

DELL EMC UNITY: FILE-LEVEL RETENTION (FLR)

A Detailed Review

Abstract

This white paper explains the concepts and benefits of File-Level Retention (FLR) for Dell EMC Unity™. The paper outlines the available commands and configurations available when using this feature and advanced features. This feature is available on Dell EMC Unity OE version 4.5 and later.

January, 2019

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license. Copyright © 2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA [01/19] [White Paper] [H17523]

Dell EMC believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	5
AUDIENCE	5
TERMINOLOGY	5
INTRODUCTION	6
OVERVIEW	6
FILE-LEVEL RETENTION (FLR) TYPES	6
REQUIREMENTS	7
FLR CONCEPTS	8
RETENTION DATES	8
FLR-C TO FLR-E COMPARISON	8
FILE STATES	9
RETENTION SETTINGS	11
AUTO-LOCK	12
AUTO-DELETE	12
TAMPER-PROOF CLOCK	12
FILE SYSTEM PROTECTION	13
DEFAULT “HARD” INFINITE RETENTION PERIOD	13
FLR-C’S DATA VERIFICATION	14
ACTIVITY LOG	14
MANAGEMENT	16
LICENSING	16
CREATING AN FLR ENABLED FILE SYSTEM	17
FILE SYSTEM DELETION	19
ENABLING FLR-C’S DATA INTEGRITY	19
HOW TO LOCK FILES	20
NFS ENVIRONMENT	20
WINDOWS ENVIRONMENT	20
HOW TO CREATE APPEND-ONLY FILES	23
INTEROPERABILITY	24
ANTIVIRUS SCANNING	24
DATA REDUCTION	24
FILE TIERING WITH CLOUD TIERING APPLIANCE (CTA)	24
NDMP BACKUP	24
SNAPSHOTS	24
<i>FLR-C</i>	24
<i>FLR-E</i>	24
REPLICATION	24
DELL EMC UNITY NATIVE FILE IMPORT	25

DESIGN CONSIDERATIONS..... 25
CONCLUSION 26
REFERENCES..... 27

EXECUTIVE SUMMARY

Being able to protect file data from modification or accidental deletion is a critical component in the operation and function of many organizations. File-Level Retention (FLR) is a feature that is used to protect file data from deletion or modification until a specified retention date. FLR enables you to create a permanent, unalterable set of files, and ensures the integrity of the data when using the FLR-C type. Locked, or protected, files are commonly referred to as WORM (Write-Once, Read-Many) files.

This white paper provides a comprehensive overview of File-Level Retention for Dell EMC Unity™, including the two FLR types:

- File-Level Retention Compliance (FLR-C)
- File-Level Retention Enterprise (FLR-E)

AUDIENCE

This white paper is intended for IT planners, storage architects, system administrators, partners, Dell EMC employees, and any others involved in evaluating, acquiring, managing, operating, or designing an FLR protected environment using Dell EMC Unity systems.

TERMINOLOGY

Append-only state – The state of a file when the data in it cannot be modified or deleted, but can have new data added at the end. Once you write to a file in the append-only state, you can transition it to the locked state.

Epoch date – An instant in time that is chosen as the origin of a particular era. In computer systems, time is expressed as the number of time units that have elapsed since a specified epoch date, also called the reference date. On UNIX systems, time is expressed in number of seconds since January 1, 1970.

Expired state – The state of a file when its retention date has passed. A file in the expired state can be reverted to the locked state or deleted from the FLR-enabled file system, but it cannot be altered. If the expired file is empty, you can transition it to the append-only state.

File-Level retention (FLR) – A feature that lets you store data on drives using NFS or SMB operations to create a permanent, unalterable set of files.

FLR clock – A non-modifiable, per-file system clock, which is used to track the retention date. It is initialized when an FLR-enabled file system is created. There is no way to advance the FLR clock, but it is possible to fall behind after a snapshot restore.

Locked state – The state of a file in an FLR-enabled file system when the file's read/write permission is changed to read-only and a retention date is set. Files committed to the locked state cannot be altered or deleted until their retention date has passed. "Locked" and "protected" are used synonymously in this paper.

NAS Server – A Dell EMC Unity storage server that uses the SMB, NFS, or FTP/SFTP protocols to catalog, organize, and transfer files within designated file system shares. A NAS Server, the basis for multitenancy, must be created before you can create file-level storage resources such as file systems or VMware file datastores.

Network Attached Storage (NAS) – File-based storage for a wide range of clients and applications that access storage over IP connectivity.

Network File System (NFS) – An access protocol that allows data access from Linux/UNIX hosts on a network.

Not locked state – The initial state of a file when it is created. A file that is not locked is treated in the same manner as a file in a file system that is without FLR. Unless the file is locked, it can be renamed, modified, or deleted.

Retention date – The date until which a locked file in an FLR-enabled file system is protected. Users and applications manage a file's retention date by using NFS or SMB to set the file's last access time to a date and time. The retention timestamp is compared with the file system's FLR clock to determine whether a file's retention date has passed.

Server Message Block (SMB) – An access protocol that allows data access from Windows/Linux hosts on a network. Also known as Common Internet File System (CIFS).

INTRODUCTION

File-Level Retention (FLR) provides a software infrastructure in the Dell EMC Unity system for files to be locked, that is, protected from deletion or modification by users or storage administrators. This functionality is also known as Write Once, Read Many (WORM). FLR is available on the physical Dell EMC Unity family as well as Dell EMC UnityVSA systems. This feature is only available for file systems and is not available for VMware NFS datastores. FLR provides a cost-effective solution for NAS files throughout their life cycle. The File-Level Retention (FLR) process can be compliant with the regulatory requirements of the United States Securities and Exchange Commission (SEC) Rule 17a-4 (f) for digital storage.

FLR is enabled per file system at creation time so that you have the flexibility to use regular file systems and FLR-enabled file systems within the same NAS Server. Keep in mind that FLR cannot be modified (enabled or disabled) after creation of the file system. Once FLR is enabled, it cannot be disabled. For which reason, it is critical to be certain that the use of FLR is required. The administrator can distinguish FLR-enabled file systems by the level of protection required: self-regulation or compliance. Individual files within FLR-enabled file systems can be locked with their own unique retention dates. Only when the retention date of a locked file has expired can that file be deleted.

