

# Quantum<sup>®</sup>

## Web Services

## StorNext



StorNext Web Services, 6-68116-01, February 2016, Product of USA.

Quantum Corporation provides this publication “as is” without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose. Quantum Corporation may revise this publication from time to time without notice.

## **COPYRIGHT STATEMENT**

© 2016 Quantum Corporation. All rights reserved.

Your right to copy this manual is limited by copyright law. Making copies or adaptations without prior written authorization of Quantum Corporation is prohibited by law and constitutes a punishable violation of the law.

## **TRADEMARK STATEMENT**

Artico, Be Certain (and the Q brackets design), DLT, DXi, DXi Accent, DXi V1000, DXi V2000, DXi V4000, FlexTier, GoVault, Lattus, NDX, the Q logo, the Q Quantum logo, Q-Cloud, Quantum (and the Q brackets design), the Quantum logo, Quantum Be Certain (and the Q brackets design), Quantum Vision, Scalar, StorageCare, StorNext, SuperLoader, Symform, the Symform logo (and design), vmPRO, and Xcellis are either registered trademarks or trademarks of Quantum Corporation and its affiliates in the United States and/or other countries. All other trademarks are the property of their respective owners.

Products mentioned herein are for identification purposes only and may be registered trademarks or trademarks of their respective companies. All other brand names or trademarks are the property of their respective owners.

Quantum specifications are subject to change.



# Contents

---

<b>Preface</b> .....	<b>v</b>
StorNext Web Services Version 2 (V2) .....	v
StorNext Web Services Version 1 (V1) .....	vi
Audience .....	vi
Ports Used By StorNext Web Services .....	vi
HTTP Port .....	vi
HTTPS Port .....	vii
View the List of Ports .....	vii
<b>Chapter 1: StorNext Web Services Commands (V2)</b> .....	<b>1</b>
StorNext Web Services V2 Commands, Usage, and Descriptions .....	1
StorNext Web Services V2 Commands, Usage, and Descriptions .....	2
Using the Web Services .....	2
Overview .....	3
Archive .....	7
Directory .....	14
Drive .....	21

File .....	27
Job .....	56
Media .....	63
Object Storage .....	122
Policy .....	127
Quota .....	139
Report .....	143
Schedule .....	155
System .....	174
Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2 .....	184
Prerequisites .....	184
Getting Started .....	185
Run the Web Services .....	187
Run the Web Services from a Client Application .....	191
Troubleshooting .....	237
<b>Chapter 2: StorNext Web Services Commands (V1) .....</b>	<b>238</b>
StorNext Web Services V1 Commands, Usage, and Descriptions .....	238
Using the APIs .....	239
API Commands .....	239
<b>Chapter 3: StorNext 4.x Web Services (WSAPI) .....</b>	<b>267</b>
StorNext 4.x Web Services (WSAPI) Commands, Usage, and Description .....	267
Using the APIs .....	268
WS-API APIs .....	270
Examples .....	286
<b>Appendix A: Sample Perl Script .....</b>	<b>292</b>
<b>Appendix B: SamplePython Script .....</b>	<b>308</b>



# Preface

---

This manual describes the commands supported by the StorNext Web Services and contains the following chapters:

- [Chapter 1: StorNext Web Services Commands \(V2\)](#)
- [Chapter 2: StorNext Web Services Commands \(V1\)](#)
- [Chapter 3: StorNext 4.x Web Services \(WSAPI\)](#)
- [Appendix A: Sample Perl Script](#)
- [Appendix B: SamplePython Script](#)

---

## StorNext Web Services Version 2 (V2)

Version 2 of StorNext web services offers better command coverage and supports XML, JSON, and TEXT response. The web services are categorized according to StorNext functional areas. For example, directories, files, media, etc.

Version 2 adds a user interface to turn the web service on or off, set the protocol (HTTP or HTTPS), and control authentication. By default, version 2 web services are turned off. Use the StorNext GUI to make changes (refer to **Web Services (V2)** in the *StorNext 5 User's Guide*).

---

**i Note:** StorNext Web Services V2 was released in StorNext 5 release 5.1.

For API commands supported by the Web Services (V2) for StorNext 5 release 5.1 (or later), see [StorNext Web Services V2 Commands, Usage, and Descriptions on page 2](#).

---

# StorNext Web Services Version 1 (V1)

Version 1 of StorNext web services is mostly used to migrate the SNAPI interface from a proprietary StorNext interface to a web service. It provides a mechanism to invoke SNAPI commands by calling a web service. Version 1 only supports XML output.

**i Note:** StorNext Web Services V1 was released in StorNext 5.

For API commands supported by the Web Service (V1) for releases prior to StorNext 5 release 5.1, see [StorNext Web Services V1 Commands, Usage, and Descriptions on page 238](#).

---

## Audience

This manual is written for StorNext operators, system administrators, and field service engineers.

---

## Ports Used By StorNext Web Services

The following table lists ports that are used by StorNext Web Services.

For a thorough explanation of StorNext's port selection algorithm, consult the `fsports(4)` man page.

Port	Protocol	StorNext Use	Notes
81	TCP	GUI (Java), Web Services	User starts at port 81, redirected to 443
443	TCP	GUI (Java), Web Services	

## HTTP Port

The default **HTTP** port is **81**.

**i Note:** If **HTTP** port **81** is not available during your installation, the first available port is used (82, 83, etc.).

# HTTPS Port

The default **HTTPS** port is **443**.

**i Note:** You do not need to specify the HTTPS port, as it will redirect to the secure port.

## View the List of Ports

You can view the list of ports (both GUI and Web Services) in the `/usr/adic/tomcat/conf/server.xml` file.

## Notational Conventions

This manual uses the following conventions:

Convention	Example
User input is shown in bold monospace font.	<b>./DARTinstall</b>
Computer output and command line examples are shown in monospace font.	<code>./DARTinstall</code>
User input variables are enclosed in angle brackets.	<b>http://&lt;ip_address&gt;/cgi-bin/stats</b>
For UNIX and Linux commands, the command prompt is implied.	<code>./DARTinstall</code> is the same as <code># ./DARTinstall</code>
File and directory names, menu commands, button names, and window names are shown in bold font.	<b>/data/upload</b>
Menu names separated by arrows indicate a sequence of menus to be navigated.	<b>Utilities &gt; Firmware</b>

The following formats indicate important information:

**i Note:** Note emphasizes important information related to the main topic.

**! Caution:** Caution indicates potential hazards to equipment or data.

**⚡ WARNING:** Warning indicates potential hazards to personal safety.

- Right side of the system - Refers to the right side as you face the component being described.
- Left side of the system - Refers to the left side as you face the component being described.

- Data sizes are reported in base 10 (decimal) rather than base  $2^{10}$  (binary). For example:  
10,995, 116,277,769 Bytes are reported as 11.0 TB (decimal/1000). In binary, this value is 10 TiB (binary/1024).

## Product Safety Statements


Quantum will not be held liable for damage arising from unauthorized use of the product. The user assumes all risk in this aspect.

This unit is engineered and manufactured to meet all safety and regulatory requirements. Be aware that improper use may result in bodily injury, damage to the equipment, or interference with other equipment.

- 
- ⚡ WARNING:** Before operating this product, read all instructions and warnings in this document and in the *Quantum Products System, Safety, and Regulatory Information Guide*.
- 
- ⚡ ADVARSEL:** Læs alle instruktioner og advarsler i dette dokument og i *Informationsvejledning vedrørende system-, sikkerheds- og lovbestemmelser for Quantum produkter, før produktet betjenes*.
- 
- ⚡ AVERTISSEMENT :** Avant d'utiliser ce produit, lisez toutes les instructions et les avertissements de ce document et du *Guide d'informations sur le système, la sécurité et la réglementation de Quantum*.
- 
- ⚡ WARNUNG:** Lesen Sie vor der Inbetriebnahme dieses Produkts alle Anleitungen und Warnungen in diesem Dokument und im *System-, Sicherheits- und Betriebsbestimmungen-Handbuch für Quantum-Produkte*.
- 
- ⚡ ADVERTENCIA:** Antes de hacer funcionar este producto, lea todas las instrucciones y advertencias de este documento y de la *Guía de información normativa, del sistema y de seguridad de los productos de Quantum*.
- 
- ⚡ WARNING:** Läs igenom alla instruktioner och varningar i detta dokument och i *Quantums produktsystem, säkerhet och reglerande informationsguide* innan denna produkt används.
- 
- ⚡ ВНИМАНИЕ!** Перед началом эксплуатации данного изделия прочтите все инструкции и предупреждения, приведенные в настоящем документе и в *Руководстве по системе, технике безопасности и действующим нормативам компании Quantum*.
- 
- ⚡ 警告 :** 本製品を使用される前に、本書と『*Quantum製品システム、安全、規制情報ガイド*』に記載されているすべての説明と警告をお読みください。
- 
- ⚡ 경고:** 본 제품을 작동하기 전에 본 문서와 *Quantum 제품 시스템, 안전 및 규제 정보 설명서*에 있는 모든 지침과 경고를 참조합니다.
- 
- ⚡ 警告 :** 在操作本产品之前，请阅读本文档和 *Quantum 产品系统、安全和法规信息指南*中的所有说明和警告。



---

 **警告：**操作此產品前，請閱讀本檔案及 *Quantum* 產品系統、安全與法規資訊指南中的指示與警告說明。

---

**אזהרה:** לפני ההפעלה של מוצר זה, קרא את כל ההוראות והאזהרות הכלולות במסמך זה וכן במדריך המידע בנושא מערכת, בטיחות ותקינה עבור מוצרי *Quantum*.

For the most up to date information on StorNext, see:

<http://www.quantum.com/serviceandsupport/get-help/index.aspx#contact-support>

## Contacts

For information about contacting Quantum, including Quantum office locations, go to:

<http://www.quantum.com/aboutus/contactus/index.aspx>

## Comments

To provide comments or feedback about this document, or about other Quantum technical publications, send e-mail to:

[doc-comments@quantum.com](mailto:doc-comments@quantum.com)

## Getting More Information or Help

StorageCare™, Quantum's comprehensive service approach, leverages advanced data access and diagnostics technologies with cross-environment, multi-vendor expertise to resolve backup issues faster and at lower cost.

Accelerate service issue resolution with these exclusive Quantum StorageCare services:

- **Service and Support Website** - Register products, license software, browse Quantum Learning courses, check backup software and operating system support, and locate manuals, FAQs, firmware downloads, product updates and more in one convenient location. Get started at:  
<http://www.quantum.com/serviceandsupport/get-help/index.aspx#contact-support>
- **eSupport** - Submit online service requests, update contact information, add attachments, and receive status updates via email. Online Service accounts are free from Quantum. That account can also be used to access Quantum's Knowledge Base, a comprehensive repository of product support information. Get started at:

<http://www.quantum.com/customercenter/>

For further assistance, or for training opportunities, contact the Quantum Customer Support Center:

Region	Support Contact
North America	1-800-284-5101 (toll free) +1-720-249-5700
EMEA	+800-7826-8888 (toll free) +49 6131 324 185
Asia Pacific	+800-7826-8887 (toll free) +603-7953-3010

For worldwide support:  
<http://www.quantum.com/serviceandsupport/get-help/index.aspx#contact-support>

## Worldwide End-User Product Warranty

For more information on the Quantum Worldwide End-User Standard Limited Product Warranty:

<http://www.quantum.com/serviceandsupport/warrantyinformation/index.aspx>



# Chapter 1: StorNext Web Services Commands (V2)

---

This chapter contains the following topics:

StorNext Web Services V2 Commands, Usage, and Descriptions .....	1
Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2 .....	184

---

## StorNext Web Services V2 Commands, Usage, and Descriptions

This section provides API commands supported by the Web Services (V2) for StorNext 5 release 5.1 (or later), along with respective:

- Description
- Parameters
- Parameters (input, output)
- Return Values
- Calling Sequence
- Usage Scenario
- Examples and Sample Output
- Related APIs (**where applicable**)
- Notes and Warnings (**where applicable**)

# StorNext Web Services V2 Commands, Usage, and Descriptions

This section provides API commands supported by the Web Services (V2) for StorNext 5 release 5.1 (or later), along with respective:

- Description
- Parameters
- Parameters (input, output)
- Return Values
- Calling Sequence
- Usage Scenario
- Examples and Sample Output
- Related APIs (**where applicable**)
- Notes and Warnings (**where applicable**)

## Using the Web Services

By default, web services are turned off. Use the StorNext GUI to turn web services on and set the protocol and authentication (refer to **Web Services (V2)** in the *StorNext 5 User's Guide*). The table below describes the allowed web service options.

**Table 1:** Allowed web service options

Option	Protocol	Description
Protocol	HTTP	Allow web services over HTTP.
	HTTPS	Allow web services over HTTPS.
	HTTP or HTTPS	Allow web services over HTTP or HTTPS.
Authentication	Off	No authentication is required to use web services.
	User	A username and password must be provided. The user is setup in the StorNext GUI.

Below is an example of a URL using the HTTP protocol and user authentication.

**i Note:** URL examples include multiple lines for clarity; your URLs will be a single line.

```
https://<<SERVER>>/sws/v2/report/fsqueue/media  
?username=wsuser  
&password=<<password>>  
&request=1928784996  
&format=xml
```

All examples are based on a protocol of HTTPS and no authentication.

### Description for Num in the Parameters Tables

The **Num** field in the parameters table displays if the parameter is a single value or a list of values can be passed.

If the parameter only accepts one value, **1** is used for **Num**. If a list of values is allowed, **N** is used for **Num**.

An example of a web service which accepts a list of values is illustrated below.

```
https://<<SERVER>>/sws/v2/file/fsfileinfo  
?file=/stornext/snfx1/smp2data/foobar0  
&file=/stornext/snfx1/smp2data/foobar1
```

## Overview

The table below provides a description for each category of a web service. The web services are organized into the high-level categories outlined in [Table 1 below](#).

**Table 1:** Web Service Categories and General Descriptions

Category	Description
Archive	Return information about an archive, query an archive port, or change the state of an archive.
Directory	Modify the class attributes of a directory or retrieve or recover files from media.
Drive	Report or change the state of drive components and storage subsystems.
File	Report, retrieve, and store files to tiered storage.
Job	Return information about jobs.
Media	Manage media – copy, clean up, move, and report.
Object Storage	Report Object Storage components.
Policy	Manage and report policies.

Category	Description
Quota	Manage and report quotas.
Report	Return information about subsystem resource requests.
Schedule	Manage and report schedules.
System	Get the status of system and Tertiary Storage Manager (TSM) components. Manage and report backups.

[Table 2 below](#) provides a brief description of each web service.

**Table 2:** Web Service Descriptions

Web Service	Description
<b>Archive</b>	
Query	Return information about an archive.
Query port	Query an archive port.
Change state	Change the state of an archive.
<b>Directory</b>	
Change attributes	Modify the class attributes of a directory.
Retrieve files	Retrieve files from media and place them on disk.
<b>Drive</b>	
State	Report the state of storage subsystem drive components and storage subsystems and Tertiary Storage Manager (TSM) software.
Change state	Change the state of a storage component in the Quantum storage subsystem.
<b>File</b>	
Information	Generate a report about files known to the Tertiary Storage Manager.
Tape location	Generate a report about a file's tape copy location.
Retrieve files	Retrieve files from media and place on disk.
Retrieve files / new location	Retrieve files from media and place into new disk file.
Relocate file from one affinity to another	Relocates a managed file from one disk affinity to another.

Web Service	Description
Retrieve files / partial	Retrieve partial files from media and place on disk.
Store	Expedite the storage of a file that currently resides on disk to media.
Remove disk copy	Remove the copy of a file from disk after the file was stored to a medium.
Modify class attributes	Modify the class attributes of a file.
<b>Job</b>	
Asynchronous job status	Returns a status of the jobs that are invoked asynchronously.
Detailed mover job status	Returns a detailed status of the mover jobs that are invoked asynchronously.
<b>Media</b>	
Information	Generate a media report based on their current status.
File background job status	Reports the status of the background job.
File inactive removal	Remove inactive versions of files.
File inactive remove by media	Clean all files on media that have been marked as logically blank.
File inactive removal list	List all media that has been marked as logically blank.
Move media	Move media from one archive to another.
Import cleaning media	Imports a list of cleaning media.
Import media	Imports a list of media.
Export media	Exports a list of media.
Copy media	Initiates copy-replace for a media.
Fragmentation report	Report on media fragmentation.
Eject media	Eject media out of an archive to be entered into another archive.
Enter media	Enter media that has been ejected out of an archive into another archive.
Purge media	Purges a list of media.
Attribute query	Queries for the attributes of one or more specified media.
Change media state	Change the class or state of a media.
Checkout media	Check media out of the Media Manager system.

Web Service	Description
List media for removal	List all media that are marked for removal.
List media by state	Lists media by state.
List media by location	Lists media by location.
<b>Object Storage</b>	
Report Object Storage components	Report Object Storage components.
<b>Policy</b>	
Class information	Report information on storage manager policies.
Modify a policy	Modify the processing parameters of a policy class.
Report policy class for a directory	Reports the policy class associated with directory.
Report policy class for a file system	Reports all policy classes with association points in a file system.
<b>Quota</b>	
Manage quotas	Manage the quota system in the StorNext file system.
<b>Report</b>	
Cancel requests	Cancels requests.
Files	Reports all files in the queue or specific files if a request identifiers or filename is specified.
Media	Reports the media movement for a request identifier or all media in queue.
Mover host	Reports the active mover host summary.
Mover request	Reports the active mover request summary.
Resource	Reports the active resource request summary.
<b>Schedule</b>	
Report	Prints the report of a feature or schedule.
Create	This web service will create a schedule.
Update	This web service will update a schedule.



Web Service	Description
Delete	This web service deletes an existing schedule.
Reset	This web service resets an existing schedule.
<b>System</b>	
Backup	Execute backup of configuration, database, and file system metadata.
Backup status	Retrieves the status of the backup operation.
File System Report	Reports the status of a file system and the status of stripe groups that belong to it.
Information	Retrieves the latest status of system components.
Parameters	Report the value for the specified Tertiary Manager System parameter.

## Archive

### Archive / Query

Return information about an archive.

This web service runs the **vsarchiveqry** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
archive	Optional	1	Identifies the archive to be queried. If not specified, all archives are queried.	-a (if absent)

Parameter	Req / Opt	Num	Description	CLI Option
showclassgroups	Optional	1	Indicates that detailed information on all MediaClass groups associated with the specified archive(s) are to be reported. Default value is <b>false</b> .	-c
showdrives	Optional	1	Indicates that all drives associated with the specified archive(s) are to be reported. Default value is <b>false</b> .	-d
showmedia	Optional	1	Indicates that all media associated with the specified archive(s) are to be reported. Default value is <b>false</b> .	-m
showports	Optional	1	Indicates that all import/export ports associated with the specified archive(s) are to be reported. Default value is <b>false</b> .	-s
showtypes	Optional	1	Indicates that detailed information on all media types associated with the specified archive(s) are to be reported. Default value is <b>false</b> .	-t
hostname	Optional	1	The host name of the Media Manager server.	-H
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries that web service will attempt if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout to the CLI software. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/archive/vsarchiveqry
?archive=myarchive
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<vsarchiveqry xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vsarchiveqry.xsd">
<header>
<commandName>vsarchiveqry</commandName>
<commandLine>/usr/adic/MSM/clibin/vsarchiveqry myarchive -F xml</commandLine>
<commandDescription>Queries for information about an
archive.</commandDescription>
<localDateISO>2014-05-15T10:44:08</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>10:44:08</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T15:44:08Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>15:44:08</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<archives>
<archive>
<archiveName>myarchive</archiveName>
<archiveType>SCSI</archiveType>
<archiveState>On-line</archiveState>
<archiveMode>Attended</archiveMode>
<fillMode>None</fillMode>
<configState>Not Being Configured</configState>
</archive>
</archives>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T10:44:08</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>10:44:08</localTimeEnd>
```

```
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T15:44:08Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>15:44:08</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0178</elapsedTimeInSeconds>  
</footer>  
</vsarchiveqry>
```

## Archive / Query Port

Query an archive port.

This web service runs the **mmportinfo** command.

**Note:** This web service returns text output.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
archive	Required	1	Identifies the archive to obtain port information about. Valid archive names may contain up to 16 alphanumeric characters, including spaces. Leading and trailing spaces are not permitted.	archive name

### Example

```
https://<<SERVER>>/sws/v2/archive/mmportinfo  
?archive=i40
```

### Output

```
Exit Code: 0  
Output: 512:LTOW,LT0:0,0,15,512
```

## Archive / Change State

Change the state of an archive.

This web service runs the **vsarchivevary** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
archive	Required	1	Identifies the archive to be queried.	archive name
state	Required	1	Identifies the state into which the archive is placed. Valid archive states are: <b>on-line</b> , <b>off-line</b> , and <b>diagnostic</b> . The archive states, <b>on-line</b> , <b>off-line</b> , and <b>diagnostic</b> are abbreviated as <b>on</b> , <b>of</b> , and <b>d</b> respectively.	-s
hostname	Optional	1	The host name of the Media Manager server.	-H
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries that web service will attempt if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout to the CLI software. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/archive/vsarchivevary  
?archive=myarchive  
&state=on  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>  
<vsarchivevary xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="vsarchivevary.xsd">  
<header>  
<commandName>vsarchivevary</commandName>  
<commandLine>/usr/adic/MSM/clibin/vsarchivevary myarchive -s on -F  
xml</commandLine>  
<commandDescription>Varies the state of an archive.</commandDescription>  
<localDateISO>2014-05-15T10:45:46</localDateISO>  
<localDate>2014-05-15</localDate>  
<localTime>10:45:46</localTime>  
<localDayOfWeek>4</localDayOfWeek>  
<gmtDateISO>2014-05-15T15:45:46Z</gmtDateISO>  
<gmtDate>2014-05-15</gmtDate>  
<gmtTime>15:45:46</gmtTime>  
<gmtDayOfWeek>4</gmtDayOfWeek>  
</header>  
<vsarchivevaryOutput>  
<vsarchivevarySuccess>  
<vsarchivevaryArchive>myarchive</vsarchivevaryArchive>  
<vsarchivevaryState>on-line</vsarchivevaryState>  
</vsarchivevarySuccess>  
</vsarchivevaryOutput>  
<footer>  
<returnCode>0</returnCode>  
<localDateISOEnd>2014-05-15T10:45:46</localDateISOEnd>  
<localDateEnd>2014-05-15</localDateEnd>  
<localTimeEnd>10:45:46</localTimeEnd>  
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T15:45:46Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>15:45:46</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0183</elapsedTimeInSeconds>  
</footer>  
</vsarchivevary>
```

# Directory

## Directory / Change Attributes

Modify the class attributes of a directory.

This web service runs the **fschdiat** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
directory	Required	N	Directory or directories to modify.	directory name
store	Optional	1	This option indicates how storage policies operate. Valid values are <b>exclude</b> and <b>policy</b> . The <b>exclude</b> option excludes files from storage when a store policy occurs. The <b>policy</b> option stores files by storage policy.	-s
reloc	Optional	1	This option indicates how relocation policies operate. Valid values are <b>exclude</b> and <b>policy</b> . The <b>exclude</b> option excludes files from storage when a store policy occurs. The <b>policy</b> option stores files by storage policy.	-r
trunc	Optional	1	This option indicates how truncation policies operate. Valid values are <b>exclude</b> and <b>policy</b> . The <b>exclude</b> option excludes files from storage when a store policy occurs. The <b>policy</b> option stores files by storage policy.	-t



Parameter	Req / Opt	Num	Description	CLI Option
clean	Optional	1	This option indicates if the database entries are to be cleaned when the file is removed from the file system. The <b>yes</b> option indicates that the database entries will be cleaned and the file will NOT be recoverable. The <b>no</b> option indicates that the database entries will NOT be cleaned and the file will be recoverable.	-D
policy	Optional	1	This specifies the class that will be associated with the directory.	-c
stubsizes	Optional	1	This indicates the stub size (in kilobytes) and is used to determine the number of bytes to leave on disk when files are truncated. It will be the minimum number of bytes left on disk (the value is rounded up to a multiple of the file system block size). If policy is specified as the value, then the policy class definitions will be used to determine the stub size.	-S
recursive	Optional	1	The directory from which to start the recursive operation.	-R

### Example

```
https://<<SERVER>>/sws/v2/directory/fschdiat  
?policy=smp4  
&directory=/stornext/snfx1/smp2data/foodir0  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8"?>
<fschdiat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fschdiat.xsd">
  <header>
    <commandName>fschdiat</commandName>
    <commandLine>/usr/adic/TSM/bin/fschdiat -c smp4 -F xml
/stornext/snfx1/smp2data/foodir0</commandLine>
    <commandDescription>Modify the class attributes of a
directory</commandDescription>
    <localDateISO>2015-12-03T11:49:12</localDateISO>
    <localDate>2015-12-03</localDate>
    <localTime>11:49:12</localTime>
    <localDayOfWeek>4</localDayOfWeek>
    <gmtDateISO>2015-12-03T18:49:12Z</gmtDateISO>
    <gmtDate>2015-12-03</gmtDate>
    <gmtTime>18:49:12</gmtTime>
    <gmtDayOfWeek>4</gmtDayOfWeek>
  </header>
  <statuses>
    <status>
      <statusCode>FS0109</statusCode>
      <statusNumber>109</statusNumber>
      <dayOfMonth>3</dayOfMonth>
      <requestId>180911</requestId>
      <commandName>/usr/adic/TSM/bin/fschdiat</commandName>
      <commandStatus>interim</commandStatus>
      <statusText>Command successful for
/stornext/snfx1/smp2data/foodir0.</statusText>
    </status>
    <status>
      <statusCode>FS0390</statusCode>
      <statusNumber>390</statusNumber>
      <dayOfMonth>3</dayOfMonth>
      <requestId>180911</requestId>
      <commandName>/usr/adic/TSM/bin/fschdiat</commandName>
      <commandStatus>completed</commandStatus>
      <statusText>1 out of 1 directory attribute changes were
successful.</statusText>
    </status>
    <status>
      <statusCode>FS0000</statusCode>
      <statusNumber>0</statusNumber>
      <dayOfMonth>3</dayOfMonth>
      <requestId>180911</requestId>
      <commandName>/usr/adic/TSM/bin/fschdiat</commandName>
      <commandStatus>completed</commandStatus>
      <statusText>Command Successful.</statusText>
    </status>
  </statuses>
</fschdiat>
```

```

    </status>
  </statuses>
  <footer>
    <returnCode>0</returnCode>
    <localDateISOEnd>2015-12-03T11:49:12</localDateISOEnd>
    <localDateEnd>2015-12-03</localDateEnd>
    <localTimeEnd>11:49:12</localTimeEnd>
    <localDayOfWeekEnd>4</localDayOfWeekEnd>
    <gmtDateISOEnd>2015-12-03T18:49:12Z</gmtDateISOEnd>
    <gmtDateEnd>2015-12-03</gmtDateEnd>
    <gmtTimeEnd>18:49:12</gmtTimeEnd>
    <gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
    <elapsedTimeInSeconds>0.0003</elapsedTimeInSeconds>
  </footer>
</fschdiat>

```

## Directory / Retrieve Files

Retrieve or recover files from media and place on disk.

This web service runs the **fsretrieve** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
directory	Required	1	The directory from which to start the recursive retrieves.	-R
updateaccesstime	Optional	1	Updates the access time for the requested files. Default value is <b>false</b> .	-a
toppriority	Optional	1	Specifies top priority and will cause all files for the retrieve request to be placed at the top of the retrieve queue. Default value is <b>false</b> .	-p

## Example

```
https://<<SERVER>>/sws/v2/directory/fsretrieve  
?directory=/stornext/snfs1/xxx  
&updateaccesstime=false  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsretrieve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsretrieve.xsd">
<header>
<commandName>fsretrieve</commandName>
<commandLine>/usr/adic/TSM/bin/fsretrieve -R /stornext/snfs1/xxx -F
xml</commandLine>
<commandDescription>Retrieve files from media and place on
disk</commandDescription>
<localDateISO>2014-05-15T11:04:42</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:04:42</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:04:42Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:04:42</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0657</statusCode>
<statusNumber>657</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387987</requestId>
<commandName>/usr/adic/TSM/bin/fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Creating map of /stornext/snfs1/xxx ...</statusText>
</status>
<status>
<statusCode>FS0658</statusCode>
<statusNumber>658</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387987</requestId>
<commandName>/usr/adic/TSM/bin/fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Mapping completed.</statusText>
</status>
<status>
<statusCode>FS0589</statusCode>
```

```
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387987</requestId>
<commandName>/usr/adic/TSM/bin/fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0655</statusCode>
<statusNumber>655</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387987</requestId>
<commandName>fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Currently processed 0 out of 100 files.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387987</requestId>
<commandName>fsretrieve</commandName>
<commandStatus>completed</commandStatus>
<statusText>100 out of 100 retrieves were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:04:51</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:04:51</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
```

```
<gmtDateISOEnd>2014-05-15T16:04:51Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>16:04:51</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>9.0595</elapsedTimeInSeconds>  
</footer>  
</fsretrieve>
```

## Drive

### Drive / State

Report the state of all Quantum storage subsystem drive components and storage subsystems and/or Tertiary Manager software.

This web service runs the **fsstate** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
drive	Optional	1	Component alias drive identifier. Submitting the <b>fsstate</b> request without this option generates a report showing all storage subsystems and drive components currently configured in the Quantum storage subsystem and the state of Tertiary Manager software.	component alias

#### Example

```
https://<<SERVER>>/sws/v2/drive/fsstate  
?drive=vt1_dr1  
&format=xml
```

## Output



```
<?xml version="1.0" encoding="UTF-8" ?>
<fsstate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsstate.xsd">
<header>
<commandName>fsstate</commandName>
<commandLine>/usr/adic/TSM/bin/fsstate vtl_dr1 -F xml</commandLine>
<commandDescription>Report the state of all Stornext drive components and storage
subsystems and/or Tertirary Manager software</commandDescription>
<localDateISO>2014-05-15T11:08:06</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:08:06</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:08:06Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:08:06</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<components>
<component>
<note>configured</note>
<alias>vtl_dr1</alias>
<driveId>1</driveId>
<driveState>ON</driveState>
<driveStatus>FREE</driveStatus>
<mediaId>NONE</mediaId>
</component>
</components>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387993</requestId>
```

```

<commandName>fsstate</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:08:06</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:08:06</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:08:06Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:08:06</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0084</elapsedTimeInSeconds>
</footer>
</fsstate>
    
```

## Drive / Change State

Change the state of a storage component in the Quantum storage subsystem.

This web service runs the **fschstate** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
drive	Required	1	Component alias drive identifier.	component alias
state	Required	1	The desired state of the drive components or subsystems. The valid values for drive components are <b>MAINT</b> , <b>ON</b> , or <b>OFF</b> . Valid values for subsystems are <b>ON</b> or <b>OFF</b> .	-s

### Example

```
https://<<SERVER>>/sws/v2/drive/fschstate  
?drive=vtl_dr1  
&state=ON  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fschstate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fschstate.xsd">
<header>
<commandName>fschstate</commandName>
<commandLine>/usr/adic/TSM/exec/fschstate vtl_dr1 -s on -F xml</commandLine>
<commandDescription>Change the state of a storage component.</commandDescription>
<localDateISO>2014-05-15T11:09:06</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:09:06</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:09:06Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:09:06</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0258</statusCode>
<statusNumber>258</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387995</requestId>
<commandName>fschstate</commandName>
<commandStatus>interim</commandStatus>
<statusText>Command attempted on vtl_dr1. New state = on. Status =
Successful.</statusText>
</status>
<status>
<statusCode>FS0270</statusCode>
<statusNumber>270</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387995</requestId>
<commandName>fschstate</commandName>
```

```
<commandStatus>completed</commandStatus>
<statusText>Overall status = Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:09:06</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:09:06</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:09:06Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:09:06</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0175</elapsedTimeInSeconds>
</footer>
</fschstate>
```

## File

### File / Information

Generate a report about files known to the Tertiary Storage Manager.

This web service runs the **fsfileinfo** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
file	Optional	1	List of files for which information is required. <b>i Note:</b> At least one file or directory path is required.	filename
directory	Optional	1	The directory from which to start the recursive operation. All entries from the specified directory and any subdirectories will be processed. <b>i Note:</b> At least one file or directory path is required.	-R
checksum	Optional	1	If checksum was turned on for the file when stored and this option is specified, the checksum value generated for the file will be displayed. Default value is <b>false</b> .	-c
objectinfo	Optional	1	If the file has one or more copies stored to Object Storage, then the object ids will be displayed. Offset information will also be displayed for each object id. Default value is <b>false</b> .	-o

### Example

```
https://<<SERVER>>/sws/v2/file/fsfileinfo  
?file=/stornext/snfs1/xxx/testfile.0  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsfileinfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsfileinfo.xsd">
<header>
<commandName>fsfileinfo</commandName>
<commandLine>/usr/adic/TSM/bin/fsfileinfo -F xml
/stornext/snfs1/xxx/testfile.0</commandLine>
<commandDescription>Generate a report about files known to the Tertiary
Manager</commandDescription>
<localDateISO>2014-05-15T11:12:56</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:12:56</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:12:56Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:12:56</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<fileInfos>
<fileInfo>
<fileName>/stornext/snfs1/xxx/testfile.0</fileName>
<storedPathFileName>N/A</storedPathFileName>
<storedPathSameAsFileName>>false</storedPathSameAsFileName>
<lastModificationDateString>15-may-2014 11:11:09</lastModificationDateString>
<lastModificationDate>2014-05-15</lastModificationDate>
<lastModificationDayOfWeek>4</lastModificationDayOfWeek>
<lastModificationTime>11:11:09</lastModificationTime>
<owner>root</owner>
<location>DISK</location>
<group>root</group>
<existingCopies>0</existingCopies>
<access>664</access>
<targetCopies>1</targetCopies>
<targetStubSize>0</targetStubSize>
```

```
<targetStubScale>1024</targetStubScale>
<existingStubSize>n/a</existingStubSize>
<fileSize>100000</fileSize>
<store>MINTIME</store>
<affinity>n/a</affinity>
<reloc>MINTIME</reloc>
<class>xxx</class>
<trunc>MINTIME</trunc>
<cleanDBInfo>NO</cleanDBInfo>
<medias>
<media>
<message>None</message>
</media>
</medias>
<checksums>
<checksum>
<summary>N</summary>
</checksum>
</checksums>
<objects>
<object>
<summary>N</summary>
</object>
</objects>
</fileInfo>
</fileInfos>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388006</requestId>
```



```

<commandName>/usr/adic/TSM/bin/fsfileinfo</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:12:56</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:12:56</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:12:56Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:12:56</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0022</elapsedTimeInSeconds>
</footer>
</fsfileinfo>
    
```

## File / Tape Location

Generate a report about a file's tape copy location.

This web service runs the **fsfiletapeloc** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

Parameter	Req / Opt	Num	Description	CLI Option
file	Required	1	The file for which location is being queried. The path name of the file is required.	file name
copynumber	Optional	1	The copy ID to generate the report for. If not specified, the information for the primary copy will be reported.	-c

### Example

```
https://<<SERVER>>/sws/v2/file/fsfiletapeloc  
?file=/stornext/snfs1/xxx/testfile.0  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsfiletapeloc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsfiletapeloc.xsd">
<header>
<commandName>fsfiletapeloc</commandName>
<commandLine>/usr/adic/TSM/bin/fsfiletapeloc -F xml
/stornext/snfs1/xxx/testfile.0</commandLine>
<commandDescription>Generate a report about a file's tape copy
location</commandDescription>
<localDateISO>2014-05-15T11:16:03</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:16:03</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:16:03Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:16:03</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<fileInfo>
<segmentNumber>1</segmentNumber>
<mediaId>E00007</mediaId>
<libraryId>myarchive</libraryId>
<format>ANTF</format>
<startBlock>25</startBlock>
<offset>128</offset>
<segmentSize>100000</segmentSize>
<blockSize>524288</blockSize>
</fileInfo>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388011</requestId>
```

```

<commandName>/usr/adic/TSM/bin/fsfiletapeloc</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:16:03</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:16:03</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:16:03Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:16:03</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0008</elapsedTimeInSeconds>
</footer>
</fsfiletapeloc>

```

## File / Retrieve Files

Retrieve truncated files from media and place on disk.

This web service runs the **fsretrieve** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
file	Optional	N	List of files that need to be retrieved. <b>i Note:</b> At least one file or directory path is required.	filename
directory	Optional	1	The directory from which to start the recursive retrieve. <b>i Note:</b> At least one file or directory path is required.	-R
copynumber	Optional	1	Used to retrieve a specific copy of filename if one exists.	-c
forcealtlocation	Optional	1	Force change in alternate retrieval location behavior. Valid values are <b>true</b> and <b>false</b> .	-x
updateaccesstime	Optional	1	Updates the access time for the requested files.	-a
toppriority	Optional	1	Specifies top priority and will cause all files for the retrieve request to be placed at the top of the retrieve queue. Default value is <b>false</b> .	-p

### Example

```
https://<<SERVER>>/sws/v2/file/fsretrieve
?file=/stornext/snfs1/xxx/testfile.0
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsretrieve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsretrieve.xsd">
<header>
<commandName>fsretrieve</commandName>
<commandLine>/usr/adic/TSM/bin/fsretrieve -a -p -F xml
/stornext/snfs1/xxx/testfile.0</commandLine>
<commandDescription>Retrieve files from media and place on
disk</commandDescription>
<localDateISO>2014-05-15T11:21:35</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:21:35</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:21:35Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:21:35</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388015</requestId>
<commandName>/usr/adic/TSM/bin/fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0347</statusCode>
<statusNumber>347</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388015</requestId>
```

```
<commandName>fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>File /stornext/snfs1/xxx/testfile.0 has been retrieved.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388015</requestId>
<commandName>fsretrieve</commandName>
<commandStatus>completed</commandStatus>
<statusText>1 out of 1 retrieves were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:21:39</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:21:39</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:21:39Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:21:39</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>4.0212</elapsedTimeInSeconds>
</footer>
</fsretrieve>
```

## File / Retrieve Files / New Location

Retrieve files from media and place them into a new disk file.

