



Spectra Stack Tape Library

REST API

Document Information

Part Number:

909700046

Revision History:

Rev. A September 2024 Updated for Stack 2.50

Chapter 1. Overview

A simple API to manage the Scalable Tape Libraries remotely over a HTTPS interface. This is requested and needed for manufacturing and for automated test / monitoring systems.

1.1. Version information

Version : 1.1.22

1.2. Contact information

Contact : BDT Storage GmbH

1.3. URI scheme

BasePath : /rest

Schemes : HTTPS

1.4. Tags

- System : System requests
- Library Information : Library Status / Information
- Library Operations : Library Operations
- Library Configuration : Library Configuration
- Library Test : Library Tests
- Drive Information : Drive Information
- Drive Configuration : Drive Configuration
- Partition Information : Partition Information
- Partition Configuration : Partition Configuration
- MLM : MLM Section
- Service : Service Request

1.5. Consumes

- `application/json`

1.6. Produces

- `application/json`

1.7. General REST API Info

Name	Value
Protocol	HTTPS
Port	3031
Authorization	Bearer token
Base Path	/rest/
Certificate	Self-Signed
Example URL	https://libraryip:3031/rest/library/baseinfo

1.8. How To Use the REST API

The REST API workflow is as follows:

1. Login to the system with the **POST** command URL <https://libraryip:3031/rest/login>, where the username and password must be sent as content in JSON format (**application/json**).

Example Content:

```
{  
  "username": "administrator",  
  "password": "password"  
}
```

2. If the login was successful a Bearer token will be returned like: "Bearer eyJ0eXAiOiJKV1QiLCJhb....zRcVud3c"
3. The returned Bearer token must be added as the "Authorization" header value to the REST calls which have a security check.
Authorization: Bearer token

NOTE: If the token is invalid or the login was invalid a 401 HTTP error will be returned.

CURL Examples:

```
curl -X GET "https://libraryip:3031/rest/library/status" -H "accept: application/json" -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhb....zRcVud3c" -k
```

```
curl -X POST "https://libraryip:3031/rest/library/io/open" -H "accept: application/json" -H "Content-Type: application/json" -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhb....zRcVud3c" -d '{"module": 1}' -k
```

Chapter 2. Resources

2.1. System

System requests

2.1.1. Get the current REST API version

```
GET /apiversion
```

Description

The REST API version used in this software release

Responses

HTTP Code	Description	Schema
200	OK	GetApiVersion
500	Error occurred at operation	Error

GetApiVersion

Name	Schema
version <i>required</i>	string

Example HTTP response

Response 200

```
{
  "version" : "1.1.10"
}
```

2.1.2. Login Request

```
POST /login
```

Description

Required Initial request to get authenticated against the tape library. The user roles/names of the library are used. This API command returns after a successful authentication an session based Bearer token, which has to be attached in the "Authorization" header of every following API command during a session.

**Example using curl:* " `curl -X GET 'https://<library-ip>:3031/rest/library/baseinfo' -H 'accept: application/json' -H 'Authorization: Bearer token'` "

Parameters

Type	Name	Description	Schema
Body	user <i>required</i>	the user/password which should be used to log in. For some product variants in case of service level login the service password (service_password) and additionally the administrator password (password) have to be sent. In case the administrator password is not required the service password has to be set through the normal password field (password). See <i>Product Variants</i> notes.	PostAuth

PostAuth

Name	Description	Schema
username <i>required</i>	Example : "administrator"	string
password <i>required</i>	Example : "password"	string
service_password <i>optional</i>	service password in case the administrator password is additionally required Example : "servicepwd"	string
totp <i>optional</i>	Two factor authentication TOTP must be sent if user has TOTP active. Example : "123456"	string

Responses

Responses

HTTP Code	Description	Schema
200	OK	LibraryInfo
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
{
  "BaseInfo" : {
    "SerialNumber" : "DE00000000",
    "MacAddress_1" : "00:d0:93:3c:76:ee",
    "MacAddress_2" : "",
    "Vendor" : "VENDOR",
    "ProductID" : "ProductID",
    "BaseFWRevision" : "SSx1",
    "BaseFWBuildDate" : "2017-02-20",
    "ExpansionFWRevision" : "0.20",
    "WWNodeName" : "5001000000000000",
    "RoboticHWRevision" : "4",
    "RoboticFWRevision" : "0.10",
    "RoboticSerialNumber" : "564EA000103",
    "NoOfModules" : "7",
    "LibraryType" : "32"
  },
  "ModulesInfo" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400022"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "2",
```



```

    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400041"
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "3",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400040"
  }, {
    "PhysicalNumber" : "4",
    "LogicalNumber" : "4",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : ""
  }, {
    "PhysicalNumber" : "5",
    "LogicalNumber" : "5",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400045"
  }, {
    "PhysicalNumber" : "6",
    "LogicalNumber" : "6",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE56400042"
  }, {
    "PhysicalNumber" : "7",
    "LogicalNumber" : "7",
    "ReadyStatus" : "TRUE",
    "SerialNumber" : "DE564000XX"
  } ]
}

```

2.2.2. Get list of recent events

```
GET /library/getevents
```

Description

Based on the "EventType" this request returns a list of the recent events. The event types are

"Ticket"

"Info"

"Config"

"Service"

The number of returns can be limited by the parameter "MaxNum". To include already closed Tickets add "IncClosed" param.

Parameters

Type	Name	Description	Schema
Query	EventType <i>required</i>	The type of event	enum (Ticket, Info, Config, Service)
Query	MaxNum <i>optional</i>	The maximum number of returned events. If parameter is not given, all events available will be returned	integer
Query	IncClosed <i>optional</i>	The closed state of the ticket.	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	OK	< EventEntry > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
[ {
  "Number" : "33",
  "TimeStamp" : "2017_03_23T15:22:34:636Z",
  "EventCode" : "9003",
  "Text" : "Move media command was executed",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "32",
  "TimeStamp" : "2017_03_23T14:53:53:629Z",
  "EventCode" : "9028",
  "Text" : "Configuration backup to base module was successful",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "31",
  "TimeStamp" : "2017_03_23T14:33:05:235Z",
  "EventCode" : "9026",
  "Text" : "Library test successfully finished",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "30",
  "TimeStamp" : "2017_03_23T14:33:04:874Z",
  "EventCode" : "9003",
  "Text" : "Move media command was executed",
  "State" : "TICKET_PENDING",
  "Closed" : "FALSE"
}, {
  "Number" : "29",
  "TimeStamp" : "2017_03_23T14:32:18:319Z",
  "EventCode" : "9003",
  "Text" : "Move media command was executed",
  "State" : "TICKET_RESOLVED",
  "Closed" : "FALSE"
} ]
```

2.2.3. Inventory of physical library

```
GET /library/inventory
```

Description

Returns the inventory of the physical library including slots, drives and mailslots. It also returns the corresponding metadata which belongs to the different types of the elements.

