 53: ekmServerTestResultList

Table 236: GET aml/system/licenses

Description: Retrieve all license resources. This will report all license features supported by the library, including feature that have been installed and those that have not been installed.

URI	***aml/system/licenses***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 97: licenseList

Table 237: POST aml/system/licenses

Description: Add a new licensable feature to the library. The object reported in the request data below should have a valid license key in the “feature” element of the “license” object.

URI	***aml/system/licenses***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 98: license
Response Codes	201, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 97: licenseList

Table 238: GET aml/system/network

Description: Retrieve the network resource information.

URI	*aml/system/network*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 112: network

Table 239: GET aml/system/network/access/certificate

Description: Retrieve the Web Server communication certificate installed on the library.

URI	*aml/system/network/access/certificate*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 16: communicationCertificate

Table 240: POST aml/system/network/access/certificate

Description: Upload the signed certificate that was generated using Table 243: POST aml/system/network/access/certificate/csr.

The multipart form data key is 'file' and the value is the file to be uploaded.

After uploading the signed certificate you will need to activate it (the library web server will start using it), see Table 242: POST aml/system/network/access/certificate/activate.

URI	*aml/system/network/access/certificate*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type multipart/formdata
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 241: GET aml/system/network/access/certificate/activate

Description: Is there an uploaded signed certificate on the library that has not yet been activated.

URI	*aml/system/network/access/certificate/activate*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:test/plain or application/json
Response Data	“true” or “false”

Table 242: POST aml/system/network/access/certificate/activate

Description: Activate the signed certificate uploaded to the library using Table 240: POST aml/system/network/access/certificate. The Web Server will start to use this certificate to authenticate communications with the Web Browser client.

URI	*aml/system/network/access/certificate/activate*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 243: POST aml/system/network/access/certificate/csr

Description: Generate a Certificate Signing Request (CSR) to be signed by a Certificate Authority. Once signed the CA Certificate can then be uploaded to the library.

The communicationCertificate fields that are required for generating the CSR are:

keySize, digestAlgorithm and subject (certificateInformation object). The required certificateInformation fields are: commonName.

The maximum allowed characters for countryCode is 2 all other fields have a maximum length of 128 characters.

URI	*aml/system/network/access/certificate/csr*
Method	**POST**
User Role Access	Admin, Service
Parameters	Optional query parameters are save with the

	following valid values: save=name where name is a file name to be used to save the CSR to. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default name is provided if none is given and is the following, library.csr.
Request Header	Content-Type: application/xml or application/json
Request Data	See Figure 16: communicationCertificate
Response Codes	200, 403
Response Header	Content-Type:application/octet-stream On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	Byte Stream

Table 244: GET aml/system/network/access/certificate/isSigned

Description: Is the certificate installed on the library that is used for web server communications signed by a certificate authority.

URI	***aml/system/network/access/certificate/isSigned***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:text/plain or application/json
Response Data	“true” or “false”

Table 245: GET aml/system/network/interfaces

Description: Retrieve the all network interfaces resources, eth0 and eth2.

URI	***aml/system/network/interfaces***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 116: netInterfaceList

Table 246: GET aml/system/network/interface/{name}

Description: Request the network interface whose name is given by URI path template "name". In the data below name was eth0.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netInterface   xmlns:ns2="http://automatedMediaLibrary/">
   <name>eth0</name>
   <macAddress>00:30:8C:06:78:B5</macAddress>
   <location>N/A</location>
   <duplexMode>Full</duplexMode>
   <autoNegotiate>false</autoNegotiate>
   <speed>1</speed>
   <linkStatus>1</linkStatus>
</ns2:netInterface>
```

URI	***aml*/system/network/interface/{name}**
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 117: netInterface

Table 247: PUT aml/system/network/interface/{name}

Description: Update the network interface whose name is given by URI path template "name".

The client request data below indicates that the interface speed should be changed to 1 (10Mb/s).

To change back to auto negotiation set autoNegotiate element to true and remove the speed element. To set a specific speed (1,2 or 3) the autoNegotiate element must be set to false.

Note: The i6k only supports setting the speed or auto negotiate elements.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netInterface   xmlns:ns2="http://automatedMediaLibrary/">
   <name>eth0</name>
   <autoNegotiate>false</autoNegotiate>
   <speed>1</speed>
</ns2:netInterface>
```

URI	***aml*/system/network/interface/{name}**
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A

Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 117: netInterface
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 117: netInterface

4.10 IPv6 Network Configuration

The IPv6 configuration information and setup is handled much differently than IPv4, since an interface (eth0 or eth2) can be configured to have multiple IPv6 addresses. With IPv4 an interface can only have one IP address.

When you make a configurations request, see Table 248: GET aml/system/network/configurations, a netConfigurationList is returned containing netConfiguration objects:

```
<netConfiguration>
    <name>eth0 or eth2</name>
    <version>1</version> 1(IPv4) or 2(IPv6)
    <hostName>dvt4</hostName> The hostname assigned to this interface
    <type>1</type> -1(Unknown), 0(None), 1(Static), 2(DHCP), 3(DHCP6) and 4(Static and DHCP6)
    <netMask>255.255.248.0</netMask>
    <netGateway>10.20.168.1</netGateway>
    <ipAddress>10.20.171.14</ipAddress>
</netConfiguration>
```

For IPv4 netConfiguration objects, version element = 1, the type element can only be 1(Static) or 2(DHCP) and the netMask, netGateway and ipAddress elements will always contain a valid IPv4 address.

With IPv6 netConfiguration objects, version element =2, there will be only one object that will have a type element of 0(None), 1(Static), 3(DHCP6) or 4(Static and DHCP6) and all the rest of the objects will have a type element of -1(Unknown). The reason for -1 (Unknown) type, is that we cannot determine how this address was created: Manually, DHCP6, Stateless Address Autoconfiguration.

The netConfiguration object that contains a type element that is NOT -1 is used to determine if a static IPv6 address was configured (Manual Configuration) for this interface, whether DHCP6 was configured, whether both static and DHCP6 was configured or whether none, Static nor DHCP6 was configured. If the type is 1 or 4 the object will contain an IP address, which will be the static IP address.

The following object reports that both a static address and DHCP6 is configure on interface eth0, the type element is 4, and the ipAddress reported is the static IP.

```
<netConfiguration>
    <name>eth0</name>
    <version>2</version>
    <hostName>dvt4</hostName>
    <type>4</type>
```

```
        <netMask>64</netMask>
        <netGateway>2001::</netGateway>
        <ipAddress>2001::abcd/64</ipAddress>
 </netConfiguration>
```

When you configure IPv6 (Static, DHCP, Both or None) for a particular interface (eth0 or eth2) you will use the netConfiguration object (see Table 250: PUT aml/system/network/configuration/{name}/{version}). Each time you request an update of the interface, you must specify all of the values you want for the new configuration.

Note: When configuring a static IPv6 address on interface eth2, you do not need to supply a netGateway element, the gateway is not required.

Table 248: GET aml/system/network/configurations

Description: Retrieve the network configurations for all the configured interfaces, eth0 and eth2, on the library.

URI	***aml/system/network/configurations***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 114: netConfigurationList

Table 249: GET aml/system/network/configuration/{name}

Description: Request the network parameters for the interface given by URI path template “name”. The XML Response Data below is for the following request:

http://10.20.171.17/aml/system/network/configuration/eth0

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netConfigurationList   xmlns:ns2="http://automatedMediaLibrary/">
   <netConfiguration>
      <name>eth0</name>
      <location>N/A</location>
      <version>1</version>
      <hostName>dvt8-jb</hostName>
      <domainName>dvt8-jb</domainName>
      <type>1</type>
      <netMask>64</netMask>
      <netGateway>10.20.168.1</netGateway>
      <ipAddress>10.20.171.17</ipAddress>
   </netConfiguration>
   <netConfiguration>
```

```
        <name>eth0</name>
        <location>N/A</location>
        <version>2</version>
        <hostName>dvt8-jb</hostName>
        <domainName>dvt8-jb</domainName>
        <type>3</type>
        <netGateway>0:0:0:0:0:0:0:0</netGateway>
        <ipAddress>2001:db8:ffff:1:230:8cff:fe06:78b5/64</ipAddress>
        <ipAddress>fe80::230:8cff:fe06:78b5/64</ipAddress>
    </netConfiguration>
</ns2:netConfigurationList>
```

URI	*/aml/system/network/configuration/{name}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 114: netConfigurationList

Table 250: PUT aml/system/network/configuration/{name}/{version}

Description: Update the network parameter whose name and version is given by the URI path template “name” and “version”.

The “version” template values are as follows: 1 (IPv4) and 2 (IPv6).

See the IPv4 examples below for more information:

Example 1: Update the hostname to “TestLibrary” using the IPv4 parameters version. The name, version (1-IPv4) and hostName elements must be specified.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netConfiguration  xmlns:ns2="http://automatedMediaLibrary/">
   <name>eth0</name>
   <version>1</version>
   <hostName>TestLibrary</hostName>
</ns2:netConfiguration>
```

Example 2: Set DHCP for IPv4 on eth0. After sending this request you session will no longer be valid and you will be disconnected from the library.

```
<ns2:netConfiguration  xmlns:ns2="http://automatedMediaLibrary/">
   <name>eth0</name>
   <version>1</version>
   <type>2</type>
</ns2:netConfiguration>
```

Example 3: Set a new Static IPv4 address on eth0. After sending this request you session will no longer be valid and you will be disconnected from the library.

```
<ns2:netConfiguration xmlns:ns2="http://automatedMediaLibrary/">
  <name>eth0</name>
  <version>1</version>
  <hostName>dvt8-jb</hostName>
  <type>1</type>
  <ipAddress>10.20.171.17</ipAddress>
</ns2:netConfiguration>
```

See the IPv6 examples below for more information:

Example 4: Set a new Static IPv6 address on interface eth0 where DHCP6 is disabled. If the current IPv6 or DHCP6 address was used to connect to the library you will be disconnected from the library.

Note: When configuring a static IPv6 address on interface eth2, you do not need to supply a netGateway element, the gateway is not required.

```
<ns2:netConfiguration xmlns:ns2="http://automatedMediaLibrary/">
  <name>eth0</name>
  <version>2</version>
  <type>1</type>
  <netMask>64</netMask>
  <netGateway>fd80::</netGateway>
  <ipAddress>fd80::ffaa:abcd</ipAddress>
</ns2:netConfiguration>
```

Example 5: Set DHCP6 on eth2, with no static address, type element = 3.

```
<ns2:netConfiguration xmlns:ns2="http://automatedMediaLibrary/">
  <name>eth2</name>
  <version>2</version>
  <type>3</type>
</ns2:netConfiguration>
```

Example 6: Configure both Static address and DHCP6 for interface eth0. Note the type element is 4 (Static and DHCP).

```
<ns2:netConfiguration xmlns:ns2="http://automatedMediaLibrary/">
  <name>eth0</name>
  <version>2</version>
  <type>4</type>
  <netMask>64</netMask>
  <netGateway>20FF::</netGateway>
  <ipAddress>20FF::ffaa:abcd</ipAddress>
</ns2:netConfiguration>
```

Example 7: Configure new host name for interface eth2 using IPv6. When you configure a new hostname you must supply the current configuration, static and or DHCP6 setting, otherwise they will get removed.

```
<ns2:netConfiguration xmlns:ns2="http://automatedMediaLibrary/">
  <name>eth2</name>
  <version>2</version>
  <hostName>TestLibrary-eth2</hostName>
  <type>4</type>
  <netMask>64</netMask>
  <netGateway>20FF::</netGateway>
  <ipAddress>20FF::ffaa:abcd</ipAddress>
</ns2:netConfiguration>
```

URI	*aml/system/network/configuration/{name}/{version}*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 115: netConfiguration
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 115: netConfiguration

Table 251: GET aml/system/network/dns

Description: Retrieve the Domain Name Servers (DNS) resource. This resource contains the IP address of the Domain Name Server configured on the library. These servers are used to resolve domain names to IP addresses.

URI	*aml/system/network/dns*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 25: DNS

Table 252: PUT aml/system/network/dns

Description: Update the current DNS IP address configuration. To disable DNS just pass down an empty DNS object.

URI	*aml/system/network/dns*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json

Request Data	See Figure 25: DNS
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 25: DNS

Table 253: GET aml/system/network/emailServer

Description: Retrieve the email server configuration resource.

URI	***aml/system/network/emailServer***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 59: emailServer

Table 254: PUT aml/system/network/emailServer

Description: Update the current email server configuration.

Example 1: Change the IP address of the email server. The required elements are server and senderEmailAddress. If authorization is turned on, the authorize element is set to true the accountName and accountPassword are required otherwise authorization will be turned off as in this example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:emailServer    xmlns:ns2="http://automatedMediaLibrary/">
     <server>10.20.169.9</server>
   <senderEmailAddress>john.doe@company.com</senderEmailAddress>
</ns2:emailServer>
```

Example 2: Disable the email server. This will prevent the library from sending email notifications and other email features. When the email server is disabled all the email server information that was persisted on the library will be deleted.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:emailServer    xmlns:ns2="http://automatedMediaLibrary/">
</ns2:emailServer>
```

URI	***aml/system/network/emailServer***
Method	**PUT**

User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 59: emailServer
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 59: emailServer

Table 255: GET aml/system/network/internal

Description: Retrieve the internal network resource. This resource provides information on the currently configured network address of the libraries internal network. It also provides the network address options that are available if a conflict exists between the internal and external network the library is connected to.

URI	***aml/system/network/internal***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 90: internalNetwork

Table 256: PUT aml/system/network/internal

Description: Update the internal network resource. This should only be used if there is a conflict between the libraries internal network and the network the library is connected to. The current element of the internalNetwork object is used to change the internal network. You must use one of the network address options provided by the GET method on this interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:internalNetwork xmlns:ns2="http://automatedMediaLibrary/">
  <current>10.247.240.0</current>
</ns2:internalNetwork>
```

URI	***aml/system/network/internal***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 90: internalNetwork
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 90: internalNetwork

Table 257: GET aml/system/network/ipv6/mode

Description: Retrieve the IPv6 current mode (enabled/disabled).

A single string value will be returned and the possible values are:

1 (enabled) or 2 (disabled)

Note: This interface has been deprecated, IPv6 is always enabled.

URI	***aml/system/network/ipv6/mode***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:text/plain or application/json
Response Data	Will always return “1”, enabled.

Table 258: PUT aml/system/network/ipv6/mode

Description: Change IPv6 mode, 1 (enable) or 2 (disable).

Note: This interface has been deprecated, you can no longer disable IPv6.

URI	***aml/system/network/ipv6/mode***
Method	**PUT**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	Content-Type:text/plain or application/json
Request Data	1 or 2
Response Codes	410
Response Header	Content-Type:text/plain or application/json
Response Data	See Figure 173: WSResultCode

Table 259: POST aml/system/network/emailServer/test

Description: Test if email server is valid. The required elements are shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:emailServer    xmlns:ns2="http://automatedMediaLibrary/">
    <server>10.20.169.9</server>
    <senderEmailAddress>admin@quantum.com</senderEmailAddress>
    <testEmailAddress>john.doe@company.com</testEmailAddress>
</ns2:emailServer>
```

URI	***aml/system/network/emailServer/test***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A

Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 59: emailServer
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 260: GET aml/system/network/snmp/MIBs

Description: Retrieve the libraries SNMP MIB's file.

URI	***aml/system/network/snmp/MIBs***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Optional query parameter is save, with the following valid values: • save="name" where name is a file name to use to save the MIB file to. The file format will be ZIP. The save="name" query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default "name" is provided if none is given and has the following format: ScalarI6K_SNMP_MIBs.zip (i6k) or iQ-Series_SNMP_MIBs.mib (non-Scalari6k products)
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type: application/octet-stream On success and when save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	Octet stream (ZIP format)

Table 261: POST aml/system/network/snmp/MIBs/email

Description: Email the libraries SNMP MIB's file.

URI	***aml/system/network/snmp/MIBs/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml, application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type: application/xml, application/json
Response Data	See Figure 173: WSResultCode

Table 262: GET aml/system/notifications/contact

Description: Retrieve the library contact information resource.

URI	*aml/system/notifications/contact*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 19: contactInformation

Table 263: PUT aml/system/notifications/contact

Description: Update the contact information resource. If any element of the contactInformation object is not included the resulting value will be an empty string.

URI	*aml/system/notifications/contact*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 19: contactInformation
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 19: contactInformation

Table 264: GET aml/system/notifications/emailRecipients

Description: Retrieve a list of emailRecipients resources. This is a list of e-mail address that are stored on the library.

URI	*aml/system/notifications/emailRecipients*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 60: emailRecipientList

Table 265: POST aml/system/notifications/emailRecipients

Description: Create a new emailRecipient resource. The required element is address.

URI	*aml/system/notifications/emailRecipients*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json Location: aml/system/notifications/emailRecipients/email address
Request Data	See Figure 61: emailRecipient
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 266: GET aml/system/notifications/emailRecipient/{id}

Description: Retrieve the emailRecipient resource whose id is given by the URI path template “id”. The id can be either the object id or address.

URI	*aml/system/notifications/emailRecipient/{id}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 61: emailRecipient

Table 267: DELETE aml/system/notifications/emailRecipient/{id}

Description: Delete the emailRecipient resource whose id is given by the URI path template “id”. The id can be either the object id or address.

URI	*aml/system/notifications/emailRecipient/{id}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404

Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 268: GET aml/system/notifications/heartbeats

Description: Retrieve a list of heartbeatNotification resource. This feature is only available on i6k.

Note: The following data is contained in the notification email.

Library Heartbeat
Library Name: Sales
Library IPv4 Address: 10.10.0.100
Library Serial Number: 2U31000001
Heartbeat Interval: 60 minutes
RAS Status:
  Connectivity : Good
  Control : Failed
  Media : Good
  Drives : Failed
  Power : Good
  Robotics : Good
Library State: Online/Ready
Date: Fri Nov 07 20:36:08 GMT 2014

URI	***aml/system/notifications/heartbeats***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 84: heartbeatNotificationList

Table 269: POST aml/system/notifications/heartbeats

Description: Create a new heartbeatNotification resource. Required fields are: interval and emailAddress. This feature is only available on i6k.

URI	***aml/system/notifications/heartbeats***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json

Request Data	See Figure 85: heartbeatNotification
Response Codes	201, 403
Response Header	Content-Type:application/xml or application/json Location: aml/system/notifications/heartbeat/{id}
Response Data	See Figure 173: WSResultCode

Table 270: GET aml/system/notifications/heartbeat/{id}

Description: Retrieve the heartbeatNotification resource whose id is given by the URI path template “id”. This feature is only available on i6k.

