Web Services for File System Pooling

6-69020-01, Rev. A

Quantum

StorNext Web Services for File System Pooling, 6-69020-01, Rev. A, November 2021, Product of USA.

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

COPYRIGHT STATEMENT

Contents

Preface

This document describes the commands supported by the StorNext Web Services for File System Pooling and contains the following chapters:

Audience

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

Notational Conventions

This manual uses the following conventions:

Convention	Example
User input is shown in bold monospace font.	**`./DARTinstall`**

Convention	Example
Computer output and command line examples are shown in monospace font.	`./DARTinstall`
User input variables are enclosed in angle brackets.	**http://<ip_address>/cgi-bin/stats**

For UNIX and Linux commands, the command prompt is implied.	`./DARTinstall`
	is the same as
	`# ./DARTinstall`

File and directory names, menu commands, button names, and window names are shown in bold font.	**/data/upload**
Menu names separated by arrows indicate a sequence of menus to be navigated.	**Utilities > Firmware**

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 *Informationsvejledning vedrørende system-, sikkerheds- og lovbestemmelser for Quantum produkter, før produktet betjenes*.

AVERTISSEMENT : Avant d'utiliser ce produit, lisez toutes les instructions et les avertissements de ce document et du *Guide d'informations sur le système, la sécurité et la réglementation de Quantum*.

WARNUNG: Lesen Sie vor der Inbetriebnahme dieses Produkts alle Anleitungen und Warnungen in diesem Dokument und im *System-, Sicherheits- und Betriebsbestimmungen-Handbuch für Quantum-Produkte*.

ADVERTENCIA: Antes de hacer funcionar este producto, lea todas las instrucciones y advertencias de este documento y de la *Guía de información normativa, del sistema y de seguridad de los productos de Quantum*.

VARNING: Läs igenom alla instruktioner och varningar i detta dokument och i *Quantums produktsystem, säkerhet och reglerande informationsguide* innan denna produkt används.

ВНИМАНИЕ! Перед началом эксплуатации данного изделия прочтите все инструкции и предупреждения, приведенные в настоящем документе и в *Руководстве по системе, технике безопасности и действующим нормативам компании Quantum*.

警告：本製品を使用される前に、本書と『*Quantum製品システム、安全、規制情報ガイド*』に記載されているすべての説明と警告をお読みください。

경고: 본 제품을 작동하기 전에 본 문서와 *Quantum 제품 시스템, 안전 및 규제 정보 설명서*에 있는 모든 지침과 경고를 참조합니다.

警告：在操作本产品之前，请阅读本文档和 *Quantum 产品系统、安全和法规信息指南*中的所有说明和警告。

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

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

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

https://www.quantum.com/serviceandsupport/get-help/index.aspx#contact-support

Contacts

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

https://www.quantum.com/aboutus/contactus/index.aspx

Comments

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

doc-comments@quantum.com

Getting More Information or Help

Quantum University, 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 University 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:

  https://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:

  https://www.quantum.com/customercenter/

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

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

https://www.quantum.com/serviceandsupport/warrantyinformation/index.aspx

Chapter 1: Introduction

This chapter contains the following topics:

Introduction

This document describes a set of web service extensions to StorNext which are intended to facilitate work-flows for moving content between sets of storage with different I/O characteristics. Some work-flows require extreme bandwidth during operation, but this speed is not needed for the complete life cycle of the data. Rather than provision a large pool of storage all capable of sustaining these data rates, we enable segmenting the storage between pools with different characteristics and then scheduling the movement of content between these pools.

Quantum does not attempt to support reorganizing content between different name spaces or collating it from several parts of a hierarchy into one, rather moving the on-disk content of a set of files between locations without impacting the naming or other attributes of the content.

Chapter 2: Concepts

This chapter contains the following topics:

Concepts

Storage Pools

A file system can be divided into a set of named pools, each pool is one or more stripe groups. These pools represent the targets for content steering. The pool is really a mechanism to assign an affinity to a set of stripe groups at run time. However, it also supports a few other configuration options such as making a pool exclusive to content labelled with the affinity and specifying a bandwidth limit for movement of content in and out of the pool.

Conceptually there could be a set of smaller fast pools and a large slow pool. The fast pools would be setup as exclusive, each one would have sufficient bandwidth for a workflow and content would be scheduled into the pool, operated on, and then moved back out of the pool to the large slower pool. The different pools could operate in parallel.

A pool may be designated as the default pool for new content that is not part of content which is explicitly targeting a pool.

