
# IBM Spectrum Scale™ Immutability Introduction, Configuration Guidance and Use cases

Nils Haustein | Senior Technical Staff Member | EMEA Storage Competence Center
Version: 2.0

# Contents


Executive Summary ..... 4
Introduction to IBM Spectrum Scale Immutability ..... 5
IBM Spectrum Scale IAM modes ..... 6
Immutable file operations ..... 7
Working with files in immutable filesets ..... 8
Creating an immutable fileset ..... 8
Working with immutable files ..... 8
POSIX commands ..... 9
IBM Spectrum Scale commands ..... 10
PowerShell ..... 10
Working with append-only files ..... 11
POSIX commands ..... 11
IBM Spectrum Scale commands ..... 13
Indefinite retention ..... 14
Working with directories ..... 15
Limitations ..... 15
Directories ..... 15
NFS exports and SMB shares ..... 15
Default retention times ..... 15
IBM Spectrum Scale limitations ..... 15
System configuration guidance ..... 17
User and Application ..... 18
IBM Spectrum Scale ..... 18
Configuration ..... 18
Administration ..... 19
Operating system configuration ..... 22
Storage system ..... 22
IBM Spectrum Archive EE ..... 23
WORM tape ..... 24
Environment ..... 24
Time server ..... 24
Networks ..... 24
Use cases ..... 25
Application supporting immutable filesets ..... 25
Automatically setting files to immutable ..... 26
Assigning different retention times based on file types ..... 28

Additional IBM Spectrum Scale archiving functions ..... 30
Appendix ..... 32
List policy example ..... 32
Sudo group example ..... 32
References ..... 33
Disclaimer ..... 35

**Acknowledgements**

Thanks to Kuei-Yu Wang-Knop and Haizhu Liu (IBM Spectrum Scale development) for the thorough review of this paper and for the great collaboration making this function available.

Thanks to Ulf Troppens (IBM Spectrum Scale architect) for his valuable feedback after reviewing this document.

Thanks to Mark Roberts of AWE, UK for the great feedback regarding system security configuration guidance.

Thanks to Felipe Knop (IBM Spectrum Scale development) for the guidance around security and especially sudo wrappers; and his thorough reviews.

Thanks to Sandeep Ramesh (IBM Spectrum Scale development) for the guidance and his thorough reviews.

 

# Executive Summary

IBM Spectrum Scale™ is a scalable parallel file system that can be used for many purposes. IBM Spectrum Scale also allows configuring immutable partitions in a file system that are called immutable filesets. IBM Spectrum Scale immutable filesets have been assessed for compliance in accordance to US regulations (SEC17a-4f) as well as German and Swiss tax laws and regulations [10].

From a user perspective a fileset is a directory within an IBM Spectrum Scale file system. Immutable filesets allows managing immutable and append-only files similar to the SnapLock® method invented by NetApp Inc. Furthermore immutable filesets can also be exported via NFS and SMB. This makes it easy for applications supporting the SnapLock® semantic to adopt IBM Spectrum Scale as an immutable file storage.

With the SnapLock® method files can be set to immutable or append-only for a given retention time using standard file system commands. During the retention time immutable files cannot be deleted or modified. When the retention time has expired immutable files can be deleted but still not modified. With indefinite retention provided by IBM Spectrum Scale deletion and modification of files can be prevented even if the retention time has expired.

With this immutability function IBM Spectrum Scale can be used for archiving where regulatory requirement demand to prevent changes and deletion of files during the life cycle. In fact the file audit logging function introduced with IBM Spectrum Scale version 5.0 uses immutable filesets to store file audit logs. Of course one and the same IBM Spectrum Scale cluster could be used for other purposes as well.

Applications not supporting the SnapLock-like immutability function can still benefit from IBM Spectrum Scale immutability because it allows to automatically set files to immutable. Such application can just use IBM Spectrum Scale as a file storage via NFS, SMB or as cluster member, and IBM Spectrum Scale takes care for making files immutable within minutes after they have been stored.

IBM Spectrum Scale immutability offers additional value adding functions beneficial for archiving solutions. One key function is the storage tiering allowing to transparently migrate files to tape and saving cost over time. The immutability function is also supported in conjunction with IBM Spectrum Scale encryption, compression and file audit logging. In addition IBM Spectrum Scale offers a comprehensive set of techniques to assure high availability, data and disaster protection.

# Introduction to IBM Spectrum Scale Immutability

This document introduces the IBM Spectrum Scale™ immutability function. It shows how to set it up and presents different ways for managing immutable and append-only files. This document also gives guidance for implementing IT security aspects in an IBM Spectrum Scale cluster addressing regulatory requirements. At the end it demonstrates two typical use cases managing immutable files. One use case involves applications that manage file immutability and another use case presents a solution to automatically set files to immutable within an IBM Spectrum Scale immutable fileset.

Immutability means to associate a file with a retention time and prevent any changes or deletion of the file data during the retention time. Immutable files are write-once-read-many protected (WORM) for a given period of time which can also be unlimited. After the retention time has expired the file can be deleted but not changed.

With the General Parallel File System version 3.4 (GPFS™ is now called IBM Spectrum Scale™) the extended attributes “immutable” and “append-only” were introduced for files and directories in the IBM Spectrum Scale file system [1]. These attributes can be set by the user who has permissions to set GPFS attributes using the command “mmchattr”. A file with the attribute “immutable” set to “yes” cannot be changed, renamed or deleted. A file where the attribute “append-only” is set to “yes”, allows append operations, but no other overwrites, renames or deletions. IBM Spectrum Scale allows setting these attributes on any file or directory, regardless if it resides in a fileset or not. The downside of this function is that these attributes can be reset by the user who has permissions to set attributes. In addition there is no concept of retention times.

IBM Spectrum Scale™ version 4.1.1 enhanced the immutability function to IBM Spectrum Scale filesets. An IBM Spectrum Scale fileset can be configured with an integrated Archive Manager (IAM) mode using the command “mmchfileset”. Files stored in such an immutable fileset can be set to immutable or append-only using standard POSIX or IBM Spectrum Scale commands.

The process to set a file to immutable is similar to the SnapLock® method[1] where a file can be set to immutable by setting the retention times via the file attribute “last access date” and by removing the write permissions [2]. These commands cause the extended IBM Spectrum Scale attributes “immutable” and “append-only” to be set implicitly for the file in the IBM Spectrum Scale file system. Depending on the IAM mode of the fileset these attributes cannot be reset. In addition there is the concept of retention times disallowing modifications and deletions of the files during a defined time period (retention or expiration time). When the retention time has expired files can be deleted but not modified.

[1] The SnapLock® method has been invented by NetApp Inc. and is state of the art to make file immutable in Network Attached Storage systems (NAS) [2]. NetApp and other storage vendors implement this method in their immutable NAS systems. Subsequently I will refer to the SnappLock® methods acknowledging that it is a trademark by NetApp Inc.

The IBM Spectrum Scale immutability function in Version 4.2 has been assessed for compliance by a globally recognized auditor in accordance to US regulations (SEC17a-4f) as well as German and Swiss tax and trade laws [10].

# IBM Spectrum Scale IAM modes

The Integrated Archive Manager (IAM) can be configured on an IBM Spectrum Scale fileset to prevent modification and deletions of files. It essentially restricts some file operations – such as update, append, overwrite, rename and delete – that are possible for files stored in normal (regular) IBM Spectrum Scale filesets. A fileset can be considered a logical partition within a file system allowing certain operations independent of the rest of the file system. From a user perspective a fileset is a directory in the file system. Functions available on a fileset include setting quota, taking snapshot, AFM caching and IAM mode[2]. The IAM mode defines the level of protection applied to files in an IBM Spectrum Scale fileset. A fileset where an IAM mode other than "none" is set is also called an immutable fileset. Otherwise the fileset is called a regular fileset.

Setting an IAM mode makes the fileset immutable. The IAM mode can be set using the "mmchfileset" command with the parameter "–iam-mode". A fileset can be set to one of the four IAM modes:

- **none**: No immutability mode is set (default), the fileset is a regular fileset
- **advisory (ad)**: Allows setting retention times and immutability, but files can be deleted with the proper file permission.
- **noncompliant (nc)**: Advisory mode plus files cannot be deleted if retention time has not expired. However retention times can be reset and files can be deleted but not changed.
- **compliant (co)**: Noncompliant mode plus retention time cannot be reset. When retention time has expired files can be deleted but not changed.
- **compliance-plus**: Compliant mode plus changes improving interoperability with SnapLock.