Responses

HTTP Code	Description	Schema
200	OK	Inventory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
{
  "Slots" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1.1",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : "TRUE",
    "Blocked" : "TRUE"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "1.2",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "CartridgeSubType" : 0,
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  }, {
    "PhysicalNumber" : "3",
```

```

"LogicalNumber" : "1.3",
"Module" : "1",
"Partition" : "1",
"Mailslot" : "FALSE",
"Cartridge" : "TRUE",
"Barcode" : "TT0013L4",
"CartridgeType" : "Data",
"CartridgeSubType" : 0,
"CartridgeGeneration" : "4",
"CartridgeEncrypted" : "FALSE",
"Access" : "TRUE",
"Blocked" : "FALSE"
}, { }, {
  "PhysicalNumber" : "280",
  "LogicalNumber" : "7.40",
  "Module" : "7",
  "Partition" : "0",
  "Mailslot" : "TRUE",
  "Cartridge" : "FALSE",
  "CartridgeType" : "N/A",
  "CartridgeSubType" : 0,
  "Access" : "TRUE",
  "Blocked" : "FALSE"
} ],
"Drives" : [ {
  "PhysicalNumber" : "10",
  "LogicalNumber" : "1",
  "Partition" : "0",
  "Vendor" : "VENDOR ",
  "Product" : "Ultrium 7-SCSI ",
  "FWRevision" : "ABCD",
  "SerialNumber" : 1234567890
}, {
  "PhysicalNumber" : "11",
  "LogicalNumber" : "2",
  "Partition" : "0",
  "Vendor" : "VENDOR ",
  "Product" : "Ultrium 7-SCSI ",
  "FWRevision" : "ABCD",
  "SerialNumber" : 1234567891
}, {
  "PhysicalNumber" : "12",
  "LogicalNumber" : "3",
  "Partition" : "0",
  "Vendor" : "VENDOR ",
  "Product" : "Ultrium 7-SCSI ",
  "FWRevision" : "ABCD",
  "SerialNumber" : 1234567892
}, {
  "PhysicalNumber" : "20",
  "LogicalNumber" : "4",

```

```
"Partition" : "0",
"Vendor" : "",
"Product" : "",
"FWRevision" : "",
"SerialNumber" : ""
} ]
}
```

2.2.4. Mailslot Status

```
GET /library/io/status
```

Description

Returns the status of the mailslots

Responses

HTTP Code	Description	Schema
200	OK	IOStatus
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
[ {
  "ModuleNo" : "7",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "6",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "5",
  "Configured" : "TRUE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "4",
  "Configured" : "TRUE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "3",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "2",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
}, {
  "ModuleNo" : "1",
  "Configured" : "FALSE",
  "OpenStatus" : "FALSE",
  "Unlocked" : "FALSE"
} ]
```

2.2.5. License List

```
GET /library/license
```

Description

Returns a list of installed feature licenses

Responses

HTTP Code	Description	Schema
200	OK	< LicenseInfo > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.2.6. List of media

GET /library/mediainfo

Description

Returns the list of media which are located in the library including detailed information about them

Responses

HTTP Code	Description	Schema
200	OK	< MediaInfoData > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
[ {
  "Barcode" : "TC070ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "2.34",
  "PhysicalNumber" : "74",
  "Cleaning" : "FALSE",
  "Partition" : "1",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "FALSE",
  "Encryption" : "FALSE",
  "NoLoads" : "148",
  "MBRead" : "153",
  "MBReadLoad" : "150",
  "MBWritten" : "226",
  "MBWrittenLoad" : "226"
}, {
  "Barcode" : "000113L5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.4",
  "PhysicalNumber" : "84",
  "Cleaning" : "FALSE",
  "Partition" : "1",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "FALSE",
  "Encryption" : "FALSE",
  "NoLoads" : "11",
  "MBRead" : "0",
  "MBReadLoad" : "0",
  "MBWritten" : "14",
  "MBWrittenLoad" : "14"
}, {
  "Barcode" : "TC084ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.28",
  "PhysicalNumber" : "108",
  "Cleaning" : "FALSE",
  "Partition" : "2",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "TRUE",
  "Encryption" : "FALSE",
  "NoLoads" : "64",
  "MBRead" : "1643",
  "MBReadLoad" : "1603",
  "MBWritten" : "943",
  "MBWrittenLoad" : "943"
} ]
```

2.2.7. Get physical library status

```
GET /library/status
```

Description

The "status" command returns the live status of the physical library like health, number of moves and other live data

Responses

HTTP Code	Description	Schema
200	OK	LibraryStatus
401	Access denied - not authenticated	No Content
403	Access forbidden – insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```

{
  "BaseStatus" : {
    "Information" : "Idle (Online)",
    "RobStatus" : "Idle",
    "MoveCount" : "1",
    "PowerUpCount" : "10",
    "PowerOnTime" : "16 days, 3 hours, 2 minutes",
    "LibHealth" : "WARNING"
  },
  "ModulesStatus" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "2",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "3",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "4",
    "LogicalNumber" : "4",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "5",
    "LogicalNumber" : "5",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "6",
    "LogicalNumber" : "6",
    "Health" : "OK"
  }, {
    "PhysicalNumber" : "7",
    "LogicalNumber" : "7",
    "Health" : "OK"
  } ]
}

```

2.3. Library Operations

Library Operations

2.3.1. Inventory scan

POST /library/inventory

Description

Executes an Inventory scan on the library

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.3.2. Open Mailslot

```
POST /library/io/open
```

Description

This command sends a request to open a mailslot. The mailslot block which is addressed has to be configured as mailslot first. The mailslot stays open for a defined time and will be locked automatically after this time. The status of the mailslot can be retrieved by issuing the IO/status request

Parameters

Type	Name	Description	Schema
Body	Module <i>required</i>	the logical module number	PostLibraryIoOpen

PostLibraryIoOpen

Name	Description	Schema
module <i>required</i>	Minimum value : 1 Maximum value : 7 Example : 1	integer

Responses

HTTP Code	Description	Schema
200	The mailslot open was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.3.3. Move a cartridge

POST /library/movemedias

Description

A cartridge can be moved from a slot or drive to another slot or drive.

Parameters

Type	Name	Description	Schema
Body	Elements <i>required</i>	the source element from where a cartridge should be taken. It has the form <Element type> and <Address>. The type can be "Slot" or "Drive". Depending on the type the Address will have the form "Logical Slot Number (m.s)" or "Logical Drivenumber"	PostLibraryMoveMedia

PostLibraryMoveMedia

Name	Description	Schema
SrcType <i>required</i>	Example : "SLOT"	enum (DRIVE, SLOT)
SrcAddress <i>required</i>	Example : "1.2"	string
DestType <i>required</i>	Example : "DRIVE"	enum (DRIVE, SLOT)
DestAddress <i>required</i>	Example : "3"	string

Responses

HTTP Code	Description	Schema
200	The move was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.3.4. Shut down the library

```
POST /library/powerdown
```

Description

The library will be shut down. During shutdown the robotic will be secured at a safe transport position. The shutdown will be initiated after response to this command

Parameters

Type	Name	Schema
Body	RoboticPos <i>required</i>	PostLibraryPowerdown

PostLibraryPowerdown

Name	Description	Schema
RoboticPos <i>optional</i>	The position of the robotic after the shutdown. "SHIPPING_POS" moves the robotic to the standard shipping position in the base module and activates the shipping lock. "RACK_SHIPPING_POS" moves the robotic to the bottom of the library independent of the base module position (shipment in rack position). "PARK_POS" moves the robotic to the default park position. Example : "SHIPPING_POS"	enum (SHIPPING_POS, RACK_SHIPPING_POS, PARK_POS)

Responses

HTTP Code	Description	Schema
200	The shutdown was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.3.5. Reboot the library

POST /library/reboot

Description

The library will be rebooted. The shutdown will be initiated after response to this command. The connection will be lost until the library restarted successfully. A new login will be required

Responses

HTTP Code	Description	Schema
200	The reboot was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.4. Library Configuration

Library Configuration

2.4.1. Set Date and Time

```
POST /library/datetime
```

Description

With this command the Date and Time can be set in format "yyyy-MM-ddTHH:mm:ssZ" for e.g. 2021-07-21T17:32:28Z

NOTE | Setting a date time can invalidate the Bearer Token and a new login is required!