Data Movement

After you create a pool, then you can initiate a data movement task. There are six types of tasks:

Task	Description
Move	Move the specified set of content to the named pool if it is not already there.
Promote	Make a copy of the specified content in the named pool, the existing copy remains as a named stream associated with the file, this original copy cannot be accessed without special code.
Demote	Move the content back to the specified pool. If an alternate copy exists from a promote and the file has not changed in the interim, then the old copy is made the live copy and the current live copy is removed. For content which has not been changed this is a very fast operation as no data movement is required. The intent is that promote and demote operate as a pair and allow for making a copy for playback in high performance storage and subsequently removing that copy without I/O.
Remove	Delete the secondary copy from a previous promote operation. **Note:** All pool labels are removed as well.
Inventory	Track the location of the file content by pool and report how many files and how much data exist in each pool in the specified set of content.

Each task defines:

- The target pool for data movement
- Do an optional pre-check pass. This sums the space needed and will not proceed if the target pool contains insufficient space.
- Should allocation sessions be used. If allocation sessions are enabled in the task then files within a directory are sorted alphanumerically and subsets of the work are performed in sequential order in their own allocation session. This leads to optimized data layout for the content and is mostly of use when targeting spinning storage.
- An optional wildcard pattern match for file names, only files which match the pattern will be used.
- The set of paths to be operated on. Paths can be specified as a list of files and or directories. Directories may be scanned recursively or non-recursively. A directory with a / at the end of the path is recursively processed, one with no / just has its immediate children scanned. This means data movement can target more than just a directory tree.

A data movement task is executed by the single host it is queued to, but any host running the service may be used to move data.

The task is split into sub-tasks and data movement is performed in parallel by a pool of workers. Using an n-way buffering scheme on a per file basis means a large file will always be doing reads and writes at the same time. Since the source and target are independent pools of storage, this is generally capable of operating at the data rate of the slower pool. For small files the metadata overhead dominates.

A task is submitted as an asynchronous operation and a handle is returned which can be used to track the status of the work. Once completed, these handles may be deleted by another rest call.

Data Movement Engine

The engine maintains a queue of tasks. Multiple tasks may be running in parallel, although typically one runs after the other. The work is persistent across restarts.

You can control the amount of parallelism and the size of the I/Os used to do the work. The queue might be suspended and resumed in order to prevent data movement work perturbing production work on the system.

Space Reservation

If you are placing content into a pool, then the pre-check pass is intended to ensure the pool has sufficient room at the start of movement to complete the task. However, other activity can interfere with this by consuming space while the task is running.

In order to guarantee resources, a reservation scheme is envisaged. A data movement task targeting a pool would specify a reservation – this could either be the size of the data being referenced, or that size plus some additional space. The task would not be executed if the reservation cannot be met by the pool. This would be implemented as a new form of quota. The quota could not be setup if the sum of the hard limits of the other quotas in the pool did not leave enough room to add in the reservation.

These quotas would then track the actual consumption by different projects in the pool and would prevent a project from over-running its reservation without first obtaining more quota.

A quota report could show the set of projects in the different pools and indicate which one should be moved out of the pool to make room, or perhaps which one could have its quota reduced to allow more space for another project.

Pinned Content

Content could be labeled in such a way that it cannot be moved unless a task uses an override capability. Some content could be pinned to slow storage, while other content could be pinned to fast storage. While not currently implemented, this would probably be a move and pin in place type operation.

Storage Manager Interactions

The implementation does not use Storage Manager and it does not move content that is currently offline.