This web service runs the **fsretrieve** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
file	Required	1	File that needs to be retrieved.	file name
newfile	Required	1	The new path and file name into which to retrieve the file. The location specified for the new file must be a local file system. Retrieval to an NFS mounted file system is not permitted.	-n
copynumber	Optional	1	Used to retrieve a specific copy of filename if one exists.	-c
updateaccesstime	Optional	1	Updates the access time for the requested files.	-a
toppriority	Optional	1	Specifies top priority and will cause all files for the retrieve request to be placed at the top of the retrieve queue. Default value is <b>false</b> .	-p

### Example

```
https://<<SERVER>>/sws/v2/file/fsretrieve/new
?file=/stornext/snfs1/xxx/testfile.0
&newfile=/stornext/snfs1/xxx/testfile.1
&format=xml
```

### Output



```
<?xml version="1.0" encoding="UTF-8" ?>
<fsretrieve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsretrieve.xsd">
<header>
<commandName>fsretrieve</commandName>
<commandLine>/usr/adic/TSM/bin/fsretrieve /stornext/snfs1/xxx/testfile.0 -n
/stornext/snfs1/xxx/testfile.1 -F xml</commandLine>
<commandDescription>Retrieve files from media and place on
disk</commandDescription>
<localDateISO>2014-05-15T11:26:44</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:26:44</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:26:44Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:26:44</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388021</requestId>
<commandName>/usr/adic/TSM/bin/fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0347</statusCode>
<statusNumber>347</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388021</requestId>
<commandName>fsretrieve</commandName>
```

```
<commandStatus>interim</commandStatus>
<statusText>File /stornext/snfs1/xxx/testfile.0 has been retrieved.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388021</requestId>
<commandName>fsretrieve</commandName>
<commandStatus>completed</commandStatus>
<statusText>1 out of 1 retrieves were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:26:46</localDateISOEnd> <localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:26:46</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:26:46Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:26:46</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>2.0689</elapsedTimeInSeconds>
</footer>
</fsretrieve>
```

## Files / Relocate file from one affinity to another

This web service relocates a managed file from one disk affinity to another.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
file	Required	N	The files that need to be relocated.	None
affinity	Required	1	The destination affinity for the files.	-a

### Example

```
https://<<SERVER>>/sws/v2/file/fsrelocate  
?file=/stornext/snfs1/pol1/file1.txt  
&affinity=tier2  
&format=json
```

### Output

```
{
  "header": {
    "commandName": "fsrelocate",
    "commandLine": "fsrelocate /stornext/snfs1/pol11/file1.txt -a tier2 -F JSON",
    "commandDescription": "Relocate a managed file from one disk affinity to another
or change the affinity association of a truncated file",
    "localDateISO": "2015-11-03T07:19:34",
    "localDate": "2015-11-03",
    "localTime": "07:19:34",
    "localDayOfWeek": 2,
    "gmtDateISO": "2015-11-03T13:19:34Z",
    "gmtDate": "2015-11-03",
    "gmtTime": "13:19:34",
    "gmtDayOfWeek": 2
  },
  "statuses": [
    {
      "statusCode": "FS0589",
      "statusNumber": 589,
      "dayOfMonth": 3,
      "requestId": 255713,
      "commandName": "fsrelocate",
      "commandStatus": "interim",
      "statusText": "Tertiary Manager software request received.
/stornext/snfs1/pol11/file1.txt"
    },
    {
      "statusCode": "FS0684",
      "statusNumber": 684,
      "dayOfMonth": 3,
      "requestId": 255713,
      "commandName": "fsrelocate",
      "commandStatus": "interim",
      "statusText": "File successfully relocated."
    },
    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 3,
      "requestId": 255713,
      "commandName": "fsrelocate",
      "commandStatus": "completed",
      "statusText": "1 out of 1 relocates were successful."
    }
  ]
}
```

```

}
],
"footer": {
"returnCode": 0,
"localDateISOEnd": "2015-11-03T07:19:34",
"localDateEnd": "2015-11-03",
"localTimeEnd": "07:19:34",
"localDayOfWeekEnd": 2,
"gmtDateISOEnd": "2015-11-03T13:19:34Z",
"gmtDateEnd": "2015-11-03",
"gmtTimeEnd": "13:19:34",
"gmtDayOfWeekEnd": 2,
"elapsedTimeInSeconds": "0.0040"
}
}
    
```

## File / Retrieve Files / Partial

Retrieve files from media and place them on disk.

This web service runs the **fsretrieve** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
file	Required	1	File that needs to be retrieved.	file name
newfile	Required	1	The new path and file name into which to retrieve the file. The location specified for the new file must be a local file system. Retrieval to an NFSmounted file system is not permitted.	-n

Parameter	Req / Opt	Num	Description	CLI Option
startbyte	Required	1	The starting byte for the file to be retrieved. The startbyte must be less than endbyte, and both must be within the byte range of the file. The byte range is inclusive.	-b
endbyte	Required	1	The last byte for the file to be retrieved.	-b
updateaccesstime	Optional	1	Updates the access time for the requested files.	-a
toppriority	Optional	1	Specifies top priority and will cause the file for the retrieve request to be placed at the top of the retrieve queue. Default value is <b>false</b> .	-p

### Example

```
https://<<SERVER>>/sws/v2/file/fsretrieve/partial  
?file=/stornext/snfs1/xxx/testfile.0  
&newfile=/stornext/snfs1/xxx/testfile.1  
&startbyte=100  
&endbyte=1000  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsretrieve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsretrieve.xsd">
<header>
<commandName>fsretrieve</commandName>
<commandLine>/usr/adic/TSM/bin/fsretrieve /stornext/snfs1/xxx/testfile.0 -n
/stornext/snfs1/xxx/testfile.1 -b 100 1000 -F xml</commandLine>
<commandDescription>Retrieve files from media and place on
disk</commandDescription>
<localDateISO>2014-05-15T11:35:23</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:35:23</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:35:23Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:35:23</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388031</requestId>
<commandName>/usr/adic/TSM/bin/fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0347</statusCode>
<statusNumber>347</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388031</requestId>
```

```
<commandName>fsretrieve</commandName>
<commandStatus>interim</commandStatus>
<statusText>File /stornext/snfs1/xxx/testfile.0 has been retrieved.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388031</requestId>
<commandName>fsretrieve</commandName>
<commandStatus>completed</commandStatus>
<statusText>1 out of 1 retrieves were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:35:24</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:35:24</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:35:24Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:35:24</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0727</elapsedTimeInSeconds>
</footer>
</fsretrieve>
```



## File / Store

Expedite the storage of a file that currently resides on disk to media.

This web service runs the **fsstore** command.

### Parameters



Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
file	Optional	N	File that needs to be stored.  <b>Note:</b> At least one file or directory path is required.	file name
directory	Optional	1	The directory from which to do start the recursive retrieve.  <b>Note:</b> At least one file or directory path is required.	-R
mediatype	Optional	1	Defines the type of medium to be used for storage.	-t
copies	Optional	1	Number of copies of the file(s) to be stored.	-c
trunc	Optional	1	The truncation mode. Valid value are <b>immediate</b> and <b>policy</b> .	-f
drivepool	Optional	1	Media Manager drive pool group used to store the file specified.	-v
minsize	Optional	1	Minimum File Size in bytes to be stored.	-z
runtime	Optional	1	Maximum allowable time in hours for the command to finish.	-u

### Example

```
https://<<SERVER>>/sws/v2/file/fsstore
?file=/stornext/snfx1/smp2data/filex1
&file=/stornext/snfx1/smp2data/filex2
&copies=1
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8"?>
<fsstore xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsstore.xsd">
  <header>
    <commandName>fsstore</commandName>
    <commandLine>/usr/adic/TSM/bin/fsstore -c 1 -F xml
/stornext/snfx1/smp2data/filex1 /stornext/snfx1/smp2data/filex2</commandLine>
    <commandDescription>Request the storage of a file that currently
resides on disk to media</commandDescription>
    <localDateISO>2015-12-03T11:43:19</localDateISO>
    <localDate>2015-12-03</localDate>
    <localTime>11:43:19</localTime>
    <localDayOfWeek>4</localDayOfWeek>
    <gmtDateISO>2015-12-03T18:43:19Z</gmtDateISO>
    <gmtDate>2015-12-03</gmtDate>
    <gmtTime>18:43:19</gmtTime>
    <gmtDayOfWeek>4</gmtDayOfWeek>
  </header>
  <statuses>
    <status>
      <statusCode>FS0589</statusCode>
      <statusNumber>589</statusNumber>
      <dayOfMonth>3</dayOfMonth>
      <requestId>180889</requestId>
      <commandName>/usr/adic/TSM/bin/fsstore</commandName>
      <commandStatus>interim</commandStatus>
      <statusText>Tertiary Manager software request
received.</statusText>
    </status>
    <status>
      <statusCode>FS0799</statusCode>
      <statusNumber>799</statusNumber>
      <dayOfMonth>3</dayOfMonth>
      <requestId>180889</requestId>
      <commandName>/usr/adic/TSM/bin/fsstore</commandName>
      <commandStatus>interim</commandStatus>
      <statusText>2 file store request(s) have been sent to Tertiary
Manager.</statusText>
    </status>
    <status>
      <statusCode>FS0346</statusCode>
      <statusNumber>346</statusNumber>
      <dayOfMonth>3</dayOfMonth>
      <requestId>180889</requestId>
      <commandName>fsstore</commandName>
      <commandStatus>interim</commandStatus>
    </status>
  </statuses>
</fsstore>
```

```
    <statusText>File /stornext/snfx1/smp2data/filex1 copy 1 has been
stored.</statusText>
  </status>
  <status>
    <statusCode>FS0346</statusCode>
    <statusNumber>346</statusNumber>
    <dayOfMonth>3</dayOfMonth>
    <requestId>180889</requestId>
    <commandName>fsstore</commandName>
    <commandStatus>interim</commandStatus>
    <statusText>File /stornext/snfx1/smp2data/filex2 copy 1 has been
stored.</statusText>
  </status>
  <status>
    <statusCode>FS0390</statusCode>
    <statusNumber>390</statusNumber>
    <dayOfMonth>3</dayOfMonth>
    <requestId>180889</requestId>
    <commandName>fsstore</commandName>
    <commandStatus>completed</commandStatus>
    <statusText>2 out of 2 statuses were successful.</statusText>
  </status>
  <status>
    <statusCode>FS0000</statusCode>
    <statusNumber>0</statusNumber>
    <dayOfMonth>3</dayOfMonth>
    <requestId>180889</requestId>
    <commandName>fsstore</commandName>
    <commandStatus>completed</commandStatus>
    <statusText>Command Successful.</statusText>
  </status>
</statuses>
<footer>
  <returnCode>0</returnCode>
  <localDateISOEnd>2015-12-03T11:43:21</localDateISOEnd>
  <localDateEnd>2015-12-03</localDateEnd>
  <localTimeEnd>11:43:21</localTimeEnd>
  <localDayOfWeekEnd>4</localDayOfWeekEnd>
  <gmtDateISOEnd>2015-12-03T18:43:21Z</gmtDateISOEnd>
  <gmtDateEnd>2015-12-03</gmtDateEnd>
  <gmtTimeEnd>18:43:21</gmtTimeEnd>
  <gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
  <elapsedTimeInSeconds>1.0867</elapsedTimeInSeconds>
</footer>
</fsstore>
```

## File / Remove Disk Copy

Remove the copy of a file from disk after the file was stored to a medium.

This web service runs the **fsrcmdiskcopy** command.

## Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
file	Optional	N	One or more files to remove from disk. <b>i Note:</b> At least one file or directory path is required.	file name
directory	Optional	1	The directory from which to start the recursive operation. All entries from the specified directory and any subdirectories will be processed. <b>i Note:</b> At least one file or directory path is required.	-R

## Example

```
https://<<SERVER>>/sws/v2/file/fsrcmdiskcopy  
?file=/stornext/snfs1/xxx/testfile.0  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsrmdiskcopy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsrmdiskcopy.xsd">
<header>
<commandName>fsrmdiskcopy</commandName>
<commandLine>/usr/adic/TSM/bin/fsrmdiskcopy -F xml
/stornext/snfs1/xxx/testfile.0</commandLine>
<commandDescription>Remove file data blocks from disk after the file was stored
to a medium</commandDescription>
<localDateISO>2014-05-15T11:43:34</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:43:34</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:43:34Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:43:34</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0266</statusCode>
<statusNumber>266</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388038</requestId>
<commandName>/usr/adic/TSM/bin/fsrmdiskcopy</commandName>
<commandStatus>interim</commandStatus>
<statusText>Data disk blocks for file /stornext/snfs1/xxx/testfile.0 were
successfully removed.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388038</requestId>
```

```

<commandName>/usr/adic/TSM/bin/fsrmdiskcopy</commandName>
<commandStatus>completed</commandStatus>

<statusText>1 out of 1 disk copy removes were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:43:34</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:43:34</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:43:34Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:43:34</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0007</elapsedTimeInSeconds>
</footer>
</fsrmdiskcopy>
    
```



## File / Modify Class Attributes

Modify the class attributes of a file.

This web service runs the **fschfiat** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

Parameter	Req / Opt	Num	Description	CLI Option
file	Optional	N	One or more file(s) having the attributes changed.  <b>Note:</b> At least one file or directory path is required.	file name
directory	Optional	1	The directory from which to do start the recursive retrieve.  <b>Note:</b> At least one file or directory path is required.	-R
store	Optional	1	This option indicates how storage policies operate on the file. Valid values are <b>exclude</b> and <b>policy</b> . The <b>exclude</b> argument excludes the file from storage when a store policy occurs. The <b>policy</b> argument stores the file by storage policy.	-s
reloc	Optional	1	This option indicates how relocation policies operate on the file. Valid values are <b>exclude</b> and <b>policy</b> . The <b>exclude</b> argument excludes the file from relocation when a relocation policy occurs. The <b>policy</b> argument relocates the file by relocation policy.	-r
trunc	Optional	1	This option indicates how truncation policies operate on the file. Valid values are <b>exclude</b> , <b>policy</b> , <b>immediate</b> and <b>clear</b> . The <b>exclude</b> argument excludes the file from truncation when a store and/or cleanup policy application occurs. The <b>immediate</b> argument truncates the file immediately when stored to a medium. The <b>policy</b> argument truncates the file by cleanup policy. The <b>clear</b> argument temporarily clears the indication that this file met truncate exclusion criteria defined in the <code>excludes.truncate</code> file.	-t
clean	Optional	1	This options indicates if the database entries are to be cleaned when the file is removed from the file system. Valid values are <b>true</b> and <b>false</b> . The <b>true</b> argument indicates that the database entries will be cleaned and the file will NOT be recoverable by the <b>fsrecover</b> command. The <b>false</b> argument indicates that the database entries will NOT be cleaned and the file will be recoverable by the <b>fsrecover</b> command.	-D
numcopies	Optional	1	Number of copies of the file(s) to be stored.	-c
policy	Optional	1	This specifies the class that will be associated with the file.	-a

Parameter	Req / Opt	Num	Description	CLI Option
stubsize	Optional	1	The truncation stub size (in kilobytes). This value is used to determine the number of bytes to leave on disk when files are truncated.	-S

### Example

```
https://<<SERVER>>/sws/v2/file/fschfiat  
?store=exclude  
&trunc=immediate  
&file=/stornext/snfs1/xxx/testfile.0  
&format=xml
```

### Output



```
<?xml version="1.0" encoding="UTF-8" ?>
<fschfiat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fschfiat.xsd">
<header>
<commandName>fschfiat</commandName>
<commandLine>/usr/adic/TSM/bin/fschfiat -s e -t i -D n -F xml
/stornext/snfs1/xxx/testfile.0</commandLine>
<commandDescription>Change file attributes for a set of file
names</commandDescription>
<localDateISO>2014-05-15T11:45:04</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:45:04</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:45:04Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:45:04</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0109</statusCode>
<statusNumber>109</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388039</requestId>
<commandName>/usr/adic/TSM/bin/fschfiat</commandName>
<commandStatus>interim</commandStatus>
<statusText>Command successful for /stornext/snfs1/xxx/testfile.0.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388039</requestId>
```

```
<commandName>/usr/adic/TSM/bin/fschfiat</commandName>
<commandStatus>completed</commandStatus>
<statusText>1 out of 1 file attribute changes were successful.</statusText>
</status>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388039</requestId>
<commandName>/usr/adic/TSM/bin/fschfiat</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:45:04</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:45:04</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:45:04Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:45:04</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0046</elapsedTimeInSeconds>
</footer>
</fschfiat>
```

## Job

### Job / Asynchronous Job Status

This web service returns a status of the jobs that are invoked asynchronously.

## Parameters

Parameter	Req / Opt	Num	Description	CLI Option
job	Required	N	The list of job ids for which status is being queried.	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	None

## Example

```
https://<<SERVER>>/sws/v2/job/info  
?job=27  
&format=json
```

## Output

```
{  
  "jobList": [  
    {  
      "jobInfo": {  
        "job": "27",  
        "state": "RUNNING"  
      }  
    }  
  ]  
}
```

## Job / Detailed Mover Job Status

This web service returns a detailed status of the mover jobs that are invoked asynchronously. This web service only supports **fsstore** and **fsretrieve** jobs.

## Parameters

Parameter	Req / Opt	Num	Description	CLI Option
job	Required	N	The list of job ids for which status is being queried.	None

## Example

```
https://<<SERVER>>/sws/v2/job/mover/info  
?job=26  
&job=27  
&job=28  
&job=29  
&job=30
```

## Output

```
[
  {
    "moverRequestList": [
      {
        "requestId": "177193",
        "requestType": "Retrieve",
        "state": "COPY",
        "positioninqueue": 1
      },
      {
        "requestId": "177194",
        "requestType": "Retrieve",
        "state": "READY",
        "positioninqueue": 2
      },
      {
        "requestId": "177205",
        "requestType": "Retrieve",
        "state": "READY",
        "positioninqueue": 3
      },
      {
        "requestId": "177213",
        "requestType": "Retrieve",
        "state": "READY",
        "positioninqueue": 4
      }
    ],
    "moverProgressList": [
      {
        "host": "REDHAT5-DEMO",
        "requestId": "177193",
        "deviceAlias": "archives_dr1",
        "runTime": "00:00:03",
        "totalFiles": "3",
      }
    ]
  }
]
```

```
        "filesCopied": "1",
        "filesFailed": "0"
    }
]
},
{
    "completedJobList": [
        {
            "header": {
                "commandName": "fsretrieve",
                "commandLine": "/usr/adic/TSM/bin/fsretrieve -F json
/stornext/snfx1/smp2data/soobar1 /stornext/snfx1/smp2data/soobar2
/stornext/snfx1/smp2data/soobar3 /stornext/snfx1/smp2data/soobar4
/stornext/snfx1/smp2data/soobar5",
                "commandDescription": "Retrieve files from media and place on
disk",
                "localDateISO": "2015-10-27T17:05:09",
                "localDate": "2015-10-27",
                "localTime": "17:05:09",
                "localDayOfWeek": 2,
                "gmtDateISO": "2015-10-27T23:05:09Z",
                "gmtDate": "2015-10-27",
                "gmtTime": "23:05:09",
                "gmtDayOfWeek": 2
            },
            "jobInfo": {
                "job": "24",
                "exitcode": 0,
                "datecompleted": "2015-10-27 17:05:20",
                "state": "COMPLETED"
            },
            "statuses": [
                {
                    "statusCode": "FS0005",
                    "statusNumber": 5,
                    "dayOfMonth": 27,
                    "requestId": 177191,
                }
            ]
        }
    ]
}
```

```
interim",
    "statusText": "File /stornext/snfx1/smp2data/soobar2 has
been retrieved."
  },
  {
    "statusCode": "FS0347",
    "statusNumber": 347,
    "dayOfMonth": 27,
    "requestId": 177191,
    "commandName": "fsretrieve",
    "commandStatus": "interim",
    "statusText": "File /stornext/snfx1/smp2data/soobar3 has
been retrieved."
  },
  {
    "statusCode": "FS0347",
    "statusNumber": 347,
    "dayOfMonth": 27,
    "requestId": 177191,
    "commandName": "fsretrieve",
    "commandStatus": "interim",
    "statusText": "File /stornext/snfx1/smp2data/soobar4 has
been retrieved."
  },
  {
    "statusCode": "FS0347",
    "statusNumber": 347,
    "dayOfMonth": 27,
    "requestId": 177191,
    "commandName": "fsretrieve",
    "commandStatus": "interim",
    "statusText": "File /stornext/snfx1/smp2data/soobar5 has
been retrieved."
  },
  {
    "statusCode": "FS0654",
    "statusNumber": 654,
    "dayOfMonth": 27,
    "requestId": 177191,
    "commandName": "fsretrieve",
    "commandStatus": "completed",
```

```
        "statusNumber": 347,  
        "dayOfMonth": 27,  
        "requestId": 177192,  
        "commandName": "fsretrieve",  
        "commandStatus": "interim",  
        "statusText": "File /stornext/snfx1/smp2data/foobar3 has  
been retrieved."  
    },  
    {  
        "statusCode": "FS0390",  
        "statusNumber": 390,  
        "dayOfMonth": 27,  
        "requestId": 177192,  
        "commandName": "fsretrieve",  
        "commandStatus": "completed",  
        "statusText": "3 out of 3 retrieves were successful."  
    }  
],  
"footer": {  
    "returnCode": 0,  
    "localDateISOEnd": "2015-10-27T17:05:26",  
    "localDateEnd": "2015-10-27",  
    "localTimeEnd": "17:05:26",  
    "localDayOfWeekEnd": 2,  
    "gmtDateISOEnd": "2015-10-27T23:05:26Z",  
    "gmtDateEnd": "2015-10-27",  
    "gmtTimeEnd": "23:05:26",  
    "gmtDayOfWeekEnd": 2,  
    "elapsedTimeInSeconds": "16.0902"  
}  
]  
},  
{  
    "pendingJobList": [  
        {  
            "jobInfo": {  
                "job": "30",  
                "positioninagentqueue": 1,  
                "exitcode": 0,  
                "state": "QUEUED"  
            }  
        }  
    ]  
}  
]
```



# Media

## Media / Information

Generate a report on media based on their current status.

This web service runs the **fsmediainfo** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
verbose	Optional	1	Produce the long form of the report that contains the same information as the short form, plus a list of the file segments on the medium. Default value is <b>false</b> .	-l
starttime	Optional	1	Used with the verbose option to indicate a start time of the files on the media to be reported. If this time and/or the end time are used with the verbose option then only the files on the media that were stored or recovered during the specified time window will be displayed. When this option or the <b>-e</b> is used, the output will be sorted per file system by the store/recovery time. The format for the time parameter is <b>YYYY:MM:DD:hh:mm:ss</b> .	-s
endtime	Optional	1	Used with the verbose option to indicate an end time of the files on the media to be reported. The format for the time parameter is <b>YYYY:MM:DD:hh:mm:ss</b> .	-e
media	Required	N	A media ID.	mediaID

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedinfo  
?verbose=true  
&starttime=2004:01:01:01:01:01  
&endtime=2013:01:01:01:01:01  
&media=E00001  
&media=E00002  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmedinfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsmedinfo.xsd">
<header>
<commandName>fsmedinfo</commandName>
<commandLine>/usr/adic/TSM/bin/fsmedinfo -l -s 2004:01:01:01:01:01 -e
2013:01:01:01:01:01 -F xml E00000 E00001</commandLine>
<commandDescription>Generate a report on media</commandDescription>
<localDateISO>2014-05-15T11:47:16</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:47:16</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:47:16Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:47:16</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<medias>
<media>
<mediaId>E00000</mediaId>
<copy>1</copy>
<mediaType>LTO</mediaType>
<storageArea>VolSub</storageArea>
<classId>sws_1400082516</classId>
<lastAccessed>15-may-2014 10:07:44</lastAccessed>
<mediaStatus>AVAIL</mediaStatus>
<writeProtect>N</writeProtect>
<markStatus>UNMARKED</markStatus>
<mediumLocation>SLOT/BIN</mediumLocation>
<formatted>Y</formatted>
<formatType>ANTF</formatType>
<numberOfSegments>10</numberOfSegments>
<externalLocation>N/A</externalLocation>
```

```
<bytesUsed>1453378</bytesUsed>
<spaceRemaining>5241831424</spaceRemaining>
<percentUsed>0.03</percentUsed>
<suspectCount>0</suspectCount>
<mountCount>5</mountCount>
<mediaClass>F0_LTO_DATA</mediaClass>
<currentState>On-line</currentState>
<assignment>Free</assignment>
<actionState>None</actionState>
<locationState>Archive</locationState>
<currentArchive>myarchive</currentArchive>
<pendingArchive>N/A</pendingArchive>
<importDate>12-may-2014 15:44:14</importDate>
<moveCount>10</moveCount>
<files />
</media>
<media>
<mediaId>E00001</mediaId>
<copy>0</copy>
<mediaType>LTO</mediaType>
<storageArea>VolSub</storageArea>
<classId>sws_1400082516</classId>
<lastAccessed>15-may-2014 10:02:43</lastAccessed>
<mediaStatus>AVAIL</mediaStatus>
<writeProtect>N</writeProtect>
<markStatus>UNMARKED</markStatus>
<mediumLocation>SLOT/BIN</mediumLocation>
<formatted>N</formatted>
<formatType>UNKNOWN</formatType>
<numberOfSegments>0</numberOfSegments>
<externalLocation>N/A</externalLocation>
<bytesUsed>0</bytesUsed>
```

```
<spaceRemaining>0</spaceRemaining>
<percentUsed>0.00</percentUsed>
<suspectCount>0</suspectCount>
<mountCount>4</mountCount>
<mediaClass>F0_LTO_DATA</mediaClass>
<currentState>On-line</currentState>
<assignment>Free</assignment>
<actionState>None</actionState>
<locationState>Archive</locationState>
<currentArchive>myarchive</currentArchive>
<pendingArchive>N/A</pendingArchive>
<importDate>12-may-2014 15:44:14</importDate>
<moveCount>8</moveCount>
</media>
</medias>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388043</requestId>
<commandName>/usr/adic/TSM/bin/fsmedinfo</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:47:16</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:47:16</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
```

```
<gmtDateISOEnd>2014-05-15T16:47:16Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>16:47:16</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0012</elapsedTimeInSeconds>  
</footer>  
</fsmedinfo>
```

## Media / File Background job status

This web service reports the status of the background job.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
requestid	Optional	1	The request id for the deletion requests. If no requestid is provided, all outstanding requests are included.	-C

### Example

```
https://<<SERVER>>/sws/v2/media/fsclean/background/status  
?requestid=177937  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8"?>
<fsclean xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsclean.xsd">
<header>
<commandName>fsclean</commandName>
<commandLine>/usr/adic/TSM/exec/fsclean -C 177937 -F xml</commandLine>
<commandDescription>Remove file version information from Tertiary Manager
knowledge.</commandDescription>
<localDateISO>2015-11-02T12:40:26</localDateISO>
<localDate>2015-11-02</localDate>
<localTime>12:40:26</localTime>
<localDayOfWeek>1</localDayOfWeek>
<gmtDateISO>2015-11-02T19:40:26Z</gmtDateISO>
<gmtDate>2015-11-02</gmtDate>
<gmtTime>19:40:26</gmtTime>
<gmtDayOfWeek>1</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>2</dayOfMonth>
<requestId>177943</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>2</dayOfMonth>
<requestId>177943</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2015-11-02T12:40:26</localDateISOEnd>
<localDateEnd>2015-11-02</localDateEnd>
<localTimeEnd>12:40:26</localTimeEnd>
<localDayOfWeekEnd>1</localDayOfWeekEnd>
```

```
<gmtDateISOEnd>2015-11-02T19:40:26Z</gmtDateISOEnd>
<gmtDateEnd>2015-11-02</gmtDateEnd>
<gmtTimeEnd>19:40:26</gmtTimeEnd>
<gmtDayOfWeekEnd>1</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0000</elapsedTimeInSeconds>
</footer>
</fsclean>
```

## Media / File Inactive Removal

The primary function of this web service is to remove inactive versions of files.

This web service runs the `fsclean` command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
fsmounts	Optional	N	One or more file systems for which to clean.	-s
policy	Optional	1	Policy class associated with data to clean.	-c
endtime	Optional	1	Endtime option. Time should be current or less than current time. If the time specified is greater than current time, current time will be used. The format for the time parameter is YYYY:MM:DD:hh:mm:ss.	-t
removedonly	Optional	1	Process inactive versions associated with removed files. Default value is false.	-d
purge	Optional	1	This option is used to indicate that a purge of the database namespace will be performed. Default value is false.	-P
media	Optional	N	A media ID.	mediaID

### Example



```
https://<<SERVER>>/sws/v2/media/fsclean  
?media=E00001  
&endtime=2013:01:01:01:01:01  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsclean xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsclean.xsd">
<header>
<commandName>fsclean</commandName>
<commandLine>/usr/adic/TSM/exec/fsclean -m E00001 -t 2013:01:01:01:01:01 -F
xml</commandLine>
<commandDescription>Remove file version information from Tertiary Manager
knowledge.</commandDescription>
<localDateISO>2014-05-15T09:55:32</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>09:55:32</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T14:55:32Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>14:55:32</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387924</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0723</statusCode>
<statusNumber>723</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387924</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
```

```

<commandStatus>completed</commandStatus>
<statusText>Nothing found to clean for request.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T09:55:32</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>09:55:32</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T14:55:32Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>14:55:32</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0003</elapsedTimeInSeconds>
</footer>
</fsclean>
    
```

## Media / File Inactive Remove by Media

Clean all Tertiary Manager knowledge of files on media that has been marked as logically blank.

This web service runs the `fsclean` command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

Parameter	Req / Opt	Num	Description	CLI Option
rminfomedia	Required	1	Media for which file version removal is required.	media ID
rminforetrieve	Optional	1	The filename file contains a list of files which need to be retrieved so that their missing copies can be regenerated.	-p
purge	Optional	1	This option is used to indicate that a purge of the database namespace will be performed. Default value is <b>false</b> .	-P

### Example

```
https://<<SERVER>>/sws/v2/media/fsclean/rminfo  
?rminfomedia=E00001  
&purge=false  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsclean xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsclean.xsd">
<header>
<commandName>fsclean</commandName>
<commandLine>/usr/adic/TSM/exec/fsclean -r E00001 -F xml</commandLine>
<commandDescription>Remove file version information from Tertiary Manager
knowledge.</commandDescription>
<localDateISO>2014-05-15T12:43:55</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:43:55</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:43:55Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:43:55</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388137</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0723</statusCode>
<statusNumber>723</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388137</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
```

```
<commandStatus>completed</commandStatus>
<statusText>Nothing found to clean for request.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:43:55</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:43:55</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T17:43:55Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>17:43:55</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0002</elapsedTimeInSeconds>
</footer>
</fsclean>
```

## Media / File Inactive Removal List

This will give a listing of all media which has been marked as logically blank.

This web service runs the `fsclean` command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

### Example

```
https://<<SERVER>>/sws/v2/media/fsclean/rminfo/list  
?format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsclean xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsclean.xsd">
<header>
<commandName>fsclean</commandName>
<commandLine>/usr/adic/TSM/exec/fsclean -r -l -F xml</commandLine>
<commandDescription>Remove file version information from Tertiary Manager
knowledge.</commandDescription>
<localDateISO>2014-05-15T09:57:17</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>09:57:17</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T14:57:17Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>14:57:17</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0589</statusCode>
<statusNumber>589</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387927</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
<commandStatus>interim</commandStatus>
<statusText>Tertiary Manager software request received.</statusText>
</status>
<status>
<statusCode>FS0723</statusCode>
<statusNumber>723</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387927</requestId>
<commandName>/usr/adic/TSM/exec/fsclean</commandName>
```



```

<commandStatus>completed</commandStatus>
<statusText>Nothing found to clean for request.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T09:57:17</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>09:57:17</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T14:57:17Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>14:57:17</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0002</elapsedTimeInSeconds>
</footer>
</fsclean>
    
```

## Media / Move Media

Move media from one archive to another.

This web service runs the **vsmove** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
media	Required	N	The media(s) that need(s) to be moved.	media ID

Parameter	Req / Opt	Num	Description	CLI Option
archive	Required	1	Specifies the name of the archive to which the specified media are to be moved.	-a
operator	Optional	1	Indicates this command is to be processed only if both the source and destination archives are operator-attended. Default value is <b>false</b> .	-i
wait	Optional	1	Indicates the Media Manager software waits until the command processing completes before returning status to the client. Default value is <b>false</b> .	-w
verbose	Optional	1	Indicates that verbose output is desired. Default value is <b>false</b> .	-v
hostname	Optional	1	The host name of the Media Manager server. The default host name is the host name of the computer where the CLI command is issued.	-H
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries the CLI software attempts if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout to the CLI software. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/media/vsmove
?media=E00008
&archive=myarchive
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<vsmove xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vsmove.xsd">
<header>
<commandName>vsmove</commandName>
<commandLine>/usr/adic/MSM/clibin/vsmove -a myarchive E00008 -F xml</commandLine>
<commandDescription>Move media from one archive to another.</commandDescription>
<localDateISO>2014-05-15T11:50:27</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:50:27</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:50:27Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:50:27</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<vsmoveOutput>
<vsmoveSuccess>
<vsmoveSuccessCompleted>0</vsmoveSuccessCompleted>
<vsmoveSuccessTotal>1</vsmoveSuccessTotal>
</vsmoveSuccess>
<errorMessage>
<errorMessageCode>VOL024</errorMessageCode>
<errorMessageText>error in the list</errorMessageText>
</errorMessage>
<mediaErrors>
<mediaError>
<mediaErrorMedium>E00008</mediaErrorMedium>
<mediaErrorText>medium already exists in target archive</mediaErrorText>
</mediaError>
</mediaErrors>
</vsmoveOutput>
```

```
<footer>
<returnCode>24</returnCode>
<localDateISOEnd>2014-05-15T11:50:27</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:50:27</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:50:27Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:50:27</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0051</elapsedTimeInSeconds>
</footer>
</vsmove>
```

## Media / Import Cleaning Media

This web service imports a list of cleaning media.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
mediaid	Required	N	The list of media IDs.	None

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedin/cleaningmedia
?mediaid=000025
&format=json
```

## Output

```
{
  "header": {
    "commandName": "fsmedin",
    "commandLine": "/usr/adic/TSM/exec/fsmedin -k -F json 000025",
    "commandDescription": "Logically enters media into the Tertiary Manager
system from the Media Manager system.",
    "localDateISO": "2015-11-02T12:47:31",
    "localDate": "2015-11-02",
    "localTime": "12:47:31",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-11-02T19:47:31Z",
    "gmtDate": "2015-11-02",
    "gmtTime": "19:47:31",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0627",
      "statusNumber": 627,
      "dayOfMonth": 2,
      "requestId": 177981,
      "commandName": "/usr/adic/TSM/exec/fsmedin",
      "commandStatus": "interim",
      "statusText": "Entering cleaning media..."
    },
    {
      "statusCode": "FS0574",
      "statusNumber": 574,
      "dayOfMonth": 2,
      "requestId": 177981,
      "commandName": "fsmedin",
      "commandStatus": "interim",
      "statusText": "Entering of medium 000025 failed. Reason: Query could
not find a match for given criteria."
    },
    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 2,
      "requestId": 177981,
      "commandName": "fsmedin",
      "commandStatus": "failed",
      "statusText": "0 out of 1 requests were successful."
    }
  ],
  "footer": {
    "returnCode": 1,
    "localDateISOEnd": "2015-11-02T12:47:31",
  }
}
```

```
"localDateEnd": "2015-11-02",  
"localTimeEnd": "12:47:31",  
"localDayOfWeekEnd": 1,  
"gmtDateISOEnd": "2015-11-02T19:47:31Z",  
"gmtDateEnd": "2015-11-02",  
"gmtTimeEnd": "19:47:31",  
"gmtDayOfWeekEnd": 1,  
"elapsedTimeInSeconds": "0.0043"  
}  
}
```

## Media / Import Media

This web service imports a list of media.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
mediaid	Required	N	The list of media IDs.	None
mediaformat	Optional	1	The media format that is requested for the media. The valid values are <b>ANTF</b> and <b>LTFS</b> .	-T
policyclass	Optional	1	The policy class this media will be associated.	-c
withholdformatting	Optional	1	This option is used to withhold formatting media immediately.	-w

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedin  
?mediaid=000017  
&format=json
```

### Output

```
{
  "header": {
    "commandName": "fsmedin",
    "commandLine": "/usr/adic/TSM/exec/fsmedin -b -F json 000017",
    "commandDescription": "Logically enters media into the Tertiary Manager
system from the Media Manager system.",
    "localDateISO": "2015-11-02T12:45:58",
    "localDate": "2015-11-02",
    "localTime": "12:45:58",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-11-02T19:45:58Z",
    "gmtDate": "2015-11-02",
    "gmtTime": "19:45:58",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0627",
      "statusNumber": 627,
      "dayOfMonth": 2,
      "requestId": 177962,
      "commandName": "/usr/adic/TSM/exec/fsmedin",
      "commandStatus": "interim",
      "statusText": "Entering media..."
    },
    {
      "statusCode": "FS0575",
      "statusNumber": 575,
      "dayOfMonth": 2,
      "requestId": 177962,
      "commandName": "fsmedin",
      "commandStatus": "interim",
      "statusText": "Entering of medium 000017 successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-11-02T12:45:58",
    "localDateEnd": "2015-11-02",
    "localTimeEnd": "12:45:58",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-11-02T19:45:58Z",
    "gmtDateEnd": "2015-11-02",
    "gmtTimeEnd": "19:45:58",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0080"
  }
}
```



```

    }
  },
  {
    "statusCode": "FS0627",
    "statusNumber": 627,
    "dayOfMonth": 2,
    "requestId": 177962,
    "commandName": "fsmedin",
    "commandStatus": "interim",
    "statusText": "Formatting media..."
  },
  {
    "statusCode": "FS0575",
    "statusNumber": 575,
    "dayOfMonth": 2,
    "requestId": 177962,
    "commandName": "fsmedin",
    "commandStatus": "interim",
    "statusText": "Formatting of medium 000017 successful."
  },
  {
    "statusCode": "FS0390",
    "statusNumber": 390,
    "dayOfMonth": 2,
    "requestId": 177962,
    "commandName": "fsmedin",
    "commandStatus": "completed",
    "statusText": "1 out of 1 requests were successful."
  }
],
"footer": {
  "returnCode": 0,
  "localDateISOEnd": "2015-11-02T12:45:59",
  "localDateEnd": "2015-11-02",
  "localTimeEnd": "12:45:59",
  "localDayOfWeekEnd": 1,
  "gmtDateISOEnd": "2015-11-02T19:45:59Z",
  "gmtDateEnd": "2015-11-02",
  "gmtTimeEnd": "19:45:59",
  "gmtDayOfWeekEnd": 1,
  "elapsedTimeInSeconds": "0.0379"
}
}

```

## Media / Export Media

This web service exports a list of media.

## Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
mediaid	Required	N	The list of media IDs.	None

## Example

```
https://<<SERVER>>/sws/v2/media/fsmedout  
?mediaid=000017  
&format=json
```

## Output

```
{
  "header": {
    "commandName": "fsmedout",
    "commandLine": "/usr/adic/TSM/exec/fsmedout -F json 000017",
    "commandDescription": "Logically removes media from the Tertiary Manager
system.",
    "localDateISO": "2015-11-02T12:48:23",
    "localDate": "2015-11-02",
    "localTime": "12:48:23",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-11-02T19:48:23Z",
    "gmtDate": "2015-11-02",
    "gmtTime": "19:48:23",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0575",
      "statusNumber": 575,
      "dayOfMonth": 2,
      "requestId": 177985,
      "commandName": "fsmedout",
      "commandStatus": "interim",
      "statusText": "Ejecting of medium 000017 successful."
    },
    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 2,
      "requestId": 177985,
      "commandName": "fsmedout",
      "commandStatus": "completed",
      "statusText": "1 out of 1 requests were successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-11-02T12:48:23",
    "localDateEnd": "2015-11-02",
    "localTimeEnd": "12:48:23",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-11-02T19:48:23Z",
    "gmtDateEnd": "2015-11-02",
    "gmtTimeEnd": "19:48:23",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0226"
  }
}
```

## Media / Copy Media

Initiates copy-replace for a media.