Parameters

Type	Name	Schema
Body	DateTime <i>required</i>	PostLibraryDateTime

PostLibraryDateTime

Name	Schema
dateTime <i>required</i>	string (date-time)

Responses

HTTP Code	Description	Schema
200	The date time config was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.4.2. Set SNTP for Date and Time

POST /library/datetime/sntp

Description

With this command the SNTP server can be set and the SNTP option can be enabled/disabled.

Parameters

Type	Name	Schema
Body	DateTimeSntp <i>required</i>	PostLibraryDateTimeSntp

PostLibraryDateTimeSntp

Name	Schema
server <i>required</i>	string

Name	Schema
enabled <i>required</i>	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The configuration was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.4.3. Set Timezone

POST /library/datetime/timezone

Description

With this command the timezone can be set like "America/Denver"

Parameters

Type	Name	Schema
Body	DateTimeZone <i>required</i>	PostLibraryDateTimeTimezone

PostLibraryDateTimeTimezone

Name	Description	Schema
timezone <i>required</i>	The timezone of the system	Timezones

Responses

HTTP Code	Description	Schema
200	The timezone config was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.4.4. Mailslot Operations

```
POST /library/io/config
```

Description

With this command the mailslot block in a magazine can be configured as mailslots or as standard storage slots. In order to be able to open mailslot slots of the magazine block they need to be configured as mailslots

Parameters

Type	Name	Description	Schema
Body	mailslotConf <i>required</i>	the logical module number and mode of mailslot. A TRUE will configure mailslot mode, a FALSE storage mode	PostLibraryIoConfig

PostLibraryIoConfig

Name	Description	Schema
module <i>required</i>	Minimum value : 1 Maximum value : 7	integer
mode <i>required</i>		enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The mailslot configuration was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.4.5. Import License key

```
POST /library/license
```

Description

With this command keys for licensed features can be imported and added to the system. The key inherits the feature and additional parameters which have to licensed

Parameters

Type	Name	Description	Schema
Body	LicenseKey <i>required</i>	the license key to add	PostLibraryLicense

PostLibraryLicense

Name	Description	Schema
licenseKey <i>required</i>	Minimum length : 15	string

Responses

HTTP Code	Description	Schema
200	The license key was added successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.4.6. Change MAC of the Library

POST /library/mac

Description

With this command the MACs can be set. After successful change the Library will be rebooted.

Parameters

Type	Name	Description	Schema
Body	setConfMac <i>required</i>	the new MACs to be set	PostLibraryMac

PostLibraryMac

Name	Description	Schema
mac0 <i>optional</i>	Length : 17	string
mac1 <i>optional</i>	Length : 17	string

Responses

HTTP Code	Description	Schema
200	The settings successfully changed.	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	ServiceSecurity

2.4.7. Change MAC and/or WWN of the Library

```
POST /library/mac_wn
```

Description

With this command the WWN (WWide Node Name) and the MACs can be configured. After successful change the Library will be rebooted.

Parameters

Type	Name	Description	Schema
Body	setConfMacWwn <i>required</i>	the new WWN and/or MACs to be set	PostLibraryMacWwn

PostLibraryMacWwn

Name	Description	Schema
wwn <i>optional</i>	Length : 16	string
mac0 <i>optional</i>	Length : 17	string

Name	Description	Schema
mac1 <i>optional</i>	Length : 17	string

Responses

HTTP Code	Description	Schema
200	The settings successfully changed.	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	ServiceSecurity

2.4.8. Reset the Configuration

POST /library/resetconfig

Description

After issuing this command the configuration of the library will be set to standard defaults or to manufacturing defaults depending on the ManufacturingMode parameter. The standard defaults will keep IP address, counters and manufacturing defaults will reset everything. In case of manual configured network the connection will possibly be lost. If no Mode parameter is submitted standard default mode will be used. Reset to Manufacturing Mode will also delete existing license keys in the system

Parameters

Type	Name	Schema
Body	ManufacturingMode <i>optional</i>	PostLibraryResetConfig

PostLibraryResetConfig

Name	Description	Schema
ManufacturingMode <i>optional</i>	The mode of reset. "TRUE" will do a full reset including IP settings and licenses Example : "FALSE"	enum (TRUE, FALSE)
RoboticPos <i>optional</i>	The position of the robotic after the shutdown."SHIPPING_POS" moves the robotic to the standard shipping position in the base module and activates the shipping lock. "RACK_SHIPPING_POS" moves the robotic to the bottom of the library independent of the base module position (shipment in rack position). "PARK_POS" moves the robotic to the default park position. Example : "SHIPPING_POS"	enum (SHIPPING_POS, RACK_SHIPPING_POS, PARK_POS)
PowerDown <i>optional</i>	"TRUE" powers the unit off after a reset. "FALSE" keeps default reboot behavior	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The reset configuration was executed successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity
apiKey	ServiceSecurity

2.4.9. Set library serial number

```
PUT /library/serialnum
```

Description

The library serial number can be set.

Note: After setting the serial number the library must be restarted for the changes to take effect.

Parameters

Type	Name	Schema
Body	SerialNumber <i>required</i>	PutLibrarySerialNumber

PutLibrarySerialNumber

Name	Description	Schema
serialnum <i>required</i>	Length : 10 - 14 Example : "DE1234567890"	string

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	ServiceSecurity

2.4.10. Upgrade Library SW

POST /library/swupgrade

Description

The command upgrades the SW on the library. It returns before the SW upgrade process is started.

After the upgrade a reboot will be initiated. The connection will be lost during this time frame

Parameters

Type	Name	Description	Schema
FormData	SWImage <i>required</i>	The SW upgrade file	file

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Consumes

- `multipart/form-data`

Security

Type	Name
apiKey	AdminSecurity

2.4.11. Change WWN of the Library

POST /library/wwn

Description

With this command the WWN (WWide Node Name) can be set. Format is 16 hex characters [0-9 A-F]. After successful change the Library will be rebooted.

Parameters

Type	Name	Description	Schema
Body	setConfWwn <i>required</i>	the new WWN to be set	PostLibraryWwn

PostLibraryWwn

Name	Description	Schema
wwn <i>required</i>	Length : 16	string

Responses

HTTP Code	Description	Schema
200	The WWN was successfully changed.	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	ServiceSecurity

2.5. Library Test

Library Tests

2.5.1. Slot to Slot Test

POST /test/slot2slot

Description

Performs the slot to slot test n times. The number of executions has to be defined in the count parameter. The command returns immediately after starting the test. The progress has to be polled by the /test/status request

Parameters

Type	Name	Description	Schema
Body	count <i>required</i>	the number of test iterations to perform	PostTestSlotToSlot

PostTestSlotToSlot

Name	Description	Schema
count <i>required</i>	Maximum value : 100 Example : 10	integer

Responses

HTTP Code	Description	Schema
200	The test could be started successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.5.2. Test Status

GET /test/status

Description

Returns the status of a test which is currently executed. A test run always asynchronously, a response is sent immediately after the start. This function is used to monitor the status of the test, it returns the current test cycle and the run status.