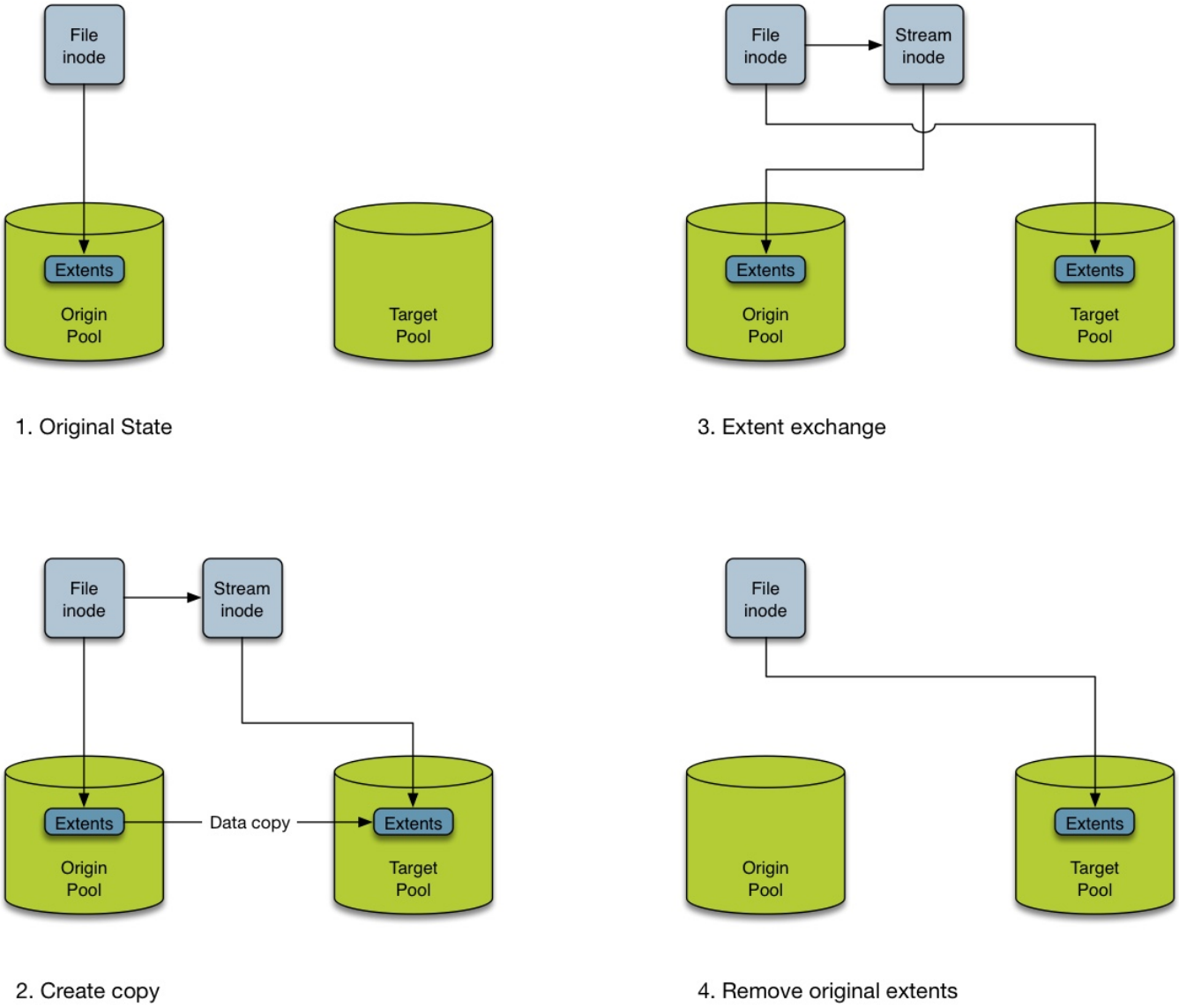
Chapter 3: Named Streams

This chapter contains the following topics:

Named Streams

All data movement is handled using named stream inodes. These are alternate data forks associated with the main inode. When there are two copies of content, there are not two named files visible in the file system; instead, there is the main inode and the named stream inode containing the copy. Should the main inode be deleted, the named stream is also deleted. If a file is renamed, the named stream remains intact. Named streams are not directly visible to the file system interface, they require special calls to access. The image below illustrates the different phases of manipulating a copy of the content.

Figure 1: Component Operations of Pooling



- A **move** operation performs all four steps.
- A **promote** operation performs **Step 1**, **Step 2** and **Step 3**.
- A **demote** operation is reverse to **promote** and starts with two copies (as in **Step 2**), where the **Origin Pool** is what was the **Target Pool** for **promote**. Then it proceeds with **Step 3** and **Step 4**.

Chapter 4: Web Service Specifications

This chapter contains the following topics:

Web Service Specifications

All operations use JSON for the request and response bodies.

Pool Services

There are three operations to create, list or remove a pool configuration, and an operation to report on the current state of the pools in a file system. These are services provided by an FSM. The pool configuration is stored persistently by the file system in its metadata and is protected in the same manner as other file system metadata.

HTTP Method	URI	Description
GET	`https://.../api/snfs/v1/config/pool`	Response body describes current pools

HTTP Method	URI	Description
PUT	`https://.../api/snfs/v1/config/pool`	Request contains new pool configuration, response body contains the same response.
DELETE	`https://.../api/snfs/v1/config/pool`	Remove the existing pool configuration
GET	`https://.../api/snfs/v1/pools`	Get current pool state including the current capacity and usage of the pools

The pool configuration is as follows. Each pool contains a set of data stripe groups, each stripe group can only be in one pool (a name which must be unique and 8 or less characters). If a pool is marked exclusive then it only takes content which has the associated affinity.

```
{
    "pools": [
        {
            "name": "fast",
            "exclusive" : true,
            "stripegroups": [
                "sg2"
            ]
        }
    ]
}
```

