Web Services Guide
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

© 2019 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

Preface .................................................................................................................. V
  StorNext Web Services Version 2 (V2) ............................................................... v
  Audience ......................................................................................................... vi
  Ports Used By StorNext Web Services .............................................................. vi
    HTTP Port .................................................................................................. vi
    HTTPS Port ................................................................................................ vi
    View the List of Ports .................................................................................. vi

Chapter 1: StorNext Web Services Commands (V2) ........................................... 1
  StorNext Web Services V2 Commands, Usage, and Descriptions .................. 2
  Using the Web Services .................................................................................. 2
  Overview ....................................................................................................... 3
  Archive .......................................................................................................... 9
  Directory ...................................................................................................... 16
  Drive ............................................................................................................. 23
  File ................................................................................................................. 29
  Job ................................................................................................................. 70
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>77</td>
</tr>
<tr>
<td>Object Storage</td>
<td>137</td>
</tr>
<tr>
<td>Policy</td>
<td>142</td>
</tr>
<tr>
<td>Quota</td>
<td>157</td>
</tr>
<tr>
<td>Report</td>
<td>161</td>
</tr>
<tr>
<td>Schedule</td>
<td>185</td>
</tr>
<tr>
<td>System</td>
<td>205</td>
</tr>
</tbody>
</table>

### Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2  

- Prerequisites ........................................... 218
- Getting Started .......................................... 218
- Run the Web Services .................................... 221
- Run the Web Services from a Client Application  225
- Troubleshooting ........................................... 271

### Appendix A: Sample Perl Script  

- Sample Perl Script ......................................... 274

### Appendix B: Sample Python Script  

- Sample Python Script ....................................... 292
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: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2**
- **Appendix A: Sample Perl Script**
- **Appendix B: Sample Python Script**

**StorNext Web Services Version 2 (V2)**

Version 2 of StorNext web services offers enhanced 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.

Use the StorNext GUI to enable or disable the web service, set the protocol (HTTP or HTTPS), and control authentication. By default, version 2 web services are turned off (see Enable Web Services on page 218).

For API commands supported by the Web Services (V2), see StorNext Web Services V2 Commands, Usage, and Descriptions on page 2.
Audience

This manual is written for StorNext 6 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 fports(4) man page.

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>StorNext Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
<td>TCP</td>
<td>GUI (Java), Web Services</td>
<td>User starts at port 81, redirected to 443</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>GUI (Java), Web Services</td>
<td></td>
</tr>
</tbody>
</table>

HTTP Port

The default HTTP port is 81.

**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.

**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:

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

The following formats indicate important information:

- **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 Informationsvedrørende system-, sikkerheds- og lovbemmelser 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.


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 produkt system, 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 설명서에 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：在操作此产品前，请閱讀本檔案及 Quantum 產品系統、安全與法規資訊指南中的指示與警告說明。

For the most up to date information on StorNext 6, see:
http://www.quantum.com/serviceandsupport/get-help/index.aspx#contact-support

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

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

<table>
<thead>
<tr>
<th>Region</th>
<th>Support Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>1-800-284-5101 (toll free) +1-720-249-5700</td>
</tr>
<tr>
<td>EMEA</td>
<td>+800-7826-8888 (toll free) +49 6131 324 185</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>+800-7826-8887 (toll free) +603-7953-3010</td>
</tr>
</tbody>
</table>

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:
Chapter 1: StorNext Web Services Commands (V2)

This chapter contains the following topics:

- StorNext Web Services V2 Commands, Usage, and Descriptions .............................................. 2
- Using the Web Services .................................................................................................................. 2
- Overview ........................................................................................................................................ 3
- Archive .......................................................................................................................................... 9
- Directory ....................................................................................................................................... 16
- Drive ............................................................................................................................................ 23
- File ............................................................................................................................................... 29
- Job ................................................................................................................................................. 70
- Media ............................................................................................................................................ 77
- Object Storage ................................................................................................................................. 137
- Policy ............................................................................................................................................ 142
- Quota ............................................................................................................................................. 157
- Report .......................................................................................................................................... 161
- Schedule ....................................................................................................................................... 185
- System ......................................................................................................................................... 205
StorNext Web Services V2 Commands, Usage, and Descriptions

This section provides API commands supported by the Web Services (V2), 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 enable or disable the web service, set the protocol (HTTP or HTTPS), and control authentication. By default, version 2 web services are turned off (see Enable Web Services on page 218).

Web Services Options

The table below describes the allowed web service options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>HTTP</td>
<td>Allow web services over HTTP.</td>
</tr>
<tr>
<td></td>
<td>HTTPS</td>
<td>Allow web services over HTTPS.</td>
</tr>
<tr>
<td></td>
<td>HTTP or HTTPS</td>
<td>Allow web services over HTTP or HTTPS.</td>
</tr>
</tbody>
</table>
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 on the next page.
Table 1: Web Service Categories and General Descriptions

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>Return information about an archive, query an archive port, or change the state of an archive.</td>
</tr>
<tr>
<td>Directory</td>
<td>Modify the class attributes of a directory or retrieve or recover files from media.</td>
</tr>
<tr>
<td>Drive</td>
<td>Report or change the state of drive components and storage subsystems.</td>
</tr>
<tr>
<td>File</td>
<td>Report, retrieve, and store files to tiered storage.</td>
</tr>
<tr>
<td>Job</td>
<td>Return information about jobs.</td>
</tr>
<tr>
<td>Media</td>
<td>Manage media – copy, clean up, move, and report.</td>
</tr>
<tr>
<td>Object Storage</td>
<td>Report Object Storage components.</td>
</tr>
<tr>
<td>Policy</td>
<td>Manage and report policies.</td>
</tr>
<tr>
<td>Quota</td>
<td>Manage and report quotas.</td>
</tr>
<tr>
<td>Report</td>
<td>Return information about subsystem resource requests.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Manage and report schedules.</td>
</tr>
<tr>
<td>System</td>
<td>Get the status of system and Tertiary Storage Manager (TSM) components. Manage and report backups.</td>
</tr>
</tbody>
</table>

Table 2 below provides a brief description of each web service and the minimum access control required to execute the command (see Create the SWS V2 Web Service User on page 220).

Table 2: Web Service Descriptions

<table>
<thead>
<tr>
<th>Web Service</th>
<th>Description</th>
<th>Minimum Access Control Required to Execute the Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Archive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query</td>
<td>Return information about an archive.</td>
<td>Destination, Read-Only</td>
</tr>
<tr>
<td>Query port</td>
<td>Query an archive port.</td>
<td>Destination, Read-Only</td>
</tr>
<tr>
<td>Change state</td>
<td>Change the state of an archive.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td><strong>Directory</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change attributes</td>
<td>Modify the class attributes of a directory.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Web Service</td>
<td>Description</td>
<td>Minimum Access Control Required to Execute the Command</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Retrieve files</td>
<td>Retrieve files from media and place them on disk.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Drive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Report the state of storage subsystem drive components and storage subsystems and Tertiary Storage Manager (TSM) software.</td>
<td>Destination, Read-Only</td>
</tr>
<tr>
<td>Change state</td>
<td>Change the state of a storage component in the Quantum storage subsystem.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>Generate a report about files known to the Tertiary Storage Manager.</td>
<td>File, Read-Only</td>
</tr>
<tr>
<td>Tape location</td>
<td>Generate a report about a file’s tape copy location.</td>
<td>File, Read-Only</td>
</tr>
<tr>
<td>Retrieve files</td>
<td>Retrieve files from media and place on disk.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Retrieve files / new location</td>
<td>Retrieve files from media and place into new disk file.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Relocate file from one affinity to another</td>
<td>Relocates a managed file from one disk affinity to another.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Retrieve files / partial</td>
<td>Retrieve partial files from media and place on disk.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Store</td>
<td>Expedite the storage of a file that currently resides on disk to media.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Remove disk copy</td>
<td>Remove the copy of a file from disk after the file was stored to a medium.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td>Modify class attributes</td>
<td>Modify the class attributes of a file.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asynchronous job status</td>
<td>Returns a status of the jobs that are invoked asynchronously.</td>
<td>None</td>
</tr>
<tr>
<td>Web Service</td>
<td>Description</td>
<td>Minimum Access Control Required to Execute the Command</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Detailed mover job status</td>
<td>Returns a detailed status of the mover jobs that are invoked asynchronously.</td>
<td>None</td>
</tr>
<tr>
<td>Media</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>Generate a media report based on their current status.</td>
<td>Destination, Read-Only</td>
</tr>
<tr>
<td>File background job status</td>
<td>Reports the status of the background job.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>File inactive removal</td>
<td>Remove inactive versions of files.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>File inactive remove by media</td>
<td>Clean all files on media that have been marked as logically blank.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>File inactive removal list</td>
<td>List all media that has been marked as logically blank.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Move media</td>
<td>Move media from one archive to another.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Import cleaning media</td>
<td>Imports a list of cleaning media.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Import media</td>
<td>Imports a list of media.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Export media</td>
<td>Exports a list of media.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Copy media</td>
<td>Initiates copy-replace for a media.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Fragmentation report</td>
<td>Report on media fragmentation.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Eject media</td>
<td>Eject media out of an archive to be entered into another archive.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Enter media</td>
<td>Enter media that has been ejected out of an archive into another archive.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Purge media</td>
<td>Purges a list of media.</td>
<td>Destination, Read-Write</td>
</tr>
</tbody>
</table>
## Chapter 1: StorNext Web Services Commands (V2)

### Overview

<table>
<thead>
<tr>
<th>Web Service</th>
<th>Description</th>
<th>Minimum Access Control Required to Execute the Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute query</td>
<td>Queries for the attributes of one or more specified media.</td>
<td>Destination, Read-Only</td>
</tr>
<tr>
<td>Change media state</td>
<td>Change the class or state of a media.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>Checkout media</td>
<td>Check media out of the Media Manager system.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>List media for removal</td>
<td>List all media that are marked for removal.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>List media by state</td>
<td>Lists media by state.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td>List media by location</td>
<td>Lists media by location.</td>
<td>Destination, Read-Write</td>
</tr>
<tr>
<td><strong>Object Storage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Object Storage components</td>
<td>Report Object Storage components.</td>
<td>Destination, Read-Only</td>
</tr>
<tr>
<td><strong>Policy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class information</td>
<td>Report information on storage manager policies.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Modify a policy</td>
<td>Modify the processing parameters of a policy class.</td>
<td>Policy, Read-Write</td>
</tr>
<tr>
<td>Report policy class for a directory</td>
<td>Reports the policy class associated with directory.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Report policy class for a file system</td>
<td>Reports all policy classes with association points in a file system.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td><strong>Quota</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage quotas</td>
<td>Manage the quota system in the StorNext file system.</td>
<td>File, Read-Write</td>
</tr>
<tr>
<td><strong>Report</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Chapter 1: StorNext Web Services Commands (V2)

### Overview

<table>
<thead>
<tr>
<th>Web Service</th>
<th>Description</th>
<th>Minimum Access Control Required to Execute the Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel requests</td>
<td>Cancels requests.</td>
<td>Policy, Read-Write</td>
</tr>
<tr>
<td>Files</td>
<td>Reports all files in the queue or specific files if a request identifiers or filename is specified.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Media</td>
<td>Reports the media movement for a request identifier or all media in queue.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Mover host</td>
<td>Reports the active mover host summary.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Mover request</td>
<td>Reports the active mover request summary.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Resource</td>
<td>Reports the active resource request summary.</td>
<td>Policy, Read-Only</td>
</tr>
<tr>
<td>Status</td>
<td>Reports the status summary information on requests.</td>
<td>Policy, Read-Only</td>
</tr>
</tbody>
</table>

### Schedule

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Description</th>
<th>Minimum Access Control Required to Execute the Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Prints the report of a feature or schedule.</td>
<td>System, Read-Only</td>
</tr>
<tr>
<td>Create</td>
<td>This web service will create a schedule.</td>
<td>System, Read-Write</td>
</tr>
<tr>
<td>Update</td>
<td>This web service will update a schedule.</td>
<td>System, Read-Write</td>
</tr>
<tr>
<td>Delete</td>
<td>This web service deletes an existing schedule.</td>
<td>System, Read-Write</td>
</tr>
<tr>
<td>Reset</td>
<td>This web service resets an existing schedule.</td>
<td>System, Read-Write</td>
</tr>
</tbody>
</table>

### System

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Minimum Access Control Required to Execute the Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup</td>
<td>Execute backup of configuration, database, and file system metadata.</td>
<td>System, Read-Write</td>
</tr>
<tr>
<td>Backup status</td>
<td>Retrieves the status of the backup operation.</td>
<td>System, Read-Only</td>
</tr>
<tr>
<td>File System Report</td>
<td>Reports the status of a file system and the status of stripe groups that belong to it.</td>
<td>File, Read-Only</td>
</tr>
<tr>
<td>Information</td>
<td>Retrieves the latest status of system components.</td>
<td>System, Read-Only</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Archive

Archive / Query

Return information about an archive.

This web service runs the `vsarchiveqry` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-F</td>
</tr>
<tr>
<td>archive</td>
<td>Optional</td>
<td>1</td>
<td>Identifies the archive to be queried. If not specified, all archives are queried.</td>
<td>-a (if absent)</td>
</tr>
</tbody>
</table>
### 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 false. | -c
showdrives | Optional | 1 | Indicates that all drives associated with the specified archive(s) are to be reported. Default value is false. | -d
showmedia | Optional | 1 | Indicates that all media associated with the specified archive(s) are to be reported. Default value is false. | -m
showports | Optional | 1 | Indicates that all import/export ports associated with the specified archive(s) are to be reported. Default value is false. | -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 false. | -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 1 (highest) to 32 (lowest) inclusive. The default priority value is 15. | -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 3. | -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 120 seconds. | -T
rpcnumber | Optional | 1 | The RPC program number for the Media Manager software. The default value for the Media Manager software program number is 300016. | -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>
    </archive>
  </archives>
</vsarchiveqry>
Archive / Query Port

Query an archive port.

This web service runs the `mmportinfo` command.

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

## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive</td>
<td>Required</td>
<td>1</td>
<td>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.</td>
<td>archive name</td>
</tr>
</tbody>
</table>
Example

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

Output

Exit Code: 0
Output: 512:LTOW,LTO:0,0,15,512

Archive / Change State

Change the state of an archive.
This web service runs the vsarchivevary command.

Parameters

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

Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.
Chapter 1: StorNext Web Services Commands (V2)

Archive

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>archive</td>
<td>Required</td>
<td>1</td>
<td>Identifies the archive to be queried.</td>
<td>archive</td>
</tr>
<tr>
<td>state</td>
<td>Required</td>
<td>1</td>
<td>Identifies the state into which the archive is placed. Valid archive states are: on-line, off-line, and diagnostic. The archive states, on-line, off-line, and diagnostic are abbreviated as on, of, and d respectively.</td>
<td>-s</td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>The host name of the Media Manager server.</td>
<td>-H</td>
</tr>
<tr>
<td>priority</td>
<td>Optional</td>
<td>1</td>
<td>The execution priority of the entered command. Assignable priority values are restricted to a range from 1 (highest) to 32 (lowest) inclusive. The default priority value is 15.</td>
<td>-P</td>
</tr>
<tr>
<td>retries</td>
<td>Optional</td>
<td>1</td>
<td>The number of retries that web service will attempt if a timeout is returned by the API software. The default retries value is 3.</td>
<td>-R</td>
</tr>
<tr>
<td>timeout</td>
<td>Optional</td>
<td>1</td>
<td>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 120 seconds.</td>
<td>-T</td>
</tr>
<tr>
<td>rpcnumber</td>
<td>Optional</td>
<td>1</td>
<td>The RPC program number for the Media Manager software. The default value for the Media Manager software program number is 300016.</td>
<td>-V</td>
</tr>
</tbody>
</table>

Example

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

```xml
<?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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>directory</td>
<td>Required</td>
<td>N</td>
<td>Directory or directories to process.</td>
<td>directory name</td>
</tr>
<tr>
<td>recursive</td>
<td>Optional</td>
<td>1</td>
<td>Only one directory can be specified when the parameter is used.</td>
<td>-R or -D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A value of false will process directory and its contents while a value of true will perform a recursive operation on the directory.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If this parameter is not used, then the directories specified by the directory parameter are processed.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>stubsize</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-S</td>
</tr>
<tr>
<td>reloc</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates how relocation policies operate. Valid values are exclude and policy. The exclude option excludes files from storage when a store policy occurs. The policy option stores files by storage policy.</td>
<td>-r</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

**Directory**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>store</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates how storage policies operate. Valid values are <code>exclude</code> and <code>policy</code>. The <code>exclude</code> option excludes files from storage when a store policy occurs. The <code>policy</code> option stores files by storage policy.</td>
<td>- s</td>
</tr>
<tr>
<td>trunc</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates how truncation policies operate. Valid values are <code>exclude</code> and <code>policy</code>. The <code>exclude</code> option excludes files from storage when a store policy occurs. The <code>policy</code> option stores files by storage policy.</td>
<td>- t</td>
</tr>
<tr>
<td>clean</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates if the database entries are to be cleaned when the file is removed from the file system. The <code>yes</code> option indicates that the database entries will be cleaned and the file will NOT be recoverable. The <code>no</code> option indicates that the database entries will NOT be cleaned and the file will be recoverable.</td>
<td>- C</td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>. <strong>Note</strong>: The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>1</td>
<td>This specifies the class that will be associated with the directory.</td>
<td>- c</td>
</tr>
</tbody>
</table>

### Example

```plaintext
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
Directory / Retrieve Files

Retrieve or recover files from media and place on disk.