Responses

HTTP Code	Description	Schema
200	Status of current test	TestStatus
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

Example HTTP response

Response 200

```
{
  "currentCycle" : "2",
  "testCycles" : "10",
  "status" : "TRUE",
  "running" : "FALSE"
}
```

2.5.3. System (Demo) Test

POST /test/systemtest

Description

Performs the integrated system (Demo) test n times. The number of executions has to be defined in the count parameter. The command returns immediately after starting the test. The progress has to be polled by the /test/status request

Parameters

Type	Name	Description	Schema
Body	testParameter <i>required</i>	the number of test iterations to perform and seating mode	PostTestSystemTest

PostTestSystemTest

Name	Description	Schema
count <i>required</i>	Maximum value : 100 Example : 10	integer
seating <i>optional</i>		enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	The test could be started successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.6. Drive Information

Drive Information

2.6.1. Get drive information

```
GET /drive/information
```

Description

Returns the information about drives like Product Id, Model, Serial number and other useful information.

Responses

HTTP Code	Description	Schema
200	OK	< DriveInfo > array

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
{
  "LogicalNumber" : "1",
  "PhysicalNumber" : "10",
  "Module" : "1",
  "Partition" : "0",
  "Generation" : "8",
  "Cartridge" : "FALSE",
  "Vendor" : "IBM",
  "Product" : "ULT3580-HH8",
  "FWRevision" : "J28F",
  "SerialNumber" : "000000005B",
  "WWNodeName" : "",
  "Interface" : "FC",
  "MFGSerialNumber" : "1013000188",
  "ErrorState" : "TRUE",
  "Power" : "FALSE",
  "Presence" : "TRUE",
  "ADTMode" : "IADT"
}
```

2.7. Drive Configuration

Drive Configuration

2.7.1. Upgrade Drive FW

```
POST /drive/fwupgrade
```


Description

The command upgrades the FW of the specified drives in the library.

Parameters

Type	Name	Description	Schema
FormData	FWImage <i>required</i>	The FW upgrade file	file
FormData	drivenumber <i>required</i>	The logical drive number(s), multiple drives possible as array (comma separated values)	< integer > array(csv)

Responses

HTTP Code	Description	Schema
200	The FW upgrade was successful	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Consumes

- `multipart/form-data`

Security

Type	Name
apiKey	AdminSecurity

2.7.2. Drive Reset

POST /drive/reset

Description

The command performs a drive reset

Parameters

Type	Name	Description	Schema
Body	DriveNumber <i>required</i>	The logical number of the drive to be reset	PostDriveReset

PostDriveReset

Name	Description	Schema
DriveNumber <i>required</i>	Minimum value : 1 Maximum value : 21 Example : 3	integer

Responses

HTTP Code	Description	Schema
200	The drive was reset successfully	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	AdminSecurity

2.8. Partition Information

Partition Information

2.8.1. Get partition (Logical Library) information

```
GET /partition/information
```

Description

The command returns the list of partitions including partition details

Responses

HTTP Code	Description	Schema
200	OK	< PartitionInfo > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
[ {
  "PartitionNumber" : "1",
  "Name" : "New Partition_1",
  "SerialNumber" : "DE00000000_LL01",
  "NumSlots" : "87",
  "NumIOSlots" : "0",
  "NumDrives" : "1",
  "LunMasterDrive" : "1",
  "LunMasterDrivePhys" : "10",
  "LunMasterDriveArr" : [ "1" ],
  "LunMasterDrivePhysArr" : [ "10" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : "8",
  "AutoClean" : "TRUE",
  "WWNode" : "50010000000000071",
  "Micw" : "FALSE"
}, {
  "PartitionNumber" : "2",
  "Name" : "New Partition_2",
  "SerialNumber" : "DE00000000_LL02",
  "NumSlots" : "90",
  "NumIOSlots" : "5",
  "NumDrives" : "1",
```

```

"LunMasterDrive" : "2",
"LunMasterDrivePhys" : "11",
"LunMasterDriveArr" : [ "2" ],
"LunMasterDrivePhysArr" : [ "11" ],
"EncryptionMode" : "ISV",
"BarcodeAlign" : "left",
"BarcodeLength" : "8",
"AutoClean" : "TRUE",
"WWNode" : "5001000000000081",
"Micw" : "FALSE"
}, {
  "PartitionNumber" : "3",
  "Name" : "New Partition_3",
  "SerialNumber" : "DE00000000_LL03",
  "NumSlots" : "85",
  "NumIOSlots" : "5",
  "NumDrives" : "1",
  "LunMasterDrive" : "3",
  "LunMasterDrivePhys" : "12",
  "LunMasterDriveArr" : [ "3" ],
  "LunMasterDrivePhysArr" : [ "12" ],
  "EncryptionMode" : "ISV",
  "BarcodeAlign" : "left",
  "BarcodeLength" : "8",
  "AutoClean" : "TRUE",
  "WWNode" : "5001000000000091",
  "Micw" : "FALSE"
} ]

```

2.8.2. Inventory of a partition

```
GET /partition/inventory
```

Description

Returns the inventory of the selected partition including slots, drives and mailslots. It also returns the corresponding metadata which belongs to the different types of the elements

Parameters

Type	Name	Description	Schema
Query	partitionNum <i>required</i>	The partition number for which the inventory should be reported	integer

Responses

HTTP Code	Description	Schema
200	OK	Inventory
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
{
  "Slots" : [ {
    "PhysicalNumber" : "1",
    "LogicalNumber" : "1.1",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "Access" : "TRUE",
    "Blocked" : "TRUE"
  }, {
    "PhysicalNumber" : "2",
    "LogicalNumber" : "1.2",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
    "Cartridge" : "FALSE",
    "CartridgeType" : "N/A",
    "Access" : "TRUE",
    "Blocked" : "FALSE"
  }, {
    "PhysicalNumber" : "3",
    "LogicalNumber" : "1.3",
    "Module" : "1",
    "Partition" : "1",
    "Mailslot" : "FALSE",
```

```

"Cartridge" : "TRUE",
"Barcode" : "TT0013L4",
"CartridgeType" : "Data",
"CartridgeSubType" : 0,
"CartridgeGeneration" : "4",
"CartridgeEncrypted" : "FALSE",
"Access" : "TRUE",
"Blocked" : "FALSE"
}, { }, {
  "PhysicalNumber" : "280",
  "LogicalNumber" : "7.40",
  "Module" : "7",
  "Partition" : "0",
  "Mailslot" : "TRUE",
  "Cartridge" : "FALSE",
  "CartridgeType" : "N/A",
  "Access" : "TRUE",
  "Blocked" : "FALSE"
} ],
"Drives" : [ {
  "PhysicalNumber" : "10",
  "LogicalNumber" : "1",
  "Partition" : "0",
  "Vendor" : "VENDOR ",
  "Product" : "Ultrium 7-SCSI ",
  "FWRevision" : "ABCD",
  "SerialNumber" : 1234567890
}, {
  "PhysicalNumber" : "11",
  "LogicalNumber" : "2",
  "Partition" : "0",
  "Vendor" : "VENDOR ",
  "Product" : "Ultrium 7-SCSI ",
  "FWRevision" : "ABCD",
  "SerialNumber" : 1234567891
}, {
  "PhysicalNumber" : "12",
  "LogicalNumber" : "3",
  "Partition" : "0",
  "Vendor" : "VENDOR ",
  "Product" : "Ultrium 7-SCSI ",
  "FWRevision" : "ABCD",
  "SerialNumber" : 1234567892
}, {
  "PhysicalNumber" : "20",
  "LogicalNumber" : "4",
  "Partition" : "0",
  "Vendor" : "",
  "Product" : "",
  "FWRevision" : "",
  "SerialNumber" : ""
} ]

```