With FLR, files that are created on a Unity file system do not need to be transferred to a specialized storage product for file-level retention. They can stay on cost-effective NAS storage, which reduces the need to invest in a more expensive storage product for data protection with compliance. Files that are stored on a Unity file system can take advantage of storage efficiency features such as thin provisioning and Data Reduction to further reduce the storage footprint.

Typical use cases for FLR include:

- Preventing deletion
 - Human error
- Data Integrity
 - Self-regulated business practices
 - Compliance (such as Federal)

OVERVIEW

FILE-LEVEL RETENTION (FLR) TYPES

FLR comes with two options that differ in the level of strictness in enforcing retention policies. Each file system can be enabled with one of the two options: FLR Enterprise (FLR-E) or FLR Compliance (FLR-C).

This list describes the difference between the two types of FLR:

FLR Enterprise (FLR-E)

- Prevents file modification and deletion by users through NAS protocols such as SMB, NFS, and FTP
- Does not prevent file system deletion by storage administrators, even if the file system has locked files

FLR Compliance (FLR-C)

- Prevents file modification and deletion by users through NAS protocols such as SMB, NFS, and FTP
- Prevents file system deletion by storage administrators if the file system has locked files
- Includes a data integrity check, which is disabled by default. Refer to **Enabling FLR-C's Data Integrity** for information about enabling the data integrity check
- FLR-C includes some snapshot restrictions
 - FLR-C only supports read-only snapshots
 - FLR-C does not support snapshot restores
- FLR-C has a hard infinite retention period, meaning that a file locked with infinite retention can never be reduced
- FLR-C meets the requirements of SEC rule 17a-4(f)
 - Intended for companies that need to comply with federal regulations

In both FLR-C and FLR-E file systems, you cannot modify or delete files that are in the locked state. Additionally, the path to a file in the locked state is protected from modification, which means that you cannot delete or rename a directory on a FLR-enabled file system if it contains protected files.

REQUIREMENTS

Some industries look to implement file-level retention policies as a form of self-regulated good business practice. To provide a robust solution that can uphold the file-level retention policies established companies, it is important that the infrastructure can protect files from accidental deletion and modification as well as from malicious attempts by individuals who have access to the NAS storage environment. Other industries look to implement file-level retention policies in response to government regulations such as those for medical, telecommunications, financial, and pharmaceutical industries.

To meet requirements for robustness and government regulations, the FLR infrastructure has multiple components to ensure protection of files and to audit events that take place on an FLR-enabled file system. Furthermore, FLR is designed to be compliant with U.S. SEC Rule 17a-4(f), which regulates the storage, retrieval, and management of electronic records for certain exchange members, brokers, and dealers. With many industries adopting strict regulations that align closely with SEC Rule 17a-4(f), it has become essential to leverage a file-level retention solution that can meet the SEC requirements.

As mentioned, FLR for Dell EMC Unity offers two options to enable a file system for file-retention capabilities: Enterprise and Compliance. FLR-E is an Enterprise-enabled file system that companies can use to regulate themselves as a good business practice. FLR-C is a Compliance-enabled file system that companies use to meet the regulations set forth by the SEC.

The SEC regulation requirements are as follow:

1. The first requirement of SEC Rule 17a-4(f), found in the SEC ruling and requirements, is to preserve the records exclusively in a non-rewritable, nonerasable format. To address this requirement, FLR-C is designed to prevent any modification or deletion of locked files by either users or administrators until a specified retention date has passed.
2. The second requirement of SEC rule 17a-4(f) is to automatically verify the quality and accuracy of the storage media recording process. To address this requirement, an FLR-C file system provides block-level checksum (bit-level verification codes) and bit-level verification (also known as disk scrubbing).
 - a. File-level checksums are calculated when the data is recorded and are maintained by Unity. When the data is read back from the disk, the system verifies the checksums to ensure that the data has not been altered since it was written.
 - b. Periodically, a bit-level verification of the physical storage media and block-level checksums is performed to ensure that there are no hardware failures at the media level.
3. The third requirement of SEC rule 17a-4(f) is to serialize the original and, if applicable, duplicate units of storage media and to timestamp the information on the storage media for the required retention period. To address this requirement, all files that are created in an FLR-C file system have a unique name, the full directory path and file name that identifies them. The “last modified” timestamp records the time at which the files were last written to before being committed. For those files that are committed, the “last accessed” timestamp records the date until which the file is protected.
4. The fourth requirement of SEC rule 17a-4(f) is to have the capacity to readily download indices and records (files) preserved on the electronic storage media to any medium acceptable under paragraph (f), as required by the commission or the self-regulatory organizations to which the exchange member, broker, or dealer belongs. To address this requirement, the record names and timestamps (metadata) and the content of the records (data) stored in an FLR-C file system can be:
 - a. Copied by using standard NAS protocols.
 - b. Replicated to an alternate location by using the native replication technologies.
 - c. Backed up through Network Data Management Protocol (NDMP).
5. The rule also requires that the organization provide “an audit system for accountability regarding the input of records into the storage system.” An FLR activity log is maintained in each FLR-enabled file system to support this requirement. Refer to the **ACTIVITY LOG** section for more information in the FLR log.

FLR CONCEPTS

The retention for each file is controlled by an attribute. The attribute identifies the file to the system as an FLR file, which has metadata needed for the NAS Server to process the file. The metadata includes the retention date and the state (for example, not locked, locked, append-only, or expired). Although the FLR attribute protects the file, the storage environment plays an important part in determining the level of protection. If an administrator can manipulate the environment by changing the system clock, the required protection solution is defeated.

RETENTION DATES

The retention date is the user-specified date and time until which a file is protected. Locked files use the file's access time attribute to store the retention date. To lock and set a retention date to a file, change the access time attribute to the intended date and set the file to read-only. Refer to the **How to Lock Files** section for more information on how to lock files.