This web service runs the `fsretrieve` command.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### Directory

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default is <strong>text</strong>.</td>
<td>-F</td>
</tr>
<tr>
<td>directory</td>
<td>Required</td>
<td>1</td>
<td>The directory from which to start the recursive retrieves.</td>
<td>-R</td>
</tr>
<tr>
<td>updateaccesstime</td>
<td>Optional</td>
<td>1</td>
<td>Updates the access time for the requested files. Default value is <strong>false</strong>.</td>
<td>-a</td>
</tr>
<tr>
<td>toppriority</td>
<td>Optional</td>
<td>1</td>
<td>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 <strong>false</strong>.</td>
<td>-p</td>
</tr>
</tbody>
</table>

#### 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>
```

- Note: The **fast** and **slow** queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.
<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>
</status>
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### Drive

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>drive</td>
<td>Optional</td>
<td>1</td>
<td>Component alias drive identifier. Submitting the fsstate 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.</td>
<td>component alias</td>
</tr>
</tbody>
</table>

### Note

The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

### Example

https://<<SERVER>>/sws/v2/drive/fsstate

?drive=vtl_dr1

&format=xml

### Output

```xml
<?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>
    </header>
</fsstate>
```
<commandDescription>Report the state of all Stornext drive components and storage subsystems and/or Tertiary 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>

$statuses$

$_statuses$

$statusCode>FS0000</statusCode>
$statusNumber>0</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>387993</requestId>
<commandName>fsstate</commandName>
<commandStatus>completed</commandStatus>
$statusText>Command Successful.</statusText>
</status>
Drive / Change State

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

This web service runs the `fschstate` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Drive

### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>drive</td>
<td>Required</td>
<td>1</td>
<td>Component alias drive identifier.</td>
<td>component alias</td>
</tr>
<tr>
<td>state</td>
<td>Required</td>
<td>1</td>
<td>The desired state of the drive components or subsystems. The valid values for drive components are MAINT, ON, or OFF. Valid values for subsystems are ON or OFF.</td>
<td>-s</td>
</tr>
</tbody>
</table>

### 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>
File

File / Information

Generate a report about files known to the Tertiary Storage Manager. This web service runs the `fsfileinfo` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>Optional</td>
<td>1</td>
<td>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 <code>false</code>.</td>
<td>-c</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>compressioninfo</td>
<td>Optional</td>
<td>1</td>
<td>If compression is enabled for the file when stored, the compression type, size, and ratio for the file is displayed. Default value is false.</td>
<td>-q</td>
</tr>
<tr>
<td>directory</td>
<td>Optional</td>
<td>1</td>
<td>The directory to process. The recursive parameter determines how the directory will be processed. Note: At least one file or directory path is required.</td>
<td>-R or -D</td>
</tr>
<tr>
<td>encryptioninfo</td>
<td>Optional</td>
<td>1</td>
<td>If encryption is enabled for the file when stored, the encryption type for the file is displayed. Default value is false.</td>
<td>-e</td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>1</td>
<td>List of files for which information is required. Note: At least one file or directory path is required.</td>
<td>filename</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast. Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>objectinfo</td>
<td>Optional</td>
<td>1</td>
<td>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 false.</td>
<td>-o</td>
</tr>
</tbody>
</table>
### Example

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

### Output

```xml
<?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>
  </header>
</fsfileinfo>
```
<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>
  </fileInfo>
</fileInfos>
<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>
<returnCode>0</returnCode>
File / Expire Copies of a File

Allows you to expire numbered copies from all versions of the specified file(s). You must first configure the copy numbers for expiration. When you create or modify a file, you must create all copies for the current version before any copies can be expired. You can manually expire copies before the expiration interval has elapsed. You cannot expire copies for Write Once Read Multiple (WORM) media types.

This web service runs the `/usr/adic/TSM/exec/fsexpcopy` command.

**Parameters**

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

**Note:** The `fast` and `slow` queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.
### Chapter 1: StorNext Web Services Commands (V2)

#### File

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>copy</td>
<td>Optional</td>
<td>1</td>
<td>The copy number to expire.</td>
<td>-c</td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>N</td>
<td>One or more file(s) to expire.</td>
<td>file name</td>
</tr>
<tr>
<td>dir</td>
<td>Optional</td>
<td>1</td>
<td>The directory from which to do start the recursive file expiration.</td>
<td>-R</td>
</tr>
<tr>
<td>scheduled</td>
<td>Optional</td>
<td>1</td>
<td>Boolean that selects scheduled processing for automatic expiration of copies based on time since last access of a file. When specified, the copy, file and dir parameters cannot be set.</td>
<td>-S</td>
</tr>
</tbody>
</table>

#### JSON Example

```
https://<<SERVER>>/sws/v2/policy/fsexpcopy
?copy=2
&file=/stornext/snfs/file1
/&stornext/snfs/file2
&format=JSON
```

#### JSON Output

```
{
```
"header": {
  "commandName": "fsexpcopy",
  "commandDescription": "Expire file copies from media",
  "localDateISO": "2018-06-05T16:05:46",
  "localDate": "2018-06-05",
  "localTime": "16:05:46",
  "localDayOfWeek": 2,
  "gmtDateISO": "2018-06-05T20:05:46Z",
  "gmtDate": "2018-06-05",
  "gmtTime": "20:05:46",
  "gmtDayOfWeek": 2
},
"statuses": [
  {
    "statusCode": "FS0850",
    "statusNumber": 850,
    "dayOfMonth": 5,
    "requestId": 78396,
    "commandName": "/usr/adic/TSM/exec/fsexpcopy",
    "commandStatus": "interim",
    "statusText": "Copy(s) successfully expired for /stornext/snfs/file1."
  },
  {
    "statusCode": "FS0850",
    "statusNumber": 850,
    "dayOfMonth": 5,
    "requestId": 78396,
    "commandName": "/usr/adic/TSM/exec/fsexpcopy",
    "commandStatus": "interim",
    "statusText": "Copy(s) successfully expired for /stornext/snfs/file2."
  },
  {
    "statusCode": "FS0390",
    "statusNumber": 390,
    "dayOfMonth": 5,
    "requestId": 78396,
    "commandName": "/usr/adic/TSM/exec/fsexpcopy",
    "commandStatus": "completed",
    "statusText": "2 out of 2 file copy expires were
Chapter 1: StorNext Web Services Commands (V2)

File

successful."
}
{
    "statusCode": "FS0000",
    "statusNumber": 0,
    "dayOfMonth": 5,
    "requestId": 78396,
    "commandName": "/usr/adic/TSM/exec/fsexpcopy",
    "commandStatus": "completed",
    "statusText": "Command Successful."
}
],
"footer": {
    "returnCode": 0,
    "localDateISOEnd": "2018-06-05T16:05:46",
    "localDateEnd": "2018-06-05",
    "localTimeEnd": "16:05:46",
    "localDayOfWeekEnd": 2,
    "gmtDateISOEnd": "2018-06-05T20:05:46Z",
    "gmtDateEnd": "2018-06-05",
    "gmtTimeEnd": "20:05:46",
    "gmtDayOfWeekEnd": 2,
    "elapsedTimeInSeconds": "0.0037"
}
}

XML Example

https://<<SERVER>>/sws/v2/policy/fsexpcopy
?copy=2
&file=/stornext/snfs/file1
/&stornext/snfs/file2
&format=XML

XML Output

<?xml version="1.0" encoding="UTF-8"?>
<fsexpcopy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
File / Tape Location

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

This web service runs the fsfiletapeloc command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### File

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>file</td>
<td>Required</td>
<td>1</td>
<td>The file for which location is being queried. The path name of the file is required.</td>
<td>file name</td>
</tr>
<tr>
<td>copynumber</td>
<td>Optional</td>
<td>1</td>
<td>The copy ID to generate the report for. If not specified, the information for the primary copy will be reported.</td>
<td>-c</td>
</tr>
</tbody>
</table>

### Example

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

### Output

```xml
<?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>2017-10-05T13:27:22</localDateISO>
  </header>
</fsfiletapeloc>
```
File / Retrieve Files

Retrieve truncated files from media and place on disk.

This web service runs the fsretrieve command.

For additional information, see How to Find the async Job ID in the Output on page 67.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity</td>
<td>Optional</td>
<td>1</td>
<td>This parameter retrieves file(s) to a specific file system affinity.</td>
<td>-t</td>
</tr>
<tr>
<td>allocation</td>
<td>Optional</td>
<td>1</td>
<td>This parameter allocates disk blocks to files in alphabetic order in an attempt to provide sequential block allocation to all files in the specified directory.</td>
<td>-A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: This parameter is only valid when the directory parameter is used and the recursive parameter is false.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The default value is false.</td>
<td></td>
</tr>
<tr>
<td>copynumber</td>
<td>Optional</td>
<td>1</td>
<td>Used to retrieve a specific copy of filename if one exists.</td>
<td>-c</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
<td>-----</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>directory</td>
<td>Optional</td>
<td>1</td>
<td>The directory to process. The recursive parameter determines how the directory will be processed.</td>
<td>-R or -D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> At least one file or directory path is required.</td>
<td></td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>N</td>
<td>List of files that need to be retrieved.</td>
<td>filename</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> At least one file or directory path is required.</td>
<td></td>
</tr>
<tr>
<td>forcealtlocation</td>
<td>Optional</td>
<td>1</td>
<td>Force change in alternate retrieval location behavior. Valid values are <strong>true</strong> and <strong>false</strong>.</td>
<td>-x</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>.</td>
<td>-F</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>recursive</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is only valid when the directory parameter is specified. A value of false will only process the contents of the directory while a value of true will perform a recursive operation on the directory. The default value is <strong>true</strong>.</td>
<td>-R or -D</td>
</tr>
</tbody>
</table>
## Chapter 1: StorNext Web Services Commands (V2)

**File**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrievaltype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter specifies the type of retrieval from glacier storage.</td>
<td>-g</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The following types of retrieval are supported:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>standard</strong>: AWS Glacier standard storage class allowing data access within 3-5 hours.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>expedited</strong>: AWS Glacier expedited storage class allowing data access within 1-5 minutes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- <strong>bulk</strong>: AWS Glacier bulk storage class allowing access to large amounts of data within a day.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If this option is not specified, then the retrieval type defaults to the configured value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note</strong>: This option is ignored if it is not supported by the media.</td>
<td></td>
</tr>
<tr>
<td>toppriority</td>
<td>Optional</td>
<td>1</td>
<td>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 <strong>false</strong>.</td>
<td>-p</td>
</tr>
<tr>
<td>updateaccesstime</td>
<td>Optional</td>
<td>1</td>
<td>Updates the access time for the requested files.</td>
<td>-a</td>
</tr>
</tbody>
</table>

### Example

https://<<SERVER>>/sws/v2/file/fsretrieve

?file=/stornext/snfs1/xxx/testfile.0

&format=xml

### Output

```xml
<?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>
    </status>
  </statuses>
</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.
For additional information, see [How to Find the async Job ID in the Output on page 67](#).

## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-F</td>
</tr>
<tr>
<td>file</td>
<td>Required</td>
<td>1</td>
<td>File that needs to be retrieved.</td>
<td>file name</td>
</tr>
<tr>
<td>newfile</td>
<td>Required</td>
<td>1</td>
<td>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.</td>
<td>-n</td>
</tr>
<tr>
<td>copynumber</td>
<td>Optional</td>
<td>1</td>
<td>Used to retrieve a specific copy of filename if one exists.</td>
<td>-c</td>
</tr>
<tr>
<td>updateaccessetime</td>
<td>Optional</td>
<td>1</td>
<td>Updates the access time for the requested files.</td>
<td>-a</td>
</tr>
<tr>
<td>toppriority</td>
<td>Optional</td>
<td>1</td>
<td>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 <code>false</code>.</td>
<td>-p</td>
</tr>
</tbody>
</table>
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>
    </status>
  </statuses>
</fsretrieve>
<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>
File / Relocate File From One Affinity To Another

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

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity</td>
<td>Required</td>
<td>1</td>
<td>The destination affinity for the files.</td>
<td>- a</td>
</tr>
<tr>
<td>allocation</td>
<td>Optional</td>
<td>1</td>
<td>This allocates disk blocks to files in alphabetic order in an attempt to provide sequential block allocation to all files in the specified directory. This parameter is only valid when the directory parameter is used and the recursive parameter is false. The default value is false.</td>
<td>-A</td>
</tr>
<tr>
<td>directory</td>
<td>Optional</td>
<td>1</td>
<td>The directory to process. The recursive parameter determines how the directory is processed.</td>
<td>- R or - D</td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>N</td>
<td>The files that need to be relocated.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are json and text. The default value is text.</td>
<td>- F</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req/Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>recursive</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is only valid when the directory parameter is specified. A value of <strong>false</strong> only process the contents of the directory, while a value of <strong>true</strong> performs a recursive operation on the directory. The default is <strong>true</strong>.</td>
<td>-R or -D</td>
</tr>
</tbody>
</table>

### Example

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

### Output

```json
{
  "header": {
    "commandName": "fsrelocate",
    "commandLine": "fsrelocate /stornext/snfs1/pol1/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",
      "statusMessage": "Operation completed successfully.
```

---

Chapter 1: StorNext Web Services Commands (V2)

File
"statusNumber": 589,
"dayOfMonth": 3,
"requestId": 255713,
"commandName": "fsrelocate",
"commandStatus": "interim",
"statusText": "Tertiary Manager software request received. /stornext/snfs1/pol1/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"
}
}
Chapter 1: StorNext Web Services Commands (V2)
File

File / Retrieve Files / Partial
Retrieve files from media and place them on disk.
This web service runs the fsretrieve command.
For additional information, see How to Find the async Job ID in the Output on page 67.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast. Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>file</td>
<td>Required</td>
<td>1</td>
<td>File that needs to be retrieved.</td>
<td>file name</td>
</tr>
<tr>
<td>newfile</td>
<td>Required</td>
<td>1</td>
<td>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.</td>
<td>-n</td>
</tr>
<tr>
<td>startbyte</td>
<td>Required</td>
<td>1</td>
<td>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.</td>
<td>-b</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### File

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>endbyte</td>
<td>Required</td>
<td>1</td>
<td>The last byte for the file to be retrieved.</td>
<td>-b</td>
</tr>
<tr>
<td>updateaccesstime</td>
<td>Optional</td>
<td>1</td>
<td>Updates the access time for the requested files.</td>
<td>-a</td>
</tr>
<tr>
<td>toppriority</td>
<td>Optional</td>
<td>1</td>
<td>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 false.</td>
<td>-p</td>
</tr>
</tbody>
</table>

### Example

```xml
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
<?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>
  </header>
</fsretrieve>
```
<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>

<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>

<statusCode>FS0390</statusCode>
<statusNumber>390</statusNumber>
<dayOfMonth>15</dayOfMonth>
<requestId>388031</requestId>
<commandName>fsretrieve</commandName>
File / Store

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

This web service runs the `fsstore` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>copies</td>
<td>Optional</td>
<td>1</td>
<td>Number of copies of the file(s) to be stored.</td>
<td>-c</td>
</tr>
<tr>
<td>directory</td>
<td>Optional</td>
<td>1</td>
<td>The directory to process. The recursive parameter determines how the directory will be processed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> At least one file or directory path is required.</td>
<td>-R or -D</td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>drivepool</td>
<td>Optional</td>
<td>1</td>
<td>Media Manager drive pool group used to store the file specified.</td>
<td>-v</td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>N</td>
<td>File that needs to be stored.</td>
<td>file name</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>mediatype</td>
<td>Optional</td>
<td>1</td>
<td>Defines the type of medium to be used for storage.</td>
<td>-t</td>
</tr>
<tr>
<td>minsize</td>
<td>Optional</td>
<td>1</td>
<td>Minimum File Size in bytes to be stored.</td>
<td>-z</td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>recursive</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is only valid when the directory parameter is specified. A value of false will only process the contents of the directory while a value of true will perform a recursive operation on the directory. The default value is true.</td>
<td>-R or -D</td>
</tr>
<tr>
<td>runtime</td>
<td>Optional</td>
<td>1</td>
<td>Maximum allowable time in hours for the command to finish.</td>
<td>-u</td>
</tr>
<tr>
<td>trunc</td>
<td>Optional</td>
<td>1</td>
<td>The truncation mode. Valid value are immediate and policy.</td>
<td>-f</td>
</tr>
</tbody>
</table>