IAM modes can be upgraded from "advisory" to "noncompliant" to "compliant"/ "compliant-plus", but not downgraded. When upgrading the IAM mode intermediate levels can also be skipped, e.g. the IAM mode can be upgraded from "none" to "compliant" in one step.

Only the IAM mode "compliant" has been assessed for compliance [10].

It is possible to create nested filesets whereby one fileset is within the other fileset. The IAM mode is not inherited. This means when fileset1 is configured in compliant mode and fileset2 within fileset1 is a regular fileset, then files stored in fileset2 cannot be managed in a compliant manner.

[2] Dependent fileset do not have an own inode-space and allow quota and IAM modes. Independent filesets have an own inode-space within the IBM Spectrum Scale file system and allow snapshots and AFM caching in addition.

## Immutable file operations

As explained above files can be set to “immutable” or “append-only” in a regular fileset, a regular file system or in an immutable fileset. A regular fileset and file system does not have an IAM mode set (IAM mode is set to none). An immutable fileset has an IAM mode set other than “none”. The table below describes the key differences between regular filesets and immutable filesets, assuming the immutable fileset is configured in compliant IAM mode:

| File or Directory Operation | Regular fileset[3] | Immutable fileset[4] |
|---|---|---|
| Reset immutability attribute | yes | no |
| Reset append-only attribute | yes | no |
| Set immutability using mmchattr -i | yes | yes |
| Set append only using mmchattr -a | yes | yes |
| Set retention time using mmchattr -E | No | Yes |
| Set immutability using chmod -w | No | yes |
| Set append only using chmod -\|+w | No | yes |
| Set retention time using touch -at | No | Yes |
| Set IAM mode on fileset | Yes | Upgrade only |
| Set directory immutable | Yes | No |

The fundamental difference between regular filesets and immutable filesets is that files in an immutable fileset can be made immutable or append-only using standard POSIX commands. In addition a retention time can be set for files in immutable filesets.

[3] A regular fileset can also be the root fileset of a file system.
[4] Assumes that the fileset is configured in compliant IAM mode.

# Working with files in immutable filesets

This section outlines the steps for setting files to immutable or append-only in an immutable fileset. File immutability can be managed with standard POSIX commands available in UNIX systems, with specific IBM Spectrum Scale commands or via SMB using Microsoft Windows PowerShell®.

Setting files to immutable or append-only using POSIX commands is appropriate for applications or users who do not directly interface with IBM Spectrum Scale commands. Applications and users can manage immutable files using POSIX commands from a cluster node with access to the IBM Spectrum Scale file system or via an NFS export. When using POSIX commands from a NFS export files cannot be set to append-only.

Setting files to immutable or append-only using IBM Spectrum Scale commands is appropriate for applications and users running on a cluster node that can directly access the immutable fileset in an IBM Spectrum Scale file system. Running commands is only possible from an IBM Spectrum Scale cluster node with administrative roles.

Setting files to immutable using PowerShell commands is appropriate for applications or users accessing the immutable fileset via SMB share. Files can be set to immutable but not to append-only via SMB. Furthermore, once the file is set to read-only the retention time cannot be changed.

## Creating an immutable fileset

The first step for working with immutable files is to create an immutable fileset:

Create a fileset (independent in this case)
```
# mmcrfileset <filesystem-name> <fileset-name> --inode-space new
```

Link the fileset to a directory within the IBM Spectrum Scale file system which must not exist at this point. This directory is the immutable fileset path:
```
# mmlinkfileset <filesystem-name> <fileset-name> -J <directory>
```

Set an IAM mode for the files. In this example we set the IAM mode “compliance”
```
# mmchfileset <filesystem> <fileset> --iam-mode compliant
```

To list the IAM mode of a fileset use the following command:
```
# mmlsfileset <filesystem> <fileset> --iam-mode
```

Note, from a compliance perspective it is recommended to set the cluster-wide parameter “indefiniteRetentionProtection” to “yes”. Setting this parameter can be done with the command, it requires the cluster to be offline:
```
# mmchconfig indefiniteRetentionProtection=yes
```

## Working with immutable files

The process to set files to immutable in an immutable fileset is comprised of two steps:

- Setting the retention time

- Setting the file to immutable or read-only

This process corresponds the SnapLock® method. There are two ways to set files to immutable:

- Using standard POSIX commands
- Using IBM Spectrum Scale command

The retention time can be extended but not reduced for immutable files stored in an immutable fileset configured in IAM mode “compliant” Furthermore an immutable file in such fileset cannot be renamed, overwritten, appended or deleted. An immutable file can be deleted if the retention time (relative to the system date and time) has expired.

## POSIX commands

POSIX commands can be either when accessing the immutable fileset within the IBM Spectrum Scale cluster from a cluster node or via an NFS export. Two POSIX command can be used set a file to immutable and assign a retention time.

The retention time is encoded in the last access date of a file and can be set or extended using the following POSIX command:

```
# touch –at 20300701000000 filename
```

The time stamp encoding the retention time has this format: [[CC]YY]MMDDhhmm[.ss]. It specifies the following date and time: 01.07.2013 00:00:00.

The file can be set to immutable by removing the write permissions with the following POSIX command:

```
# chmod a-w filename
```

A file in an immutable fileset is implicitly set to immutable once the above two steps have been performed. This means the IBM Spectrum Scale file attributes “immutable” and “expiration time” are set accordingly. To display the retention setting of the file the following POSIX command can be used:

```
# stat filename
File: ‘filename’
Size: 1               Blocks: 0          IO Block: 1048576
Device: 23h/35d Inode: 353793      Links: 1
Access: (0444/-r--r--r--)
Access: 2030-07-01 00:00:00.000000000 +0200
Modify: 2016-05-03 14:31:59.462278718 +0200
Change: 2016-05-03 14:32:16.915516097 +0200
```

The access pattern shows the permissions of the file indicating read-only. The access time stamp indicates the expiration time. The retention time can be extended using the same “touch” command as shown above[5].

[5] Requires IBM Spectrum Scale version 4.2.0.2 or higher.

## IBM Spectrum Scale commands

IBM Spectrum Scale allows applications and users to directly set the retention time and the immutability attribute using IBM Spectrum Scale commands. There is one IBM Spectrum Scale command with different parameters that can be used for this.

The retention time can be set using the IBM Spectrum Scale command:

```
# mmchattr –E 2030-07-01@00:00:00 filename
```

The time stamp encoding the retention time has this format: YYYY-MM-DD@hh:mm:ss

The attribute immutable can be set using the following IBM Spectrum Scale command:

```
# mmchattr –i filename
```

A file in an immutable fileset is implicitly set to immutable once the above two steps have been performed. The following IBM Spectrum Scale command can be used to display the file attributes:

```
# mmlsattr –L filename
```

The output looks like this. Notice the attribute “immutable” is set to “yes” and the expiration time is set to the time given with the mmchattr –E command:

```
#mmlsattr -L filename
file name:            filename
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            yes
appendOnly:           no
indefiniteRetention:  no
expiration Time:      Tue Jul  1 00:00:00 2030
flags:
storage pool name:    system
fileset name:         WORM
snapshot name:
creation time:        Tue Dec 21 21:18:58 2015
Windows attributes:   ARCHIVE
```

The retention time can be extended using the same “mmchattr –E” command as shown above.

## PowerShell

File immutability in regard to retention times and read-only setting can be managed using PowerShell. This requires that the immutable fileset or part of it is exported via SMB.

The retention time can be set using the PowerShell command:

```
# (dir filename).LastAccessTime = "2030-07-01 00:00:00"
```

The time stamp encoding the retention time has this format: YYYY-MM-DD hh:mm:ss

To set the file read-only use the following PowerShell command:

```
# (dir filename).Attributes = "ReadOnly"
```

The file in an immutable fileset is implicitly set to immutable once the above two steps have been performed. The following PowerShell command can be used to display the file attributes:

```
# (dir filename | select Name,LastAccessTime,Attributes
```

The output looks like this. Notice the attribute “ReadOnly” is set and the retention time is encoded in the last access date.

```
PS Z:\> dir filename | select Name,LastAccessTime,Attributes
Name           LastAccessTime             Attributes
----           --------------             ----------
filename       01.07.2030 00:00:00        ReadOnly
```

Note, retention time cannot be extended via SMB once the file has been set to read-only.

# Working with append-only files

The process to set files to append-only in an IBM Spectrum Scale immutable fileset is comprised of three steps:

- Setting retention time for the file
- Setting the file to read-only
- Setting the file to read-write

This process corresponds the SnapLock® method. There are two ways to set files to append-only:

- Using standard POSIX commands
- Using IBM Spectrum Scale command

Managing append-only files is only possible when accessing the immutable fileset from a cluster node, not through NFS or SMB.