URI	***aml/system/notifications/heartbeat/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 85: heartbeatNotification

Table 271: PUT aml/system/notifications/heartbeat/{id}

Description: Modify the heartbeatNotification resource whose id is given by the URI path template “id”. This feature is only available on i6k.

URI	***aml/system/notifications/heartbeat/{id}***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 85: heartbeatNotification
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 85: heartbeatNotification

Table 272: DELETE aml/system/notifications/heartbeat/{id}

Description: Delete the heartbeatNotification whose id is given by the URI path template “id”. This feature is only available on i6k.

URI	***aml/system/notifications/heartbeat/{id}***
Method	**DELETE**

User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 273: GET aml/system/notifications/mediaSecurity

Description: Get the media security policy resource.

URI	*aml/system/notifications/mediaSecurity*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 108: mediaSecurityPolicy

Table 274: PUT aml/system/notifications/mediaSecurity

Description: Update the media security policy resource. You can configure the library to automatically notify you via a RAS ticket when media is moved in or out of the library, either intentionally or unintentionally. First, you must choose under which circumstances you wish to be notified, and then you must enable automatic inventory on the library.

Note: The feature requires an Advanced Reporting license.

URI	*aml/system/notifications/mediaSecurity*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: application/xml or application/json
Request Data	See Figure 108: mediaSecurityPolicy
Response Header	Content-Type: application/xml or application/json
Response Codes	200, 403, 404
Response Data	See Figure 108: mediaSecurityPolicy

Table 275: GET aml/system/notifications/reports

Description: Retrieve a list of reportNotification resources. These report notifications are scheduled to be e-mailed to recipients on a periodic basis.

URI	*aml/system/notifications/reports*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 144: reportNotificationList

Table 276: POST aml/system/notifications/reports

Description: Create a new report notification resource. The required reportNotification object elements are:

emailAddress – The e-mail address of the recipient you want this report to be sent to.

reportTemplateName – A valid reportTemplate name, the available templates can be found at the following URI Table 307: GET aml/system/reports/templates.

interval – See Figure 145: reportNotification.

If not specified the enable element will default to false, so it is recommended to include this and set it to true.

URI	*aml/system/notifications/reports*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json Location: aml/system/notifications/report/{id}
Request Data	See Figure 145: reportNotification
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 277: GET aml/system/notifications/report/{id}

Description: Retrieve the reportNotification resource whose id is given by the URI path template “id”.

URI	*aml/system/notifications/report/{id}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A

Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 145: reportNotification

Table 278: PUT aml/system/notifications/report/{id}

Description: Modify the reportNotification resource whose id is given by the URI path template “id”.

URI	***aml/system/notifications/report/{id}***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 145: reportNotification
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 145: reportNotification

Table 279: DELETE aml/system/notifications/report/{id}

Description: Delete the reportNotification whose id is given by the URI path template “id”.

URI	***aml/system/notifications/report/{id}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 280: GET aml/system/notifications/snmpTraps

Description: Retrieve the list of snmp trap notification resources.

URI	***aml/system/notifications/snmpTraps***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A

Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 157: trapNotificationList

Table 281: POST aml/system/notifications/snmpTraps

Description: Create a new trapNotification resource. The required fields are host and port.

URI	***aml/system/notifications/snmpTraps***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 158: trapNotification
Response Codes	201, 403
Response Header	Content-Type:application/xml or application/json Location: aml/system/notifications/snmpTrap/{id}
Response Data	See Figure 173: WSResultCode

Table 282: GET aml/system/notifications/snmpTrap/{id}

Description: Retrieve the snmp trap notification resources whose id is given by the URI path template “id”.

URI	***aml/system/notifications/snmpTrap/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 158: trapNotification

Table 283: PUT aml/system/notifications/snmpTrap/{id}

Description: Update the snmp trap notification resources whose id is given by the URI path template “id”.

URI	**aml/system/notifications/snmpTrap/{id}**
Method	PUT
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 158: trapNotification
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 158: trapNotification

Table 284: DELETE aml/system/notifications/snmpTrap/{id}

Description: Delete the snmp trap notification resources whose id is given by the URI path template “id”.

URI	***aml/system/notifications/snmpTrap/{id}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 285: GET aml/system/notifications/tickets

Description: Retrieve a list of rasNotification resources. These RAS notifications are e-mailed to recipients when a RAS ticket is opened. You can also filter the tickets sent depending on the severity of the ticket.

URI	***aml/ system/notifications/tickets***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 139: rasNotificationList

Table 286: POST aml/system/notifications/tickets

Description: Create a new rasNotification resource. The required rasNotification object elements are:

emailAddress – e-mail address to send the notification to.

enabled – set to true, otherwise the notification will not be sent.

severity1 … severity5 – Set the severities to true that you want to receive notification on.

URI	***aml/system/notifications/tickets***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 140: rasNotification
Response Codes	201, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 287: GET aml/system/notifications/ticket/{id}

Description: Retrieve the rasNotification resource whose id is given by the URI path template “id”.

URI	***aml/system/notifications/ticket/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 140: rasNotification

Table 288: PUT aml/system/notifications/ticket/{id}

Description: Modify the rasNotification resource whose id is given by the URI path template “id”.

URI	***aml/system/notifications/ticket/{id}***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json

Request Data	See Figure 140: rasNotification
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 140: rasNotification

Table 289: DELETE aml/system/notifications/ticket/{id}

Description: Delete the rasNotification whose id is given by the URI path template “id”.

URI	*aml/system/notifications/ticket/{id}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 290: POST aml/system/operations/reboot

Description: Reboot the library.

URI	*aml/system/operations/reboot*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 291: POST aml/system/operations/shutdown

Description: Shutdown the library. After the library has been shutdown, you will need to physically power the library down by using the power button on the libraries Control Module.

URI	*aml/system/operations/shutdown*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A

Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 292: GET aml/system/policy/externalApplicationServers

Description: Retrieve a list of externalApplicationServers resources. These servers can be used when configuring Extended Data Life Management (EDLM) policies. If StorNext Storage Manager is managing your partition, you can configure a policy to use StorNext with EDLM to automatically copy data off of bad or suspect tapes or to trigger media scans. This feature requires an external application server(s) (StorNext) to be configured.

URI	***aml/system/policy/externalApplicationServers***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 69: externalApplicationServersList

Table 293: POST aml/system/policy/externalApplicationServers

Description: Configure a new external application (StorNext) servers. The following externalApplicationServers elements are required:

name: the name value must be unique.

server: at least one server element is required, and both name and port elements of the server are required. The externalApplicationName is required and must point to a configured externalApplication plugin, which the library will use to communicate with the external servers.

To see what externalApplication are configured use this URI: Table 326: GET aml/system/software/externalApplications.

URI	***aml/system/policy/externalApplicationServers***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 70: externalApplicationServers
Response Codes	201, 403
Response Header	Content-Type:application/xml or application/json Location: aml/system/externalApplicationServers/{name}

Response Data	See Figure 173: WSResultCode

Table 294: GET aml/system/policy/externalApplicationServer/{name}

Description: Retrieve the externalApplicationServers with the name given by URI path template “name”.

URI	***aml/system/policy/externalApplicationServer/{name}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 70: externalApplicationServers

Table 295: PUT aml/system/policy/externalApplicationServer/{name}

Description: Modify the externalApplicationServers with the name given by URI path template “name”.

The example below will change the name of the externalApplicationServers from EDLMservers to StorNext and reconfigure the policy to use server 10.20.169.88 only using the following URI:

“aml/system/policy/externalApplicationServer/EDLMservers”

```
<ns2:externalApplicationServers  xmlns:ns2="http://automatedMediaLibrary/">
      <name>StorNext</name>
      <server>
         <name>10.20.169.88</name>
         <port>61776</port>
      </server>
</ns2:externalApplicationServers>
```

If you just want to change the name, then just specify the name

```
<ns2:externalApplicationServers  xmlns:ns2="http://automatedMediaLibrary/">
      <name>StorNext5.0</name>
</ns2:externalApplicationServers>
```

If you just want to change the external application name plugin

```
<ns2:externalApplicationServers  xmlns:ns2="http://automatedMediaLibrary/">
      externalApplicationName>snapi-2.0.1</externalApplicationName>
</ns2:externalApplicationServers>
```

You can specify any combination of the above.

URI	***aml/system/policy/externalApplicationServer/{name}***
Method	**PUT**
User Role Access	Admin, Service

Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 70: externalApplicationServers
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 70: externalApplicationServers

Table 296: DELETE aml/system/policy/externalApplicationServer/{name}

Description: Delete the externalApplicationServers with the name given by URI path template “name”.

URI	***aml/system/policy/externalApplicationServer/{name}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 297: GET aml/system/ras

Description: Retrieve the Reliability, Accessibility and Serviceability (RAS) group status resource list.

Note: On the i6k the group 0 (Library) is not include in the list.

URI	***aml/system/ras***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 131: RASGroupStatusList

Table 298: GET aml/system/ras/{group}

Description: Retrieve the RAS group status resource given by URI path template “group”.

Note: On the i6k to get the overall Library RAS status, use 0 for the group.

URI	*aml/system/ras/{group}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 132: RASGroupStatus

Table 299: GET aml/system/ras/tickets

Description: Retrieve the ticket summary resource instances using the query parameter filters.

To retrieve all drive and control ticket summaries that are currently opened you would specify the following:

“aml/system/ras/tickets?group=20&state=4”

If no query parameters are given, all tickets are reported.

Treat the above group, status and state values as bit masks, like in the example above group 20 is 4(control) + 16(drive).

URI	*aml/system/ras/tickets*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameters are group, status, state. • group values 255 (All), 2 (Connectivity), 4 (Control), 8 (Media), 16 (Drive), 32 (Power), 64 (Robotics) • status values 255 (All) 4 (Failed), 8 (Degraded), 16 (Warning) • state values 255 (All), 4 (Opened), 16 (Closed), 32 (Verified)

	These Query parameters work like a bit mask, for instance if you want find tickets belonging to Control and Media you would specify 12 (4+8).
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 136: RASTicketList

Table 300: GET aml/system/ras/ticket/{id}

Description: Retrieve the ticket resource instance with the value given by the URI path template “id”.

URI	***aml/system/ras/ticket/{id}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Optional query parameter is save, with the following valid values: • save=”name” where name is a file name to use to save the ticket information and resolution to. The file format will be in zip format. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: ticket_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.zip (i6k) or ticket_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.tar (non-Scalari6k products)
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: application/xml, application/json or application/octet-stream On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 137: RASTicket

Table 301: PUT aml/system/ras/ticket/{id}

Description: Change the ticket resource instance state with the value given by URI

path template “id”. The state will be changed to Closed. Only tickets that are in an ‘Open’ state can be closed.

URI	*aml/system/ras/ticket/{id}*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml or application/json
Response Data	See Figure 137: RASTicket

Table 302: POST aml/system/ras/ticket/{id}/email

Description: Email the ticket information to the recipients contained in the email object.

The email will have an attachment containing the resolution and the body will contain ticket information.

URI	*aml/system/ras/ticket/{id}/email*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 303: GET aml/system/ras/ticket/{id}/history

Description: Retrieve the ticket resource history list of related tickets for the ticket with the “id” given by URI path template “id”.

URI	*aml/system/ras/ticket/{id}/history*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 136: RASTicketList

Table 304: GET aml/system/ras/ticket/{id}/reports

Description: Retrieve the ticket resource reports list for the ticket with the “id” given by URI path template “id”.

URI	***aml/system/ras/ticket/{id}/reports***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 135: RASTicketReports

Table 305: GET aml/system/ras/ticket/{id}/resolution

Description: Retrieve the RAS ticket resolution information. You can request a byte stream or a html page.

URI	***aml/system/ras/ticket/{id}/resolution***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Optional query parameter is save with the following valid values: • save=”name” where name is a file name to use to save the ticket resolution to. The file format is HTML The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: “resolution.html” for i6k and “resolution.htm” for non-Scalari6k products.
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type: application/octet-stream or text/html On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	Byte Stream or text/html

Table 306: GET aml/system/reports

Description: Retrieve the list of supported reports.

URI	*aml/system/reports*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 141: reportList

Table 307: GET aml/system/reports/templates

Description: Retrieve the list of report template resource instances. A report template is used to save report filtering data that can be used a later date to query a report using these saved filters.

URI	*aml/system/reports/templates*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 146: reportTemplateList

Table 308: POST aml/system/reports/templates

Description: Create a new reportTemplate resource. When you specify a reportCriteria for the new reportTemplate, the supported reportCriteria options will depend on the report type, you will need to look at the following URI's to see what reportCriteria options are available for each report. The reportTemplate name can only contain the following characters A-Z a-z 0-9 _ and spaces.

Table 117: GET aml/media/reports/tapeAlerts
Table 119: GET aml/media/reports/usage
Table 115: GET aml/media/reports/securityEvents
Table 111: GET aml/media/reports/crossPartitionMoves
Table 113: GET aml/media/reports/inventory
Table 226: GET aml/system/ekm/reports/audit/mediaStatus
Table 228: GET aml/system/ekm/reports/audit/partitionActivity

The reportTemplate name can only contain the following characters A-Z a-z 0-9 _ and spaces. The maxium number of character allowed is 64.

URI	*aml/system/reports/templates*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 147: reportTemplate
Response Codes	201, 403
Response Header	Content-Type:application/xml or application/json Location: aml/system/reports/templates/{id}
Response Data	See Figure 146: reportTemplateList

Table 309: GET aml/system/reports/template/{name}

Description: Retrieve the report template resource instance with username name.

URI	*aml/system/reports/template/{name}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 147: reportTemplate

Table 310: PUT aml/system/reports/template/{name}

Description: Update a reportTemplate resource instance with the name represented by the parameter name.

If you are modifying the reportTemplate name you need to specify the current reportTemplate id.

URI	*aml/system/reports/template/{name}*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	Figure 147: reportTemplate
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 147: reportTemplate

Table 311: DELETE aml/system/reports/template/{name}

Description: Delete the reportTemplate resource instance whose name is represented by the URI path template “name”.

URI	***aml*/system/reports/template/{name}**
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 312: GET aml/system/saveRestore

Description: Create a restore (library configuration) image. This image will be the payload in the response body. This image can be used to restore the libraries configuration information.

URI	***aml*system/saveRestore**
Method	**GET**
User Role Access	Admin, Service
Parameters	Optional query parameter is save, with the following valid values: • save=”name” where name is a file name to use to save the rescue image. The file format will be gzip-ed tar file ‘.tar.gz’. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: saveRescueConfigurationImage_librarySerialNumber_yyyy-MM-dd_HH.mm.ss.tar.gz
Request Header	N/A
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type: application/x-tar On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	Byte Stream (Compressed tar format)

Table 313: POST aml/system/saveRestore

Description: Restore a saved configuration. Upload the configuration file using multipart form data, where the key is 'file' and the value is the file to be uploaded. After the restore is completed the library will be rebooted.

URI	*/amlsystem/saveRestore*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: multipart/form-data
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml, application/json
Response Data	See Figure 173: WSResultCode

Table 314: GET aml/system/saveRestore/rescue

Description: Determine if a rescue image has been created on the library.

URI	*/aml/system/saveRestore/rescue*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:text/plain or application/json
Response Data	true or false

Table 315: POST aml/system/saveRestore/rescue

Description: Create a rescue image on the library's file system. This image contains the library's current configuration information.

URI	*/aml/system/saveRestore/rescue*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type: application/xml, application/json
Response Data	See Figure 173: WSResultCode

Table 316: PUT aml/system/saveRestore/rescue

Description: Restore a library's configuration using the rescue image. The rescue

image allows you to roll back the library's configuration settings to a previous state.

Note: After the operation has completed the library will automatically be rebooted.

URI	*aml/system/saveRestore/rescue*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type: application/xml, application/json
Response Data	See Figure 173: WSResultCode

Table 317: GET aml/system/saveRestore/revert

Description: Determine if a revert image has been created on the library. The revert image is automatically created and stored locally as the first step of any restore or rescue operation. The purpose of the Revert process is to revert to the last configuration that was used before a restore image was applied. If an incorrect restore image was applied, the Revert feature allows the MCB to revert back to its prior configuration.

URI	*aml/system/saveRestore/revert*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:text/plain or application/json
Response Data	true or false

Table 318: PUT aml/system/saveRestore/revert

Description: Restore a library's configuration using the Revert image.

Note: After the operation has completed the library will automatically be rebooted.

URI	*aml/system/saveRestore/revert*
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 319: GET aml/system/sensors

Description: Retrieve the sensor resource instances for the library. The sensor list will include cooling, power, temperature, voltage and humidity sensors.

URI	***aml/system/sensors***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	Query parameter names are “type” with the following valid values • type = “cooling” or “temperature” or “voltage” or “power” or “humidity”
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 152: sensorList

Table 320: GET aml/system/snapshot

Description: Capture a snapshot of type specified by the given query parameter. If no query parameter is specified, an extended snapshot is captured.

The snapshot will be return in the body of the response.

URI	***aml/system/snapshot***
Method	**GET**
User Role Access	Admin, Service
Parameters	Query parameters are type and save where the valid values are as follows: • type=basic or extended. • save=”name” where name is a file name to use to save the snap shot to. The file format will be tgz. The save=”name” query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default “name” is provided if none is given and has the following format: Quantum_i6000_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss_snapshot.tgz
Request Header	N/A
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type: application/x-tar, application/xml or application/json On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done

Response Data	Compress tar file (byte stream).

Table 321: POST aml/system/snapshot/email

Description: Email Snapshot report.

The information will be in an email attachment and the file format will be TGZ.