### Example

```plaintext
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>
    </status>
  </statuses>
</fsstore>
<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>
$statusText>File /stornext/snfx1/smp2data/filex1 copy 1 has been stored.<$/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>
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 `fsrmdiskcopy` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>directory</td>
<td>Optional</td>
<td>1</td>
<td>The directory to process. The recursive parameter determines how the directory will be processed.</td>
<td>-R or -D</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>N</td>
<td>One or more files to remove from disk.</td>
<td>file name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-----</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast. <strong>Note:</strong> The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>recursive</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is only valid when the directory parameter is specified. A value of false will only process the contents of the directory while a value of true will perform a recursive operation on the directory. The default value is true.</td>
<td>-R or -D</td>
</tr>
</tbody>
</table>

**Example**

```
https://<<SERVER>>/sws/v2/file/fsrmdiskcopy
?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>
  </header>
</fsrmdiskcopy>
```
<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>
File / Modify Class Attributes

Modify the class attributes of a file.
This web service runs the `fschfiat` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>clean</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates if the database entries are to be cleaned when the file is removed from the file system. Valid values are <code>true</code> and <code>false</code>. The <code>true</code> argument indicates that the database entries will be cleaned and the file will NOT be recoverable by the <code>fsrecover</code> command. The <code>false</code> argument indicates that the database entries will NOT be cleaned and the file will be recoverable by the <code>fsrecover</code> command.</td>
<td>-C</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| directory    | Optional  | 1   | The directory to process. The recursive parameter determines how the directory will be processed.  
|              |           |     | Note: At least one file or directory path is required.                     | -R or -D   |
| file         | Optional  | N   | One or more file(s) having the attributes changed.                          | file name  |
|              |           |     | Note: At least one file or directory path is required.                      |            |
| format       | Optional  | 1   | The output format requested. Valid values are xml, json and text. The default is text. | -F         |
| 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 fast and slow. The default value is fast.  
<p>|              |           |     | Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues. | None       |
| mode         | Optional  | 1   | This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync. | None       |
| 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         |
| recursive    | Optional  | 1   | This parameter is only valid when the directory parameter is specified. A value of false will only process the contents of the directory while a value of true will perform a recursive operation on the directory. The default value is true. | -R or -D   |
| reloc        | Optional  | 1   | This option indicates how relocation policies operate on the file. Valid values are exclude and policy. The exclude argument excludes the file from relocation when a relocation policy occurs. The policy argument relocates the file by relocation policy. | -r         |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>store</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates how storage policies operate on the file. Valid values are <code>exclude</code> and <code>policy</code>. The <code>exclude</code> argument excludes the file from storage when a store policy occurs. The <code>policy</code> argument stores the file by storage policy.</td>
<td>-s</td>
</tr>
<tr>
<td>stubsize</td>
<td>Optional</td>
<td>1</td>
<td>The truncation stub size (in kilobytes). This value is used to determine the number of bytes to leave on disk when files are truncated.</td>
<td>-S</td>
</tr>
<tr>
<td>trunc</td>
<td>Optional</td>
<td>1</td>
<td>This option indicates how truncation policies operate on the file. Valid values are <code>exclude</code>, <code>policy</code>, <code>immediate</code> and <code>clear</code>. The <code>exclude</code> argument excludes the file from truncation when a store and/or cleanup policy application occurs. The <code>immediate</code> argument truncates the file immediately when stored to a medium. The <code>policy</code> argument truncates the file by cleanup policy. The clear argument temporarily clears the indication that this file met truncate exclusion criteria defined in the <code>excludes.truncate</code> file.</td>
<td>-t</td>
</tr>
</tbody>
</table>

**Example**

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

**Output**

```xml
<?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>
```

StorNext 6 Web Services Guide 65
<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>
Chapter 1: StorNext Web Services Commands (V2)

File

How to Find the async Job ID in the Output

Below is an example command that initiates the async request to the StorNext MDC over HTTP or HTTPS:

```bash
```
Below is an example of the output:

```json
{
  "job" : "23",
  "message" : "Job has been successfully submitted."
}
```

Below is an example command that queries for the status for the job number:

```
# /opt/quantum/curl/bin/curl "http://$MDC:81/sws/v2/job/info?job=23"
```

Below is an example of the output:

```json
{
  "jobList": [ 
    {
      "jobInfo": {
        "job": "17",
        "state": "COMPLETED",
        "exitcode": 0,
        "datecompleted": "2018-10-30 19:13:05"
      },
      "response": {
        "header": {
          "commandName": "fsretrieve",
          "commandLine": "/usr/adic/TSM/bin/fsretrieve -F JSON /stornext/snfs1/foobar",
          "commandDescription": "Retrieve files from media and place on disk",
          "localDateISO": "2018-10-30T19:12:51",
          "localDate": "2018-10-30",
          "localTime": "19:12:51",
          "localDayOfWeek": 2,
          "gmtDateISO": "2018-10-30T23:12:51Z",
          "gmtDate": "2018-10-30",
          "gmtTime": "23:12:51",
          "gmtDayOfWeek": 2
        },
        "statuses": [ 
          {
            "statusCode": "FS0589",
            "statusNumber": 589,
```
"dayOfMonth": 30,
"requestId": 248266,
"commandName": "/usr/adic/TSM/bin/fsretrieve",
"commandStatus": "interim",
"statusText": "Tertiary Manager software request received."
},
{
 "statusCode": "FS0347",
 "statusNumber": 347,
 "dayOfMonth": 30,
 "requestId": 248266,
 "commandName": "/usr/adic/TSM/bin/fsretrieve",
 "commandStatus": "interim",
 "statusText": "File /stornext/snfs1/foobar has been retrieved."
},
{
 "statusCode": "FS0390",
 "statusNumber": 390,
 "dayOfMonth": 30,
 "requestId": 248266,
 "commandName": "/usr/adic/TSM/bin/fsretrieve",
 "commandStatus": "completed",
 "statusText": "1 out of 1 retrieves were successful."
}]
,"footer": {
 "returnCode": 0,
 "localDateISOEnd": "2018-10-30T19:13:05",
 "localDateEnd": "2018-10-30",
 "localTimeEnd": "19:13:05",
 "localDayOfWeekEnd": 2,
 "gmtDateISOEnd": "2018-10-30T23:13:05Z",
 "gmtDateEnd": "2018-10-30",
 "gmtTimeEnd": "23:13:05",
 "gmtDayOfWeekEnd": 2,
 "elapsedTimeInSeconds": "13.0942"
}
Chapter 1: StorNext Web Services Commands (V2)

Job

Job / Asynchronous Job Status

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

Parameters

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

Example

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

Output

```json
{
  "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 \texttt{fsstore} and \texttt{fsretrieve} jobs.

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>job</td>
<td>Required</td>
<td>N</td>
<td>The list of job ids for which status is being queried.</td>
<td>None</td>
</tr>
</tbody>
</table>

### 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",
"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",
"commandName": "fsretrieve",
"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",
"commandName": "fsretrieve",
"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",
"commandName": "fsretrieve",
"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",
"commandName": "fsretrieve",
"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",
"commandName": "fsretrieve",
"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,
"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
Chapter 1: StorNext Web Services Commands (V2)

Job

/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." }

] ,
Chapter 1: StorNext Web Services Commands (V2)

Job

```
"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."
Media

Media / Information

Generate a report on media based on their current status.

This web service runs the `fsmedinfo` command.
## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-F</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>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 <code>false</code>.</td>
<td>-l</td>
</tr>
<tr>
<td>starttime</td>
<td>Optional</td>
<td>1</td>
<td>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 <code>-e</code> is used, the output will be sorted per file system by the store/recovery time. The format for the time parameter is <code>YYYY:MM:DD:hh:mm:ss</code>.</td>
<td>-s</td>
</tr>
<tr>
<td>endtime</td>
<td>Optional</td>
<td>1</td>
<td>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 <code>YYYY:MM:DD:hh:mm:ss</code>.</td>
<td>-e</td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>A media ID.</td>
<td>mediaID</td>
</tr>
</tbody>
</table>

### Note
The `fast` and `slow` queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

- F
- l
- s
- e

### Example

https://<<SERVER>>/sws/v2/media/fsmedinfo
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 E00002</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>
    </media>
  </medias>
</fsmedinfo>
<classId>sws_1400082516</classId>
<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Media

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default value is <strong>text</strong>.</td>
<td>- F</td>
</tr>
<tr>
<td>requestid</td>
<td>Optional</td>
<td>1</td>
<td>The request id for the deletion requests. If no requestid is provided, all outstanding requests are included.</td>
<td>- C</td>
</tr>
</tbody>
</table>

Example

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

Output

```xml
<?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>
</header>
```
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td><code>- F</code></td>
</tr>
<tr>
<td>fsmounts</td>
<td>Optional</td>
<td>N</td>
<td>One or more file systems for which to clean.</td>
<td><code>- s</code></td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>1</td>
<td>Policy class associated with data to clean.</td>
<td><code>- c</code></td>
</tr>
<tr>
<td>endtime</td>
<td>Optional</td>
<td>1</td>
<td>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 <code>YYYY:MM:DD:hh:mm:ss</code>.</td>
<td><code>- t</code></td>
</tr>
<tr>
<td>removedonly</td>
<td>Optional</td>
<td>1</td>
<td>Process inactive versions associated with removed files. Default value is false.</td>
<td><code>- d</code></td>
</tr>
<tr>
<td>purge</td>
<td>Optional</td>
<td>1</td>
<td>This option is used to indicate that a purge of the database namespace will be performed. Default value is false.</td>
<td><code>- P</code></td>
</tr>
<tr>
<td>media</td>
<td>Optional</td>
<td>N</td>
<td>A media ID.</td>
<td><code>mediaID</code></td>
</tr>
</tbody>
</table>

## Example

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

Output

```xml
<?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>
  </statuses>
</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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>rminfomedia</td>
<td>Required</td>
<td>1</td>
<td>Media for which file version removal is required.</td>
<td>media ID</td>
</tr>
<tr>
<td>rminforetrieve</td>
<td>Optional</td>
<td>1</td>
<td>The filename file contains a list of files which need to be retrieved so that their missing copies can be regenerated.</td>
<td>-p</td>
</tr>
<tr>
<td>purge</td>
<td>Optional</td>
<td>1</td>
<td>This option is used to indicate that a purge of the database namespace will be performed. Default value is false.</td>
<td>-P</td>
</tr>
</tbody>
</table>

Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

Example

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

```xml
<?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>
        </status>
    </statuses>
</fsclean>
```
<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

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

Example

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

Output

```<?xml version="1.0" encoding="UTF-8" ?>
<fs clean 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>
</fs clean>```
Chapter 1: StorNext Web Services Commands (V2)
Media

<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>
</footer>
Media / Move Media

Move media from one archive to another.
This web service runs the `vsmove` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>The media(s) that need(s) to be moved.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-----</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>archive</td>
<td>Required</td>
<td>1</td>
<td>Specifies the name of the archive to which the specified media are to be moved.</td>
</tr>
<tr>
<td>operator</td>
<td>Optional</td>
<td>1</td>
<td>Indicates this command is to be processed only if both the source and destination archives are operator-attended. Default value is false.</td>
</tr>
<tr>
<td>wait</td>
<td>Optional</td>
<td>1</td>
<td>Indicates the Media Manager software waits until the command processing completes before returning status to the client. Default value is false.</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>Indicates that verbose output is desired. Default value is false.</td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>priority</td>
<td>Optional</td>
<td>1</td>
<td>The execution priority of the entered command. Assignable priority values are restricted to a range from 1 (highest) to 32 (lowest) inclusive. The default priority value is 15.</td>
</tr>
<tr>
<td>retries</td>
<td>Optional</td>
<td>1</td>
<td>The number of retries the CLI software attempts if a timeout is returned by the API software. The default retries value is 3.</td>
</tr>
<tr>
<td>timeout</td>
<td>Optional</td>
<td>1</td>
<td>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 120 seconds.</td>
</tr>
<tr>
<td>rpcnumber</td>
<td>Optional</td>
<td>1</td>
<td>The RPC program number for the Media Manager software. The default value for the Media Manager software program number is 300016.</td>
</tr>
</tbody>
</table>

**Example**

https://<<SERVER>>/sws/v2/media/vsmove
?media=E00008
&archive=myarchive
Chapter 1: StorNext Web Services Commands (V2)

Media

&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>
Media / Import Cleaning Media

This web service imports a list of cleaning media.

Parameters

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

**Note:** The **fast** and **slow** queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req/Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are <strong>json</strong> and <strong>text</strong>. The default value is <strong>text</strong>.</td>
<td>-F</td>
</tr>
<tr>
<td>mediaid</td>
<td>Required</td>
<td>N</td>
<td>The list of media IDs.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Example**

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

**Output**

```json
{
   "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
   }
}
```
Chapter 1: StorNext Web Services Commands (V2)

Media

},
"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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>. Note: The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are <strong>json</strong> and <strong>text</strong>. The default value is <strong>text</strong>.</td>
<td>- F</td>
</tr>
<tr>
<td>mediaid</td>
<td>Required</td>
<td>N</td>
<td>The list of media IDs.</td>
<td>None</td>
</tr>
<tr>
<td>mediaformat</td>
<td>Optional</td>
<td>1</td>
<td>The media format that is requested for the media. The valid values are <strong>ANTF</strong> and <strong>LTFS</strong>.</td>
<td>- T</td>
</tr>
<tr>
<td>policyclass</td>
<td>Optional</td>
<td>1</td>
<td>The policy class this media will be associated.</td>
<td>- c</td>
</tr>
<tr>
<td>withholdformatting</td>
<td>Optional</td>
<td>1</td>
<td>This option is used to withhold formatting media immediately.</td>
<td>- w</td>
</tr>
</tbody>
</table>

Example

https://<<SERVER>>/sws/v2/media/fsmedin
?mediaid=000017
Output

```json
{
   "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."
   },
   "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"
Media / Export Media

This web service exports a list of media.

Parameters

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

Example

https://<<SERVER>>/sws/v2/media/fsmedout
?mediaid=000017
Chapter 1: StorNext Web Services Commands (V2)

Media

&format=json

Output

```json
{
    "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
    }
}
```
Media / Copy Media

Initiates copy-replace for a media.