```
} ]  
}
```

2.8.3. List of media of a partition

```
GET /partition/mediainfo
```

Description

Returns the list of media which are assigned to the partition including detailed information about them

Parameters

Type	Name	Description	Schema
Query	partitionNum <i>required</i>	The partition number for which the media information should be reported	integer

Responses

HTTP Code	Description	Schema
200	OK	< MediaInfoData > array
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Security

Type	Name
apiKey	UserSecurity

Example HTTP response

Response 200

```
[ {
  "Barcode" : "TD239ML4",
  "LocationType" : "SLOT",
  "LogicalNumber" : "3.35",
  "PhysicalNumber" : "115",
  "Cleaning" : "FALSE",
  "Partition" : "2",
  "Generation" : "4",
  "SubType" : "0",
  "Protection" : "FALSE",
  "Encryption" : "FALSE",
  "NoLoads" : "0",
  "MBRead" : "0",
  "MBReadLoad" : "0",
  "MBWritten" : "0",
  "MBWrittenLoad" : "0"
}, {
  "Barcode" : "TC084ML5",
  "LocationType" : "SLOT",
  "LogicalNumber" : "5.10",
  "PhysicalNumber" : "170",
  "Cleaning" : "FALSE",
  "Partition" : "2",
  "Generation" : "5",
  "SubType" : "0",
  "Protection" : "TRUE",
  "Encryption" : "FALSE",
  "NoLoads" : "64",
  "MBRead" : "1643",
  "MBReadLoad" : "1603",
  "MBWritten" : "943",
  "MBWrittenLoad" : "943"
} ]
```

2.9. Partition Configuration

Partition Configuration

2.9.1. Create partitions in simple mode

```
POST /partition/createsimple
```

Description

Creates the number of partitions which are requested in the parameters. This command corresponds to the Basic Wizard in RMI to create partitions. The maximum number of partitions which can be created are limited by the number of installed drives.

Parameters

Type	Name	Description	Schema
Body	partitionParameters <i>required</i>	the parameters to create the partitions	PostPartitionCreateSimple

PostPartitionCreateSimple

Name	Description	Schema
numPartitions <i>required</i>	The number of partitions to create Minimum value : 1 Maximum value : 21 Example : 1	integer
barcodeLength <i>optional</i>	Minimum value : 6 Maximum value : 16 Example : 8	integer
barcodeAlignment <i>optional</i>	Alignment of the barcode for SCSI element status. Can be "left" or "right" Default : "left"	enum (left, right)
autoClean <i>optional</i>	Auto Cleaning for this partition. 'TRUE' for enabling 'FALSE' to disable it. Note: <i>At some product variants the default value is 'TRUE'</i> Default : "FALSE"	enum (TRUE, FALSE)
sequentialMode <i>optional</i>	Enable Sequential Mode for this partition. 'TRUE' for enabling 'FALSE' to disable it. Note: <i>Only applicable for some variants</i> Default : "FALSE"	enum (TRUE, FALSE)
sequentialModeLoop <i>optional</i>	Restarting sequence when finished. Requires sequentialMode to be 'TRUE' Note: <i>Only applicable for some variants</i> Default : "FALSE"	enum (TRUE, FALSE)

Name	Description	Schema
sequentialModeAutoload <i>optional</i>	Load first cartridge automatically in sequential mode. Requires sequentialMode to be 'TRUE' Note: <i>Only applicable for some variants</i> Default : "FALSE"	enum (TRUE, FALSE)

Responses

HTTP Code	Description	Schema
200	OK	No Content
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Consumes

- `application/json`

Security

Type	Name
apiKey	AdminSecurity

2.10. MLM

MLM Section

2.10.1. Save MLM Database

```
GET /mlm/database
```

Description

Save MLM Database from the stack in zip format returned as "application/octet-stream".

Responses

HTTP Code	Description	Schema
200	OK	file (binary)
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content
500	Error occurred at operation	Error

Produces

- `application/octet-stream`

Security

Type	Name
apiKey	AdminSecurity

2.11. Service

Service Request

2.11.1. Get library logs dump