URI	***aml/system/snapshot/email***
Method	**POST**
User Role Access	Admin, Service
Parameters	Query parameters are type where the valid values are as follows: • type=basic or extended. If no type is given an extended snapshot will be captured.
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 322: GET aml/system/software

Description: Retrieve the current library firmware components/distributions and versions saved on the library. The list will contain the currently installed version, component = Current, the previous distribution that was installed, component = Rollback and a distribution that has been uploaded to the library, component = Uploaded. This uploaded component can be installed at any time if the version of this component is greater than the currently installed version.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareList   xmlns:ns2="http://automatedMediaLibrary/">
   <lastInstallDate>2014-06-02 09:57:50 -0600</lastInstallDate>
   <firmware>
      <component>Current</component>
      <version>665H.TS07401</version>
   </firmware>
   <firmware>
      <component>Rollback</component>
      <version>665H.TS06701</version>
   </firmware>
   <firmware>
      <component>Uploaded</component>
      <version></version>
   </firmware>
</ns2:firmwareList>
```

URI	*aml/system/software*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 74: firmwareList

Table 323: POST aml/system/software

Description: Upload the software component/distribution file using multipart form data, where the key is 'file' and the value is the file to be uploaded. This will upload the distribution software to the library, it does not install this new distribution, you must now use the interface, Table 324: GET aml/system/software/operations/update to install this distribution.

Note: For i6k libraries the description element of the WSResultCode object will contain a colon ":" separated name value pair list as following, for successful request:

STATUS="OK" or "WARNING"

MESSAGE="Some message"

Example: <description>STATUS=OK:MESSAGE=Signed by Production Certificate</description>

URI	*aml/system/software*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: multipart/form-data
Request Data	Software distribution file
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 324: GET aml/system/software/operations/update

Description: Retrieve the progress of the current software update operation. This will report the current installation status. Initially the status element may be empty but as the installation progresses it will report a % complete. When the software update has completed the status components will all report 100% or Failed if there was a problem.

The updateState element will report the overall state of the software update, see Figure 80: firmwareStatusList for details.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareStatusList  xmlns:ns2="http://automatedMediaLibrary/">
   <updateState>2</updateState>
   <firmwareStatus>
      <component>FDM</component>
      <status>Awaiting RCU -- Overall  4%</status>
   </firmwareStatus>
   <firmwareStatus>
      <component>AMC</component>
      <status>100%</status>
   </firmwareStatus>
   <firmwareStatus>
      <component>MCB</component>
      <status>100%</status>
   </firmwareStatus>
   <firmwareStatus>
      <component>RCU</component>
      <status>5%</status>
   </firmwareStatus>
   <firmwareStatus>
      <component>CMB</component>
      <status>100%</status>
   </firmwareStatus>
   <firmwareStatus>
      <component>EEB</component>
      <status>100%</status>
   </firmwareStatus>
</ns2:firmwareStatusList>
```

Note: After the software update has completed successfully the library will automatically be rebooted, so the connection to the library will be lost and any requests to the library at this time will receive a 503 status code (Service Unavailable).

URI	***aml*/system/software/operations/update**
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 80: firmwareStatusList

Table 325: POST aml/system/software/operations/update

Description: Start the software installation for the given firmware object. In the example below we decided to install the previously installed software distribution (Rollback).

Note: After the install/update has completed the library will automatically be rebooted.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmware   xmlns:ns2="http://automatedMediaLibrary/">
   <component>Rollback</component>
   <version>665H.TS07401</version>
</ns2:firmware>
```

URI	*aml/system/software/operations/update*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 75: firmware
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 326: GET aml/system/software/externalApplications

Description: Retrieve a list of externalApplication resources. This will report the StorNext plugins installed on the library.

URI	*aml/system/software/externalApplications*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 67: externalApplicationList

Table 327: POST aml/system/software/externalApplications

Description: Upload external application file using form data, where the key is 'file' and the value is the filename.

URI	*aml/system/software/externalApplications*
Method	**GET**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type: multipart/form-data
Request Data	The external application file
Response Codes	201, 403
Response Header	Content-Type:application/xml or application/json Location: aml/system/software/externalApplication/{name}

Response Data	See Figure 173: WSResultCode

Table 328: GET aml/system/software/externalApplication/{name}

Description: Retrieve the externalApplication resources given by the URI path template "name".

URI	*aml/system/software/externalApplication/{name}*
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 68: externalApplication

Table 329: DELETE aml/system/software/externalApplication/{name}

Description: Delete the external application file given by the URI path template "name".

URI	*aml/system/software/externalApplication/{name}*
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 330: GET aml/users

Description: Retrieve the user resource instances.

URI	*aml/users*
Method	**GET**
User Role Access	Admin, Service, User (If User then only show this users information)
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 168: userList

Table 331: POST aml/users

Description: Create a new user resource instance. You must specify the name, password and role. The valid user roles are 0 (ADMIN) or 1 (USER). If you do not specify a role element an ADMIN user will be created.

The following user names are reserved and cannot be used to create a new user: service, admin, cvtl and security.

The name element only allows the following characters:

"A-Za-z0-9 _" for HP branded libraries "A-Za-z0-9\\. _@-".

Note: When your create a new user the name will be stored in all lower case characters, because joe = Joe = JOE.

The password element accepts: all printable characters except back tick and tilde.

The minimum number of characters for password is 8 and the maximum number of characters for name or password is 64.

URI	***aml/users***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 169: user
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 169: user

Table 332: GET aml/users/ldap

Description: Retrieve the LDAP configuration resource.

URI	***aml/users/ldap***
Method	**GET**
User Role Access	Admin, Service, User (If User request, then only return information on whether LDAP is enabled/disabled)
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 92: ldap

Table 333: PUT aml/users/ldap

Description: Configure the LDAP resource. Initially all elements of the ldap object must be completed except for the alternateServer which may be null. In the case where LDAP is being disabled, only the enable element is required. After the LDAP configuration is saved on the library, the library will persist this data. When the user disables LDAP the configuration is still persisted but it is not used. If you change the state from disabled to enabled you will still need to pass down all the required elements even though the library has this data persisted. The exception to this is the ldap.searchUserPassword element which does not need to be included; if it is empty or null we will use the persisted value.

URI	***aml/users/ldap***
Method	**PUT**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 92: ldap
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 92: ldap

Table 334: POST aml/users/ldap

Description: Upload LDAP certificate(s) to the library. These certificates are generally distributed public keys from the LDAP server. When uploading the file(s), the key is 'file' and the value is the 'file name'.

Note: This interface has been deprecated and is being replaced with, Table 335: POST aml/users/ldap/certificates

URI	***aml/users/ldap***
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:multipart/form-data
Request Data	LDAP Certificate file
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 335: POST aml/users/ldap/certificates

Description: Upload LDAP certificate(s) to the library. These certificates are generally distributed public keys from the LDAP server. When uploading the file(s), the key is 'file'

and the value is the ‘file name’.

URI	*aml/users/ldap/certificates*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:multipart/form-data
Request Data	LDAP Certificate file
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 336: POST aml/users/ldap/test

Description: Test the LDAP user and configuration settings provided in the request. When testing LDAP configuration, you are not required to provide a user or password in the ldapTest object. If you want to verify the LDAP user name and password you are required to provide the user and password and also the LDAP configuration settings, ldapTest.Ldap.

URI	*aml/users/ldap/test*
Method	**POST**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 93: ldapTest
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 337: POST aml/users/login

Description: Login to the library. A successful login returns a session cookie that need to be use for successive requests. If a service user is currently logged in a 503 status code is returned and the login will be denied.

If the maximum number of ADMIN user logins has been exceeded the customCode element of the WSResultCode will be set to 1.

URI	*aml/users/login*
Method	**POST**
User Role Access	Admin, Service, User
Parameters	The form parameters are name, password, ldap and forceLogin. The valid values for each parameter are as follows: • name = valid user name

	• password = the user password • ldap = “true” or “false” • forceLogin = “true” or “false” • ldap=true, if you want to use LDAP to authenticate The forceLogin parameter only applies to an admin user. This can be used when the number of admin logins has exceed the maximum allowed.
Request Header	Content-Type: application/x-www-form-urlencoded
Request Data	HTTP form data
Response Codes	200, 404, 503
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 338: POST aml/users/login

Description: Login to the library using the user XML or JSON object. A successful login returns a session cookie that need to be use for successive requests. If a service user is currently logged in a 503 status code is returned and the login will be denied.

URI	***aml/users/login***
Method	**POST**
User Role Access	Admin, Service, User
Parameters	The query parameters forceLogin. The valid values for each parameter are as follows: • forceLogin = “true” or “false” The forceLogin parameter only applies to an admin user. This can be used when the number of admin logins has exceed the maximum allowed. If used it will force one of the currently logged in admins to be logged out. If the maximum number of ADMIN user logins has been exceeded the customCode element of the WSResultCode will be set to 1.
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 169: user
Response Codes	200, 404, 503
Response Header	Content-Type:application/xml or application/json Warning: Default Password Supplied. This header will be returned if it is the first login for a new user or if an Admin has reset the users password. This is not supported on the i6k.

Response Data	See Figure 169: user

Table 339: DELETE aml/users/login

Description: Logout of the library web services.

URI	***aml/users/login***
Method	**DELETE**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 340: GET aml/users/reports/login

Description: Retrieve the list of user login activity records.

URI	***aml/users/reports/login***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	The following query parameters are supported, start, length, period, date, save with the following values: • start=0-n • length=1-n or -1 for all records • period=the last number of days to include in the report. So if you want to report for the last week, you would specify 7. • date=At what date you want to start your query. The data returned will include all records that are equal or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z”, the Z (time zone) will be ignored.. • save=”name” where name is a file name to use to save the tape alert information to. The file format will be CSV. The data separator is a “!”

	(exclamation point character) not a comma ",". The save="name" query parameter should be used by a client browser to allow the data to be saved by the browser to a file. A default "name" is provided if none is given and has the following format: loginActivity_librarySerialNumber_ yyyy-MM-dd_HH.mm.ss.csv If no query parameters are used the request will return all the login activity data.
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type: application/xml, application/json, text/plain, application/octet-stream On success and save= parameter is used Cookie: name= FileDownloadingProgressCookie, value=Done
Response Data	See Figure 99: loginActivityList or octet-stream or text

Table 341: POST aml/users/reports/login/email

Description: Email the list of user login activity records.

The information will be in an email attachment and the file format will be CSV. The data separator is a "!" (exclamation point character) not a comma ",".

URI	***aml/users/reports/login/email***
Method	**GET**
User Role Access	Admin, Service
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 58: email
Response Codes	200, 403
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 342: GET aml/users/sessions

Description: Retrieve the user sessions (current logged in users) resources.

URI	***aml/users/sessions***

Method	**GET**
User Role Access	Admin, Service, User (If User return only this users session information)
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 170: userSessionList

Table 343: DELETE aml/users/session/{id}

Description: Log off the user with the id given by URI path template “id”. The user id's are reported in the userSession objects URI aml/users/sessions. Only uses with Admin privileges (role 0) can log off other users.

URI	***aml/users/session/{id}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 344: GET aml/user

Description: Retrieve the user resource instance based on the cookie header parameter.

URI	***aml/user***
Method	**GET**
User Role Access	Admin, Service, User (If User return on the User information)
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 169: user

Table 345: GET aml/user/{name}

Description: Retrieve the user resource instance with username name. To retrieve the user who name is JohnDoe use the following URI: “aml/user/JohnDoe”.

URI	***aml/user/{name}***
Method	**GET**
User Role Access	Admin, Service, User
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 169: user

Table 346: PUT aml/user/{name}

Description: Update a user resource instance with the username represented by the parameter name.

The name element only allows the following characters:

“A-Za-z0-9 _” for HP branded libraries “A-Za-z0-9\\. _@-“

If changing the password element the valid characters are all printable characters except back tick and tilde.

The minimum number of characters for password is 8 and the maximum number of characters for name or password is 64.

A standard/regular (role = 1) user can only modify the password. So in this case you will need to set the role element of the user object to -1 so it will be ignored.

When an ADMIN user (role = 0) is modifying a standard user you must always specify the role element, because if you don’t the role element will default to 0 which mean that standard user will now have ADMIN privileges.

Also you must always specify the partition access list, because if you don’t that user will have access to no partitions.

The password element can be null which will indicate that no password change is to be made.

URI	***aml/user/{name}***
Method	**PUT**
User Role Access	Admin, Service, User (If User, then only allow the user to change his/her password)
Parameters	N/A
Request Header	Content-Type:application/xml or application/json
Request Data	See Figure 169: user
Response Codes	200, 403, 404

Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

Table 347: DELETE aml/user/{name}

Description: Delete the user resource instance whose username is represented by the URI path template “name”.

URI	***aml/user/{name}***
Method	**DELETE**
User Role Access	Admin, Service
Parameters	N/A
Request Header	N/A
Request Data	N/A
Response Codes	200, 403, 404
Response Header	Content-Type:application/xml or application/json
Response Data	See Figure 173: WSResultCode

5. Appendix

5.1 Resource Objects

The following figures list the supported resource objects (XML representation) available by the Web Services interfaces.

Figure 1: accessDeviceList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accessDeviceList   xmlns:ns2="http://automatedMediaLibrary/">
   List of access devices, see Figure 2: accessDevice
</ns2:accessDeviceList>
```

Figure 2: accessDevice

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accessDevice   xmlns:ns2="http://automatedMediaLibrary/">
    <serialNumber>F001396025</serialNumber>          <!-- drive or partition serial number -->
    <type>1</type>                                   <!--1 (drive) 2 (Partition) -->
    <port>
      <id>1</id>
      <access>true</access>
    </port>
     <port>
       <id>2</id>
       <access>false</access>
     </port>
</ns2:accessDevice>
```

Figure 3: accessGroupList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accessGroupList   xmlns:ns2="http://automatedMediaLibrary/">
   <accessGroup/> List of accessGroup objects, see Figure 4: accessGroup
</ns2:accessGroupList>
```

Figure 4: accessGroup

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:accessGroup   xmlns:ns2="http://automatedMediaLibrary/">
      <name>AG2</name>
      <hostList/> See Figure 86: hostList
      <accessDeviceList/> See Figure 1: accessDeviceList
</accessGroup>
```

Figure 5: activeVaultPolicyList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:activeVaultPolicyList  xmlns:ns2="http://automatedMediaLibrary/">
```

```
    <activeVaultPolicy/>    <!--  A list of activeVaultPolicy objects, see Figure 6: activeVaultPolicy -->
</ns2:activeVaultPolicyList>
```

Figure 6: activeVaultPolicy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:activeVaultPolicy   xmlns:ns2="http://automatedMediaLibrary/">
        <partitionName>LL1</partitionName>
        <!-- The partition will have either a vaultDefinedExport policy or a externalDefinedExport policy -->
        <vaultDefinedExport> <!-- Redirect exports to specific Active Vault -->
            <activeVaultName>AV Partition</activeVaultName>
            <mediaFilter>*00LTO5</mediaFilter>
        </vaultDefinedExport>
        <externalDefinedExport> <!-- Redirect exports to external application (StorNext) defined Active Vault
-->
            <externalApplicationServersName>The external application server configuration
name</externalApplicationServersName> <!-- See Figure 70: externalApplicationServers -->
        </externalDefinedExport>
    </activeVaultPolicy>
</ns2:activeVaultPolicyList>
```

Figure 7: autoExportPolicyList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:autoExportPolicyList  xmlns:ns2="http://automatedMediaLibrary/">
    <autoExportPolicy/> <!-- A list of autoExportPolict objects -->
</ns2:autoExportPolicyList>
```

Figure 8: autoExportPolicy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:autoExportPolicy xmlns:ns2="http://automatedMediaLibrary/">
    <partitionName>LL2</partitionName> <!-- The partition you want to apply the policy too -->
    <destinationAmpPartitionName>AMP</destinationAmpPartitionName> <!-- The AMP partition the
media will be moved too -->
</ns2:autoExportPolicy>
```

Figure 9: autoImportPolicyList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:autoImportPolicyList  xmlns:ns2="http://automatedMediaLibrary/">
    <autoImportPolicy/>    <!--  A list of autoImportPolicy objects -->
</ns2:autoImportPolicyList>
```

Figure 10: autoImportPolicy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:autoImportPolicy  xmlns:ns2="http://automatedMediaLibrary/">
    <partitionName>The name of the partition to apply the policy too</partitionName>
   <mediaBarcodeFilter>0000500-0000700</mediaBarcodeFilter> <!-- The media barcode filter range, the
following regex applies "([a-zA-Z0-9]{5,15}-[a-zA-Z0-9]{5,15};?\\s*)*" -->
</ns2:autoImportPolicy>
```

Figure 11: blade

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:blade xmlns:ns2="http://automatedMediaLibrary/">
  <coordinate/> See Figure 20: coordinate
  <name>name</name>
  <type>1</type> <!-- 1(FC IO Blade), 2(EEB), 3(MCB) -->
  <firmwareVersion>firmwareVersion</firmwareVersion>
  <serialNumber>serialNumber</serialNumber>
</ns2:blade>
```

Figure 12: cleanDriveTask

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:cleanDriveTask   xmlns:ns2="http://automatedMediaLibrary/">
  <serialNumber>The drives physical or logical serial number</serialNumber>
  <coordinate>The coordinate of the media to use for cleaning</barcode> <!-- see Figure 20: coordinate -->
</ns2:cleanDriveTask>
```

Figure 13: cleaningMediaList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:cleaningMediaList  xmlns:ns2="http://automatedMediaLibrary/">
  </cleaningMedia> <!-- A list of cleaningMedia objects -->
 </ns2:cleaningMediaList>
```

Figure 14: cleaningMedia

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:cleaningMedia  xmlns:ns2="http://automatedMediaLibrary/">
  </media> <!—see Figure 105: media -->
  <useCount>The number of times the media was used to clean a drive</useCount>
  <state>1</state> <!-- 0(Unknown), 1(Valid), 2(Invalid), 3(Expired) -->
 </ns2:cleaningMedia>
```

Figure 15: communicationCertificateList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:communicationCertificateList xmlns:ns2="http://automatedMediaLibrary/">
   </communicationCertificate> <!-- A list of communicationCertificate objects -->
</ns2:communicationCertificateList>
```

Figure 16: communicationCertificate

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:communicationCertificate xmlns:ns2="http://automatedMediaLibrary/">
   <keySize>2048</keySize> <!-- 1024, 2048 0r 4096 -->
   <digestAlgorithm>0</digestAlgorithm> <!-- 0(Unknown), 1(MD5), 2(SHA1), 3(SHA224), 4(SHA256),
5(SHA384), 6(SHA512), 7(MDC2), 8(RIPEMD160) -->
   </issuer> <!-- certificateInformation object -->
   </subject> <!-- certificateInformation object -->
   <type>Root</type> <!-- types are Root, Admin, Client, Identity or Unknown -->
   <validNotBefore>Apr 19 13:51:18 2011 GMT</validNotBefore>
   <validNotAfter>Apr 17 13:51:18 2021 GMT</validNotAfter>
</ns2:communicationCertificate>
```

Figure 17: componentList (ENUM)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<componentList xmlns:ns2="http://automatedMediaLibrary/">
    <component name="component name"> <!-- A list of one or more components -->
        <element name="component element name">
            <entry key="1" value="Connectivity"/> <!-- A list of one or more entry elements, which are key,
value pairs -->
        </element>
    </component>
 </componentList>
```

Figure 18: certificateInformation

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:certificateImformation xmlns:ns2="http://automatedMediaLibrary/">
   <countryCode>US</countryCode>
   <state>Colorado</state>
   <locality>Denver</locality>
   <organization>DVT</organization>
   <organizationalUnit>DVT</organizationalUnit>
   <commonName>Your Company Name</commonName>
   <emailAddress>E-mail address</emailAddress>
</ns2: certificateImformation >
```

Figure 19: contactInformation

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:contactInformation xmlns:ns2="http://automatedMediaLibrary/">
   <!-- each element can have a maximum of 256 characters -->
   <name>The contact name</name>
   <company>Company name</company>
   <phone>Contact phone number</phone>
   <email>Contact e-mail address</email>
   <description>A description of the library, i6k, i500 ..</description>
</ns2:contactInformation>
```