This web service runs the `fsmedcopy` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>includeinactive</td>
<td>Optional</td>
<td>1</td>
<td>Indicates all files, active and inactive versions will be copied from the source medium. Default value is <code>false</code>.</td>
<td><code>-a</code></td>
</tr>
<tr>
<td>blank</td>
<td>Optional</td>
<td>1</td>
<td>Specify blank media for media destination. Default value is <code>false</code>.</td>
<td><code>-b</code></td>
</tr>
<tr>
<td>destination</td>
<td>Optional</td>
<td>1</td>
<td>Copy data file(s) to destination media.</td>
<td><code>-d</code></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td><code>-F</code></td>
</tr>
<tr>
<td>checksum</td>
<td>Optional</td>
<td>1</td>
<td>Generate and maintain a checksum for each copied file. Valid values are <code>y</code> and <code>n</code>.</td>
<td><code>-G y or n</code></td>
</tr>
<tr>
<td>limit</td>
<td>Optional</td>
<td>1</td>
<td>Use this option to throttle and limit the number of concurrent streams used for the request.</td>
<td><code>-L</code></td>
</tr>
</tbody>
</table>

**Note:** This option is useful with object storage media.

The default value is **10**.
## Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>recreate</td>
<td>Optional</td>
<td>1</td>
<td>Use this option when issues exist with a medium such that files cannot be read from it. Files on the medium can be recreated from alternate media which have different copies of the files. The argument specifies which copy to use. If 0 is specified, then the command uses any copy to recreate copies of files for the bad medium. Valid values are 0 through 8.</td>
<td>-R</td>
</tr>
<tr>
<td>mediatype</td>
<td>Optional</td>
<td>1</td>
<td>Defines the type of medium to be used.</td>
<td>-t</td>
</tr>
<tr>
<td>runtime</td>
<td>Optional</td>
<td>1</td>
<td>Maximum allowable time in hours for the command to finish.</td>
<td>-u</td>
</tr>
<tr>
<td>drivepool</td>
<td>Optional</td>
<td>1</td>
<td>Media Manager drive pool group from which the destination drive will be selected when copying the specified media.</td>
<td>-v</td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>Media IDs of the media(s).</td>
<td>media ID</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
</tbody>
</table>

### Example

https://<<SERVER>>/sws/v2/media/fsmedcopy/replace
Chapter 1: StorNext Web Services Commands (V2)

Media

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>
</statuses>
</fsmedcopy>
Medial / Fragmentation Report

Report on media fragmentation.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default is <strong>text</strong>.</td>
<td>-F</td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>The media IDs for which we need the fragmentation report.</td>
<td>media ID</td>
</tr>
<tr>
<td>fill</td>
<td>Optional</td>
<td>1</td>
<td>Fill level threshold between 0 and 100 percent. The percentage of the medium that has been written, including active and inactive file versions. A default of 0 percent is used if not specified.</td>
<td>-f</td>
</tr>
<tr>
<td>fragmentation</td>
<td>Optional</td>
<td>1</td>
<td>The percent (0 – 100) of wasted space out of the filled space on a medium. A default of 0 (zero) percent is used if not specified.</td>
<td>-w</td>
</tr>
</tbody>
</table>

### Example

```
https://<<SERVER>>/sws/v2/media/fsmedcopy/fragreport
?media=E00001
&format=xml
```
Chapter 1: StorNext Web Services Commands (V2)

Media

Output

```xml
<?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>
    </status>
  </statuses>
</fsmedcopy>
```
Media / Eject Media

Eject media out of an archive to be entered into another archive.
This web service runs the `vsarchiveeject` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <em>sync</em> and <em>async</em>. The default value is <em>sync</em>.</td>
<td>None</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### Media

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>Media IDs that need to be ejected.</td>
<td>mediaID</td>
</tr>
<tr>
<td>port</td>
<td>Optional</td>
<td>1</td>
<td>Specifies the import/export port the media is ejected to (if applicable).</td>
<td>-s</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>Indicates that verbose output is desired. Default value is false.</td>
<td>-v</td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>The host name of the Media Manager server.</td>
<td>-H</td>
</tr>
<tr>
<td>priority</td>
<td>Optional</td>
<td>1</td>
<td>The execution priority of the entered command. Assignable priority values are restricted to a range from 1 (highest) to 32 (lowest) inclusive. The default priority value is 15.</td>
<td>-P</td>
</tr>
<tr>
<td>retries</td>
<td>Optional</td>
<td>1</td>
<td>The number of retries that are attempted if a timeout is returned by the API software. The default retries value is 3.</td>
<td>-R</td>
</tr>
<tr>
<td>timeout</td>
<td>Optional</td>
<td>1</td>
<td>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 120 seconds.</td>
<td>-T</td>
</tr>
<tr>
<td>rpcnumber</td>
<td>Optional</td>
<td>1</td>
<td>The RPC program number for the Media Manager software. The default value for the Media Manager software program number is 300016.</td>
<td>-V</td>
</tr>
</tbody>
</table>
Example

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

Output

```xml
<?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>
        </errorMessage>
    </vsarchiveejectOutput>
</vsarchiveeject>
```
<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>
</vsarchiveeject>

## Media / Enter Media

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

This web service runs the `vsarchiveenter` command.
# Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code></td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td><code>-F</code></td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>Media IDs that need to be moved.</td>
<td><code>mediaID</code></td>
</tr>
<tr>
<td>port</td>
<td>Optional</td>
<td>1</td>
<td>Specifies the import/export port the media is entered from (if applicable).</td>
<td><code>-s</code></td>
</tr>
<tr>
<td>archive</td>
<td>Required</td>
<td>1</td>
<td>Specifies the name of the archive to which the specified media are to be moved.</td>
<td><code>-a</code></td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>Indicates that verbose output is desired.</td>
<td><code>-v</code></td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>The host name of the Media Manager server.</td>
<td><code>-H</code></td>
</tr>
<tr>
<td>priority</td>
<td>Optional</td>
<td>1</td>
<td>The execution priority of the entered command. Assignable priority values are restricted to a range from 1 (highest) to 32 (lowest) inclusive. The default priority value is 15.</td>
<td><code>-P</code></td>
</tr>
<tr>
<td>retries</td>
<td>Optional</td>
<td>1</td>
<td>The number of retries that are attempted if a timeout is returned by the API software. The default retries value is 3.</td>
<td><code>-R</code></td>
</tr>
<tr>
<td>timeout</td>
<td>Optional</td>
<td>1</td>
<td>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 120 seconds.</td>
<td><code>-T</code></td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpcnumber</td>
<td>Optional</td>
<td>1</td>
<td>The RPC program number for the Media Manager software. The default value for the Media Manager software program number is <strong>300016</strong>.</td>
</tr>
</tbody>
</table>

### 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>
</vsarchiveenter>
```
Media / Purge Media

This web service purges a list of media.

Parameters

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

Example

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

Output

```json
{
    "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"
Chapter 1: StorNext Web Services Commands (V2)

Media

Media / Attribute Query

Queries for the attributes of one or more specified media. This web service runs the `vsmedqry` command.

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

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>Specifies a media to be queried.</td>
<td>mediaID</td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>The host name of the Media Manager server.</td>
<td>-H</td>
</tr>
<tr>
<td>priority</td>
<td>Optional</td>
<td>1</td>
<td>The execution priority of the entered command. Assignable priority values are restricted to a range from 1 (highest) to 32 (lowest) inclusive. The default priority value is 15.</td>
<td>-P</td>
</tr>
<tr>
<td>retries</td>
<td>Optional</td>
<td>1</td>
<td>The number of retries that are attempted if a timeout is returned by the API software. The default retries value is 3.</td>
<td>-R</td>
</tr>
<tr>
<td>timeout</td>
<td>Optional</td>
<td>1</td>
<td>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 120 seconds.</td>
<td>-T</td>
</tr>
<tr>
<td>rpcnumber</td>
<td>Optional</td>
<td>1</td>
<td>The RPC program number for the Media Manager software. The default value for the Media Manager software program number is 300016.</td>
<td>-V</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td><code>-F</code></td>
</tr>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>One or more 16-character media identifiers.</td>
<td><code>mediaID</code></td>
</tr>
</tbody>
</table>
| state     | Optional  | 1   | Media state. Valid values for states are as follows:  
  - `unsusp`: Reset error count associated with media to 0.  
  - `protect`: Mark media as write-protected. Files can be read, but no data can be written to the media.  
  - `unprotect`: Mark the media as unprotected for data storage on media.  
  - `avail`: Media are available for storage and retrieval.  
  - `unavail`: Media are unavailable to Tertiary Manager software.  
  - `unmark`: Unmark media that are marked Error, or Check out.  
  Either the state or policy is required. | `-s` |
| policy    | Optional  | 1   | Change the policy class name of blank media. Either the state or policy is required. | `-c` |
| blank     | Optional  | 1   | Change policy class for a blank medium to system blank pool. Default value is `false`. | `-b` |
Example

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

Output

```xml
<?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>
    </status>
  </statuses>
</fschmedstate>
```
**Media / Checkout Media**

Check media out of the Media Manager system.
This web service runs the `vscheckout` command.
Note: This web service returns text output.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>media</td>
<td>Required</td>
<td>N</td>
<td>Specifies a list of one or more media to be queried.</td>
<td>mediaID</td>
</tr>
<tr>
<td>comment</td>
<td>Optional</td>
<td>1</td>
<td>Provide a comment to be associated with each checked-out media.</td>
<td>-t</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>Indicates that verbose output is needed. Default value is false.</td>
<td>-v</td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>The host name of the Media Manager server.</td>
<td>-H</td>
</tr>
<tr>
<td>priority</td>
<td>Optional</td>
<td>1</td>
<td>The execution priority of the entered command. Assignable priority values are restricted to a range from 1 (highest) to 32 (lowest) inclusive. The default priority value is 15.</td>
<td>-P</td>
</tr>
<tr>
<td>retries</td>
<td>Optional</td>
<td>1</td>
<td>The number of retries that are attempted if a timeout is returned by the API software. The default retries value is 3.</td>
<td>-R</td>
</tr>
<tr>
<td>timeout</td>
<td>Optional</td>
<td>1</td>
<td>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 120 seconds.</td>
<td>-T</td>
</tr>
<tr>
<td>rpcnumber</td>
<td>Optional</td>
<td>1</td>
<td>The RPC program number for the Media Manager software. The default value for the Media Manager software program number is 300016.</td>
<td>-V</td>
</tr>
</tbody>
</table>

Example

https://<<SERVER>>/sws/v2/media/vscheckout
?media=E00001
Chapter 1: StorNext Web Services Commands (V2)

Media

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default is <strong>text</strong>.</td>
<td>-F</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>N</td>
<td>Policy class(es) for which the report is to be generated.</td>
<td>-c</td>
</tr>
<tr>
<td>blank</td>
<td>Optional</td>
<td>1</td>
<td>Used to report on the blank media in the general scratch pool. Default value is <strong>false</strong>.</td>
<td>-g</td>
</tr>
<tr>
<td>checkout</td>
<td>Optional</td>
<td>1</td>
<td>List all media marked for check out. Default value is <strong>false</strong>.</td>
<td>-lk</td>
</tr>
</tbody>
</table>

Example

https://<<SERVER>>/sws/v2/media/fsmedlist/removal

?policy=myaxrpol1
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>
    </status>
  </statuses>
</fsmedlist>
Media / List Media by State

Lists media by state.

This web service runs the fsmedlist command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>-----</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>N</td>
<td>Policy class(es) for which the report is to be generated.</td>
<td>-c</td>
</tr>
<tr>
<td>blank</td>
<td>Optional</td>
<td>1</td>
<td>Used to report on the blank media in the general scratch pool. Default value is false.</td>
<td>-g</td>
</tr>
<tr>
<td>unformatted</td>
<td>Optional</td>
<td>1</td>
<td>List unformatted blank media. Default value is false.</td>
<td>-lb</td>
</tr>
<tr>
<td>formatted</td>
<td>Optional</td>
<td>1</td>
<td>List formatted blank media. Default value is false.</td>
<td>-lf</td>
</tr>
<tr>
<td>protect</td>
<td>Optional</td>
<td>1</td>
<td>List all write protected media. Default value is false.</td>
<td>-lp</td>
</tr>
<tr>
<td>error</td>
<td>Optional</td>
<td>1</td>
<td>List media with mark error. Default value is false.</td>
<td>-lq</td>
</tr>
<tr>
<td>available</td>
<td>Optional</td>
<td>1</td>
<td>List all available media. Default value is false.</td>
<td>-la</td>
</tr>
<tr>
<td>inaccessible</td>
<td>Optional</td>
<td>1</td>
<td>List all media unavailable to Tertiary Manager software. Default value is false.</td>
<td>-ln</td>
</tr>
<tr>
<td>unavailable</td>
<td>Optional</td>
<td>1</td>
<td>List all media marked as unavailable, but still located within the Tertiary Manager system. Default value is false.</td>
<td>-lu</td>
</tr>
<tr>
<td>suspect</td>
<td>Optional</td>
<td>1</td>
<td>List all suspect media. Default value is false.</td>
<td>-ls</td>
</tr>
</tbody>
</table>
Example

https://<<SERVER>>/sws/v2/media/fsmedlist/state
?policy=myaxrpol1
&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>
</class>
</classes>
</header>
<inDrive>
<total>0</total>
</inDrive>
<inBin>
<total>0</total>
</inBin>
<exiting>
<total>0</total>
</exiting>
<outOfStorNext>
<total>0</total>
</outOfStorNext>
<markedForCheckOut>
<total>0</total>
</markedForCheckOut>
<markError>
<total>0</total>
</markError>
Chapter 1: StorNext Web Services Commands (V2)

Media

</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>
</footer>
</localDateISOEnd>
2014-07-01T16:05:08
</localDateISOEnd>
</localDateEnd>
Media / List Media by Location

Lists media by location.

This web service runs the `fsmedlist` command.

Parameters

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>N</td>
<td>Policy class(es) for which the report is to be generated.</td>
<td>-c</td>
</tr>
<tr>
<td>blank</td>
<td>Optional</td>
<td>1</td>
<td>Used to report on the blank media in the general scratch pool. Default value is false.</td>
<td>-g</td>
</tr>
<tr>
<td>drive</td>
<td>Optional</td>
<td>1</td>
<td>List all media located in a drive. Default value is false.</td>
<td>-ld</td>
</tr>
<tr>
<td>home</td>
<td>Optional</td>
<td>1</td>
<td>List all media located in their home slot/bin. Default value is false.</td>
<td>-lh</td>
</tr>
<tr>
<td>exit</td>
<td>Optional</td>
<td>1</td>
<td>List all media exiting a storage area. Default value is false.</td>
<td>-lz</td>
</tr>
<tr>
<td>checkout</td>
<td>Optional</td>
<td>1</td>
<td>List all media that are checked out of the storage areas. Default value is false.</td>
<td>-lo</td>
</tr>
</tbody>
</table>

### Example

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

### Output

```
<?xml version="1.0" encoding="UTF-8" ?>
```
Chapter 1: StorNext Web Services Commands (V2)
Media

<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>
    </class>
  </classes>
</fsmedlist>
Chapter 1: StorNext Web Services Commands (V2)

Media

```
</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>
```
Chapter 1: StorNext Web Services Commands (V2)
Object Storage

Object Storage

Report Object Storage Components

Report all Object Storage components in Quantum storage system.
This web service runs the \texttt{fsobjcfg} command.
Parameters

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

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>
  </header>
</fsobjcfg>
<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>
</appliance>
</configurationReport>
<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>
</iopath>
</iopaths>
</node>
<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>
</status>
</statuses>
Policy

Policy / Class Information

This web service reports information on storage manager policies.

This web service runs the `fsclassinfo` command.
## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default is <strong>text</strong>.</td>
<td>- F</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>N</td>
<td>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.</td>
<td>policy name</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>List information in the long report format. Default value is <strong>false</strong>.</td>
<td>- l</td>
</tr>
</tbody>
</table>

## 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>
Policy / Modify a Policy

Modify the processing parameters of a policy class.
This web service runs the `fsmodclass` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>Optional</td>
<td>1</td>
<td>Up to five characters can be used for the account number.</td>
<td>-o</td>
</tr>
<tr>
<td>affinity</td>
<td>Optional</td>
<td>1</td>
<td>A space-separated list of disk affinities that the files in this policy class will traverse throughout their life cycle.</td>
<td>-a</td>
</tr>
<tr>
<td>autostore</td>
<td>Optional</td>
<td>1</td>
<td>This option decides if we allow the policy engine to automatically store files for this policy class. Valid values are <strong>true</strong> and <strong>false</strong>.</td>
<td>-p</td>
</tr>
<tr>
<td>blankpool</td>
<td>Optional</td>
<td>1</td>
<td>Media classification cleanup action. Valid values are policy and system.</td>
<td>-r</td>
</tr>
<tr>
<td>checksumretrieve</td>
<td>Optional</td>
<td>1</td>
<td>Verify the checksum of each retrieved file. Valid values are <strong>true</strong> and <strong>false</strong>.</td>
<td>-V</td>
</tr>
</tbody>
</table>