This web service runs the **fsmedcopy** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
media	Required	N	Media IDs of the media(s).	media ID
includeinactive	Optional	1	Indicates all files, active and inactive versions will be copied from the source medium. Default value is <b>false</b> .	-a
destination	Optional	1	Copy data file(s) to destination media.	-d
blank	Optional	1	Specify blank media for media destination. Default value is <b>false</b> .	-b
mediatype	Optional	1	Defines the type of medium to be used.	-t
runtime	Optional	1	Maximum allowable time in hours for the command to finish.	-u
drivepool	Optional	1	Media Manager drive pool group from which the destination drive will be selected when copying the specified media.	-v
checksum	Optional	1	Generate and maintain a checksum for each copied file. Valid values are <b>y</b> and <b>n</b> .	-G y or n

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedcopy/replace
?media=E00001
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmedcopy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsmedcopy.xsd">
<header>
<commandName>fsmedcopy</commandName>
<commandLine>/usr/adic/TSM/exec/fsmedcopy E00001 -r -G n -F xml</commandLine>
<commandDescription>Replace and/or defragment media.</commandDescription>
<localDateISO>2014-05-15T10:02:36</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>10:02:36</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T15:02:36Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>15:02:36</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0575</statusCode>
<statusNumber>575</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387943</requestId>
<commandName>fsmedcopy</commandName>
<commandStatus>interim</commandStatus>
<statusText>Copying files from medium E00001 successful.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387943</requestId>
<commandName>fsmedcopy</commandName>
```

```

<commandStatus>completed</commandStatus>
<statusText>1 out of 1 requests were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T10:02:42</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>10:02:42</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T15:02:42Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>15:02:42</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>5.0470</elapsedTimeInSeconds>
</footer>
</fsmedcopy>
</footer>
</fsclean>

```

## Media / Fragmentation Report

Report on media fragmentation.

This web service runs the **fsmedcopy** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
media	Required	N	The media IDs for which we need the fragmentation report.	media ID
fill	Optional	1	Fill level threshold between <b>0</b> and <b>100</b> percent. The percentage of the medium that has been written, including active and inactive file versions. A default of <b>0</b> percent is used if not specified.	-f
fragmentation	Optional	1	The percent (0 – 100) of wasted space out of the filled space on a medium. A default of <b>0</b> (zero) percent is used if not specified.	-w

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedcopy/fragreport  
?media=E00001  
&format=xml
```

### Output



```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmedcopy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsmedcopy.xsd">
<header>
<commandName>fsmedcopy</commandName>
<commandLine>/usr/adic/TSM/exec/fsmedcopy E00001 -F xml</commandLine>
<commandDescription>Replace and/or defragment media.</commandDescription>
<localDateISO>2014-05-15T11:52:30</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:52:30</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:52:30Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:52:30</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<media>
<medium>
<mediaId>E00001</mediaId>
<fillLevel>0.00</fillLevel>
<wastedSpace>0.00</wastedSpace>
<segmentCount>0</segmentCount>
<available>Y</available>
<vaulted>N</vaulted>
</medium>
</media>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388046</requestId>
<commandName>/usr/adic/TSM/exec/fsmedcopy</commandName>
```

```

<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T11:52:30</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:52:30</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:52:30Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:52:30</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0005</elapsedTimeInSeconds>
</footer>
</fsmedcopy>
    
```

## Media / Eject Media

Eject media out of an archive to be entered into another archive.

This web service runs the **vsarchiveeject** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
media	Required	N	Media IDs that need to be ejected.	mediaID

Parameter	Req / Opt	Num	Description	CLI Option
port	Optional	1	Specifies the import/export port the media is ejected to (if applicable).	-s
verbose	Optional	1	Indicates that verbose output is desired. Default value is <b>false</b> .	-v
hostname	Optional	1	The host name of the Media Manager server.	-H
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries that are attempted if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/media/vsarchiveject
?media=E00008
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<vsarchiveeject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vsarchiveeject.xsd">
<header>
<commandName>vsarchiveeject</commandName>
<commandLine>/usr/adic/MSM/clibin/vsarchiveeject E00008 -F xml</commandLine>
<commandDescription>Ejects media out of an archive to be entered into another
archive.</commandDescription>
<localDateISO>2014-05-15T11:53:41</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:53:41</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:53:41Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:53:41</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<vsarchiveejectOutput>
<vsarchiveejectReport>
<successfullyEjected>0</successfullyEjected>
<totalRequested>1</totalRequested>
</vsarchiveejectReport>
<errorMessage>
<errorMessageCode>VOL024</errorMessageCode>
<errorMessageText>error in the list</errorMessageText>
</errorMessage>
<mediaErrors>
<mediaError>
<mediaErrorMedium>E00008</mediaErrorMedium>
<mediaErrorText>port not available</mediaErrorText>
</mediaError>
</mediaErrors>
</vsarchiveejectOutput>
<footer>
```

```
<returnCode>24</returnCode>
<localDateISOEnd>2014-05-15T11:53:41</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:53:41</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:53:41Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:53:41</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0080</elapsedTimeInSeconds>
</footer>
</vsarchiveject>
```

## Media / Enter Media

Enter media that has been ejected out of an archive into another archive.

This web service runs the **vsarchiveenter** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
media	Required	N	Media IDs that need to be moved.	mediaID
port	Optional	1	Specifies the import/export port the media is entered from (if applicable).	-s
archive	Required	1	Specifies the name of the archive to which the specified media are to be moved.	-a
verbose	Optional	1	Indicates that verbose output is desired.	-v

Parameter	Req / Opt	Num	Description	CLI Option
hostname	Optional	1	The host name of the Media Manager server.	-H
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries that are attempted if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/media/vsarchiveenter  
?media=E00008  
&archive=myarchive  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<vsarchiveenter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="vsarchiveenter.xsd">
<header>
<commandName>vsarchiveenter</commandName>
<commandLine>/usr/adic/MSM/clibin/vsarchiveenter E00008 -a myarchive -F
xml</commandLine>
<commandDescription>Enters media that has been ejected out of an archive into
another archive.</commandDescription>
<localDateISO>2014-05-15T11:54:54</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:54:54</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:54:54Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:54:54</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<vsarchiveenterOutput>
<vsarchiveenterReport>
<successfullyEntered>0</successfullyEntered>
<totalRequested>1</totalRequested>
</vsarchiveenterReport>
<errorMessage>
<errorMessageCode>VOL024</errorMessageCode>
<errorMessageText>error in the list</errorMessageText>
</errorMessage>
<mediaErrors>
<mediaError>
<mediaErrorMedium>E00008</mediaErrorMedium>
<mediaErrorText>invalid action or location state for operation</mediaErrorText>
</mediaError>
</mediaErrors>
```

```
</vsarchiveenterOutput>
<footer>
<returnCode>24</returnCode>
<localDateISOEnd>2014-05-15T11:54:54</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:54:54</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:54:54Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:54:54</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0091</elapsedTimeInSeconds>
</footer>
</vsarchiveenter>
```

## Media / Purge Media

This web service purges a list of media.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
mediaid	Required	N	The list of media IDs.	None

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedout/purge
?mediaid=000017
&format=json
```



## Output

```
{
  "header": {
    "commandName": "fsmedout",
    "commandLine": "/usr/adic/TSM/exec/fsmedout -k -F json 000017",
    "commandDescription": "Logically removes media from the Tertiary Manager
system.",
    "localDateISO": "2015-11-02T12:45:24",
    "localDate": "2015-11-02",
    "localTime": "12:45:24",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-11-02T19:45:24Z",
    "gmtDate": "2015-11-02",
    "gmtTime": "19:45:24",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0575",
      "statusNumber": 575,
      "dayOfMonth": 2,
      "requestId": 177957,
      "commandName": "fsmedout",
      "commandStatus": "interim",
      "statusText": "Ejecting of medium 000017 successful."
    },
    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 2,
      "requestId": 177957,
      "commandName": "fsmedout",
      "commandStatus": "completed",
      "statusText": "1 out of 1 requests were successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-11-02T12:45:25",
    "localDateEnd": "2015-11-02",
    "localTimeEnd": "12:45:25",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-11-02T19:45:25Z",
    "gmtDateEnd": "2015-11-02",
    "gmtTimeEnd": "19:45:25",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0554"
  }
}
```

## Media / Attribute Query

Queries for the attributes of one or more specified media.

This web service runs the **vsmedqry** command.

**i Note:** This web service returns text output.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
media	Required	N	Specifies a media to be queried.	mediaID
hostname	Optional	1	The host name of the Media Manager server.	-H
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries that are attempted if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/media/vsmedqry  
?media=000001
```

### Output

```
-----  
Media Query Report 03-Jan-2014 11:58:23 1  
-----
```

```
Media ID: 000001  
-----
```

```
Media Type: LTO  
Media Class: F0_LTO_DATA  
Assignment: Free  
Location State: Archive  
Current Archive: mylib  
Pending Archive:  
Action State: Checkout  
Import Date: 15-Aug-2013 08:08:21  
Last Access: 11-Dec-2013 11:08:40  
Mount Count: 99  
Move Count: 198  
Manufacturer:  
Batch:  
Current State: On-line
```

## Media / Change Media State

Change the class or state of a media.

This web service runs the **fschmedstate** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
media	Required	N	One or more 16-character media identifiers.	mediaID
state	Optional	1	Media state. Valid values for states are as follows: <b>unsusp</b> : Reset error count associated with media to 0. <b>protect</b> : Mark media as write-protected. Files can be read, but no data can be written to the media. <b>unprotect</b> : Mark the media as unprotected for data storage on media. <b>avail</b> : Media are available for storage and retrieval. <b>unavail</b> : Media are unavailable to Tertiary Manager software. <b>unmark</b> : Unmark media that are marked Error, or Check out. Either the <b>state</b> or <b>policy</b> is required.	-s
policy	Optional	1	Change the policy class name of blank media. Either the <b>state</b> or <b>policy</b> is required.	-c
blank	Optional	1	Change policy class for a blank medium to system blank pool. Default value is <b>false</b> .	-b

### Example

```
https://<<SERVER>>/sws/v2/media/fschmedstate
?media=E00001
&state=avail
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fschmedstate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fschmedstate.xsd">
<header>
<commandName>fschmedstate</commandName>
<commandLine>/usr/adic/TSM/exec/fschmedstate E00001 -s avail -F xml</commandLine>
<commandDescription>Change the class or state of a medium.</commandDescription>
<localDateISO>2014-05-15T11:56:21</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>11:56:21</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T16:56:21Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>16:56:21</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0249</statusCode>
<statusNumber>249</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388051</requestId>
<commandName>/usr/adic/TSM/exec/fschmedstate</commandName>
<commandStatus>interim</commandStatus>
<statusText>Media E00001 is already in state avail.</statusText>
</status>
<status>
<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388051</requestId>
<commandName>/usr/adic/TSM/exec/fschmedstate</commandName>
```

```
<commandStatus>failed</commandStatus>
<statusText>0 out of 1 media state change requests were successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>1</returnCode>
<localDateISOEnd>2014-05-15T11:56:21</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>11:56:21</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T16:56:21Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>16:56:21</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0001</elapsedTimeInSeconds>
</footer>
</fschmedstate>
```

## Media / Checkout Media

Check media out of the Media Manager system.

This web service runs the **vscheckout** command.

**i Note:** This web service returns text output.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
media	Required	N	Specifies a list of one or more media to be queried.	mediaID
comment	Optional	1	Provide a comment to be associated with each checked-out media.	-t
verbose	Optional	1	Indicates that verbose output is needed. Default value is <b>false</b> .	-v
hostname	Optional	1	The host name of the Media Manager server.	-H

Parameter	Req / Opt	Num	Description	CLI Option
priority	Optional	1	The execution priority of the entered command. Assignable priority values are restricted to a range from <b>1</b> (highest) to <b>32</b> (lowest) inclusive. The default priority value is <b>15</b> .	-P
retries	Optional	1	The number of retries that are attempted if a timeout is returned by the API software. The default retries value is <b>3</b> .	-R
timeout	Optional	1	The amount of time (in seconds) the API software waits for status from the Media Manager software before returning a timeout. Total wait time for a command is (retries plus 1) multiplied by time-out value. The default time-out value is <b>120 seconds</b> .	-T
rpcnumber	Optional	1	The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <b>300016</b> .	-V

### Example

```
https://<<SERVER>>/sws/v2/media/vscheckout
?media=E00001
```

### Output

```
Check out 1 of 1 media was successful
```

## Media / List Media for Removal

List all media that are marked for removal.

This web service runs the **fsmedlist** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
policy	Optional	N	Policy class(es) for which the report is to be generated.	-c
blank	Optional	1	Used to report on the blank media in the general scratch pool. Default value is <b>false</b> .	-g
checkout	Optional	1	List all media marked for check out. Default value is <b>false</b> .	-lk

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedlist/removal  
?policy=myaxrpol1  
&format=xml
```

### Output



```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmedlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsmedlist.xsd">
<header>
<commandName>fsmedlist</commandName>
<commandLine>
/usr/adic/TSM/bin/fsmedlist -c myaxrpol1 -l -m -F xml
</commandLine>
<commandDescription>
Generate a list of media that have a specific status
</commandDescription>
<localDateISO>2014-07-01T16:01:10</localDateISO>
<localDate>2014-07-01</localDate>
<localTime>16:01:10</localTime>
<localDayOfWeek>2</localDayOfWeek>
<gmtDateISO>2014-07-01T22:01:10Z</gmtDateISO>
<gmtDate>2014-07-01</gmtDate>
<gmtTime>22:01:10</gmtTime>
<gmtDayOfWeek>2</gmtDayOfWeek>
</header>
<classes>
<class>
<classId>myaxrpol1</classId>
<marked>
<total>0</total>
</marked>
<totalMediaInClass>0</totalMediaInClass>
</class>
</classes>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>1</dayOfMonth>
<requestId>323000</requestId>
<commandName>/usr/adic/TSM/bin/fsmedlist</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
```

```
<localDateISOEnd>2014-07-01T16:01:10</localDateISOEnd>
<localDateEnd>2014-07-01</localDateEnd>
<localTimeEnd>16:01:10</localTimeEnd>
<localDayOfWeekEnd>2</localDayOfWeekEnd>
<gmtDateISOEnd>2014-07-01T22:01:10Z</gmtDateISOEnd>
<gmtDateEnd>2014-07-01</gmtDateEnd>
<gmtTimeEnd>22:01:10</gmtTimeEnd>
<gmtDayOfWeekEnd>2</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0002</elapsedTimeInSeconds>
</footer>
</fsmedList>
```

## Media / List Media by State

Lists media by state.

This web service runs the **fsmedList** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
policy	Optional	N	Policy class(es) for which the report is to be generated.	-c
blank	Optional	1	Used to report on the blank media in the general scratch pool. Default value is <b>false</b> .	-g
unformatted	Optional	1	List unformatted blank media. Default value is <b>false</b> .	-lb
formatted	Optional	1	List formatted blank media. Default value is <b>false</b> .	-lf

Parameter	Req / Opt	Num	Description	CLI Option
protect	Optional	1	List all write protected media. Default value is <b>false</b> .	-lp
error	Optional	1	List media with mark error. Default value is <b>false</b> .	-lq
available	Optional	1	List all available media. Default value is <b>false</b> .	-la
inaccessible	Optional	1	List all media unavailable to Tertiary Manager software. Default value is <b>false</b> .	-ln
unavailable	Optional	1	List all media marked as unavailable, but still located within the Tertiary Manager system. Default value is <b>false</b> .	-lu
suspect	Optional	1	List all suspect media. Default value is <b>false</b> .	-ls

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedlist/state  
?policy=myaxrp011  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmedlist xsi:noNamespaceSchemaLocation="fsmedlist.xsd">
<header>
<commandName>
fsmedlist
</commandName>
<commandLine>
/usr/adic/TSM/bin/fsmedlist -c myaxrpol1 -l -F xml
</commandLine>
<commandDescription>
Generate a list of media that have a specific status
</commandDescription>
<localDateISO>
2014-07-01T16:05:08
</localDateISO>
<localDate>
2014-07-01
</localDate>
<localTime>
16:05:08
</localTime>
<localDayOfWeek>
2
</localDayOfWeek>
<gmtDateISO>
2014-07-01T22:05:08Z
</gmtDateISO>
<gmtDate>
2014-07-01
</gmtDate>
<gmtTime>
22:05:08
</gmtTime>
<gmtDayOfWeek>
2
</gmtDayOfWeek>
</header>
<classes>
<class>
<classId>
myaxrpol1
</classId>
<inDrive>
```

```
<total>
0
</total>
</inDrive>
<inBin>
<total>
0
</total>
</inBin>
<inTransit>
<total>
0
</total>
</inTransit>
<exiting>
<total>
0
</total>
</exiting>
<outOfStorNext>
<total>
0
</total>
</outOfStorNext>
<markedForCheckOut>
<total>
0
</total>
</markedForCheckOut>
<markError>
<total>
0
</total>
</markError>
<marked>
<total>
0
</total>
</marked>
<total>
0
</total>
</formattedBlank>
```

```
<unformattedBlank>
<total>
0
</total>
</unformattedBlank>
<writeProtected>
<total>
0
</total>
</writeProtected>
<suspect>
<total>
0
</total>
</suspect>
<available>
<total>
0
</total>
</available>
<unavailable>
<total>
0
</total>
</unavailable>
<unavailableToTertiaryManager>
<total>
0
</total>
</unavailableToTertiaryManager>
<totalMediaInClass>
0
</totalMediaInClass>
</class>
</classes>
<statuses>
<status>
<statusCode>
FS0000
</statusCode>
<statusNumber>
0
```

```
</statusNumber>
<dayOfMonth>
1
</dayOfMonth>
<requestId>
323010
</requestId>
<commandName>
/usr/adic/TSM/bin/fsmedlist
</commandName>
<commandStatus>
completed
</commandStatus>
<statusText>
Command Successful.
</statusText>
</status>
</statuses>
<footer>
<returnCode>
0
</returnCode>
<localDateISOEnd>
2014-07-01T16:05:08
</localDateISOEnd>
<localDateEnd>
2014-07-01
</localDateEnd>
<localTimeEnd>
16:05:08
</localTimeEnd>
<localDayOfWeekEnd>
2
</localDayOfWeekEnd>
<gmtDateISOEnd>
2014-07-01T22:05:08Z
</gmtDateISOEnd>
<gmtDateEnd>
2014-07-01
</gmtDateEnd>
<gmtTimeEnd>
22:05:08
</gmtTimeEnd>
```

```
<gmtDayOfWeekEnd>
2
</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>
0.0001
</elapsedTimeInSeconds>
</footer>
</fsmedlist>
```

## Media / List Media by Location

Lists media by location.

This web service runs the **fsmedlist** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
policy	Optional	N	Policy class(es) for which the report is to be generated.	-c
blank	Optional	1	Used to report on the blank media in the general scratch pool. Default value is <b>false</b> .	-g
drive	Optional	1	List all media located in a drive. Default value is <b>false</b> .	-ld
home	Optional	1	List all media located in their home slot/bin. Default value is <b>false</b> .	-lh
transit	Optional	1	List all media transitioning between locations. Default value is <b>false</b> .	-lt
exit	Optional	1	List all media exiting a storage area. Default value is <b>false</b> .	-lz
checkout	Optional	1	List all media that are checked out of the storage areas. Default value is <b>false</b> .	-lo



## Example

```
https://<<SERVER>>/sws/v2/media/fsmedlist/location  
?policy=myaxrp01  
&drive=true  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmedlist xsi:noNamespaceSchemaLocation="fsmedlist.xsd">
<header>
<commandName>
fsmedlist
</commandName>
<commandLine>
/usr/adic/TSM/bin/fsmedlist -c myaxrpol1 -l -d -F xml
</commandLine>
<commandDescription>
Generate a list of media that have a specific status
</commandDescription>
<localDateISO>
2014-07-01T16:08:05
</localDateISO>
<localDate>
2014-07-01
</localDate>
<localTime>
16:08:05
</localTime>
<localDayOfWeek>
2
</localDayOfWeek>
<gmtDateISO>
2014-07-01T22:08:05Z
</gmtDateISO>
<gmtDate>
2014-07-01
</gmtDate>
<gmtTime>
22:08:05
</gmtTime>
<gmtDayOfWeek>
2
</gmtDayOfWeek>
</header>
<classes>
<class>
<classId>
myaxrpol1
</classId>
<inDrive>
```

```
<total>
0
</total>
</inDrive>
<totalMediaInClass>
0
</totalMediaInClass>
</class>
</classes>
<statuses>
<status>
<statusCode>
FS0000
</statusCode>
<statusNumber>
0
</statusNumber>
<dayOfMonth>
1
</dayOfMonth>
<requestId>
323016
</requestId>
<commandName>
/usr/adic/TSM/bin/fsmedlist
</commandName>
<commandStatus>
completed
</commandStatus>
<statusText>
Command Successful.
</statusText>
</status>
</statuses>
<footer>
<returnCode>
0
</returnCode>
<localDateISOEnd>
2014-07-01T16:08:05
</localDateISOEnd>
<localDateEnd>
2014-07-01
```

```
</localDateEnd>  
<localTimeEnd>  
16:08:05  
</localTimeEnd>  
<localDayOfWeekEnd>  
2  
</localDayOfWeekEnd>  
<gmtDateISOEnd>  
2014-07-01T22:08:05Z  
</gmtDateISOEnd>  
<gmtDateEnd>  
2014-07-01  
</gmtDateEnd>  
<gmtTimeEnd>  
22:08:05  
</gmtTimeEnd>  
<gmtDayOfWeekEnd>  
2  
</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>  
0.0001  
</elapsedTimeInSeconds>  
</footer>  
</fsmedList>
```

## Object Storage

### Report Object Storage Components

Report all Object Storage components in Quantum storage system.

This web service runs the **fsobjcfg** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
action	Required	1	The only valid value is <b>list</b> . It generates a report showing all Quantum Object Storage components that are currently configured.	no CLI option.

### Example

```
https://<<SERVER>>/sws/v2/objectstorage/lattus/fsobjcfg  
?action=list  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsobjcfg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsobjcfg.xsd">
<header>
<commandName>fsobjcfg</commandName>
<commandLine>/usr/adic/TSM/exec/fsobjcfg -F xml</commandLine>
<commandDescription>Object Storage configuration</commandDescription>
<localDateISO>2014-05-15T12:03:43</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:03:43</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:03:43Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:03:43</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<configurationReport>
<appliance>
<alias>MDH1</alias>
<type>appliance</type>
<maxStreams>0</maxStreams>
<protocol>http</protocol>
<restapi />
<urlstyle />
<ipAddress>10.65.166.61</ipAddress>
<port>80</port>
<provider>LATTUS</provider>
<nodes>
<node>
<alias>C1</alias>
<type>node</type>
<maxStreams>48</maxStreams>
<protocol />
```

```
<restapi />
<urlstyle />
<ipAddress />
<port />
<provider />
<iopaths>
<iopath>
<alias>P1</alias>
<type>iopath</type>
<maxStreams>0</maxStreams>
<protocol>http</protocol>
<restapi>AXR</restapi>
<urlstyle>PATH</urlstyle>
<ipAddress>10.65.166.62:8080</ipAddress>
<port />
<provider />
</iopath>
</iopaths>
</node>
<node>
<alias>C2</alias>
<type>node</type>
<maxStreams>48</maxStreams>
<protocol />
<restapi />
<urlstyle />
<ipAddress />
<port />
<provider />
<iopaths>
<iopath>
<alias>P3</alias>
```

```
<type>iopath</type>
<maxStreams>0</maxStreams>
<protocol>http</protocol>
<restapi>AXR</restapi>
<urlstyle>PATH</urlstyle>
<ipAddress>10.65.166.65:8080</ipAddress>
<port />
<provider />
</iopath>
</iopaths>
</node>
</nodes>
</appliance>
</configurationReport>
<mediaReport>
<appliance>
<alias>MDH1</alias>
<medias>
<media>
<namespace>george1</namespace>
<mediaId>george1</mediaId>
<copy>1</copy>
<mediatype>AXR</mediatype>
</media>
</medias>
</appliance>
</mediaReport>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388056</requestId>
```



```
<commandName>/usr/adic/TSM/exec/fsobjcfg</commandName>  
<commandStatus>completed</commandStatus>  
<statusText>Command Successful.</statusText>  
</status>  
</statuses>  
<footer>  
<returnCode>0</returnCode>  
<localDateISOEnd>2014-05-15T12:03:43</localDateISOEnd>  
<localDateEnd>2014-05-15</localDateEnd>  
<localTimeEnd>12:03:43</localTimeEnd>  
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:03:43Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:03:43</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0004</elapsedTimeInSeconds>  
</footer>  
</fsobjcfg>
```

## Policy

### Policy / Class Information

This web service reports information on storage manager policies.

This web service runs the **fsclassinfo** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None

Parameter	Req / Opt	Num	Description	CLI Option
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
policy	Optional	N	Policy class(es) for which the report is to be generated. If no policy classes are specified, a short report is displayed for all policy classes.	policy name
verbose	Optional	1	List information in the long report format. Default value is <b>false</b> .	-l

### Example

```
https://<<SERVER>>/sws/v2/policy/fsclassinfo  
?policy=xxx  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsclassinfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsclassinfo.xsd">
<header>
<commandName>fsclassinfo</commandName>
<commandLine>/usr/adic/TSM/bin/fsclassinfo xxx -F xml</commandLine>
<commandDescription>Report policy class processing parameters, associated
directory paths, and affinity lists.</commandDescription>
<localDateISO>2014-05-15T12:05:02</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:05:02</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:05:02Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:05:02</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<classes>
<class>
<classId>xxx</classId>
<softLimit>20000</softLimit>
<hardLimit>25000</hardLimit>
<drivePool>fs_F0drivepool</drivePool>
<securityCode>NONE</securityCode>
<acctNumber>12345</acctNumber>
<defCopies>1</defCopies>
<maxCopies>4</maxCopies>
<maxInactiveVersions>10</maxInactiveVersions>
<mediaType>LTO</mediaType>
<fileCleanup>MINTIME</fileCleanup>
<mediaCleanup>SYSTEM</mediaCleanup>
<storeMinTime>5m</storeMinTime>
<storeMaxSetAge>n/a</storeMaxSetAge>
```

```
<storeMinSetSize>n/a</storeMinSetSize>
<storeAutomatically>yes</storeAutomatically>
<relocMinTime>7d</relocMinTime>
<truncMinTime>3d</truncMinTime>
<generateChecksum>DISABLED</generateChecksum>
<validateChecksum>DISABLED</validateChecksum>
<cleanOnRemove>DISABLED</cleanOnRemove>
<targetStubSize>0</targetStubSize>
</class>
</classes>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388058</requestId>
<commandName>/usr/adic/TSM/bin/fsclassinfo</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:05:02</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:05:02</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T17:05:02Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>17:05:02</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0002</elapsedTimeInSeconds>
```

```
</footer>  
</fsclassinfo>
```

## Policy / Modify a Policy

Modify the processing parameters of a policy class.

This web service runs the **fsmodclass** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
policy	Required	1	The policy class for which the report is to be generated.	policy class
softlimit	Optional	1	The warning limit for the number of media that can be allocated to this policy class.	-s
hardlimit	Optional	1	The maximum number of media that are allowed in this policy class.	-h
stubsizes	Optional	1	The truncation stub size (in kilobytes).	-S
mediatype	Optional	1	Defines the type of medium to be used.	-t
security	Optional	1	Up to four characters can be used for security code.	-l
account	Optional	1	Up to five characters can be used for the account number.	-o
maxcopies	Optional	1	The maximum number of copies (including the primary copy) that are allowed for each file in this policy class.	-x

Parameter	Req / Opt	Num	Description	CLI Option
defaultcopies	Optional	1	The total number of copies that will be stored (including the primary copy) for each file in this policy class.	-d
minstoretime	Optional	1	The minimum time that a file must reside unmodified on disk before being considered a candidate for storage on media.	-m
mintruncetime	Optional	1	The minimum time that a stored file must reside unaccessed on disk before being considered a candidate for truncation (the clearing of disk blocks).	-c
minreloctime	Optional	1	The minimum time that a stored file must reside unaccessed on disk before being considered a candidate for relocation (the clearing of disk blocks).	-i
affinity	Optional	1	A space-separated list of disk affinities that the files in this policy class will traverse throughout their life cycle.	-a
retrieveaffinity	Optional	1	The affinity to retrieve a truncated file to.	-R
drivepool	Optional	1	The Media Manager drive pool group used to store or retrieve data for this policy class.	-v
maxversions	Optional	1	This is the maximum number of inactive versions to keep for a file (the current version is active, all others are inactive).	-k
truncmode	Optional	1	Truncation mode. Valid values are <b>immediate</b> and <b>policy</b> .	-f
blankpool	Optional	1	Media classification cleanup action. Valid values are policy and system.	-r
autostore	Optional	1	This option decides if we allow the policy engine to automatically store files for this policy class. Valid values are <b>true</b> and <b>false</b> .	-p
minsetsize	Optional	1	The minimum set size of the policy class.	-z
maxsetage	Optional	1	Candidate expiration time (in hours) of the policy class.	-g

Parameter	Req / Opt	Num	Description	CLI Option
copy	Optional	1	Used to apply the copy options to a specific copy.	-C
drivelimit	Optional	1	The maximum number of drives to use when the policy is run.	-L
checksumstore	Optional	1	Generate and maintain a checksum for each stored file. Valid values are <b>true</b> and <b>false</b> .	-G
checksumretrieve	Optional	1	Verify the checksum of each retrieved file. Valid values are <b>true</b> and <b>false</b> .	-V
dbclean	Optional	1	Remove database information when a file is removed. Valid values are <b>true</b> and <b>false</b> .	-D
mediaformattype	Optional	1	Specifies the media format type that will be used when formatting or selecting media.	-T

### Example

```
https://<<SERVER>>/sws/v2/policy/fsmodclass  
?policy=xxx  
&defaultcopies=3  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsmodclass xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsmodclass.xsd">
<header>
<commandName>fsmodclass</commandName>
<commandLine>/usr/adic/TSM/exec/fsmodclass xxx -d 3 -p no -G n -V n -D n -F
xml</commandLine>
<commandDescription>Modify the processing parameters of a policy
class.</commandDescription>
<localDateISO>2014-05-15T12:06:20</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:06:20</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:06:20Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:06:20</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388059</requestId>
<commandName>/usr/adic/TSM/exec/fsmodclass</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful. Warning: AUTOSTORE is off for the policy class
and there is no spolicy schedule for it.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:06:20</localDateISOEnd>
```



```
<localDateEnd>2014-05-15</localDateEnd>  
<localTimeEnd>12:06:20</localTimeEnd>  
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:06:20Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:06:20</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0022</elapsedTimeInSeconds>  
</footer>  
</fsmoduleclass>
```

## Policy / Report Policy Class For A Directory

This web service reports the policy class associated with directory.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
directory	Required	1	The directory for which the policy class is required.	None

### Example

```
https://<<SERVER>>/sws/v2/policy/fsdirclass  
?directory=/stornext/snfx1/smp1data  
&format=json
```

### Output

```
{
  "header": {
    "commandName": "fsdirclass",
    "commandLine": "/usr/adic/TSM/bin/fsdirclass /stornext/snfx1/smpldata -F
json",
    "commandDescription": "Report the policy class associated with a
directory.",
    "localDateISO": "2015-11-02T12:50:56",
    "localDate": "2015-11-02",
    "localTime": "12:50:56",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-11-02T19:50:56Z",
    "gmtDate": "2015-11-02",
    "gmtTime": "19:50:56",
    "gmtDayOfWeek": 1
  },
  "directories": [
    {
      "directoryName": "/stornext/snfx1/smpldata",
      "classId": "smpl"
    }
  ],
  "statuses": [
    {
      "statusCode": "FS0000",
      "statusNumber": 0,
      "dayOfMonth": 2,
      "requestId": 177998,
      "commandName": "/usr/adic/TSM/bin/fsdirclass",
      "commandStatus": "completed",
      "statusText": "Command Successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-11-02T12:50:56",
    "localDateEnd": "2015-11-02",
    "localTimeEnd": "12:50:56",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-11-02T19:50:56Z",
    "gmtDateEnd": "2015-11-02",
    "gmtTimeEnd": "19:50:56",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0003"
  }
}
```

## Policy / Report Policy Class For A File System

This web service reports all policy classes with association points in a file system.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
filesystem	Required	1	The file system name on which to generate policy class relation information.	None

### Example

```
https://<<SERVER>>/sws/v2/policy/fsgetclasses  
?filesystem=/stornext/snfx1  
&format=json
```

### Output

```
{
  "header": {
    "commandName": "fsgetclasses",
    "commandLine": "/usr/adic/TSM/bin/fsgetclasses /stornext/snfx1 -F json",
    "commandDescription": "Report all policy classes with association points
in a file system.",
    "localDateISO": "2015-11-02T12:52:04",
    "localDate": "2015-11-02",
    "localTime": "12:52:04",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-11-02T19:52:04Z",
    "gmtDate": "2015-11-02",
    "gmtTime": "19:52:04",
    "gmtDayOfWeek": 1
  },
  "filesystems": [
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "smp1"
    },
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "_adic_backup"
    },
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "smp2"
    },
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "smp3"
    },
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "smp4"
    },
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "smp67"
    },
    {
      "filesystemName": "/stornext/snfx1",
      "classId": "smp78"
    },
    {
      "filesystemName": "/stornext/snfx1",

```

```
        "classId": "smpqcc"
      },
      {
        "filesystemName": "/stornext/snfx1",
        "classId": "smpltomedia"
      }
    ],
    "statuses": [
      {
        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 2,
        "requestId": 178002,
        "commandName": "/usr/adic/TSM/bin/fsgetclasses",
        "commandStatus": "completed",
        "statusText": "Command Successful."
      }
    ],
    "footer": {
      "returnCode": 0,
      "localDateISOEnd": "2015-11-02T12:52:04",
      "localDateEnd": "2015-11-02",
      "localTimeEnd": "12:52:04",
      "localDayOfWeekEnd": 1,
      "gmtDateISOEnd": "2015-11-02T19:52:04Z",
      "gmtDateEnd": "2015-11-02",
      "gmtTimeEnd": "19:52:04",
      "gmtDayOfWeekEnd": 1,
      "elapsedTimeInSeconds": "0.0012"
    }
  }
}
```

## Quota

### Quota / Manage Quotas

This web service manipulates the quota system in the StorNext file system.

This web service runs the **snquota** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-o
fsname	Required	1	Specify <b>FileSystemName</b> as the file system to manipulate.	-F
path	Optional	1	Specify the file system containing Path as the file system to manipulate.	-P
action	Required	1	Specify the action. Valid values are <b>create</b> , <b>delete</b> , <b>list</b> , <b>listall</b> , <b>mark</b> , <b>rebuild</b> , and <b>set</b> .	create = -C delete = -D list = -G listall = -L mark = -M rebuild = -R set = -S
absolute	Optional	1	Specifies if the path is absolute. Default value is <b>false</b> .	-a
directory	Optional	1	This option specifies a DQNS on a StorNext file system to be used with the actions.	-d
exact	Optional	1	When used with the <b>list</b> or <b>listall</b> actions, numbers will be printed as exact values.	-e
files	Optional	1	This option is only useful with the list and set actions and the directory option. When this option is present, limits and values represent the number of regular files contained in the DQNS.	-f
groupname	Optional	1	This option specifies the name of a group to get or set with the list or set action.	-g

Parameter	Req / Opt	Num	Description	CLI Option
hostname	Optional	1	Use a hostname in a StorNext cluster that is different from the cluster the command is being run on. This option is rarely needed.	-H
hardlimit	Optional	1	This option specifies a hard limit to set when used with the set action.	-h
softlimit	Optional	1	This option specifies a soft limit to set when used with the set action.	-s
graceperiod	Optional	1	This option specifies a grace period to set when used with the set action.	-t
user	Optional	1	This option specifies the name of a user to get or set with the list or set action.	-u

Usage and Limits are printed in a human-readable form, suffixed with "K", "M", "G", "T", or "P" for kilobytes, megabytes, gigabytes, terabytes, or petabytes (respectively).

These are base-2 values (in other words, 1K = 1024). A value without a suffix is in bytes.

### Example

```
https://<<SERVER>>/sws/v2/quota/snquota  
?fsname=snfs1  
&action=listall  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<snquota xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="snquota.xsd">
<userQuotas>
<quota>
<hardLimit>0</hardLimit>
<softLimit>0</softLimit>
<gracePeriod>0</gracePeriod>
<curSize>327M</curSize>
<status>NoLimit</status>
<type>user</type>
<name>root</name>
</quota>
</userQuotas>
<groupQuotas>
<quota>
<hardLimit>0</hardLimit>
<softLimit>0</softLimit>
<gracePeriod>0</gracePeriod>
<curSize>327M</curSize>
<status>NoLimit</status>
<type>group</type>
<name>root</name>
</quota>
</groupQuotas>
<directoryQuotas>
<quota>
<hardLimit>5.0T</hardLimit>
<softLimit>2.0T</softLimit>
<gracePeriod>2w</gracePeriod>
<curSize>0</curSize>
<status>Under</status>
```



```
<type>dir</type>
<name>/foobar</name>
</quota>
<quota>
<hardLimit>0</hardLimit>
<softLimit>0</softLimit>
<gracePeriod>0</gracePeriod>
<curSize>0</curSize>
<status>NoLimit</status>
<type>dirfiles</type>
<name>/foobar</name>
</quota>
</directoryQuotas>
<returnCode>0</returnCode>
</snquota>
```

## Report

### Report / Cancel Requests

This web service cancels requests.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
requestid	Required	1	The request id that needs to be canceled.	-r

## Example

```
https://<<SERVER>>/sws/v2/report/fscancel  
?request=177397
```

## Output

```
{  
  "header": {  
    "commandName": "fscancel",  
    "commandLine": "fscancel 177937 -F json",  
    "commandDescription": "Cancel media, file, and resource queued  
requests",  
    "localDateISO": "2015-11-02T12:58:29",  
    "localDate": "2015-11-02",  
    "localTime": "12:58:29",  
    "localDayOfWeek": 1,  
    "gmtDateISO": "2015-11-02T19:58:29Z",  
    "gmtDate": "2015-11-02",  
    "gmtTime": "19:58:29",  
    "gmtDayOfWeek": 1  
  },  
  "statuses": [  
    {  
      "statusCode": "FS0000",  
      "statusNumber": 0,  
      "dayOfMonth": 2,  
      "requestId": 178014,  
      "commandName": "fscancel",  
      "commandStatus": "completed",  
      "statusText": "Command Successful."  
    }  
  ],  
  "footer": {  
    "returnCode": 0,  
    "localDateISOEnd": "2015-11-02T12:58:29",  
    "localDateEnd": "2015-11-02",  
    "localTimeEnd": "12:58:29",  
    "localDayOfWeekEnd": 1,  
    "gmtDateISOEnd": "2015-11-02T19:58:29Z",  
    "gmtDateEnd": "2015-11-02",  
    "gmtTimeEnd": "19:58:29",  
    "gmtDayOfWeekEnd": 1,  
    "elapsedTimeInSeconds": "0.0000"  
  }  
}
```

## Report / Files

Reports all files in the queue or specific files if a request identifiers or filename is specified.