Figure 20: coordinate

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:coordinate xmlns:ns2="http://automatedMediaLibrary/">
  <frame>1</frame>
  <rack>0</rack>
  <section>0</section>
  <column>0</column>
  <row>0</row>
  <type>1</type> <!—0(N/A), 1(Robot), 2(Storage), 3(IE), 4(Drive), 5(XIE), 6(Cleaning) -->
</ns2:coordinate>
```

Figure 21: crossPartitionMovesList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:crossPartitionMovesList xmlns:ns2="http://automatedMediaLibrary/">
  <crossPartitionMoves/> A list of crossPartitionMoves objects
```

```
</ns2:crossPartitionMovesList>
```

Figure 22: crossPartitionMoves

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:crossPartitionMoves xmlns:ns2="http://automatedMediaLibrary/">
  <barcode>barcode</barcode>
  <sourcePartition>sourcePartition</sourcePartition>
  <sourceElementAddress>250</sourceElementAddress>
  <destinationPartition>destinationPartition</destinationPartition>
  <destinationElementAddress>4096</destinationElementAddress>
  <dateTime>2001-12-31 12:00:00</dateTime>
</ns2:crossPartitionMoves>
```

Figure 23: dateTime

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:dateTime  xmlns:ns2="http://automatedMediaLibrary/">
    <date>2012-08-14</date>
    <time>21:34:02</time>
    <datetime>2012-08-14 21:34:02 -0600</datetime>
    <timezone>America/Denver</timezone>
    <itime>1345001642141</itime> <!-- time in seconds since EPOCH -->
</ns2:dateTime>
```

Figure 24: detailedDriveActivityStatistics

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:detailedDriveActivityStatistics xmlns:ns2="http://automatedMediaLibrary/">
    <drive> <!-- One of these for each drive installed in the library -->
        <physicalSerialNumber>1210007896</physicalSerialNumber>
        <logicalSerialNumber>F001397049</logicalSerialNumber>
        <partition>LL1</partition> <!-- NULL if the drive does not belong to a partition -->
        <driveActivityStatisticsList>
            <driveActivityStatistics> <!-- 24 of these, see Figure 28: driveActivityStatistics -->
            </driveActivityStatistics>
        </driveActivityStatisticsList>
     <drive>
</ns2:detailedDriveActivityStatistics>
```

Figure 25: DNS

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:DNS xmlns:ns2="http://automatedMediaLibrary/">
  <ipAddress>10.20.88.2</ipAddress>
  <ipAddress>192.168.23.5</ipAddress>
</ns2:DNS>
```

Figure 26: driveList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveList  xmlns:ns2="http://automatedMediaLibrary/">
   <drive/> A list of drive objects, see Figure 27: drive
</ns2:driveList>
```

Figure 27: drive

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:drive xmlns:ns2="http://automatedMediaLibrary/">
  <physicalSerialNumber>physicalSerialNumber</physicalSerialNumber>
  <logicalSerialNumber>logicalSerialNumber</logicalSerialNumber>
  <address>address</address>              <!-- WWPN -->
  <coordinate /> See Figure 20: coordinate
  <elementAddress>2048</elementAddress> <!-- Logical element address -->
  <firmwareVersion>firmwareVersion</firmwareVersion>
  <type>
    <domainType>3</domainType> <!-- 3(LTO2), 4(LTO3), 5(LTO4), 6(LTO5), 7(LT06), 8(LTO7),
554(Mixed Types) -->
    <vendor>IBM or HP</vendor>
    <productId>Ultrium 6-SCSI</productId>
    <sledType>1</sledType>       <!—0(Unknown), 1(Standard), 2(EDLM) -->
    <interface>1</interface>       <!-- 0(Unknown), 1(SCSI), 2(Fibre), 3(SAS) -->
    <formFactor>FH or HH</formFactor>
  </type>
  <mode>0</mode>                  <!-- 1(Online), 2(Offline) -->
  <state>1</state>                <!-- 1(Varied On), 2(Varied Off), 3(Pending) -->
  <status>0</status> <!-- 0(Unknown), 1(Good), 2(Failed), 3(Degraded), 4(Not Installed), 5(Initializing) -->
  <owner>Owning partition or NULL</owner>
  <settings>
    <controlPath>
      <primary>true</primary>
      <type>1</type>                <!-- 1(None), 2(Standard/CP), 3(Basic/CPF), 4(Advanced/ACPF), 5(Multi
Control Path), 6(Advanced Control Path IBM) -->
      <subType>1</subType>       <!-- 1(None), 2(Active), 3(Standby) -->
    </controlPath>
    <dataPath>1</dataPath>       <!-- 1(None), 2(Active), 3(Advanced) -->
    <encryption>
      <method>0</method>         <!-- 0(None), 1(AME), 2(LME), 4(SME) -->
      <fips>true</fips>
    </encryption>
    <ipAddress>10.10.3.50</ipAddress>
    <bladeCoordinate/>   See Figure 20: coordinate
    <bladeAttachedType>0</bladeAttachedType>            <!-- 0(None), 1(FC IO Blade), 2(EEB) -->
    <license>7</license>            <!-- 7(EKM), 10(SM), 11(SNW) -->
  </settings>
  <barcode>media barcode if drive is loaded</barcode>
  <mediaHomeCoordinate>The slot coordinate where the media came from</ mediaHomeCoordinate> <!-
- If the is no media loaded in the drive this will be null -->
  <sledSerialNumber>sledSerialNumber</sledSerialNumber>
  <sledBootFirmwareVersion>sledBootFirmwareVersion</sledBootFirmwareVersion>
  <sledAppFirmwareVersion>sledAppFirmwareVersion</sledAppFirmwareVersion>
  <portCount>0</portCount>
</ns2:drive>
```

Figure 28: driveActivityStatistics

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveActivityStatisticsList xmlns:ns2="http://automatedMediaLibrary/">
    <driveActivityStatistics>
       <hourOfDay>16</hourOfDay> <!-- 23 hours ago -->
       <mbRead>0</mbRead>
```

```
    <mbWritten>5677</mbWritten>
    <mountCount>2</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>17</hourOfDay> <!-- 22 hours ago -->
    <mbRead>0</mbRead>
    <mbWritten>22000</mbWritten>
    <mountCount>4</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>18</hourOfDay> <!-- 21 hours ago -->
    <mbRead>44567</mbRead>
    <mbWritten>0</mbWritten>
    <mountCount>4</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>19</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>0</mbWritten>
    <mountCount>0</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>20</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>0</mbWritten>
    <mountCount>0</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>21</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>562209</mbWritten>
    <mountCount>5</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>22</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>3399</mbWritten>
    <mountCount>3</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>23</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>0</mbWritten>
    <mountCount>0</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>0</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>0</mbWritten>
    <mountCount>0</mountCount>
  </driveActivityStatistics>
  <driveActivityStatistics>
    <hourOfDay>1</hourOfDay>
    <mbRead>0</mbRead>
    <mbWritten>0</mbWritten>
    <mountCount>0</mountCount>
```

```
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>2</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>3</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>4</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>5</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>6</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>7</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>8</hourOfDay>
   <mbRead>344</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>2</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>9</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
   <hourOfDay>10</hourOfDay>
   <mbRead>0</mbRead>
   <mbWritten>0</mbWritten>
   <mountCount>0</mountCount>
</driveActivityStatistics>
<driveActivityStatistics>
```

```
      <hourOfDay>11</hourOfDay>
      <mbRead>0</mbRead>
      <mbWritten>0</mbWritten>
      <mountCount>0</mountCount>
   </driveActivityStatistics>
   <driveActivityStatistics>
      <hourOfDay>12</hourOfDay>
      <mbRead>0</mbRead>
      <mbWritten>0</mbWritten>
      <mountCount>0</mountCount>
   </driveActivityStatistics>
   <driveActivityStatistics>
      <hourOfDay>13</hourOfDay> <!-- 2 hours ago -->
      <mbRead>0</mbRead>
      <mbWritten>0</mbWritten>
      <mountCount>0</mountCount>
   </driveActivityStatistics>
   <driveActivityStatistics>
      <hourOfDay>14</hourOfDay> <!-- 1 hour ago -->
      <mbRead>0</mbRead>
      <mbWritten>0</mbWritten>
      <mountCount>0</mountCount>
   </driveActivityStatistics>
   <driveActivityStatistics>
      <hourOfDay>15</hourOfDay> <!-- Current hour of day -->
      <mbRead>23344</mbRead>
      <mbWritten>0</mbWritten>
      <mountCount>4</mountCount>
   </driveActivityStatistics>
</ns2:driveActivityStatisticsList>
```

Figure 29: driveCleaningPolicyList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:driveCleaningPolicyList xmlns:ns2="http://automatedMediaLibrary/">
  <driveCleaningPolicy/> <!-- A list of driveCleaningPolicy objects -->
</ns2:driveCleaningPolicyList>
```

Figure 30: driveCleaningPolicy

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:driveCleaningPolicy xmlns:ns2="http://automatedMediaLibrary/">
  <partition>partition name</partition>  <driveCleaning> <!—This option is only available for HP drives,
IBM drive do not support this feature -->
    <enabled>true</enabled> <!-- Determines whether cleaning is enable or disabled -->
    <motionTime>100</motionTime> <!-- 0, 100, 200, 400, 800 or 1000 -->
    <mountCount>0</mountCount> <!—0, 10 or 25 -->
  </driveCleaning>
  <libraryCleaning> <!-- When a library should perform periodic drive cleaning for a the named partition -->
    <hour>0</hour>
    <dayOfWeek>0</dayOfWeek>
    <period>1</period>
  </libraryCleaning>
  <relations/>
</ns2:driveCleaningPolicy>
```

Figure 31: driveLevelingPolicy

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveLevelingPolicy
   xmlns:ns2="http://automatedMediaLibrary/">
   <partition>LL1</partition>
   <type>2</type>
   <firmwareFile/> <!-- A list of firmwareFile objects, see Figure 79: firmwareFile -->
</ns2:driveLevelingPolicy>
```

Figure 32: driveLogList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveLogList xmlns:ns2=http://automatedMediaLibrary/>
  <driveLog/> A list of driveLog objects
</ns2:driveLogList>
```

Figure 33: driveLog

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveLog xmlns:ns2=http://automatedMediaLibrary/>
  <name>The log file name</name>
  <created>The date and time the log file was created</created>
  <size>The log file size, in bytes</size>
</ns2:driveLog>
```

Figure 34: drivePortList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:drivePortList xmlns:ns2=http://automatedMediaLibrary/>
  <drivePorts/> A list of drivePorts objects, see Figure 35: drivePorts
</ns2:drivePortList>
```

Figure 35: drivePorts

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:drivePorts xmlns:ns2="http://automatedMediaLibrary/">
  <serialNumber>serialNumber</serialNumber>  <!-- The drives logical serial number -->
  <physicalSerialNumber>serial number</physicalSerialNumber>  <!-- The drive's physical serial number -
->
  <ports/> A list of port objects, see Figure 130: port
</ns2:drivePorts>
```

Figure 36: driveSerialNumberList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveSerialNumberList xmlns:ns2="http://automatedMediaLibrary/">
    <serialNumber>F00139603D</serialNumber> <!-- one or more serialNumber elements -->
</ns2:driveSerialNumberList>
```

Figure 37: driveUtilizationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:driveUtilizationList xmlns:ns2=http://automatedMediaLibrary/>
  <driveUtilization/> A list of driveUtilization objects
```

```
</ns2:driveUtilizationList>
```

Figure 38: driveUtilization

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:driveUtilization xmlns:ns2="http://automatedMediaLibrary/">
  <coordinate>
    <frame>1</frame>
    <rack>0</rack>
    <section>0</section>
    <column>0</column>
    <row>0</row>
    <type>4</type>
    <relations/>
  </coordinate>
  <driveSerialNumber>the drive  serial number</driveSerialNumber>
  <partition>partition name</partition>
  <mountTime>2001-12-31 12:00:00</mountTime>
  <unmountTime>2001-12-31 12:02:00</unmountTime>
  <mbRead>0</mbRead>
  <mbWrite>0</mbWrite>
  <barcode>barcode</barcode>
</ns2:driveUtilization>
```

Figure 39: edlmMediaList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmMediaList xmlns:ns2="http://automatedMediaLibrary/">
  </edlmMedia> <!-- A list of edlmMedia objects -->
</ns2:edlmMediaList>
```

Figure 40: edlmMedia

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmMedia xmlns:ns2="http://automatedMediaLibrary/">
  <barcode>ABC123000L6</barcode>
  <resultId>19</resultId> <!-- The id of the edlmMediaResult object -->
  <coordinate>
    <frame>1</frame>
    <rack>0</rack>
    <section>0</section>
    <column>0</column>
    <row>0</row>
    <type>1</type>
  </coordinate>
  <owner>owner</owner> <!-- What partition owns this media -->
  <testType>0</testType> <!-- 0(None), 1(Quick Scan), 2(Normal Scan), 3(Full Scan) -->
  <testPriority>0</testPriority> <!-- 0(Immediate), 1(High), 2(Medium), 3(Low) -->
  <testState>1</testState> <!-- 0(Unknown), 1(Pending), 2(In Progress), 3(Complete), 4(Stopped),
5(Paused), 6(Resume) -->
  <testResult>3</testResult> <!-- 0(Not Complete), 1(Good), 2(Unsupported), 3(Suspect), 4(Failed) -->
  <lastTested>2001-12-31T12:00:00</lastTested> <!-- When was media last tested -->
  <supported>true</supported> <!-- Is media supported for EDLM Scan tests -->
</ns2:edlmMedia>
```

Figure 41: edlmMediaResultList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:edlmMediaResultList xmlns:ns2="http://automatedMediaLibrary/">
    </edlmMediaResult> A list of edlmMediaResult objects
</ns2:edlmMediaResultList>
```

Figure 42: edlmMediaResult

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmMediaResult xmlns:ns2="http://automatedMediaLibrary/">
  <id>0</id>
  <sessionId>0</sessionId>
  <barcode>barcode</barcode>
  <driveSerialNumber>driveSerialNumber</driveSerialNumber>
  <mediaSerialNumber>1070748083</mediaSerialNumber>
  <testDate>2001-12-31 12:00:00</testDate>
  <startDate>2001-12-31 11:48:05</startDate>
  <testState>0</testState> <!-- 0(Unknown), 1(Pending), 2(In Progress), 3(Complete), 4(Stopped),
5(Paused), 6(Resume) -->
  <testType>0</testType> <!-- 0(None), 1(Quick Scan), 2(Normal Scan), 3(Full Scan) -->
  <testResult>0</status> <!-- 0(Not Complete), 1(Good), 2(Unsupported), 3(Suspect), 4(Failed) -->
  <staticTestStatus>0</staticTestStatus> <!-- 512(Test Completed), 513(Test Paused), 514(Test Pending),
515(Test Not Run),  516(Test In Progress) -->
  <dynamicTestStatus>0</dynamicTestStatus> <!—1024(Test Completed),  1025(Test Paused), 1026(Test
Pending), 1027(Test Not Run), 1028(Test In Progress), 1029(Test Not Configured) -->
  <staticTestErrorCode>0</staticTestErrorCode> <!—768(Good), 769(N/A), 770(Failed to get CM data),
771(CM Hardware Failure), 772(99% EOL based on thread count), 773(99% EOL based on number of
writes), 774(Uncorrected errors), 775(Load failure), 776(Unload failure) -->
  <dynamicTestErrorCode>0</dynamicTestErrorCode> <!—1280(Good), 1281(N/A), 1282(Failed to
communicate to IO Blade), 1283Failed to receive scan data), 1284(Unexpected EOD, Possible corrupt
CM), 1285(Unformatted tape), 1286(Failed to read tape data),  1287(Un-recovered read errors on tape),
1289(Corrupt data format), 1296(Tape experienced a mechanical error), 1297(Tape performance is
severely degraded), 1298(Unable to load tape), 1299(Unable to unload tape), 1300(Tape is a cleaning
cartridge),  1301(Cartridge memory fault encountered), 1302(Unknown media type detected), 1303(Scan
was aborted), 1304(Drive reports no media present), 1305(Media is encrypted), 1312(Media is blank),
1313(Block size exceeds maximum) -->
</ns2:edlmResult>
```

Figure 43: edlmPolicyList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmPolicyList xmlns:ns2="http://automatedMediaLibrary/">
   <edlmPolicy/> A list of edlmPolicy objects
</ns2:edlmPolicyList>
```

Figure 44: edlmPolicy

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmPolicy xmlns:ns2="http://automatedMediaLibrary/">
  <partitionName>partitionName</partitionName>
  <tapeAlert>
    <scanType>0</scanType> <!-- 0(None), 1(Quick), 2(Normal) and 3(Full) -->
    <count>0</count> <!-- The threshold count on when to do the scan -->
  </tapeAlert>
```

```
  <timeInterval> <!-- The interval is in days -->
    <quickScan>0</quickScan>
    <normalScan>0</normalScan>
    <fullScan>0</fullScan>
  </timeInterval>
  <onImport>0</onImport> <!—Perform scan when media is imported 0(No Scan), 1(Quick), 2(Normal)
and 3(Full) -->
  <scanPriotity>0</scanPriotity> <!-- 0(Immediate), 1(Low), 2(Medium), 3(High) -->
  <concurrentScans>0</concurrentScans> <!-- The maximum number of concurrent scan allow (depends
on the number of EDLM dries configured), 0 for No Limit and then 1 through N where N is the number of
EDLM drives installed -->
  <continueOnError>true</continueOnError>
  <disableRasTicketGeneration>true</disableRasTicketGeneration>
  <externalPolicies>
    <externalApplicationServersName>The StorNext server name</externalApplicationServersName>
    <mediaCopyPolicy>0</mediaCopyPolicy> <!-- 0(Disabled), 1(Bad Media), 2(Suspect Media), 3(Bad or
Suspect Media) -->
    <suspectCountScanType>0</suspectCountScanType> <!-- 0(None), 1(Quick), 2(Normal), 3(Full) -->
  </externalPolicies>
  <relations/>
</ns2:edlmPolicy>
```

Figure 45: edlmSessionList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmSessionList xmlns:ns2="http://automatedMediaLibrary/">
  </edlmSession> <!-- A list of edlmSession objects -->
</ns2:edlmSessionList>
```