**Note:** If you enable checksum, then it might not be possible to keep the drives with compression enabled, running at their highest rate due to the overhead of generating the checksum.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksumstore</td>
<td>Optional</td>
<td>1</td>
<td>Generate and maintain a checksum for each stored file. Valid values are true and false.</td>
<td>-G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> If you enable checksum, then it might not be possible to keep the drives with compression enabled, running at their highest rate due to the overhead of generating the checksum.</td>
<td></td>
</tr>
<tr>
<td>copy</td>
<td>Optional</td>
<td>1</td>
<td>Used to apply the copy options to a specific copy.</td>
<td>-C</td>
</tr>
<tr>
<td>dbclean</td>
<td>Optional</td>
<td>1</td>
<td>Remove database information when a file is removed. Valid values are true and false.</td>
<td>-D</td>
</tr>
<tr>
<td>defaultcopies</td>
<td>Optional</td>
<td>1</td>
<td>The total number of copies that will be stored (including the primary copy) for each file in this policy class.</td>
<td>-d</td>
</tr>
<tr>
<td>drivelimit</td>
<td>Optional</td>
<td>1</td>
<td>The maximum number of drives to use when the policy is run.</td>
<td>-L</td>
</tr>
<tr>
<td>drivepool</td>
<td>Optional</td>
<td>1</td>
<td>The Media Manager drive pool group used to store or retrieve data for this policy class.</td>
<td>-v</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### Policy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>expirationtime</td>
<td>Optional</td>
<td>1</td>
<td>The minimum time per copy that a stored file must reside without being accessed before the copy becomes eligible for deletion. The default value of 0 (zero) indicates that the copy is not eligible for expiration. Expiration time can be set on any copy, but at least one copy must be configured as ineligible for expiration. The default unit for the copy expiration value is days.</td>
<td>-E</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Examples:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- A value of <strong>15m</strong> equates to 15 minutes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- A value of <strong>3h</strong> equates to 3 hours.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- A value of <strong>180d</strong> equates to 180 days.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- A value of <strong>10</strong> equates to 10 days.</td>
<td></td>
</tr>
<tr>
<td>forceallcopiesexpire</td>
<td>Optional</td>
<td>1</td>
<td>Boolean to override the restriction where the user is not allowed to configure a copy expiration time on all configured copies. The expiration option must also be specified with an expiration time greater than 0. When the last copy of a file expires and is removed, the file is automatically removed from the file-system name space and is no longer recoverable.</td>
<td>-X</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>hardlimit</td>
<td>Optional</td>
<td>1</td>
<td>The maximum number of media that are allowed in this policy class.</td>
<td>-h</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow.  The default value is fast. Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>maxcopies</td>
<td>Optional</td>
<td>1</td>
<td>The maximum number of copies (including the primary copy) that are allowed for each file in this policy class.</td>
<td>-x</td>
</tr>
<tr>
<td>maxsetage</td>
<td>Optional</td>
<td>1</td>
<td>Candidate expiration time (in hours) of the policy class.</td>
<td>-g</td>
</tr>
<tr>
<td>maxversions</td>
<td>Optional</td>
<td>1</td>
<td>This is the maximum number of inactive versions to keep for a file (the current version is active, all others are inactive).</td>
<td>-k</td>
</tr>
<tr>
<td>mediaformattype</td>
<td>Optional</td>
<td>1</td>
<td>Specifies the media format type that will be used when formatting or selecting media.</td>
<td>-T</td>
</tr>
<tr>
<td>mediatype</td>
<td>Optional</td>
<td>1</td>
<td>Defines the type of medium to be used.</td>
<td>-t</td>
</tr>
<tr>
<td>minreloctime</td>
<td>Optional</td>
<td>1</td>
<td>The minimum time that a stored file must reside unaccessed on disk before being considered a candidate for relocation (the clearing of disk blocks).</td>
<td>-i</td>
</tr>
<tr>
<td>minsetsize</td>
<td>Optional</td>
<td>1</td>
<td>The minimum set size of the policy class.</td>
<td>-z</td>
</tr>
<tr>
<td>minstoretime</td>
<td>Optional</td>
<td>1</td>
<td>The minimum time that a file must reside unmodified on disk before being considered a candidate for storage on media.</td>
<td>-m</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>mintrunctime</td>
<td>Optional</td>
<td>1</td>
<td>The minimum time that a stored file must reside unaccessed on disk before being considered a candidate for truncation (the clearing of disk blocks).</td>
<td>-c</td>
</tr>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>policy</td>
<td>Required</td>
<td>1</td>
<td>The policy class for which the report is to be generated.</td>
<td>policy class</td>
</tr>
<tr>
<td>restoreonreference</td>
<td>Optional</td>
<td>1</td>
<td>Boolean to indicate if copies are to be recreated when referenced after being expired. The default value of false specifies that the copy is not to be recreated. This option is not allowed with the forceallcopiesexpire option or when the expirationtime is set to 0.</td>
<td>-P</td>
</tr>
<tr>
<td>retrieveaffinity</td>
<td>Optional</td>
<td>1</td>
<td>The affinity to retrieve a truncated file to.</td>
<td>-R</td>
</tr>
<tr>
<td>retrieveorder</td>
<td>Optional</td>
<td>1</td>
<td>A comma-separated list of copy numbers specifying the order in which copies will be selected when retrieving files. Specifying none will re-enable the default retrieve order. If this option is not specified, the default retrieve order is used.</td>
<td>-O</td>
</tr>
<tr>
<td>security</td>
<td>Optional</td>
<td>1</td>
<td>Up to four characters can be used for security code.</td>
<td>-l</td>
</tr>
<tr>
<td>softlimit</td>
<td>Optional</td>
<td>1</td>
<td>The warning limit for the number of media that can be allocated to this policy class.</td>
<td>-s</td>
</tr>
<tr>
<td>stubsize</td>
<td>Optional</td>
<td>1</td>
<td>The truncation stub size (in kilobytes).</td>
<td>-S</td>
</tr>
<tr>
<td>truncmode</td>
<td>Optional</td>
<td>1</td>
<td>Truncation mode. Valid values are immediate and policy.</td>
<td>-f</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)
Policy

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>
    </status>
  </statuses>
</fsmodclass>
Policy / Report Policy Class For A Directory

This web service reports the policy class associated with directory.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Policy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>. <strong>Note:</strong> The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are <strong>json</strong> and <strong>text</strong>. The default value is <strong>text</strong>.</td>
<td>-F</td>
</tr>
<tr>
<td>directory</td>
<td>Required</td>
<td>1</td>
<td>The directory for which the policy class is required.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Example**

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

**Output**

```json
{
    "header": {
        "commandName": "fsdirclass",
        "commandLine": "/usr/adic/TSM/bin/fsdirclass /stornext/snfx1/smp1data -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",
```
Policy / Report Policy Class For A File System

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

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

#### Example

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

#### Output

```json
{
    "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"
  },
Quota

Quota / Manage Quotas

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

This web service runs the `snquota` command.
## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-o</td>
</tr>
<tr>
<td>fsname</td>
<td>Required</td>
<td>1</td>
<td>Specify <code>FileSystemName</code> as the file system to manipulate.</td>
<td>-F</td>
</tr>
<tr>
<td>path</td>
<td>Optional</td>
<td>1</td>
<td>Specify the file system containing <code>Path</code> as the file system to manipulate.</td>
<td>-p</td>
</tr>
<tr>
<td>action</td>
<td>Required</td>
<td>1</td>
<td>Specify the action. Valid values are <code>create</code>, <code>delete</code>, <code>list</code>, <code>listall</code>, <code>mark</code>, <code>rebuild</code>, and <code>set</code>.</td>
<td><code>create = -C</code>&lt;br&gt;<code>delete = -D</code>&lt;br&gt;<code>list = -G</code>&lt;br&gt;<code>listall = -L</code>&lt;br&gt;<code>mark = -M</code>&lt;br&gt;<code>rebuild = -R</code>&lt;br&gt;<code>set = -S</code></td>
</tr>
<tr>
<td>absolute</td>
<td>Optional</td>
<td>1</td>
<td>Specifies if the path is absolute. Default value is <code>false</code>.</td>
<td>-a</td>
</tr>
<tr>
<td>directory</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies a DQNS on a StorNext file system to be used with the actions.</td>
<td>-d</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### Quota

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>Optional</td>
<td>1</td>
<td>When used with the list or listall actions, numbers will be printed as exact values.</td>
<td>-e</td>
</tr>
<tr>
<td>files</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-f</td>
</tr>
<tr>
<td>groupname</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies the name of a group to get or set with the list or set action.</td>
<td>-g</td>
</tr>
<tr>
<td>hostname</td>
<td>Optional</td>
<td>1</td>
<td>Use a hostname in a StorNext cluster that is different from the cluster the command is being run on. This option is rarely needed.</td>
<td>-H</td>
</tr>
<tr>
<td>hardlimit</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies a hard limit to set when used with the set action.</td>
<td>-h</td>
</tr>
<tr>
<td>softlimit</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies a soft limit to set when used with the set action.</td>
<td>-s</td>
</tr>
<tr>
<td>graceperiod</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies a grace period to set when used with the set action.</td>
<td>-t</td>
</tr>
<tr>
<td>user</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies the name of a user to get or set with the list or set action.</td>
<td>-u</td>
</tr>
</tbody>
</table>

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

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

```xml
<?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>
    </quota>
  </directoryQuotas>
</snquota>
```
<report>

<report>

<cancel_requests/>

This web service cancels requests.

</cancel_requests>

</report>
## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are <code>json</code> and <code>text</code>. The default value is <code>text</code>.</td>
<td><code>- F</code></td>
</tr>
<tr>
<td>requestid</td>
<td>Required</td>
<td>1</td>
<td>The request id that needs to be canceled.</td>
<td><code>- r</code></td>
</tr>
</tbody>
</table>

## Example

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

## Output

```json
{
    "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 `f scqueue` command.

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously.</td>
<td>None</td>
</tr>
</tbody>
</table>

The valid values are `sync` and `async`. The default value is `sync`.
### Chapter 1: StorNext Web Services Commands (V2)

#### Report

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast. Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>request</td>
<td>Optional</td>
<td>1</td>
<td>The request identifier of the request to be reported.</td>
<td>-r</td>
</tr>
<tr>
<td>file</td>
<td>Optional</td>
<td>N</td>
<td>The absolute path of the file(s).</td>
<td>-f</td>
</tr>
</tbody>
</table>

### Example

https://<<SERVER>>/sws/v2/report/fsqueue/file

?file=/stornext/snfs1/xxx/testfile.0

&format=xml

### Output

```xml
<?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>
</header>
```
Chapter 1: StorNext Web Services Commands (V2)

Report

<?xml version="1.0" encoding="UTF-8" ?><header><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>
Chapter 1: StorNext Web Services Commands (V2)

Report

Report / Media

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

This web service runs the `fsqueue` command.

Parameters

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

**Note:** The `fast` and `slow` queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

| format     | Optional  | 1   | The output format requested. Valid values are `xml`, `json` and `text`. The default is `text`. | -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
Chapter 1: StorNext Web Services Commands (V2)

Report

Output

```xml
<?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>
  </footer>
</fsqueue>
```
Report / Mover Host

Active mover host summary.

This web service runs the `fsqueue` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-F</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Example

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>
</fsqueue>
Report / Mover Request

Active mover request summary.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <strong>sync</strong> and <strong>async</strong>. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>fast</strong> and <strong>slow</strong> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default is <strong>text</strong>.</td>
<td>- F</td>
</tr>
</tbody>
</table>

Example

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

Output

```xml
<?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>
    </header>
</fsqueue>
```
Chapter 1: StorNext Web Services Commands (V2)

Report

<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>
estatuses
estatus
<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>
Report / Resource

Active resource request summary.

This web service runs the `fsqueue` command.

Parameters

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

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>
  </footer>
</fsqueue>
Report / List Available Software Backups

Provides a listing of available SNSM software backups which you can use for a restore operation. The information is listed in a tabular format. Full backups and all subsequent partial backups are grouped together. Each copy is also grouped together. The media required for each set of backups is shown in the MEDIA column.

This web service runs the snbkpreport command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. This web service only supports text output. The default is text.</td>
<td>None</td>
</tr>
</tbody>
</table>

Example

```
/opt/quantum/curl/bin/curl -k "https://192.168.1.75/sws/v2/report/snbackup"
```
### StorNext Web Services Commands (V2) Report

## Output

--- ADIC Backup Report ---

Sdisk and Object Storage Device Information (They may or may not be used for storing backups)

<table>
<thead>
<tr>
<th>ALIAS</th>
<th>TYPE</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysdisk</td>
<td>SDISK</td>
<td>/stornext/u1/mysdisk/adic.xps-centos7-c</td>
</tr>
</tbody>
</table>

Backup ID: 0

Time Range: 2019.06.28 - 2019.06.28

Copy: 1

<table>
<thead>
<tr>
<th>DATE</th>
<th>TYPE</th>
<th>STATUS</th>
<th>MEDIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019.06.28:10:19:42</td>
<td>full</td>
<td>PASS</td>
<td>mysdisk</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Report

Report / Get Status for Software Backups

Get status for currently running backup or results of previous backup.

This web service runs the `snbackup -s` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code> and <code>json</code>. The default is <code>json</code>.</td>
<td>-F</td>
</tr>
</tbody>
</table>

Example

```
curl -k "https://192.168.1.75/sws/v2/report/snbackup/status?format=JSON"
```

Output

```
{
    "header": {
        "gmtDateISO": "2019-06-28T15:23:39Z",
        "localDate": "2019-06-28",
        "localTime": "10:23:39",
        "gmtTime": "15:23:39",
        "gmtDayOfWeek": 5,
        "gmtDate": "2019-06-28",
        "commandName": "snbackup",
        "commandDescription": "Execute backup of configuration, database, and file system metadata."
    },
    "localDateISO": "2019-06-28T08:23:39",
    "localDayOfWeek": 5,
    "commandLine": "snbackup -s -F JSON"
}
```
Report / Status

Displays status summary information on requests.

This web service runs the `fsqueue` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is <strong>sync</strong>.</td>
<td>None</td>
</tr>
<tr>
<td>jobpriority</td>
<td>Optional</td>
<td>1</td>
<td>This parameter describes the priority of an asynchronous job. Valid values are <strong>high</strong>, <strong>medium</strong> and <strong>low</strong>.</td>
<td>None</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

**Report**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast. Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td>None</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>queuedonly</td>
<td>Optional</td>
<td>1</td>
<td>This parameter will limit the output to include requests that are queued waiting on resource allocation.</td>
<td>-q</td>
</tr>
<tr>
<td>moveronly</td>
<td>Optional</td>
<td>1</td>
<td>This parameter will limit the output to include requests that have data movers which are working on requests.</td>
<td>-i</td>
</tr>
</tbody>
</table>
| timelimit    | Optional  | 1   | This parameter will limit the output to include requests that have completed within the specified amount of time. The time value can be set in units of seconds, minutes, hours or days using the following suffix values:  
  - s: seconds (default if no suffix specified)  
  - m: minutes  
  - h: hours  
  - d: days  | -c         |
| request      | Optional  | 1   | The request identifier of the request to be reported. | -r         |

**Example**

https://<<SERVER>>/sws/v2/report/fsqueue/status

?  
format=json
Output

```json
{
   "header": {
      "commandName": "fsqueue",
      "commandLine": "fsqueue -s -i -q -c 10 -F JSON",
      "commandDescription": "View subsystem resource requests",
      "localDateISO": "2017-02-06T14:44:42",
      "localDate": "2017-02-06",
      "localTime": "14:44:42",
      "localDayOfWeek": 1,
      "gmtDateISO": "2017-02-06T20:44:42Z",
      "gmtDate": "2017-02-06",
      "gmtTime": "20:44:42",
      "gmtDayOfWeek": 1
   },
   "requestQueue": [
      {
         "queuePos": 1,
         "requestId": 420,
         "requestState": "COPY",
         "requestType": "STR",
         "startTime": 1486413864,
         "stopTime": 0,
         "runTime": 18,
         "totalFiles": 3000,
         "filesCopied": 2371,
         "filesFailed": 0,
         "totalDataSize": 76579736,
         "dataCopied": 52649600,
         "mediaId": "E00000",
         "deviceAlias": "vtl1_drv1",
         "deviceAlias2": "",
         "host": "snt756465-mdc1"
      },
      {
         "queuePos": 2,
         "requestId": 420,
         "requestState": "QUEUE",
         "requestType": "STR",
         "startTime": 1486413872,
         "stopTime": 0,
         "runTime": 10,
         "totalFiles": 3000,
```
"filesCopied": 0,
"filesFailed": 0,
"totalDataSize": 197321367,
"dataCopied": 0,
"mediaId": ",
"deviceAlias": ",
"deviceAlias2": ",
"host": ",
},
{
"queuePos": 3,
"requestId": 420,
"requestState": "QUEUE",
"requestType": "STR",
"startTime": 1486413872,
"stopTime": 0,
"runTime": 10,
"totalFiles": 3000,
"filesCopied": 0,
"filesFailed": 0,
"totalDataSize": 197321367,
"dataCopied": 0,
"mediaId": ",
"deviceAlias": ",
"deviceAlias2": ",
"host": ",
},
{
"queuePos": 4,
"requestId": 420,
"requestState": "QUEUE",
"requestType": "STR",
"startTime": 1486413872,
"stopTime": 0,
"runTime": 10,
"totalFiles": 3000,
"filesCopied": 0,
"filesFailed": 0,
"totalDataSize": 197321367,
"dataCopied": 0,
"mediaId": ",
"deviceAlias": ",
"deviceAlias2": ",
"host": ",
}
"host": ""}
},
{
"queuePos": 5,
"requestId": 420,
"requestState": "QUEUE",
"requestType": "STR",
"startTime": 1486413872,
"stopTime": 0,
"runTime": 10,
"totalFiles": 3000,
"filesCopied": 0,
"filesFailed": 0,
"totalDataSize": 197321367,
"dataCopied": 0,
"mediaId": "",
"deviceAlias": "",
"deviceAlias2": "",
"host": ""
},
{
"queuePos": 6,
"requestId": 420,
"requestState": "ALLOCATE",
"requestType": "STR",
"startTime": 1486413864,
"stopTime": 0,
"runTime": 18,
"totalFiles": 3000,
"filesCopied": 2372,
"filesFailed": 0,
"totalDataSize": 76579736,
"dataCopied": 52649472,
"mediaId": "E00028",
"deviceAlias": "vt11_drv2",
"deviceAlias2": "",
"host": "snt756465-mdc1"
},
{
"queuePos": 7,
"requestId": 420,
"requestState": "ALLOCATE",
"requestType": "STR",
"startTime": 1486413868,
"stopTime": 0,
"runTime": 18,
"totalFiles": 3000,
"filesCopied": 2372,
"filesFailed": 0,
"totalDataSize": 76579736,
"dataCopied": 52649472,
"mediaId": "E00028",
"deviceAlias": "vt11_drv2",
"deviceAlias2": "",
"host": "snt756465-mdc1"
"startTime": 1486413864,
"stopTime": 0,
"runTime": 18,
"totalFiles": 3000,
"filesCopied": 2371,
"filesFailed": 0,
"totalDataSize": 76579736,
"dataCopied": 52649600,
"mediaId": "E00027",
"deviceAlias": "vt11_drv3",
"deviceAlias2": "",
"host": "snt756465-mdc1"
},
{
"queuePos": 8,
"requestId": 420,
"requestState": "ALLOCATE",
"requestType": "STR",
"startTime": 1486413864,
"stopTime": 0,
"runTime": 18,
"totalFiles": 3000,
"filesCopied": 2372,
"filesFailed": 0,
"totalDataSize": 76579736,
"dataCopied": 52649472,
"mediaId": "E00026",
"deviceAlias": "vt11_drv4",
"deviceAlias2": "",
"host": "snt756465-mdc1"
},
{
"queuePos": 9,
"requestId": 420,
"requestState": "QUEUE",
"requestType": "STR",
"startTime": 1486413881,
"stopTime": 0,
"runTime": 1,
"totalFiles": 2778,
"filesCopied": 0,
"filesFailed": 0,
"totalDataSize": 33271651,
<table>
<thead>
<tr>
<th>Queue Position</th>
<th>Request ID</th>
<th>Request State</th>
<th>Request Type</th>
<th>Start Time</th>
<th>Stop Time</th>
<th>Run Time</th>
<th>Total Files</th>
<th>Files Copied</th>
<th>Files Failed</th>
<th>Total Data Size</th>
<th>Data Copied</th>
<th>Media ID</th>
<th>Device Alias</th>
<th>Device Alias 2</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>420</td>
<td>QUEUE</td>
<td>STR</td>
<td>1486413881</td>
<td>0</td>
<td>1</td>
<td>2778</td>
<td>0</td>
<td>0</td>
<td>33271651</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>420</td>
<td>QUEUE</td>
<td>STR</td>
<td>1486413881</td>
<td>0</td>
<td>1</td>
<td>2778</td>
<td>0</td>
<td>0</td>
<td>33271651</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Schedule