This web service runs the **fsqueue** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
request	Optional	1	The request identifier of the request to be reported.	-r
file	Optional	N	The absolute path of the file(s).	-f

### Example

```
https://<<SERVER>>/sws/v2/report/fsqueue/file  
?file=/stornext/snfs1/xxx/testfile.0  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsqueue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsqueue.xsd">
<header>
<commandName>fsqueue</commandName>
<commandLine>/usr/adic/TSM/exec/fsqueue -f /stornext/snfs1/xxx/testfile.0 -F
xml</commandLine>
<commandDescription>View subsystem resource requests</commandDescription>
<localDateISO>2014-05-15T12:11:25</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:11:25</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:11:25Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:11:25</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388061</requestId>
<commandName>fsqueue</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful. There are no requests found.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:11:25</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:11:25</localTimeEnd>
```

```
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:11:25Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:11:25</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0001</elapsedTimeInSeconds>  
</footer>  
</fsqueue>
```

## Report / Media

Reports the media movement for a request identifier or all media in queue.

This web service runs the **fsqueue** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
request	Optional	1	The request identifier of the request to be reported.	-r

### Example

```
https://<<SERVER>>/sws/v2/report/fsqueue/media  
?request=1928784996  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsqueue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsqueue.xsd">
<header>
<commandName>fsqueue</commandName>
<commandLine>/usr/adic/TSM/exec/fsqueue -m -r 1928784996 -F xml</commandLine>
<commandDescription>View subsystem resource requests</commandDescription>
<localDateISO>2014-05-15T12:13:06</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:13:06</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:13:06Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:13:06</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388064</requestId>
<commandName>fsqueue</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful. There are no requests found.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:13:06</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:13:06</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
```

```
<gmtDateISOEnd>2014-05-15T17:13:06Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:13:06</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0000</elapsedTimeInSeconds>  
</footer>  
</fsqueue>
```

## Report / Mover Host

Active mover host summary.

This web service runs the **fsqueue** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

### Example

```
https://<<SERVER>>/sws/v2/report/fsqueue/moverhost  
?format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsqueue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsqueue.xsd">
<header>
<commandName>fsqueue</commandName>
<commandLine>/usr/adic/TSM/exec/fsqueue -a -F xml</commandLine>
<commandDescription>View subsystem resource requests</commandDescription>
<localDateISO>2014-05-15T12:14:17</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:14:17</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:14:17Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:14:17</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<movers>
<move>
<host>gp-rh63.mdh.quantum.com</host>
<state>Enabled</state>
<activeDataMovers>0</activeDataMovers>
</move>
</movers>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388066</requestId>
<commandName>fsqueue</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
```



```
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:14:17</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:14:17</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T17:14:17Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>17:14:17</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0001</elapsedTimeInSeconds>
</footer>
</fsqueue>
```

## Report / Mover Request

Active mover request summary.

This web service runs the **fsqueue** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

### Example

```
https://<<SERVER>>/sws/v2/report/fsqueue/moverrequest
?format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsqueue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsqueue.xsd">
<header>
<commandName>fsqueue</commandName>
<commandLine>/usr/adic/TSM/exec/fsqueue -a -v -F xml</commandLine>
<commandDescription>View subsystem resource requests</commandDescription>
<localDateISO>2014-05-15T12:15:07</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:15:07</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:15:07Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:15:07</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388068</requestId>
<commandName>fsqueue</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful. There are no requests found.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:15:07</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:15:07</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
```

```
<gmtDateISOEnd>2014-05-15T17:15:07Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:15:07</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0007</elapsedTimeInSeconds>  
</footer>  
</fsqueue>
```

## Report / Resource

Active resource request summary.

This web service runs the **fsqueue** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
request	Optional	1	The request identifier of the request to be reported.	-r

### Example

```
https://<<SERVER>>/sws/v2/report/fsqueue/resource  
?request=1928784996  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsqueue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsqueue.xsd">
<header>
<commandName>fsqueue</commandName>
<commandLine>/usr/adic/TSM/exec/fsqueue -r 1928784996 -F xml</commandLine>
<commandDescription>View subsystem resource requests</commandDescription>
<localDateISO>2014-05-15T12:17:05</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:17:05</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:17:05Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:17:05</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388071</requestId>
<commandName>fsqueue</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful. There are no requests found.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:17:05</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:17:05</localTimeEnd>
```

```
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:17:05Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:17:05</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0000</elapsedTimeInSeconds>  
</footer>  
</fsqueue>
```

## Schedule

### Schedule / Report

Prints the report of a feature or schedule.

This web service runs the `fsschedule` command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
feature	Optional	1	The name of the feature. Valid types are: <b>clninfo</b> , <b>clnver</b> , <b>defrag</b> , <b>rebuild</b> , <b>p_backup</b> , <b>f_backup</b> , <b>spolicy</b> , <b>healthck</b> , <b>activevault</b> , and <b>archive_cmp</b> .	-f
schedule	Optional	1	The schedule name.	-n

#### Example

```
https://<<SERVER>>/sws/v2/schedule/fsschedule/info  
?feature=defrag  
&schedule=schedule1  
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsschedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsschedule.xsd">
<header>
<commandName>fsschedule</commandName>
<commandLine>/usr/adic/TSM/exec/fsschedule -n schedule1 -F xml</commandLine>
<commandDescription>Insert, modify, delete, reset, or report all maintenance
features in the Quantum storage subsystem.</commandDescription>
<localDateISO>2014-05-15T12:21:43</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:21:43</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:21:43Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:21:43</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<schedules>
<schedule>
<name>schedule1</name>
<feature>defrag</feature>
<period>daily</period>
<lastRun>None</lastRun>
<lastRunISODate>None</lastRunISODate>
<lastRunTime>None</lastRunTime>
<calendar>
<dailyflags>YYYYYYY</dailyflags>
</calendar>
<startWindowStartHr>00</startWindowStartHr>
<startWindowStartMin>00</startWindowStartMin>
<startWindowEndHr>01</startWindowEndHr>
<startWindowEndMin>00</startWindowEndMin>
<lastRunStatus>None</lastRunStatus>
</schedule>
```

```
</schedules>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388079</requestId>
<commandName>/usr/adic/TSM/exec/fsschedule</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:21:43</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:21:43</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T17:21:43Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>17:21:43</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0001</elapsedTimeInSeconds>
</footer>
</fsschedule>
```

## Schedule / Create

This web service will create a schedule.

This web service runs the **fsschedule** command.

### Parameters



Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
feature	Required	1	The name of the feature. Valid types are: <b>clninfo</b> , <b>clnver</b> , <b>defrag</b> , <b>rebuild</b> , <b>p_backup</b> , <b>f_backup</b> , <b>spolicy</b> , <b>healthck</b> , <b>activevault</b> , and <b>archive_cmp</b> .	-f
schedule	Required	1	The schedule name.	-n
period	Optional	1	The period of the schedule. Valid options are: <b>daily</b> , <b>weekly</b> , and <b>monthly</b> .	-p
weekday	Optional	1	The day of the week: Valid options are: <b>Sun</b> , <b>Mon</b> , <b>Tue</b> , <b>Wed</b> , <b>Thu</b> , <b>Fri</b> , or <b>Sat</b> .	-e
monthday	Optional	1	The day of the month: 1 – 31.	-y
runtime	Required	1	The start time of the feature defined as HHMM.	-t
window	Optional	1	The window or runtime offset defined as HHMM.	-w
option	Optional	1	The option used by the feature. Currently, only the spolicy feature requires an option, which is an existing policy class.	-o
archive	Optional	N	Applicable only for activevault feature. Media to be vaulted are selected from this list of archives.	--
vault	Optional	1	Applicable only for activevault feature. The destination archive vault media.	-vault
copy	Optional	1	Applicable only for activevault feature. A list of copy numbers to query on.	-copy

Parameter	Req / Opt	Num	Description	CLI Option
used	Optional	1	Applicable only for activevault feature. Select only media that have used size capacity. Size is in <b>bytes</b> by default, but a suffix of B(ytes), K(ibibytes), M(ebibytes), G(ibibytes) or T(ebibytes) may be used to specify capacity.	-used
free	Optional	1	Applicable only for activevault feature. Select only media that have size capacity remaining. Size is in <b>bytes</b> by default, but a suffix of B(ytes), K(ibibytes), M(ebibytes), G(ibibytes) or T(ebibytes) may be used to specify capacity.	-remaining
age	Optional	1	Applicable only for activevault feature. Vault media according to age. age by default is in seconds, but a time unit may also be provided to specify seconds, days, weeks, months or years. A specific date may also be specified with the <b>YYYY:MM:DD:hh:mm:ss</b> format.	-age
sort	Optional	1	Applicable only for activevault feature. Sort results based according to column where column can be age, id, full, remaining or used. Age will sort by last access time of the media. Id will sort by media ID. Full will sort by the percentage full of the media. This is the default behavior if no sort option is specified. Remaining will sort by space remaining on the media. Used will sort by the amount of space used on the media.	-sort
migrate	Optional	1	Applicable only for activevault feature. Valid values are true and false. If true, Select from media in the MIGRATE media class. If false, Ignore media in the MIGRATE media class.	-migrate or -nomigrate
pending	Optional	1	Applicable only for activevault feature. Valid values are <b>true</b> and <b>false</b> . If <b>true</b> , Select media where a vaulting operation is pending. If <b>false</b> , Select media where no vaulting operation is pending.	-pending or -nopending

Parameter	Req / Opt	Num	Description	CLI Option
highmark	Optional	1	Applicable only for activevault feature. Override the ACTIVEVAULT_HIGH_USE sysparm value to start vaulting if the used capacity of the Storage Manager license is at or above the pct percent.	-highmark
lowmark	Optional	1	Applicable only for activevault feature. Override the ACTIVEVAULT_LOW_USE sysparm value to stop vaulting if the used capacity of the Storage Manager license is below the pct percent.	-lowmark
fill	Optional	1	Applicable only for activevault feature. Override the ACTIVEVAULT_FULL_PERCENT sysparm value to consider vaulting media that is at or above the specified percent.	-fullpct
report	Optional	1	Applicable only for activevault feature. Generate a media report based upon the selection criteria.	-report
includepolicy	Optional	N	Applicable only for activevault feature. Select media belonging to the list of policy classes. If this option is used, only media belonging to the list will be selected.	-include_policy
excludepolicy	Optional	N	Applicable only for activevault feature. Excludes media belonging to the listed policy classes.	-exclude_policy
capacity	Optional	1	Applicable only for activevault feature. Report the current licensed capacity and total available licensed capacity.	-capacity
dryrun	Optional	1	Applicable only for activevault feature. Show what media would be vaulted according to the selection criteria.	-dryrun
limit	Optional	1	Applicable only for activevault feature. Limit the number of vaulted media to this number.	-limit

Parameter	Req / Opt	Num	Description	CLI Option
notify	Optional	1	Applicable only for activevault feature. Set the email notification level for active vault policy admin alerts where level is either <b>none</b> , <b>error</b> , <b>warn</b> , or <b>info</b> . <b>None</b> will suppress all email notifications. <b>Error</b> will only send notifications when an error occurs, such as database errors or licensing errors. <b>Warn</b> will send notifications for warnings, such as being unable to vault enough media to satisfy the low water mark. <b>Info</b> will cause email notifications to be sent indicating that the active vault policy completed successfully. The default notification level is <b>warn</b> .	-notify
noheader	Optional	1	Applicable only for the activevault feature. Do not display header or result count in the media report.	-noheader
verbose	Optional	1	Applicable only for activevault feature.	-verbose
policy	Optional	1	Applicable only for the activevault feature. The name of the Active Vault policy to use for email notifications.	-policy

### Example

```
https://<<SERVER>>/sws/v2/schedule/fsschedule/create
?feature=f_backup
&schedule=schedule1
&period=daily
&weekday=Sun
&runtime=1000
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsschedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsschedule.xsd">
<header>
<commandName>fsschedule</commandName>
<commandLine>/usr/adic/TSM/exec/fsschedule -a -n schedule2 -f f_backup -p daily -
e Sun -t 1000 -F xml</commandLine>
<commandDescription>Insert, modify, delete, reset, or report all maintenance
features in the Quantum storage subsystem.</commandDescription>
<localDateISO>2014-05-15T12:25:28</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:25:28</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:25:28Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:25:28</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388090</requestId>
<commandName>/usr/adic/TSM/exec/fsschedule</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:25:28</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
```

```
<localTimeEnd>12:25:28</localTimeEnd>
<localDayOfWeekEnd>4</localDayOfWeekEnd>
<gmtDateISOEnd>2014-05-15T17:25:28Z</gmtDateISOEnd>
<gmtDateEnd>2014-05-15</gmtDateEnd>
<gmtTimeEnd>17:25:28</gmtTimeEnd>
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>
<elapsedTimeInSeconds>0.0049</elapsedTimeInSeconds>
</footer>
</fsschedule>
```

## Schedule / Update

This web service will update a schedule.

This web service runs the `fsschedule` command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
schedule	Required	1	The schedule name.	-n
period	Optional	1	The period of the schedule. Valid options are: <b>daily</b> , <b>weekly</b> , or <b>monthly</b> .	-p
weekday	Optional	1	The day of the week: Valid options are: Sun, Mon, Tue, Wed, Thu, Fri, or Sat.	-e
monthday	Optional	1	The day of the month: 1 – 31.	-y

Parameter	Req / Opt	Num	Description	CLI Option
runtime	Optional	1	The start time of the feature defined as HHMM.	-t
window	Optional	1	The window or runtime offset defined as HHMM.	-w
option	Optional	1	The option used by the feature. Currently, only the spolicy feature requires an option, which is an existing policy class.	-o
archive	Optional	N	Applicable only for the activevault feature. Media to be vaulted are selected from this list of archives.	--
vault	Optional	1	Applicable only for the activevault feature. The destination archive where to vault media.	-vault
copy	Optional	1	Applicable only for the activevault feature. A list of copy numbers to query.	-copy
used	Optional	1	Applicable only for the activevault feature. Select only media that have used size capacity. Size is in <b>bytes</b> by default, but a suffix of B(ytes), K (ibibytes), M(ebibytes), G(ibibytes) or T (ebibytes) may be used to specify capacity.	-used
free	Optional	1	Applicable only for the activevault feature. Select only media that have size capacity remaining. Size is in <b>bytes</b> by default, but a suffix of B(ytes), K (ibibytes), M(ebibytes), G(ibibytes) or T (ebibytes) may be used to specify capacity.	-remaining
age	Optional	1	Applicable only for the activevault feature. Vault media according to age. Age by default is in <b>seconds</b> , but a time unit may also be provided to specify seconds, days, weeks, months or years. A specific date may also be specified with the YYYY:MM:DD:hh:mm:ss format.	-age

Parameter	Req / Opt	Num	Description	CLI Option
sort	Optional	1	Applicable only for the activevault feature. Sort results based according to column where column can be age, id, full, remaining or used. Age will sort by last access time of the media. Id will sort by media ID. Full will sort by the percentage full of the media. This is the default behavior if no sort option is specified. Remaining will sort by space remaining on the media. Used will sort by the amount of space used on the media.	-sort
migrate	Optional	1	Applicable only for the activevault feature. Valid values are <b>true</b> and <b>false</b> . If <b>true</b> , select from media in the MIGRATE media class. If <b>false</b> , ignore media in the MIGRATE media class.	-migrate or - nomigrate
pending	Optional	1	Applicable only for the activevault feature. Valid values are <b>true</b> and <b>false</b> . If <b>true</b> , select media where a vaulting operation is pending. If <b>false</b> , select media where no vaulting operation is pending.	-pending or -nopending
highmark	Optional	1	Applicable only for the activevault feature. Override the ACTIVEVAULT_HIGH_USE sysparm value to start vaulting if the used capacity of the Storage Manager license is at or above the pct percent.	-highmark
lowmark	Optional	1	Applicable only for the activevault feature. Override the ACTIVEVAULT_LOW_USE sysparm value to stop vaulting if the used capacity of the Storage Manager license is below the pct percent.	-lowmark
fill	Optional	1	Applicable only for the activevault feature. Override the ACTIVEVAULT_FULL_PERCENT sysparm value to consider vaulting media that is at or above the pct percent.	-fullpct



Parameter	Req / Opt	Num	Description	CLI Option
report	Optional	1	Applicable only for the activevault feature. Generate a media report based upon the selection criteria.	-report
includepolicy	Optional	N	Applicable only for the activevault feature. Select media belonging to the list of policy classes. If this option is used, only media belonging to the list will be selected.	-include_policy
excludepolicy	Optional	N	Applicable only for the activevault feature. Excludes media belonging to the listed policy classes.	-exclude_policy
capacity	Optional	1	Applicable only for the activevault feature. Report the current licensed capacity and total available licensed capacity.	-capacity
dryrun	Optional	1	Applicable only for the activevault feature. Show media that would be vaulted according to the selection criteria.	-dryrun
limit	Optional	1	Applicable only for the activevault feature. Limit the number of vaulted media to this number.	-limit
notify	Optional	1	Applicable only for the activevault feature. Set the email notification level for active vault policy admin alerts where level is either none, error, warn, or info. Nnone will suppress all email notifications. Error will only send notifications when an error occurs, such as database errors or licensing errors. Warn will send notifications for warnings, such as being unable to vault enough media to satisfy the low water mark. Info will cause email notifications to be sent indicating that the active vault policy completed successfully. The default notification level is warn.	-notify
noheader	Optional	1	Applicable only for the activevault feature. Do not display header or result count in the media report.	-noheader

Parameter	Req / Opt	Num	Description	CLI Option
verbose	Optional	1	Applicable only for activevault feature.	-verbose
policy	Optional	1	Applicable only for the activevault feature. The name of the Active Vault policy to use for email notifications.	-policy

### Example

```
https://<<SERVER>>/sws/v2/schedule/fsschedule/update  
?feature=defrag  
&schedule=schedule1  
&period=daily  
&weekday=Sun  
&runtime=1000  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsschedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsschedule.xsd">
<header>
<commandName>fsschedule</commandName>
<commandLine>/usr/adic/TSM/exec/fsschedule -m -n schedule1 -p daily -e Sun -t
1000 -F xml</commandLine>
<commandDescription>Insert, modify, delete, reset, or report all maintenance
features in the Quantum storage subsystem.</commandDescription>
<localDateISO>2014-05-15T12:27:59</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:27:59</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:27:59Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:27:59</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388095</requestId>
<commandName>/usr/adic/TSM/exec/fsschedule</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:27:59</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
```

```
<localTimeEnd>12:27:59</localTimeEnd>  
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:27:59Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:27:59</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0057</elapsedTimeInSeconds>  
</footer>  
</fsschedule>
```

## Schedule / Delete

This web service deletes an existing schedule.

This web service runs the `fsschedule` command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
schedule	Required	1	The schedule name.	-n

### Example

```
https://<<SERVER>>/sws/v2/schedule/fsschedule/delete  
?schedule=schedule1  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsschedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsschedule.xsd">
<header>
<commandName>fsschedule</commandName>
<commandLine>/usr/adic/TSM/exec/fsschedule -d -n schedule1 -F xml</commandLine>
<commandDescription>Insert, modify, delete, reset, or report all maintenance
features in the Quantum storage subsystem.</commandDescription>
<localDateISO>2014-05-15T12:29:23</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:29:23</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:29:23Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:29:23</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388098</requestId>
<commandName>/usr/adic/TSM/exec/fsschedule</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:29:24</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:29:24</localTimeEnd>
```

```
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:29:24Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:29:24</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0613</elapsedTimeInSeconds>  
</footer>  
</fsschedule>
```

## Schedule / Reset

This web service resets an existing schedule.

This web service runs the **fsschedule** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
feature	Required	1	The name of the feature. Valid types are: <b>clninfo</b> , <b>clnver</b> , <b>defrag</b> , <b>rebuild</b> , <b>p_backup</b> , <b>f_backup</b> , <b>spolicy</b> , <b>healthck</b> , <b>activevault</b> , or <b>archive_cmp</b> .	-f

### Example

```
https://<<SERVER>>/sws/v2/schedule/fsschedule/reset  
?feature=defrag  
&format=xml
```

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<fsschedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fsschedule.xsd">
<header>
<commandName>fsschedule</commandName>
<commandLine>/usr/adic/TSM/exec/fsschedule -r -f defrag -F xml</commandLine>
<commandDescription>Insert, modify, delete, reset, or report all maintenance
features in the Quantum storage subsystem.</commandDescription>
<localDateISO>2014-05-15T12:30:26</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>12:30:26</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T17:30:26Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>17:30:26</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388101</requestId>
<commandName>/usr/adic/TSM/exec/fsschedule</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
</statuses>
<footer>
<returnCode>0</returnCode>
<localDateISOEnd>2014-05-15T12:30:26</localDateISOEnd>
<localDateEnd>2014-05-15</localDateEnd>
<localTimeEnd>12:30:26</localTimeEnd>
```

```
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T17:30:26Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>17:30:26</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0139</elapsedTimeInSeconds>  
</footer>  
</fsschedule>
```

## System

### System / Backup

Execute backup of configuration, database, and file system metadata.

This web service runs the **snbackup** command.

#### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

#### Example

```
https://<<SERVER>>/sws/v2/system/snbackup  
?partial=true  
&meta=true
```

#### Output



```
Exit Code: 0
Output: =====
=====--- StorNext Backup ---=====
=====

Backup Start: Thu Apr 17 12:56:30 2014
Log File: /usr/adic/TSM/logs/reports/backup.status
Type: Partial Backup

=====
=====--- Begin Processing ---=====
=====

- Connecting to database
Database connection successful

- Configuring backup information
- Backup using storage manager
- Querying database for _adic_backup class information
- Getting list of file systems with metadata files
- Checking for BACKUPFS fs_sysparm setting
- Scanning for defined file systems
- Updating fs_sysparm for selected mount point

- Checking for required components
- Verifying Relation Point
- Get identifier information and sequence numbers
Using Backup ID 15 for partial backup
- Check Failures

- Getting list of existing backup files
```

```
- Package Configuration Files

- Package Database Files
- Dumping database
Backup scanned up to lsn 71894126433
mysqlbackup incremental backup succeeded
- Create archive file for database backup

- Package Filesystem Metadata Files
- Suspending metadata archive processing for target
- Updating reference copy for target
- Compressing package for target
Packaged target metadata - took 83s
- Resuming metadata archive processing for target

- Storing files to media

- Removing obsolete backup versions
- Update database backup catalog
- Syncing backup manifest files
Wrote out snbackup_manifest file /usr/adic/mysql/snbackup_manifest
Wrote out snbackup_manifest file /usr/adic/TSM/internal/status_dir/snbackup_
manifest
Wrote out device_manifest file /usr/adic/mysql/device_manifest
Wrote out device_manifest file /usr/adic/TSM/internal/status_dir/device_manifest

- Checking for files to revert

=====
----- Backup End -----
=====
```

```
== End Time: Thu Apr 17 13:08:51 2014  
== Backup successfully completed
```

## System / Backup Status

Retrieves the status of the backup operation.

This web service runs the **snbackup** command.

---

**i Note:** This web service returns text or json output. The default is text.

### Parameters

This web service does not contain any parameters.

### Example

```
https://<<SERVER>>/sws/v2/system/snbackup/status
```

### Output

```
Exit Code: 0
Output: Backup Complete: Backup completed successfully
Output from last run of snbackup:
2014-04-17-12:56:30: Backup Start: Thu Apr 17 12:56:30 2014
2014-04-17-12:56:30: - Connecting to database
2014-04-17-12:56:30: Database connection successful
2014-04-17-12:56:30: - Configuring backup information
2014-04-17-12:56:30: - Backup using storage manager
2014-04-17-12:56:30: - Querying database for _adic_backup class information
2014-04-17-13:03:56: - Compressing package for target
2014-04-17-13:05:19: Packaged target metadata - took 83s
2014-04-17-13:05:19: - Resuming metadata archive processing for target
2014-04-17-13:05:19: - Storing files to media
2014-04-17-13:08:36: - Removing obsolete backup versions
2014-04-17-13:08:36: - Update database backup catalog
2014-04-17-12:56:30: - Getting list of file systems with metadata files
2014-04-17-12:56:30: - Checking for BACKUPFS fs_sysparm setting
2014-04-17-12:56:30: - Scanning for defined file systems
2014-04-17-12:56:30: - Updating fs_sysparm for selected mount point
2014-04-17-12:56:30: - Checking for required components
2014-04-17-12:56:30: - Verifying Relation Point
2014-04-17-12:56:31: - Get identifier information and sequence numbers
2014-04-17-12:56:31: - Check Failures
2014-04-17-12:56:31: - Getting list of existing backup files
2014-04-17-12:56:31: - Package Configuration Files
2014-04-17-12:56:58: - Package Database Files
2014-04-17-12:56:58: - Dumping database
2014-04-17-13:02:10: Backup scanned up to lsn 71894126433
2014-04-17-13:02:10: mysqlbackup incremental backup succeeded
2014-04-17-13:02:10: - Create archive file for database backup
2014-04-17-13:02:14: - Package Filesystem Metadata Files
2014-04-17-13:02:14: - Suspending metadata archive processing for target
```

```
2014-04-17-13:02:14: - Updating reference copy for target
2014-04-17-13:08:37: - Syncing backup manifest files
2014-04-17-13:08:37: Wrote out snbackup_manifest file /usr/adic/mysql/snbackup_
manifest
2014-04-17-13:08:37: Wrote out snbackup_manifest file
/usr/adic/TSM/internal/status_dir/snbackup_manifest
2014-04-17-13:08:39: Wrote out device_manifest file /usr/adic/mysql/device_
manifest
2014-04-17-13:08:39: Wrote out device_manifest file
/usr/adic/TSM/internal/status_dir/device_manifest
2014-04-17-13:08:51: - Checking for files to revert
2014-04-17-13:08:51: == End Time: Thu Apr 17 13:08:51 2014
2014-04-17-13:08:51: == Backup successfully completed
systems with relation points
```

## System / File System Report

This web service reports the status of a file system and the status of stripe groups that belong to it.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. The valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default value is <b>text</b> .	-F
filesystem	Required	1	The file system name for which status is required.	None
showstripegroups	Optional	1	This option will also enable the status information of stripe groups. The valid values are <b>true</b> and <b>false</b> . The default value is <b>false</b> .	

### Example

```
https://<<SERVER>>/sws/v2/system/filesystem/info  
?filesystem=snfx1  
&showstripegroups=true  
&format=json
```

## Output

```
{  
  "status" : "Online",  
  "totalSpace" : "19.991 GB",  
  "usedSpace" : "9.1 GB",  
  "usedPercent" : "45%",  
  "freeSpace" : "10.937 GB",  
  "freePercent" : "55%",  
  "sgList" : [ {  
    "name" : "sg0",  
    "totalSpace" : "19.99 GB",  
    "reservedSpace" : "4.13 GB",  
    "freeSpace" : "15.07 GB"  
  } ]  
}
```

## System / Information

Retrieves the latest status of system components.

This web service runs control scripts for different components and gathers the output.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F

### Example

```
https://<<SERVER>>/sws/v2/system/info  
?format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<xml-fragment>
<sws:SoftwareVersion
xmlns:sws="http://www.quantum.com/sws">5.0.0</sws:SoftwareVersion>
<sws:System xmlns:sws="http://www.quantum.com/sws">ONLINE</sws:System>
<sws:Tsm xmlns:sws="http://www.quantum.com/sws">ONLINE</sws:Tsm>
<sws:Msm xmlns:sws="http://www.quantum.com/sws">ONLINE</sws:Msm>
<sws:Dsm xmlns:sws="http://www.quantum.com/sws">ONLINE</sws:Dsm>
<sws:Database xmlns:sws="http://www.quantum.com/sws">ONLINE</sws:Database>
<sws:Srvclg xmlns:sws="http://www.quantum.com/sws">ONLINE</sws:Srvclg>
</xml-fragment>
```

## System / Parameters

Report the value for the specified Tertiary Manager system parameter.

This web service runs the **showsysparm** command.

### Parameters

Parameter	Req / Opt	Num	Description	CLI Option
mode	Optional	1	This option specifies if the job will be run synchronously or asynchronously. The valid values are <b>sync</b> and <b>async</b> . The default value is <b>sync</b> .	None
jobtype	Optional	1	This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <b>fast</b> and <b>slow</b> . The default value is <b>fast</b> .	None
format	Optional	1	The output format requested. Valid values are <b>xml</b> , <b>json</b> and <b>text</b> . The default is <b>text</b> .	-F
param	Required	N	Name of the system parameter.	no CLI option

### Example

```
https://<<SERVER>>/sws/v2/system/showsysparm
?param=CLASS_DEFAULT_COPIES
&format=xml
```

## Output

```
<?xml version="1.0" encoding="UTF-8" ?>
<showsysparm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="showsysparm.xsd">
<header>
<commandName>showsysparm</commandName>
<commandLine>/usr/adic/TSM/util/showsysparm CLASS_DEFAULT_COPIES -F
xml</commandLine>
<commandDescription>Report the value for the specified Tertiary Manager system
parameter.</commandDescription>
<localDateISO>2014-05-15T10:39:59</localDateISO>
<localDate>2014-05-15</localDate>
<localTime>10:39:59</localTime>
<localDayOfWeek>4</localDayOfWeek>
<gmtDateISO>2014-05-15T15:39:59Z</gmtDateISO>
<gmtDate>2014-05-15</gmtDate>
<gmtTime>15:39:59</gmtTime>
<gmtDayOfWeek>4</gmtDayOfWeek>
</header>
<sysparms>
<sysparm>
<name>CLASS_DEFAULT_COPIES</name>
<value>1</value>
</sysparm>
</sysparms>
<statuses>
<status>
<statusCode>FS0000</statusCode>
<statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387968</requestId>
<commandName>/usr/adic/TSM/util/showsysparm</commandName>
<commandStatus>completed</commandStatus>
<statusText>Command Successful.</statusText>
</status>
```