Figure 46: edlmSession

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:edlmSession xmlns:ns2="http://automatedMediaLibrary/">
  <id>0</id>
  <startDate>2001-12-31 12:00:00</startDate>
  <endDate>2001-12-31 12:00:00</endDate>
  <state>0</state> <!-- 0(Unknown), 1(Complete), 2(Pending), 4(Paused), 8(In Progress), 16(Stopped) -->
  <goodCount>0</goodCount> <!-- The number of scans in the session that completed with good status -->
  <suspectCount>0</suspectCount> <!-- The number of scans in the session that completed with suspect
status -->
  <badCount>0</badCount> <!-- The number of scans in the session that completed with bad status -->
  <notCompleteCount>0</notCompleteCount> <!-- The number of scans in the session that did not
complete -->
  <unSupportedCount>0</unSupportedCount> <!-- The number of scans in the session that were not
supported -->
  <continueOnError>true</continueOnError> <!—Was the continue on error flag set for this session -->
</ns2:edlmSession>
```

Figure 47: ekmMediaStatusList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ekmMediaStatusList xmlns:ns2="http://automatedMediaLibrary/">
  < ekmMediaStatus/> A list of ekmMediaStatus objects
</ns2:ekmMediaStatusList>
```

Figure 48: ekmMediaStatus

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ekmMediaStatus xmlns:ns2="http://automatedMediaLibrary/">
  <driveSerialNumber>driveSerialNumber</driveSerialNumber>
  <barcode>barcode</barcode>
  <mediaSerialNumber>mediaSerialNumber</mediaSerialNumber>
  <partitionName>partitionName</partitionName>
  <encrypted>true</encrypted>
  <mountTime>2001-12-31T12:00:00</mountTime>
  <unmountTime>2001-12-31T12:00:00</unmountTime>
</ns2:ekmMediaStatus>
```

Figure 49: ekmPartitionActivityList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ekmPartitionActivityList xmlns:ns2="http://automatedMediaLibrary/">
  < ekmPartitionActivity/> A list of ekmPartitionActivity objects
</ns2:ekmPartitionActivityList>
```

Figure 50: ekmPartitionActivity

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ekmPartitionActivity xmlns:ns2="http://automatedMediaLibrary/">
  <partition>partition</partition>
  <method>method</method>
  <user>user</user>
  <loginFrom>loginFrom</loginFrom>
  <dateTime>2001-12-31T12:00:00</dateTime>
</ns2:ekmPartitionActivity>
```

Figure 51: ekmServersList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ekmServersList   xmlns:ns2="http://automatedMediaLibrary/">
    <ekmServers/> <!-- A list of ekmServers objects -->
</ns2:ekmServersList>
```

Figure 52: ekmServers

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ekmServers xmlns:ns2="http://automatedMediaLibrary/">
        </ekmServerType> <!-- 2(RKM), 4(KMIP), 8(QEKM), 16(SKM), 32(TKLM) -->
        <!-- Note: RKM is no longer supported -->
        <server>
            <hostName>10.20.9.18</hostName>
            <port>3801</port>
            <status>Active</status>
        </server>
        <server>
            <hostName>10.20.9.6</hostName>
            <port>3801</port>
            <status>Standby</status>
        </server>
        <ekmPathDiagnosticsInterval>0-60</ekmPathDiagnosticsInterval> <!-- 0-60 minutes, where 0
```

```
means disabled -->
        <ssl>false</ssl>
</ns2:ekmServers>
```

Figure 53: ekmServerTestResultList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ekmServerTestResultList xmlns:ns2="http://automatedMediaLibrary/">
  </ekmServerTestResult> <!-- A list of ekmServerTestResult objects -->
</ns2:ekmServerTestResultList>
```

Figure 54: ekmServerTestResult

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ekmServerTestResult xmlns:ns2="http://automatedMediaLibrary/">
  <server>server</server> <!-- The EKM server name/IP -->
  <test> <!-- The list of test objects -->
    <type>1</type> <!-- 1(Ping Test), 2(Path Test), 3(Configure Test) -->
    <result>1</result> <!-- 1(Passed), 2(Failed) -->
  </test>
</ns2:ekmServerTestResult>
```

Figure 55: elementList (ENUM)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<elementList   xmlns:ns2="http://automatedMediaLibrary/">
    <component name="component name">
        <element name="element name"> <!-- One or more elements -->
            <entry key="1" value="Connectivity"/> <!-- One or more key value entry pairs -->
        </element>
    </component>
</elementList>
```

Figure 56: elementList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:elementList  xmlns:ns2="http://automatedMediaLibrary/">
        <element/> <!-- A list of element objects -->
</ns2:elementList>
```

Figure 57: element

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:element  xmlns:ns2="http://automatedMediaLibrary/">
        <coordinate/> <!-- See Figure 20: coordinate -->
        <address>256</address> <!-- The logical SCSI element address -->
        <status>2</status> <!-- 1 (Not Installed), 2 (Accessible), 3 (Not Accessible) -->
        <barcode></barcode>
        <owner>Test Partition</owner>
        <configuredType>0</configuredType>
</ns2:element>
```

Figure 58: email

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

```
<ns2:email  xmlns:ns2="http://automatedMediaLibrary/">
   <!-- A list of recipients -->
   <recipient>john.doe@company.com</recipient>
   <recipient>jane.doe@company.com</recipient>
   <comment>Needs your urgent attention</comment>
   </reportCriteria> <!-- Figure 143: reportCriteria -->
</ns2:email>
```

Figure 59: emailServer

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:emailServer    xmlns:ns2="http://automatedMediaLibrary/">
   <server>10.20.169.2</server>
   <accountName>john.doe</accountName>
   <accountPassword>secret_password</accountPassword>
   <senderEmailAddress>dvt8@quantum.com</senderEmailAddress>
   <authorize>true</authorize>
   <testEmailAddress>john.doe@company.com</testEmailAddress>
</ns2:emailServer>
```

Figure 60: emailRecipientList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:emailRecipientList xmlns:ns2="http://automatedMediaLibrary/">
   <emailRecipient>
      <id>1</id>  <!-- Automatically generated by the library -->
      <address>john.doe@company.com</address> <!-- e-mail address -->
   </emailRecipient>
</ns2:emailRecipientList>
```

Figure 61: emailRecipient

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:emailRecipient xmlns:ns2="http://automatedMediaLibrary/">
   <id>1</id>  <!-- Automatically generated by the library -->
   <address>john.doe@company.com</address> <!-- e-mail address -->
</ns2:emailRecipient>
```

Figure 62: emmc

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:emmc xmlns:ns2="http://automatedMediaLibrary/">
  <sectorUsage>Good</sectorUsage> <!-- Good, Warning, Alert and Degraded -->
  <spareBlocksUsage>Good</spareBlocksUsage> <!-- Good, Warning and Alert -->
</ns2:emmc>
```

Figure 63: entryList (ENUM)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<entryList   xmlns:ns2="http://automatedMediaLibrary/">
   <component name="component name">
      <element name="element name">
         <entry key="1" value="SCSI"/> <!-- One or more key value entry pairs -->
      </element>
   </component>
</entryList>
```

Figure 64: ethernetExpansionBladeList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ethernetExpansionBladeList xmlns:ns2="http://automatedMediaLibrary/">
  <ethernetExpansionBlade/> A list of ethernetExpansionBlade objects, see Figure 65:
ethernetExpansionBlade
</ns2:ethernetExpansionBladeList>
```

Figure 65: ethernetExpansionBlade

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ethernetExpansionBlade xmlns:ns2="http://automatedMediaLibrary/">
  <blade/> See Figure 11: blade
  <ethernetExpansionBladePort/> A list of 6 ethernetExpansionBladePort objects, see Figure 66:
ethernetExpansionBladePort
</ns2:ethernetExpansionBlade>
```

Figure 66: ethernetExpansionBladePort

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ethernetExpansionBladePort xmlns:ns2="http://automatedMediaLibrary/">
  <number>0</number>
  <linkStatus>linkStatus</linkStatus>
  <speed>0</speed>
  <duplex>duplex</duplex>
</ns2:ethernetExpansionBladePort>
```

Figure 67: externalApplicationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:externalApplicationList   xmlns:ns2="http://automatedMediaLibrary/">
     </externalApplication> A list of externalApplication objects.
</ns2:externalApplicationList>
```

Figure 68: externalApplication

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:externalApplication   xmlns:ns2="http://automatedMediaLibrary/">
     <name>snapi-2.0.1</name>
     <version>110i.EE002</version>
     <description>StorNext Plugin (3.5 to 4.1)</description>
</ns2:externalApplication>
```

Figure 69: externalApplicationServersList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:externalApplicationServersList   xmlns:ns2="http://automatedMediaLibrary/">
   </externalApplicationServers> A list of externalApplicationServers objects
</ns2:externalApplicationServersList>
```

Figure 70: externalApplicationServers

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:externalApplicationServers   xmlns:ns2="http://automatedMediaLibrary/">
      <name>StorNext</name>
      <server>
         <name>10.20.169.88</name>
         <port>61776</port>
      </server>
      <server>
         <name>10.20.9.18</name>
         <port>61776</port>
      </server>
      <externalApplicationName>snappy-2.0.1</externalApplicationName>
</ns2:externalApplicationServersList>
```

Figure 71: fcBladeList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:fcBladeList xmlns:ns2=http://automatedMediaLibrary/>
   <fcBlade/> A list of fcBlade objects, see Figure 72: fcBlade
</ns2:fcBladeList>
```

Figure 72: fcBlade

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:fcBlade xmlns:ns2=http://automatedMediaLibrary/>
  <blade/> See Figure 11: blade
  <wwnn>wwnn</wwnn>
  <status>1</status> <!-- 1(Unknown), 2(OK), 3(Warning), 4(Failed) -->
  <hostPortFailover>true</hostPortFailover>
  <fcBladePort/> A list of 6 fcBladePort objects, see Figure 73: fcBladePort
  <host/> A list of host objects, see Figure 87: host
</ns2:fcBlade>
```

Figure 73: fcBladePort

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:fcBladePort xmlns:ns2="http://automatedMediaLibrary/">
  <wwpn>wwpn</wwpn>
  <number>0</number>
  <loopId>0</loopId>
  <!-- See blade.xsd for the following element values -->
  <mode>0</mode>
  <topology>0</topology>
  <speed>0</speed>
  <frameSize>512</frameSize>
  <status>0</status>
</ns2:fcBladePort>
```

Figure 74: firmwareList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareList   xmlns:ns2="http://automatedMediaLibrary/">
    <lastInstallDate>2014-06-02 09:57:50 -0600</lastInstallDate>
```

```
   <firmware/> <!-- A list of firmware objects -->
</ns2:firmwareList>
```

Figure 75: firmware

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmware  xmlns:ns2="http://automatedMediaLibrary/">
       <component>Current</component> <!-- Values are 'Current' (i6k only), Rollback' and 'Uploaded' -->
       <version>665H.TS07401</version> <!-- The firmware/software version -->
       <signingStatus>Signed by Production Certificate</signingStatus>
</ns2:firmware>
```

Figure 76: firmwareFileList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:firmwareFileList xmlns:ns2="http://automatedMediaLibrary/">
  <firmwareFile/> A list of firmwareFile objects, see Figure 79: firmwareFile
</ns2:firmwareFileList>
```

Figure 77: firmwareUpdateList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareUpdateList   xmlns:ns2="http://automatedMediaLibrary/">
   </firmwareUpdate> A list of firmwareUpdate objects.
</ns2:firmwareUpdateList>
```

Figure 78: firmwareUpdate

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareUpdate   xmlns:ns2="http://automatedMediaLibrary/">
       <serialNumber>HU19487U51</serialNumber>
       </firmwareFile> A firmwareFile object
</ns2:firmwareUpdate>
```

Figure 79: firmwareFile

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:firmwareFile xmlns:ns2="http://automatedMediaLibrary/">
  <name>name</name>
  <version>version</version>
  <vendor>vendor</vendor> <!-- HP or IBM -->
  <type>type</type> <!-- LTO2, LTO3, LTO4, LTO5, LTO6 or LTO7 -->
  <formFactor>HH or FH</formFactor> <!-- HH (Half Height) or FH (Full Height) -->
  <interface>Fibre</interface> <!-- Fibre or SAS -->
</ns2:firmwareFile>
```

Figure 80: firmwareStatusList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareStatusList  xmlns:ns2="http://automatedMediaLibrary/">
   <updateState>2</updateState> <!-- 0(None), 1(Pending), 2(In Progress), 3(Aborting), 4(Aborted),
5(Success), 6(Failure) -->
   <firmwareStatus/> A list of firmwareStatus objects, see Figure 83: firmwareStatus
</ns2:firmwareStatusList>
```

Figure 81: frameList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:frameList  xmlns:ns2="http://automatedMediaLibrary/">
    <frame/> A list of frame objects
</ns2:frameList>
```

Figure 82: frame

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:frame xmlns:ns2="http://automatedMediaLibrary/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://automatedMediaLibrary/ ../xsd/system.xsd ">
  <id>0</id>
  <type>1</type>
  <serialNumber>serialNumber</serialNumber>
  <HDDM>true</HDDM>
  <HDDMLocalMode>true</HDDMLocalMode>
  <rack1>
    <driveCount>0</driveCount>
    <usedDriveCount>0</usedDriveCount>
    <storageSegmentCount>0</storageSegmentCount>
    <usedStorageSegmentCount>0</usedStorageSegmentCount>
    <usedXieSegmentCount>0</usedXieSegmentCount>
    <iobCount>0</iobCount>
    <cmbCount>0</cmbCount>
    <eebCount>0</eebCount>
  </rack1>
  <rack2>
    <storageSegmentCount>0</storageSegmentCount>
    <usedStorageSegmentCount>0</usedStorageSegmentCount>
    <usedXieSegmentCount>0</usedXieSegmentCount>
    <ie>0</ie>
    <ieSegmentCount>0</ieSegmentCount>
    <usedIeSegmentCount>0</usedIeSegmentCount>
  </rack2>
  <relations/>
</ns2:frame>
```

Figure 83: firmwareStatus

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:firmwareStatus  xmlns:ns2="http://automatedMediaLibrary/">
     <component>Drive SN: HU19487U51</component>
     <status>In Progress</status>
</ns2:firmwareStatus>
```

Figure 84: heartbeatNotificationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:heartbeatNotificationList xmlns:ns2="http://automatedMediaLibrary/">
    <heartbeatNotification/> A list of heartbeatNotification objects
</ns2:heartbeatNotificationList>
```

Figure 85: heartbeatNotification

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:heartbeatNotification xmlns:ns2="http://automatedMediaLibrary/">
        <id>1</id>
        <interval>60</interval> <!-- Interval in minutes -->
        <emailAddress>john.doe@company.com</emailAddress> <!-- Who to send the notification too. -->
</ns2:heartbeatNotification>
```

Figure 86: hostList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:hostList  xmlns:ns2="http://automatedMediaLibrary/">
    <host/> A list of host objects, see Figure 87: host
</ns2:hostList>
```

Figure 87: host

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:host  xmlns:ns2="http://automatedMediaLibrary/">
        <name>Test</name>
        <type>2</type>                  <!-- See access.xsd -->
        <mode>0</mode>                  <!--0(Offline), 1(Online) -->
        <WWPN>12345678:ABCDABCD</WWPN>
</ns2:host>
```

Figure 88: ieStationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ieStationList xmlns:ns2="http://automatedMediaLibrary/">
  <ieStation/> A list of ieStation objects, see Figure 89: ieStation
</ns2:ieStation>
```

Figure 89: ieStation

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ieStation xmlns:ns2="http://automatedMediaLibrary/">
  <number>0</number>
  <coordinate/> See Figure 20: coordinate
  <slotCount>0</slotCount>
  <opened>false</opened>
  <status>1</status>                <!-- 1(locked), 2(unlocked) -->
  <mode>0</mode>                    <!-- 1(Online), 2(Offline) -->
  <state>1</state>                  <!-- 1(Varied On), 2(Varied Off) -->
</ns2:ieStation>
```

Figure 90: internalNetwork

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:internalNetwork xmlns:ns2="http://automatedMediaLibrary/">
  <current>192.19.240.0</current>
  <options>10.247.240.0</options> <!-- One or more options elements -->
</ns2:internalNetwork>
```

Figure 91: inventoryTask

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:inventoryTask   xmlns:ns2="http://automatedMediaLibrary/">
  <partitionName>LL1</ partitionName> <!-- The partition name or NULL for physical library inventory -->
  <offline>true</offline> <!-- Take the partition/physical library offline while doing the inventory. You should
almost always set this to true -->
  <startElement>2048</startElement>
  <elementCount>2058</elementCount>
</ns2:inventoryTask>
```

Figure 92: ldap

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ldap
    xmlns:ns2="http://automatedMediaLibrary/">
    <enabled>true</enabled>
    <primaryServer>10.20.170.3</primaryServer>
    <alternateServer>10.20.170.4</alternateServer>
    <serverPort>389</serverPort> <!-- 389 (Standard port), 636 (Secure port) -->
    <secureMode>1</secureMode> <!-- 1(LDAP standard), 2(LDAPS, TLS connection), 3(StartTLS, start a
TLS session within an already established LDAP connection -->
    <searchUser>cn=administrator,cn=Users,dc=hyjal,dc=hw,dc=quantum,dc=com</searchUser>
    <searchUserPassword></searchUserPassword>
    <usersContext>cn=Users,dc=hyjal,dc=hw,dc=quantum,dc=com</usersContext>
    <groupContext>cn=Users,dc=hyjal,dc=hw,dc=quantum,dc=com</groupContext>
<libraryAccessGroupsUser>cn=people,cn=Users,dc=hyjal,dc=hw,dc=quantum,dc=com</libraryAccessGr
oupsUser>
<libraryAccessGroupsAdmin>cn=admins,cn=Users,dc=hyjal,dc=hw,dc=quantum,dc=com</libraryAccess
GroupsAdmin>
<!-- The following are used to support ldap-kerberos and will only be supported on non-Scalari6k products
-->
<realm>string</realm>
<keyDistributionCenter>string</keyDistributionCenter>
<domainMapping>string</domainMapping>
</ns2:ldap>
```

Figure 93: ldapTest

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ldapTest  xmlns:ns2="http://automatedMediaLibrary/">
    <user>JIM</user> <!-- The LDAP user -->
    <password>********</password> <!-- The LDAP user password -->
    </ldap> <!-- The ldap object to test see Figure 92: ldap -->
   </ns2:ldapTest>
```

Figure 94: libraryControlBladeList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:libraryControlBladeList xmlns:ns2=http://automatedMediaLibrary/>
   <libraryControlBlade/> A list of libraryControlBlade objects
</ns2:libraryControlBladeList>
```