Schedule / Report
Prints the report of a feature or schedule.
This web service runs the fsschedule command.
Chapter 1: StorNext Web Services Commands (V2)

Schedule

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are xml, json and text. The default is text.</td>
<td>-F</td>
</tr>
<tr>
<td>feature</td>
<td>Optional</td>
<td>1</td>
<td>The name of the feature. Valid types are: clninfo, clnver, defrag, rebuild, p_backup, f_backup, spolicy, healthck, activevault, and archive_cmp.</td>
<td>-f</td>
</tr>
<tr>
<td>schedule</td>
<td>Optional</td>
<td>1</td>
<td>The schedule name.</td>
<td>-n</td>
</tr>
</tbody>
</table>

Example

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

Output

```
<?xml version="1.0" encoding="UTF-8"?>
```
Chapter 1: StorNext Web Services Commands (V2)
Schedule

<fschedule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fschedule.xsd">
  <header>
    <commandName>fschedule</commandName>
    <commandLine>/usr/adic/TSM/exec/fschedule -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>
    </schedule>
  </schedules>
</fschedule>
Schedule / Create

This web service will create a schedule.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td>-F</td>
</tr>
<tr>
<td>feature</td>
<td>Required</td>
<td>1</td>
<td>The name of the feature. Valid types are: <code>clninfo</code>, <code>clnver</code>, <code>defrag</code>, <code>rebuild</code>, <code>p_backup</code>, <code>f_backup</code>, <code>spolicy</code>, <code>healthck</code>, <code>activevault</code>, and <code>archive_cmp</code>.</td>
<td>-f</td>
</tr>
<tr>
<td>schedule</td>
<td>Required</td>
<td>1</td>
<td>The schedule name.</td>
<td>-n</td>
</tr>
<tr>
<td>period</td>
<td>Optional</td>
<td>1</td>
<td>The period of the schedule. Valid options are: <code>daily</code>, <code>weekly</code>, and <code>monthly</code>.</td>
<td>-p</td>
</tr>
<tr>
<td>weekday</td>
<td>Optional</td>
<td>1</td>
<td>The day of the week: Valid options are: <code>Sun</code>, <code>Mon</code>, <code>Tue</code>, <code>Wed</code>, <code>Thu</code>, <code>Fri</code>, or <code>Sat</code>.</td>
<td>-e</td>
</tr>
<tr>
<td>monthday</td>
<td>Optional</td>
<td>1</td>
<td>The day of the month: 1 – 31.</td>
<td>-y</td>
</tr>
<tr>
<td>runtime</td>
<td>Required</td>
<td>1</td>
<td>The start time of the feature defined as HHMM.</td>
<td>-t</td>
</tr>
<tr>
<td>window</td>
<td>Optional</td>
<td>1</td>
<td>The window or runtime offset defined as HHMM.</td>
<td>-w</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>option</td>
<td>Optional</td>
<td>1</td>
<td>The option used by the feature. Currently, only the spolicy feature requires an option, which is an existing policy class.</td>
<td>-o</td>
</tr>
<tr>
<td>archive</td>
<td>Optional</td>
<td>N</td>
<td>Applicable only for activevault feature. Media to be vaulted are selected from this list of archives.</td>
<td>- -</td>
</tr>
<tr>
<td>vault</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. The destination archive vault media.</td>
<td>-vault</td>
</tr>
<tr>
<td>copy</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. A list of copy numbers to query on.</td>
<td>-copy</td>
</tr>
<tr>
<td>used</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Select only media that have used size capacity. Size is in bytes by default, but a suffix of B(yches), K(ibibytes), M (ebibytes), G(ibibytes) or T(ebibytes) may be used to specify capacity.</td>
<td>-used</td>
</tr>
<tr>
<td>free</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Select only media that have size capacity remaining. Size is in bytes by default, but a suffix of B(yches), K(ibibytes), M (ebibytes), G(ibibytes) or T(ebibytes) may be used to specify capacity.</td>
<td>-remaining</td>
</tr>
<tr>
<td>age</td>
<td>Optional</td>
<td>1</td>
<td>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 YYYY:MM:DD:hh:mm:ss format.</td>
<td>-age</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-----</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>sort</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-sort</td>
</tr>
<tr>
<td>migrate</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-migrate</td>
</tr>
<tr>
<td>pending</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Valid values are true and false. If true, Select media where a vaulting operation is pending. If false, Select media where no vaulting operation is pending.</td>
<td>-pending</td>
</tr>
<tr>
<td>highmark</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-highmark</td>
</tr>
<tr>
<td>lowmark</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-lowmark</td>
</tr>
<tr>
<td>fill</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Override the ACTIVEVAULT_FULL_PERCENT sysparm value to consider vaulting media that is at or above the specified percent.</td>
<td>-fullpct</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-----</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>report</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Generate a media report based upon the selection criteria.</td>
<td>-report</td>
</tr>
<tr>
<td>includepolicy</td>
<td>Optional</td>
<td>N</td>
<td>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.</td>
<td>-include_policy</td>
</tr>
<tr>
<td>excludepolicy</td>
<td>Optional</td>
<td>N</td>
<td>Applicable only for activevault feature. Excludes media belonging to the listed policy classes.</td>
<td>-exclude_policy</td>
</tr>
<tr>
<td>capacity</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Report the current licensed capacity and total available licensed capacity.</td>
<td>-capacity</td>
</tr>
<tr>
<td>dryrun</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Show what media would be vaulted according to the selection criteria.</td>
<td>-dryrun</td>
</tr>
<tr>
<td>limit</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Limit the number of vaulted media to this number.</td>
<td>-limit</td>
</tr>
<tr>
<td>notify</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature. Set the email notification level for active vault policy admin alerts where level is either none, error, warn, or info. None 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.</td>
<td>-notify</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Schedule

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>noheader</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Do not display header or result count in the media report.</td>
<td>-noheader</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature.</td>
<td>-verbose</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. The name of the Active Vault policy to use for email notifications.</td>
<td>-policy</td>
</tr>
</tbody>
</table>

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>
  </header>
  <localDateISO>2014-05-15T12:25:28</localDateISO>
  <localDate>2014-05-15</localDate>

StorNext 6 Web Services Guide
< 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 >
</ footer>
Schedule / Update

This web service will update a schedule.
This web service runs the `fsschedule` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <code>fast</code> and <code>slow</code>. The default value is <code>fast</code>.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The <code>fast</code> and <code>slow</code> queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. Valid values are <code>xml</code>, <code>json</code> and <code>text</code>. The default is <code>text</code>.</td>
<td><code>-F</code></td>
</tr>
<tr>
<td>schedule</td>
<td>Required</td>
<td>1</td>
<td>The schedule name.</td>
<td><code>-n</code></td>
</tr>
<tr>
<td>period</td>
<td>Optional</td>
<td>1</td>
<td>The period of the schedule. Valid options are: <code>daily</code>, <code>weekly</code>, or <code>monthly</code>.</td>
<td><code>-p</code></td>
</tr>
<tr>
<td>weekday</td>
<td>Optional</td>
<td>1</td>
<td>The day of the week: Valid options are: Sun, Mon, Tue, Wed, Thu, Fri, or Sat.</td>
<td><code>-e</code></td>
</tr>
<tr>
<td>monthday</td>
<td>Optional</td>
<td>1</td>
<td>The day of the month: 1 – 31.</td>
<td><code>-y</code></td>
</tr>
<tr>
<td>runtime</td>
<td>Optional</td>
<td>1</td>
<td>The start time of the feature defined as HHMM.</td>
<td><code>-t</code></td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-----</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>window</td>
<td>Optional</td>
<td>1</td>
<td>The window or runtime offset defined as HHMM.</td>
<td>-w</td>
</tr>
<tr>
<td>option</td>
<td>Optional</td>
<td>1</td>
<td>The option used by the feature. Currently, only the spolicy feature requires an option, which is an existing policy class.</td>
<td>-o</td>
</tr>
<tr>
<td>archive</td>
<td>Optional</td>
<td>N</td>
<td>Applicable only for the activevault feature. Media to be vaulted are selected from this list of archives.</td>
<td>--</td>
</tr>
<tr>
<td>vault</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. The destination archive where to vault media.</td>
<td>-vault</td>
</tr>
<tr>
<td>copy</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. A list of copy numbers to query.</td>
<td>-copy</td>
</tr>
<tr>
<td>used</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Select only media that have used size capacity. Size is in bytes by default, but a suffix of B(abytes), K (ibibytes), M(ebibytes), G(ibibytes) or T (ebibytes) may be used to specify capacity.</td>
<td>-used</td>
</tr>
<tr>
<td>free</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Select only media that have size capacity remaining. Size is in bytes by default, but a suffix of B(abytes), K (ibibytes), M(ebibytes), G(ibibytes) or T (ebibytes) may be used to specify capacity.</td>
<td>-remaining</td>
</tr>
<tr>
<td>age</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the 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 YYYY:MM:DD:hh:mm:ss format.</td>
<td>-age</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-----</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>sort</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-sort</td>
</tr>
<tr>
<td>migrate</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the 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.</td>
<td>-migrate or -nomigrate</td>
</tr>
<tr>
<td>pending</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Valid values are true and false. If true, select media where a vaulting operation is pending. If false, select media where no vaulting operation is pending.</td>
<td>-pending or -nopending</td>
</tr>
<tr>
<td>highmark</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-highmark</td>
</tr>
<tr>
<td>lowmark</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-lowmark</td>
</tr>
<tr>
<td>fill</td>
<td>Optional</td>
<td>1</td>
<td>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.</td>
<td>-fullpct</td>
</tr>
<tr>
<td>Parameter</td>
<td>Req / Opt</td>
<td>Num</td>
<td>Description</td>
<td>CLI Option</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>report</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Generate a media report based upon the selection criteria.</td>
<td>-report</td>
</tr>
<tr>
<td>includepolicy</td>
<td>Optional</td>
<td>N</td>
<td>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.</td>
<td>-include_policy</td>
</tr>
<tr>
<td>excludepolicy</td>
<td>Optional</td>
<td>N</td>
<td>Applicable only for the activevault feature. Excludes media belonging to the listed policy classes.</td>
<td>-exclude_policy</td>
</tr>
<tr>
<td>capacity</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Report the current licensed capacity and total available licensed capacity.</td>
<td>-capacity</td>
</tr>
<tr>
<td>dryrun</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Show media that would be vaulted according to the selection criteria.</td>
<td>-dryrun</td>
</tr>
<tr>
<td>limit</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Limit the number of vaulted media to this number.</td>
<td>-limit</td>
</tr>
<tr>
<td>notify</td>
<td>Optional</td>
<td>1</td>
<td>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. None 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.</td>
<td>-notify</td>
</tr>
</tbody>
</table>
Chapter 1: StorNext Web Services Commands (V2)

Schedule

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>noheader</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. Do not display header or result count in the media report.</td>
<td>-noheader</td>
</tr>
<tr>
<td>verbose</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for activevault feature.</td>
<td>-verbose</td>
</tr>
<tr>
<td>policy</td>
<td>Optional</td>
<td>1</td>
<td>Applicable only for the activevault feature. The name of the Active Vault policy to use for email notifications.</td>
<td>-policy</td>
</tr>
</tbody>
</table>

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" xmlns="">
  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>
  </header>
  <localDate>2014-05-15</localDate>
Chapter 1: StorNext Web Services Commands (V2)

Schedule

<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>
Schedule / Delete

This web service deletes an existing schedule. This web service runs the `fsschedule` command.

Parameters

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

**Note:** The `fast` and `slow` queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

| format | Optional | 1 | The output format requested. Valid values are `xml`, `json` and `text`. The default is `text`. | -F |
| schedule | Required | 1 | The schedule name. | -n |

Example

```
https://<<SERVER>>/sws/v2/schedule/fsschedule/delete
?schedule=schedule1
&format=xml
```
Chapter 1: StorNext Web Services Commands (V2)
Schedule

Output

```xml
<?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>
```
Schedule / Reset

This web service resets an existing schedule.
This web service runs the `fsschedule` command.

Parameters

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

Note: The `fast` and `slow` queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.
### Parameter | Req / Opt | Num | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>feature</td>
<td>Required</td>
<td>1</td>
<td>The name of the feature. Valid types are: clninfo, clnver, defrag, rebuild, p_backup, f_backup, spolicy, healthck, activevault, or archive_cmp.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example

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

### Output

```xml
<?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>
```

StorNext 6 Web Services Guide 204
<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

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

#### Note:
The `fast` and `slow` queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

### 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.

> **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: - 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
System / File System Report

This web service reports the status of a file system and the status of stripe groups that belong to it.
Chapter 1: StorNext Web Services Commands (V2)
System

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are sync and async. The default value is sync.</td>
<td>None</td>
</tr>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are fast and slow. The default value is fast.</td>
<td>None</td>
</tr>
<tr>
<td>Note: The fast and slow queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>1</td>
<td>The output format requested. The valid values are xml, json and text. The default value is text.</td>
<td>-F</td>
</tr>
<tr>
<td>filesystem</td>
<td>Required</td>
<td>1</td>
<td>The file system name for which status is required.</td>
<td>None</td>
</tr>
<tr>
<td>showstripegroups</td>
<td>Optional</td>
<td>1</td>
<td>This option will also enable the status information of stripe groups. The valid values are true and false. The default value is false.</td>
<td>None</td>
</tr>
</tbody>
</table>

Example

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

Output

```json
{}
```
"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

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

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

Output

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

System / Parameters

Report the value for the specified Tertiary Manager system parameter.
This web service runs the `showsysparm` command.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Optional</td>
<td>1</td>
<td>This option specifies if the job will be run synchronously or asynchronously. The valid values are <code>sync</code> and <code>async</code>. The default value is <code>sync</code>.</td>
<td>None</td>
</tr>
</tbody>
</table>
### Chapter 1: StorNext Web Services Commands (V2)

#### System

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Req / Opt</th>
<th>Num</th>
<th>Description</th>
<th>CLI Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobtype</td>
<td>Optional</td>
<td>1</td>
<td>This parameter is used for async mode only and enables the user to run it as a fast or slow job. Valid values are <strong>fast</strong> and <strong>slow</strong>. The default value is <strong>fast</strong>.</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** The **fast** and **slow** queues provide a method for segregating jobs that run quickly from jobs that take a long time to complete. Each queue is configured by default to run up to four jobs simultaneously in order within their respective queues.

<table>
<thead>
<tr>
<th>format</th>
<th>Optional</th>
<th>1</th>
<th>The output format requested. Valid values are <strong>xml</strong>, <strong>json</strong> and <strong>text</strong>. The default is <strong>text</strong>.</th>
<th>-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>Required</td>
<td>N</td>
<td>Name of the system parameter.</td>
<td>no CLI option</td>
</tr>
</tbody>
</table>

### Example

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

### Output

```xml
<?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>
  </header>
```

<header>
<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>
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.