```
</statuses>  
<footer>  
<returnCode>0</returnCode>  
<localDateISOEnd>2014-05-15T10:39:59</localDateISOEnd>  
<localDateEnd>2014-05-15</localDateEnd>  
<localTimeEnd>10:39:59</localTimeEnd>  
<localDayOfWeekEnd>4</localDayOfWeekEnd>  
<gmtDateISOEnd>2014-05-15T15:39:59Z</gmtDateISOEnd>  
<gmtDateEnd>2014-05-15</gmtDateEnd>  
<gmtTimeEnd>15:39:59</gmtTimeEnd>  
<gmtDayOfWeekEnd>4</gmtDayOfWeekEnd>  
<elapsedTimeInSeconds>0.0000</elapsedTimeInSeconds>  
</footer>  
</showsysparm>
```

# Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

StorNext Web Services (SWS) V2 is a collection of web services that provide an interface to important functionalities within the product. SWS V2 enables the user to make quick and reliable web service calls and perform different operations and obtain reports from the StorNext enabled system.

This section provides an overview of how to use the SWS V2 web services. The section demonstrates different ways the web services can be invoked and also provides insight into the various options.

## Prerequisites

**Table 3:** StorNext Web Services V2 Prerequisites

Prerequisite	Notes
StorNext 5 release 5.3 (or later) should be installed on the system.	For information on how to install StorNext, contact a Quantum representative or visit the <i>StorNext 5 Installation Guide</i> online at <a href="http://www.quantum.com/sn5docs">http://www.quantum.com/sn5docs</a> .
Web services should be enabled.	For information on how to enable web services, see the section <a href="#">Getting Started on the next page</a> .
Web service authentication and user credentials should be configured.	For information on how to configure authentication and user credentials, see the section <a href="#">Getting Started on the next page</a> .
If authentication is required, web service user(s) needs to be created.	For information on how to create web service users, see the section <a href="#">Getting Started on the next page</a> .
If SWS V2 web services need to be executed via a web browser, it should be installed on the system.	SWS V2 can be invoked from various browsers. For example: <ul style="list-style-type: none"> <li>• Microsoft Windows Internet Explorer</li> <li>• Mozilla Firefox</li> <li>• Google Chrome</li> </ul>
If SWS V2 web services are invoked from customer source code, relevant libraries need to be installed and referenced from the source code.	For additional information, see the section <a href="#">Run the Web Services from a Client Application on page 191</a> .

## Getting Started

This section provides the procedures required to begin using SWS V2 web services. The procedures assume you have access to the StorNext Graphical User Interface (GUI).

### Enable Web Services

Enable or disable the SWS V2 web services using the StorNext GUI. Ensure the GUI is accessible via a supported browser (see [Prerequisites on the previous page](#)). If StorNext is installed and configured correctly, access the StorNext GUI from `https://<mdc_ip_address>[:<mdc_port>]`.

1. On the StorNext GUI, click **Service**, and then click **Web Services (V2)**. The **Service > Web Services (V2)** page appears.
2. In the **State** box, select **On** to enable web services.
3. Click **Apply**.

**Figure 1:** Enable Web Services

**Service > Web Services (V2)**

State: On

Protocol: On

Authentication Type: User

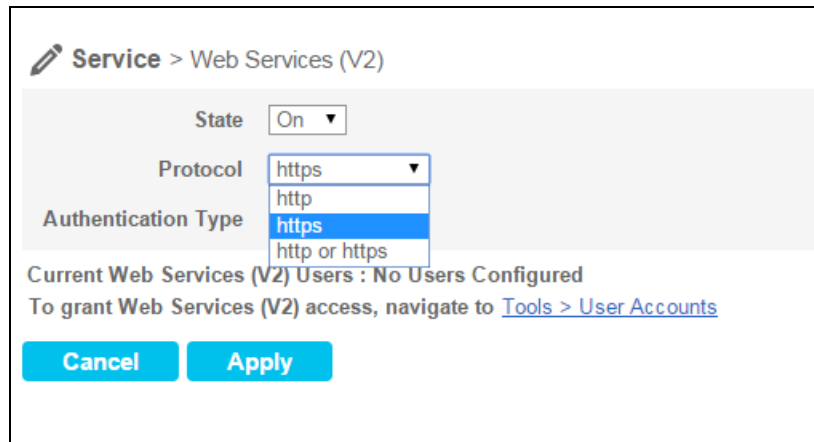
Current Web Services (V2) Users : No Users Configured  
To grant Web Services (V2) access, navigate to [Tools > User Accounts](#)

Cancel Apply

### Configure Protocol for Access

1. On the StorNext GUI, click **Service**, and then click **Web Services (V2)**. The **Service > Web Services (V2)** page appears.
2. In the **Protocol** box, select the protocol (**http**, **https** or **http or https**).
3. Click **Apply**.

**Figure 2:** Configure Web Services Protocol



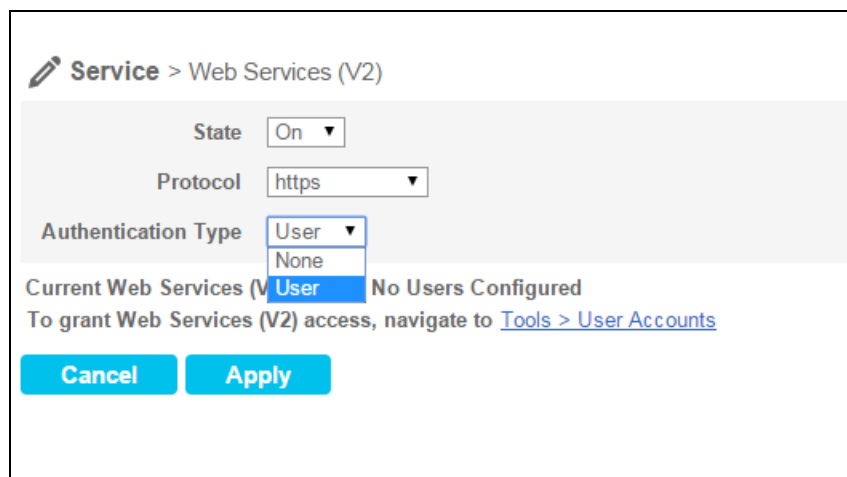
## Configure Authentication Type

Configure Web Services to authenticate web service requests with a **username** and **password** by configuring the authentication type to **User**.

**i Note:** If **User** is selected, all web service requests require a valid **username** and **password**. For this purpose, a valid web service user needs to be created. To grant Web Services (V2) access, on the StorNext GUI navigate to **Tools > User Accounts**.

1. On the StorNext GUI, click **Service**, and then click **Web Services (V2)**. The **Service > Web Services (V2)** page appears.
2. In the **Authentication Type** box, select the type (**User**, or **None**).
3. Click **Apply**.

**Figure 3:** Configure Authentication for Web Services



## Create the SWS V2 Web Service User

1. On the **Tools** menu, click **User Accounts**. The **Tools >User Accounts** page appears. All existing users and the admin are shown.
2. Click **New**. The **Tools >User Accounts > New** page appears.
3. In the **User Name** field, type the name the new user will enter at the **User ID** field when he or she logs on to StorNext.
4. In the **Password** field, type the password the new user will enter when logging on to StorNext.
5. In the **Session Timeout** field, type a number, and then select Minutes or Hours as unit of time measurement from the drop-down. The **Session Timeout** specifies the predetermined amount of time that should elapse before the user is logged out of the system. The default is 30 minutes, and the valid range is from 10 minutes to 12 hours.

---

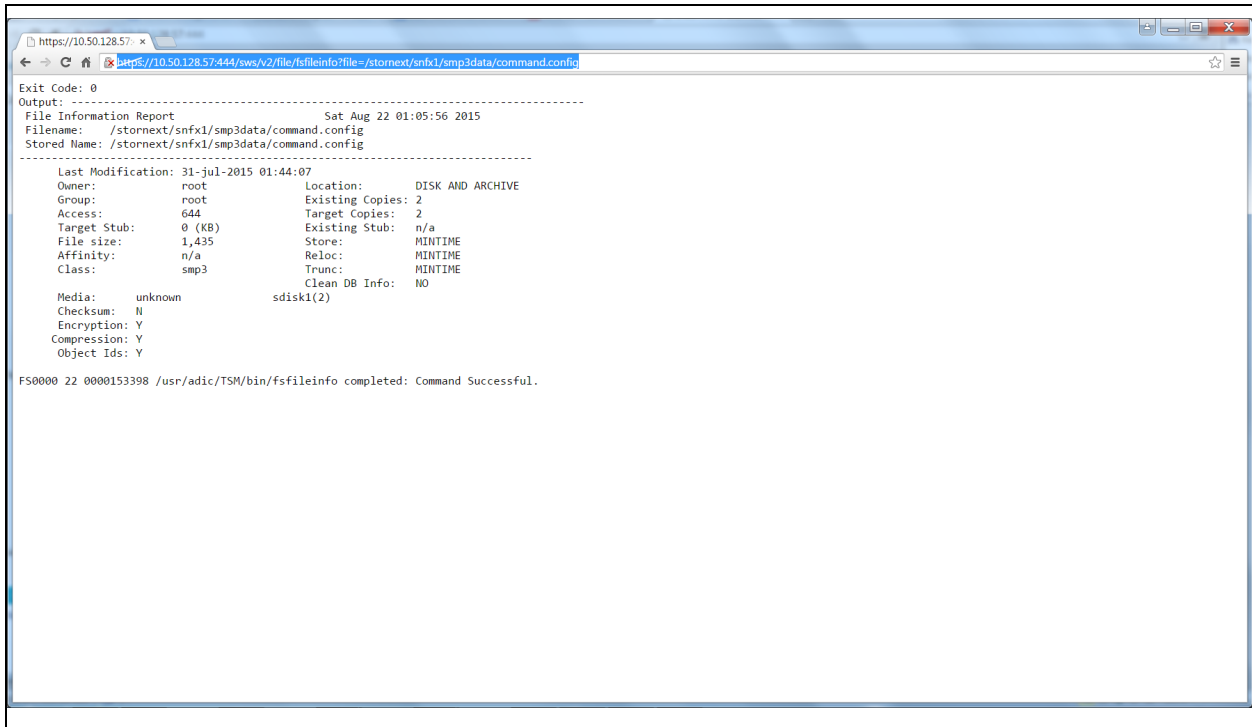
**i Note:** Access to the **Session Timeout** feature is available when a user has the **Manage Users** privilege checked within the **Admin Functions** section.

6. Roles are grouped according to **Admin Functions**, **Operator Functions** and **General User Functions**. You can automatically pre-select all the functions for one of these three roles by clicking at the **Access Control** field **Admin Defaults**, **Operator Defaults**, **General User Defaults** or **Web Services**. Selecting one of these roles for the new user makes it easy for you to automatically add or remove functions.
  - a. Under **Access Control**, click **Web Services**.
  - b. Under **General User Functions**, click **Use Web Services**.
7. When you are satisfied with the permissions you have assigned, click **Apply** to save your changes. To exit without saving, click **Cancel**.
8. When a message informs you that the new user was successfully added, click **OK**.

## Run the Web Services

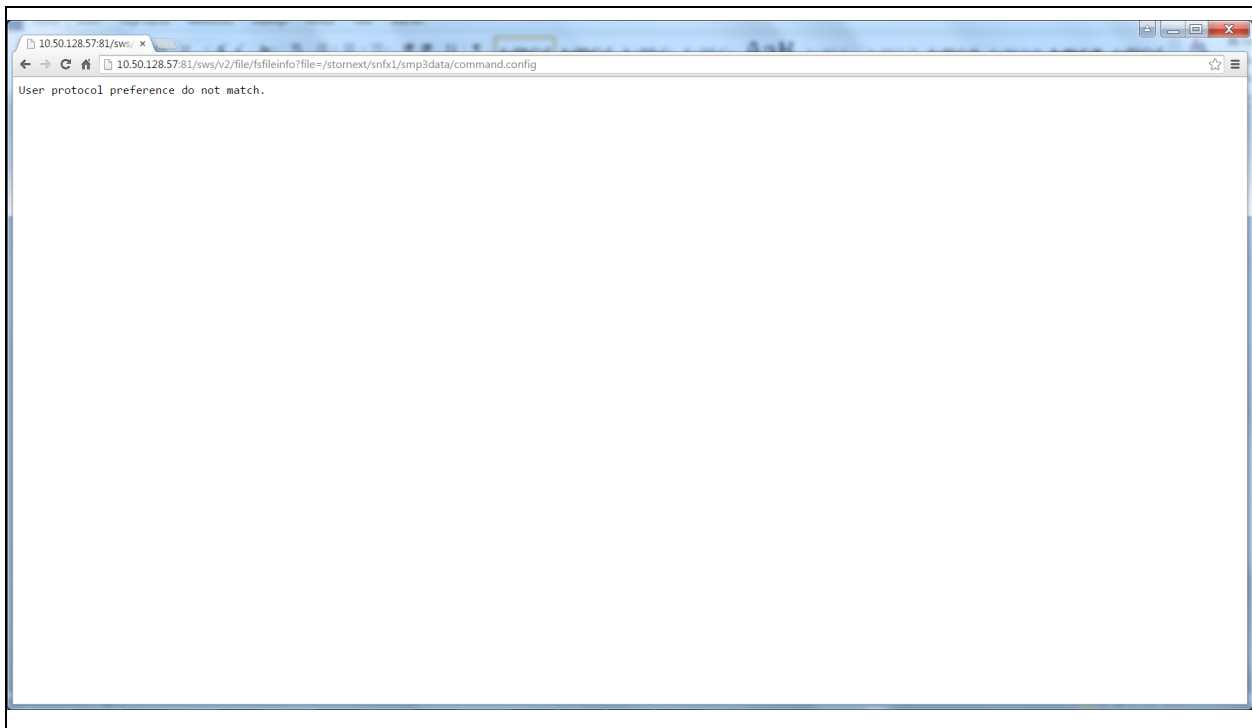
Run SWS V2 web services from a supported web browser (see [Prerequisites on page 184](#)). Below are examples of a sample run of the web services using Google Chrome. In the example, the web service is run with the **Protocol** configured to **https** and **Authentication Type** configured to **None**.

Figure 4: Example of SWS V2 run using Google Chrome



SWS V2 displays an error if a mismatch in the protocol exists, as illustrated below.

Figure 5: Example of SWS V2 Protocol Failure using Google Chrome



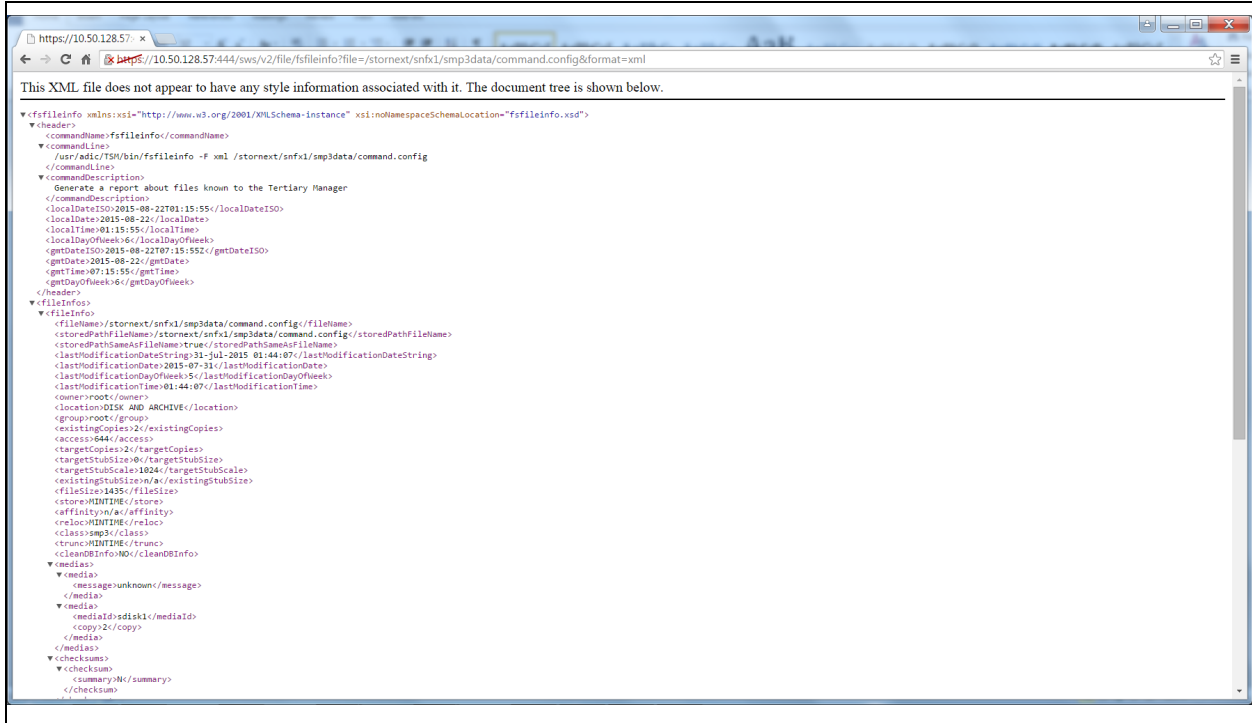
Similarly, if you provide a **username** and/or **password** in the query parameters, it is ignored.

SWS V2 supports different formats for response. The supported formats are **TEXT**, **XML** and **JSON**. However, there are certain web services that only support **TEXT** and **JSON**. For information on the supported formats for a particular web service, see the *StorNext Web Services Guide* online at <http://www.quantum.com/sn5docs>.

Quantum recommends **JSON** as the preferred format since most web services support this format. **XML** is also supported, but it is primarily used for compatibility with existing applications.

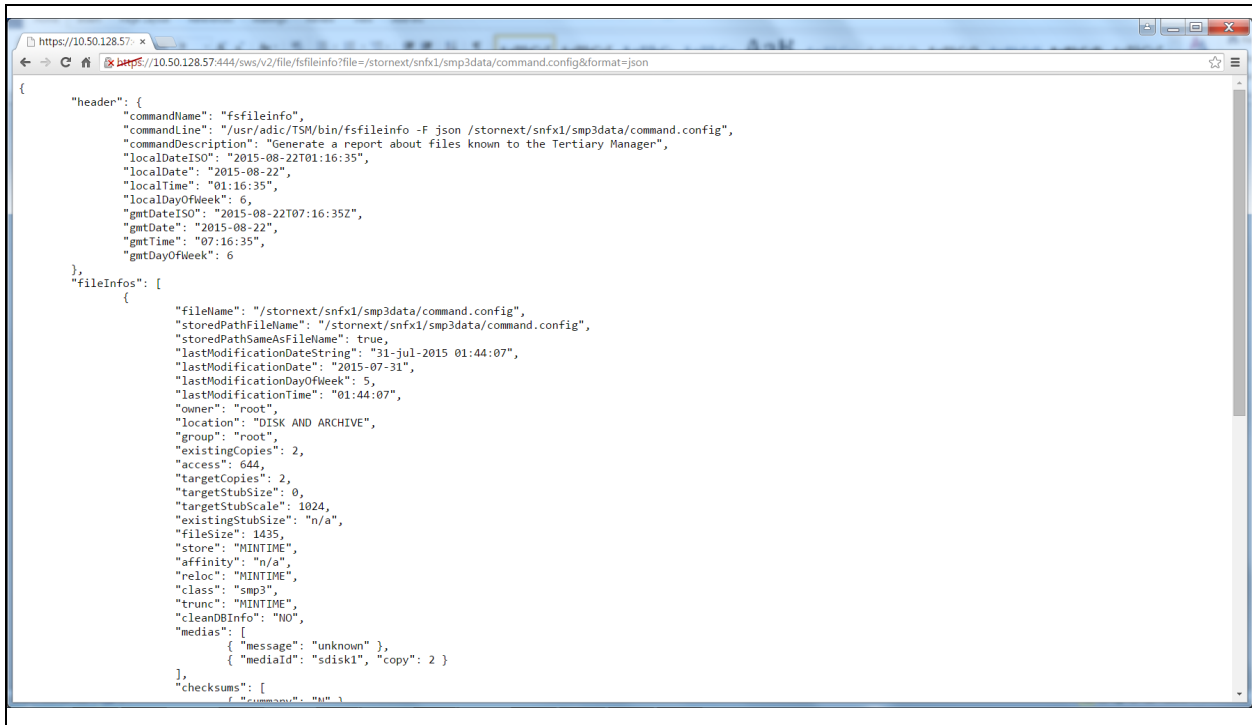
Below is an example of the same web service run with the **XML** format request.

**Figure 6:** Example of Web Service run with the XML Format using Google Chrome



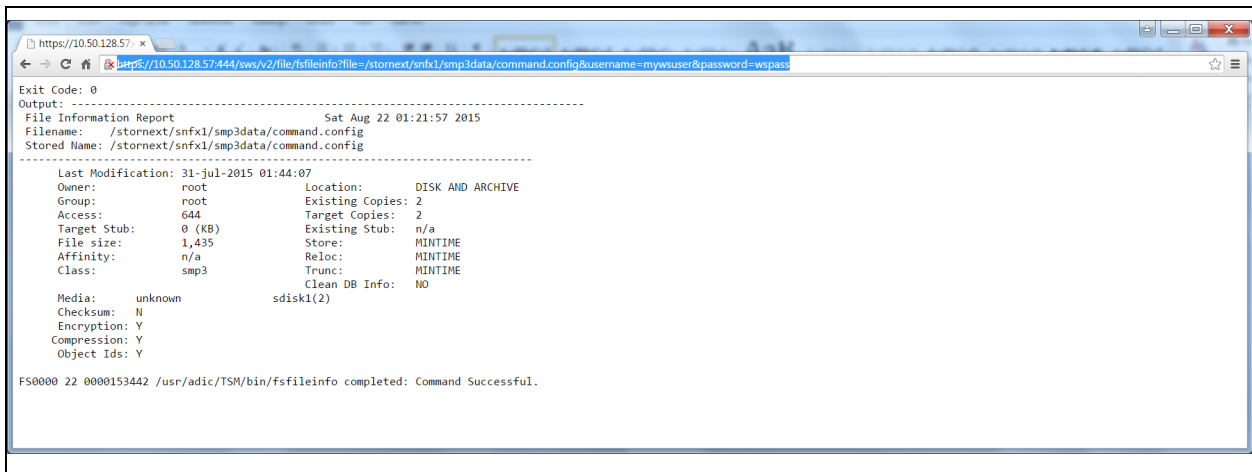
Below is an example of the same web service run with the **JSON** format request.

Figure 7: Example of Web Service run with the JSON Format using Google Chrome



If the **Authentication Type** is configured to **User**, the web service request requires a valid **username** and **password**.

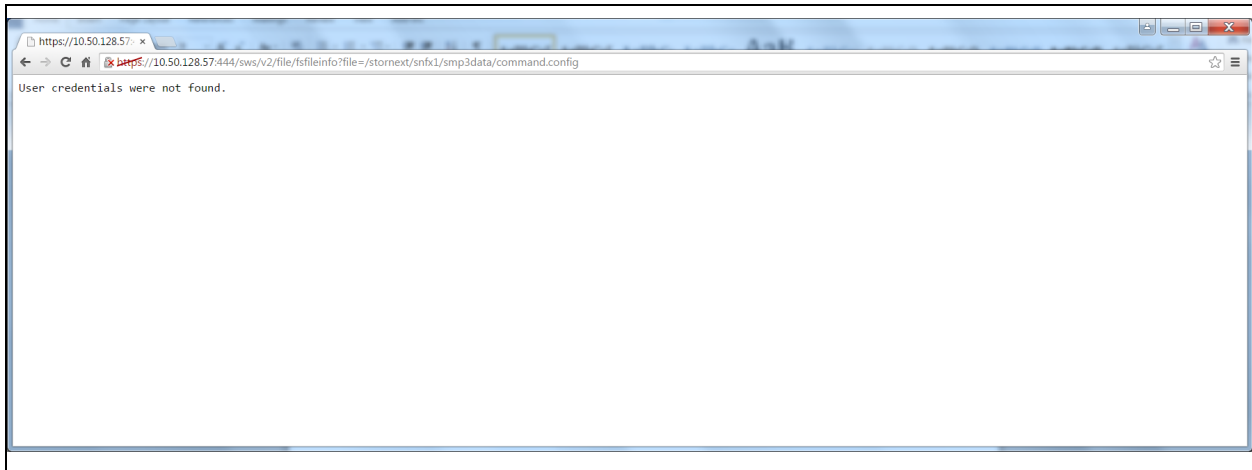
Figure 8: Example of a SWS V2 run using Google Chrome



If the user credentials are not provided, an error appears.

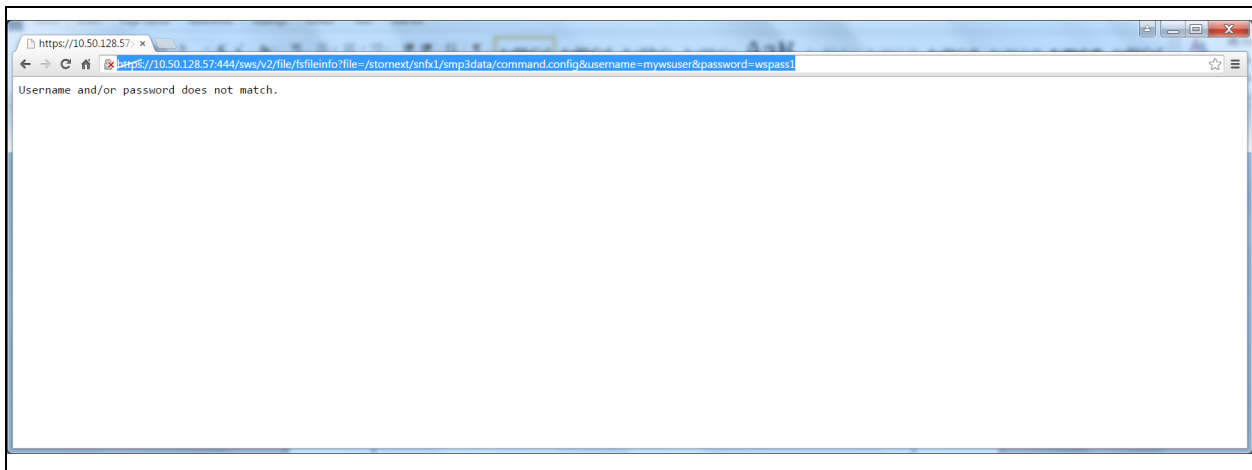


**Figure 9:** Example of a SWS V2 User Credential Failure using Google Chrome



If the incorrect **username** and **password** are provided in the credentials, a corresponding error appears.

**Figure 10:** Example of SWS V2 User Credentials Incorrect Error using Google Chrome



## Run the Web Services from a Client Application

This section provides the following:

- Demonstrate how to use the SWS V2 from client applications.
- Provide a step-by-step approach to integrate the SWS V2 web services in the client applications.
- Demonstrate the following scenarios.

Scenario	Web Service Used
Generate a report about files known to the Tertiary Storage Manager	fsfileinfo

Scenario	Web Service Used
Generate a report about the policy class associated with directory.	fsdirclass
Generate a report about storage manager policies.	fsclassinfo
Expedite the storage of a file that currently resides on disk to media.	fsstore
Remove the copy of a file from disk after the file was stored to a medium.	fsrmdiskcopy
Retrieve or recover files from media and place on disk.	fsretrieve
Create a directory quota on a managed file system.	snquota
Set the quota limits for the directory.	
List all available quotas on a file system.	

- Provide samples on how to asynchronously execute web services calls in SWS V2.
- Demonstrate how to use web services for object storage activities.
- SWS V2 web services supports several software languages; this section also demonstrates how to use the web services in customer applications written in Java, Perl and Python.

## Java

The example below uses the Jersey library. Download the latest version of Jersey library online at: <https://jersey.java.net/download.html>.

In the following example, a file info call is performed, similar to the example illustrated in the section [Run the Web Services on page 187](#).

```

// Get the client configuration
ClientConfig config = getClientConfig();

// Setup SSL for https connection
setupSSL(useHttps, config);

// Configure the client
Client client = configureClient(userName, password, config);

// Create the service
WebResource service = client.resource(getBaseURI(hostName, port, useHttps));

// Obtain the right media type based on the format requested by the user
String mediaType = getMediaType(format);

// Invoke the web service and obtain the response back as a String
String response = service.path("sws/v2/file/fsfileinfo")
    .queryParams("file", filePath)
    .queryParams("format", format)
    .accept(mediaType).get(String.class);

```

First, get the client configuration.

```

private ClientConfig getClientConfig() {
    ClientConfig config = new DefaultClientConfig();
    return config;
}

```

Now, configure for an **https** request. The code is optional and only required if **https** is the protocol.

```
private void setupSSL(boolean useHttps, ClientConfig config) {
    if (useHttps) {
        TrustManager[] certs = new TrustManager[] { new X509TrustManager() {
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }

            public void checkServerTrusted(X509Certificate[] chain,
                String authType) throws CertificateException {
            }

            public void checkClientTrusted(X509Certificate[] chain,
                String authType) throws CertificateException {
            }
        } };

        SSLContext ctx = null;
        try {
            ctx = SSLContext.getInstance("TLS");
            ctx.init(null, certs, new SecureRandom());
        } catch (java.security.GeneralSecurityException ex) {
        }
        HttpURLConnection.setDefaultSSLSocketFactory(ctx.getSocketFactory
    ());

        try {
            config.getProperties().put(
                HTTPSProperties.PROPERTY_HTTPS_PROPERTIES,
                new HTTPSProperties(new HostnameVerifier() {
                    public boolean verify(String hostname,
                        SSLSession session) {
                        return true;
                    }
                }, ctx));
        } catch (Exception e) {
        }
    }
}
```

Next, configure the client.

```
private static final int CONNECT_TIMEOUT = 30000; // 30 secs
private static final int READ_TIMEOUT = 30000; // 30 secs

private Client configureClient(String userName, String password,
    ClientConfig config) {
    Client client = Client.create(config);
    client.setConnectTimeout(CONNECT_TIMEOUT);
    client.setReadTimeout(READ_TIMEOUT);
    if (userName != null && userName.length() > 0 && password != null) {
        client.addFilter(new HTTPDigestAuthFilter(userName, password));
    }
    return client;
}
```

In the next step, create the service.

```
// Create the service
WebResource service = client.resource(getBaseURI(hostName, port, useHttps));

protected URI getBaseURI(String host, String port, boolean https)
    {
        String protocol = https ? "https://" : "http://";
        if (port != null) {
            return UriBuilder.fromUri(protocol + host + ":" + port +
"/").build();
        } else {
            return UriBuilder.fromUri(protocol + host + "/").build();
        }
    }
}
```

Next, obtain the proper media type in order to send the response correctly.

```
private String getMediaType(String format) {
    String mediaType = MediaType.TEXT_PLAIN;
    if (format != null) {
        if (format.equalsIgnoreCase("json")) {
            mediaType = MediaType.APPLICATION_JSON;
        } else if (format.equalsIgnoreCase("xml")) {
            mediaType = MediaType.APPLICATION_XML;
        }
    }
    return mediaType;
}
```

Finally, invoke the web service and obtain the response as a **String**.

```
String response = service.path("sws/v2/file/fsfileinfo")
                        .queryParams("file", filePath)
                        .queryParams("format", format)
                        .accept(mediaType).get(String.class);
```

Below is an example of the response appears.

```

{
  "header": {
    "commandName": "fsfileinfo",
    "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json
/stornext/snfx1/smp3data/command.config",
    "commandDescription": "Generate a report about files known to the
Tertiary Manager",
    "localDateISO": "2015-08-21T23:48:16",
    "localDate": "2015-08-21",
    "localTime": "23:48:16",
    "localDayOfWeek": 5,
    "gmtDateISO": "2015-08-22T05:48:16Z",
    "gmtDate": "2015-08-22",
    "gmtTime": "05:48:16",
    "gmtDayOfWeek": 5
  },
  "fileInfos": [
    {
      "fileName": "/stornext/snfx1/smp3data/command.config",
      "storedPathFileName": "/stornext/snfx1/smp3data/command.config",
      "storedPathSameAsFileName": true,
      "lastModificationDateString": "31-jul-2015 01:44:07",
      "lastModificationDate": "2015-07-31",
      "lastModificationDayOfWeek": 5,
      "lastModificationTime": "01:44:07",
      "owner": "root",
      "location": "DISK AND ARCHIVE",
      "group": "root",
      "existingCopies": 2,
      "access": 644,
      "targetCopies": 2,
      "targetStubSize": 0,
      "targetStubScale": 1024,
      "existingStubSize": "n/a",
      "fileSize": 1435,
      "store": "MINTIME",
      "affinity": "n/a",
      "reloc": "MINTIME",
      "class": "smp3",
      "trunc": "MINTIME",
      "cleanDBInfo": "NO",
      "medias": [
        { "message": "unknown" },
        { "mediaId": "sdisk1", "copy": 2 }
      ],
      "checksums": [

```

```

        { "summary": "N" }
    ],
    "encryptions": [
        { "summary": "Y" }
    ],
    "compressions": [
        { "summary": "Y" }
    ],
    "objects": [
        { "summary": "Y" }
    ]
    }
],
"statuses": [
    {
        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 21,
        "requestId": 153265,
        "commandName": "/usr/adic/TSM/bin/fsfileinfo",
        "commandStatus": "completed",
        "statusText": "Command Successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-08-21T23:48:16",
    "localDateEnd": "2015-08-21",
    "localTimeEnd": "23:48:16",
    "localDayOfWeekEnd": 5,
    "gmtDateISOEnd": "2015-08-22T05:48:16Z",
    "gmtDateEnd": "2015-08-22",
    "gmtTimeEnd": "05:48:16",
    "gmtDayOfWeekEnd": 5,
    "elapsedTimeInSeconds": "0.0013"
}
}

```

If the **Authentication Type** is configured to **User**, the **username** and **password** also requires to be passed as a query string.

```

String response = service.path("sws/v2/file/fsfileinfo")
    .queryParams("file", filePath)
    .queryParams("format", format)
    .queryParams("username", userName)
    .queryParams("password", password)
    .accept(mediaType).get(String.class);

```

Execute the same request using **POST**; the response is the same.



```
String inputX = "file=" + filePath;  
String response = service.path("sws/v2/file/fsfileinfo")  
    .accept(mediaType).post(String.class, inputX);
```

Below is the full source code with other functions.

```

import java.net.URI;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.UriBuilder;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.WebResource;
import com.sun.jersey.api.client.config.ClientConfig;
import com.sun.jersey.api.client.config.DefaultClientConfig;
import com.sun.jersey.api.client.filter.HTTPDigestAuthFilter;
import com.sun.jersey.client.urlconnection.HTTPSPProperties;

public class SWSV2Samples {
    private static final int CONNECT_TIMEOUT = 30000;
    private static final int READ_TIMEOUT = 30000;

    public String getFsFileInfo(boolean useHttps, String userName,
        String password, String hostName, String port, String format,
        String filePath) {
        ClientConfig config = getClientConfig();

        setupSSL(useHttps, config);
        Client client = configureClient(userName, password, config);
        WebResource service = client.resource(getBaseURI(hostName, port,
            useHttps));
        String mediaType = getMediaType(format);
        String response = service.path("sws/v2/file/fsfileinfo")
            .queryParams("file", filePath).queryParams("format", format)
            .queryParams("username", userName).queryParams("password", password)
            .accept(mediaType).get(String.class);

        return response;
    }
}

```

```

public String postFsFileInfo (boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String filePath) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String inputX = "file=" + filePath;
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/file/fsfileinfo")
        .accept(mediaType).post(String.class, inputX);

    return response;
}

public String getFsClassInfo(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String className) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/policy/fsclassinfo")
        .queryParams("policy", className).queryParams("format", format)
        .queryParams("username", userName).queryParams("password", password)
        .accept(mediaType).get(String.class);

    return response;
}

public String getFsDirClass(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String filePath) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,

```

```

        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/policy/fsdirclass")
        .queryParams("directory", filePath).queryParams("format", format)
        .queryParams("username", userName).queryParams("password", password)
        .accept(mediaType).get(String.class);

    return response;
}

public String getFsStore(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String filePath, int copies) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/file/fsstore")
        .queryParams("file", filePath)
        .queryParams("copies", Integer.toString(copies))
        .queryParams("format", format).accept(mediaType)
        .queryParams("username", userName).queryParams("password", password)
        .get(String.class);

    return response;
}

public String getFsRmDiskCopy(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String filePath) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/file/fsrmdiskcopy")
        .queryParams("file", filePath).queryParams("format", format)
        .queryParams("username", userName).queryParams("password", password)

```

```

        .accept(mediaType).get(String.class);

    return response;
}

public String getFsRetrieve(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String filePath) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/file/fsretrieve")
        .queryParams("file", filePath).queryParams("format", format)
        .queryParams("username", userName).queryParams("password", password)
        .accept(mediaType).get(String.class);

    return response;
}

public String getCreateSnQuota(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String fspath, String dirPath) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/quota/snquota")
        .queryParams("path", fspath).queryParams("directory", dirPath)
        .queryParams("action", "create").queryParams("format", format)
        .queryParams("username", userName).queryParams("password", password)
        .accept(mediaType).get(String.class);

    return response;
}

public String getSetSnQuota(boolean useHttps, String userName,

```

```

        String password, String hostName, String port, String format,
        String fspath, String dirPath, String hardLimit, String softLimit,
        String gracePeriod) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/quota/snquota")
        .queryParams("path", fspath).queryParams("directory", dirPath)
        .queryParams("action", "set").queryParams("hardlimit", hardLimit)
        .queryParams("softlimit", softLimit)
        .queryParams("graceperiod", gracePeriod)
    .queryParams("username", userName).queryParams("password", password)
        .queryParams("format", format).accept(mediaType)
        .get(String.class);

    return response;
}

public String getListSnQuota(boolean useHttps, String userName,
    String password, String hostName, String port, String format,
    String fsname) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    String response = service.path("sws/v2/quota/snquota")
        .queryParams("fsname", fsname).queryParams("action", "listall")
    .queryParams("username", userName).queryParams("password", password)
        .queryParams("format", format).accept(mediaType)
        .get(String.class);

    return response;
}

private String getMediaType(String format) {
    String mediaType = MediaType.TEXT_PLAIN;

```

```

    if (format != null) {
        if (format.equalsIgnoreCase("json")) {
            mediaType = MediaType.APPLICATION_JSON;
        } else if (format.equalsIgnoreCase("xml")) {
            mediaType = MediaType.APPLICATION_XML;
        }
    }
    return mediaType;
}

private Client configureClient(String userName, String password,
    ClientConfig config) {
    Client client = Client.create(config);
    client.setConnectTimeout(CONNECT_TIMEOUT);
    client.setReadTimeout(READ_TIMEOUT);
    if (userName != null && userName.length() > 0 && password != null) {
        client.addFilter(new HTTPDigestAuthFilter(userName, password));
    }
    return client;
}

private void setupSSL(boolean useHttps, ClientConfig config) {
    if (useHttps) {
        TrustManager[] certs = new TrustManager[] { new X509TrustManager() {
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }

            public void checkServerTrusted(X509Certificate[] chain,
                String authType) throws CertificateException {
            }

            public void checkClientTrusted(X509Certificate[] chain,
                String authType) throws CertificateException {
            }
        }
    }
}

SSLContext ctx = null;
try {
    ctx = SSLContext.getInstance("TLS");
    ctx.init(null, certs, new SecureRandom());
} catch (java.security.GeneralSecurityException ex) {
}

```





The output is shown below. Notice the location of the file is reported as **DISK** since the file has not yet been stored.

```

{
  "header": {
    "commandName": "fsfileinfo",
    "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json
/stornext/snfx1/smpltomediamedia/foobar0",
    "commandDescription": "Generate a report about files known to the
Tertiary Manager",
    "localDateISO": "2015-10-26T19:42:34",
    "localDate": "2015-10-26",
    "localTime": "19:42:34",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:42:34Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:42:34",
    "gmtDayOfWeek": 1
  },
  "fileInfos": [
    {
      "fileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathFileName": "N/A",
      "storedPathSameAsFileName": false,
      "lastModificationDateString": "26-oct-2015 19:42:08",
      "lastModificationDate": "2015-10-26",
      "lastModificationDayOfWeek": 1,
      "lastModificationTime": "19:42:08",
      "owner": "root",
      "location": "DISK",
      "group": "root",
      "existingCopies": 0,
      "access": 664,
      "targetCopies": 1,
      "targetStubSize": 0,
      "targetStubScale": 1024,
      "existingStubSize": "n/a",
      "fileSize": 10485760,
      "store": "MINTIME",
      "affinity": "n/a",
      "reloc": "MINTIME",
      "class": "smpltomediamedia",
      "trunc": "MINTIME",
      "cleanDBInfo": "NO",
      "altStoreLocation": "Disabled",
      "medias": [
        { "message": "None" }
      ],
      "checksums": [

```

```

        { "summary": "N" }
    ],
    "encryptions": [
        { "summary": "N" }
    ],
    "compressions": [
        { "summary": "N" }
    ],
    "objects": [
        { "summary": "N" }
    ]
}
],
"statuses": [
    {
        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 26,
        "requestId": 176855,
        "commandName": "/usr/adic/TSM/bin/fsfileinfo",
        "commandStatus": "completed",
        "statusText": "Command Successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:42:34",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:42:34",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:42:34Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:42:34",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0006"
}
}

```

Now, try to find out what policy this file path is associated with. Use the **fsdirclass** web service.

```

String response = caller.getFsDirClass(true, "wsuser", "wspass",
    "192.168.36.128", "444", "text",
"/stornext/snfx1/smpltomediamedia");
System.out.println(response);

```

The output is shown below. Notice the path is associated with the **smpltomediamedia** policy.

```

{
  "header": {
    "commandName": "fsdirclass",
    "commandLine": "/usr/adic/TSM/bin/fsdirclass /stornext/snfx1/smpltome-
-F json",
    "commandDescription": "Report the policy class associated with a
directory.",
    "localDateISO": "2015-10-26T19:44:07",
    "localDate": "2015-10-26",
    "localTime": "19:44:07",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:44:07Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:44:07",
    "gmtDayOfWeek": 1
  },
  "directories": [
    {
      "directoryName": "/stornext/snfx1/smpltome-
",
      "classId": "smpltome-
"
    }
  ],
  "statuses": [
    {
      "statusCode": "FS0000",
      "statusNumber": 0,
      "dayOfMonth": 26,
      "requestId": 176858,
      "commandName": "/usr/adic/TSM/bin/fsdirclass",
      "commandStatus": "completed",
      "statusText": "Command Successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:44:07",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:44:07",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:44:07Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:44:07",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0001"
  }
}

```

Now, review the `smpltome-` policy. Use the `fsclassinfo` web service.

```
String response = caller.getFsClassInfo(true, "wsuser", "wspass",  
                                         "192.168.36.128", "444", "json", "smpltomedia");  
System.out.println(response);
```

The output is shown below.

```

{
  "header": {
    "commandName": "fsclassinfo",
    "commandLine": "/usr/adic/TSM/bin/fsclassinfo smpltomedia -F json",
    "commandDescription": "Report policy class processing parameters,
associated directory paths, and affinity lists.",
    "localDateISO": "2015-10-26T19:45:13",
    "localDate": "2015-10-26",
    "localTime": "19:45:13",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:45:13Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:45:13",
    "gmtDayOfWeek": 1
  },
  "classes": [
    {
      "classId": "smpltomedia",
      "softLimit": 20000,
      "hardLimit": 25000,
      "drivePool": "fs_F0drivepool",
      "securityCode": "NONE",
      "acctNumber": 12345,
      "defCopies": 1,
      "maxCopies": 4,
      "maxInactiveVersions": 10,
      "mediaType": "LTO",
      "fileCleanup": "MINTIME",
      "mediaCleanup": "SYSTEM",
      "storeMinTime": "5m",
      "storeMaxSetAge": "n/a",
      "storeMinSetSize": "n/a",
      "storeAutomatically": "yes",
      "relocMinTime": "7d",
      "truncMinTime": "3d",
      "generateChecksum": "DISABLED",
      "validateChecksum": "DISABLED",
      "cleanOnRemove": "DISABLED",
      "targetStubSize": 0,
      "encryption": "None",
      "altStoreLocation": "DISABLED",
      "masterKeyName": "",
      "compression": "None"
    }
  ],
  "statuses": [
    {

```

```

        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 26,
        "requestId": 176861,
        "commandName": "/usr/adic/TSM/bin/fsclassinfo",
        "commandStatus": "completed",
        "statusText": "Command Successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:45:13",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:45:13",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:45:13Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:45:13",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0002"
}
}

```

Store the foobar0 file using the **fsstore** web service; only store one copy of the file.

```

String response = caller.getFsStore(true, "wsuser", "wspass",
    "192.168.36.128", "444", "json",
    "/stornext/snfx1/smpltomediamedia/foobar0", 1);
System.out.println(response);

```

The output is shown below.

```

{
  "header": {
    "commandName": "fsstore",
    "commandLine": "/usr/adic/TSM/bin/fsstore -c 1 -F json
/stornext/snfx1/smpltomediam/foobar0",
    "commandDescription": "Request the storage of a file that currently
resides on disk to media",
    "localDateISO": "2015-10-26T19:46:30",
    "localDate": "2015-10-26",
    "localTime": "19:46:30",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:46:30Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:46:30",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0589",
      "statusNumber": 589,
      "dayOfMonth": 26,
      "requestId": 176864,
      "commandName": "/usr/adic/TSM/bin/fsstore",
      "commandStatus": "interim",
      "statusText": "Tertiary Manager software request received."
    },
    {
      "statusCode": "FS0799",
      "statusNumber": 799,
      "dayOfMonth": 26,
      "requestId": 176864,
      "commandName": "/usr/adic/TSM/bin/fsstore",
      "commandStatus": "interim",
      "statusText": "1 file store request(s) have been sent to Tertiary
Manager."
    },
    {
      "statusCode": "FS0346",
      "statusNumber": 346,
      "dayOfMonth": 26,
      "requestId": 176864,
      "commandName": "fsstore",
      "commandStatus": "interim",
      "statusText": "File /stornext/snfx1/smpltomediam/foobar0 copy 1 has
been stored."
    }
  ]
}

```



```

    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 26,
      "requestId": 176864,
      "commandName": "fsstore",
      "commandStatus": "completed",
      "statusText": "1 out of 1 statuses were successful."
    },
    {
      "statusCode": "FS0000",
      "statusNumber": 0,
      "dayOfMonth": 26,
      "requestId": 176864,
      "commandName": "fsstore",
      "commandStatus": "completed",
      "statusText": "Command Successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:46:33",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:46:33",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:46:33Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:46:33",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "3.0097"
  }
}

```

When the file is stored, execute the `fsfileinfo` web service again to ensure the file is stored.

```

String response = caller.getFsFileInfo(true, "wsuser", "wspass",
    "192.168.36.128", "444", "json",
    "/stornext/snfx1/smpltomedia/foobar0");
System.out.println(response);

```

Notice the location reads **DISK AND ARCHIVE**, which signifies the file is stored.

```

{
  "header": {
    "commandName": "fsfileinfo",
    "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json
/stornext/snfx1/smpltomediamedia/foobar0",
    "commandDescription": "Generate a report about files known to the
Tertiary Manager",
    "localDateISO": "2015-10-26T19:47:45",
    "localDate": "2015-10-26",
    "localTime": "19:47:45",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:47:45Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:47:45",
    "gmtDayOfWeek": 1
  },
  "fileInfos": [
    {
      "fileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathFileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathSameAsFileName": true,
      "lastModificationDateString": "26-oct-2015 19:42:08",
      "lastModificationDate": "2015-10-26",
      "lastModificationDayOfWeek": 1,
      "lastModificationTime": "19:42:08",
      "owner": "root",
      "location": "DISK AND ARCHIVE",
      "group": "root",
      "existingCopies": 1,
      "access": 664,
      "targetCopies": 1,
      "targetStubSize": 0,
      "targetStubScale": 1024,
      "existingStubSize": "n/a",
      "fileSize": 10485760,
      "store": "MINTIME",
      "affinity": "n/a",
      "reloc": "MINTIME",
      "class": "smpltomediamedia",
      "trunc": "MINTIME",
      "cleanDBInfo": "NO",
      "altStoreLocation": "Disabled",
      "medias": [
        { "mediaId": "000005", "copy": 1 }
      ],
      "checksums": [

```

```

        { "summary": "N" }
    ],
    "encryptions": [
        { "summary": "N" }
    ],
    "compressions": [
        { "summary": "N" }
    ],
    "objects": [
        { "summary": "N" }
    ]
}
],
"statuses": [
    {
        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 26,
        "requestId": 176872,
        "commandName": "/usr/adic/TSM/bin/fsfileinfo",
        "commandStatus": "completed",
        "statusText": "Command Successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:47:45",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:47:45",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:47:45Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:47:45",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0007"
}
}

```

Remove the disk copy of the file using the **fsrmdiskcopy** web service.