The epoch time is an incrementing signed integer, which will eventually overflow. Maximum values for retention dates are depending on the host's operating system type. For 32-bit systems, the maximum epoch time is January 19, 2038 at 03:14:08, and UTC so; attempting to set a retention year greater than 2038 returns an error. For 64-bit systems, retention periods can be set up to the year 2106, with the maximum date being February 7, 2106 at 06:28:13 UTC.

To set retention years between 2039 – 2084 for 32-bit operating systems, a “base year” formula was added to the system. To trigger this formula, set the retention date to a year in the past (between 1971 and 2017). When a “base year” is used for the retention date, the system uses the following formula to calculate the actual retention year desired.

Formula: $2038 - 1970 + \text{base_year} = \text{Actual Retention Year}$

For example, $2038 - 1970 + 1971 = 2039$

For example, $2038 - 1970 + 2016 = 2084$

For example:

```
[root@VM test]# touch -at 201601010000 file
[root@VM test]# chmod -w file
[root@VM test]# ls -lui --time-style=long-iso
total 32
9445 -r--r--r--. 1 root root 5 2084-01-01 00:00 file
```

FLR-C TO FLR-E COMPARISON

Table 1 compares the features that are available in the FLR-C vs FLR-E file systems.

Table 1. FLR-C and FLR-E features

Feature	FLR-C	FLR-E
Default/minimum/maximum retention periods	✓	✓
Auto-lock and auto-delete	✓	✓
Tamper-proof clock	✓	✓
File system protection	Cannot delete FS with protected files	Can delete FS with protected files
Data verification	✓	X
Default “hard” infinite retention	✓	X

Activity log	✓	✓
Append-only files	✓	✓
Only supports read-only snapshots	Restriction for FLR-C Only	N/A
Does not support snapshot restores	Restriction for FLR-C Only	N/A

FILE STATES

Files in a FLR-enabled file system can have one of the following states:

- **Not Locked**
 - The initial state of a new file
 - Treated in the same manner as a file in a non-FLR file system (can be modified, deleted, and so on)
- **Locked**
 - A locked file has a set retention period that prevents users from modifying the file data, deleting, moving, or renaming the file
 - A locked file remains in this state until its retention period expires. An administrator can perform two actions on a locked file:
 - Modify the file retention date to extend the existing retention period.
 - If the locked file is initially empty, move the file to the append-only state.
 - Files can be manually locked by a user or automatically locked by the system
 - A locked file can have its retention period extended, but not shortened
- **Append Only**
 - You cannot delete, move, or rename the file, and you cannot modify the existing data in an append-only file, but new data can be added to the end of the file
 - Since existing data cannot be changed and new data can be added, append-only files are useful for log files
 - A state that you can set only on an empty file in the locked state
 - An append-only file does not have a retention period, but it cannot be deleted unless it is empty
 - After modifying an append-only file, it can be converted back to a traditional locked file or can remain in the append-only state forever
 - Transitioning to the locked state uses the retention period set by the user or the file system's default retention period
 - After data is written to an append-only file and the file is converted to the locked state, you cannot change the file back to the append-only state
 - Some applications interpret appending a file as extending a new file to the desired size and then writing the new data afterwards
 - This is seen as creating empty space on a file, then modifying the empty space, which is not allowed
- **Expired**
 - When the retention period ends, the file transitions from the locked state to the expired state
 - You cannot modify, move, or rename a file in the expired state, but you can delete the file
 - An expired file can have its retention period extended, to transition the file back to the locked state
 - An empty, expired file can also transition to the append-only state

The following section demonstrated the life cycle of a file in an FLR-enabled file system.

1. Once a file is locked, either manually or automatically, it goes into a **Locked** state,



Figure 1 – File is Locked

- a. If a file is locked and unlocked while is empty, it goes to an **Append-only** state

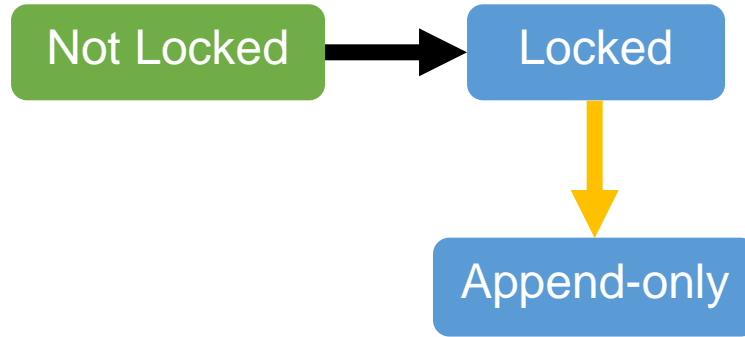


Figure 2 – Locked to Append-only

- b. If the retention period ends, the file can:
- Go into an **Expired** state (a)
 - Which then can be **Locked** again (b)