This chapter contains the following topics:

- Prerequisites .................................................................................................................. 218
- Getting Started ............................................................................................................. 218
- Run the Web Services ................................................................................................... 221
- Run the Web Services from a Client Application ......................................................... 225
- Troubleshooting ............................................................................................................. 271
Table 3: StorNext Web Services V2 Prerequisites

<table>
<thead>
<tr>
<th>Prerequisite</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>StorNext 5 release 5.3 (or later) should be installed on the system.</td>
<td>For information on how to install StorNext, contact a Quantum representative or visit the StorNext Installation Guide online at <a href="http://www.quantum.com/snsdocs">http://www.quantum.com/snsdocs</a>.</td>
</tr>
<tr>
<td>Web services should be enabled.</td>
<td>For information on how to enable web services, see the section Getting Started below.</td>
</tr>
<tr>
<td>Web service authentication and user credentials should be configured.</td>
<td>For information on how to configure authentication and user credentials, see the section Getting Started below.</td>
</tr>
<tr>
<td>If authentication is required, web service user(s) needs to be created.</td>
<td>For information on how to create web service users, see the section Getting Started below.</td>
</tr>
</tbody>
</table>
| 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:  
  - Microsoft Windows Internet Explorer  
  - Mozilla Firefox  
  - Google Chrome  
| If SWS V2 web services are invoked from customer source code, relevant libraries need to installed and referenced from the source code. | For additional information, see the section Run the Web Services from a Client Application on page 225. |

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 above). 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

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**.
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

Getting Started

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.
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.

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
Run the Web Services

Run SWS V2 web services from a supported web browser (see Prerequisites on page 218). Below are examples of a sample run of the web services using Google Chrome.

Example of SWS V2 Run Using Google Chrome

In the example, the web service is run with the Protocol configured to https and Authentication Type configured to None.
Example of SWS V2 Protocol Failure Using Google Chrome

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

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/snsdocs](http://www.quantum.com/snsdocs).

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.
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

Run the Web Services

Example of Web Service Run With The XML Format Using Google Chrome

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

Example of Web Service Run With The JSON Format Using Google Chrome

Below is an example of the same web service run with the JSON format request.
Example of a SWS V2 Run Using Google Chrome

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

Example of a SWS V2 User Credential Failure Using Google Chrome

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

Example of SWS V2 User Credentials Incorrect Error Using Google Chrome

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

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

- 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.
How to Determine File States Using the fsfileinfo Command

You can determine the store and truncation states using the `fsfileinfo` command and checking the json/xml output for the following key words:

- `location`
- `targetStubSize`
- `targetStubScale`

⚠️ **Note:** The `targetStubScale` is 1024, stubs defined in 1KB blocks.

- `existingStubSize`
- `existingStubScale`

⚠️ **Note:** The `existingStubScale` is 1024, not provided if file not truncated.

⚠️ **Note:** While you can define the stub size in 1KB blocks, the actual truncation only leaves stubs rounded up to multiples of the file system block size. The block size for managed file systems is typically 4KB. For example, if you have three files with stub sizes of 1k, 2k and 3k they are all left with 4096 bytes in their stubs at truncation time.

**Example of File State: On Disk Only (Not Stored)**

```bash
% fsfileinfo -F json junk | egrep "location|Stub"
"location": "DISK",
"targetStubSize": 0,
"targetStubScale": 1024,
"existingStubSize": "n/a",
```

- The location of **DISK** means the file is not stored; no copies in the **ARCHIVE**.
- The **targetStubSize** of **0** means no stub has been requested for this file at truncation time.
- The **targetStubScale** is always **1024**.
- The **existingStubSize** of **n/a** means the file is not truncated.

**Example of File State: File Stored but Not Truncated (No Stub Defined)**

```bash
% fsfileinfo -F json junk | egrep "location|Stub"
"location": "DISK AND ARCHIVE",
```
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2
Run the Web Services from a Client Application

Example of File State: File Stored and Truncated (No Stub Defined)

```
% fsfileinfo -F json junk | grep "location|Stub"
"location": "ARCHIVE",
"targetStubSize": 0,
"targetStubScale": 1024,
"existingStubSize": "n/a",
"existingStubScale": 1024,
```

- The location of **DISK AND ARCHIVE** means the file is stored and not truncated.
- The **targetStubSize** of 0 means no stub has been requested for this file at truncation time.
- The **targetStubScale** is always **1024**.
- The **existingStubSize** of **n/a** means the file is not truncated.

Example of File State: File Stored and Truncated (2k Stub Defined)

```
% fsfileinfo -F json junk | grep "location|Stub"
"location": "ARCHIVE",
"targetStubSize": 2,
"targetStubScale": 1024,
"existingStubSize": 4,
"existingStubScale": 1024,
```

- The location of **ARCHIVE** means the file is stored and truncated.
- The **targetStubSize** of 0 means no stub has been requested for this file at truncation time.
- The **targetStubScale** is always **1024**.
- The **existingStubSize** of 0 means no stub has been left on disk.
- The **existingStubScale** is always **1024**.
The location of ARCHIVE means the file is stored and truncated.

- The `targetStubSize` of 2 means a stub of 2k has been requested.
- The `targetStubScale` is always 1024.
- The `existingStubSize` of 4 means a stub of 4k exists on disk; the 2k was rounded up to the 4k file system block size.
- The `existingStubScale` is always 1024.

**Example of File State: File Stored and Truncated (8k Stub Defined)**

```
% fsfileinfo -F json junk | egrep "location|Stub"
"location": "ARCHIVE",
"targetStubSize": 8,
"targetStubScale": 1024,
"existingStubSize": 8,
"existingStubScale": 1024,
```

- The location of ARCHIVE means the file is stored and truncated.
- The `targetStubSize` of 8 means a stub of 8k has been requested.
- The `targetStubScale` is always 1024.
- The `existingStubSize` of 8 means a stub of 8k exists on disk.
- The `existingStubScale` is always 1024.

### Java

The example below uses the Jersey library. Download the latest version of Jersey library online at: [https://jersey.java.net/download.html](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 221.

```java
// 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")
    .QueryParam("file", filePath)
    .QueryParam("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.

```java
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) {
        }
        HttpsURLConnection.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.

```java
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.

```java
// 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.

```java
private String getMediaType(String format) {
    String mediaType = MediaType.TEXTPLAIN;
    if (format != null) {
        if (format.equalsIgnoreCase("json")) {
            mediaType = MediaType.APPLICATIONJSON;
        }
    }
    return mediaType;
}
```
Finally, invoke the web service and obtain the response as a **String**.

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

Below is an example of the response appears.

```json
{
    "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",
  }
]
If the Authentication Type is configured to User, the username and password also requires to be passed as a query string.

```java
String response = service.path("sws/v2/file/fsfileinfo")
    .QueryParam("file", filePath)
    .QueryParam("format", format)
    .QueryParam("username", userName)
    .QueryParam("password", password)
    .Accept(mediaType).Get(String.class);
```

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

```java
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.

```java
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.HTTPSProperties;

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")
                .queryParam("file", filePath).queryParam("format", format)
                .queryParam("username", userName).queryParam("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")
        .queryParam("policy", className).queryParam("format", format)
        .queryParam("username", userName).queryParam("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();

```java
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")
    .queryParam("directory", filePath).queryParam("format", format)
    .queryParam("username", userName).queryParam("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")
        .queryParam("file", filePath)
        .queryParam("copies", Integer.toString(copies))
        .queryParam("format", format).accept(mediaType)
        .queryParam("username", userName).queryParam("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);
```
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/fsrmdiskcopy")
            .queryParam("file", filePath).queryParam("format", format)
            .queryParam("username", userName).queryParam("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")
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2
Run the Web Services from a Client Application

```java
public String getSetSnQuota(boolean useHttps, String userName,
        String password, String hostName, String port,
        String fsname, 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")
            .queryParam("path", fspath).queryParam("directory", dirPath)
            .queryParam("action", "create").queryParam("format", format)
            .queryParam("username", userName).queryParam("password", password)
            .accept(mediaType).get(String.class);

    return response;
}

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

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

StorNext 6 Web Services Guide 239
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) {
}
HttpsURLConnection.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) {
}

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

protected URI getBaseURI(String host, String port, boolean https) {
    String protocol = https ? "https://" : "http://";
    if (port != null) {
        protocol += port;
    }
    return new URI(protocol + host);
}
In the next few examples, the assumption is that the Protocol is https and Authentication Type is User. The username is wsuser and the password is wspass. Also, JSON is the format of response.

Consider a managed file system smpltomedia mounted at /stornext/snfx1/smpltomedia and a file named foobar0 in the file system. We attempt to store the file, remove the disk copy and retrieve it using web services available in SWS V2. Refer the source code above.

First, check the file information. Use the fsfileinfo web service.

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

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/smpltomedia/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/smpltomedia/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": "smpltomedia",
"trunc": "MINTIME",
"cleanDBInfo": "NO",
"altStoreLocation": "Disabled",
"medias": [ { "message": "None" } ],
"checksums": [ { "summary": "N" } ],
"encryptions": [ { "summary": "N" } ],
"compressions": [ { "summary": "N" } ],
"objects": [ { "summary": "N" } ] },
Now, try to find out what policy this file path is associated with. Use the **fsdirclass** web service.

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

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

```json
{
   "header": {
      "commandName": "fsdirclass",
      "commandLine": "/usr/adic/TSM/bin/fsdirclass /stornext/snfx1/smpltomedia -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/smpltomedia",
"classId": "smpltomedia"
}
],
"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"
}
}
String response = caller.getFsClassInfo(true, "wsuser", "wspass", "192.168.36.128", "444", "json", "smpltomedia" );
System.out.println(response);

The output is shown below.

```json
{
   "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"
      }
   ]
}
```
Store the foobar0 file using the fsstore web service; only store one copy of the file.

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

The output is shown below.
Run the Web Services from a Client Application

```json
{
    "header": {
        "commandName": "fsstore",
        "commandLine": "/usr/adic/TSM/bin/fsstore -c 1 -F json /stornext/snfx1/smpltomedical/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": "Tertiary Manager software request received."
        }
    ]
}
```
When the file is stored, execute the **fsfileinfo** web service again to ensure the file is stored.

```java
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.
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2
Run the Web Services from a Client Application

```json
{
    "header": {
        "commandName": "fsfileinfo",
        "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json /stornext/snfx1/smpltomedia/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/smpltomedia/foobar0",
            "storedPathFileName": "/stornext/snfx1/smpltomedia/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": "smpltomedia",
            "trunc": "MINTIME",
            "cleanDBInfo": "NO",
            "altStoreLocation": "Disabled",
            "medias": [
```
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

Run the Web Services from a Client Application

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

```java
String response = caller.getFsRmDiskCopy(true, "wsuser", "wspass",
```
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

Run the Web Services from a Client Application

```
"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."
        }
    ]
}
```
Verify the file is removed using the `fsfileinfo` command.

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

Notice the location of the file is **ARCHIVE** which signifies the disk copy is removed.
Run the Web Services from a Client Application

```json
{
    "fileName": "/stornext/snfx1/smpltomedia/foobar0",
    "storedPathFileName": "/stornext/snfx1/smpltomedia/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": "smpltomedia",
    "trunc": "MINTIME",
    "cleanDBInfo": "NO",
    "altStoreLocation": "Disabled",
    "medias": [
        {
            "mediaId": "000005",
            "copy": 1
        }
    ],
    "checksums": [
        {
            "summary": "N"
        }
    ],
    "encryptions": [
        {
            "summary": "N"
        }
    ],
    "compressions": [
        {
            "summary": "N"
        }
    ],
    "objects": [
        {
            "summary": "N"
        }
    ]
}
```
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

Run the Web Services from a Client Application

```
"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/smpltomedia/foobar0");
System.out.println(response);
```

Notice the retrieve operation is successful.

```
{
    "header": {
        "commandName": "fsretrieve",
        "commandLine": "/usr/adic/TSM/bin/fsretrieve -F json 
/stornext/snfx1/smpltomedia/foobar0",
        "commandDescription": "Retrieve files from media and place on disk",
```
Run the Web Services from a Client Application

```json
"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/smpltomedia/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",
"gmtDateISOEnd": "2015-10-27T01:54:45Z",
"gmtDateEnd": "2015-10-27",
"gmtTimeEnd": "01:54:45",
"gmtDayOfWeekEnd": 1
}
```
Execute `fsfileinfo` one last time to ensure the file is retrieved.

```java
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.

```json
{
   "header": {
      "commandName": "fsfileinfo",
      "commandLine": "/usr/adic/TSM/bin/fsfileinfo -F json /stornext/snfx1/smpltomedia/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/smpltomedia/foobar0",
         "storedPathFileName": "/stornext/snfx1/smpltomedia/foobar0",
         "storedPathSameAsFileName": true,
         "lastModificationDateString": "26-oct-2015 19:42:08",
```
Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2
Run the Web Services from a Client Application

"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": "smpltomedia",
"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,
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.

**Note:** Ensure quotas are enabled before you run the web services. For information on how to enable quotas, see Add a File System online at the StorNext Documentation Center.

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

```java
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.

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

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

```java
String response = caller.getSetSnQuota(true, "wsuser", "wspass", 
```
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"
        }
    ]
}
```
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.

```java
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 fsretrive
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.
### Chapter 2: Software Development Kit (SDK) on Developing Applications Using StorNext Web Services V2

#### Run the Web Services from a Client Application

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;moverRequestList&quot;</td>
<td>This section displays the jobs that are executed by the storage manager. This section displays a requestId assigned by the storage manager.</td>
</tr>
<tr>
<td></td>
<td>- The requestType can either be Store or Retrieve.</td>
</tr>
<tr>
<td></td>
<td>- The state parameter displays the current state of the job. The valid values for state are READY, QUEUE, PROCESS, FORMAT, COPY and COMPLETE.</td>
</tr>
<tr>
<td></td>
<td>- The positioninqueue parameter displays the position of the job in the storage manager queue.</td>
</tr>
</tbody>
</table>

| "moverProgressList" | This section displays the job that is currently running in the storage manager.                                                                                                                             |
|                    | - The host parameter displays the hostname of the system in which this job is being executed. The storage manager assigns the requestId when the job is registered under storage manager for execution. |
|                    | - The deviceAlias is the alias for the drive on which this job is being executed.                                                                                                                           |
|                    | - The runTime parameter displays the time that has elapsed since the job started.                                                                                                                          |
|                    | - The totalFiles parameter displays the number of files that will be copied for this job.                                                                                                                  |
|                    | - The filesCopied parameter displays the files that have already been completed.                                                                                                                          |
|                    | - The filesFailed parameter displays the number of files that could not be copied.                                                                                                                       |

| "completedJobList" | This section displays the jobs that have completed execution and begins with header information in the header node.                                                                                       |
|                    | - The jobInfo node displays details about a particular job.                                                                                                                                               |
|                    | - The statuses node contains individual status of files that are being copied in this particular job.                                                                                                     |
|                    | - The footer section contains footer information.                                                                                                                                                    |

| "pendingJobList"   | This section displays the jobs that are waiting to be executed by the storage manager.                                                                                                                     |
|                    | - The job parameter displays the job Id.                                                                                                                                                                |
|                    | - The positioninagentqueue is the position of the job in the agent queue.                                                                                                                               |
|                    | - The exitcode parameter displays the current exit code.                                                                                                                                                 |
|                    | - The state of waiting jobs is shown. In general, state for a waiting job is QUEUED.                                                                                                                 |
[{
    "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"
        }
    ]
}]

Run the Web Services from a Client Application

```json
{
    "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."
}
Run the Web Services from a Client Application