```

String response = caller.getFsRmDiskCopy(true, "wsuser", "wspass",
    "192.168.36.128", "444", "json",
    "/stornext/snfx1/smpltomedia/foobar0");
System.out.println(response);

```

The output is shown below.

```

{
  "header": {
    "commandName": "fsrmdiskcopy",
    "commandLine": "/usr/adic/TSM/bin/fsrmdiskcopy -F json
/stornext/snfx1/smpltomedia/foobar0",
    "commandDescription": "Remove file data blocks from disk after the file
was stored to a medium",
    "localDateISO": "2015-10-26T19:49:58",
    "localDate": "2015-10-26",
    "localTime": "19:49:58",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:49:58Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:49:58",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0266",
      "statusNumber": 266,
      "dayOfMonth": 26,
      "requestId": 176881,
      "commandName": "/usr/adic/TSM/bin/fsrmdiskcopy",
      "commandStatus": "interim",
      "statusText": "Data disk blocks for file
/stornext/snfx1/smpltomedia/foobar0 were successfully removed."
    },
    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 26,
      "requestId": 176881,
      "commandName": "/usr/adic/TSM/bin/fsrmdiskcopy",
      "commandStatus": "completed",
      "statusText": "1 out of 1 disk copy removes were successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:49:58",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:49:58",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:49:58Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:49:58",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0005"
  }
}

```

```
    }  
}
```

Verify the file is removed using the `fsfileinfo` command.

```
String response = caller.getFsFileInfo(true, "wsuser", "wspass",  
    "192.168.36.128", "444", "json",  
    "/stornext/snfx1/smpltomediamedia/foobar0");  
System.out.println(response);
```

Notice the location of the file is **ARCHIVE** which signifies the disk copy is removed.

```

{
  "header": {
    "commandName": "fsfileinfo",
    "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json
/stornext/snfx1/smpltomediamedia/foobar0",
    "commandDescription": "Generate a report about files known to the
Tertiary Manager",
    "localDateISO": "2015-10-26T19:53:15",
    "localDate": "2015-10-26",
    "localTime": "19:53:15",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:53:15Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:53:15",
    "gmtDayOfWeek": 1
  },
  "fileInfos": [
    {
      "fileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathFileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathSameAsFileName": true,
      "lastModificationDateString": "26-oct-2015 19:42:08",
      "lastModificationDate": "2015-10-26",
      "lastModificationDayOfWeek": 1,
      "lastModificationTime": "19:42:08",
      "owner": "root",
      "location": "ARCHIVE",
      "group": "root",
      "existingCopies": 1,
      "access": 664,
      "targetCopies": 1,
      "targetStubSize": 0,
      "targetStubScale": 1024,
      "existingStubSize": 0,
      "existingStubScale": 1024,
      "fileSize": 10485760,
      "store": "MINTIME",
      "affinity": "n/a",
      "reloc": "MINTIME",
      "class": "smpltomediamedia",
      "trunc": "MINTIME",
      "cleanDBInfo": "NO",
      "altStoreLocation": "Disabled",
      "medias": [
        { "mediaId": "000005", "copy": 1 }
      ]
    }
  ],
}

```

```

        "checksums": [
            { "summary": "N" }
        ],
        "encryptions": [
            { "summary": "N" }
        ],
        "compressions": [
            { "summary": "N" }
        ],
        "objects": [
            { "summary": "N" }
        ]
    }
],
"statuses": [
    {
        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 26,
        "requestId": 176888,
        "commandName": "/usr/adic/TSM/bin/fsfileinfo",
        "commandStatus": "completed",
        "statusText": "Command Successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:53:15",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:53:15",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:53:15Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:53:15",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0004"
}
}

```

Retrieve the file using the **fsretrieve** web service.

```

String response = caller.getFsRetrieve(true, "wsuser", "wspass",
    "192.168.36.128", "444", "json",
    "/stornext/snfx1/smpltmedia/foobar0");
System.out.println(response);

```

Notice the retrieve operation is successful.

```

{
  "header": {
    "commandName": "fsretrieve",
    "commandLine": "/usr/adic/TSM/bin/fsretrieve -F json
/stornext/snfx1/smpltmedia/foobar0",
    "commandDescription": "Retrieve files from media and place on disk",
    "localDateISO": "2015-10-26T19:54:45",
    "localDate": "2015-10-26",
    "localTime": "19:54:45",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:54:45Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:54:45",
    "gmtDayOfWeek": 1
  },
  "statuses": [
    {
      "statusCode": "FS0589",
      "statusNumber": 589,
      "dayOfMonth": 26,
      "requestId": 176893,
      "commandName": "/usr/adic/TSM/bin/fsretrieve",
      "commandStatus": "interim",
      "statusText": "Tertiary Manager software request received."
    },
    {
      "statusCode": "FS0347",
      "statusNumber": 347,
      "dayOfMonth": 26,
      "requestId": 176893,
      "commandName": "fsretrieve",
      "commandStatus": "interim",
      "statusText": "File /stornext/snfx1/smpltmedia/foobar0 has been
retrieved."
    },
    {
      "statusCode": "FS0390",
      "statusNumber": 390,
      "dayOfMonth": 26,
      "requestId": 176893,
      "commandName": "fsretrieve",
      "commandStatus": "completed",
      "statusText": "1 out of 1 retrieves were successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:54:46",

```



```
    "localDateEnd": "2015-10-26",  
    "localTimeEnd": "19:54:46",  
    "localDayOfWeekEnd": 1,  
    "gmtDateISOEnd": "2015-10-27T01:54:46Z",  
    "gmtDateEnd": "2015-10-27",  
    "gmtTimeEnd": "01:54:46",  
    "gmtDayOfWeekEnd": 1,  
    "elapsedTimeInSeconds": "0.0679"  
  }  
}
```

Execute `fsfileinfo` one last time to ensure the file is retrieved.

```
String response = caller.getFsFileInfo(true, "wsuser", "wspass",  
    "192.168.36.128", "444", "json",  
    "/stornext/snfx1/smpltomedia/foobar0");  
System.out.println(response);
```

Notice the location reads **DISK AND ARCHIVE** which signifies the file is retrieved.

```

{
  "header": {
    "commandName": "fsfileinfo",
    "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json
/stornext/snfx1/smpltomediamedia/foobar0",
    "commandDescription": "Generate a report about files known to the
Tertiary Manager",
    "localDateISO": "2015-10-26T19:57:04",
    "localDate": "2015-10-26",
    "localTime": "19:57:04",
    "localDayOfWeek": 1,
    "gmtDateISO": "2015-10-27T01:57:04Z",
    "gmtDate": "2015-10-27",
    "gmtTime": "01:57:04",
    "gmtDayOfWeek": 1
  },
  "fileInfos": [
    {
      "fileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathFileName": "/stornext/snfx1/smpltomediamedia/foobar0",
      "storedPathSameAsFileName": true,
      "lastModificationDateString": "26-oct-2015 19:42:08",
      "lastModificationDate": "2015-10-26",
      "lastModificationDayOfWeek": 1,
      "lastModificationTime": "19:42:08",
      "owner": "root",
      "location": "DISK AND ARCHIVE",
      "group": "root",
      "existingCopies": 1,
      "access": 664,
      "targetCopies": 1,
      "targetStubSize": 0,
      "targetStubScale": 1024,
      "existingStubSize": "n/a",
      "fileSize": 10485760,
      "store": "MINTIME",
      "affinity": "n/a",
      "reloc": "MINTIME",
      "class": "smpltomediamedia",
      "trunc": "MINTIME",
      "cleanDBInfo": "NO",
      "altStoreLocation": "Disabled",
      "medias": [
        { "mediaId": "000005", "copy": 1 }
      ],
      "checksums": [

```

```

        { "summary": "N" }
    ],
    "encryptions": [
        { "summary": "N" }
    ],
    "compressions": [
        { "summary": "N" }
    ],
    "objects": [
        { "summary": "N" }
    ]
}
],
"statuses": [
    {
        "statusCode": "FS0000",
        "statusNumber": 0,
        "dayOfMonth": 26,
        "requestId": 176898,
        "commandName": "/usr/adic/TSM/bin/fsfileinfo",
        "commandStatus": "completed",
        "statusText": "Command Successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-26T19:57:04",
    "localDateEnd": "2015-10-26",
    "localTimeEnd": "19:57:04",
    "localDayOfWeekEnd": 1,
    "gmtDateISOEnd": "2015-10-27T01:57:04Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "01:57:04",
    "gmtDayOfWeekEnd": 1,
    "elapsedTimeInSeconds": "0.0007"
}
}

```

There are several other operations to perform using the SWS V2 web services. For example, you can create a quota on a directory in a managed file system and set the quota limits.

**i Note:** Ensure quotas are enabled before you run the web services. For information on how to enable quotas, see the *StorNext 5 User's Guide* online at <http://www.quantum.com/sn5docs>.

Create a quota for a directory `/smpltomedia/media` in the `snfx1` filesystem. Refer to sample code above.

```

String response = caller.getCreateSnQuota(true, "wsuser", "wspass",
    "192.168.36.128", "444", "json", "/stornext/snfx1",
    "/smpltomedia/media");
System.out.println(response);

```

The output is shown below. An exit code of 0 signifies the quota is created.

```
{
  "returnCode": 0
}
```

Set the high limit to 10 GB, soft limit to 1 GB and grace period to 1 week.

```
String response = caller.getSetSnQuota(true, "wsuser", "wspass",
    "192.168.36.128", "444", "text", "/stornext/snfx1",
    "/smpltomedia/media", "10g", "1g", "1w");
System.out.println(response);
```

The output displays the quota limits are set.

```
{
  "directoryQuotas": [
    {
      "hardLimit": "10G",
      "softLimit": "1.0G",
      "gracePeriod": "1w",
      "curSize": 0,
      "status": "Under",
      "type": "dir",
      "name": "/smpltomedia/media"
    }
  ],
  "returnCode": 0
}
```

To verify, use the **snquota** web service to retrieve a list of quotas for snfx1.

```
String response = caller.getListSnQuota(true, "wsuser", "wspass",
    "192.168.36.128", "444", "text", "snfx1");
System.out.println(response);
```

The output shown below displays the quota values are set.

```

{
  "userQuotas": [
    {
      "hardLimit": 0,
      "softLimit": 0,
      "gracePeriod": "0m",
      "curSize": "4.6G",
      "status": "NoLimit",
      "type": "user",
      "name": "root"
    }
  ],
  "groupQuotas": [
    {
      "hardLimit": 0,
      "softLimit": 0,
      "gracePeriod": "0m",
      "curSize": "4.6G",
      "status": "NoLimit",
      "type": "group",
      "name": "root"
    }
  ],
  "directoryQuotas": [
    {
      "hardLimit": "10G",
      "softLimit": "1.0G",
      "gracePeriod": "1w",
      "curSize": 0,
      "status": "Under",
      "type": "dir",
      "name": "/smpltomedia/media"
    },
    {
      "hardLimit": 0,
      "softLimit": 0,
      "gracePeriod": "0m",
      "curSize": 0,
      "status": "NoLimit",
      "type": "dirfiles",
      "name": "/smpltomedia/media"
    }
  ],
  "returnCode": 0
}

```

You can also run SWS V2 web services asynchronously. This is especially helpful for long running processes like storing several files or retrieving them. First, write the code to send a `fsretrieve` request with async mode. For this example, we will assume that we are running a `fsretrieve` operation asynchronously on 10 files in a particular directory.

```

public String getFsRetrieveAsync(boolean useHttps, String userName,
    String password, String hostName, String port, String format) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType(format);
    MultivaluedMap<String, String> params = new MultivaluedMapImpl();
    for (int i=1; i<=10; i++) {
        params.add("file", "/stornext/snfx1/smp1data/foobar" + i);
    }

    String response = service.path("sws/v2/file/fsretrieve")
        .queryParams(params)
        .QueryParam("mode", "async")
        .QueryParam("format", format).accept(mediaType)
        .get(String.class);

    return response;
}

```

Notice the use of a `MultivaluedMap` to populate a list of 10 files. If you have just one file, ignore this. Next, the code must find the job status (for example, if the job is running or completed).

```

public String getJobStatus(boolean useHttps, String userName,
    String password, String hostName, String port, String jobID) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    String mediaType = getMediaType("text");
    String response = service.path("sws/v2/job/info")
        .QueryParam("job", jobID).accept(mediaType)
        .get(String.class);

    return response;
}

```

A detailed job status with interim transfer details is available for store (**fsstore**) and retrieve (**fsretrieve**) operations.

```

// Only available for fsstore and fsretrieve
public String getMoverJobStatus(boolean useHttps, String userName,
    String password, String hostName, String port, String... jobIDList) {
    ClientConfig config = getClientConfig();

    setupSSL(useHttps, config);
    Client client = configureClient(userName, password, config);
    WebResource service = client.resource(getBaseURI(hostName, port,
        useHttps));
    MultivaluedMap<String, String> params = new MultivaluedMapImpl();
    for (String job: jobIDList) {
        params.add("job", job);
    }
    String mediaType = getMediaType("json");
    String response = service.path("sws/v2/job/mover/info")
        .queryParams(params).accept(mediaType)
        .get(String.class);

    return response;
}

```

Below is an example of a job status query. The **job** parameter displays the job ID of the job. The **state** parameter displays the current state of the job. The valid values for state are **READY**, **QUEUED**, **RUNNING**, **COMPLETED** and **ERROR**.

```

{
  "jobList": [
    {
      "jobInfo": {
        "job": "27",
        "state": "RUNNING"
      }
    }
  ]
}

```

Below is an example of a detailed mover job status query.

Parameter	Description
"moverRequestList"	<p>This section displays the jobs that are executed by the storage manager. This section displays a <b>requestId</b> assigned by the storage manager.</p> <ul style="list-style-type: none"> <li>• The <b>requestType</b> can either be <b>Store</b> or <b>Retrieve</b>.</li> <li>• The <b>state</b> parameter displays the current state of the job. The valid values for <b>state</b> are <b>READY</b>, <b>QUEUE</b>, <b>PROCESS</b>, <b>FORMAT</b>, <b>COPY</b> and <b>COMPLETE</b>.</li> <li>• The <b>positioninqueue</b> parameter displays the position of the job in the storage manager queue.</li> </ul>
"moverProgressList"	<p>This section displays the job that is currently running in the storage manager.</p> <ul style="list-style-type: none"> <li>• The <b>host</b> parameter displays the hostname of the system in which this job is being executed. The storage manager assigns the <b>requestId</b> when the job is registered under storage manager for execution.</li> <li>• The <b>deviceAlias</b> is the alias for the drive on which this job is being executed.</li> <li>• The <b>runTime</b> parameter displays the time that has elapsed since the job started.</li> <li>• The <b>totalFiles</b> parameter displays the number of files that will be copied for this job.</li> <li>• The <b>filesCopied</b> parameter displays the files that have already been completed.</li> <li>• The <b>filesFailed</b> parameter displays the number of files that could not be copied.</li> </ul>
"completedJobList"	<p>This section displays the jobs that have completed execution and begins with header information in the header node.</p> <ul style="list-style-type: none"> <li>• The <b>jobInfo</b> node displays details about a particular job.</li> <li>• The <b>statuses</b> node contains individual status of files that are being copied in this particular job.</li> <li>• The footer section contains footer information.</li> </ul>
"pendingJobList"	<p>This section displays the jobs that are waiting to be executed by the storage manager.</p> <ul style="list-style-type: none"> <li>• The <b>job</b> parameter displays the job Id.</li> <li>• The <b>positioninagentqueue</b> is the position of the job in the agent queue.</li> <li>• The <b>exitcode</b> parameter displays the current exit code.</li> <li>• The <b>state</b> of waiting jobs is shown. In general, state for a waiting job is <b>QUEUED</b>.</li> </ul>



```
[
  {
    "moverRequestList": [
      {
        "requestId": "177193",
        "requestType": "Retrieve",
        "state": "COPY",
        "positioninqueue": 1
      },
      {
        "requestId": "177194",
        "requestType": "Retrieve",
        "state": "READY",
        "positioninqueue": 2
      },
      {
        "requestId": "177205",
        "requestType": "Retrieve",
        "state": "READY",
        "positioninqueue": 3
      },
      {
        "requestId": "177213",
        "requestType": "Retrieve",
        "state": "READY",
        "positioninqueue": 4
      }
    ],
    "moverProgressList": [
      {
        "host": "REDHAT5-DEMO",
        "requestId": "177193",
        "deviceAlias": "archives_dr1",
        "runTime": "00:00:03",
        "totalFiles": "3",
        "filesCopied": "1",
        "filesFailed": "0"
      }
    ]
  },
  {
    "completedJobList": [
      {
```

```

    "header": {
      "commandName": "fsretrieve",
      "commandLine": "/usr/adic/TSM/bin/fsretrieve -F json
/stornext/snfx1/smp2data/soobar1 /stornext/snfx1/smp2data/soobar2
/stornext/snfx1/smp2data/soobar3 /stornext/snfx1/smp2data/soobar4
/stornext/snfx1/smp2data/soobar5",
      "commandDescription": "Retrieve files from media and place on
disk",
      "localDateISO": "2015-10-27T17:05:09",
      "localDate": "2015-10-27",
      "localTime": "17:05:09",
      "localDayOfWeek": 2,
      "gmtDateISO": "2015-10-27T23:05:09Z",
      "gmtDate": "2015-10-27",
      "gmtTime": "23:05:09",
      "gmtDayOfWeek": 2
    },
    "jobInfo": {
      "job": "24",
      "exitcode": 0,
      "datecompleted": "2015-10-27 17:05:20",
      "state": "COMPLETED"
    },
    "statuses": [
      {
        "statusCode": "FS0005",
        "statusNumber": 5,
        "dayOfMonth": 27,
        "requestId": 177191,
        "commandName": "/usr/adic/TSM/bin/fsretrieve",
        "commandStatus": "interim",
        "statusText": "No retrieve needed, the data is already
present on the disk for file /stornext/snfx1/smp2data/soobar1."
      },
      {
        "statusCode": "FS0589",
        "statusNumber": 589,
        "dayOfMonth": 27,
        "requestId": 177191,
        "commandName": "/usr/adic/TSM/bin/fsretrieve",
        "commandStatus": "interim",
        "statusText": "Tertiary Manager software request
received."
      }
    ]
  }

```

```

        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,
            "requestId": 177191,
            "commandName": "fsretrieve",
            "commandStatus": "interim",
            "statusText": "File /stornext/snfx1/smp2data/soobar2 has
been retrieved."
        },
        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,
            "requestId": 177191,
            "commandName": "fsretrieve",
            "commandStatus": "interim",
            "statusText": "File /stornext/snfx1/smp2data/soobar3 has
been retrieved."
        },
        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,
            "requestId": 177191,
            "commandName": "fsretrieve",
            "commandStatus": "interim",
            "statusText": "File /stornext/snfx1/smp2data/soobar4 has
been retrieved."
        },
        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,
            "requestId": 177191,
            "commandName": "fsretrieve",
            "commandStatus": "interim",
            "statusText": "File /stornext/snfx1/smp2data/soobar5 has
been retrieved."
        },
        {
            "statusCode": "FS0654",
            "statusNumber": 654,
            "dayOfMonth": 27,

```

```

        "requestId": 177191,
        "commandName": "fsretrieve",
        "commandStatus": "completed",
        "statusText": "1 out of 5 files were already on disk."
    },
    {
        "statusCode": "FS0390",
        "statusNumber": 390,
        "dayOfMonth": 27,
        "requestId": 177191,
        "commandName": "fsretrieve",
        "commandStatus": "completed",
        "statusText": "5 out of 5 retrieves were successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-27T17:05:20",
    "localDateEnd": "2015-10-27",
    "localTimeEnd": "17:05:20",
    "localDayOfWeekEnd": 2,
    "gmtDateISOEnd": "2015-10-27T23:05:20Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "23:05:20",
    "gmtDayOfWeekEnd": 2,
    "elapsedTimeInSeconds": "10.0878"
}
},
{
    "header": {
        "commandName": "fsretrieve",
        "commandLine": "/usr/adic/TSM/bin/fsretrieve -F json
/stornext/snfx1/smp2data/foobar1 /stornext/snfx1/smp2data/foobar2
/stornext/snfx1/smp2data/foobar3",
        "commandDescription": "Retrieve files from media and place on
disk",
        "localDateISO": "2015-10-27T17:05:09",
        "localDate": "2015-10-27",
        "localTime": "17:05:09",
        "localDayOfWeek": 2,
        "gmtDateISO": "2015-10-27T23:05:09Z",
        "gmtDate": "2015-10-27",
        "gmtTime": "23:05:09",

```

```

        "gmtDayOfWeek": 2
    },
    "jobInfo": {
        "job": "25",
        "exitcode": 0,
        "datecompleted": "2015-10-27 17:05:26",
        "state": "COMPLETED"
    },
    "statuses": [
        {
            "statusCode": "FS0589",
            "statusNumber": 589,
            "dayOfMonth": 27,
            "requestId": 177192,
            "commandName": "/usr/adic/TSM/bin/fsretrieve",
            "commandStatus": "interim",
            "statusText": "Tertiary Manager software request
received."
        },
        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,
            "requestId": 177192,
            "commandName": "fsretrieve",
            "commandStatus": "interim",
            "statusText": "File /stornext/snfx1/smp2data/foobar1 has
been retrieved."
        },
        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,
            "requestId": 177192,
            "commandName": "fsretrieve",
            "commandStatus": "interim",
            "statusText": "File /stornext/snfx1/smp2data/foobar2 has
been retrieved."
        },
        {
            "statusCode": "FS0347",
            "statusNumber": 347,
            "dayOfMonth": 27,

```

```

        "requestId": 177192,
        "commandName": "fsretrieve",
        "commandStatus": "interim",
        "statusText": "File /stornext/snfx1/smp2data/foobar3 has
been retrieved."
    },
    {
        "statusCode": "FS0390",
        "statusNumber": 390,
        "dayOfMonth": 27,
        "requestId": 177192,
        "commandName": "fsretrieve",
        "commandStatus": "completed",
        "statusText": "3 out of 3 retrieves were successful."
    }
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2015-10-27T17:05:26",
    "localDateEnd": "2015-10-27",
    "localTimeEnd": "17:05:26",
    "localDayOfWeekEnd": 2,
    "gmtDateISOEnd": "2015-10-27T23:05:26Z",
    "gmtDateEnd": "2015-10-27",
    "gmtTimeEnd": "23:05:26",
    "gmtDayOfWeekEnd": 2,
    "elapsedTimeInSeconds": "16.0902"
}
}
]
},
{
    "pendingJobList": [
        {
            "jobInfo": {
                "job": "30",
                "positioninagentqueue": 1,
                "exitcode": 0,
                "state": "QUEUED"
            }
        }
    ]
}
]

```

## Perl

See [Sample Perl Script on page 292](#).

## Python

See [Sample Python Script on page 308](#).

# Troubleshooting

Issue	Tips
Web Service Reported Off	On the StorNext GUI, click <b>Services</b> , and then click <b>Web Services (V2)</b> . Ensure the <b>State</b> parameter is enabled
Browser Error on Invoking a Web Service	<p><b>Check the protocol and the port number for the request.</b></p> <p>If the protocol and the port number do not match the web server configuration, SWS V2 does not receive the request. For example, if the <b>Protocol</b> is <b>http</b> and port used is <b>443</b>, the web service request may not work.</p>
Web Service is Not Working	<p><b>The root cause may be that the web server that hosts the web services is offline.</b></p> <p>Ensure you can start the StorNext GUI using the same IP and port. If the StorNext GUI cannot be started, the web server is down. Contact your System Administrator.</p>
Sent a XML Request, but Response is Not XML	<p><b>There are certain web services that only support TEXT and JSON.</b></p> <p>For information on the supported formats for a particular web service, see the <i>StorNext Web Services Guide</i> online at <a href="http://www.quantum.com/sn5docs">http://www.quantum.com/sn5docs</a>.</p> <p>Quantum recommends <b>JSON</b> as the preferred format since most web services support this format. <b>XML</b> is also supported, but it is primarily used for compatibility with existing applications.</p>



## Chapter 2: StorNext Web Services Commands (V1)

---

This chapter contains the following topics:

StorNext Web Services V1 Commands, Usage, and Descriptions .....	238
--	-----

---

### StorNext Web Services V1 Commands, Usage, and Descriptions

This section provides API commands supported by the Web Services (V1) for releases prior to StorNext 5 release 5.1, along with respective:

- Description
- Parameters (**input, output**)
- Return Values
- Calling Sequence
- Usage Scenario
- Examples and Sample Output
- Related APIs (**where applicable**)
- Notes and Warnings (**where applicable**)



## Using the APIs

In order to perform any command on the remote server, a matching password must be specified with the call. The server verifies this password against the one stored in the `/usr/adic/.snapipassword` file on the server.

Make sure this file is the same on all metadata controllers (MDCs) you will be accessing. This is especially important if you are using virtual IP (VIP) addresses, because you can be communicating to different machines across different API calls or even retries of the same call.

The `.snapipassword` file is stored as clear text, and it should be set to be readable only by root. If this file does not exist, no calls will succeed.

### Description for Num in the Parameters Tables

The **Num** field in the parameters table displays if the parameter is a single value or a list of values can be passed.

If the parameter only accepts one value, **1** is used for **Num**. If a list of values is allowed, **N** is used for **Num**.

An example of a web service which accepts a list of values is illustrated below.

```
https://<<SERVER>>/sws/v1/getmediastatus  
?password=<<PASSWORD>>  
&mediaID=000001  
&mediaID=000002
```

## API Commands

### Backup

This API initiates a backup operation.

#### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.

#### Example

```
https://<<SERVER>>/sws/v1/backup  
?password=<<PASSWORD>>
```

#### Output

```
<?xml version="1.0"?>  
<RESPONSE name="Backup" statusCode="0" status="SUCCESS" statusDescription="Backup  
initiation successful">  
<STATUSDETAIL statusCode="0" status="SUCCESS" statusDescription="Backup  
initiation successful"/>  
</RESPONSE>
```

## Check out Media

This API allows you to check out media with the specified media IDs.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	N	This is the list of media IDs that need to be checked out.

### Example

```
https://<<SERVER>>/sws/v1/checkoutmedia  
?password=<<PASSWORD>>  
&mediaid=000001
```

### Output

```
<?xml version="1.0"?>  
<RESPONSE name="CheckOutMedia" statusCode="0" status="SUCCESS" statusDescription=  
"Command Successful">  
<STATUSDETAIL name="mediaID" value="000001" statusCode="0" status="SUCCESS" statu  
sDescription="Command Successful"/>  
</RESPONSE>
```

## Clean Media

This API cleans media by removing inactive files from the specified media.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	N	This is the list of media IDs that need to be cleaned.

### Example

```
https://<<SERVER>>/sws/v1/cleanmedia  
?password=<<PASSWORD>>  
&mediaid=025311  
&mediaid=025312
```

### Output

```
<?xml version="1.0">  
<RESPONSE name="CleanMedia" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="mediaID" value="025311"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
<STATUSDETAIL name="mediaID" value="025312"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Copy Media

This API copies the content of all specified media to a piece of blank media.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	N	This is the list of media IDs that need to be copied.

### Example

```
https://<<SERVER>>/sws/v1/copymedia  
?password=<<PASSWORD>>  
&mediaid=025311  
&mediaid=025312
```

## Output

```
<?xml version="1.0">  
<RESPONSE name="CopyMedia" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="mediaID" value="025311"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
<STATUSDETAIL name="mediaID" value="025312"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Eject Media

This API moves the specified media from the archive to a media I/E slot. **EjectMedia** does an automated media eject. This API must be used with the [Move Media on page 259](#) API.

After running this API, you must call the [Enter Media on the next page](#) API to physically move the media from the mailbox into the destination library, or in the case of a vault, to logically enter the media into the vault.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	1	The ID of the media to eject.
portid	Required	1	The port used for ejecting media. The port ID is in the format 0,0,15,###. Only the digits following the final comma are needed, indicated as ### in the format example. It is not needed for vault archives.

### Example

```
https://<<SERVER>>/sws/v1/ejectmedia  
?password=<<PASSWORD>>  
&mediaid=012345  
&portid=16
```

## Output

```
<?xml version="1.0">  
<RESPONSE name="EjectMedia" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="mediaID" value="012345"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Enter Media

This API adds (inserts) media into the specified archive from the I/E slot. This API must be used after you run the [Eject Media on the previous page](#) API, and must be used with the [Move Media on page 259](#) API.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
archiveid	Required	1	The archive ID to which media is added.
mediaid	Required	1	The media ID to insert into the archive.
portid	Required	1	The port used for ejecting media. The port ID is in the format 0,0,15,###. Only the digits following the final comma are needed, indicated as ### in the format example. It is not needed for vault archives.

### Example

```
https://<<SERVER>>/sws/v1/entermedia  
?password=<<PASSWORD>>  
&archiveid=2  
&mediaid=012345  
&portid=16
```

## Output

```
<?xml version="1.0">  
<RESPONSE name="EnterMedia" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="mediaID" value="012345"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## File Retrieve

This API retrieves the specified single file from secondary storage to primary storage.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
filename	Required	1	The pathname of the file to be retrieved.
newfilename	Optional	1	The name you want to give the retrieved file. The default value is an empty string, which means the file data is retrieved to disk using the original file name. This argument is optional if used by itself, but required if you use <code>startByte</code> and <code>endByte</code> .
startbyte	Optional	1	The starting byte for a partial retrieval. The default value is 0, which means a full retrieval is expected. Do not use this argument (in conjunction with the <code>endByte</code> argument) if you use the <code>copyId</code> argument.
endbyte	Optional	1	The end byte for a partial retrieval. The default value is 0, which means a full retrieval is expected. Do not use this argument (in conjunction with the <code>startByte</code> argument) if you use the <code>copyId</code> argument.
modaccesstime	Optional	1	true or false. Indicates whether the retrieve will modify the file's access time. The default value is false.

Parameter	Req / Opt	Num	Description
copyid	Optional	1	Indicates which copy to be retrieved. The default value is 0, which refers to the primary copy. Do not use this argument if you use the startByte and endByte arguments.

### Example

```
https://<<SERVER>>/sws/v1/fileretrieve  
?password=<<PASSWORD>>  
&filename=/csofs/storage/temp
```

### Output

```
<?xml version="1.0">  
<RESPONSE name="FileRetrieve" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<RETRIEVEINFO name="filename" value="/csofs/storage/temp"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful" />  
</RESPONSE>
```

## Get Archive Capacity

This API provides the amount of remaining storage capacity for all archives.

The value returned in **remainingMediaCapacity** is in gigabytes (GB).

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.

### Example

```
https://<<SERVER>>/sws/v1/getarchivecapacity  
?password=<<PASSWORD>>
```

### Output

```
<?xml version="1.0"?>  
<RESPONSE name="GetArchiveCapacity" statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<STRINGINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<INFO name="remainingMediaCapacity" value="6800"/>  
</STRINGINFO>  
</RESPONSE>
```

## Get Archive List

This API provides the current configuration settings for all archives.

The values returned in **totalSpace** and **remainingSpace** is in gigabytes (GB).

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.

### Example

```
https://<<SERVER>>/sws/v1/getarchivelist  
?password=<<PASSWORD>>
```

### Output



```
<?xml version="1.0"?>
<RESPONSE name="GetArchiveList" statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<ARCHIVEINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<INFO name="archiveName" value="mylib"/>
<INFO name="state" value="online"/>
<INFO name="archiveType" value="SCSI"/>
<INFO name="serialNumber" value="0000000091_Defau"/>
<INFO name="model" value="Scalar 1000"/>
<INFO name="numberSlots" value="40"/>
<INFO name="numberSlotsUsed" value="17"/>
<INFO name="firmwareVersion" value="2.23"/>
<INFO name="totalSpace" value="6800"/>
<INFO name="remainingSpace" value="6800"/>
</ARCHIVEINFO>
</RESPONSE>
```

## Get Backup Status

This API provides the current status or progress of a backup in progress.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.

### Example

```
https://<<SERVER>>/sws/v1/getbackupstatus
?password=<<PASSWORD>>
```

### Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<RESPONSE name="GetBackupStatus" statusCode="0" status="SUCCESS"
statusDescription="Command Successful">
<STRINGINFO statusCode="0" status="SUCCESS"
statusDescription="Command Successful">
<!-- An INFO tag will surround each output line. -->
<INFO value="Response text goes here."/>
<INFO value="Response text goes here."/>
<INFO value="Response text goes here."/>
<INFO value="Response text goes here."/>
</STRINGINFO>
</RESPONSE>
```

## Get Drive List

This API provides a list of available drives and their attributes.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.

### Example

```
https://<<SERVER>>/sws/v1/getdrivelist
?password=<<PASSWORD>>
```

### Output

```
<?xml version="1.0"?>
<RESPONSE name="GetDriveList" statusCode="0" status="SUCCESS" statusDescription="
Command Successful">
<DRIVEINFO statusCode="0" status="SUCCESS" statusDescription="Command
Successful">
<INFO name="driveName" value="mylib_dr1"/>
<INFO name="state" value="ON"/>
<INFO name="serialNumber" value="ADIC000094"/>
<INFO name="type" value="LTO"/>
<INFO name="mountState" value="FREE"/>
<INFO name="mediaID" value="NONE"/>
<INFO name="firmwareVersion" value="FW3W"/>
</DRIVEINFO>
<DRIVEINFO statusCode="0" status="SUCCESS" statusDescription="Command
Successful">
<INFO name="driveName" value="mylib_dr2"/>
<INFO name="state" value="ON"/>
<INFO name="serialNumber" value="ADIC000095"/>
<INFO name="type" value="LTO"/>
<INFO name="mountState" value="FREE"/>
<INFO name="mediaID" value="NONE"/>
<INFO name="firmwareVersion" value="FW3W"/>
</DRIVEINFO>
<DRIVEINFO statusCode="0" status="SUCCESS" statusDescription="Command
Successful">
<INFO name="driveName" value="mylib_dr3"/>
<INFO name="state" value="ON"/>
<INFO name="serialNumber" value="ADIC000092"/>
<INFO name="type" value="LTO"/>
<INFO name="mountState" value="FREE"/>
<INFO name="mediaID" value="NONE"/>
<INFO name="firmwareVersion" value="FW3W"/>
</DRIVEINFO>
<DRIVEINFO statusCode="0" status="SUCCESS" statusDescription="Command
```

```
Successful">  
<INFO name="driveName" value="mylib_dr4"/>  
<INFO name="state" value="ON"/>  
<INFO name="serialNumber" value="ADIC000093"/>  
<INFO name="type" value="LTO"/>  
<INFO name="mountState" value="FREE"/>  
<INFO name="mediaID" value="NONE"/>  
<INFO name="firmwareVersion" value="FW3W"/>  
</DRIVEINFO>  
</RESPONSE>
```

## Get File Attribute

This API returns attribute information for the specified file, which is located in either primary storage or secondary storage (tape or storage disk).

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
filename	Required	1	The pathname of the file.

### Example

```
https://<<SERVER>>/sws/v1/getfileattribute  
?password=<<PASSWORD>>  
&filename=/stornext/snfs2/myfiles/cfg.sql
```

### Output

```
<?xml version="1.0"?>
<RESPONSE name="GetFileAttribute" statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<FILEINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<INFO name="fileName" value="/stornext/snfs2/myfiles/cfg.sql"/>
<INFO name="location" value="DISK AND TAPE"/>
<INFO name="policyClass" value="pc1"/>
<INFO name="numberExistingCopies" value="2"/>
<INFO name="numberTargetCopies" value="3"/>
<INFO name="mediaID" value="mysdisk(2)"/>
<INFO name="mediaID" value="000000(3)"/>
</FILEINFO>
</RESPONSE>
```

## Get File Tape Location

This API returns location information for the specified file, including the copy ID, segment number, starting block, and segment size.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
filename	Required	1	The pathname of the file.
copyid	Optional	1	Indicates which copy from which to get segment information. The default value is 1, the primary copy.