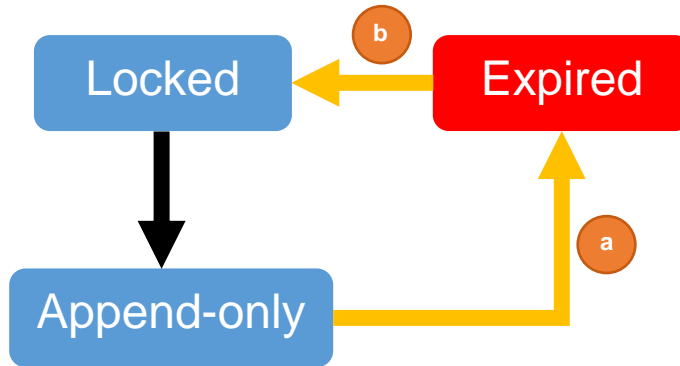


Figure 3 – Append-only file to Expired, then Locked

2. An expired empty file can be unlocked, which becomes an **Append-only** file

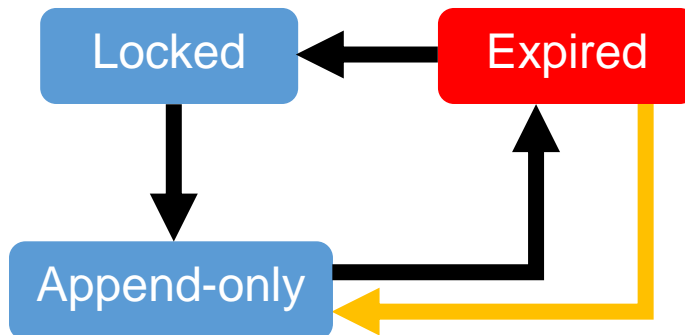


Figure 4 – Expired empty file is unlocked become Append-only

- 3. An **Append-only** file can be locked again

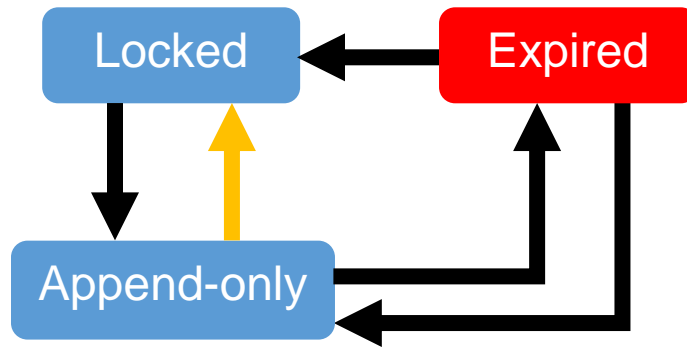


Figure 5 – Append-only file is Locked again

With all the possibilities ending in a never ending cycle between the different states, as follows:

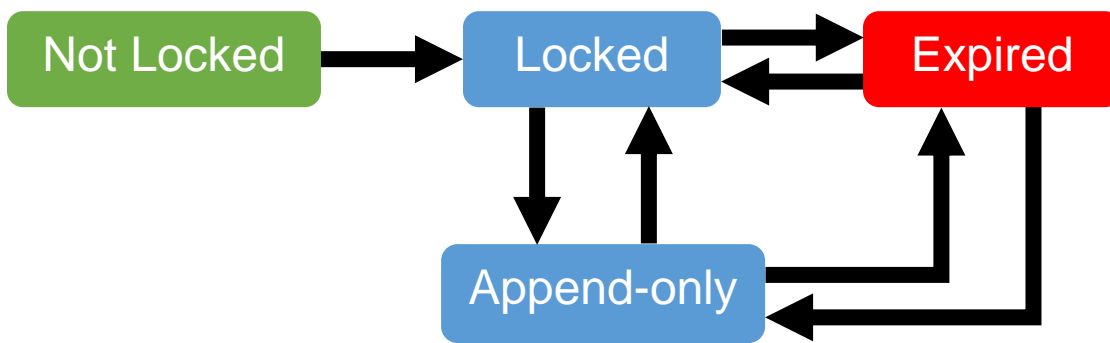


Figure 6 – File’s cycle through states

RETENTION SETTINGS

When enabling FLR at file-system creation, as stated before, FLR can only be enabled at creation time and cannot be modified afterwards. Once FLR is enabled, you have the option to set the default, minimum, and maximum retention periods. You can change these settings after file system creation.

- **Minimum Retention Period:** Specifies the shortest period of time that files can be locked for. The possible units are days, months, and years. Any attempts to lock files with a lower retention period than the minimum uses this setting instead. The minimum retention period must be less than or equal to the maximum retention period.

Table 2. Minimum Retention Period limits

Default Value	Minimum Value	Maximum Value
1 Day	0 Days	87 Years or Unlimited

- **Default Retention Period:** Specifies the default retention period, which is used if a file is locked without setting an explicit retention date. The default value is Unlimited for FLR-E file systems and one year for FLR-C file systems. **Note:** For FLR-C, if a file is locked with unlimited retention, it means that the file and file system can never be deleted. The default retention period must be greater than or equal to the minimum retention period and must be less than or equal to the maximum retention period.

Table 3. Default Retention Period limits

Default Value	Minimum Value	Maximum Value
Unlimited (FLR-E) 1 Year (FLR-C)	0 Days	87 Years or Unlimited

- **Maximum Retention Period:** Specifies the longest period of time for that files can be locked for. Any attempts to lock files with a higher retention period than the maximum uses this setting instead. The maximum retention period must be greater than or equal to the minimum retention period.

Table 4. Maximum Retention Period limits

Default Value	Minimum Value	Maximum Value
Unlimited	1 Day	87 Years or Unlimited

The minimum and maximum retention periods allow the administrator to enforce retention dates that fall within a specified range. For example, you can set the minimum and maximum retention periods to 30 days and 1 year, respectively. If a user attempts to set a retention period of 20 days on a file, FLR automatically locks it with the minimum retention period of 30 days. If a user attempts to set a retention period of 5 years on a file, FLR automatically locks it with the maximum retention period of 1 year. If a user attempts to lock a file for 90 days, it is allowed. Note that modifying the minimum, maximum, or default retention periods do not apply to already locked files.

The following two attributes are configured at the file system level once the file system is created as FLR-enabled.

AUTO-LOCK

Auto-lock can be enabled after creating a FLR file system. Auto-lock automatically locks files in the file system with the default retention period if they have not been modified for a user-specified period of time. A parameter called the Policy Interval is used to configure the user-specified period of time.

Once auto-lock is enabled, this will cause periodically scans of the file system for files that meet the criteria set for auto-lock. The scan interval is a factor of the policy interval. It may take up ½ of the time past the policy interval for the auto-lock to be triggered.

AUTO-DELETE

Auto-delete is another feature that can be enabled after creating the FLR file system. When this feature is enabled, FLR automatically deletes files with expired retention periods. Note: The auto-delete happens at 7-day intervals. The timer starts when auto-delete is enabled.

For more granular or additional options for auto-lock and auto-delete, the FLR Toolkit's Monitor Service can be used.