Figure 95: libraryControlBlade

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2: libraryControlBlade xmlns:ns2=http://automatedMediaLibrary/>
  <blade/> See Figure 11: blade
  <fcBladePort/> A list of fcBladePort objects, see Figure 73: fcBladePort
</ns2: libraryControlBlade>
```

Figure 96: libraryStatus

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:libraryStatus   xmlns:ns2="http://automatedMediaLibrary/">
   <state>1</state> <!—1(Ready), 2(Not Ready), 3(Becoming Ready) -->
   <mode>1</mode> <!-- 1(Online) 2(Offline) -->
   <ras>
      <status> <!-- See Figure 132: RASGroupStatus -->
         <group>0</group>
         <status>4</status>
      </status>
      <openedTickets>5</openedTickets>
   </ras>
   <partition> <!-- 0 or n partition entries -->
      <name>AV</name>
      <mode>1</mode> <!-- 1(Online) 2(Offline) -->
      <type>1</type> <!-- 1(Standard), 2(EDLM), 3(AMP), 4(Active Vault), 5(CVTL) -->
   </ partition >
   <drive> <!-- 0 or n drive entries -->
      <logicalSerialNumber>drive logical serial number</ logicalSerialNumber>
      <mode>1</mode> <!-- 1(Online) 2(Offline) -->
      <state>2</state> <!-- 1(Varied On), 2(Varied Off) and 3(Pending/Initializing) -->
   </drive>
   <robot>  <!-- 0 or n robot entries -->
      <serialNumber>The robot serial number</serialNumber>
      <location>Right or Left robot</location> <!-- Left or Right -->
      <status>The robot status</status> <!—0(Unknown), 1(Good), 2(Not initialized), 3(Initializing),
4(Failed), 5(N/A)
      <state>2</state> <!-- 1(Varied On), 2(Varied Off) and 3(Pending/Initializing) -->
   </robot>
   <tower> <!-- 0 or n tower entries -->
      <frameNumber>The frame number the tower is in</frameNumber>
      <status>The robot status</status> <!—0(Unknown), 1(Not Present), 2(Failed), 3(Not Ready),
4(Initializing), 5(Ready)
      <state>2</state> <!-- 1(Varied On), 2(Varied Off) and 3(Pending/Initializing) -->
      <mode>1</mode> <!-- 1(Online) 2(Offline) -->
   </tower>
   <systemStatus>0</systemStatus> <!-- 1(Good), 2(Degraded), 3(Failed) -->
   <saveConfigurationRequired>true</saveConfigurationRequired> <!-- Was there a library configuration
change made that requires a new rescue image the be generated. -->
   <time>2014-06-27 11:00:15 -0600</time>
   <timeZoneOffset> Returns the offset, measured in minutes, for the local time zone relative to
UTC</timeZoneOffset>
      <timeSinceEPOC>1410958379212</timeSinceEPOC> <!-- Returns the number of milliseconds
elapsed since January 1st 1970 -->
    <snmpStarted>true</snmpStarted> <!-- This element is used on the i6k to indicate if the SNMP
daemon is running when the LMC server starts up. It should always be true, but we have seen cases in
the past where the daemon does not start up before LMC starts.-->
```

```
</ns2:libraryStatus>
```

Figure 97: licenseList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:licenseList  xmlns:ns2="http://automatedMediaLibrary/">
   <license/> <!-- A list of license objects. -->
</ns2:licenseList>
```

Figure 98: license

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:license  xmlns:ns2="http://automatedMediaLibrary/">
      <feature>The name of the feature</feature>
      <quantity>1</quantity> <!-- Some licenses are controlled by a number, for instance COD, the rest
will be 1 0r 0 is not installed -->
      <description>6 Drives</description>
      <expiration>Permanent</expiration> <!-- for i6k it will be 'Permanent' or 'Month Day, Year' example,
Jan 12, 2015 and for Quatro YYYY-MM-DD -->
      <installed>true</installed> <!-- Is the license installed -->
      <type> License type</type> <!-- 1(Automated Media Pool), 2(Advanced Reporting), 3(Active Vault),
4(Capacity On Demand), 5(Control Path Failover), 6(Extended Data Lifecycle Management), 7(Encryption
Key Management), 8(Partition), 9(Partition Utilization), 10(Secure Manager), 11(Path Failover/Native
Storage Networking), 12(LTFS) -->
      <usedQuantity>2</usedQuantity> <!-- This element only applies to the following license type:
Capacity on Demand, Partition, Encryption Key Management and Path Failover/Native Storage
Networking -->
   </license>
</ns2:license>
```

Figure 99: loginActivityList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2: loginActivityList xmlns:ns2="http://automatedMediaLibrary/">
  < loginActivity /> A list of loginActivity objects
</ns2: loginActivity List>
```

Figure 100: loginActivity

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:loginActivity xmlns:ns2="http://automatedMediaLibrary/">
      <user>admin</user>
      <role>0</role> <!-- 0(Admin User), 1(Standard User), 2(Service User), 3(Guest User) -->
      <sessionId>1</sessionId>
      <command>Login</command> <!-- The command that was executed login, logout, create partition,
etc.. -->
      <loginFrom>10.20.9.68</loginFrom>
      <time>2014-06-26 15:35:31 -0600</time>
      <description>More details of the command that was run</description>
</ns2:loginActivity>
```

Figure 101: logNames

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

```
<ns2:logNames xmlns:ns2="http://automatedMediaLibrary/">
  <name>name</name> <!-- A list of log names -->
</ns2:logNames>
```

Figure 102: logTableInfoList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:logTableInfoList xmlns:ns2="http://automatedMediaLibrary/">
    <logTableInfo/> <!-- A list of logTableInfo objects -->
</ns2:logTableInfoList>
```

Figure 103: logTableInfo

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:logTableInfo xmlns:ns2="http://automatedMediaLibrary/">
    <type>1</type> <!-- 0(N/A), 1(VT), 2(SCSI), 3(GET/PUT Statistics) -->
    <licensed>true</licensed>
    <format>TXT</format> <!-- N/A, TXT, CSV -->
</ns2:logTableInfo>
```

Figure 104: mediaList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:mediaList xmlns:ns2="http://automatedMediaLibrary/">
  <media/> A list of media objects, see Figure 105: media
</ns2:mediaList>
```

Figure 105: media

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:media xmlns:ns2="http://automatedMediaLibrary/">
  <barcode>barcode</barcode>
  <mediaType>0</mediaType>   <!—2(LTO1), 3(LTO2), 4(LTO3), 5(LTO4), 6(LTO5), 7(LTO6), 8(LTO7),
20(Cleaning) -->
  <cartridgeType>0</cartridgeType>       <!-- 0(Data), 1(Cleaning), 2(FUP), 3(Diagnostic) This element is
not supported on the i6k -->
  <currentOwner>N/A</currentOwner>
  <previousOwner>N/A</previousOwner>
  <encryption>Unknown</encryption>     <!-- Unknown, Encrypted, Not Encrypted -->
  <coordinate/> See Figure 20: coordinate
  <elementAddress>4096</elementAddress> <!-- The SCSI logical element address of where the media
is located -->
</ns2:media>
```

Figure 106: mediaSecurityEventList

```
<?xml version='1.0' encoding='UTF-8'?>
<ns2:mediaSecurityEventList xmlns:ns2="http://automatedMediaLibrary/">
   < mediaSecurityEvent/> A list of mediaSecurityEvent objects
</ns2:mediaSecurityEventList>
```

Figure 107: mediaSecurityEvent

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

```
<ns2:mediaSecurityEvent xmlns:ns2="http://automatedMediaLibrary/">
   <date>2001-12-31 12:00:00</date>
   <barcode>barcode</barcode>
   <coordinate/> <!-- See Figure 20: coordinate -->
   <removalExpected>true</removalExpected> <!-- Was the media removal expected or unexpected -->
</ns2:mediaSecurityEvent>
```

Figure 108: mediaSecurityPolicy

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:mediaSecurityPolicy xmlns:ns2="http://automatedMediaLibrary/">
  <unexpectedRemovalAfterReboot>true</unexpectedRemovalAfterReboot>
  <unexpectedRemovalDuringLibraryOperation>true</unexpectedRemovalDuringLibraryOperation>
  <expectedRemovalFromIe>true</expectedRemovalFromIe>
  <unexpectedRemovalFromIe>true</unexpectedRemovalFromIe>
</ns2:mediaSecurityPolicy>
```

Figure 109: mediaUsageList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:mediaUsageList xmlns:ns2="http://automatedMediaLibrary/">
  <mediaUsage/> A list of mediaUsage objects
</ns2:mediaUsageList>
```

Figure 110: mediaUsage

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:mediaUsage xmlns:ns2="http://automatedMediaLibrary/">
  <updated>2001-12-31T12:00:00</updated>
  <barcode>barcode</barcode>
  <serialNumber>Media serial number</serialNumber>
  <manufacturer>manufacturer</manufacturer>
  <type>type</type>
  <manufacturerDate>2001-12-31T12:00:00</manufacturerDate>
  <threadCount>0</threadCount>
  <MBread>0</MBread>
  <MBwrite>0</MBwrite>
  <recoveredReadErrors>0</recoveredReadErrors>
  <recoveredWriteErrors>0</recoveredWriteErrors>
  <unRecoveredReadErrors>0</unRecoveredReadErrors>
  <unRecoveredWriteErrors>0</unRecoveredWriteErrors>
  <encrypted>true</encrypted>
</ns2:mediaUsage>
```

Figure 111: moveMedium

```
<ns2:moveMedium  xmlns:ns2="http://automatedMediaLibrary/">
   <mode>2</mode> <!-- This is an optional field that provides the capability to take the partition(s) offline
that are involved in the move medium, only standard p[partitions apply. The default behavior is to do
nothing -->
   </sourceCoordinate> A coordinate object, see Figure 20: coordinate
  </destinationCoordinate> A coordinate object, see Figure 20: coordinate
</ns2:moveMedium>
```

Figure 112: network

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:network xmlns:ns2="http://automatedMediaLibrary/">
      </netInterfaceList> See Figure 116: netInterfaceList
      </netConfigurationList> See Figure 114: netConfigurationList
</ns2:network>
```

Figure 113: networkAccess

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:networkAccess
   xmlns:ns2="http://automatedMediaLibrary/">
   <icmp>
      <enabled>true</enabled>
   </icmp>
   <ssh>
      <enabled>false</enabled>
   </ssh>
   <cli>
      <enabled>true</enabled>
   </cli>
   <snmp>
      <enabled>true</enabled>
      <v1v2>true</v1v2>
   </snmp>
   <smis>
      <enabled>true</enabled>
      <secure>false</secure>
   </smis>
   <cvtl>
      <enabled>false</enabled>
   </cvtl>
   <xml>
      <enabled>true</enabled>
   </xml>
</ns2:networkAccess>
```

Figure 114: netConfigurationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netConfigurationList xmlns:ns2="http://automatedMediaLibrary/">
   </netConfiguration> A list of netConfiguration objects
</ns2:netConfigurationList>
```

Figure 115: netConfiguration

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netConfiguration xmlns:ns2="http://automatedMediaLibrary/">
   <name>eth0</name>
   <location>N/A</location>
   <version>1</version> <!-- 1(IPv4), 2(IPv6) -->
   <hostName>dvt4</hostName> <!-- Valid characters, regx "^[a-zA-Z][a-zA-Z0-9\\-.]*$" -->
   <domainName>company.com</domainName>
   <type>1</type> <!-- -1(Unknown), 0(None), 1(Static), 2(DHCP), 3(DHCPv6), 4(Static and DHCPv6) -->
```

```
  <netMask>255.255.248.0</netMask>
  <netGateway>10.20.168.1</netGateway>
  <ipAddress>10.20.171.14</ipAddress>
</ns2:netConfiguration>
```

Figure 116: netInterfaceList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netInterfaceList xmlns:ns2="http://automatedMediaLibrary/">
     </netInterface> A list of netInterface objects, see Figure 117: netInterface
</ns2:netInterfaceList>
```

Figure 117: netInterface

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:netInterface  xmlns:ns2="http://automatedMediaLibrary/">
     <name>eth0</name>
     <macAddress>00:30:8C:06:78:D7</macAddress>
     <duplexMode>full</duplexMode>
     <autoNegotiate>false</autoNegotiate>
     <speed>1</speed>
     <linkStatus>1</linkStatus>
</ns2:netInterface>
```

Figure 118: NTP

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:NTP xmlns:ns2="http://automatedMediaLibrary/">
  <server>0.us.pool.ntp.org</server> <!--You can have 0 or more of these -->
</ns2:NTP>
```

Figure 119: partitionList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:partitionList xmlns:ns2="http://automatedMediaLibrary/">
  <partition/> A list of partition objects, see Figure 120: partition
</ns2:partitionList>
```

Figure 120: partition

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:partition xmlns:ns2="http://automatedMediaLibrary/">
  <id>0</id>       <!-- Used internally -->
  <name>name</name>
  <serialNumber>serialNumber</serialNumber>
  <type>1</type>           <!-- 1(Standard), 2(EDLM), 3(AMP), 4(Vault), 5(CVTL) -->
  <driveDomainType>554</driveDomainType>    <!—0(Unknown), 3(LTO2), 4(LTO3), 5(LTO4), 6(LTO5),
7(LTO6), 8(LTO7), 554(Mixed) -->
  <mode>0</mode>                  <!-- 1(Online), 2(Offline) -->
  <storageSlotCount>0</storageSlotCount>
  <driveCount>0</driveCount>
  <ieSlotCount>0</ieSlotCount>
  <xieSlotCount>0</xieSlotCount>
```

```
  <ampExtensionsCount>0</ampExtensionsCount>
  <mediaCount>0</mediaCount>
  <barcodeReporting>4</barcodeReporting>       <!-- 1(Prefix / Media ID first), 2(Suffix / Media ID last),
3(Disabled / Standard), 4(Pass Through / Extended), 5(Standard 6), 6(Plus 6) -->
  <vendorId>1</vendorId>                      <!-- 0(ADIC), 1(Quantum), 2(HP) -->
  <productId>7</productId>                    <!-- 1((ADIC) Scalar 24), 2((ADIC) Scalar 100), 3((ADIC)
Scalar 1000), 4((ADIC) Scalar 10k), 5((ADIC/Quantum) Scalar i500), 6((ADIC/Quantum) Scalar i2000),
7((ADIC/Quantum) Scalar i6000), 8((ADIC/Quantum) Scalar i40-i80), 9((ADIC/Quantum) Scalar iScalar
i3-i6) -->
  <controlPathProvider>0</controlPathProvider> <!-- 0(None), 1(Drive), 2(Management Control Blade) -->
  <policySettings>
       <driveFirmwareLevelingEnabled>false</driveFirmwareLevelingEnabled>
       <driveCleaningEnabled>false</driveCleaningEnabled>
       <driveSpoofingEnabled>true</driveSpoofingEnabled>
       <encryptionEnabled>true</encryptionEnabled>
       <activeVaultEnabled>false</activeVaultEnabled>
       <autoImportEnabled>true</autoImportEnabled>
       <autoExportEnabled>true</autoExportEnabled>
       <edlmEnabled>false</edlmEnabled>
   </policySettings>
</ns2:partition>
```

Figure 121: partitionEncryptionPolicyList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:partitionEncryptionPolicyList xmlns:ns2="http://automatedMediaLibrary/">
  </partitionEncryptionPolicy> <!-- A list of partitionEncryptionPolicy objects -->
</ns2:partitionEncryptionPolicyList>
```

Figure 122: partitionEncryptionPolicy

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:partitionEncryptionPolicy xmlns:ns2="http://automatedMediaLibrary/">
  <partitionName>partitionName</partitionName>
  </ekmServerType> <!—0(None), 2(RKM), 4(KMIP), 8(QEKM), 16(SKM), 32(TKLM) -->
  <!-- Note: RKM is no longer supported -->
  <libraryManaged>true</libraryManaged>
  <fipsEnabled>true</fipsEnabled>
  <keyReuse>true</keyReuse>
  <keyType>0</keyType> <!-- 1(Key per Media), 2(Key per Partition), 3(Key per Library) -->
</ns2:partitionEncryptionPolicy>
```