When using POSIX command the file which is set to append-only must be an empty file. When using IBM Spectrum Scale command the last two steps of setting the file to read-only and read-write can be combined into one step.

The retention time can be extended but not reduced for append-only files stored in an immutable fileset configured in IAM mode “compliant”. Furthermore data can be appended to an append-only file. However an append-only file cannot be renamed, overwritten or deleted in a fileset configured in IAM mode “compliant”. An append-only file can be deleted if the retention time (relative to the system date and time) has expired. An append-only file can also be set to immutable removing the capability to append any data to it.

## POSIX commands

Setting files to append-only mode using POSIX commands is only possible from a cluster node accessing the immutable fileset directly, not through NFS or SMB.

The file to be managed in append-only mode must be empty with no content. Consequently the first step is to create an empty file and give it a retention time. This can be achieved with the following POSIX command:

```
# touch –at 20300701000000 filename
```

The time stamp encoding the retention time has this format: [[CC]YY]MMDDhhmm[.ss]. It specifies the following date and time: 01.07.2013 00:00:00.

The empty file can be set to append-only by removing and adding the write permissions with the following POSIX command:

```
# chmod a-w filename
# chmod +w filename
```

A file in append only mode cannot be deleted (unless its retention time has expired) or overwritten, but data can be appended to it. The retention time can be extended using the same "touch" command as shown above[6].

A file in an immutable fileset is implicitly set to append-only once the above two steps have been performed. Thereby the IBM Spectrum Scale file attributes "append-only" and "expiration time" are set accordingly. To display the retention setting of the file the following POSIX command can be used:

```
# stat filename
File: 'filename'
Size: 1               Blocks: 0          IO Block: 1048576
Device: 23h/35d Inode: 353793      Links: 1
Access: (0644/-rw-r--r--)
Access: 2030-07-01 00:00:00.000000000 +0200
Modify: 2016-05-03 14:31:59.462278718 +0200
Change: 2016-05-03 14:32:16.915516097 +0200
```

The access pattern shows the permissions of the file indicating read-write for the user. The access time stamp indicates the expiration time. There is no easy way to see that the file is append-only since this is not a file state in POSIX. The retention time can be extended using the same "touch" command as shown above[7].

When no more data has to be appended to the file it can be set to immutable using the following POSIX command:

```
# chmod a-w filename
```

Consequently this last step will set the attribute "immutable" to "yes" and the attribute "appendOnly" to "no" and disallow any appends. In IBM Spectrum Scale the attribute "immutable" is set implicitly. . To display the retention setting of the file the following POSIX command can be used:

```
# stat filename
File: 'filename'
Size: 1               Blocks: 0          IO Block: 1048576
Device: 23h/35d Inode: 353793      Links: 1
Access: (0444/-r--r--r--)
Access: 2030-07-01 00:00:00.000000000 +0200
Modify: 2016-05-03 14:31:59.462278718 +0200
Change: 2016-05-03 14:32:16.915516097 +0200
```

This method is similar to the method used with SnapLock®. However IBM Spectrum Scale does not support managing files in append-only mode via NFS or SMB.

[6] Requires IBM Spectrum Scale version 4.2.0.2 or higher.
[7] Requires IBM Spectrum Scale version 4.2.0.2 or higher.

## IBM Spectrum Scale commands

One fundamental difference between POSIX and IBM Spectrum Scale commands for managing append-only files is that with IBM Spectrum Scale commands the file does not have to be empty in order make it append-only. Consequently a normal file can be created and filled up with content before it is set to append-only. Prior to setting a file to append-only it is recommended to set a retention time using the following IBM Spectrum Scale command:

```
# mmchattr –E 2030-07-01@00:00:00 filename
```

The time stamp encoding the retention time has this format: YYYY-MM-DD@hh:mm:ss

Now the file can be set to append-only using the following IBM Spectrum Scale command:

```
# mmchattr –a yes filename
```

A file in append only mode cannot be deleted (unless its retention time has expired) or overwritten, but data can be appended to it. The retention time can be extended using the same "mmchattr –E" command as shown above[8].

A file in an immutable fileset is implicitly set to append-only once the above two steps have been performed. The following IBM Spectrum Scale command can be used to display the file attributes within the IBM Spectrum Scale file system:

```
# mmlsattr –L filename
```

The output looks like this. Notice the attribute "appendOnly" is set to "yes" and the Expiration time is set to the time encoded in the last access date.

```
#mmlsattr -L filename
file name:            filename
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            no
appendOnly:           yes
indefiniteRetention:  no
expiration Time:      Tue Jul  1 00:00:00 2030
flags:
storage pool name:    system
fileset name:         WORM
snapshot name:
creation time:        Tue Dec 21 21:18:58 2015
Windows attributes:   ARCHIVE
```

When no more data has to be appended to the file it can be set to immutable using the following IBM Spectrum Scale command:

```
# mmchattr –i yes filename
```

Consequently this last step will set the attribute "immutable" to "yes" and make the file immutable. The IBM Spectrum Scale command (mmlsattr –L filename) will reflect this:

```
#mmlsattr -L filename
```

[8] Requires IBM Spectrum Scale version 4.2.0.2 or higher.

```
file name:            filename
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            yes
appendOnly:           yes
indefiniteRetention:  no
expiration Time:      Tue Jul  1 00:00:00 2030
flags:
storage pool name:    system
fileset name:         WORM
snapshot name:
creation time:        Tue Dec 21 21:18:58 2015
Windows attributes:   ARCHIVE
```

Note, it is normal that at this point the append-only and immutable attributes are set to yes. It is not possible to reset the append-only attribute before setting the immutable attribute.

## Indefinite retention

Immutable files can also be configured with an indefinite retention time. Indefinite retention essentially defines an indefinite retention time for the immutable file. It can be set and reset by the user who has permissions to change file attributes using an IBM Spectrum Scale command. It cannot be set with POSIX commands.

Indefinite retention does not require retention times to be set. If a retention time and indefinite retention is set on a file then the file cannot be deleted even if the retention time has expired. If the indefinite retention is remove from a file the underlying retention time take precedence. If this has expired or was not set then the file can be deleted.

One use case for indefinite retention is the implementation of a deletion hold function. A deletion hold does not allow the deletion or expiration of a file even if the retention time is expired. It provides an additional level of protection above the retention time. For example, if an application or user wants to prevent a file from expiring it can set the indefinite retention attribute. To release the deletion hold the indefinite retention attribute can be reset. If the indefinite retention is reset (set to “no”) the actual retention time of the file takes precedence. If this is expired the file can be deleted, if not the file cannot be deleted.

To set indefinite retention for an immutable file the following IBM Spectrum Scale command can be used:

```
# mmchattr –-indefinite-retention yes filename
```

To reset indefinite retention issue the command:

```
# mmchattr –-indefinite-retention no filename
```

Indefinite retention is only useful if the file is already set to immutable or append-only. Otherwise it will not have an effect.

# Working with directories

Directories within an immutable fileset cannot be set to immutable (or append-only) explicitly. Directories within an immutable fileset become immutable automatically as soon as files stored within this directory. An immutable directory cannot be renamed or deleted unless there are no files within the directory. An empty directory (not containing any files) can still be deleted and renamed.

This differs from the SnapLock® mode where even empty directories can be set to immutable. But what is the purpose of setting empty directories to immutable?

# Limitations

In this section some limitations with IBM Spectrum Scale immutable filesets are explained that also influence the compatibility with SnapLock®.

## Directories

Most of the POSIX file system commands in an immutable fileset have the same effect as with the standard SnapLock® method invented by NetApp Inc. [2]. One of the key differences from the standard SnapLock® method is the concept of immutable directories. With the standard SnapLock® method directories can be explicitly set to immutable which applies limitations to file operations within this directory. In IBM Spectrum Scale immutable fileset a directory becomes immutable implicitly and as soon as a file is immutable in the directory. Immutable directories within an immutable fileset cannot be deleted and renamed.

## NFS exports and SMB shares

An immutable filesets can also be exported via NFS or SMB. Some of the limitations are:

- It is not possible to set and manage files in append-only mode
- Extended attributes (POSIX) cannot be set via NFS

Note, with IBM Spectrum Scale version 5.0.3.2 and above it is possible to extend retention period and delete expired files over SMB. This requires IAM mode of “compliant-plus”. With prior Spectrum Scale versions these operations are not possible.