TAMPER-PROOF CLOCK

The retention date is compared to the current FLR Clock for that file system to determine when files are expired. A software clock mechanism in FLR addresses the issue of malicious administrators attempting to delete protected content before its expiration date by tampering with the system clock. FLR includes a tamper-proof and nonmodifiable software clock set once for each file system. The value of the FLR clock is initialized by synchronizing it with the current Storage Processor (SP) time when the FLR file system is first created.

The FLR clock is periodically updated on the file system. Because the FLR clock might not always be synchronized with the system time, the FLR clock can adjust itself to changes in the system clock. If the system time is turned back, and the FLR clock is ahead of the system clock, the FLR clock will be synced back to the system clock. If the system time is ahead of the FLR clock, the FLR clock gradually adjusts to that change. The FLR clock can advance 138 seconds per hour if it is behind the current value of the system clock. If necessary, the FLR clock can catch up at most two weeks per year. This implementation prevents administrators from deleting protected files early because the slow rate of change makes attacks of this nature impractical.

FLR enables authorized administrators to restore from previous snapshots on FLR-E file systems. The FLR clock adjusts accordingly to the FLR clock from the snapshot. Snapshot restores are prohibited on FLR-C file systems. Refer to Figure 7 for an example of the FLR clock.

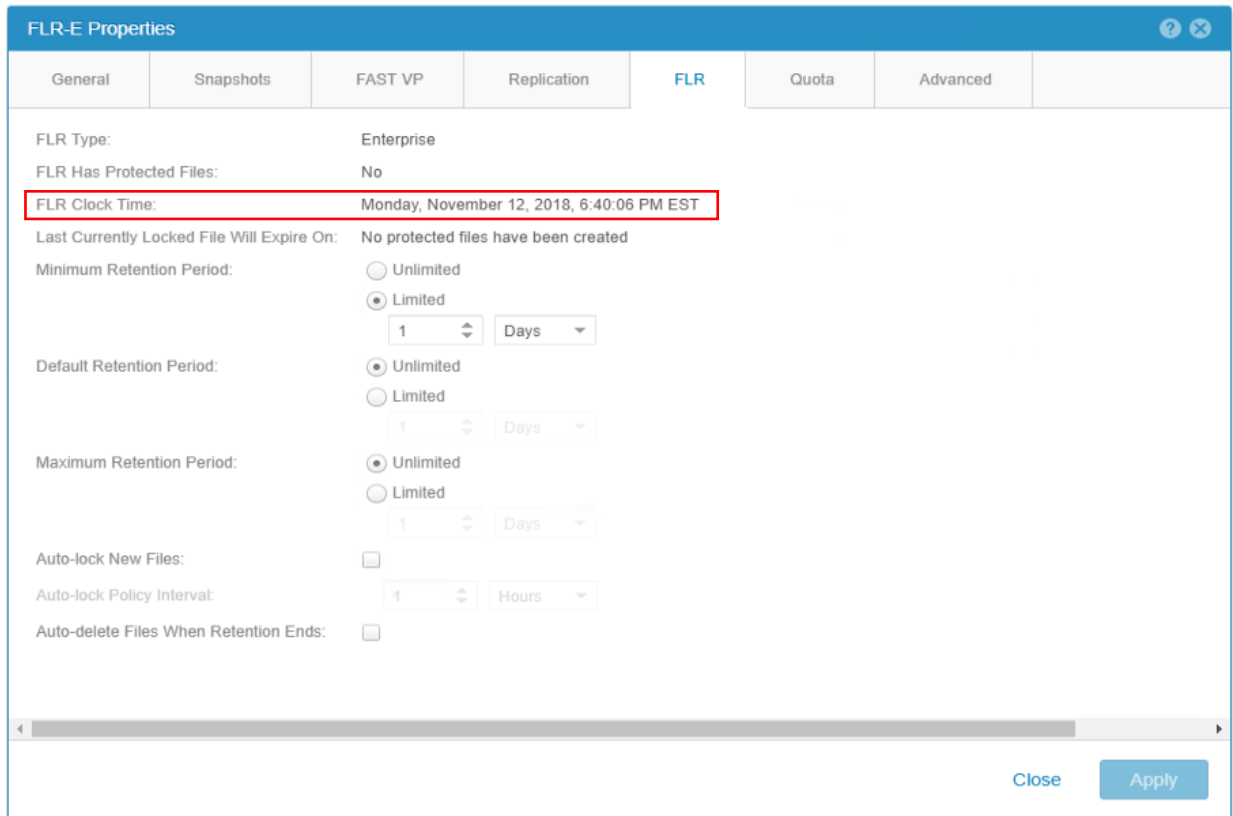


Figure 7 – FLR Clock Time

FILE SYSTEM PROTECTION

To meet the SEC Rule 17a-4(f) requirement, FLR-C prohibits destructive operations on an FLR-C file system that contains protected files. With FLR-C, a file locked with unlimited retention means the file and the file system can never be deleted. If there are no protected files, you are allowed to delete the file system.

Although this feature prevents destructive actions against protected content at the file-system level for FLR-C file systems, it does not protect committed content at the disk level. However, the intent of SEC Rule 17a-4(f) requirement is to prevent modification of individual files without trace or record. FLR-C meets this requirement because an administrator cannot target individual files for deletion.

FLR-E treats this scenario differently. Unity alerts the administrator to the presence of protected content on the FLR-E file system and requires a confirmation from the administrator before deletion. With FLR-E, a file that is locked with unlimited retention can be updated with a specific retention date at a later time.

DEFAULT “HARD” INFINITE RETENTION PERIOD

FLR-E retains the “soft” infinite retention period in this scenario. This means that the infinite retention period can be reduced by setting a date afterwards.

FLR-C has a hard infinite retention date that can never be decreased. This means both the file and file system can never be deleted. Figure 8 shows the confirmation message when selecting an Unlimited Default Retention Period for an FLR-C enabled file system.