The pool status output looks similar with extra information:

```
{
    "pools": [
        {
            "name": "slow",
            "stripegroups": [
                "sg1"
```

```
            ],
            "disks": [
                {
                    "name": "clone_data"
                }
            ],
            "fullstripe": 4194304,
            "free_blocks": 1312164352,
            "total_blocks": 1317143552
        },
        {
            "name": "fast",
            "exclusive": true,
            "stripegroups": [
                "sg2"
            ],
            "disks": [
                {
                    "name": "CvfsDisk_nvme1"
                }
            ],
            "fullstripe": 4194304,
            "free_blocks": 97574240,
            "total_blocks": 97674240
        }
    ]
```


The set of disks involved in the pool is here for internal consumption by the system. The **fullstripe** represents the I/O size needed to hit all the LUNs in the stripe groups in the job. This will be factored into the io size used by reads and writes so that data is accessed from all LUNs in parallel. Because files may not be aligned on stripe boundaries it can lead to one device getting two I/Os per read or write.

Data Movement Operations

Data movement operations are to a specific data mover service. All data movement requests are **PUT** operations to a queue, they return immediately with either an error or with a position in the queue. The record remains in the queue and represents the current status of the operation until it is deleted.

HTTP Method	URI	Description
PUT	`https://.../api/sntier/v1/queue`	Submit data movement operation, returns the queue job id or an error.
GET	`https://.../api/sntier/v1/queue`	List the data movement jobs and their status.
GET	`https://.../api/sntier/v1/queue/job-id`	List the status of a specific data movement job.
DELETE	`https://.../api/sntier/v1/queue`	Delete all completed data movement jobs.
DELETE	`https://.../api/sntier/v1/queue/job-id`	Delete a specific data movement job, this can act as cancellation of a job prior to it running as well as cleanup after termination.

The contents of a data movement operation:

```
{
    "action" : "promote",
"pool" : "fast",
“precheck” : false,
    "sessions" : true,
    "paths" : [
        "/snfs/flexsync/video/",
       “/snfs/flexsync/backgrounds”
    ]
}
```

A promote to the fast pool is performed on the contents of the video folder recursively and the backgrounds folder non-recursively. The response is:

```
[
    {
        "job": 1000,
        "type": "promote",
        "state": "completed",
        "queueTime": "Fri Jul 20 15:17:48 2018",
        "childjobs": 19,
        "startTime": "Fri Jul 20 15:17:48 2018",
        "elapsedSec": 25.092348999999999,
        "filesScanned": 400,
        "filesMoved": 400,
        "bytesMoved": 20394803200,
        "bytesPerSec": "775M",
        "finishTime": "Fri Jul 20 15:18:13 2018",
        "parameters": {
            "action": "promote",
            "pool": "fast",
            "sessions": true,
            "paths": [
                "/snfs/flexsync/video/"
            ],
            "mount": "/snfs/flexsync",
            "io_size": 4194304
        }
    }
]
```


A **GET** on this URI at the same data mover responds with the job status, a **GET** with the added parameter of **?details=true** produces more verbose output.


```
[
    {
```

```
        "job": 1000,
        "type": "promote",
        "state": "completed",
        "queueTime": "Fri Jul 20 15:17:48 2018",
        "childjobs": 19,
        "startTime": "Fri Jul 20 15:17:48 2018",
        "elapsedSec": 25.092348999999999,
        "filesScanned": 400,
        "filesMoved": 400,
        "bytesMoved": 20394803200,
        "bytesPerSec": "775M",
        "finishTime": "Fri Jul 20 15:18:13 2018",
        "parameters": {
            "action": "promote",
            "pool": "fast",
            "sessions": true,
            "paths": [
                "/snfs/flexsync/video/"
            ],
            "mount": "/snfs/flexsync",
            "io_size": 4194304
        }
    }
]
```


An **inventory** job contains different details about the content:


```
[
    {
        "job": 1001,
        "type": "inventory",
```

```
        "state": "completed",
        "queueTime": "Fri Jul 20 15:46:31 2018",
        "childjobs": 19,
        "startTime": "Fri Jul 20 15:46:31 2018",
        "elapsedSec": 0.044970000000000003,
        "inventory": [
            {
                "pool": "fast",
                "directories": 1,
                "files": 400,
                "file_blocks": 39833600,
                "tier_copies": 400,
                "tier_blocks": 39833600,
                "tier_invalid": 0
            }
        ],
        "finishTime": "Fri Jul 20 15:46:31 2018",
        "parameters": {
            "action": "inventory",
            "paths": [
                "/snfs/flexsync/video/"
            ],
            "mount": "/snfs/flexsync"
        }
    }
]
```


A **DELETE** on the queue removes all terminated jobs, a **DELETE** on queue/jobid deletes a specific job if in queued or completed state. It cannot cancel a job in mid operation.