## Default retention times

An immutable IBM Spectrum Scale fileset does not support minimum, default and maximum retention times as it is available with NetApps’ SnapLock®. It could however be implemented using IBM Spectrum Scale policies as explained in section Automatically setting files to immutable.

## IBM Spectrum Scale limitations

When an immutable fileset is managed within the IBM Spectrum Scale cluster some limitations apply:

 

- Immutable filesets are not supported for AFM cache, primary or secondary filesets
- Non-empty immutable filesets configured in IAM mode “compliant” cannot be deleted using the mmdelfileset command
- File systems cannot be deleted if the cluster-wide parameter “indefiniteRetentionProtection=yes” has been set, regardless if the file system contains an immutable fileset or not.
- When copying an immutable file within the immutable fileset then the immutability attributes are not retained. In order to make such file immutable it is necessary to set the file to read-write (chmod +x) and read-only (chmod -w) and set the expiration time (touch -at).

# System configuration guidance

After explaining how to configure and manage immutable files within IBM Spectrum Scale immutable filesets, this section provides guidance addressing IT security aspects. IT security aspects must be considered for the entire solution including the application and user creating and managing immutable files, the IBM Spectrum Scale cluster and file systems storing immutable files, the operating system running on the cluster nodes, the storage system where data is stored and the infrastructure environment. Figure 1 gives an overview of the key components that are considered from an IT security perspective:

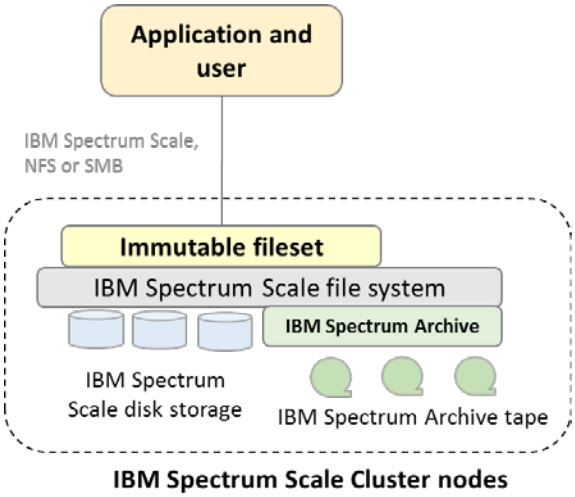


Figure 1: IBM Spectrum Scale architecture with immutable filesets

The IBM Spectrum Scale cluster is configured with file systems and immutable filesets. The application and user access the immutable file set via the IBM Spectrum Scale file system client, NFS or SMB to store and retrieves files and manages their immutability. The files are initially stored on IBM Spectrum Scale disk storage. Optionally the files stored on disk can be migrated to tapes formatted in the Linear Tape File system format (LTFS) by the Spectrum Archive Enterprise Edition software.

Configuration guidance provided in this chapter apply to the application and user, the IBM Spectrum Scale cluster node configuration and administration, the IBM Spectrum Scale storage, the Spectrum Archive Enterprise Edition software and WORM tapes and the operating system of the servers used for IBM Spectrum Scale, Spectrum Archive and by the user and application. This guidance is of an abstract nature suggesting an appropriate security configuration and setup for IBM Spectrum Scale and Spectrum Archive. The client organization should leverage security architects to understand the implication of changes to their security policy and review their security architecture from end to end. This document does not assure that the system cannot be compromised by leveraging exploits nor is it intending to imply any specific audit guarantees (see Disclaimer).

For more detailed technical guidance regarding security related configuration aspects refer to the IBM Spectrum Scale Security paper [14]. In addition the general recommendations provided by the external auditor have been considered [10].

# User and Application

The user and application - subsequently referred as applications - use the immutable filesets to read and write files and make files immutable with appropriate retention times. The application can access immutable filesets via the IBM Spectrum Scale file system client, NFS or SMB. From the application perspective the following guidance should be considered:

- The application server should use the same time source for automated time synchronization as the IBM Spectrum Scale cluster hosting the immutable filesets.
- The application server should be hardened to disallow tampering with data and metadata. Consider the guidance provided for IBM Spectrum Scale administration and Operating system configuration for the application as well.
- The use of mmap (mapping files in memory) is not supported with files stored in immutable filesets.
- The application should set the file retention time and immutability in a timely manner (immediately after storing a file).
- The application should track files where retention was not set (e.g. due to failures) and set it later.
- The application should manage retention within GPFS file system (GPFS client) using either POSIX or GPFS commands. Using GPFS command `mmchattr -i | -E | -a` is recommended.
    - Managing retention via NFS or SMB has some limitations (see section Limitations)
- The IBM Elastic Storage Server (ESS) provides internal check summing for data stored in immutable filesets. When not using ESS as storage for immutable files consider calculating and storing checksums with immutable files upon file creation. Checksums can be stored in IBM Spectrum Scale extended attributes (such as `user.cksum=`*`checksum:value`*) using the command: `mmchattr --set-attr user.chksum=`*`checksum:value filename`*. Periodically validate the checksums stored in extended attributes and create and retain an audit protocols for this validation process. The audit trail shall be stored in an immutable manner for the retention time of immutable files.
- Implement authentication and authorization according to the organizational standards and the methods supported with IBM Spectrum Scale [16].

# IBM Spectrum Scale

The IBM Spectrum Scale file system provides immutable filesets allowing to retain files in an immutable manner. This function has been assessed for compliance by an independent auditor [10]. The immutability is managed by the application and IBM Spectrum Scale enforces it. Consider the following guidance for IBM Spectrum Scale configuration and administration.

## Configuration

IBM Spectrum Scale provides a comprehensive set of IT security configuration and functions [14]. In this section we elaborate on some important setting that should be considered in combination with immutable files:

- Configure immutable filesets in compliant mode (`iam-mode=compliant`). Filesets configured in compliant mode cannot be set to non-compliant or advisory mode.

- Disallow file system deletion by setting the cluster-wide parameter using the command: `mmchconfig indefiniteRetentionProtection=yes`. Once this parameter is set it is not possible to delete file systems in the cluster. In addition this parameter cannot be reset.
- Implement audit logging for administrative IBM Spectrum Scale commands using the command: `mmchconfig commandAudit=yes|syslogOnly`. With this setting the IBM Spectrum Scale administrative command can be logged in the IBM Spectrum Scale log file and / or to the syslog file [17]. The audit log file must be kept in an immutable manner for the duration of the retention time of the files stored in immutable filesets.
- Implement file audit logging using the audit logging function provided with IBM Spectrum Scale version 5.0 and higher [23]. For prior IBM Spectrum Scale version use the audit logging function for NFS and SMB provided by Varonis [11]. With file audit logging all major file operations such as open, close, rename, ACL and attribute changes as well as deletions are logged into audit logs. The audit logs are stored as immutable files residing in immutable filesets of an IBM Spectrum Scale file system. The retention time of audit log files can be configured.
- When migrating files to an external pool ensure that these are stored on a WORM medium, like WORM tapes.
- Implement business continuity (leveraging quorum, replication, backup and clustering) based on SLAs.
- Implement IBM Spectrum Scale file encryption when required by the security policies [12].

# Administration

## Secure administration with sudo wrappers

IBM Spectrum Scale does not have its own user management for user using the command line interface (CLI). CLI users are authenticated by the operating system and have privileges to perform IBM Spectrum Scale specific commands. By default the IBM Spectrum Scale cluster requires root privileges for administration. The IBM Spectrum Scale sudo wrappers eliminate the need to allow ssh access for the root user to other nodes. In addition it allows using named users (non-root) to administer the cluster by leveraging sudo definitions [4].

Note: The use of IBM Spectrum Scale sudo wrappers does not guarantee that named users cannot elevate their privileges. Additional operational measures are required according to the security policies deployed within the organization.

Please note, that IBM Spectrum Scale sudo wrappers are currently not supported with:

- Cluster Export Services, NFS, SMB and Object
- Installation toolkit (command: `spectrumscale`)
- IBM Spectrum Scale call home
- With Windows nodes in the cluster

Users administering a cluster containing immutable filesets should not be able to tamper immutable files. Therefore it might be appropriate to allow only commands (IBM Spectrum Scale and operating system) that are required for normal operations for these users. In special cases the privileges of these users can be temporarily

increased by another user. This allows for the implementation of the four-eye principle where the administrative user requires the authorization of another user (security administrator) to perform special task.