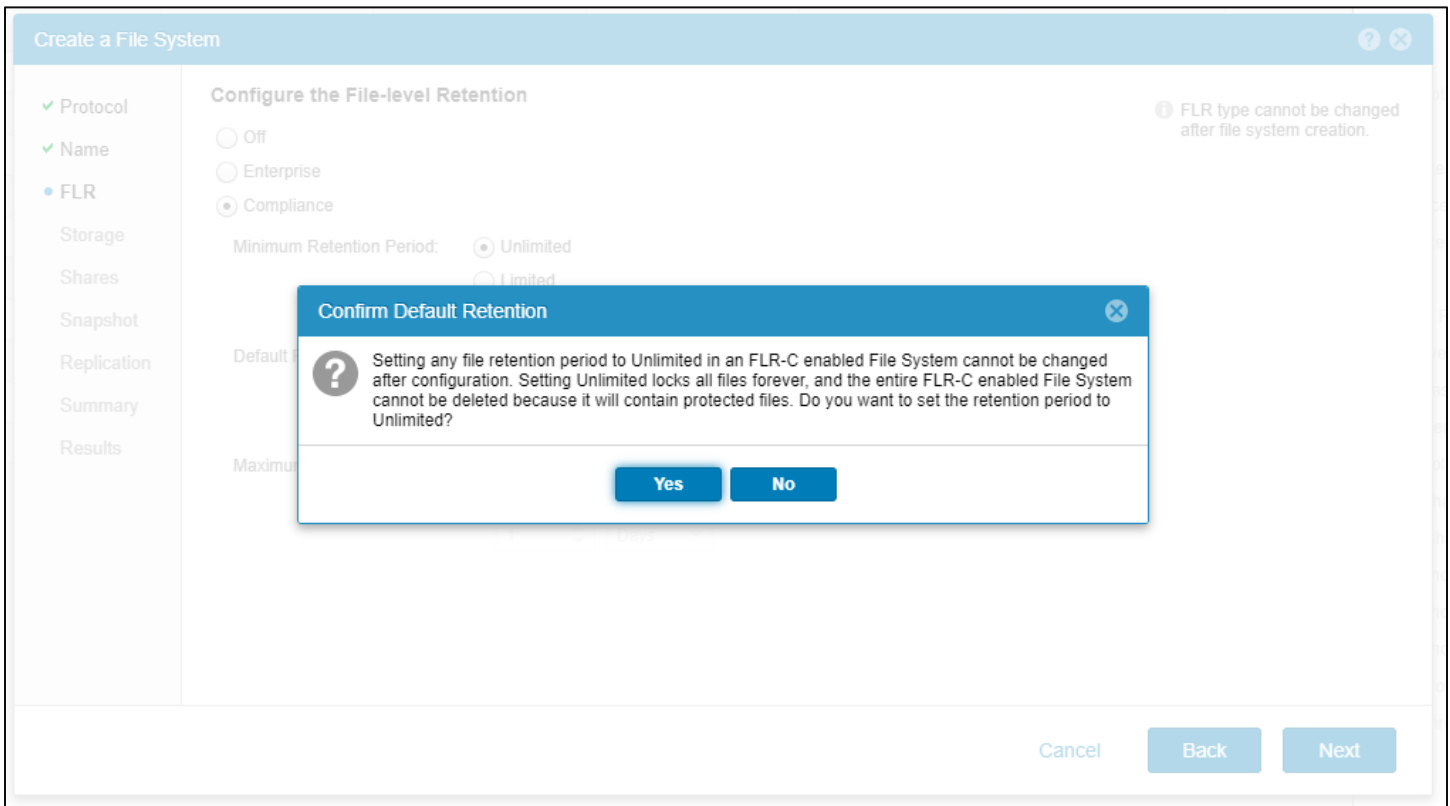


Figure 8 – Unlimited Default Retention

FLR-C'S DATA VERIFICATION

FLR-C (but not FLR-E) includes an enhancement for write verification. SEC Rule 17a- 4(f) requires that the storage system ensures the integrity of the stored data by reading back the data that was written to the file system.

If the data that is read back from the storage system does not match the data in memory, the system attempts to write and read back two more times. If a mismatch still occurs, the system reports an error and generates an event on the system to inform the administrator.

Users may experience performance degradation because of the write verification feature. When this feature is enabled, every write results in an additional read of the data that was just written. As a result, a write operation to an FLR-C file system will have an impact on file system performance.

A NAS Server parameter controls the data verification feature. By default, data verification is disabled. Changes to the parameter take effect immediately without the need for a system reboot. Refer to **Enabling FLR-C's Data Integrity** for more details.

ACTIVITY LOG

Because an FLR file system is used in environments that are subject to strict regulations, both FLR-C and FLR-E record events that pertain to successful and unsuccessful attempts to change protected data on the file system. The activity log records the user, time of event, and the type of action taken against protected files.

The FLR log has a fixed naming convention, flrLog[timestamp], and is stored in FLR_Logs folder under the root directory of each FLR file system. The FLR log has a maximum size of 10 MB. Logs that meet the limit are converted to locked files, and subsequent events are written to another file with the same naming convention. A converted log file is set with the maximum retention period of locked files in the file system.

The administrator is responsible for deleting old and expired log files. If there is insufficient space in the file system to update the activity log, operations cannot proceed, and warnings are logged in the system logs, and posted as an alert to the administrator.

The activity log captures the following events:

- Creation append-only file
- File set to locked state

- Retention period extended on a locked or expired file
- Attempt to make a locked or expired file writable
- Deletion or attempt deletion of a protected (locked, expired, append-only) file
- Changes to the FLR settings

The following information accompanies a recorded event:

- Time of event (software clock maintained by the file system)
- Action (events described above)
- Inode number
- Generation number (epoch timestamp) for the inode
- User identifier (UID) and group identifier (GID) of the user who performed or attempted the action
- File permissions
- File size
- Event-specific information:
 - The retention date and time if a locked file is committed
 - Success or failure if a locked file is deleted or if there is an attempt to delete it

Here is an example of the FLR log after following these steps:

1. Initial file system's settings
2. Locking a file
3. Attempting to delete/rename/move/edit/chmod a locked file
4. Creating an append-only file
5. Changing the FLR file system's settings
6. Locking an append-only file
7. Extending retention date on a locked file

```
[root@VM mnt]# ls -l
total 8
drwxr-xr-x. 2 root bin 152 Sep 6 10:26 FLR_Logs
drwxr-xr-x. 2 root root 8192 Sep 6 10:26 lost+found
[root@VM mnt]# cd FLR_Logs/
[root@VM FLR_Logs]# ls -l
total 8
-rw----- . 1 root bin 137 Sep 6 10:26 flrLog20181130154921
[root@VM FLR_Logs]# cat flrLog20181130154921
Fri Nov 30 15:49:21 2018 : Activity log file created
Fri Nov 30 15:49:21 2018 : Initial fs rp range: max = infinite, default = infinite, min =
0D
Fri Nov 30 15:54:45 2018 : Worm commit clean file : Inode No = 9442 : Generation No =
1543592978 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize = 8 : RP = Fri Nov 30
18:54:39 2018 : Passed
Fri Nov 30 15:54:59 2018 : Worm commit clean file : Inode No = 9443 : Generation No =
1543592979 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize = 8 : RP = Fri Nov 30
17:54:59 2018 : Passed
```

```
Fri Nov 30 15:55:35 2018 : Delete worm committed file : Inode No = 9443 : Generation No =
1543592979 : Uid = 10000 : Gid = 100000000 : FileMode = 444 : FileSize = 8 : RP = Fri Nov
30 17:54:59 2018 : Failed

Fri Nov 30 16:19:18 2018 : Worm commit clean file : Inode No = 9444 : Generation No =
1543592980 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize = 8 : RP = Sat Dec 1
06:59:00 2018 : Passed

Fri Nov 30 16:20:47 2018 : Worm commit clean file : Inode No = 9445 : Generation No =
1543592981 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize = 8 : RP = Fri Nov 30
18:00:00 2018 : Passed

Fri Nov 30 16:21:30 2018 : Create a worm append-only file : Inode No = 9446 : Generation
No = 1543592982 : Uid = 10000 : Gid = 100000000 : FileMode = 644 : FileSize = 0 : RP =
Infinite : Passed

Fri Nov 30 16:22:59 2018 : Change fs default retention period: oldVal = Infinite : newVal
= 2 D : Passed

Fri Nov 30 16:23:07 2018 : Set auto delete feature: oldVal = disable : newVal = enable :
Passed

Fri Nov 30 16:23:07 2018 : Set auto lock feature: oldVal = disable : newVal = enable :
Passed

Fri Nov 30 16:23:07 2018 : Set auto lock policy Interval: oldVal = 3600 : newVal = 60 :
Passed

Fri Nov 30 16:23:08 2018 : Worm commit worm append-only file : Inode No = 9446 :
Generation No = 1543592982 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize = 37 :
RP = Sun Dec 2 16:24:05 2018 : Passed

Fri Nov 30 16:24:06 2018 : Worm commit clean file : Inode No = 9447 : Generation No =
1543592983 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize = 9 : RP = Sun Dec 2
16:25:05 2018 : Passed

Fri Nov 30 16:26:52 2018 : Make writeable worm committed file : Inode No = 9442 :
Generation No = 1543592978 : Uid = 10000 : Gid = 100000000 : FileMode = 644 : FileSize =
8 : RP = Fri Nov 30 18:54:39 2018 : Failed

Fri Nov 30 16:26:58 2018 : Delete worm committed file : Inode No = 9442 : Generation No =
1543592978 : Uid = 10000 : Gid = 100000000 : FileMode = 444 : FileSize = 8 : RP = Fri Nov
30 18:54:39 2018 : Failed

Fri Nov 30 16:27:02 2018 : Delete worm committed file : Inode No = 9442 : Generation No =
1543592978 : Uid = 10000 : Gid = 100000000 : FileMode = 444 : FileSize = 8 : RP = Fri Nov
30 18:54:39 2018 : Failed

Fri Nov 30 16:31:25 2018 : Extend Retention Period on worm committed file : Inode No =
9442 : Generation No = 1543592978 : Uid = 10000 : Gid = 10000 : FileMode = 444 : FileSize
= 8 : Old RP = Fri Nov 30 18:54:39 2018 : New RP = Fri Nov 30 23:54:39 2018 : Passed
```

MANAGEMENT

LICENSING

Dell EMC Unity File-Level Retention (FLR) is licensed with all Dell EMC Unity systems, including physical and Dell EMC UnityVSA systems, at no additional cost. To use FLR, the system must be running Dell EMC Unity OE version 4.5 or later. After upgrading the system to Dell EMC Unity OE version 4.5 or later, the File-Level Retention (FLR) feature will be automatically enabled, with no additional steps needed.

FLR is supported in newly created File Systems. FLR is not supported in VMware NFS Datastores.

To verify which version of Dell EMC Unity OE your system is running, select the **View System Status** icon on the top blue menu bar of Unisphere. Alternatively, you can view the license status for File-Level Retention by clicking the **Update System Settings** icon, denoted by a gear icon on the top blue menu bar, and finding **File Level Retention Management** in the License Management list. An entry of File Level Retention Management and a green checkmark besides it confirms that the feature is licensed on the system.

The following sections outline how to create and manage FLR enabled file systems. Unisphere examples for each of these areas are shown. For more information about using the Unisphere CLI, refer to the *Unisphere Command Line Interface User Guide* on Dell EMC Online Support. For information about managing Dell EMC Unity File-Level Retention from REST API, consult the REST API documentation which can be accessed directly from any Dell EMC Unity system:

- **REST API Programmer's Guide** – https://<Management_IP>/apidocs/programmers-guide/index.html
- **REST API Reference Guide** – https://<Management_IP>/apidocs/index.html
 - Where <Management_IP> is the management IP of your system.

CREATING AN FLR ENABLED FILE SYSTEM

The FLR setting must be enabled during file system creation. The user must choose either the Enterprise (FLR-E) or Compliance (FLR-C) type to enable the FLR functionality. The FLR setting cannot be modified after creation. You cannot enable FLR on an existing file system. If FLR is enabled, you can also customize the retention period options, either at creation time or modified at a later time.