```
GET /library/logs
```

Description

Generates and returns a log file from the stack in binary format "application/octet-stream". Other responses are returned as "application/json"

Responses

HTTP Code	Description	Schema
200	Log file (zipped)	file (binary)
401	Access denied - not authenticated	No Content
403	Access forbidden - insufficient rights	No Content

HTTP Code	Description	Schema
500	Error occurred at operation	Error

Produces

- `application/octet-stream`
- `application/json`

Security

Type	Name
apiKey	AdminSecurity

Chapter 3. Definitions

3.1. BaseInfoData

The definition of the physical base module information

Name	Description	Schema
SerialNumber <i>optional</i>	The serial number for the complete library/stack which is also reported through SCSI	string
MacAddress_1 <i>optional</i>	The MAC address of the Ethernet port 1	string
MacAddress_2 <i>optional</i>	The MAC address of the Ethernet port 2	string
Vendor <i>optional</i>	The product vendor name	string
ProductID <i>optional</i>	The Product ID	string
BaseFWRevision <i>optional</i>	Firmware Revision of the base library	string
BaseFWBuildDate <i>optional</i>	Build date of base library firmware YYYY-MM-DD	string (date)
ExpansionFWRevision <i>optional</i>	Firmware Revision of the expansion modules	string
WWNodeName <i>optional</i>	World Wide Node Name of the library	string
RoboticHWRevision <i>optional</i>	Hardware Revision Level of the Robotic	string

Name	Description	Schema
RoboticFWRevision <i>optional</i>	Firmware Revision Level of the Robotic	string
RoboticSerialNumber <i>optional</i>	Internal serial number of the Robotic	string
NoOfModules <i>optional</i>	Number of connected expansion modules	integer
LibraryType <i>optional</i>	Library Type (32/40) (Product specific)	integer

3.2. BaseStatusData

Overview status data of the library

Name	Description	Schema
Information <i>optional</i>	Activity Status of the library	string
RobStatus <i>optional</i>	Activity Status of the Robotic	enum (Idle, Moving)
MoveCount <i>optional</i>	Number of moves over lifetime	integer
PowerUpCount <i>optional</i>	Number of power cycles over lifetime	integer
PowerOnTime <i>optional</i>	Power up time over lifetime	string
LibHealth <i>optional</i>	Health Status of library (Summary)	string

3.3. DriveInfo

Name	Description	Schema
LogicalNumber <i>optional</i>	The logical drive number starts always at 1 with the first inserted drive in the lowest available module	integer
PhysicalNumber <i>required</i>	The physical drive number starting at 1. Formula is (ModuleNo - 1) * (total Drive Slots in single Module) + (DriveSlotNumber in Module)	integer
Module <i>optional</i>	The Expansion Module where the drive is located in	integer
Partition <i>optional</i>	The partition (logical library) where the drive is assigned to. If no partition is assigned the value will be 0	integer
Generation <i>optional</i>	LTO Generation of Drive	integer
Cartridge <i>optional</i>	TRUE - Cartridge is loaded / FALSE - no Cartridge loaded	enum (TRUE, FALSE)
Barcode <i>optional</i>	The Barcode Label of the cartridge which is loaded in the drive	string
Vendor <i>optional</i>	The drive vendor name	string
Product <i>optional</i>	The Product ID of the drive	string
FWRevision <i>optional</i>	The FW revision of the drive	string
SerialNumber <i>optional</i>	The serial number of the drive (Spoofed)	string
WWNodeName <i>optional</i>	The World Wide Node Name of the drive	string
Interface <i>optional</i>	Interface Type - SAS / Fibre Channel	string

Name	Description	Schema
MFGSerialNumber <i>optional</i>	The Manufacturing Serial number	string
ErrorState <i>optional</i>	Is Drive in an error state	enum (TRUE, FALSE)
Power <i>optional</i>	Is drive powered	enum (TRUE, FALSE)
Presence <i>optional</i>	Is drive present	enum (TRUE, FALSE)
ADTMode <i>optional</i>	ADT Transport Mode	string

3.4. DriveInfoInventory

Name	Description	Schema
PhysicalNumber <i>required</i>	The physical drive number starting at 1. Formula is (ModuleNo - 1) * (total Drive Slots in single Module) + (DriveSlotNumber in Module)	integer
LogicalNumber <i>optional</i>	The logical drive number starts always at 1 with the first inserted drive in the lowest available module	integer
Module <i>optional</i>	The Expansion Module where the drive is located in	integer
Partition <i>optional</i>	The partition (logical library) where the drive is assigned to. If no partition is assigned the value will be 0	integer
Barcode <i>optional</i>	The Barcode Label of the cartridge which is loaded in the drive	string
Vendor <i>optional</i>	The drive vendor name	string
Product <i>optional</i>	The Product ID of the drive	string

Name	Description	Schema
FWRevision <i>optional</i>	The FW revision of the drive	string
SerialNumber <i>optional</i>	The serial number of the drive (Spoofed)	string

3.5. Error

Name	Schema
code <i>optional</i>	string
message <i>optional</i>	string

3.6. EventEntry

Structure of event entry

Name	Description	Schema
Number <i>required</i>	Number of event in list	integer
TimeStamp <i>optional</i>	Date/ Time of event	string
EventCode <i>optional</i>	Eventcode, describing the event which occurred	integer
Text <i>optional</i>	Short description of EventCode	string
State <i>optional</i>	Shows the current state for the ticket.	enum (INFO, CONFIG, SERVICE, TICKET_PENDING, TICKET_RESOLVED, SINGULAR_TICKET)
Closed <i>optional</i>	Reflects the closed state for tickets	enum (TRUE, FALSE)

3.7. IOStatus

IO Station (Mailslot) status data

Type : < [IOStatus](#) > array

IOStatus

Name	Description	Schema
ModuleNo <i>required</i>	logical module number where the IO Station is located	integer
Configured <i>optional</i>	FALSE - IO Station Slots are configured as storage slots. TRUE - IO Station Slots are configured for IO Operation	enum (TRUE, FALSE)
Unlocked <i>optional</i>	FALSE - IO Station is locked. TRUE - IO Station is unlocked	enum (TRUE, FALSE)
OpenStatus <i>optional</i>	FALSE - IO Station is closed. TRUE - IO Station is opened	enum (TRUE, FALSE)

3.8. Inventory

Name	Schema
Slots <i>optional</i>	< Slot > array
Drives <i>optional</i>	< DriveInfoInventory > array

3.9. LibraryInfo

The definition of the physical library information

Name	Description	Schema
BaseInfo <i>required</i>		BaseInfoData
ModulesInfo <i>required</i>	Module Information	< Module > array

3.10. LibraryStatus

Overview status information of the library

Name	Schema
BaseStatus <i>required</i>	BaseStatusData
ModulesStatus <i>required</i>	< ModuleStatusData > array

3.11. LicenseInfo

Licensed Feature data

Name	Description	Schema
Feature <i>required</i>	Name of licensed feature	string
LicenseKey <i>required</i>	License Key string	string

3.12. MediaInfoData

Information about a tape media.

Name	Description	Schema
Barcode <i>optional</i>	The Barcode Label of the media	string
LocationType <i>optional</i>	Type of location where the media is stored. "Drive" or "Slot"	enum (DRIVE, SLOT)
LogicalNumber <i>optional</i>	The logical number of media location	string