For the implementation of the four-eye principle create two user groups and assign named users to each of these groups in the operating system. In this example one group is named *gpfsadmin* and another group is named *secadmin*. The following principles should apply:

- Users in the group *gpfsadmin* should be able to administer IBM Spectrum Scale and Spectrum Archive during normal operations.
- Users in the group *secadmin* should be able to manage the privileges of the users in the group *gpfsadmin*, for example to temporarily allow additional commands in case of problems. For this purpose the user in the group *secadmin* may leverage sudo definitions.

The four-eye principle allows users of the group *gpfsadmin* to execute a limited set of commands that are required for normal operations. If a user of the group *gpfsadmin* requires more privileges (e.g. due to a problem) he can contact a user of the group *secadmin*. The user of the group *secadmin* can now grant the user of the group *gpfsadmin* further privileges temporary by changing the sudo definitions.

Configure IBM Spectrum Scale sudo wrappers according to the instructions. Allow users in group *gpfsadmin* to run commands that are required to administer the IBM Spectrum Scale cluster, file systems and Spectrum Archive. Use the default configuration in the `sudoers.sample` file, adjust and test this according to the needs. Find an example of the sudo definitions in section Sudo group example. In general consider the following guidance

- Users in the group *gpfsadmin* should not be able to run all commands; instead limit this to the IBM Spectrum Scale and Spectrum Archive commands that are required.
- Users in the group *gpfsadmin* should not be able to obtain full root permission without being authorized by a user of the group *secadmin*.
- Users in the group *gpfsadmin* should not be able to manage sudo configuration.
- Users in the group *gpfsadmin* should not be able to change the time or time related configurations in the operating system.
- Do not alter the set of commands for the group *gpfsadmin* in the sudo definitions that have to run without password.
- Allow users in the group *secadmin* to change the sudo configuration (visudo or the like). This allows the user of the *secadmin* group to temporary allow certain commands to the user in the *gpfsadmin* group.
- Configure logging of all commands run by all user groups in the sudo context (e.g. by configuring the sudo parameters `logfile` and / or `iolog_dir` in the sudoers file). These logs shall be immediately send to a remote log server and kept for the retention time of the files.

The IBM Spectrum Scale GUI has a separate user management. It is recommended to use the same user names in the GUI and the CLI for the same user. In addition the roles of the user in GUI should match the roles the user has in CLI.

When required disable root logon via SSH (`PermitRootLogin No` in sshd_config)

 

## Monitoring

Monitoring assures that resource overload and system misbehavior is detected in time. It is highly recommended to configure event notifications via the IBM Spectrum Scale GUI and send the appropriate events to the administrators via email or SNMP. Especially monitor file system capacities and ensure sufficient capacity and inodes are available at all times. For more information regarding IBM Spectrum Scale monitoring consider these [21].

Alternatively the IBM Spectrum Scale REST API can be used to monitor the cluster [22].

## Access control

Accessing immutable files in the immutable fileset must be controlled by file permissions and access control lists. This is typically done by the application that creates and works with files.

In general an IBM Spectrum Scale administrator - users in group *gpfsadmin* and *secadmin* - should not be able to access immutable files, unless the business process requires this. Do not use the GUI to manage access to immutable filesets since this may allow an administrator to gain data access privileges.

## Backup

IBM Spectrum Scale offers the capability to backup immutable files to the Spectrum Protect server using the command: `mmbackup` [8]. It is recommended to backup immutable files. The backup process should be monitored and failures should be corrected in a timely manner.

Immutable files must be restored to immutable filesets, otherwise the immutability attributes (immutable, append-only and retention time) are lost for the file that has been restored. These attributes however remain in the backup copy stored in the IBM Spectrum Protect server.

Note, immutable filesets cannot be replicated using IBM Spectrum Scale Active File Management (AFM) in any AFM mode.

## Manage file immutability

Setting files to immutable is either managed by the application or automatically using policies with external scripts (see section Automatically setting files to immutable). It should be periodically assured that files within an immutable fileset are set to immutable with the appropriate retention times. For this purpose LIST policies can be used to find files that are not set to immutable (see section List policy example). The administrator should be notified if there are files in an immutable fileset that are not set to immutable. The check for non-immutable files should be performed periodically and create an audit trail that includes the name of the files that are not immutable. This audit trail must be preserved in an immutable manner for the duration of the retention time of the immutable files.

When required implement file level encryption by leveraging the IBM Spectrum Scale encryption function [12].

When required immutable files can also be migrated to WORM tapes using Spectrum Archive Enterprise Edition.

## Operating system configuration

The operating system comes into play for all IBM Spectrum Scale cluster nodes as well as on application servers accessing the cluster via IBM Spectrum Scale, NFS or SMB. Find below some abstract guidance for securing operating systems:

- Configure NTP or similar time synchronization method.
- Control and restrict access to remote time server as well as the client.
- Configure password rules (length, complexity, expiration) according to security standards.
- Implement system log message forwarding (e.g. rsyslog). Store the system log messages in an immutable manner for the duration of the retention time. Note, the IBM Spectrum Scale commands should be logged to the system log (parameter `commandAudit=yes|syslogOnly`).
- Capture the sudo audit logs and forward these to a remote location (similar to rsyslog). These logs shall be kept in an immutable manner or the duration of the retention time of files.
- Capture operating system commands executed by the users in groups *gpfsadmin* and *secadmin*. Forward these logs to a remote location (similar to rsyslog). These logs shall be kept in an immutable manner or the duration of the retention time of files.
- Periodically gather GUI logs (`/var/log/cnlog`) and send them to remote location (similar to rsyslog). These logs shall be kept in an immutable manner or the duration of the retention time of files.
- Disable unnecessary TCP ports and services (see [19]) using the firewall.
- Implement patch management to immediately address vulnerabilities.
- Periodically audit the sudo environment, in particular the sudo configuration. Audit trails must be kept in an immutable manner for the duration of the file retention time.
- Harden the operating system according to client standards and test this with IBM Spectrum Scale.

## Storage system

The storage is used by IBM Spectrum Scale to store immutable files on disk (hard disk, flash, SSD). IBM Spectrum Scale can use different kind of storage:

- Elastic Storage Server (ESS) with Native RAID functionality (GNR): The ESS I/O nodes are IBM Spectrum Scale cluster nodes and the underlying disks are managed by the IBM Spectrum Scale GNR software. Protection can be established by protecting the IBM Spectrum Scale nodes (see section IBM Spectrum Scale Administration and Operating system configuration).
- Internal disk within the NSD server: The measures outline for the operating system configuration help to protect the internal storage (see section Operating system configuration).
- External storage systems: The disk system is external to the IBM Spectrum Scale nodes and attached via SAN, LAN or Infiniband. It may require addition configuration (see below).

Consider the following guidance for the configuration of the external storage system:

- Administrators of the external storage system should be named users.
- Role based access control for administrators should be enforced.
- Storage system administrators should be different from IBM Spectrum Scale administrators.
- The access to the external storage system should be audited and the audit trails should be stored in an immutable manner for the duration of the file retention time.
- The administrative network (typically LAN) should be encapsulated and isolated from other administrative networks.
- The administration of the storage system should leverage secure authentication and messaging (https, TLS).
- Data network (SAN, LAN and Infiniband) might be encapsulated and isolated from other data networks depending on the general security practices.
- Data should be encrypted over the data network when required.
- Physical access to the storage system should be restricted and audited (data center access) and audit trails should be stored in an immutable manner for the duration of the retention time.

# IBM Spectrum Archive EE

In deployments where IBM Spectrum Archive Enterprise Edition (EE) is used the IBM Spectrum Archive software runs on a subset of IBM Spectrum Scale cluster nodes. The IBM Spectrum Scale cluster node configuration has been discussed above (see sections IBM Spectrum Scale configuration and IBM Spectrum Scale Administration and Operating system configuration).
Find below some guidance to configure Spectrum Archive EE in conjunction with IBM Spectrum Scale:

- Enable Logical Block Protection which will validate the data on the flight and at rest using CRC or Red Solomon algorithm [18]. Please note that LBP has performance impacts, which is higher with Reed Solomon (approx. 80%) than with CRC32 (approx. 12 %, measured with TS1150 drives)
- Assure that the administrator for IBM Spectrum Archive EE does not use direct root login. IBM Spectrum Scale sudo wrappers can be used to enable administration of Spectrum Archive EE (see section IBM Spectrum Scale Administration).
    - Prevent access of administrators to the underlying LTFS file system (/ltfs).
- Immutable files should be migrated to WORM tapes. If a non-immutable file is migrated to a WORM tape and later on the file is changed then it cannot be migrated to the same WORM tape pool again.
- Do not use reclamation and reconciliation in conjunction with WORM tape pools. This is not supported.
- Normal export and import of LTFS tapes are not supported in conjunction with WORM tape pools. Offline export is supported, but it must export a copy tape and never the primary tape. Hence offline exporting requires two copies of files.
- Implement two copies of files on two tape in different libraries and fire zones when possible.
- Since the dual copy function in Spectrum Archive is not a backup consider implementing the IBM Spectrum Scale backup function (mmbackup) in conjunction with Spectrum Protect.