```
"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."
}
```


"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."}
Run the Web Services from a Client Application

[JSON data]

},
{
  "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"}
Troubleshooting

<table>
<thead>
<tr>
<th>Issue</th>
<th>Tips</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Service Reported Off</td>
<td>On the StorNext GUI, click Services, and then click Web Services (V2). Ensure the State parameter is enabled</td>
</tr>
</tbody>
</table>
## Issue | Tips
--- | ---
Browser Error on Invoking a Web Service | **Check the protocol and the port number for the request.**
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 **Protocol** is **http** and port used is **443**, the web service request may not work.

Web Service is Not Working | **The root cause may be that the web server that hosts the web services is offline.**
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.

Sent a XML Request, but Response is Not XML | **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/snsdocs](http://www.quantum.com/snsdocs).
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.
Appendix A: Sample Perl Script
Sample Perl Script

Download the Sample Perl Script

Download a .zip file containing a text version of the 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 ($PROTOCOL eq 'https')) {  
033:    print "Protocol should be either 'http' or 'https'\n";  
034:    exit;  
035: }  
036:  
037: # check format, should be json, text, or xml  
038: #  
039: unless (($FORMAT eq 'json') or ($FORMAT eq 'text') or ($FORMAT eq 'xml')) {  
040:    print "Format should be either 'json', 'text', or 'xml'\n";  
041:    exit;  
042: }  
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";
071: print "FORMAT: $FORMAT\n";
072: print "PROTOCOL: $PROTOCOL\n";
073: print "AUTHENTICATE: $AUTHENTICATE\n";
074:  
075:  
076: sub check_job_status {

my ($rsp, @more) = @_;  
print "$rsp\n";
if ($rsp =~ /Job has been successfully submitted/i) {
    $rsp =~ /"job": "(\d+)"/i;
    my $jobno = $1;
    my $myrsp;
    my $myrsp2;
    $myrsp2 = do_webservices_cmd( '/job/mover/info',
        "job=$jobno");
    print "$myrsp2\n";
    $myrsp = do_webservices_cmd( '/job/info',
        "job=$jobno");
    my $count = 0;
    while ($myrsp =~ /RUNNING/i) {
        sleep(1);
        $myrsp2 = do_webservices_cmd( '/job/mover/info',
            "job=$jobno");
        print "$myrsp2\n";
        $myrsp = do_webservices_cmd( '/job/info',
            "job=$jobno");
        print "$myrsp\n";
        $count += 1;
    }
    $myrsp2 = do_webservices_cmd( '/job/mover/info',
        "job=$jobno");
    print "$myrsp2\n";
    print "Checked for completion: $count times\n";
    print "$myrsp\n";
    if ($myrsp =~ /ERROR/i) {
        print "Job had error -------------------------------\n";
        $FAILURES += 1;
    }
sub do_webservices_cmd {
    my ($cmd, @pieces) = @_;
    #? print "cmd: $cmd\n"; #?
    #? print "pieces: @pieces\n"; #?
    #? for my $piece (@pieces) { #?
        #?     print "piece: $piece\n"; #?
    #? } #?
    #? set up the URL prefix
    #?
    my $prefix;
    if ($cmd =~ /wsconfig/i) {
        $prefix = "http://$IP:81/sws/v2";
    } elsif ($PROTOCOL =~ /https/i) {
        $prefix = "https://$IP/sws/v2";
    } else {
        $prefix = "http://$IP:81/sws/v2";
    }
    my $ws = $prefix . $cmd;
Appendix A: Sample Perl Script

138:   if ($AUTHENTICATE ne '') {
139:       push( @pieces, "username=$USERNAME");
140:       push( @pieces, "password=$PASSWORD");
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) {

Appendix A: Sample Perl Script

```
169:     print "\n";
170:     print "WS: $ws\n";
171:     print "="*70="\n" if ($ws =~ /mover/);
172: }
173:
174:     my $req = new HTTP::Request GET => $ws;
175:     my $res = $ua->request($req);
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_webervices_cmd( '/system/info',
199:     "format=text" );
```
Appendix A: Sample Perl Script

```perl
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"
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';
```
Appendix A: Sample Perl Script

```perl
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: for my $i (0,1) {
236:   my $filename = "file.$i";
237:   push @singles_paths_sync, "$singles_path/$filename";
238:   push @dirs_paths_sync, "$dirs_path_sync/$filename";
239:   push @dirs_paths_async, "$dirs_path_async/$filename";
240:   push @multi_paths_sync, "$multi_path/$filename";
241: } 
242:
243: for my $i (2,3) {
244:   my $filename = "file.$i";
245:   push @singles_paths_async, "$singles_path/$filename";
246:   push @multi_paths_async, "$multi_path/$filename";
247: } 
248:
249: # 5) Use WS fsstore to save both files from the first directory to TAPE
250: for my $filepath (@singles_paths_sync) {
251:   $ws_rsp = do_webservices_cmd( '/file/fsstore',
```
Appendix A: Sample Perl Script

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>
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:
Appendix A: Sample Perl Script

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:
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:
Appendix A: Sample Perl Script

```perl
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
351: 
352:  # /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
353:  $ws_rsp = do_webservices_cmd( '/file/fsretrieve',
354:                       "file=$multi_paths_sync[0]",
```
$multi_paths_sync[1];

356: print "$ws_rsp\n";

357:

358: # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
359: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
360:   "file=$multi_paths_sync[0]",
361:   "file=$multi_paths_sync[1]");

362: print "$ws_rsp\n";

363: # 12-18) Repeat steps 5-11 using async mode
364:
365: # /sws/v2/file/fsstore?file=<filepath>
366: for my $filepath (@singles_paths_async) {
367:   $ws_rsp = do_webservices_cmd( '/file/fsstore',
368:     "file=$filepath",
369:     "mode=async");
370:   check_job_status( $ws_rsp);
371: }

372: # /sws/v2/file/fsfileinfo?file=<filepath>
373: for my $filepath (@singles_paths_async) {
374:   $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
375:     "file=$filepath");
376:   print "$ws_rsp\n";
377: }

378: # 13) Use WS fsstore to save the second directory to TAPE
386: # /sws/v2/file/fsstore?directory=<dirpath>
387: $ws_rsp = do_webservices_cmd( '/file/fsstore',
388:     "directory=$dirs_path_async",
389:     "mode=async"); 
390: check_job_status( $ws_rsp);
391: # /sws/v2/file/fsfileinfo?directory=<dirpath>
392: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
393:     "directory=$dirs_path_async");
394: print "$ws_rsp\n";
395: # 14) Use WS fsstore to save both files from the third directory to TAPE
396: # /sws/v2/file/fsstore?file=<f1>&file=<f2>
397: $ws_rsp = do_webservices_cmd( '/file/fsstore',
398:     "file=$multi_paths_async[0]",
399:     "file=$multi_paths_async[1]",
400:     "mode=async");
401: check_job_status( $ws_rsp);
402: # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
403: $ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
404:     "file=$multi_paths_async[0]",
405:     "file=$multi_paths_async[1]" );
406: print "$ws_rsp\n";
407: # 15) Use WS rmdiskcopy to truncate both files in each of 3 directories
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: # 15) Use WS rmdiskcopy to truncate both files in each of 3 directories
415: 
416:
Appendix A: Sample Perl Script

```perl
417: # /sws/v2/file/fsrmdiskcopy
418: for my $filepath (@singles_paths_async) {
419:     $ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
420:                                     "file=$filepath");
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) {
```
Appendix A: Sample Perl Script

```perl
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>
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]",
```
"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: $ws_rsp = do_webservices_cmd( '/policy/fsclassinfo',
490: "policy=policy_min_i");
491: print "$ws_rsp\n";
492:
493:
494: print "\n";
495: print "Number of failures: $FAILURES\n";
496:
497:
498: #
Copyright 2015 Quantum Corporation
Appendix B: Sample Python Script
Sample Python Script

Download the Sample Python Script

Download a .zip file containing a text version of the sample Python script.

```python
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 })
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
Appendix B: Sample Python Script

```python
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'.")
```
Appendix B: Sample Python Script

```python
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
106:    #
107:    m = re.compile( '\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$')
```
if not m.match(args.ip):
    print "ip not in correct format: %s" % (args.ip,)
    sys.exit(1)

# check authentication

if args.authenticate == '':
    username = ''
    password = ''
elif '/' in args.authenticate:
    username, password = args.authenticate.split('/')
    username = username + '_py'
    password = password + '_py'
else:
    print 'Authenticate argument must have '/' between the username and password:'
    print "value: %s" % (args.authenticate,)
    sys.exit(1)

def do_webservices_cmd(cmd, pieces={}):
    global args
    global failures
    global username
    global password
    global verbose

    # set up the URL prefix
    #
    if 'wsconfig' in cmd:
        prefix = "http://%s:81/sws/v2" % (args.ip,)
    elif 'https' in args.protocol:
prefix = "https://%s:81/sws/v2" % (args.ip,)
else:
    prefix = "http://%s:81/sws/v2" % (args.ip,)

if 'format' not in pieces:
    pieces['format'] = args.format

if 'job' not in cmd:
    pieces['job'] = cmd

if args.authenticate != '':
    pieces['username'] = username
    pieces['password'] = password

# all web-service URLs are lowercase
ws = ws.lower()

response = requests.get( ws, pieces, verify=False)
if verbose:
    print "WS: %s" % (urllib.unquote(response.url),)

rsp = response.text

# print "rsp: %s" % (rsp,) #?
if response.status_code == 200:
    return rsp
else:
    failures += 1
    return "Fail: %s" % (rsp,)

if __name__ == '__main__':
    global username
global password

    process_args()

    # Description of Web Services sample programs:

    # 0) Please turn on web-services using the StorNext GUI
    # Also choose protocol and authentication through the SN GUI

    # 1) Do WS system info, returning TEXT, XML and JSON

    # /sws/v2/system/info?format=text

    ws_rsp = dowebservices_cmd( '/system/info',
                                { 'format' : 'text' } )
    print ws_rsp

    # /sws/v2/system/info?format=xml

    ws_rsp = dowebservices_cmd( '/system/info',
                                { 'format' : 'xml' } )
    print ws_rsp
# /sws/v2/system/info?format=json

ws_rsp = do_webservices_cmd( '/system/info',
    { 'format' : 'json' } )

print ws_rsp

# 2) Create a policy for a managed file system

# 3) 3 directories should exist:
# a) Directory for single-file manipulation
singles_path = '/stornext/snfs1/sample_dir_singles/py'
singles_paths_sync = []
singles_paths_async = []

# b) Directory for directory manipulation
dirs_path_sync = '/stornext/snfs1/sample_dir_dirs_sync/py'
dirs_paths_sync = []
dirs_path_async = '/stornext/snfs1/sample_dir_dirs_async/py'
dirs_paths_async = []

# c) Directory for multi-file manipulation
multi_path = '/stornext/snfs1/sample_dir_multi/py'
multi_paths_sync = []
multi_paths_async = []

# 4) Create arrays with path names of both files in each of those directories
# for single and multi commands, files 0 and 1 are for sync, files 2 and 3 are for async
# for directory commands, there is a directory for sync and one for async
for i in [0,1]:
    filename = "file.%d" % (i,)
    singles_paths_sync.append( "%s/%s" % (singles_path, filename))
    dirs_paths_sync.append( "%s/%s" % (dirs_path_sync, filename))
    multi_paths_sync.append( "%s/%s" % (multi_path, filename))

for i in [2,3]:
    filename = "file.%d" % (i,)
    singles_paths_async.append( "%s/%s" % (singles_path, filename))
    multi_paths_async.append( "%s/%s" % (multi_path, filename))

# 5-11) steps 5-11 use sync mode, steps 12-18 use async mode
# 5) Use WS fsstore to save both files from the first directory to TAPE
# /sws/v2/file/fsstore?file=<filepath>
for filepath in singles_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsstore',
        { 'file' : filepath } )
    print ws_rsp

# /sws/v2/file/fsfileinfo?file=<filepath>
for filepath in singles_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
        { 'file' : filepath } )
    print ws_rsp
# 6) Use WS fsstore to save the second directory to TAPE
263:     # /sws/v2/file/fsstore?directory=<dirpath>
264:     ws_rsp = do_webservices_cmd( '/file/fsstore',
265:         { 'directory' : dirs_path_sync })
266:     print ws_rsp
267:     # /sws/v2/file/fsfileinfo?directory=<dirpath>
268:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
269:         { 'directory' : dirs_path_sync })
270:     print ws_rsp
271:     # 7) Use WS fsstore to save both files from the third directory to TAPE
272:     # /sws/v2/file/fsstore?file=<f1>&file=<f2>
273:     ws_rsp = do_webservices_cmd( '/file/fsstore',
274:         { 'file' : [ multi_paths_sync[0],
276:     print ws_rsp
277:     # /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
278:     ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
279:         { 'file' : [ multi_paths_sync[0],
281:     print ws_rsp
282:     # 8) Use WS rmdiskcopy to truncate both files in each of 3 directories
# /sws/v2/file/fsrmdiskcopy

for filepath in singles_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
    { 'file' : filepath })
    print ws_rsp

for filepath in dirs_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
    { 'file' : filepath })
    print ws_rsp

for filepath in multi_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
    { 'file' : filepath })
    print ws_rsp

# 9) Use WS fsretrieve to restore both files to first directory from TAPE
# /sws/v2/file/fsretrieve?file=<filepath>

for filepath in singles_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsretrieve',
    { 'file' : filepath })
    print ws_rsp

for filepath in singles_paths_sync:
    ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
    { 'file' : filepath })
    print ws_rsp
Appendix B: Sample Python Script

```python
# 10) Use WS fsretrieve to restore the second directory from TAPE
# /sws/v2/file/fsretrieve?directory=<dirpath>
ws_rsp = do_webservices_cmd( '/file/fsretrieve',
    { 'directory' : dirs_path_sync } )
print ws_rsp

# /sws/v2/file/fsfileinfo?directory=<dirpath>
ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
    { 'directory' : dirs_path_sync } )
print ws_rsp

# 11) Use WS fsretrieve to restore both files in the third directory from TAPE
# /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
ws_rsp = do_webservices_cmd( '/file/fsretrieve',
    { 'file' : [ multi_paths_sync[0],
               multi_paths_sync[1] ] } )
print ws_rsp

# /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
    { 'file' : [ multi_paths_sync[0],
               multi_paths_sync[1] ] } )
print ws_rsp
```

StorNext 6 Web Services Guide 303
# 12-18) Repeat steps 5-11 using async mode

# 12) Use WS fsstore to save both files from the first directory to TAPE

# /sws/v2/file/fsstore?file=<filepath>

for filepath in singles_paths_async:
    ws_rsp = do_webservices_cmd('/file/fsstore',
                              { 'file': filepath,
                                'mode': 'async' })
    check_job_status(ws_rsp)

# /sws/v2/file/fsfileinfo?file=<filepath>

for filepath in singles_paths_async:
    ws_rsp = do_webservices_cmd('/file/fsfileinfo',
                              { 'file': filepath })
    print ws_rsp

# 13) Use WS fsstore to save the second directory to TAPE

# /sws/v2/file/fsstore?directory=<dirpath>

ws_rsp = do_webservices_cmd('/file/fsstore',
                             { 'directory': dirs_path_async,
                               'mode': 'async' })
check_job_status(ws_rsp)

# /sws/v2/file/fsfileinfo?directory=<dirpath>

ws_rsp = do_webservices_cmd('/file/fsfileinfo',
                             { 'directory': dirs_path_async })
print ws_rsp
# 14) Use WS fsstore to save both files from the third directory to TAPE
# /sws/v2/file/fsstore?file=<f1>&file=<f2>
ws_rsp = do_webservices_cmd( '/file/fsstore',
    { 'file' : [ multi_paths_async[0],
                 multi_paths_async[1] ],
       'mode' : 'async' } )
check_job_status( ws_rsp)

# /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
    { 'file' : [ multi_paths_async[0],
                 multi_paths_async[1] ] } )
print ws_rsp

# 15) Use WS rmdiskcopy to truncate both files in each of 3 directories
# /sws/v2/file/fsrmdiskcopy
for filepath in singles_paths_async:
    ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
        { 'file' : filepath } )
    print ws_rsp
for filepath in dirs_paths_async:
    ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
        { 'file' : filepath } )
    print ws_rsp
for filepath in multi_paths_async:
    ws_rsp = do_webservices_cmd( '/file/fsrmdiskcopy',
                              { 'file': filepath } )
print ws_rsp

# 16) Use WS fsretrieve to restore both files to first directory from TAPE
# /sws/v2/file/fsretrieve?file=<filepath>
for filepath in singles_paths_async:
    ws_rsp = do_webservices_cmd( '/file/fsretrieve',
                                 { 'file': filepath,
                                   'mode': 'async' } )
check_job_status( ws_rsp)

# /sws/v2/file/fsfileinfo?file=<filepath>

# 17) Use WS fsretrieve to restore the second directory from TAPE
# /sws/v2/file/fsretrieve?directory=<dirpath>
for filepath in singles_paths_async:
    ws_rsp = do_webservices_cmd( '/file/fsretrieve',
                                 { 'file': filepath } )
print ws_rsp

# /sws/v2/file/fsfileinfo?directory=<dirpath>
ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
    { 'directory' : dirs_path_async } )
print ws_rsp

# 18) Use WS fsretrieve to restore both files in the third directory from TAPE
# /sws/v2/file/fsretrieve?file=<f1>&file=<f2>
ws_rsp = do_webservices_cmd( '/file/fsretrieve',
    { 'file' : [ multi_paths_async[0],
        multi_paths_async[1] ],
        'mode' : 'async' } )
check_job_status( ws_rsp)

# /sws/v2/file/fsfileinfo?file=<f1>&file=<f2>
ws_rsp = do_webservices_cmd( '/file/fsfileinfo',
    { 'file' : [ multi_paths_async[0],
        multi_paths_async[1] ] } )

# 19) Get policy information
ws_rsp = do_webservices_cmd( '/policy/fsdirclass',
    { 'directory' : dirs_path_async} )
print ws_rsp

ws_rsp = do_webservices_cmd( '/policy/fsclassinfo',
    { 'policy' : 'policy_min_i' } )
print ws_rsp

print "Number of failures: %d" % (failures,)
Copyright 2015 Quantum Corporation