PhysicalNumber <i>optional</i>	The physical number of media location	string

Name	Description	Schema
Cleaning <i>optional</i>	TRUE - media is a cleaning tape / FALSE - media is not a cleaning tape	enum (FALSE, TRUE)
Partition <i>optional</i>	The partition (logical library) where the media is assigned to. If no partition is assigned the value will be 0	integer
Generation <i>optional</i>	LTO Generation of media	integer
SubType <i>optional</i>	LTO SubType of media (0 = default, 1= Type M)	integer
Protection <i>optional</i>	TRUE - media is write protected / FALSE - media is not protected	enum (FALSE, TRUE)
Encryption <i>optional</i>	TRUE - media is encrypted / FALSE - media is not encrypted / null if not identified by drive	enum (FALSE, TRUE)
NoLoads <i>optional</i>	Number of loads of media. Shows how many times the media was loaded into a tape drive	integer
MBRead <i>optional</i>	Bytes read from this media in MB	integer
MBReadLoad <i>optional</i>	Bytes read from this media in MB during last load	integer
MBWritten <i>optional</i>	Bytes written to this media in MB	integer
MBWrittenLoad <i>optional</i>	Bytes written to this media in MB during last load	integer

3.13. Module

The definition of an expansion module informational data

Name	Description	Schema
PhysicalNumber <i>required</i>	The physical number of the expansion module. the numbering starts always at 1 with the bottom module counting to 7 regardless of the physical presence of this module. The base module has always the number 4	integer
LogicalNumber <i>optional</i>	The number of the module in an existing library stack. The lowest module always starts with 1	integer
ReadyStatus <i>optional</i>	This entry returns the current status of the module. If it is correctly initialized and running it will be set to TRUE	enum (TRUE, FALSE)
SerialNumber <i>optional</i>	The internal manufacturing serial number of the module	string

3.14. ModuleStatusData

Current Health Status of a module

Name	Description	Schema
PhysicalNumber <i>required</i>	Module Number (physically)	integer
LogicalNumber <i>optional</i>	Module Number (logical)	integer
Health <i>optional</i>	Current health status of Module	string

3.15. Modules

List of expansion module data

Type : < [Module](#) > array

3.16. PartitionInfo

Partition data

Name	Description	Schema
PartitionNumber <i>required</i>	Number of partition	integer
Name <i>optional</i>	Name of partition	string
SerialNumber <i>optional</i>	Serialnumber of partition, based on serialnumber of physical library	string
NumSlots <i>optional</i>	Number of slots assigned to partition	integer
NumIOSlots <i>optional</i>	Number of IO slots (Mailslots) assigned to partition	integer
NumDrives <i>optional</i>	Number of drives assigned to partition	integer
LunMasterDrive <i>optional</i>	Lun Master Drive number (logical)	integer
LunMasterDrivePhys <i>optional</i>	Lun Master Drive number (physical)	integer
LunMasterDriveArr <i>optional</i>	Lun Master Drive number (logical)	< integer > array
LunMasterDrivePhysArr <i>optional</i>	Lun Master Drive number (physical)	< integer > array

Name	Description	Schema
EncryptionMode <i>optional</i>	Encryption Mode which is currently active for this partition. The following values are possible <ul style="list-style-type: none"> • OFF - No encryption policy activated • ISV - Encryption can be controlled by host application • KMIP - KMIP Key Management is activated. A KMIP Key Server is required • ESKM - ESKM Key Management is activated. A ESKM Server is required • PLK - Local Key Management over a PLK Token is active. A PLK token is required 	enum (OFF, ISV, KMIP, ESKM, PLK, IPP)
BarcodeAlign <i>optional</i>	Barcode Label Alignment (left,right)	enum (left, right)
BarcodeLength <i>optional</i>	Barcode Label Length	integer
AutoClean <i>optional</i>	AutoClean enabled	enum (TRUE, FALSE)
WWNode <i>optional</i>	WWN of Partition	string
Micw <i>optional</i>	Multi-initiator conflict warning enabled	enum (TRUE, FALSE)

3.17. Slot

The definition of cartridge slot data

Name	Description	Schema
PhysicalNumber <i>required</i>	The physical slot number starting at 1. Formula is (ModuleNo - 1) * (total Slots in single Module) + (SlotNumber in Module)	integer
LogicalNumber <i>optional</i>	The logical slot number includes the module number as prefix and starts always at 1 in a module. Format - ModuleNo.(SlotNumber in Module)	string

Name	Description	Schema
Module <i>optional</i>	The Expansion Module where the slot is located in	integer
Partition <i>optional</i>	The partition (logical library) where the slot is assigned to. If no partition is assigned the value will be 0	integer
Mailslot <i>optional</i>	If the slot is mapped as a mailslot (I/O Station) this entry is set to TRUE	enum (TRUE, FALSE)
Cartridge <i>optional</i>	If a cartridge is detected in the slot this entry is set to TRUE	enum (TRUE, FALSE)
CartridgeType <i>optional</i>	Type of cartridge Data,Cleaning,Worm etc.	enum (Unknown, Data, Cleaning, Worm, N/A)
CartridgeSub Type <i>optional</i>	SubType of a cartridge 0 = Default, 1= Type M	integer
CartridgeGeneration <i>optional</i>	Generation of the cartridge	integer
CartridgeEncrypted <i>optional</i>	Cartridge encryption state	enum (TRUE, FALSE)
Barcode <i>optional</i>	The Barcode Label of the cartridge which is sitting in the slot	string
Access <i>optional</i>	If the slot is accessible by the system the entry is set to TRUE	enum (TRUE, FALSE)
Blocked <i>optional</i>	TRUE - The slot is not reachable by the robot / FALSE - Slot is reachable by the robot Default : "FALSE"	enum (TRUE, FALSE)

3.18. TestStatus

Status of current running test

Name	Description	Schema
currentCycle <i>optional</i>	current test cycle number	integer
testCycles <i>optional</i>	number of test cycles planned	integer
status <i>optional</i>	status of the test like: not started, running, passed, failed or stopped by user request.	string
running <i>optional</i>	TRUE - test is still running, FALSE - no test running	enum (TRUE, FALSE)

3.19. Timezones

The available timezones in the system