Find some more guidance using Spectrum Archive EE in combination with WORM tapes in the Spectrum Archive Redbook [20] chapter 7.21.

# WORM tape

When IBM Spectrum Archive EE is used within the solution then WORM tapes should be used to store migrated files. The IBM TS1100 WORM tape technology has also been assessed for compliance [15]. Find below some guidance for configuring WORM tapes:

- Physical access to the tape library should be restricted and audited. The audit trails should be stored in an immutable manner for the duration of the retention time of files.
- Configure Library Managed tape encryption in case the network (SAN) between the server and the WORM tape drive is not secure.

It is important to use WORM tapes and not standard read-write tapes in combination with immutable files.

# Environment

In this section some general guidance is provided for the IT environment of the IBM Spectrum Scale solution.

## Time server

The time server used for time synchronization of the IBM Spectrum Scale nodes and the application nodes should be identical. In addition this time server should not be configurable by IBM Spectrum Scale or application administrators. Any manipulation of the time of the time server must be prohibited.

In addition the IBM Spectrum Scale and application administrators should not be able to manipulate the time service client (ntpd, chronyd) on the IBM Spectrum Scale and application nodes. This can be accomplished with the proper sudo configuration.

## Networks

The networks between application server and IBM Spectrum Scale nodes as well as the cluster network between the cluster nodes and the network to disk and tape (SAN) must be secure. If this is not the case consider using additional network security techniques such as TLS or VPN or end-to-end encryption.

For this purpose IBM Spectrum Scale provides secure authentication and data transfer between the cluster nodes. The level of security can be configure with the configuration parameter `cipherList`. By default this parameter is set to `AuthOnly` which means that the authentication between nodes is protected. The parameter can also be set to a cipher which causes the authentication between nodes and the data transfer to be secured. A list of supported ciphers can be shown using the command: `mmauth show ciphers`

# Use cases

Managing immutable files is typically done for archiving purposes when the regulatory requirements demand storing files in an immutable manner. An archive system is composed of an archive managing system and the archive storage. The archive management system is an archiving application driving the archiving process by collecting files to be archived from different sources (email servers, ERP systems, Databases, file servers, etc.), indexing the content and storing files in the archive storage. The archive storage is responsible to store the files in an immutable manner. The archive storage is represented by IBM Spectrum Scale immutable filesets in the context of these use cases.

In this chapter two use cases for IBM Spectrum Scale immutable files are presented. In the first use case the archiving application supports the immutability function provided by IBM Spectrum Scale and manages immutability of files (see section Application supporting immutable filesets).

In the second use case the archiving application does not support the immutability function to manage immutable files. It stores the files in an immutable fileset and IBM Spectrum Scale is configured to automatically set files to immutable in the background. This is based on IBM Spectrum Scale policies in combination with a custom script that periodically identify files and set the retention time and immutability attribute for these files (see section Automatically setting files to immutable). In order to understand the functionality of this use case knowledge about the IBM Spectrum Scale policy engine is required [5].

Finally additional functions provided by IBM Spectrum Scale and relevant for archiving are discussed. These functions address requirements for high availability, disaster protection and tiered storage (see section Additional IBM Spectrum Scale archiving functions).

## Application supporting immutable filesets

Applications supporting the standard SnapLock® method invented by NetApp Inc. via the file system interface might support the immutability function in IBM Spectrum Scale, because this is works similar to the SnapLock® method. However, it is important to test and certify any application with IBM Spectrum Scale immutability prior.

As shown in figure 2 the archiving application accesses the immutable fileset configured in an IBM Spectrum Scale file system to store files and manage file-level immutability. The application can either run on a server that is an IBM Spectrum Scale cluster member or it can access the immutable fileset through a NFS export. In the first case the application can directly access the immutable fileset and use either POSIX or IBM Spectrum Scale commands to manage immutable files. In the second case the application can use POSIX commands to manage immutable files.

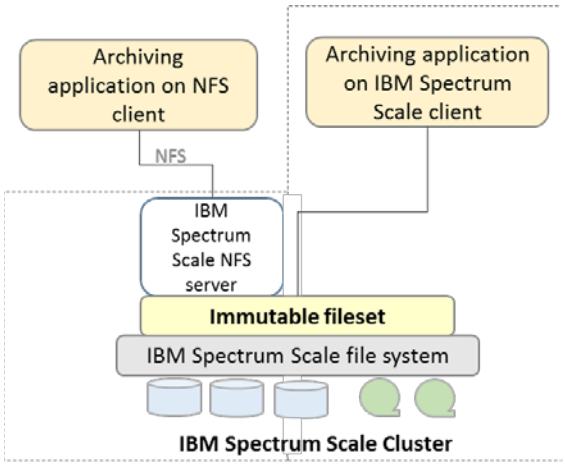

Figure 2: Architecture with application using IBM Spectrum Scale immutable filesets

After the application has stored files in the immutable fileset it makes the file immutable by setting the retention time and immutability using POSIX or IBM Spectrum Scale commands (see section Working with immutable files). IBM Spectrum Scale assures that immutable files cannot be changed or deleted. During the lifecycle of the file the application can extend the retention time. The application typically tracks the retention time of files in its own index database and if a file has expired the application deletes the file. Alternatively the application reads the last access data of the file which encodes the retention time in IBM Spectrum Scale in order to decide whether the file can be deleted.

In a similar way applications can manage append-only files (see section Working with append-only files). The only difference is that the application cannot manage append-only files via NFS. Hence the application must run on an IBM Spectrum Scale client (see figure 2).

The archive application can also leverage indefinite retention feature. For example when the application needs to prevent a file from being expired it can set the indefinite retention for the file, just like a deletion hold (see section Indefinite retention). This assures that the file remains immutable, even if the retention time expires. Setting indefinite retention requires the application to run on an IBM Spectrum Scale client because it can only be set and reset with IBM Spectrum Scale commands.

While this example shows the application connected to the IBM Spectrum Scale cluster as cluster member or via NFS, it can also be connected via SMB. IBM Spectrum Scale also provides file sharing via SMB. Setting retention times and immutability via SMB however requires different sets of commands or the use of UNIX toolkits, because SMB is not POSIX compatible [9].

# Automatically setting files to immutable

In this section a solution is presented to set files to immutable with a pre-defined retention time using the IBM Spectrum Scale policy engine. In order to understand the functionality behind this use case some basic knowledge about IBM Spectrum Scale policies is required [5].

Applications that do not support managing immutable files in an IBM Spectrum Scale immutable fileset can still benefit from the immutability function. Similar to the architecture presented in figure 2 the archiving application connects to the immutable fileset via NFS or SMB or as IBM Spectrum Scale client directly.

The archive application stores file in the immutable fileset of the IBM Spectrum Scale file system. The IBM Spectrum Scale cluster is configured to automatically set files to immutable by leveraging the IBM Spectrum Scale policy engine. The policy engine can be configured to run periodically (for example every 30 minutes), possibly using the cron-daemon.

The policy engine in IBM Spectrum Scale allows to identify files based on file attributes and processes these files with custom scripts. The identification criteria for files and the script processing these files are defined in a set of rules, also called policy. For the purpose of this use case all non-immutable files in a given immutable fileset need to be identified and processed. The processing executed by a custom scripts sets the pre-defined retention time and make the files immutable.

Accordingly two rules are required. The first rule defines the name of the custom script to be executed for all identified files. In this example the script is named "makeimmutable.sh"[9]:

```
RULE EXTERNAL LIST 'makeworm' EXEC 'makeimmutable.sh'
```

The second rule describes the identification criteria of files that need to be processed by the custom script "makeimmutable.sh". The identification criteria is to select all files in the immutable fileset "archive" where the extended attribute "immutable" is not set:

```
RULE ‘notworm' LIST ‘makeworm‘ FOR FILESET (‘archive’)
WHERE NOT (MISC_ATTRIBUTES LIKE '%X%')
```

For testing purposes these two rules can be written to a file (policyfile.txt). Replace the token `EXEC ‘makeimmutable.sh’` by `EXEC ‘’` and execute these rules using the policy engine in test mode:

```
# mmapplypolicy <filesystem-name> -P <policyfile.txt> -I test
```

This will show a list of files identified without processing these.