Figure 123: partitionNames

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:partitionNames xmlns:ns2="http://automatedMediaLibrary/">
    <name>Partition1</name> <!-- One or more partition names -->
</ns2:partitionNames>
```

Figure 124: physicalLibrary

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:physicalLibrary  xmlns:ns2="http://automatedMediaLibrary/">
    <name>dvt4</name> <!-- The library host name -->
    <serialNumber>273190049</serialNumber> <!-- The library serial number -->
```

```
   <vendorId>quantum</vendorId> <!-- The library vendor  name -->
   <productId>Scalar i6000</productId> <!-- The library type, i6k, i500, etc.. -->
   <mode>1</mode> <!—1 (Online) or 2 (Offline) -->
   <state>1</state> <!—1 (Ready) or 2 (Not Ready) -->
   <mediaInStorageCount>108</mediaInStorageCount> <!—The number of media in storage slots -->
   <mediaInDriveCount>1</mediaInDriveCount> <!—The number of media in drives -->
   <mediaInIeCount>3</mediaInIeCount> <!—The number of media in IE slots -->
   <storageSlotCount>756</storageSlotCount> <!—The number of storage slots -->
   <usedStorageSlotCount>102</usedStorageSlotCount> <!—The number of storage slots owned by
partitions-->
   <ieSlotCount>24</ieSlotCount> <!—The number of ie slots -->
   <usedIeSlotCount>6</usedIeSlotCount> <!—The number of ie slots owned by partitions-->
   <driveCount>9</driveCount> <!—The number of drives in the library -->
   <usedDriveCount>2</usedDriveCount> <!—The number of drives owned by partitions -->
   <libraryController>MCB2B</libraryController>
   <roboticController>RCU2</roboticController>
   <firmwareVersion>665Q.DS02101</firmwareVersion> <!-- The library firmware version -->
   <roboticsGeneration>2</roboticsGeneration> <!-- The library robot type 1 (Gen1) or 2 (Gen2) -->
   <moduleCount>3</moduleCount> <!-- The number of frames/modules in the library -->
   <towerCount>0</towerCount> <!-- The number of towers in the library -->
   <parkingModules>true</parkingModules> <!-- Does the library have parking modules -->
   <cleaningSlotCount>0</cleaningSlotCount> <!-- The number of cleaning slots in the library -->
   <cleaningMediaCount>0</cleaningMediaCount> <!-- The number of cleaning media in the library -->
   <libraryUpTime>6 days 18h:21m:40s</libraryUpTime> <!—How long has the library been powered up
since last power down -->
   <phySettings>
      <driveSerialNumberSpoofing>
         <enabled>true</enabled>
      </driveSerialNumberSpoofing>
      <autoInventory>
         <enabled>false</enabled>
      </autoInventory>
      <autoCalibration> <!-- Deprecated in i12.X  -->
         <enabled>false</enabled>
      </autoCalibration>
      <autoConfiguration> <!-- Deprecated in i12.X  -->
         <enabled>false</enabled>
      </autoConfiguration>
      <autoCleaning> <!-- Deprecated in i12.X, this is now provided on a partition basis
(aml/partitions/policy/driveCleaning) -->
         <enabled>true</enabled>
      </autoCleaning>
      <autoDriveUnload>
         <enabled>true</enabled>
      </autoDriveUnload>
      <ipv6>
         <enabled>true</enabled> <!-- As of i12.3 you can no longer disable IPv6, it will always be enabled
-->
      </ipv6>
      <extendedIe>
         <enabled>true</enabled>
      </extendedIe>
      <sendUsageStatistics>
         <interval>1</interval> <!-- 0(Disabled), 1(Monthly) and 2(Quarterly) -->
      </sendUsageStatistics>
      <healthCheck>
```

```
        <railInterval>0</railInterval> <!-- 0-180 days (0 to disable) -->
        <robotInterval>0</robotInterval> <!-- 0-180 days (0 to disable) -->
        <towerInterval>0</towerInterval> <!-- 0-180 days (0 to disable) -->
      </healthCheck>
      <aisleLights> <!--  Set the duration for aisle lighting -->
        <interval>0</interval> <!-- 0 (off), 30 minutes or 60 minutes -->
      </aisleLights>
      <webCamera> <!-- The IP of the host where the Camera (i6000) Application is running -->
        <ipAddress>10.20.9.1</ipAddress>
      </webCamera>
      <icmpService> <!-- Allow PINGing the library -->
        <enabled>true</enabled>
      </icmpService>
      <sshService> <!-- Secure Shell connections to the library -->
        <enabled>true</enabled>
      </sshService>
      <cliService> <!-- HP CLI interface to the library -->
        <enabled>true</enabled>
      </cliService>
      <xmlInterfaceService> <!-- Currently the interface used by Vision to get library information -->
        <enabled>true</enabled>
      </xmlInterfaceService>
      <serviceLogin> <!-- Allow/disallow service user login -->
        <enabled>true</enabled>
        <enabledRemoteAccess>true</enabledRemoteAccess> <!-- Enable a service user to login from a
remote interface (browser) -->
        <remoteAccessTimeout>0</remoteAccessTimeout> <!-- The amount of time before a service
users remote access will be diasabled after it has been enabled. Values are 0 to 72 hours, where 0 is
indefinitely -->
        <enableLocalAccess>true</enableLocalAccess> <!-- Enable a service user to login from the local
user interface -->
        <localAccessTimeout>0</localAccessTimeout> <!-- The amount of time before a service users
local access will be diasabled after it has been enabled. Values are 0 to 72 hours, where 0 is indefinitely -->
        <serviceUserSession>240</serviceUserSession> <!-- length of the service user session, in
minutes (Not modifiable) -->
        <remoteAccessGranted>2013-05-29 13:17:36</remoteAccessGranted> <!-- date/time remote
access was granted -->
        <localAccessGranted>2013-05-29 13:17:36</localAccessGranted> <!-- date/time local access
was granted -->
      </serviceLogin>
      <sessionTimeout> <!--  The user session timeout -->
        <minutes>30</minutes> <!-- (1 - 1440) -->
      </sessionTimeout>
      <snmp>
        <communityString>publicCmtyStr</communityString>
        <enabled>true</enabled>
        <enableVersion1And2>true</enableVersion1And2>
        <enableVersion3>false</enableVersion3> <!-- Not supported on i6k -->
        <enableAuthenticationTraps>false</enableAuthenticationTraps> <!-- Not supported on i6k -->
      </snmp>
      <smis>
        <enabled>true</enabled>
        <enableSecureSmis>false</enableSecureSmis>
      </smis>
    </phySettings>
</ns2:physicalLibrary>
```

Figure 125: physicalLibraryConfiguration

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:physicalLibraryConfiguration xmlns:ns2="http://automatedMediaLibrary/">
  <name>dvt4</name> <!-- The library host name -->
  <serialNumber>273190049</serialNumber> <!-- The library serial number -->
  <vendorId>quantum</vendorId> <!-- The library vendor  name -->
  <productId>Scalar i6000</productId> <!-- The library type, i6k, i500, etc.. -->
  <mode>1</mode> <!—1 (Online) or 2 (Offline) -->
  <state>1</state> <!—1 (Ready) or 2 (Not Ready) -->
  <libraryController>MCB2B</libraryController>
  <roboticController>RCU2</roboticController>
  <firmwareVersion>firmwareVersion</firmwareVersion>
  <roboticsGeneration>0</roboticsGeneration> <!-- The library robot type 1 or 2 -->
  <libraryUpTime>6 days 18h:21m:40s</libraryUpTime> <!—How long has the library been powered up
since last power down -->
</ns2:physicalLibraryConfiguration>
```

Figure 126: physicalLibraryRemoteAccess

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:physicalLibraryRemoteAccess xmlns:ns2="http://automatedMediaLibrary/">
  <ipv6>
    <enabled>true</enabled> <!-- As of i12.3 you can no longer disable IPv6, it will always be enabled -->

  </ipv6>
  <icmpService> <!-- Allow/disallow PINGing the library -->
    <enabled>true</enabled>
  </icmpService>
  <sshService> <!-- Secure Shell connections to the library -->
    <enabled>true</enabled>
  </sshService>
  <cliService> <!-- HP CLI interface to the library -->
    <enabled>true</enabled>
  </cliService>
  <xmlInterfaceService> <!-- Enable Vision Interface -->
    <enabled>true</enabled>
  </xmlInterfaceService>
  <serviceLogin> <!-- Allow/disallow service user login -->
    <enabled>true</enabled>
    <enabledRemoteAccess>true</enabledRemoteAccess> <!-- Enable a service user to login from a
remote interface (browser) -->
        <remoteAccessTimeout>0</remoteAccessTimeout> <!-- The amount of time before a service
users remote access will be diasabled after it has been enabled. Values are 0 to 72 hours, where 0 is
indefinitely -->
        <enableLocalAccess>true</enableLocalAccess> <!-- Enable a service user to login from the local
user interface -->
        <localAccessTimeout>0</localAccessTimeout> <!-- The amount of time before a service users
local access will be diasabled after it has been enabled. Values are 0 to 72 hours, where 0 is indefinitely -->
        <serviceUserSession>240</serviceUserSession> <!-- length of the service user session, in
minutes (Not modifiable) -->
        <remoteAccessGranted>2013-05-29 13:17:36</remoteAccessGranted> <!-- date/time remote
access was granted -->
        <localAccessGranted>2013-05-29 13:17:36</localAccessGranted> <!-- date/time local access
was granted -->
```

```
  </serviceLogin>
  <sessionTimeout> <!--  The user session timeout in minutes, 1 – 1440 -->
    <minutes>0</minutes>
  </sessionTimeout>
</ns2:physicalLibraryRemoteAccess>
```

Figure 127: physicalLibraryResources

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:physicalLibraryResources xmlns:ns2="http://automatedMediaLibrary/">
  <mediaInStorageCount>0</mediaInStorageCount>
  <mediaInDriveCount>0</mediaInDriveCount>
  <mediaInIeCount>0</mediaInIeCount>
  <storageSlotCount>0</storageSlotCount>
  <assignedStorageSlotCount>0</assignedStorageSlotCount>
  <ieSlotCount>0</ieSlotCount>
  <assignedIeSlotCount>0</assignedIeSlotCount>
  <driveCount>0</driveCount>
  <assignedDriveCount>0</assignedDriveCount>
  <moduleCount>0</moduleCount>
  <towerCount>0</towerCount>
  <parkingModules>true</parkingModules>
  <cleaningSlotCount>0</cleaningSlotCount>
  <cleaningMediaCount>0</cleaningMediaCount>
</ns2:physicalLibraryResources>
```

Figure 128: physicalLibrarySettings

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:physicalLibrarySettings xmlns:ns2="http://automatedMediaLibrary/">
  <driveSerialNumberSpoofing>
    <enabled>true</enabled>
  </driveSerialNumberSpoofing>
  <autoInventory>
    <enabled>true</enabled>
  </autoInventory>
  <autoCalibration> <!-- Deprecated in i12.X  -->
    <enabled>true</enabled>
  </autoCalibration>
  <autoConfiguration> <!-- Deprecated in i12.X  -->
    <enabled>true</enabled>
  </autoConfiguration>
  <autoCleaning> <!-- Deprecated in i12.X, this is now provided on a partition basis
(aml/partitions/policy/driveCleaning) -->
    <enabled>true</enabled>
  </autoCleaning>
  <autoDriveUnload>
    <enabled>true</enabled>
  </autoDriveUnload>
  <extendedIe>
    <enabled>true</enabled>
  </extendedIe>
  <sendUsageStatistics>
    <interval>0</interval> <!-- 0(Disabled), 1(Monthly) and 2(Quarterly) -->
  </sendUsageStatistics>
  <healthCheck>
```

```
    <railInterval>0</railInterval> <!-- 0-180 days (0 to disable) -->
    <robotInterval>0</robotInterval> <!-- 0-180 days (0 to disable) -->
    <towerInterval>0</towerInterval> <!-- 0-180 days (0 to disable) -->
  </healthCheck>
  <aisleLights> <!--  Set the duration for aisle lighting 0 (off), 30 minutes or 60 minutes -->
    <interval>0</interval>
  </aisleLights>
  <webCamera> <!-- The IP of the host where the Camera (i6000) Application is running -->
    <ipAddress>ipAddress</ipAddress>
  </webCamera>
  <snmp>
    <communityString>communityString</communityString>
    <enabled>true</enabled>
    <enableVersion1And2>true</enableVersion1And2>
    <enableVersion3>true</enableVersion3> <!-- Not supported on i6k -->
    <enableAuthenticationTraps>true</enableAuthenticationTraps> <!-- Not supported on i6k -->
  </snmp>
  <smis>
    <enabled>true</enabled>
    <enableSecureSmis>true</enableSecureSmis>
  </smis>
</ns2:physicalLibrarySettings>
```

Figure 129: ping

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ping   xmlns:ns2="http://automatedMediaLibrary/">
    <firmwareVersion>665Q.DS02401</firmwareVersion>
    <productName>Scalar i6000</productName> <!-- “Scalar i6000”, “Scalar i350” or “Scalar i700” -->
    <serialNumber>273190048</serialNumber>
    <vendor>Vendor Name</vendor> <!-- “Quantum” or “HP” -->
</ns2:ping>
```

Figure 130: port

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:port xmlns:ns2="http://automatedMediaLibrary/">
  <id>Port 1, Port A, Left, Right …</id>
  <type>1</type> <!-- 1(SCSI), 2(Fibre), 3(SAS) -->
  <address>WWPN</address>
  <topology>
    <actual>2</actual>    <!--1(Loop Preferred “LN”), 2(P2P/Fabric “N”), 3(Loop “L”), 4(P2P/Fabric
preferred “NL”) -->
    <requested>2</requested>    <!--1(Loop Preferred “LN”), 2(P2P/Fabric “N”), 3(Loop “L”), 4(P2P/Fabric
preferred “NL”) -->
  </topology>
  <loopId>0</loopId>
  <speed>
    <actual>4</actual>              <!-- 0(Auto), 1(1Gb/s), 2(2Gb/s), 4(4Gb/s), 8(8Gb/s) -->
    <requested>4</requested>
  </speed>
  <status>0</status>                <!-- 0(N/A), 1(Down), 2(Active), 4(Passive) -->
</ns2:port>
```

Figure 131: RASGroupStatusList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASGroupStatusList  xmlns:ns2="http://automatedMediaLibrary/">
   </RASGroupStatus> A list of RASGroupStatus objects
</ns2:RASGroupStatusList>
```

Figure 132: RASGroupStatus

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASGroupStatus  xmlns:ns2="http://automatedMediaLibrary/">
      <group>1</group> <!--
0 (All/Library), 0 to represent the overall library status
1 (Connectivity),
2 (Control),
3 (Media),
4 (Drives),
5 (Power),
6 (Robotics)
.-->
      <status>1</status> <!-- 1 (Good), 2 (Failed), 3 (Degraded), 4 (Warning), 5 (Informational), 6
(Unknown), 7 (Invalid), 8(Attention) -->
</ns2:RASGroupStatus>
```

Figure 133: RASReportList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASReportList   xmlns:ns2="http://automatedMediaLibrary/">
         </RASReport> A list of RASReport objects
</ns2:RASReportList>
```

Figure 134: RASReport

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASReport   xmlns:ns2="http://automatedMediaLibrary/">
         <reportId>42</reportId>
         <postedDate>2013-05-29 13:17:36 +0000</postedDate>
         <duplicates>0</duplicates>
         </RASGroupStatus> <!-- See Figure 132: RASGroupStatus -->
         <eventCode>09_09_18_00_00000000</eventCode>
         <modifier>0x0</modifier>
         <summary>Drive [1, 1, 1, 12, 1, 1] lost network link, will be unable to reconfigure</summary>
         <description>Control of Tape Drive at [1,1,1,12,1,1] communication has failed</description>
         <headReport>41</headReport>
         <keyReportId>0</keyReportId>
         <serialNumber>GB120401FD</serialNumber>
         <repairLink>09_09_18_htm</repairLink>
</ns2:RASReport>
```

Figure 135: RASTicketReports

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASTicketReports   xmlns:ns2="http://automatedMediaLibrary/">
   <ticketId>15</ticketId>
   </RASReportList> A RASReportList object, see Figure 133: RASReportList
```

```
</ns2:RASTicketReports>
```

Figure 136: RASTicketList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASTicketList  xmlns:ns2="http://automatedMediaLibrary/">
   <RASTicket>
      <ticketId>5</ticketId>
      <name></name>
      <description>Control of Tape Drive at [1,1,1,10,1,1] communication has failed</description>
      <closed>1970-01-01 00:00:00 +0000</closed>
      <opened>2013-05-16 20:36:43 +0000</opened>
      <eventCode>0</eventCode>
      <groupStatus>
         <group>4</group>
         <status>2</status>
      </groupStatus>
      <RASTicketState>
         <state>2</state>
      </RASTicketState>
      <duplicates>0</duplicates>
      <lastUpdate>2013-06-11 10:20:05 +0000</lastUpdate>
      <serialNumber>HU1231PJTT</serialNumber>
      <repairLink>09_09_18_htm</repairLink>
      <keyReportId>17</keyReportId>
      <rasTicketDetails/>
   </RASTicket>
</ns2:RASTicketList>
```

Figure 137: RASTicket

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASTicket xmlns:ns2="http://automatedMediaLibrary/">
      <ticketId>5</ticketId>
      <name></name>
      <description>Control of Tape Drive at [1,1,1,10,1,1] communication has failed</description>
      <closed>1970-01-01 00:00:00 +0000</closed>
      <opened>2013-05-16 20:36:43 +0000</opened>
      <eventCode>0</eventCode>
      </groupStatus> <!-- RASGroupStatus object, see Figure 132: RASGroupStatus -->
      </RASTicketState> <!-- A RASTicketState object -->
      <duplicates>0</duplicates>
      <lastUpdate>2013-06-11 10:20:05 +0000</lastUpdate>
      <serialNumber>HU1231PJTT</serialNumber>
      <repairLink>09_09_18_htm</repairLink>
      <keyReportId>17</keyReportId>
      <rasTicketDetails/>
      <severity>1</severity> <-- 1(critical), 2(degraded), 3(warning), 4(attention) and 5(informational)
      <fruLocation>1,1,1,1,1,3</fruLocation> <!-- Where the component is location in the library -->
</ns2:RASTicket>
```

Figure 138: RASTicketState

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RASTicketState xmlns:ns2="http://automatedMediaLibrary/">
```

```
    <state>5</state>  <!-- 1 (New), 2 (Open), 3 (Suspended), 4 (Closed), 5 (Verified) -->
</ns2:RASTicketState>
```

Figure 139: rasNotificationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:rasNotificationList   xmlns:ns2="http://automatedMediaLibrary/">
        </rasNotification> A list of rasNotification objects
</ns2:rasNotificationList>
```

Figure 140: rasNotification

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:rasNotification xmlns:ns2="http://automatedMediaLibrary/">
  <id>1</id>  <!-- This id is generated and is used to lookup a reportNotification object. Each
reportNotification gets a unique id. -->
  <enabled>true</enabled>  <!-- Enable/Disable the notification, when disabled the notification will not be
sent -->
  <emailAddess>jim.boyd@quantum.com </emailAddress> <!-- E-mail address -->
  <severity1>true</severity1> <!-- Send Severity 1 Tickets -->
  <severity2>true</severity2> <!-- Send Severity 2 Tickets -->
  <severity3>true</severity3> <!-- Send Severity 3 Tickets -->
  <severity4>true</severity4> <!-- Send Severity 4 Tickets -->
  <severity5>true</severity5> <!-- Send Severity 5 Tickets -->
  <includeSnapshot>false</includeSnapshot>  <!-- Do we include a snapshot with the notification. This is
not supported on the i6k -->
  <lastSent>Date</lastSent> <!-- The last time the notification was sent -->
</ns2:rasNotification>
```

Figure 141: reportList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:reportList   xmlns:ns2="http://automatedMediaLibrary/">
        </report> A list of report objects
</ns2:reportList>
```

Figure 142: report

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:report  xmlns:ns2="http://automatedMediaLibrary/">
      <name>Drive Utilization</name> <!-- The report name -->
      <type>The Report Type</type> <!-- 1, "Drive Utilization", 2, "Media Integrity Analysis", 3, "Media
Usage", 4, "Media Security", 5, "Cross Partition Media Moves", 6, "Tickets", 7, "LUN Mapping", 8, "Media
Inventory", 9, "EKM Partition Activity", 10, "EKM Media Status", 11, "Login Activity", 12, "Verification Test",
13, "Library Configuration", 14, "Partition Utilization", 15, “Drive Cleaning” -->
      <license>Advanced Reporting</license> <!-- The required license -->
      <format>CSV</format> <!-- The format supported when the report is saved -->
      <recordCount>123</recordCount> <!-- The number of records, or if TXT format -->
      <template /> <!--  A list of reportTemplate objects, see Figure 147: reportTemplate -->
      <relations>
        <link rel="item" title="Drive Utilization" href="http://library/aml/drives/reports/utilization"/>
      </relations>
</ns2:report>
```