Type : enum (Africa/Abidjan, Africa/Accra, Africa/Addis_Ababa, Africa/Algiers, Africa/Asmara, Africa/Bamako, Africa/Bangui, Africa/Banjul, Africa/Bissau, Africa/Blantyre, Africa/Brazzaville, Africa/Bujumbura, Africa/Cairo, Africa/Casablanca, Africa/Ceuta, Africa/Conakry, Africa/Dakar, Africa/Dar_es_Salaam, Africa/Djibouti, Africa/Douala, Africa/El_Aaiun, Africa/Freetown, Africa/Gaborone, Africa/Harare, Africa/Johannesburg, Africa/Juba, Africa/Kampala, Africa/Khartoum, Africa/Kigali, Africa/Kinshasa, Africa/Lagos, Africa/Libreville, Africa/Lome, Africa/Luanda, Africa/Lubumbashi, Africa/Lusaka, Africa/Malabo, Africa/Maputo, Africa/Maseru, Africa/Mbabane, Africa/Mogadishu, Africa/Monrovia, Africa/Nairobi, Africa/Ndjamena, Africa/Niamey, Africa/Nouakchott, Africa/Ouagadougou, Africa/Porto-Novo, Africa/Sao_Tome, Africa/Tripoli, Africa/Tunis, Africa/Windhoek, America/Adak, America/Anchorage, America/Anguilla, America/Antigua, America/Araguaina, America/Argentina/Buenos_Aires, America/Argentina/Catamarca, America/Argentina/Cordoba, America/Argentina/Jujuy, America/Argentina/La_Rioja, America/Argentina/Mendoza, America/Argentina/Rio_Gallegos, America/Argentina/Salta, America/Argentina/San_Juan, America/Argentina/San_Luis, America/Argentina/Tucuman, America/Argentina/Ushuaia, America/Aruba, America/Asuncion, America/Atikokan, America/Bahia, America/Bahia_Banderas, America/Barbados, America/Belem, America/Belize, America/Blanc-Sablon, America/Boa_Vista, America/Bogota, America/Boise, America/Cambridge_Bay, America/Campo_Grande, America/Cancun, America/Caracas, America/Cayenne, America/Cayman, America/Chicago, America/Chihuahua, America/Costa_Rica, America/Creston, America/Cuiaba, America/Curacao, America/Danmarkshavn, America/Dawson, America/Dawson_Creek, America/Denver, America/Detroit, America/Dominica, America/Edmonton, America/Eirunepe, America/El_Salvador, America/Fort_Nelson, America/Fortaleza, America/Glace_Bay, America/Goose_Bay, America/Grand_Turk, America/Grenada, America/Guadeloupe, America/Guatemala, America/Guayaquil, America/Guyana, America/Halifax, America/Havana, America/Hermosillo, America/Indiana/Indianapolis, America/Indiana/Knox, America/Indiana/Marengo, America/Indiana/Petersburg, America/Indiana/Tell_City, America/Indiana/Vevay, America/Indiana/Vincennes, America/Indiana/Winamac, America/Inuvik, America/Iqaluit, America/Jamaica, America/Juneau, America/Kentucky/Louisville, America/Kentucky/Monticello, America/Kralendijk, America/La_Paz, America/Lima,

America/Los_Angeles, America/Lower_Princes, America/Maceio, America/Managua,
America/Manaus, America/Marigot, America/Martinique, America/Matamoros, America/Mazatlan,
America/Menominee, America/Merida, America/Metlakatla, America/Mexico_City,
America/MiquelonF, America/Moncton, America/Monterrey, America/Montevideo,
America/Montserrat, America/Nassau, America/New_York, America/Nipigon, America/Nome,
America/Noronha, America/North_Dakota/Beulah, America/North_Dakota/Center,
America/North_Dakota/New_Salem, America/Nuuk, America/Ojinaga, America/Panama,
America/Pangnirtung, America/Paramaribo, America/Phoenix, America/Port-au-Prince,
America/Port_of_Spain, America/Porto_Velho, America/Puerto_Rico, America/Punta_Arenas,
America/Rainy_River, America/Rankin_Inlet, America/Recife, America/Regina, America/Resolute,
America/Rio_Branco, America/Santarem, America/Santiago, America/Santo_Domingo,
America/Sao_Paulo, America/Scoresbysund, America/Sitka, America/St_Barthelemy,
America/St_Johns, America/St_Kitts, America/St_Lucia, America/St_Thomas, America/St_Vincent,
America/Swift_Current, America/Tegucigalpa, America/Thule, America/Thunder_Bay,
America/Tijuana, America/Toronto, America/Tortola, America/Vancouver, America/Whitehorse,
America/Winnipeg, America/Yakutat, America/Yellowknife, Antarctica/Casey, Antarctica/Davis,
Antarctica/DumontDUrville, Antarctica/Macquarie, Antarctica/Mawson, Antarctica/McMurdo,
Antarctica/Palmer, Antarctica/Rothera, Antarctica/Syowa, Antarctica/Troll, Antarctica/Vostok,
Arctic/Longyearbyen, Asia/Aden, Asia/Almaty, Asia/Amman, Asia/Anadyr, Asia/Aqtau, Asia/Aqtobe,
Asia/Ashgabat, Asia/Atyrau, Asia/Baghdad, Asia/Bahrain, Asia/Baku, Asia/Bangkok, Asia/Barnaul,
Asia/Beirut, Asia/Bishkek, Asia/Brunei, Asia/Chita, Asia/Choibalsan, Asia/Colombo, Asia/Damascus,
Asia/Dhaka, Asia/Dili, Asia/Dubai, Asia/Dushanbe, Asia/Famagusta, Asia/Gaza, Asia/Hebron,
Asia/Ho_Chi_Min, Asia/Hong_Kong, Asia/Hovd, Asia/Irkutsk, Asia/Jakarta, Asia/Jayapura,
Asia/Jerusalem, Asia/Kabul, Asia/Kamchatka, Asia/Karachi, Asia/Kathmandu, Asia/Khandyga,
Asia/Kolkata, Asia/Krasnoyarsk, Asia/Kuala_Lumpur, Asia/Kuching, Asia/Kuwait, Asia/Macau,
Asia/Magadan, Asia/Makassar, Asia/Manila, Asia/Muscat, Asia/Nicosia, Asia/Novokuznetsk,
Asia/Novosibirsk, Asia/Omsk, Asia/Oral, Asia/Phnom_Penh, Asia/Pontianak, Asia/Pyongyang,
Asia/Qatar, Asia/Qostanay, Asia/Qyzylorda, Asia/Riyadh, Asia/Sakhalin, Asia/Samarkand, Asia/Seoul,
Asia/Shanghai, Asia/Singapore, Asia/Srednekolymsk, Asia/Taipei, Asia/Tashkent, Asia/Tbilisi,
Asia/Tehran, Asia/Thimphu, Asia/Tokyo, Asia/Tomsk, Asia/Ulaanbaatar, Asia/Urumqi, Asia/Ust-Nera,
Asia/Vientiane, Asia/Vladivostok, Asia/Yakutsk, Asia/Yangon, Asia/Yekaterinburg, Asia/Yerevan,
Atlantic/Azores, Atlantic/Bermuda, Atlantic/Canary, Atlantic/Cape_Verde, Atlantic/Faroe,
Atlantic/Madeira, Atlantic/Reykjavik, Atlantic/South_Georgia, Atlantic/St_Helena, Atlantic/Stanley,
Australia/Adelaide, Australia/Brisbane, Australia/Broken_Hill, Australia/Currie, Australia/Darwin,
Australia/Eucla, Australia/Hobart, Australia/Lindeman, Australia/Lord_Howe, Australia/Melbourne,
Australia/Perth, Australia/Sydney, Europe/Amsterdam, Europe/Andorra, Europe/Astrakhan,
Europe/Athens, Europe/Belgrade, Europe/Berlin, Europe/Bratislava, Europe/Brussels,
Europe/Bucharest, Europe/Budapest, Europe/Busingen, Europe/Chisinau, Europe/Copenhagen,
Europe/Dublin, Europe/Gibraltar, Europe/Guernsey, Europe/Helsinki, Europe/Isle_of_Man,
Europe/Istanbul, Europe/Jersey, Europe/Kaliningrad, Europe/Kiev, Europe/Kirov, Europe/Lisbon,
Europe/Ljubljana, Europe/London, Europe/Luxembourg, Europe/Madrid, Europe/Malta,
Europe/Mariehamn, Europe/Minsk, Europe/Monaco, Europe/Moscow, Europe/Oslo, Europe/Paris,
Europe/Podgorica, Europe/Prague, Europe/Riga, Europe/Rome, Europe/Samara, Europe/San_Marino,
Europe/Sarajevo, Europe/Saratov, Europe/Simferopol, Europe/Skopje, Europe/Sofia,
Europe/Stockholm, Europe/Tallinn, Europe/Tirane, Europe/Ulyanovsk, Europe/Uzhgorod,
Europe/Vaduz, Europe/Vatican, Europe/Vienna, Europe/Vilnius, Europe/Volgograd, Europe/Warsaw,
Europe/Zagreb, Europe/Zaporozhye, Europe/Zurich, Indian/Antananarivo, Indian/Chagos,
Indian/Christmas, Indian/Cocos, Indian/Comoro, Indian/Kerguelen, Indian/Mahe, Indian/Maldives,

Indian/Mauritius, Indian/Mayotte, Indian/Reunion, Pacific/Apia, Pacific/Auckland,
Pacific/Bougainville, Pacific/Chatham, Pacific/Chuuk, Pacific/Easter, Pacific/Efate, Pacific/Enderbury,
Pacific/Fakaofu, Pacific/Fiji, Pacific/Funafuti, Pacific/Galapagos, Pacific/Gambier,
Pacific/Guadalcanal, Pacific/Guam, Pacific/Honolulu, Pacific/Kiritimati, Pacific/Kosrae,
Pacific/Kwajalein, Pacific/Majuro, Pacific/Marquesas, Pacific/Midway, Pacific/Nauru, Pacific/Niue,
Pacific/Norfolk, Pacific/Noumea, Pacific/Pago_Pago, Pacific/Palau, Pacific/Pitcairn, Pacific/Pohnpei,
Pacific/Port_Moresby, Pacific/Rarotonga, Pacific/Saipan, Pacific/Tahiti, Pacific/Tarawa,
Pacific/Tongatapu, Pacific/Wake, Pacific/Wallis, GMT, UTC)

Chapter 4. Security

4.1. UserSecurity

User role API Key, only allowing status requests.

Type : apiKey

Name : Authorization

In : HEADER

4.2. AdminSecurity

Admin role API Key, allowing status, Operations and operational requests

Type : apiKey

Name : Authorization

In : HEADER

4.3. ServiceSecurity

Service role API Key, allowing special service requests including Admin level

Type : apiKey

Name : Authorization

In : HEADER