Now the custom script needs to be implemented. This script (makeimmutable.sh) is invoked by the policy engine with two arguments:

- $1 is the policy operation which can be TEST or LIST
- $2 for the operation TEST this is the file system name, for the operation LIST this is the name of the policy file including all selected files.

Consequently the custom script needs to parse these two arguments and act appropriately. For the TEST operation it should check if the file system given with the second argument exists. For the LIST operation a file list is passed which can be processed. Here is some pseudo code for this:

```
# pseudo code to for an external script setting files to immutable
# default retention time is 1 year
Def_ret=365
case $1 in
```

[9] The script name must be specified with the full qualified path name in the EXTERNAL LIST rule.

```
TEST )
  # $2 is the file system path, check if this exists.
  if [ $2 exists ] then
    rc=0
  else
    rc=1
  fi
  ;;
LIST )
  # $2 is the name of the policy file
  # process the file list
  for each line in $2
  do
    extract_the_filename()
    if [ filename exists ] then
      set retentiontime: mmchattr –E $(current + $def_ret) filename
      set immutable: mmchattr –i yes filename
    fi
  done
  rc=0
  ;;
exit $rc
```

The file list passed to the custom script by the policy engine consists of multiple line. Each line has five fields and contains one file name. Here is an example for one line with five fields:

```
      48900 1741777473 0   -- /mnt/filesystem/file1
```

The first three fields are GPFS internal numbers (inodenumber, inodegeneration, snapid). The fourth field is meaningless for this purpose. The file name is the 5th field in the file list. The function to extract the file name has to parse this file list line by line and extract the file name. Be aware that the file name can include blanks.

## Assigning different retention times based on file types

The retention time being set for each non-immutable file can also be different for files with different characteristics, for example different file types. This requires some minor changes to the policy and to the custom script.

Let's assume the retention time for files ending with ".tiff" should be set to 5 years and the retention time for files ending with ".pdf" should be set to 7 years. In order to pass the retention time based on file type to the custom script the EXTERNAL LIST rule can be configured with an optional parameter denoted by the name: OPTS 'years'. The option "years" is an integer number specifying the number of years to retain the specific file types. Four rules are required for this, two for teach file type. The example below also includes some macros which make it easier to read rules:

```
/* define some macros */
define( exclude_list, (PATH_NAME LIKE '%/.SpaceMan/%' OR PATH_NAME LIKE
'%/.snapshots/%' OR NAME LIKE '%mmbackup%' ))
define( immutable, MISC_ATTRIBUTES LIKE '%X%')

/* rule to set .tiff files to 5 years retention
RULE EXTERNAL LIST 'settiff' EXEC 'makeimmutable.sh' OPTS '5'
```

```
RULE 'mp3' LIST 'settiff' FOR FILESET ('archive') WHERE NOT
(exclude_list) and NOT (immutable) and (NAME LIKE '%.tiff')

/* rule to set .pdf files to 7 years retention
RULE EXTERNAL LIST 'setpdf' EXEC 'makeimmutable.sh' OPTS '7'
RULE 'pdf' LIST 'setpdf' FOR FILESET ('archive') WHERE NOT
(exclude_list) and NOT (immutable) and (NAME LIKE '%.pdf')
```

The custom script obtains a third argument from the policy engine which is the value encoded in the OPTS clause. In our case this is an integer number specifying the number of years to retain a file of a certain type. Find below some pseudo code implementing the variable retention time by file type:

```
# pseudo code to for an external script setting files to immutable
# default retention time is 1 year
Def_ret=365
case $1 in
TEST )
  # $2 is the file system path, check if this exists.
  if [ $2 exists ] then
    rc=0
  else
    rc=1
  fi
  ;;
LIST )
  # $2 is the name of the policy file
  # $3 is the retention time, if not set give it a default time
  if [ $3 is not set ] then
    retime=$Def_ret
  else
    retime=$3
  fi
  # process the file list
  for each line in $2
  do
    extract_the_filename()
    if [ filename exists ] then
      set retentiontime: mmchattr –E $(current + $retime) filename
      set immutable: mmchattr –i yes filename
    fi
  done
  rc=0
  ;;
exit $rc
```

The example above just gives an idea how retention times can be flexibly assigned to files based on their attributes. The assignment of retention times can be based on file attributes such as file type (extension), path name, size, user ID owning the file, extended user attributes, etc. (see [5] and [6]).

## Additional IBM Spectrum Scale archiving functions

Archiving is characterized by medium to large volumes of data that have to be kept for long period of time. During this time access to the files should always be possible even if an unwanted situation like a disaster has occurred. Laws and regulations may demand to prevent deletion and changes of data during the retention time. Because of the long lifetimes of archived files it is important to manage cost for operations, power and cooling. The IBM Spectrum Scale cluster can be configured to provide high availability, disaster protection, tiered storage, compression, encryption and regulatory compliance of archived data.

*High availability* can be achieved by leveraging IBM Spectrum Scale synchronous replication in combination with intelligent quorum techniques [7]. IBM Spectrum Scale Quorum techniques assure that the cluster remains online even if a subset of cluster nodes has failed. Synchronous replication copies files to two or three storage systems attached to the IBM Spectrum Scale cluster. In combination with multiple file system descriptor disk the synchronous replication feature in IBM Spectrum Scale allows access to all data even if one storage system has failed.

*Disaster protection* can be achieved by leveraging the integrated IBM Spectrum Scale backup function in combination with Spectrum Protect (formerly known as Tivoli Storage Manager, TSM). This function uses the “mmbackup” command to quickly create copies for immutable files in the IBM Spectrum Scale server. The backup copies can be stored in a disk pool, a deduplicated disk pool, in the cloud or directly on WORM tape.

*The tiered storage* function in IBM Spectrum Scale allows to place files on the most appropriate storage technology during the entire lifecycle. This function is important for archiving, especially for large volume of data that have to be retained for long periods of time because it provides optimal total cost of ownership over the lifetime of data. For examples archived files can be stored on a first tier of storage for a first period of time where access to the files is likely. When access to files diminishes or disappears files can be transparently migrated to the next tier of storage. Using tape as a next storage tier helps to reduce cost for maintenance, power consumption and cooling.

IBM Spectrum Scale offers the *hierarchical storage management function* in combination with IBM Spectrum Protect™ for Space Management (TSM HSM) or IBM Spectrum Archive Enterprise Edition, to identify and migrated files to tape. File identification is based on file attributes such as file types, access time and file size. Files that have been identified are migrated to the next storage tier where they can be stored on WORM tape. Access to file in the IBM Spectrum Scale immutable fileset remains transparent. This means the application or user still sees the migrated file and upon access the file is fetched from tape by the WORM tape.

*File encryption* only allows authorized users to read the content of files and is one of the important data security function in IBM Spectrum Scale [12]. Data encryption supports stringent requirements for data security that are demanded for example by the Payment Card Industry.

*File compression* provides storage capacity optimization and is another value adding function provided by IBM Spectrum Scale [13]. File compression can be controlled on an individual file basis. In the context of archiving compression is useful for data that

does not change and has to be kept for long period of time in order to archive cost savings.

*File and command audit logging* allow to create comprehensive audit trails. File audit logging introduces with IBM Spectrum Scale Version 5.0 audits file operations and stores the resulting audit logs in an immutable fileset. Command audit logging allows to send all commands causing changes to the IBM Spectrum Scale cluster configuration to the syslog and from there to a remote log server. These audit trails can be used to determine changes to files and the IBM Spectrum Scale cluster configuration.

*Regulatory compliance* can be achieved leveraging immutable filesets. This functionality has been assessed for compliance by KPMG AG in accordance to US regulations (SEC17a-4f) as well as German and Swiss tax and trade laws [10].

# Appendix

## List policy example

The list policy example shows how to identify immutable files. It might need further adjustments for particular environments.

```
/* define macros */
define( exclude_list, (PATH_NAME LIKE '%/.SpaceMan/%' OR PATH_NAME LIKE
'%/.snapshots/%' OR NAME LIKE '%mmbackup%' OR PATH_NAME like ‘%working-
directory/%’ ))
define( immutable, MISC_ATTRIBUTES LIKE '%X%')

/* external list rule */
RULE EXTERNAL LIST 'immut' EXEC ''
RULE 'listimmut' LIST 'immut' FOR FILESET ('worm') WHERE NOT
(exclude_list) and NOT (immutable)

/* to run this policy: mmapplypolicy fsname -P policy -f ./gpfs -I
defer */
/* result is written to ./gpfs.list.immut */
```

The tag ‘*%working-directory/* stands for the directory in the file system that might be used as the working directory for the policy engine (parameters `-s / -g` of the `mmapplypolicy` command)

## Sudo group example

The example below shows the sudo configuration for groups: gpfsadmin and secadmin. Users in group gpfsadmin are allowed to run most of the GPFS and LTFS commands. They are not allowed to run certain commands. These commands could be temporarily enabled by secadmin users. Users in group secadmin are allowed to change the sudo configuration.