### Example

```
https://<<SERVER>>/sws/v1/getfiletapelocation
?password=<<PASSWORD>>
&filename=/snfs/myDirectory/myFile.dat
```

### Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<RESPONSE name="GetFileTapeLocation" statusCode="0"
status="SUCCESS"
statusDescription="Command Successful">
<LOCATIONINFO statusCode="0" status="SUCCESS"
statusDescription="Command Successful">
<INFO name="fileName" value="/stornnnnext/snfs1/d1/regfile" />
<INFO name="copyID" value="1" />
<SEGMENTINFO name="segmentNumber" value="1">
<INFO name="mediaID" value="000455" />
<INFO name="archiveID" value="scsi_archive1" />
<INFO name="startBlock" value="7" />
<INFO name="offset" value="128" />
<INFO name="segmentSize" value="100000" />
<INFO name="blockSize" value="524288" />
</SEGMENTINFO>
<SEGMENTINFO name="segmentNumber" value="2">
<INFO name="mediaID" value="000455" />
<INFO name="archiveID" value="scsi_archive1" />
<INFO name="startBlock" value="7" />
<INFO name="offset" value="100256" />
<INFO name="segmentSize" value="100000" />
<INFO name="blockSize" value="524288" />
</SEGMENTINFO>
<SEGMENTINFO name="segmentNumber" value="3">
<INFO name="mediaID" value="000455" />
<INFO name="archiveID" value="scsi_archive1" />
<INFO name="startBlock" value="7" />
<INFO name="offset" value="200384" />
<INFO name="segmentSize" value="100000" />
<INFO name="blockSize" value="524288" />
</SEGMENTINFO>
```

```
</LOCATIONINFO>  
</RESPONSE>
```

This API returns either a list of media IDs or the number of media that meets the specified criteria. All arguments are optional. If you do not specify any arguments, the total number of media is returned.

## Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
format	Optional	1	Valid values are <b>count</b> or <b>list</b> .
location	Optional	1	Valid values are <b>archive</b> , <b>vault</b> , <b>checkout</b> , or <b>unknown</b> . If no argument is specified, archive is used as the default value.
archiveid	Optional	1	The archive ID.
classification	Optional	1	Valid values are <b>data</b> , <b>backup</b> , or <b>cleaning</b> .
availability	Optional	1	Valid values are <b>available</b> or <b>unavailable</b> .
writeaccess	Optional	1	Valid values are <b>writeprotected</b> or <b>notwriteprotected</b> .
integrity	Optional	1	Valid values are <b>suspect</b> or <b>notSuspect</b> .
space	Optional	1	Valid values are <b>full</b> or <b>blank</b> .
percentused	Optional	1	The percentage of media space used, from 0.00 to 100.00.
copyid	Optional	1	The number of the copy used, from 1 - 8.

## Example

```
https://<<SERVER>>/sws/v1/getmedialist  
?password=<<PASSWORD>>  
&format=count  
&location=archive  
&availability=available
```

## Output

```
<?xml version="1.0"?>  
<RESPONSE name="GetMediaList" statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<MEDIALIST statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<INFO name="location" value="archive"/>  
<INFO name="availability" value="available"/>  
<INFO name="count" value="16"/>  
</MEDIALIST>  
</RESPONSE>
```

## Get Media Status

This API returns status for the specified piece of media.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	N	A list of mediaIDs that are being queried.

### Example

```
https://<<SERVER>>/sws/v1/getmediastatus  
?password=<<PASSWORD>>  
&mediaID=000001  
&mediaID=000002
```

### Output



```
<?xml version="1.0"?>
<RESPONSE name="GetMediaStatus" statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<MEDIAINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<INFO name="mediaID" value="000001"/>
<INFO name="policyClass" value="_adic_backup"/>
<INFO name="location" value="mylib"/>
<INFO name="type" value="LTO"/>
<INFO name="writeProtected" value="N"/>
<INFO name="mountCount" value="99"/>
<INFO name="status" value="AVAIL"/>
<INFO name="suspectCount" value="0"/>
<INFO name="lastAccessTime" value="11-dec-2013 11:08:40"/>
<INFO name="spaceUsed" value="818516168"/>
<INFO name="spaceFree" value="178639601664"/>
<INFO name="percentUsed" value="0.46"/>
<INFO name="classification" value="backup"/>
</MEDIAINFO>
<MEDIAINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">
<INFO name="mediaID" value="000002"/>
<INFO name="policyClass" value="_adic_backup"/>
<INFO name="location" value="mylib"/>
<INFO name="type" value="LTO"/>
<INFO name="writeProtected" value="N"/>
<INFO name="mountCount" value="2"/>
<INFO name="status" value="AVAIL"/>
<INFO name="suspectCount" value="0"/>
<INFO name="lastAccessTime" value="11-dec-2013 12:30:29"/>
<INFO name="spaceUsed" value="148167235"/>
<INFO name="spaceFree" value="179839172608"/>
<INFO name="percentUsed" value="0.08"/>
```

```
<INFO name="classification" value="backup"/>  
</MEDIAINFO>  
</RESPONSE>
```

## Get Policy

This API retrieves the specified policy.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
policyclass	Optional	1	A valid policy class name or "all" will get all policy classes. If no argument is specified, all is used as the default value.

### Example

```
https://<<SERVER>>/sws/v1/getpolicy  
?password=<<PASSWORD>>  
&policyclass=pc1
```

### Output

```
<?xml version="1.0"?>  
<RESPONSE name="GetPolicy" statusCode="0" status="SUCCESS" statusDescription="Com  
mand Successful">  
<POLICYINFO statusCode="0" status="SUCCESS" statusDescription="Command  
Successful">  
<INFO name="policyClass" value="pc1"/>  
<INFO name="numberCopies" value="3"/>  
<INFO name="maxInactiveVersions" value="10"/>  
</POLICYINFO>  
</RESPONSE>
```

## Get Port List

This API retrieves the import/export port IDs for a specified archive. An archive can have multiple ports.

**Note:** Vaults do not have a port, so the portID is always 0.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
archiveid	Required	1	The ID of the archive whose import/export port information you want to retrieve.

### Example

```
https://<<SERVER>>/sws/v1/getportlist  
?password=<<PASSWORD>>  
&archiveid=mylib
```

### Output

```
<?xml version="1.0"?>  
<RESPONSE name="GetPortList" statusCode="0" status="SUCCESS" statusDescription="C  
ommand Successful">  
<PORTINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<INFO name="portID" value="16"/>  
</PORTINFO>  
</RESPONSE>
```

## Get Schedule

This API provides information about previously scheduled events such as backups.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
scheduletype	Required	1	The only valid value is <b>backup</b> .

### Example

```
https://<<SERVER>>/sws/v1/getschedule  
?password=<<PASSWORD>>  
&scheduletype=backup
```

## Output

```
<?xml version="1.0"?>  
<RESPONSE name="GetSchedule" statusCode="0" status="SUCCESS" statusDescription="C  
ommand Successful">  
<SCHEDULEINFO statusCode="0" status="SUCCESS" statusDescription="Command  
Successful">  
<INFO name="scheduleType" value="backup"/>  
<INFO name="scheduleTime" value="23:59"/>  
<INFO name="scheduleLastAttempt" value="Dec 10,2013 23:00"/>  
<INFO name="scheduleLastStatus" value="Successful"/>  
</SCHEDULEINFO>  
</RESPONSE>
```

## Get System Status

This API provides the current system status.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.

### Example

```
https://<<SERVER>>/sws/v1/getsystemstatus  
?password=<<PASSWORD>>
```

## Output

```
<?xml version="1.0"?>  
<RESPONSE name="GetSystemStatus" statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<SYSTEMINFO statusCode="0" status="SUCCESS" statusDescription="Command Successful">  
<INFO name="version" value="5.0.0(38795)"/>  
<INFO name="systemState" value="online"/>  
<INFO name="tsm" value="online"/>  
<INFO name="msm" value="online"/>  
<INFO name="dsm" value="online"/>  
<INFO name="database" value="online"/>  
<INFO name="svclog" value="online"/>  
</SYSTEMINFO>  
</RESPONSE>
```

## Move Media

This API logically marks a media to move from one archive to another. It does not do the physical moving. In order to both physically and logically move a media between two archives, you must take three steps:

1. Call **MoveMedia**.
2. Call **EjectMedia** to physically move the media to the mailbox at the source archive.
3. Call **EnterMedia** to physically move the media from mailbox into the destination archive.

**i Note:** If the media is entered or ejected from a vault, **EjectMedia** and **EnterMedia** are logical operations.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	1	The media ID to move.
destarchiveid	Required	1	The destination archive ID to move to.

### Example

```
https://<<SERVER>>/sws/v1/movemedi  
?password=<<PASSWORD>>  
&mediaid=000455  
&destarchiveid=vault1
```

## Output

```
<?xml version="1.0">  
<RESPONSE name="MoveMedia" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="mediaID" value="000455"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Remove Disk Copy

This API removes the disk copy of the specified file through explicit truncation.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
filename	Required	1	The pathname of the file whose on-disk copy will be removed.

### Example

```
https://<<SERVER>>/sws/v1/rmdiskcopy  
?password=<<PASSWORD>>  
&filename=/snfs/myDirectory/myFile.dat
```

## Output

```
<?xml version="1.0">
<RESPONSE name="RmDiskCopy" statusCode="0" status="SUCCESS"
statusDescription="Command Successful">
<STATUSDETAIL name="filename" value="/snfs/myDirectory/
myFile.dat"
statusCode="0" status="SUCCESS"
statusDescription="Command Successful"/>
</RESPONSE>
```

## Set Archive State

This API allows you to set the archive state to on or off.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
archivename	Required	1	The name of the archive for which you want to set the state.
state	Required	1	<b>on or off.</b>

### Example

```
https://<<SERVER>>/sws/v1/setarchivestate
?password=<<PASSWORD>>
&archivename=mylib
&state=on
```

### Output

```
<?xml version="1.0"?>
<RESPONSE name="SetArchiveState" statusCode="0" status="SUCCESS" statusDescriptio
n="Command Successful">
<STATUSDETAIL name="archiveName" value="mylib" statusCode="0" status="SUCCESS" st
atusDescription="Command Successful"/>
</RESPONSE>
```

## Set Directory Attributes

This API allows you to set the following directory attributes:

- **store** (enable or disable)
- **truncate** (enable or disable)
- policy class name for the directory

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
directoryname	Required	1	The name of the directory for which you want to set attributes.
nottruncate	Optional	1	<b>true</b> or <b>false</b> .
nostore	Optional	1	<b>true</b> or <b>false</b> .
policyclass	Optional	1	The name of the policy class you want to apply to the directory.

### Example

```
https://<<SERVER>>/sws/v1/setdirattributes  
?password=<<PASSWORD>>  
&directoryname=/snfs/mydir  
&nottruncate=true  
&nostore=true  
&policyclass=mypolicyclass
```

### Output

```
<?xml version="1.0">  
<RESPONSE name="SetDirAttributes" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="directoryName" value="/snfs/mydir"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Set Drive State

This API allows you to set the drive state to on or off.

### Parameters



Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
drivename	Required	1	The name of the drive whose state you want to set.
state	Required	1	<b>on or off.</b>

### Example

```
https://<<SERVER>>/sws/v1/setdrivestate  
?password=<<PASSWORD>>  
&drivename=lto2-001  
&state=on
```

### Output

```
<?xml version="1.0"?>  
<RESPONSE name="SetDriveState" statusCode="0" status="SUCCESS" statusDescription=  
"Command Successful">  
<STATUDETAIL name="driveName" value="lto2-  
001" statusCode="0" status="SUCCESS" statusDescription="Command Successful"/>  
</RESPONSE>
```

## Set File Attributes

This API allows you to set the following attributes for a file:

- **store** (enable or disable)
- **truncate** (enable or disable)

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
filename	Required	1	The name of the file for which you want to set attributes.
nottruncate	Optional	1	<b>true or false.</b>
nostore	Optional	1	<b>true or false.</b>

### Example

```
https://<<SERVER>>/sws/v1/setfileattributes  
?password=<<PASSWORD>>  
&filename=/snfs/mydir/myFile.dat  
&nottruncate=true  
&nostore=true
```

## Output

```
<?xml version="1.0">  
<RESPONSE name="SetFileAttributes" statusCode="0"  
status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="filename" value="/snfs/mydir/  
myFile.dat"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Set Media State

This API allows you to set the state for one or more piece of media.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
mediaid	Required	N	The ID of the media whose state you want to set.
state	Required	1	Valid values are <b>avail</b> (available), <b>unavail</b> (not available), <b>protect</b> (write protected), <b>unprotect</b> (not write protected), <b>unmark</b> (not marked), or <b>unsusp</b> (not suspect).

### Example

```
https://<<SERVER>>/sws/v1/setmediastate  
?password=<<PASSWORD>>  
&mediaid=004351  
&state=avail
```

## Output

```
<?xml version="1.0"?>  
<RESPONSE name="SetMediaState" statusCode="0" status="SUCCESS" statusDescription=  
"Command Successful">  
<STATUSDETAIL name="mediaID" value="004351" statusCode="0" status="SUCCESS" statu  
sDescription="Command Successful"/>  
</RESPONSE>
```

## Set Policy

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
policyclass	Required	1	The name of the policy class for which you want to set arguments, or enter "all" to apply arguments to all policy classes.
numberofcopies	Optional	1	The number of copies to maintain for the policy class.
maxinactiveversions	Optional	1	The maximum number of versions to maintain for the policy class.

### Example

```
https://<<SERVER>>/sws/v1/setpolicy  
?password=<<PASSWORD>>  
&policyclass=myPolicyClass  
&numberofcopies=2  
&maxinactiveversions=10
```

### Output

```
<?xml version="1.0">  
<RESPONSE name="SetPolicy" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful">  
<STATUSDETAIL name="mypolicyClass" value="all"  
statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```

## Set Schedule

This API allows you to specify a schedule for a backup. Running this API affects only system default full backups and default partial backups, not any user-configured backup events.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	A password that is required to run the web service.
schedulertype	Required	1	<b>backup</b> or other scheduled event type.
timeofday	Required	1	The time the scheduled event begins, in 24-hour format (HH:MM).

### Example

```
https://<<SERVER>>/sws/v1/setschedule  
?password=<<PASSWORD>>  
&schedulertype=backup  
&timeofday=23:59
```

### Output

```
<?xml version="1.0"?>  
<RESPONSE name="SetSchedule" statusCode="0" status="SUCCESS" statusDescription="C  
ommand Successful">  
<STATUSDETAIL name="scheduleType" value="backup" statusCode="0" status="SUCCESS"  
statusDescription="Command Successful"/>  
</RESPONSE>
```



# Chapter 3: StorNext 4.x Web Services (WSAPI)

---

This chapter contains the following topics:

StorNext 4.x Web Services (WSAPI) Commands, Usage, and Description .....	267
--	-----

---

## StorNext 4.x Web Services (WSAPI) Commands, Usage, and Description

**i Note:** Web services in this topic were introduced in StorNext 4.0. Web services in this topic are still supported in StorNext 5. If you are creating new integrations, use the **Web Services (V1)** commands, or the **Web Services (V2)** commands.

The Web Services Application Programming Interface feature (WS-API) provides an HTTP-based interface to a subset of the StorNext Storage Manager User Commands. WS-API provides basic control over StorNext Storage Manager systems to track media and drives, and to store/truncate/retrieve files from any computer capable of creating a Web Services connection, which includes Windows, Macintosh, and Linux-based systems, among others.

When first released in StorNext 4.0, WS-API returned the text-format output of the commands. The most recent WS-API release changes the commands listed below to add structured Extensible Markup Language (XML) or JavaScript Object Notation (JSON) optional output formats.

The following subset of Storage Manager Commands supports the -F parameter to select XML, JSON or text output, and their associated WS-API interfaces (in parentheses):

- `fscancel` (`doCancel`)
- `fsfileinfo` (`getFileLocation`)
- `fsfiletapeloc` (`getFileTapeLocation`)
- `fsmedinfo` (`getMediaReport`)
- `fsmedlist` (`getMediaInfo`)
- `fsqueue` (`getSMQueue`)
- `fsrelocate` (`relocateFiles`)
- `fsretrieve` (`doRetrieve`)
- `fsrmdiskcopy` (`doTruncate`)
- `fsstate` (`getDriveReport`)
- `fsstore` (`doStore`)
- `fswascfg` (`getWasConfiguration`)
- `vsmove` (`doMediaMove`)

The new `fsxsd` and `vsxsd` commands provide access to XML schemas describing the output of those commands.

## Using the APIs

Accessing commands through WS-API can be done through a Web browser, by constructing an appropriate URL, or programmatically through a machine-generated library that supports the SOAP HTTP Client Protocol. Examples of these are provided in the notes listed below and in an example client application written in C# that is included in the StorNext installation package.

The available WS-API commands and the structure of their arguments and parameters are defined in the Web Service Definition Language file (WSDL) for StorNext, which is available from the StorNext Metadata Controller (MDC). The StorNext WSDL file can be displayed by entering the following URL in the address window of a web browser:

```
http://<MDC>:81/axis2/services/stornext?wsdl
```

Microsoft Visual Studio provides a menu selection for adding a "Service Reference" to the WSDL, which gleans the interfaces from the WSDL to aid programmers. Visual Studio also provides a `wsdl.exe` command that can generate source code in supported programming languages for compiling into a dynamic-link library (DLL) that supports the SOAP HTTP Client Protocol. Instructions for doing this are included in the example client code provided with StorNext. The sample WS-API client-application code, written in C# for Visual Studio on Windows, is included in StorNext distributions for MDCs. The following pathname displays a typical location for the file:

```
/tmp/stornext/stornext_full/RedHat50AS_26x86_64/examples/SNAPITest.zip
```

In order to perform any command on the remote server (except for the `getSNAPIVersion` call), a matching password must be specified with the call. The server verifies this password against the one stored in the `/usr/adic/.snapipassword` file on the server.

Make sure this file is the same on all metadata controllers (MDCs) you will be accessing. This is especially important if you are using virtual IP (vIP) addresses, because you can be communicating to different machines across different API calls or even retries of the same call.

The `.snapipassword` file is stored as clear text, and it should be set to be readable only by root. If this file does not exist, no calls other than `getSNAPIVersion` will succeed.

## Using APIs With the High Availability MDC Feature

The StorNext High Availability MDC Feature (HA) uses a pair of redundant MDCs to maximize the availability of StorNext file systems. Under normal operation, one of the MDCs is the Primary server for all Storage Manager processing, and the Secondary server is ready to take over operations if the Primary server stops. This is called an HA Failover.

The Virtual IP Address (vIP) is a feature of StorNext HA which allows using a single IP address to automatically connect with whichever MDC is currently Primary. Using the vIP for WS-API calls simplifies development of client code. However, a client application must handle some transitional behaviors that occur during an HA Failover.

During an HA Failover, there is typically a delay of one to two minutes between the stop of the original Primary MDC and the transition of the Secondary MDC to full Primary MDC status. The former Primary MDC can stop abruptly without closing TCP/IP connections. For a period of time after that, attempts to make new connections to the vIP will fail until the new Primary brings up the vIP interface. Next is a period of time when the MySQL database has not fully started. Finally, there is a period when TSM has not fully started. Each of these periods presents a different type of error to the client application.

When using WS-API with a StorNext High Availability (HA) cluster, the following notes apply:

1. Configure a Virtual IP Address (vIP). The vIP is a static IP address that is automatically configured as a virtual interface on the Primary MDC. The vIP allows a client application to use a single IP address to locate the WS-API server.
2. Set a timeout that allows a client application to recover if it was connected to a server that stopped abruptly. (Since the client software is responsible for setting a timeout, the actual method of specifying a timeout value on each Web service request varies. Check with your client documentation for specific instructions.)

Stores and retrieves to tape may take longer than the timeout, but they will continue to run to completion if they can. Store and retrieve requests can also be submitted multiple times without impacting in-process transfers. When the transfer completes, all the identical requests will complete and return status if there is still a connection to the requestor.

3. When a request fails because the vIP server is not available, MySQL is not running, or TSM is not fully started, wait a few moments and retry the request.

The WS-API Server does not time-out until the target command completes with either success or failure. The Client application's Microsoft .NET `SoapHttpClientProtocol` class has a `Timeout` property with a default of 100 seconds. This is more than enough for status-type requests, but it is too short for tape-transfer-type requests, which are dependent on file size, system configuration and contention for resources. For example,

a 100 GB file transferring at 100 MB/s will take about twenty minutes to complete. Setting the SoapHttpClientProtocol Timeout property to -1 is equivalent to infinity.

## WS-API APIs

This section provides descriptions for the APIs included with WS-API.

### Prerequisites

- A password is required to run the WS-API web services. The password is stored in the **/usr/adic.snapipassword** file on the server, and the file must contain only one word, for instance, the password without a prefix or suffix. For example:

```
[root@mdc01 ~]# cat /usr/adic/.snapipassword  
password123
```

### The doCancel API

Given a requestID (which can be retrieved by running [The getSMQueue API on page 283](#)), running the doCancel API aborts an operation in progress. Running this API is equivalent to running the fscancel command.

#### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=password123
requestID	Required	1	The request identifier of the request to be cancelled. Operations in progress can be retrieved by running the getSMQueue API. Example: requestID=12345
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

#### Example



```
http://host:81/axis2/services/stornext/doCancel  
?password=stornext  
&requestID=12345  
&format=json
```

## The doMediaMove API

Use the doMediaMove API to move media from one archive to another. Running this API is equivalent to running the vsmove command.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
mediaIDs	Required	N	Specifies one or more media to be moved. A valid medium identifier may contain up to 16 alphanumeric characters, including spaces. Leading and trailing spaces are not permitted. The number of media that can be specified is restricted by the CLI software. Currently, the maximum allowed number is 64.  Example: mediaIDs=1234
archiveName	Required	1	Specifies the name of the archive to which the specified media are to be moved. Valid archive names may contain up to 16 alphanumeric characters, including spaces. Leading and trailing spaces are not permitted.  Example: archiveName=lto
format	Optional	1	Output format: json, text, or xml. The default output format is text.  Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

### Example

```
http://host:81/axis2/services/stornext/doMediaMove  
?password=stornext  
&mediaIDs=1234  
&mediaIDs=5678  
&archiveName=lto  
&format=json
```

## The doRetrieve API

Use the doRetrieve API to retrieve or recover a file from media and place it on disk. By default, the primary copy of a file is retrieved. Running this API is equivalent to running the fsretrieve command.

### Parameters

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
files	Optional	N	The path and file name of the file(s) to retrieve. The full path name starting from the root directory is required as input to the command. Each file path must specify a file in a migration directory. Example: files=/stornext/dir/a
updateATime	Optional	1	Updates the access time of the requested files. Example: updateATime=1
copy	Optional	1	Used to retrieve a specific copy of filename if one exists. A number from 1 to N. Example: copy=2
newFileName	Optional	1	The new path and file name into which to retrieve the file. The location specified for the new file must be a local file system. Retrieval to an NFS-mounted file system is not permitted. Example: newFileName=/dir/new/filename
startByte	Optional	1	The startByte must be less than endByte, and both must be within the byte range of the file. The byte range is inclusive. To retrieve a single byte, the startByte is equal to the endByte. If the startByte and endByte are specified, the newFileName must be specified. Otherwise, the command is rejected. The byte range is zero relative; therefore a specified byte range must be zero to the end byte minus 1. Example: startByte=123
endByte	Optional	1	See the description for startByte Example: endByte=456

Parameter	Req / Opt	Num	Description
directory	Optional	1	The directory from which to start the recursive retrieve. All files from the specified directory and any subdirectories will be retrieved. Depending upon the number of files in the directory and subdirectories, running this option may use extensive Tertiary Manager resources.  Example: directory=/stornext/managed/pooldir/mp3
topPriority	Optional	1	Specifies top priority and will cause all files for the retrieve request to be placed at the top of the retrieve queue.  Example: topPriority=1
format	Optional	1	Output format: json, text, or xml. The default output format is text.  Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

### Exceptions

Parameter	Exception	Description
copy	startByte	If you specify a copy, you can NOT specify a startByte.
directory	copy	If you specify a directory, you can NOT specify a copy.
newFileName	files	If you specify a newFileName, you can NOT specify more than one (1) files.
newFileName	directory	If you specify a newFileName, you can NOT specify a directory.
startByte	newFileName	If you specify a startByte, you MUST specify a newFileName.

### Examples

Restore a file and give it a new file name.

```
http://host:81/axis2/services/stornext/doRetrieve
?password=stornext
&files=/stornext/managed/a
&newFileName=/stornext/managed/b
&format=json
```

Restore the second (2nd) copy of a file and give it a new file name.

```
http://host:81/axis2/services/stornext/doRetrieve
?password=stornext
&files=/stornext/managed/a
&copy=2
&newFileName=/stornext/managed/a2
&format=xml
```

Restore a portion of a file and give it a new file name.

```
http://host:81/axis2/services/stornext/doRetrieve
?password=stornext
&files=/stornext/managed/a
&newFileName=/stornext/managed/a-123-456
&startByte=123
&endByte=456
&format=xml
```

## The doStore API

Use the doStore API to store files as specified by its respective policy. Running this API is equivalent to running the fsstore command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
files	Required	N	The names of the file(s) on disk to store to media. The full path name starting from the root directory is required as input to the command. Multiple filenames must be separated by commas. Example: files=/stornext/dir/a
mediaType	Optional	1	Defines the type of medium to be used for storage. Depending on the type of platform used, the following media types are supported by Tertiary Manager software: AIT, AITW, LTO, LTOW, SDISK, 3590, 3592, 9840, 9940, T10K, DLT4, DLT2 (SDLT600 media) Example: mediaType=LTOW

Parameter	Req / Opt	Num	Description
<code>copies</code>	Optional	1	<p>Number of copies of the file(s) to be stored. The value is the total number of copies, including the primary copy of the file. This number cannot exceed the number of copies defined in the policy class <code>maxcopies</code> parameter. To store more copies of the file than are specified in the default <code>copies</code> parameter in the policy class, the file's copy attribute must be modified. Use <code>fschfiat filename -c</code> to change this attribute. Then the <code>-c</code> option can be used with the <code>doStore</code> API to store additional copies of the file. If the number of copies stored is less than the number specified by the policy class definition or by the <code>fschfiat</code> command, the remaining copies are stored when the storage policy is applied.</p> <p>Example: <code>copies=2</code></p>
<code>retention</code>	Optional	1	<p>The file retention policy for the filename specified. The files can be truncated immediately (<code>i</code>) or at policy application time (<code>p</code>) once all file copies are stored on a medium. If the <code>retention</code> option is not used, the file retention policy will be specified by the policy class definition.</p> <p>Example: <code>retention=i</code></p>
<code>drivePool</code>	Optional	1	<p>Media Manager drive pool group used to store the file specified. The drive pool must be defined in Media Manager software. If the <code>drivePool</code> option is not used, the default drive pool group will be specified by the policy class definition. The special <code>"_"</code> character is permitted to identify the drive pool group.</p> <p>Example: <code>drivePool=pool13</code></p>
<code>minSize</code>	Optional	1	<p>Minimum file size in bytes to be stored. Files larger than or equal to the specified <code>minSize</code> will be stored. Files with a size less than specified <code>minSize</code> will not be stored.</p> <p>Example: <code>minSize=123456</code></p>
<code>runTime</code>	Optional	1	<p>Maximum allowable time in hours for the command to finish. This command normally runs until it completes. This option can be used to limit how long it should remain active. If the store has not completed in the specified amount of time, then any outstanding activity will be cancelled.</p> <p>Example: <code>runTime=2</code></p>

Parameter	Req / Opt	Num	Description
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

### Examples

```
http://host:81/axis2/services/stornext/doStore
?password=stornext
&files=/stornext/managed/a
&format=json
```

## The doTruncate API

The doTruncate API allows a user to remove the copy/copies of the specified file(s) from disk after the file is copied to media. The cleanup policy is the system administration method of routinely freeing disk space by removing files after being stored on media.

The doTruncate API is used by users to maintain a desired level of disk space by truncating individual files. Files specified for removal from disk with the doTruncate API command must have an exact copy on media.

Suggested uses for the **fsrmdiskcopy** command include the following:

- After viewing a migrated file. Viewing the file caused it to be retrieved to disk. If the file is not modified, the disk copy can be removed.
- After storing the file to medium with the **fsstore** command without using the option to immediately truncate the file from disk.
- Between the application of the storage and truncation policies by the system administrator.

Running this API is equivalent to running the **fsrmdiskcopy** command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
files	Required	N	The names of the file(s) on disk to store to media. The file path(s) must also be in a migration directory. The full path name starting from the root directory is required as input to the command. Multiple filenames must be separated by commas. Example: files=/stornext/dir/a

Parameter	Req / Opt	Num	Description
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

### Example

```
http://host:81/axis2/services/stornext/doTruncate
?password=stornext
&files=/stornext/managed/a
&format=json
```

## The getDriveReport API

The getDriveReport API is a user command that can be executed when Tertiary Manager software is active or nonactive. The getDriveReport API reports the state of the Tertiary Manager software and/or all storage subsystems and drive components configured in the Quantum storage subsystem.

Submitting the getDriveReport API with the componentAlias option generates a report for a single Quantum component, i.e. drive(s), drive identifier(s), and Media Manager system(s).

Running this API is equivalent to running the fsstate command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
componentAlias	Optional	1	The alias for storage subsystem and drives. The system administrator configures the possible values for component aliases during system configuration or by using the <b>fsconfig</b> command. Example: componentAlias=sdisk
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

### Examples

Status of the 'sdisk' component.

```
http://host:81/axis2/services/stornext/getDriveReport
?password=stornext
&componentAlias=sdisk
&format=xml
```

## The getFileLocation API

The `getFileLocation` API reports the current location(s) of files, whether on disk, on a particular medium, or not in the SNMS system. It also displays file attribute information.

Running this API is equivalent to running the `fsfileinfo` command.

### Important Information About the getFileLocation API

The `getFileLocation` API hangs when there are 150 or more calls being processed by the MDC. The queue is usually due to lengthy tape processing calls such as `doRetrieve`, `doStore`, or `fsmedcopy`. The process might take up to an hour until the retrieves complete.

The number of incoming call services is restricted by the Tomcat `maxThreads` parameter in the file, `/usr/adic/gui/config/server.xml`; there are 2 lines in the `server.xml` file where the `maxThreads` parameter exists.

Quantum recommends you increase the `maxThreads` value to cover the maximum expected number of lengthy commands, and then restart the Web Services. The default setting is 150.

For example, in the `server.xml` file, locate the 2 lines that contain the following parameter:

```
maxThreads="150"
```

Update the `maxThreads` value accordingly. For example:

```
maxThreads="600"
```

- Tomcat settings are found in `/usr/adic/tomcat/conf/server.xml`.
- The **maxThreads** value specifies “The maximum number of request processing threads to be created which therefore determines the maximum number of simultaneous requests that can be handled.”
- The default **maxThreads** setting for StorNext is **150**, which is conservative and can be increased if a higher number of simultaneous requests are expected or required.
- The value should be set to the maximum number of simultaneous requests that the customer wants to be able to service.



- A high number of threads utilizes more memory. If the selected value is too high for your system, then it might cause system resource constraints and should be reduced and/or the Tomcat memory increased.
- If you are seeing slow performance and/or frequent memory errors in the GUI and in the GUI logs, then increase the available memory for Tomcat to use.
- On MDCs with multiple cores, setting this value between **150** and **1000** should not cause a problem. If you detect performance or memory issues, then you should allocate more Tomcat memory by altering the **-Xmx** value in `/etc/init.d/stornext_web` from the default **512m**. To increase the Tomcat memory, edit `/etc/init.d/stornext_web` and change the **-Xmx** value to a larger value. For example, **-Xmx1024m** or **-Xmx2048m** for 1 GB and 2 GB, respectively:

```
export JAVA_OPTS="$JAVA_OPTS -Ddas.lto.worm=${daslto worm} -Xms128m -Xmx1024m
-Dcom.amazonaws.sdk.disableCertChecking=true"
```

- After making changes, run **service stornext\_web restart** to restart Tomcat. Repeat all steps on the second node, for an HA configuration.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
files	Required	N	The path name of at least one file is required. The full path name starting from the root directory is required. Example: files=/stornext/dir/a
checksum	Optional	1	If checksum was turned on for the file when stored and this option is specified, the checksum value generated for the file will be displayed. Example: checksum=1
showIds	Optional	1	Show whether the file has any objects stored to the Wide Area Storage. Example: showIds=1
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

## Examples

Basic file information.

```
http://host:81/axis2/services/stornext/getFileLocation
?password=stornext
&files=/stornext/managed/a
&format=xml
```

Information about two files, with checksum and object IDs.

```
http://host:81/axis2/services/stornext/getFileLocation
?password=stornext
&files=/stornext/managed/a
&files=/stornext/managed/b
&checksum=1
&showIds=1
&format=xml
```

## The getFileTapeLocation API

The `getFileTapeLocation` API reports the location information of the file's on-tape copies. The report will list all the segments belonging to the specified file copy.

Running this API is equivalent to running the `fsfiletapeloc` command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
file	Required	N	The full path name starting from the root directory is required. Example: file=/stornext/dir/a
copy	Optional	1	The copy id to generate report for. If not specified, the information for the primary copy will be reported. Example: copy=2
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

### Examples

Basic file information.

```
http://host:81/axis2/services/stornext/getFileTapeLocation  
?password=stornext  
&file=/stornext/managed/a  
&format=xml
```

## The getMediaInfo API

The `getMediaInfo` API produces a list of media. The organization of the media list is defined by the use of options. If no options are used, the `getMediaInfo` API generates a short report that lists the total quantity of media in each policy class, including the general scratch pool.

Running this API is equivalent to running the `fsmedlist` command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
scratchPoolOnly	Required	1	Used to report on the blank media in the general scratch pool. Example: scratchPoolOnly=1
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

### Examples

Media report, no scratch pool.

```
http://host:81/axis2/services/stornext/getMediaInfo  
?password=stornext  
&scratchPoolOnly=0  
&format=xml
```

Media report, general scratch pool.

```
http://host:81/axis2/services/stornext/getMediaInfo  
?password=stornext  
&scratchPoolOnly=1  
&format=xml
```

## The getMediaReport API

The getMediaReport API produces either a short report or a long report on the specified media, based on the options entered. One or more media identifiers must be entered.

Running this API is equivalent to running the `fsmediainfo` command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
mediaIDs	Required	N	One or more media identifiers on which to report. Example: mediaIDs=sdisk
longReport	Optional	1	Produce the long form of the report that contains the same information as the short form, plus a list of the file segments on the medium. The pathname that is shown is the name of the file at the time the file was stored. If the file has been renamed since that time that will not be reflected in this report. If the parent or name of an individual file cannot be found, the getMediaReport API will indicate that fact but still report the key. Example: longReport=1
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

### Examples

Media report, short form.

```
http://host:81/axis2/services/stornext/getMediaReport
?password=stornext
&mediaIDs=sdisk
&format=xml
```

Media report, long form.

```
http://host:81/axis2/services/stornext/getMediaReport
?password=stornext
&mediaIDs=sdisk
&longReport=1
&format=xml
```

## The getSMQueue API

The getSMQueue API checks the request queue for the specified request identifier(s), filename(s), or media. Requests awaiting resources (drives and media) are displayed.

Issuing the getSMQueue API without any options will report all resource requests associated with storage subsystems, for example, drive-media mount requests.

Running this API is equivalent to running the **fsqueue** command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
report	Optional	1	The report type. Valid names are: file, media, moverhost, or moverrequest. Example: report=file
requestID	Optional	1	The request identifier of the request to be reported. Valid when the report type is file or media. Example: requestId=1234
file	Optional	N	The file option reports the current status and request identifier associated with a specified file or files. This method gets the active request identifier associated with a store or retrieve if a filename is known. However, because of potential dependencies, the original request identifier for <a href="#">The doStore API on page 274</a> or <a href="#">The doRetrieve API on page 272</a> issued by a particular user may not be discernible. Valid when the report type is file. Example: file=/stornext/dir/a
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, x=1&x=2&x=3.

### Examples

Check request queue for a few file.

```
http://host:81/axis2/services/stornext/getSMQueue
?password=stornext
&report=file
&file=/stornext/managed/a
&format=xml
```

Check request queue for a specific request ID.

```
http://host:81/axis2/services/stornext/getSMQueue  
?password=stornext  
&report=media  
&requestId=1234  
&format=xml
```

## The getSNAPIVersion API

The `getSNAPIVersion` API returns the Web Service version and the output from the `cvversions` command.

Running this API is equivalent to running the `cvversions` command.

Parameter	Req / Opt	Num	Description
<code>in</code>	Required	1	The parameter <code>in</code> is required, but need not contain anything.

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

### Examples

Get the file system server, client, and host OS versions.

```
http://host:81/axis2/services/stornext/getSNAPIVersion  
?in=
```

## The getSystemStatus API

The `getSystemStatus` API returns the software version and the status of DSM, TSM, MSM, MySQL and SRVCLOG.

Running this API is equivalent to running the `cvversions`, `cvadmin`, `TSM_control`, `vsping`, `mysql_control`, and `SRVCLOG_control` commands.

Parameter	Req / Opt	Num	Description
<code>password</code>	Required	1	Example: <code>password=stornext</code>
<code>format</code>	Optional	1	Output format: <code>json</code> , <code>text</code> , or <code>xml</code> . The default output format is <code>text</code> . Example: <code>format=xml</code>

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

Get the file system server and client versions.

```
http://host:81/axis2/services/stornext/getSystemStatus  
?password=stornext  
&format=xml
```

## The getWasConfiguration API

The `getWasConfiguration` API reports configuration settings for Wide Area Storage components in the storage system. Wide Area Storage components are: **Appliances**, **Controllers**, **I/O Paths** and **Namespaces**. These components and their attributes, provide the addressing information required to form the URL to store and retrieve objects from the Wide Area Storage.

Running this API is equivalent to running the `fswascfg` command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext
format	Optional	1	Output format: json, text, or xml. The default output format is text. Example: format=xml

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

### Example

```
http://host:81/axis2/services/stornext/getWasConfiguration  
?password=stornext  
&format=xml
```

## The relocateFiles API

Use the `relocateFiles` API to relocate a managed file from one disk affinity to another, or change the affinity association of a truncated file.

Running this API is equivalent to running the `fsrelocate` command.

Parameter	Req / Opt	Num	Description
password	Required	1	Example: password=stornext

Parameter	Req / Opt	Num	Description
affinity	Required	1	The destination affinity. This affinity must be defined for the file system in which the file resides. The file will be relocated to this affinity if it has not been truncated. If the file has been truncated, only the file's affinity association will change. Example: <code>affinity=tier1</code>
file	Required	N	One or more files to be relocated. The files must reside in a managed directory. Example: <code>file=/stornext/dir/a</code>
format	Optional	1	Output format: <code>json</code> , <code>text</code> , or <code>xml</code> . The default output format is <code>text</code> . Example: <code>format=xml</code>

**Num** is the number of arguments: 1 = only one allowed, N = multiple allowed, the parameter name is repeated for multiple entries, for example, `x=1&x=2&x=3`.

### Examples

Relocate a single file.

```
http://host:81/axis2/services/stornext/relocateFiles
?password=stornext
&affinity=tier1
&file=/stornext/managed/a
&format=xml
```

Relocate multiple files.