You can suspend a job with a **PUT** on queue/jobid?state=**suspended** and you can resume a job with a **PUT** on queue/jobid?state=**running**.

Data Mover Control

You can control various parameters of the data mover.

HTTP Method	URI	Description
GET	`https://.../api/sntier/v1/queue/config`	Retrieve the configuration parameters of the mover.
PUT	`https://.../api/sntier/v1/queue/config`	Update the configuration parameters of the mover.

The current set of parameters as which can be controlled are represented by the JSON:

```
{
    "queuestate": "running",
    "job_threads": 8,
    "buffer_size": 8388608,
    "job_size": 1073741824,
    "min_job_size": 16777216,
    "job_entries": 64,
    "io_threads": 2,
    "queue_depth": 4,
}
```

Parameter	Description
queuestate	You can set to **running** or **suspended**.
job_threads	Defines how many different batches of work can be processed at once. A job is broken up into smaller jobs handled by these threads.
buffer_size	The size used for reads and writes, all I/O is O_DIRECT.
job_size	The size in bytes which will be tackled sequentially by one **job_thread** When allocation sessions are in use, this is the amount of data which will end up targeted to a single allocation session. **Note:** A really large file is handled by a single sub job.
min_job_size	The number of files considered as a single sub job.

Parameter	Description
job_entries	The minimum amount of bytes in a sub job, if files are small, more files are placed in the single job for processing.
io_threads	The number of threads doing reads or writes for each sub job.
queue_depth	The number of buffers used by each queue thread.

Chapter 5: Policy and Template Settings

This chapter contains the following topics:

Policy and Template Settings

A template is a partial definition of a job or policy that you can use as a predefined set of rules for a **job** or **policy**. A template allows an administrator to predefine a set of tasks that an end user can apply to content. A templates does not allow specification of the content; the specification of the content is derived from a job or a policy definition.

You can create a template with a **POST** on **api/sntier/v1/template** with parameters in the request body:

```
{
    "name": "SSD",
    "task": {
        "action": "move",
        "target_pool": "veryfast"
    },
    "criteria": {
        "pool": "fast",
        "min_age": 300,
```

```
        "file_only" : true
    },
    "schedule": {
        "frequency": 120
    }
}
```

The response is:

```
{
    "name": "SSD",
    "task": {
        "action": "move",
        "target_pool": "veryfast"
    },
    "criteria": {
        "pool": "fast",
        "min_age": 300,
        "file_only": true
    },
    "schedule": {
        "frequency": 120
    },
    "id": "ae3ac2f1-8118-45d7-98c9-fd7e383bb224"
}
```

You can list the templates with a **GET** on **api/sntier/v1/template**.

You can create a policy with a **POST** on **api/sntier/v1/manage** with parameters in the request body. You can use a previously created template as one of parameters:

```
{
    "content" : {
        "search" : "/stornext/snfs1"
    },
    "template" : "SSD"
}
```

You can list the policies with a **GET** on **api/sntier/v1/manage**. To obtain time in human readable format, use the additional parameter ? time_fmt=1. The response is:

```
{
    "id": "9ea256a7-ff51-4036-8d92-6b161cf1c8a8",
    "policy": {
        "content": {
            "search": "/stornext/snfs1",
            "mount": "/stornext/snfs1"
        },
        "name": "SSD",
        "task": {
            "action": "move",
            "target_pool": "veryfast"
        },
        "criteria": {
            "pool": "fast",
            "min_age": 300,
            "file_only": true
        },
        "schedule": {
            "frequency": 120
        }
    },
    "nextScanTime": "Tue Nov 16 14:47:10 2021"
}
```

Quantum

Quantum technology, software, and services provide the solutions that today's organizations need to make video and other unstructured data smarter – so their data works for them and not the other way around. With over 40 years of innovation, Quantum's end-to-end platform is uniquely equipped to orchestrate, protect, and enrich data across its lifecycle, providing enhanced intelligence and actionable insights. Leading organizations in cloud services, entertainment, government, research, education, transportation, and enterprise IT trust Quantum to bring their data to life, because data makes life better, safer, and smarter. Quantum is listed on Nasdaq (QMCO) and the Russell 2000® Index. For more information visit www.quantum.com.

www.quantum.com | 800-677-6268