This example of the group definitions has not been fully tested and may require further adjustments in the context of a particular implementation.

```
#### Allow members of gpfsadmin to run all mm-commands and ltfsee
commands
%gpfsadmin ALL=(ALL) PASSWD: LOG_INPUT: LOG_OUTPUT:
/usr/lpp/mmfs/bin/mm*, /opt/ibm/ltfsee/bin/ltfsee
*,/usr/local/bin/ltfs, NOPASSWD: LOG_INPUT: LOG_OUTPUT:
/usr/lpp/mmfs/bin/mmremote, /usr/bin/scp, /bin/echo,
/usr/lpp/mmfs/bin/mmsdrrestore

#### disallow certain mm-commands for members of the GPFS admin group
%gpfsadmin ALL=(ALL) LOG_INPUT: LOG_OUTPUT: !/usr/lpp/mmfs/bin/mmfsadm,
!/usr/lpp/mmfs/bin/mmaddcallback, !/usr/lpp/mmfs/bin/mmchcluster,
!/usr/lpp/mmfs/bin/mmchconfig, !/usr/lpp/mmfs/bin/mmauth

#### allow members of gpfsadmin the following commands temporarily
%gpfsadmin ALL=(ALL) PASSWD: LOG_INPUT: LOG_OUTPUT:
/root/software/itdt/ITDT/itdt, /usr/bin/umount /ltfs
```

```
#### Allow members of the security admin group to change the sudo
configuration
%secadmin ALL=(ALL) PASSWD: LOG_INPUT: LOG_OUTPUT: /usr/sbin/visudo

#### Add logging
#%admins         ALL=(ALL) NOPASSWD: LOG_INPUT: LOG_OUTPUT: ALL
Defaults iolog_dir=/var/log/sudo-io/%{user}
```

Note: The use of GPFS sudo wrappers does not guarantee that users in the gpfsadmin group cannot elevate their privileges. In particular the password-less use of /usr/bin/scp and /usr/bin/echo in the definition for the group gpfsadmin makes the solution vulnerable. Additional operational measures are required.

# References


[1] IBM Spectrum Scale Knowledge Center: Immutability and append-only: http://www-01.ibm.com/support/knowledgecenter/STXKQY_4.1.1/com.ibm.spectrum.scale.v4r11.adv.doc/bl1adv_integratedarchiveplatform.htm?lang=en

[2] Information about the standard SnapLock® method invented by NetApp Inc. http://www.netapp.com/uk/products/protection-software/snaplock.aspx

[3] End-to-End checksums with IBM Spectrum Scale Native RAID http://www-01.ibm.com/support/knowledgecenter/SSYSP8_3.5.0/com.ibm.spectrum.scale.raid.v4r11.adm.doc/bl1adv_introe2echecksum.htm?lang=en

[4] Configuring sudo wrappers in an IBM Spectrum Scale cluster https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1adm_sudowrapper.htm

[5] IBM Spectrum Scale Archiving Policies - GPFS Policy Guide for LTFS EE: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102518

[6] IBM Spectrum Scale Information Lifecycle Management Policies: http://www-01.ibm.com/support/knowledgecenter/STXKQY_4.2.0/com.ibm.spectrum.scale.v4r2.adv.doc/bl1adv_policyrules.htm?lang=en

[7] Article: Configuring IBM Spectrum Scale for reliability https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20%28GPFS%29/page/Configuring%20GPFS%20for%20Reliability

[8] IBM Spectrum Scale backup function using “mmbackup” https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.0/com.ibm.spectrum.scale.v4r2.adm.doc/bl1adm_backupusingmmbackup.htm

[9] Managing immutability with IBM Spectrum Scale via SMB:

https://www.ibm.com/developerworks/community/blogs/c8abdf40-97a5-47e6-93be-47abc3ed45b4/entry/Achieving_WORM_like_functionality_from_NFS_SMB_clients_for_data_on_Spectrum_Scale?lang=en

[10] Detailed KPMG assessment report for IBM Spectrum Scale V 4.2 immutable filesets:
http://www.kpmg.de/bescheinigungen/RequestReport.aspx?41742
KPMG software certificate for IBM Spectrum Scale immutable filesets:
https://www.kpmg.de/bescheinigungen/RequestReport.aspx?41743

[11] IBM Spectrum Scale audit logging for file system activity using Varonis
https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1adv_dpauditlogging.htm

[12] IBM Spectrum Scale encryption
https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1adv_encryption.htm

[13] IBM Spectrum Scale compression
http://www.ibm.com/support/knowledgecenter/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adm_compression.htm

[14] IBM Spectrum Scale security red paper
https://www.redbooks.ibm.com/redbooks.nsf/RedpieceAbstracts/redp5426.html

[15] Assessment report for IBM TS1100 WORM tapes
http://www.kpmg.de/bescheinigungen/RequestReport.aspx?AFC70A11EC864E189CE2F5BCF42753CB

[16] IBM Spectrum Scale Authentication and Authorization
https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1ins_authconcept.htm

[17] IBM Spectrum Scale command audit logging
https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1pdg_syslog.htm

[18] Logical Block Protection with Spectrum Archive
https://www.ibm.com/support/knowledgecenter/STZMZN/com.ibm.storage.hollywood.doc/ltfs_le_lbp.html

[19] IBM Spectrum Scale Firewall considerations
https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1adv_firewall.htm

[20] Spectrum Archive EE Redbook
http://www.redbooks.ibm.com/redpieces/abstracts/sg248333.html?Open

[21] IBM Spectrum Scale Monitoring overview
https://developer.ibm.com/storage/2017/11/16/spectrum-scale-monitoring-know/

[22] IBM Spectrum Scale REST API Overview

https://www.ibm.com/support/knowledgecenter/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1adm_restapi_functionaloverview.htm

[23] IBM Spectrum Scale version 5.0 native file audit logging
https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.0/com.ibm.spectrum.scale.v5r00.doc/bl1ins_adlgintro.htm#adlgintro

# Disclaimer

This document reflects the understanding of the author in regard to questions asked about archiving solutions with IBM hardware and software. This document is presented “As-Is” and IBM does not assume responsibility for the statements expressed herein. It reflects the opinions of the author. These opinions are based on several years of joint work with the IBM Systems group. If you have questions about the contents of this document, please direct them to the Author (nils_haustein@de.ibm.com).

The Techdocs information, tools and documentation ("Materials") are being provided to IBM Business Partners to assist them with customer installations. Such Materials are provided by IBM on an "as-is" basis. IBM makes no representations or warranties regarding these Materials and does not provide any guarantee or assurance that the use of such Materials will result in a successful customer installation. These Materials may only be used by authorized IBM Business Partners for installation of IBM products and otherwise in compliance with the IBM Business Partner Agreement.”

This document provides guidance for certain configuration and operational aspects. IBM does not guarantee that this guidance complies with laws or regulation. In order to obtain a compliance assessment an independent auditor has to be engaged by the client. IBM cannot be made liable for any findings or violations of laws and regulations.

The software nature of the solution may allow malicious hackers to exploit the system and elevate privileges of users. The use of GPFS sudo wrappers does not completely assure that there is no way to escape the sudo environment and elevate privileges of users. In addition it might be possible to use certain undocumented GPFS commands to exploit the system and elevate privileges of users. IBM cannot be made liable for any damage resulting of the use of exploits.

The guidance given herein does not imply warranty that the commands given or their intention satisfies the purpose. IBM cannot be made liable upon damage caused by any of the commands or guidance.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States or other countries or both: IBM, IBM Spectrum Scale and IBM Spectrum Protect.

Linux is a registered trademark of Linus Torwald

SnapLock® is a registered trade mark of NetApp Inc. in the United States and other countries.

Microsoft® Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.

Microsoft® Windows® PowerShell™ is a registered trademark of Microsoft Corporation in the United States and other countries.

Varonis provides software that protects data from insider threats and cyberattacks, and enables organizations to analyze, secure, manage, and migrate their volumes of unstructured data.

Other company, product, and service names may be trademarks or service marks of others.