Figure 143: reportCriteria

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:reportCriteria xmlns:ns2="http://automatedMediaLibrary/">
    <start>0</start> <!-- The start offset in the list of records, 0 is the default, the first record -->
    <length>0</length> <!-- The number of records, you want to retrieve, 0 is default and means all records after the start offset -->
    <period>0</period> <!-- The last number of days to report. If I want a report for the last week, this value would be 7 -->
    <date>Query start date</date> <!-- At what date you want to start your query. The data returned will include all records that are equal to or older than the date specified. When used with the period parameter, the data returned will include all records that are equal or older than the date specified up to the period (number of days) specified. The date format expected is “yyyy-MM-dd HH:mm:ss” or “yyyy-MM-dd HH:mm:ss Z” the Z (time zone) field will be ignored. -->
    <partition>partition name</partition> <!-- Some reports can be filtered on partition name -->
    <driveSerialNumber>serial number</driveSerialNumber> <!-- Some reports can be filtered on the drive physical serial number -->
    <barcode>drive vendor</barcode> <!-- Some reports can be filtered on the media barcode-->
</ns2:reportCriteria>
```

Figure 144: reportNotificationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:reportNotificationList xmlns:ns2="http://automatedMediaLibrary/">
   <reportNotification/> A list of reportNotification objects, See Figure 145: reportNotification
</ns2:reportNotificationList>
```

Figure 145: reportNotification

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:reportNotification xmlns:ns2="http://automatedMediaLibrary/">
   <id>4</id>  <!-- This id is generated and is used to lookup a reportNotification object. Each reportNotification gets a unique id. -->
   <enabled>true</enabled>  <!-- Enable/Disable the notification, when disabled the notification will not be sent -->
   <emailAddress>john.doe@company.com </emailAddress>  <!-- The e-mail address where the notification will be sent too -->
   <reportTemplateName>LoginActivity</reportTemplateName>  <!-- The report template name, See Figure 147: reportTemplate. -->
   <interval>
      <frequency>1</frequency>  <!-- The frequency the report will be sent, valid values are 1 (Daily), 2 (Weekly), 3 (Monthly) and 4 (Quarterly) -->
      <dayOfWeek>4</dayOfWeek>   <!-- The day of the week the report should be sent, 0 is Sunday, 6 is Saturday -->
      <hourOfDay>0</hourOfDay>   <!-- The hour of the day the report should be sent, 0 - 23 -->
    </interval>
</ns2:reportNotification>
```

Figure 146: reportTemplateList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:reportTemplateList xmlns:ns2="http://automatedMediaLibrary/">
  <reportTemplate/> A list of reportTemplate objects, see Figure 147: reportTemplate
</ns2:reportTemplateList>
```

Figure 147: reportTemplate

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:reportTemplate xmlns:ns2="http://automatedMediaLibrary/">
  <id>Template ID</id> <!-- Only used to lookup template -->
  <name>name</name>  <!-- The name can only contain the following characters A-Z a-z 0-9 _ and
spaces. The maximum number of character allowed is 64 -->
  <reportType>1</reportType> <!-- 1, "Drive Utilization", 2, "Media Integrity Analysis", 3, "Media Usage", 4,
"Media Security", 5, "Cross Partition Media Moves", 6, "Tickets", 7, "LUN Mapping", 8, "Media Inventory",
9, "EKM Partition Activity", 10, "EKM Media Status", 11, "Login Activity", 12, "Verification Test", 13, "Library
Configuration", 14, "Partition Utilization", 15, “Drive Cleaning” -->
  <reportCriteria/> <!-- A reportCriteria object Figure 143: reportCriteria -->
</ns2:reportTemplate>
```

Figure 148: robotList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:robotList xmlns:ns2="http://automatedMediaLibrary/">
  <robot/> A list of 1 or 2 robot objects, see Figure 149: robot
</ns2:robotList>
```

Figure 149: robot

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:robot xmlns:ns2="http://automatedMediaLibrary/">
  <name>name</name>
  <status>0</status>                <!-- 1(Active), 2(Passive), 3(Pending), 4(Failed) -->
  <state>0</state>                  <!-- 1(Varied On), 2(Varied Off) -->
  <serialNumber>serialNumber</serialNumber>
  <firmwareVersion>firmwareVersion</firmwareVersion>
  <parked>true</parked>
  <present>true</present>
  <generation>generation</generation>
</ns2:robot>
```

Figure 150: segmentList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segment   xmlns:ns2="http://automatedMediaLibrary/">
       </segment> <!-- A list of segment objects -->
 </ns2:segment>
```

Figure 151: segment

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:segment   xmlns:ns2="http://automatedMediaLibrary/">
       <coordinate/> <!-- See Figure 20: coordinate -->
       <size>6</size>
       <owner>LL1</owner> <!—The partition who owns the segment -->
       <configuredType>0</configuredType> <!-- On the i500 IE segments can be configured as storage -->
 </ns2:segment>
```

Figure 152: sensorList

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

```
<ns2:sensorList   xmlns:ns2="http://automatedMediaLibrary/">
  </sensor> A list of sensor objects
</ns2:sensorList>
```

Figure 153: sensor

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:sensor  xmlns:ns2="http://automatedMediaLibrary/">
      <!-- All elements are strings -->
      <name>RCS FAN1</name>
      <type>Cooling</type>
      <status>Nominal</status>
      <value>5818</value>
      <unit>RPM</unit>
      <location>Library (LPC) Cooling Fan #1</location>
</ns2:sensor>
```

Figure 154: serviceLogList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:serviceLogList xmlns:ns2="http://automatedMediaLibrary/">
   <serviceLog/> <!-- A list of serviceLog objects -->
</ns2:serviceLogList>
```

Figure 155: serviceLog

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:serviceLog   xmlns:ns2="http://automatedMediaLibrary/">
      <name>vt_2U31090031_2015-06-18_17.21.48.log</name> <!-- The name of the log -->
      <type>1</type> <!-- 0(N/A), 1(VT), 2(SCSI), 3(GET/PUT Statistics) -->
      <date>2015-06-18 17:22:59</date> <!-- The date the log file was last modified -->
</ns2:serviceLog>
```

Figure 156: shutdownTask

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:shutdownTask xmlns:ns2="http://automatedMediaLibrary/">
  <reboot>false</reboot> <!-- Set to true if you want the library to reboot, otherwise the library needs to be
power cycled -->
</ns2:shutdownTask>
```

Figure 157: trapNotificationList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:trapNotificationList xmlns:ns2="http://automatedMediaLibrary/">
   <trapNotification/> A list of trapNotification objects.
</ns2:trapNotificationList>
```

Figure 158: trapNotification

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:trapNotification xmlns:ns2="http://automatedMediaLibrary/">
      <id>Unique ID</id> <!-- Used to lookup object -->
      <host>10.20.1.100</host> <!-- The host or IP of the host that will receive the trap -->
      <port>162</port>
```

```
    <transportType>1</transportType> <!-- 1(UDP IPv4), 2(UDP IPv6), 3(TCP IPv4), 4(TCP IPv6) This is
not supported on i6k -->
    <communityString>publicCmtyStr</communityString> <!-- Not required for Quantum branded i6k
libraries -->
    <version>1</version> <!-- 1(SNMPv1), 2(SNMPv2c), 3(SNMPv3) currently not supported on i6k -->
</ns2:trapNotification>
```

Figure 159: tapeAlertList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:tapeAlertList xmlns:ns2="http://automatedMediaLibrary/">
  <tapeAlert/> A list of tapeAlert objects
</ns2:tapeAlertList>
```

Figure 160: tapeAlert

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:tapeAlert xmlns:ns2="http://automatedMediaLibrary/">
  <driveSerialNumber>driveSerialNumber</driveSerialNumber>
  <barcode>barcode</barcode>
  <tapeAlert>0</tapeAlert>
  <dateTime>2001-12-31T12:00:00</dateTime>
</ns2:tapeAlert>
```

Figure 161: taskList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:taskList xmlns:ns2="http://automatedMediaLibrary/">
    <task/> A list of task objects
</ns2:taskList>
```

Figure 162: task

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:task xmlns:ns2="http://automatedMediaLibrary/">
    <id>91</id> <!-- A unique task id number, used to look up a particular task -->
    <componentId>LL1</componentId> <!-- Drive serial number, partition name, blade coordinate, etc..-->
    <type>1</type>
<!-- 0(All types),
1 (Inventory),
2 (Library Shutdown),
3 (Library Reboot),
4 (Identify Drive),
5 (Drive Clean),
6 (Power Cycle FC IO Blade),
7 (Reset FC IO Blade),
8 (Identify FC IO Blade),
9 (Identify Ethernet Expansion Blade),
10 (Auto Import Media),
11 (Generate Command History Logs)
```

Figure 163: ticketFilterList

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ticketFilterList xmlns:ns2="http://automatedMediaLibrary/">
```

```
  </ticketFilter> <!-- A list of ticketFilter objects -->
</ns2:ticketFilterList>
```

Figure 164: ticketFilter

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ticketFilter xmlns:ns2="http://automatedMediaLibrary/">
  <id>0</id>
  <filter>0</filter> <!-- 0(No Suppression, e-mail notifications will be sent), 1(No E-mail to Technical
Support), 2(No E-mail will be sent to anybody), 3(The RAS ticket will not be generated) -->
  <description>description</description> <!-- RAS Ticket description -->
  <tapeAlertNumber>22</tapeAlertNumber>
  <category>1</category> <!-- 1(Connectivity), 2(Control), 3(Media), 4(Drives), 5(Power) 6(Robotics) -->
</ns2:ticketFilter>
```

Figure 165: timeZoneIDs

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:timeZoneIDs   xmlns:ns2="http://automatedMediaLibrary/">
    <ID>(GMT-12:00) Etc/GMT+12 (GMT-12:00)</ID>
    <ID>(GMT-11:00) Etc/GMT+11 (GMT-11:00)</ID>
    <ID>(GMT+13:00) MIT (WSDT)</ID>
    <ID>(GMT+13:00) Pacific/Apia (WSDT)</ID>
    <ID>(GMT-11:00) Pacific/Midway (SDT)</ID>
    <ID>(GMT-11:00) Pacific/Niue (NUST)</ID>
    <ID>(GMT-11:00) Pacific/Pago_Pago (SDT)</ID>
    <ID>(GMT-11:00) Pacific/Samoa (SDT)</ID>
    <ID>(GMT-11:00) US/Samoa (SDT)</ID>
    <ID>(GMT-10:00) America/Adak (HADT)</ID>
    ……..
</ns2:timeZoneIDs>
```

Figure 166: towerList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:towerList xmlns:ns2="http://automatedMediaLibrary/">
  <tower/> A list of tower objects, see Figure 167: tower
</ns2:tower>
```

Figure 167: tower

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:tower xmlns:ns2="http://automatedMediaLibrary/">
  <coordinate/> See Figure 20: coordinate
  <serialNumber>serialNumber</serialNumber>
  <firmwareVersion>firmwareVersion</firmwareVersion>
  <scannerPresent>true</scannerPresent>
  <doorOpened>true</doorOpened>
  <mode>0</mode>       <!-- 1(Online), 2(Offline) -->
  <state>1</state>     <!-- 1(Varied On), 2(Varied Off) -->
  <status>0</status>   <!-- 1(Not Present), 2(Failed), 3(Not Ready), 4(Initializing), 5(Ready) -->
</ns2:tower>
```

Figure 168: userList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:userList  xmlns:ns2="http://automatedMediaLibrary/">
   </user> A list of user objects
</ns2:userList>
```

Figure 169: user

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:user   xmlns:ns2="http://automatedMediaLibrary/">
   <name>JohnDoe</name>
   <role>1</role> <!-- -1(Ignore this field), 0 (Admin), 1 (User), 2 (Service), 3 (Guest)
   <partitionAccess>Test Partition</partitionAccess>
   <partitionAccess>Sales Partition</partitionAccess>
   <activeCount>0</activeCount>
   <ldap>false</ldap>
</ns2:user>
```

Figure 170: userSessionList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:userSessionList   xmlns:ns2="http://automatedMediaLibrary/">
   </userSession> A list of userSession objects
</ns2:userSessionList>
```

Figure 171: userSession

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:userSession   xmlns:ns2="http://automatedMediaLibrary/">
      <id>3</id>
      <loginTime>2013-01-10 14:43:29 -0700</loginTime>
      <name>johndoe</name>
      <role>1</role>
      <lastActivityTime>2013-01-10 14:43:29 -0700</lastActivityTime>
      <loginFrom>10.20.9.123</loginFrom>
</ns2:userSession>
```

Figure 172: vtReportList

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:vtReportList  xmlns:ns2="http://automatedMediaLibrary/">
   <report>vt_2U31090031_2014-06-23_16.09.52</report> <!-- The report name -->
   <report>vt_2U31090031_2014-06-23_16.49.36</report>
</ns2:vtReportList>
```

Figure 173: WSResultCode

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:WSResultCode   xmlns:ns2="http://automatedMediaLibrary/">
   <code>200</code> <!-- HTTP Status code -->
   <description>OK</description> <!-- A brief description of this code -->
   <summary>Operation Completed Successfully</summary> <!-- A summary of the result of the
operation -->
   <action>Upload library firmware</action> <!-- A description of the action/operation that was requested -
```

```
->
   <customCode>0</customCode> <!-- An optional custom code -->
</ns2:WSResultCode>
```

5.2 Scalar i6k WSResultCode Custom Codes

The Scalar i6k WSResultCode object contains a customCode element which may provide more information when an error occurs for any given Web Service request. The customCode is only used when a Web Services request encounters an error, regardless of it being a client or server problem.

The following lists the codes and describes their meaning. If a customCode is not available, a value of 0 (zero) will be returned.

Table 348: SCSI Custom Codes 10000 – 80000

ASC	ASCQ	Custom Code	Description
		10001	Reservation Conflict
00h	17h	10023	Drive requests cleaning
04h	00h	11024	Robotics not ready due to an unknown cause
04h	01h	11025	Robotics are becoming ready
04h	03h	11027	Robotics not ready; manual intervention required
04h	83h	11155	Robotics not ready; aisle power disabled
04h	8Dh	11165	Robotics not ready
15h	01h	15377	Robotics positioning error
15h	80h	15504	Robotics dropped a cartridge
15h	81h	15505	Robotics could not pick a cartridge
15h	83h	15507	Robotics could not place a cartridge
15h	84h	15508	Move completed with cartridge Get recoveries
15h	85h	15509	Move completed with cartridge Put recoveries
15h	86h	15510	Move completed but cartridge was placed in alternate location
15h	91h	15521	Media stranded in picker
15h	92h	15522	Media stranded in picker
15h	93h	15523	Operation needs to be retried
1Ah	00h	16656	Logic error: Parameter list length error
20h	00h	18192	Logic error: Illegal OpCode in CDB
21h	01h	18449	Logic error: Invalid element address in CDB
24h	00h	19216	Logic error: Invalid field in CDB
24h	80h	19344	Logic error: Attempt to write a read only buffer
25h	00h	19472	Logic error: Illegal LUN
26h	00h	19728	Logic error: Invalid field in Parameter List
26h	01h	19729	Logic error: Parameter not supported
26h	02h	19730	Logic error: Parameter value invalid
26h	80h	19856	Logic error: Parameter data checksum failure
26h	81h	19857	Logic error: Parameter value already in use

28h	00h	20240	Robotics transitioned to ready state: Element status may have changed.
28h	01h	20241	Insert/Eject area opened and closed: I/E element status may have changed
29h	00h	20496	Power-on or reset occurred
29h	01h	20497	Power on occurred
29h	04h	20500	Internal reset occurred
29h	81h	20625	Library reset into degraded mode of operation
2Ah	01h	20753	Mode parameters have been changed
2Ah	80h	20880	Library inventory completed
30h	00h	22288	Incompatible medium installed
30h	07h	22295	Cleaning failure
30h	82h	22418	Drive cleaning operation complete
30h	90h	22432	Drive firmware update complete
30h	91h	22433	Drive firmware update failed
30h	92h	22434	Invalid firmware image
3Bh	0Dh	25117	Destination element is full
3Bh	0Eh	25118	Source element is empty
3Bh	12h	25122	Media magazine not installed
3Bh	85h	25237	Logic Error: Destination of MOVE cannot be picker
3Bh	A0h	25264	Logic error: Media type does not match destination media type
3Eh	00h	25872	Robotics discovery/teach not complete
3Eh	03h	25875	Robotics self-test failed
3Fh	01h	26129	New firmware loaded
3Fh	04h	26132	Drive added
3Fh	80h	26256	EEPROM failed to erase
3Fh	84h	26260	EEPROM program failure
3Fh	90h	26272	Robotics status changed
3Fh	91h	26273	Robotics access timer started
3Fh	92h	26274	Robot replacement succeeded
3Fh	93h	26275	Robot replacement failed
3Fh	94h	26276	Replacement robot detected
40h	80h	26512	Component failure
40h	91h	26529	Gripper failure
40h	A0h	26544	Robotics vertical axis motion failure
40h	A1h	26545	Robotics vertical axis homing failure
40h	B0h	26560	Robotics horizontal axis motion failure
40h	B1h	26561	Robotics horizontal axis homing failure
40h	C0h	26576	Robotics motion failure
40h	E0h	26608	Robotics power failure
44h	00h	27408	Internal logic failure
44h	81h	27537	Drive communication not established
44h	82h	27538	Drive communication lost
44h	83h	27539	Drive powered off
4Ch	00h	29456	Robotics discovery/teach failure
53h	00h	31248	Drive did not load or unload a tape cartridge
53h	01h	31249	A drive did not unload a tape cartridge

53h	02h	31250	A drive is preventing media removal
53h	80h	31376	Tape cartridge in I/E station not properly inserted
53h	81h	31377	Insert/Eject station door is open
53h	82h	31378	I/E station lock failure
53h	83h	31379	I/E station unlock failure
53h	84h	31380	Medium in drive is unloaded and robot accessible
5Dh	00h	33808	Tape Alert reported or failure prediction threshold exceeded
83h	00h	43536	Barcode label too short or too long
83h	01h	43537	Cannot read a barcode label due to scanner problem
83h	02h	43538	Barcode label questionable
83h	03h	43539	Cell status and barcode label questionable
83h	04h	43540	Drive not installed
83h	05h	43541	Drive varied off and not accessible for library operations
83h	06h	43542	Element contained within an offline tower or I/E station and is not accessible for library operations
83h	07h	43543	Tower is offline
83h	08h	43544	Tower transitioned to online
83h	09h	43545	Drive is offline
83h	0Ah	43546	Drive transitioned to online
83h	0Bh	43547	Drive enabled library managed encryption
83h	0Ch	43548	Drive enabled application managed encryption
83h	0Dh	43549	Drive assigned valid Ethernet IP address
83h	0Eh	43550	Drive disabled encryption