```
http://host:81/axis2/services/stornext/relocateFiles
?password=stornext
&affinity=tier1
&file=/stornext/managed/a
&file=/stornext/managed/b
&format=xml
```

## Examples

This section contains examples for Web Services URLs, sample XML output, sample JSON output, and sample text output.



## Example: Web Services URLs

Following are example URLs which demonstrate how to use the output format parameter. The URLs are formatted as multiple lines to make them easier to read.

**i Note:** Two fields in the URL must be modified for local use: MDC\_address (or vIP) and Password.

1. Run the **fsfileinfo** command (**getFileLocation** API) for three files, and return output in structured XML format:

```
http://<MDC_address>:81/axis2/services/stornext/getFileLocation
?password=<Password>
&files=/stornext/dir/fileA
&files=/stornext/dir/fileB
&files=/stornext/dir/fileC
&format=xml
```

2. Run the **fsstore** command (**doStore** API) for one file, with the "-f i" option, and return output in structured XML format:

```
http://<MDC_address>:81/axis2/services/stornext/doStore
?password=<Password>
&files=/stornext/dir/fileA
&retention=i
&format=xml
```

3. Run the **fsrmdiskcopy** command (**doTruncate** API) for one file, and return output in JSON format:

```
http://<MDC_address>:81/axis2/services/stornext/doTruncate
?password=<Password>
&files=/stornext/fileA
&format=json
```

4. Run the **fsretrieve** command (**doRetrieve** API) for a directory, with the "-a" parameter, and return output in text format:

```
http://<MDC_address>:81/axis2/services/stornext/doRetrieve
?password=<Password>
&updateATime=1
&directory=/stornext/dir/
&format=text
```

## Sample XML Output

```
[tester1@smo4 p1]# fsfileinfo -F xml /stornext/dir/fileA
<?xml version="1.0" encoding="UTF-8"?>
<fsfileinfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="fsfileinfo.xsd">
  <header>
    <commandName>fsfileinfo</commandName>
    <commandLine>fsfileinfo -F xml /stornext/dir/fileA</commandLine>
    <commandDescription>Generate a report about files known to the Tertiary
      Manager</commandDescription>
    <localDateISO>2011-09-07T11:00:20</localDateISO>
    <localDate>2011-09-07</localDate>
    <localTime>11:00:20</localTime>
    <localDayOfWeek>3</localDayOfWeek>
    <gmtDateISO>2011-09-07T16:00:20Z</gmtDateISO>
    <gmtDate>2011-09-07</gmtDate>
    <gmtTime>16:00:20</gmtTime>
    <gmtDayOfWeek>3</gmtDayOfWeek>
  </header>
  <fileInfos>
    <fileInfo>
      <fileName>/stornext/dir/fileA</fileName>
      <storedPathFileName>/stornext/dir/fileA</storedPathFileName>
      <storedPathSameAsFileName>false</storedPathSameAsFileName>
      <lastModificationDateString>03-aug-2011
        15:49:36</lastModificationDateString>
      <lastModificationDate>2011-08-03</lastModificationDate>
      <lastModificationDayOfWeek>3</lastModificationDayOfWeek>
      <lastModificationTime>15:49:36</lastModificationTime>
      <owner>root</owner>
      <location>DISK AND TAPE</location>
      <group>root</group>
      <existingCopies>1</existingCopies>
      <access>644</access>
      <targetCopies>1</targetCopies>
    </fileInfo>
  </fileInfos>
</fsfileinfo>
```

```

<targetStubSize>0</targetStubSize>
  <targetStubScale>1024</targetStubScale>
  <existingStubSize>n/a</existingStubSize>
  <fileSize>1936636</fileSize>
  <store>MINTIME</store>
  <affinity>n/a</affinity>
  <reloc>MINTIME</reloc>
  <class>pool</class>
  <trunc>MINTIME</trunc>
  <cleanDBInfo>NO</cleanDBInfo>
  <medias>
    <media>
      <mediaId>sdisk</mediaId>
      <copy>1</copy>
    </media>
  </medias>
  <checksums>
    <checksum>
      <summary>N</summary>
    </checksum>
  </checksums>
</fileInfo>
</fileInfos>
<statuses>
  <status>
    <statusCode>FS0000</statusCode>
    <statusNumber>0</statusNumber>
    <dayOfMonth>7</dayOfMonth>
    <requestId>2125016624</requestId>
    <commandName>fsfileinfo</commandName>
    <commandStatus>completed</commandStatus>
    <statusText>Command Successful.</statusText>
  </status>
</statuses>
<footer>
  <returnCode>0</returnCode>
  <localDateISOEnd>2011-09-07T11:00:20</localDateISOEnd>
  <localDateEnd>2011-09-07</localDateEnd>
  <localTimeEnd>11:00:20</localTimeEnd>
  <localDayOfWeekEnd>3</localDayOfWeekEnd>
  <gmtDateISOEnd>2011-09-07T16:00:20Z</gmtDateISOEnd>
  <gmtDateEnd>2011-09-07</gmtDateEnd>
  <gmtTimeEnd>16:00:20</gmtTimeEnd>
  <gmtDayOfWeekEnd>3</gmtDayOfWeekEnd>
  <elapsedTimeInSeconds>0.0077</elapsedTimeInSeconds>
</footer>
</fsfileinfo>

```

## Sample JSON Output

```
[tester1@smo4 p1]# fsfileinfo -F json /stornext/dir/fileA
{
  "header": {
    "commandName": "fsfileinfo",
    "commandLine": "fsfileinfo -F json /stornext/dir/fileA",
    "commandDescription": "Generate a report about files known to the Tertiary
Manager",
    "localDateISO": "2011-09-07T11:07:36",
    "localDate": "2011-09-07",
    "localTime": "11:07:36",
    "localDayOfWeek": 3,
    "gmtDateISO": "2011-09-07T16:07:36Z",
    "gmtDate": "2011-09-07",
    "gmtTime": "16:07:36",
    "gmtDayOfWeek": 3
  },
  "fileInfos": [
    {
      "fileName": "/stornext/dir/fileA",
      "storedPathFileName": "/stornext/dir/fileA",
      "storedPathSameAsFileName": false,
      "lastModificationDateString": "03-aug-2011 15:49:36",
      "lastModificationDate": "2011-08-03",
      "lastModificationDayOfWeek": 3,
      "lastModificationTime": "15:49:36",
      "owner": "root",
      "location": "DISK AND TAPE",
      "group": "root",
      "existingCopies": 1,
      "access": 644,
      "targetCopies": 1,
      "targetStubSize": 0,
      "targetStubScale": 1024,
      "existingStubSize": "n/a",
      "fileSize": 1936636,
      "store": "MINTIME",
      "affinity": "n/a",
      "reloc": "MINTIME",
      "class": "pool",
      "trunc": "MINTIME",
      "cleanDBInfo": "NO",
      "medias": [
        { "mediaId": "sdisk", "copy": 1 }
      ],
      "checksums": [
        { "summary": "N" }
      ]
    }
  ]
}
```

```
],
  "statuses": [
    {
      "statusCode": "FS0000",
      "statusNumber": 0,
      "dayOfMonth": 7,
      "requestId": 2125016625,
      "commandName": "fsfileinfo",
      "commandStatus": "completed",
      "statusText": "Command Successful."
    }
  ],
  "footer": {
    "returnCode": 0,
    "localDateISOEnd": "2011-09-07T11:07:36",
    "localDateEnd": "2011-09-07",
    "localTimeEnd": "11:07:36",
    "localDayOfWeekEnd": 3,
    "gmtDateISOEnd": "2011-09-07T16:07:36Z",
    "gmtDateEnd": "2011-09-07",
    "gmtTimeEnd": "16:07:36",
    "gmtDayOfWeekEnd": 3,
    "elapsedTimeInSeconds": "0.0011"
  }
}
```

## Sample Text Output

```
[tester1@smo4 p1]# fsfileinfo -F text /stornext/dir/fileA
-----
File Information Report                               Wed Sep  7 11:10:35 2011
Filename:      /stornext/dir/fileA
Stored path:   /stornext/dir/fileA
-----

Last Modification: 03-aug-2011 15:49:36
Owner:            root                               Location:       DISK AND TAPE
Group:           root                               Existing Copies: 1
Access:          644                               Target Copies:  1
Target Stub:     0 (KB)                            Existing Stub:  n/a
File size:       1,936,636                          Store:         MINTIME
Affinity:        n/a                               Reloc:         MINTIME
Class:           pool                               Trunc:         MINTIME
Clean DB Info:   NO

Media:   sdisk(1)
Checksum: N
FS0000 07 2125016626 fsfileinfo completed: Command Successful.
```



## Appendix A: Sample Perl Script

---

## Appendix A: Sample Perl Script

```
001: #!/usr/bin/env perl
002:
003: use LWP::UserAgent;
004: use LWP::Simple;
005: use File::Find;
006: use Getopt::Long;
007:
008: use strict;
009: use warnings;
010:
011: our $VERBOSE          = 0;
012: our $IP               = '';
013: our $FORMAT           = '';
014: our $PROTOCOL         = '';
015: our $AUTHENTICATE     = '';
016: our $FAILURES         = 0;
017: our $USERNAME         = '';
018: our $PASSWORD         = '';
019:
020: our $ua               = LWP::UserAgent->new;
021: our $rsp;
022:
023: GetOptions ("authenticate=s" => \$AUTHENTICATE,
024:           "protocol=s"      => \$PROTOCOL,
025:           "ip=s"            => \$IP,
026:           "format=s"        => \$FORMAT,
027:           "verbose"         => \$VERBOSE)
028: or die("Error in command line arguments\n");
029:
030: # check protocol, should be either http or https
031: #
032: unless (($PROTOCOL eq 'http') or ($PROTOCOL eq 'https')) {
033:     print "Protocol should be either 'http' or 'https'\n";
034:     exit;
035: }
```

## Appendix A: Sample Perl Script

```
036:
037:
038: # check format, should be json, text, or xml
039: #
040: unless (($FORMAT eq 'json') or ($FORMAT eq 'text') or ($FORMAT eq 'xml')) {
041:     print "Format should be either 'json', 'text', or 'xml'\n";
042:     exit;
043: }
044:
045:
046: # check the ip, standard 4 numbers '.' separated
047: #
048: if ($IP !~ /^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$/) {
049:     print "ip not in correct format: $IP\n";
050:     exit;
051: }
052:
053: # check authentication
054: #
055: if ($AUTHENTICATE eq '') {
056:     ;
057: } elsif ($AUTHENTICATE =~ /\//) {
058:     my ($username, $password) = split( /\//, $AUTHENTICATE);
059:     $USERNAME = $username . '_pl';
060:     $PASSWORD = $password . '_pl';
061: } else {
062:     print 'Authenticate argument must have '/' between the username and password:\n';
063:     print "value: $AUTHENTICATE\n";
064:     exit;
065: }
066:
067: # $prefix = "#{protocol}://sws:sws@#{$ip}:81/sws/v2"
068:
069: print "VERBOSE:      $VERBOSE\n";
070: print "IP:           $IP\n";
```



## Appendix A: Sample Perl Script

```
071: print "FORMAT:          $FORMAT\n";
072: print "PROTOCOL:        $PROTOCOL\n";
073: print "AUTHENTICATE:     $AUTHENTICATE\n";
074:
075:
076: sub check_job_status {
077:     my ($rsp,@more) = @_ ;
078:     print "$rsp\n";
079:     if ($rsp =~ /Job has been successfully submitted/i) {
080:         $rsp =~ /"job" : "(\\d+)"/i;
081:         my $jobno = $1;
082:         my $myrsp;
083:         my $myrsp2;
084:         $myrsp2 = do_webservices_cmd( '/job/mover/info',
085:                                     "job=$jobno");
086:         print "$myrsp2\n";
087:         $myrsp = do_webservices_cmd( '/job/info',
088:                                     "job=$jobno");
089:         my $count = 0;
090:         while ($myrsp =~ /RUNNING/i) {
091:             sleep(1);
092:             $myrsp2 = do_webservices_cmd( '/job/mover/info',
093:                                         "job=$jobno");
094:             print "$myrsp2\n";
095:             $myrsp = do_webservices_cmd( '/job/info',
096:                                         "job=$jobno");
097:             print "$myrsp\n";
098:             $count += 1;
099:         }
100:         $myrsp2 = do_webservices_cmd( '/job/mover/info',
101:                                     "job=$jobno");
102:         print "$myrsp2\n";
103:         print "Checked for completion: $count times\n";
104:         print "$myrsp\n";
105:         if ($myrsp =~ /ERROR/i) {
```

## Appendix A: Sample Perl Script

```
106:         print "Job had error -----\n";
107:         $FAILURES += 1;
108:     }
109: } else {
110:     print "Job not submitted correctly -----\n";
111:     $FAILURES += 1;
112: }
113: }
114:
115:
116: sub do_webservices_cmd {
117:     my ($cmd, @pieces) = @_ ;
118:     #? print "cmd: $cmd\n"; #?
119:     #? print "pieces: @pieces\n"; #?
120:     #? for my $piece (@pieces) { #?
121:         #?     print "piece: $piece\n"; #?
122:     #? } #?
123:
124:     # set up the URL prefix
125:     #
126:     my $prefix;
127:     if ($cmd =~ /wsconfig/i) {
128:         $prefix = "http://$IP:81/sws/v2";
129:     } elsif ($PROTOCOL =~ /https/i) {
130:         $prefix = "https://$IP/sws/v2";
131:     } else {
132:         $prefix = "http://$IP:81/sws/v2";
133:     }
134:
135:
136:     my $ws = $prefix . $cmd;
137:
138:     if ($AUTHENTICATE ne '') {
139:         push( @pieces, "username=$USERNAME");
140:         push( @pieces, "password=$PASSWORD");
```

## Appendix A: Sample Perl Script

```
141:     }
142:
143:     # add any passed parameters to URL
144:     #
145:     if (scalar(@pieces) > 0) {
146:         $ws .= '?' . join( '&', sort(@pieces));
147:     }
148:
149:     # if we do not already have a format parameter
150:     #
151:     unless ($ws =~ /format=/i) {
152:         # add '?' if this is the first parameter, otherwise add '&'
153:         #
154:         if ($ws =~ /\?/) {
155:             $ws .= '&';
156:         } else {
157:             $ws .= '?';
158:         }
159:         $ws .= "format=$FORMAT";    # ask for appropriate formatting for the web-service
160:     }
161:
162:     # add login parameters if we need to authenticate
163:     #
164:     # all web-service URLs are lowercase
165:     #
166:     $ws = lc( $ws);
167:
168:     if ($VERBOSE) {
169:         print "\n";
170:         print "WS: $ws\n";
171:         print "===== \n" if ($ws =~ /mover/);
172:     }
173:
174:     my $req = new HTTP::Request GET => $ws;
175:     my $res = $ua->request($req);
```

## Appendix A: Sample Perl Script

```
176:
177:     if ( $res->is_success ) {
178:         my $rsp = $res->content;
179:         return $rsp;
180:     }
181:     else {
182:         print "**** HTTP ERROR: " . $res->status_line . "...\\n";
183:         $FAILURES += 1;
184:         return 1;
185:     }
186: }
187:
188: my $ws_rsp;
189:
190: #Description of Web Services sample programs:
191:
192: # 0) Please turn on web-services using the StorNext GUI
193: #     Also choose protocol and authentication through the SN GUI
194:
195: # 1) Do WS system info, returning TEXT, XML and JSON
196:
197: # /sws/v2/system/info?format=text
198: $ws_rsp = do_webservices_cmd( '/system/info',
199:                               "format=text");
200: print "$ws_rsp\\n";
201:
202: # /sws/v2/system/info?format=xml
203: $ws_rsp = do_webservices_cmd( '/system/info',
204:                               "format=xml");
205: print "$ws_rsp\\n";
206:
207: # /sws/v2/system/info?format=json
208: $ws_rsp = do_webservices_cmd( '/system/info',
209:                               "format=json");
210: print "$ws_rsp\\n";
```

## Appendix A: Sample Perl Script

```
211:
212: # 2) Create a policy for a managed file system
213:
214: # 3) 3 directories should exist:
215: # a) Directory for single-file manipulation
216:
217: my $singles_path = '/stornext/snfs1/sample_dir_singles/pl';
218: my @singles_paths_sync = ();
219: my @singles_paths_async = ();
220:
221: # b) Directory for directory manipulation
222:
223: my $dirs_path_sync = '/stornext/snfs1/sample_dir_dirs_sync/pl';
224: my @dirs_paths_sync = ();
225: my $dirs_path_async = '/stornext/snfs1/sample_dir_dirs_async/pl';
226: my @dirs_paths_async = ();
227:
228: # c) Directory for multi-file manipulation
229:
230: my $multi_path = '/stornext/snfs1/sample_dir_multi/pl';
231: my @multi_paths_sync = ();
232: my @multi_paths_async = ();
233:
234: # 4) Create arrays with path names of both files in each of those directories
235:
236: for my $i (0,1) {
237:     my $filename = "file.$i";
238:
239:     push @singles_paths_sync, "$singles_path/$filename";
240:
241:     push @dirs_paths_sync, "$dirs_path_sync/$filename";
242:
243:     push @dirs_paths_async, "$dirs_path_async/$filename";
244:
245:     push @multi_paths_sync, "$multi_path/$filename";
```

## Appendix A: Sample Perl Script

```
246: }
247:
248: for my $i (2,3) {
249:     my $filename = "file.$i";
250:
251:     push @singles_paths_async, "$singles_path/$filename";
252:
253:     push @multi_paths_async, "$multi_path/$filename";
254: }
255:
256:
257: # 5) Use WS fsstore to save both files from the first directory to TAPE
258:
259: # /sws/v2/file/fsstore?file=<filepath>
260: for my $filepath (@singles_paths_sync) {
261:     $ws_rsp = do_webservices_cmd( '/file/fsstore',
262:                                   "file=$filepath");
263:     print "$ws_rsp\n";
264: }
265:
266: # /sws/v2/file/fsfileinfo?file=<filepath>
267: for my $filepath (@singles_paths_sync) {
268:     $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
269:                                   "file=$filepath");
270:     print "$ws_rsp\n";
271: }
272:
273: # 6) Use WS fsstore to save the second directory to TAPE
274:
275: # /sws/v2/file/fsstore?directory=<dirpath>
276: $ws_rsp = do_webservices_cmd( '/file/fsstore',
277:                               "directory=$dirs_path_sync");
278: print "$ws_rsp\n";
279:
280: # /sws/v2/file/fsfileinfo?directory=<dirpath>
```

## Appendix A: Sample Perl Script

```
281: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
282:                               "directory=$dirs_path_sync");
283: print "$ws_rsp\n";
284:
285: # 7) Use WS fsstore to save both files from the third directory to TAPE
286:
287: # /sws/v2/file/fsstore?file=<f1>&file=<f2>
288: $ws_rsp = do_webservices_cmd( '/file/fsstore',
289:                               "file=$multi_paths_sync[0]",
290:                               "file=$multi_paths_sync[1]");
291: print "$ws_rsp\n";
292:
293:
294: # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
295: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
296:                               "file=$multi_paths_sync[0]",
297:                               "file=$multi_paths_sync[1]");
298: print "$ws_rsp\n";
299:
300:
301: # 8) Use WS rmdiskcopy to truncate both files in each of 3 directories
302:
303: # /sws/v2/file/fsrmdiskcopy
304: for my $filepath (@singles_paths_sync) {
305:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
306:                                   "file=$filepath");
307:     print "$ws_rsp\n";
308: }
309:
310: for my $filepath (@dirs_paths_sync) {
311:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
312:                                   "file=$filepath");
313:     print "$ws_rsp\n";
314: }
315:
```

## Appendix A: Sample Perl Script

```
316: for my $filepath (@multi_paths_sync) {
317:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
318:                                   "file=$filepath");
319:     print "$ws_rsp\n";
320: }
321:
322: # 9) Use WS fsretrieve to restore both files to first directory from TAPE
323:
324: # /sws/v2/file/fsretrieve?file=<filepath>
325: for my $filepath (@singles_paths_sync) {
326:     $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
327:                                   "file=$filepath");
328:     print "$ws_rsp\n";
329: }
330:
331: # /sws/v2/file/fsfileinfo?file=<filepath>
332: for my $filepath (@singles_paths_sync) {
333:     $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
334:                                   "file=$filepath");
335:     print "$ws_rsp\n";
336: }
337:
338: # 10) Use WS fsretrieve to restore the second directory from TAPE
339:
340: # /sws/v2/file/fsretrieve?directory=<dirpath>
341: $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
342:                               "directory=$dirs_path_sync");
343: print "$ws_rsp\n";
344:
345: # /sws/v2/file/fsfileinfo?directory=<dirpath>
346: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
347:                               "directory=$dirs_path_sync");
348: print "$ws_rsp\n";
349:
350: # 11) Use WS fsretrieve to restore both files in the third directory from TAPE
```



## Appendix A: Sample Perl Script

```
351:
352: # /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
353: $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
354:                               "file=$multi_paths_sync[0]",
355:                               "file=$multi_paths_sync[1]");
356: print "$ws_rsp\n";
357:
358:
359: # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
360: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
361:                               "file=$multi_paths_sync[0]",
362:                               "file=$multi_paths_sync[1]");
363: print "$ws_rsp\n";
364:
365:
366: # 12-18) Repeat steps 5-11 using async mode
367:
368: # 12) Use WS fsstore to save both files from the first directory to TAPE
369:
370: # /sws/v2/file/fsstore?file=<filepath>
371: for my $filepath (@singles_paths_async) {
372:     $ws_rsp = do_webservices_cmd( '/file/fsstore',
373:                                   "file=$filepath",
374:                                   "mode=async");
375:     check_job_status( $ws_rsp);
376: }
377:
378: # /sws/v2/file/fsfileinfo?file=<filepath>
379: for my $filepath (@singles_paths_async) {
380:     $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
381:                                   "file=$filepath");
382:     print "$ws_rsp\n";
383: }
384:
385: # 13) Use WS fsstore to save the second directory to TAPE
```

## Appendix A: Sample Perl Script

```
386:
387: # /sws/v2/file/fsstore?directory=<dirpath>
388: $ws_rsp = do_webservices_cmd( '/file/fsstore',
389:                               "directory=$dirs_path_async",
390:                               "mode=async");
391: check_job_status( $ws_rsp);
392:
393: # /sws/v2/file/fsfileinfo?directory=<dirpath>
394: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
395:                               "directory=$dirs_path_async");
396: print "$ws_rsp\n";
397:
398: # 14) Use WS fsstore to save both files from the third directory to TAPE
399:
400: # /sws/v2/file/fsstore?file=<f1>&file=<f2>
401: $ws_rsp = do_webservices_cmd( '/file/fsstore',
402:                               "file=$multi_paths_async[0]",
403:                               "file=$multi_paths_async[1]",
404:                               "mode=async");
405: check_job_status( $ws_rsp);
406:
407:
408: # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
409: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
410:                               "file=$multi_paths_async[0]",
411:                               "file=$multi_paths_async[1]");
412: print "$ws_rsp\n";
413:
414:
415: # 15) Use WS rmdiskcopy to truncate both files in each of 3 directories
416:
417: # /sws/v2/file/fsrmdiskcopy
418: for my $filepath (@singles_paths_async) {
419:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
420:                                   "file=$filepath");
```

## Appendix A: Sample Perl Script

```
421:     print "$ws_rsp\n";
422: }
423:
424: for my $filepath (@dirs_paths_async) {
425:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
426:                                   "file=$filepath");
427:     print "$ws_rsp\n";
428: }
429:
430: for my $filepath (@multi_paths_async) {
431:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
432:                                   "file=$filepath");
433:     print "$ws_rsp\n";
434: }
435:
436: # 16) Use WS fsretrieve to restore both files to first directory from TAPE
437:
438: # /sws/v2/file/fsretrieve?file=<filepath>
439: for my $filepath (@singles_paths_async) {
440:     $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
441:                                   "file=$filepath",
442:                                   "mode=async");
443:     check_job_status( $ws_rsp);
444: }
445:
446: # /sws/v2/file/fsfileinfo?file=<filepath>
447: for my $filepath (@singles_paths_async) {
448:     $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
449:                                   "file=$filepath");
450:     print "$ws_rsp\n";
451: }
452:
453: # 17) Use WS fsretrieve to restore the second directory from TAPE
454:
455: # /sws/v2/file/fsretrieve?directory=<dirpath>
```

## Appendix A: Sample Perl Script

```
456: $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
457:                               "directory=$dirs_path_async",
458:                               "mode=async");
459: check_job_status( $ws_rsp);
460:
461: # /sws/v2/file/fsfileinfo?directory=<dirpath>
462: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
463:                               "directory=$dirs_path_async");
464: print "$ws_rsp\n";
465:
466: # 18) Use WS fsretrieve to restore both files in the third directory from TAPE
467:
468: # /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
469: $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
470:                               "file=$multi_paths_async[0]",
471:                               "file=$multi_paths_async[1]",
472:                               "mode=async");
473: check_job_status( $ws_rsp);
474:
475:
476: # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
477: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
478:                               "file=$multi_paths_async[0]",
479:                               "file=$multi_paths_async[1]");
480: print "$ws_rsp\n";
481:
482:
483: # 19) Get policy information
484:
485: $ws_rsp = do_webservices_cmd( '/policy/fsdirclass',
486:                               "directory=$dirs_path_async");
487: print "$ws_rsp\n";
488:
489: $ws_rsp = do_webservices_cmd( '/policy/fsclassinfo',
490:                               "policy=policy_min_i");
```

## Appendix A: Sample Perl Script

```
491: print "$ws_rsp\n";
492:
493:
494: print "\n";
495: print "Number of failures: $FAILURES\n";
496:
497:
498: #           Copyright 2015 Quantum Corporation
499: # vim:ts=4:sw=4:et:ic:bg=dark:
```



## Appendix B: Sample Python Script

---

## Appendix B: SamplePython Script

```
001: #! /opt/quantum/python27/bin/python
002:
003: import requests
004: import argparse
005: import re
006: import sys
007: import urllib
008: from time import sleep
009:
010: global failures
011: failures = 0
012: global args
013: global username
014: global password
015: global verbose
016:
017: def check_job_status( ws_rsp):
018:     global args
019:     global failures
020:     global verbose
021:
022:     print "%s" % (ws_rsp,)
023:     m = re.search( '"message" : "Job has been successfully submitted."', ws_rsp)
024:     if m:
025:         m = re.search( '"job" : "(\\d+)"', ws_rsp)
026:         if m:
027:             jobno = m.group(1)
028:         else:
029:             print "job not found -----"
030:             return
031:         ws_rsp2 = do_webservices_cmd( '/job/mover/info',
032:                                     { 'job' : jobno } )
033:         print ws_rsp2
034:         ws_rsp = do_webservices_cmd( '/job/info',
035:                                     { 'job' : jobno } )
```

## Appendix B: SamplePython Script

```
036:         count = 0
037:         while re.search( 'RUNNING', ws_rsp):
038:             sleep(1)
039:             ws_rsp2 = do_webservices_cmd( '/job/mover/info',
040:                                         { 'job' : jobno } )
041:             print ws_rsp2
042:             ws_rsp = do_webservices_cmd( '/job/info',
043:                                         { 'job' : jobno } )
044:             print ws_rsp
045:             count += 1
046:             ws_rsp2 = do_webservices_cmd( '/job/mover/info',
047:                                         { 'job' : jobno } )
048:             print ws_rsp2
049:             print "Checked for completion: %d times" % (count,)
050:             print ws_rsp
051:             if re.search( 'Status: ERROR', ws_rsp):
052:                 print "Job had error -----"
053:                 failures += 1
054:         else:
055:             print "Job not submitted correctly -----"
056:             failures += 1
057:
058:
059:
060: def process_args():
061:     global args
062:     global username
063:     global password
064:     global verbose
065:
066:     parser = argparse.ArgumentParser()
067:     parser.add_argument("--verbose",
068:                         action='store_true',
069:                         dest='verbose',
070:                         help=" Be verbose in output.")
```



```

071:     parser.add_argument("--authenticate",
072:                          action='store',
073:                          dest='authenticate',
074:                          metavar='[authentication]',
075:                          default='',
076:                          help="Authenticate all WS calls, value is 'username/password'.")
077:     parser.add_argument("--protocol",
078:                          action='store',
079:                          dest='protocol',
080:                          metavar='[PROTOCOL]',
081:                          choices=['http', 'https'],
082:                          help="The protocol to use.")
083:     parser.add_argument("--format",
084:                          action='store',
085:                          dest='format',
086:                          choices=['text', 'json', 'xml'],
087:                          default='json',
088:                          help="The format for response.")
089:     parser.add_argument("--ip",
090:                          action='store',
091:                          dest='ip',
092:                          default='',
093:                          help="The ip of the MDC.")
094:
095:     # Parse the command-line
096:     args = parser.parse_args()
097:
098:     print "authenticate: %s" % (args.authenticate,)
099:     print "format:      %s" % (args.format,)
100:     print "ip:         %s" % (args.ip,)
101:     print "protocol:   %s" % (args.protocol,)
102:     print "verbose:    %s" % (args.verbose,)
103:     verbose = args.verbose
104:
105:     # check the ip, standard 4 numbers '.' separated

```

## Appendix B: SamplePython Script

```
106:     #
107:     m = re.compile( '^\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}$')
108:     if not m.match( args.ip):
109:         print "ip not in correct format: %s" % (args.ip,)
110:         sys.exit(1)
111:
112:     # check authentication
113:     #
114:     if args.authenticate == '':
115:         username = ''
116:         password = ''
117:     elif '/' in args.authenticate:
118:         username, password = args.authenticate.split( '/')
119:         username = username + '_py'
120:         password = password + '_py'
121:     else:
122:         print 'Authenticate argument must have '/' between the username and password:'
123:         print "value: %s" % (args.authenticate,)
124:         sys.exit(1)
125:
126:
127: def do_webservices_cmd( cmd, pieces={}):
128:     global args
129:     global failures
130:     global username
131:     global password
132:     global verbose
133:
134:     # set up the URL prefix
135:     #
136:     if 'wsconfig' in cmd:
137:         prefix = "http://%s:81/sws/v2" % (args.ip,)
138:     elif 'https' in args.protocol:
139:         prefix = "https://%s:81/sws/v2" % (args.ip,)
140:     else:
```

## Appendix B: SamplePython Script

```
141:         prefix = "http://%s:81/sws/v2" % (args.ip,)
142:
143:     #? print "prefix: %s" % (prefix,) #?
144:
145:     ws = prefix + cmd
146:
147:     # if we do not already have a format parameter
148:     #
149:     if 'format' not in pieces:
150:         #if 'job' not in cmd:
151:             pieces['format'] = args.format
152:
153:     # add login parameters if we need to authenticate
154:     #
155:     if args.authenticate != '':
156:         pieces['username'] = username
157:         pieces['password'] = password
158:
159:     # all web-service URLs are lowercase
160:     #
161:     ws = ws.lower()
162:
163:     response = requests.get( ws, pieces, verify=False)
164:     if verbose:
165:         print
166:         print "WS: %s" % (urllib.unquote(response.url),)
167:
168:     rsp = response.text
169:     #? print "rsp; %s" % (rsp,) #?
170:     if response.status_code == 200:
171:         return rsp
172:     else:
173:         failures += 1
174:         return "Fail: %s" % (rsp,)
175:
```

## Appendix B: SamplePython Script

```
176:
177: if __name__ == '__main__':
178:     global username
179:     global password
180:
181:     process_args()
182:
183:     #Description of Web Services sample programs:
184:
185:     # 0) Please turn on web-services using the StorNext GUI
186:     #     Also choose protocol and authentication through the SN GUI
187:
188:     # 1) Do WS system info, returning TEXT, XML and JSON
189:     # /sws/v2/system/info?format=text
190:
191:     ws_rsp = do_webservices_cmd( '/system/info',
192:                                 { 'format' : 'text' } )
193:     print ws_rsp
194:
195:     # /sws/v2/system/info?format=xml
196:
197:     ws_rsp = do_webservices_cmd( '/system/info',
198:                                 { 'format' : 'xml' } )
199:     print ws_rsp
200:
201:     # /sws/v2/system/info?format=json
202:
203:     ws_rsp = do_webservices_cmd( '/system/info',
204:                                 { 'format' : 'json' } )
205:     print ws_rsp
206:
207:     # 2) Create a policy for a managed file system
208:
209:     # 3) 3 directories should exist:
210:     # a) Directory for single-file manipulation
```

```

211:
212:     singles_path = '/stornext/snfs1/sample_dir_singles/py'
213:     singles_paths_sync = []
214:     singles_paths_async = []
215:
216:     # b) Directory for directory manipulation
217:
218:     dirs_path_sync = '/stornext/snfs1/sample_dir_dirs_sync/py'
219:     dirs_paths_sync = []
220:     dirs_path_async = '/stornext/snfs1/sample_dir_dirs_async/py'
221:     dirs_paths_async = []
222:
223:     # c) Directory for multi-file manipulation
224:
225:     multi_path = '/stornext/snfs1/sample_dir_multi/py'
226:     multi_paths_sync = []
227:     multi_paths_async = []
228:
229:     # 4) Create arrays with path names of both files in each of those directories
230:     # for single and multi commands, files 0 and 1 are for sync, files 2 and 3 are for async
231:     # for directory commands, there is a directory for sync and one for async
232:
233:     for i in [0,1]:
234:         filename = "file.%d" % (i,)
235:         singles_paths_sync.append( "%s/%s" % (singles_path, filename))
236:         dirs_paths_sync.append(     "%s/%s" % (dirs_path_sync, filename))
237:         dirs_paths_async.append(    "%s/%s" % (dirs_path_async, filename))
238:         multi_paths_sync.append(    "%s/%s" % (multi_path, filename))
239:
240:     for i in [2,3]:
241:         filename = "file.%d" % (i,)
242:         singles_paths_async.append( "%s/%s" % (singles_path, filename))
243:         multi_paths_async.append(    "%s/%s" % (multi_path, filename))
244:
245:

```

## Appendix B: SamplePython Script

```
246:     # 5-11) steps 5-11 use sync mode, steps 12-18 use async mode
247:
248:     # 5) Use WS fsstore to save both files from the first directory to TAPE
249:     # /sws/v2/file/fsstore?file=<filepath>
250:
251:     for filepath in singles_paths_sync:
252:         ws_rsp = do_webservices_cmd( '/file/fsstore',
253:                                     { 'file' : filepath } )
254:         print ws_rsp
255:
256:     # /sws/v2/file/fsfileinfo?file=<filepath>
257:
258:     for filepath in singles_paths_sync:
259:         ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
260:                                     { 'file' : filepath } )
261:         print ws_rsp
262:
263:     # 6) Use WS fsstore to save the second directory to TAPE
264:     # /sws/v2/file/fsstore?directory=<dirpath>
265:
266:     ws_rsp = do_webservices_cmd( '/file/fsstore',
267:                                 { 'directory' : dirs_path_sync } )
268:     print ws_rsp
269:
270:     # /sws/v2/file/fsfileinfo?directory=<dirpath>
271:
272:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
273:                                 { 'directory' : dirs_path_sync } )
274:     print ws_rsp
275:
276:     # 7) Use WS fsstore to save both files from the third directory to TAPE
277:     # /sws/v2/file/fsstore?file=<f1>&file=<f2>
278:
279:     ws_rsp = do_webservices_cmd( '/file/fsstore',
280:                                 { 'file' : [ multi_paths_sync[0],
```

```

281:                                     multi_paths_sync[1] ] } )
282:     print ws_rsp
283:
284:
285:     # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
286:
287:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
288:                                 { 'file' : [ multi_paths_sync[0],
289:                                             multi_paths_sync[1] ] } )
290:     print ws_rsp
291:
292:
293:     # 8) Use WS rmdiskcopy to truncate both files in each of 3 directories
294:     # /sws/v2/file/fsrmdiskcopy
295:
296:     for filepath in singles_paths_sync:
297:         ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
298:                                     { 'file' : filepath } )
299:         print ws_rsp
300:
301:     for filepath in dirs_paths_sync:
302:         ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
303:                                     { 'file' : filepath } )
304:         print ws_rsp
305:
306:     for filepath in multi_paths_sync:
307:         ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
308:                                     { 'file' : filepath } )
309:         print ws_rsp
310:
311:     # 9) Use WS fsretrieve to restore both files to first directory from TAPE
312:     # /sws/v2/file/fsretrieve?file=<filepath>
313:
314:     for filepath in singles_paths_sync:
315:         ws_rsp = do_webservices_cmd( '/file/fsretrieve',

```

## Appendix B: SamplePython Script

```
316:             { 'file' : filepath } )
317:     print ws_rsp
318:
319:     # /sws/v2/file/fsfileinfo?file=<filepath>
320:
321:     for filepath in singles_paths_sync:
322:         ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
323:             { 'file' : filepath } )
324:         print ws_rsp
325:
326:     # 10) Use WS fsretrieve to restore the second directory from TAPE
327:     # /sws/v2/file/fsretrieve?directory=<dirpath>
328:
329:     ws_rsp = do_webservices_cmd( '/file/fsretrieve',
330:         { 'directory' : dirs_path_sync } )
331:     print ws_rsp
332:
333:     # /sws/v2/file/fsfileinfo?directory=<dirpath>
334:
335:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
336:         { 'directory' : dirs_path_sync } )
337:     print ws_rsp
338:
339:     # 11) Use WS fsretrieve to restore both files in the third directory from TAPE
340:     # /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
341:
342:     ws_rsp = do_webservices_cmd( '/file/fsretrieve',
343:         { 'file' : [ multi_paths_sync[0],
344:             multi_paths_sync[1] ] } )
345:     print ws_rsp
346:
347:
348:     # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
349:
350:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
```



```

351:             { 'file' : [ multi_paths_sync[0],
352:                       multi_paths_sync[1] ] } )
353: print ws_rsp
354:
355:
356: # 12-18) Repeat steps 5-11 using async mode
357:
358: # 12) Use WS fsstore to save both files from the first directory to TAPE
359: # /sws/v2/file/fsstore?file=<filepath>
360:
361: for filepath in singles_paths_async:
362:     ws_rsp = do_webservices_cmd( '/file/fsstore',
363:                                 { 'file' : filepath,
364:                                   'mode' : 'async' } )
365:     check_job_status( ws_rsp)
366:
367: # /sws/v2/file/fsfileinfo?file=<filepath>
368:
369: for filepath in singles_paths_async:
370:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
371:                                 { 'file' : filepath } )
372:     print ws_rsp
373:
374: # 13) Use WS fsstore to save the second directory to TAPE
375: # /sws/v2/file/fsstore?directory=<dirpath>
376:
377: ws_rsp = do_webservices_cmd( '/file/fsstore',
378:                             { 'directory' : dirs_path_async,
379:                               'mode' : 'async' } )
380: check_job_status( ws_rsp)
381:
382: # /sws/v2/file/fsfileinfo?directory=<dirpath>
383:
384: ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
385:                             { 'directory' : dirs_path_async } )

```

```
386:     print ws_rsp
387:
388:     # 14) Use WS fsstore to save both files from the third directory to TAPE
389:     # /sws/v2/file/fsstore?file=<f1>&file=<f2>
390:
391:     ws_rsp = do_webservices_cmd( '/file/fsstore',
392:                                 { 'file' : [ multi_paths_async[0],
393:                                             multi_paths_async[1] ],
394:                                 'mode' : 'async' } )
395:     check_job_status( ws_rsp)
396:
397:     # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
398:
399:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
400:                                 { 'file' : [ multi_paths_async[0],
401:                                             multi_paths_async[1] ] } )
402:     print ws_rsp
403:
404:
405:     # 15) Use WS rmdiskcopy to truncate both files in each of 3 directories
406:     # /sws/v2/file/fsrmdiskcopy
407:
408:     for filepath in singles_paths_async:
409:         ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
410:                                     { 'file' : filepath } )
411:         print ws_rsp
412:
413:     for filepath in dirs_paths_async:
414:         ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
415:                                     { 'file' : filepath } )
416:         print ws_rsp
417:
418:     for filepath in multi_paths_async:
419:         ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
420:                                     { 'file' : filepath } )
```

## Appendix B: SamplePython Script

```
421:         print ws_rsp
422:
423:     # 16) Use WS fsretrieve to restore both files to first directory from TAPE
424:     # /sws/v2/file/fsretrieve?file=<filepath>
425:
426:     for filepath in singles_paths_async:
427:         ws_rsp = do_webservices_cmd( '/file/fsretrieve',
428:                                     { 'file' : filepath,
429:                                       'mode' : 'async' } )
430:         check_job_status( ws_rsp)
431:
432:     # /sws/v2/file/fsfileinfo?file=<filepath>
433:
434:     for filepath in singles_paths_async:
435:         ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
436:                                     { 'file' : filepath } )
437:         print ws_rsp
438:
439:     # 17) Use WS fsretrieve to restore the second directory from TAPE
440:     # /sws/v2/file/fsretrieve?directory=<dirpath>
441:
442:     ws_rsp = do_webservices_cmd( '/file/fsretrieve',
443:                                 { 'directory' : dirs_path_async,
444:                                   'mode' : 'async' } )
445:     check_job_status( ws_rsp)
446:
447:     # /sws/v2/file/fsfileinfo?directory=<dirpath>
448:
449:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
450:                                 { 'directory' : dirs_path_async } )
451:     print ws_rsp
452:
453:     # 18) Use WS fsretrieve to restore both files in the third directory from TAPE
454:     # /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
455:
```

## Appendix B: SamplePython Script

```
456:     ws_rsp = do_webservices_cmd( '/file/fsretrieve',
457:                                   { 'file' : [ multi_paths_async[0],
458:                                               multi_paths_async[1] ],
459:                                   'mode' : 'async' } )
460:     check_job_status( ws_rsp)
461:
462:     # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
463:
464:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
465:                                   { 'file' : [ multi_paths_async[0],
466:                                               multi_paths_async[1] ] } )
467:
468:     # 19) Get policy information
469:
470:     ws_rsp = do_webservices_cmd( '/policy/fsdirclass',
471:                                   { 'directory' : dirs_path_async } )
472:     print ws_rsp
473:
474:     ws_rsp = do_webservices_cmd( '/policy/fsclassinfo',
475:                                   { 'policy' : 'policy_min_i' } )
476:     print ws_rsp
477:
478:     print
479:     print "Number of failures: %d" % (failures,)
480:
481:
482: #           Copyright 2015 Quantum Corporation
483: # vim:ts=4:sw=4:et:ic:bg=dark:
```