To create an FLR-enabled file system, if a NAS Server has been created in the system already, go to the **File > File Systems** page and click **Add**. Figure 9 shows the new **FLR** step as part of creating a file system.

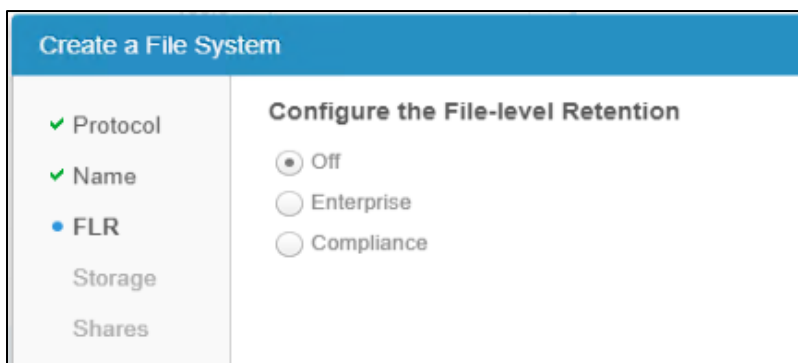


Figure 9 – FLR step in the **Create a File System** wizard

When selecting either **Enterprise** or **Compliance** in the **FLR** step, the UI shows a confirmation dialog. This is to confirm that the user wants to enable the FLR setting since the feature cannot be disabled once the file system is created, as shown in Figure 10.

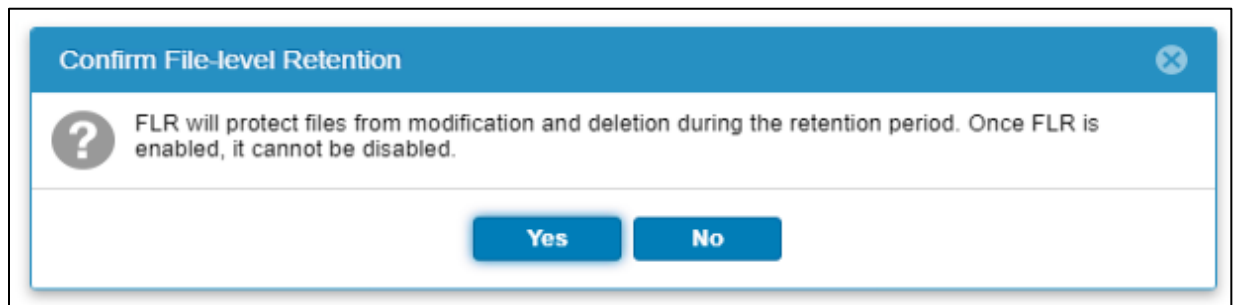


Figure 10 – Confirm dialog when enabling FLR in the **Create a File System** wizard

After confirming the dialog, the FLR attributes that can be configured on the file system are displayed, as shown in Figure 11, which includes the Minimum Retention Period, Default Retention Period, and Maximum Retention Period.

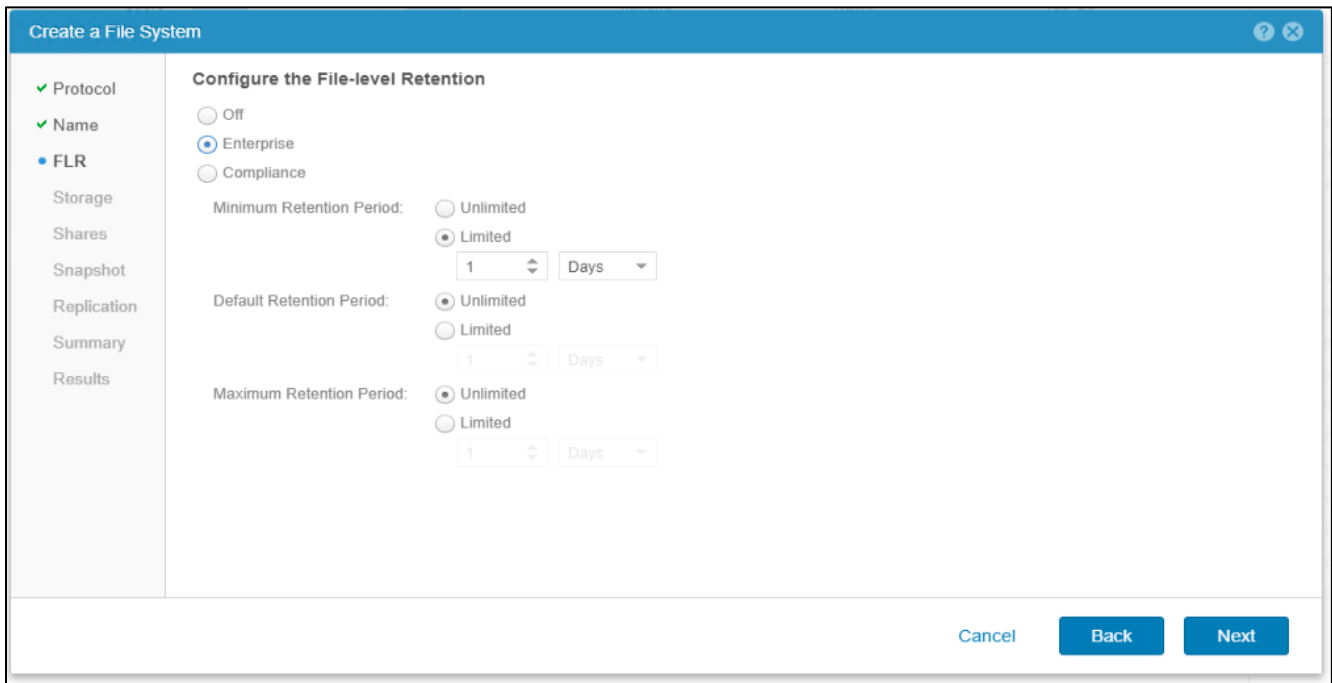


Figure 11 – FLR retention period options in the Create a File System wizard

After the file system is created, the FLR attributes and additional FLR features, such as Auto-Lock and Auto-Delete, can be modified from the file system’s properties, as shown in Figure 12.

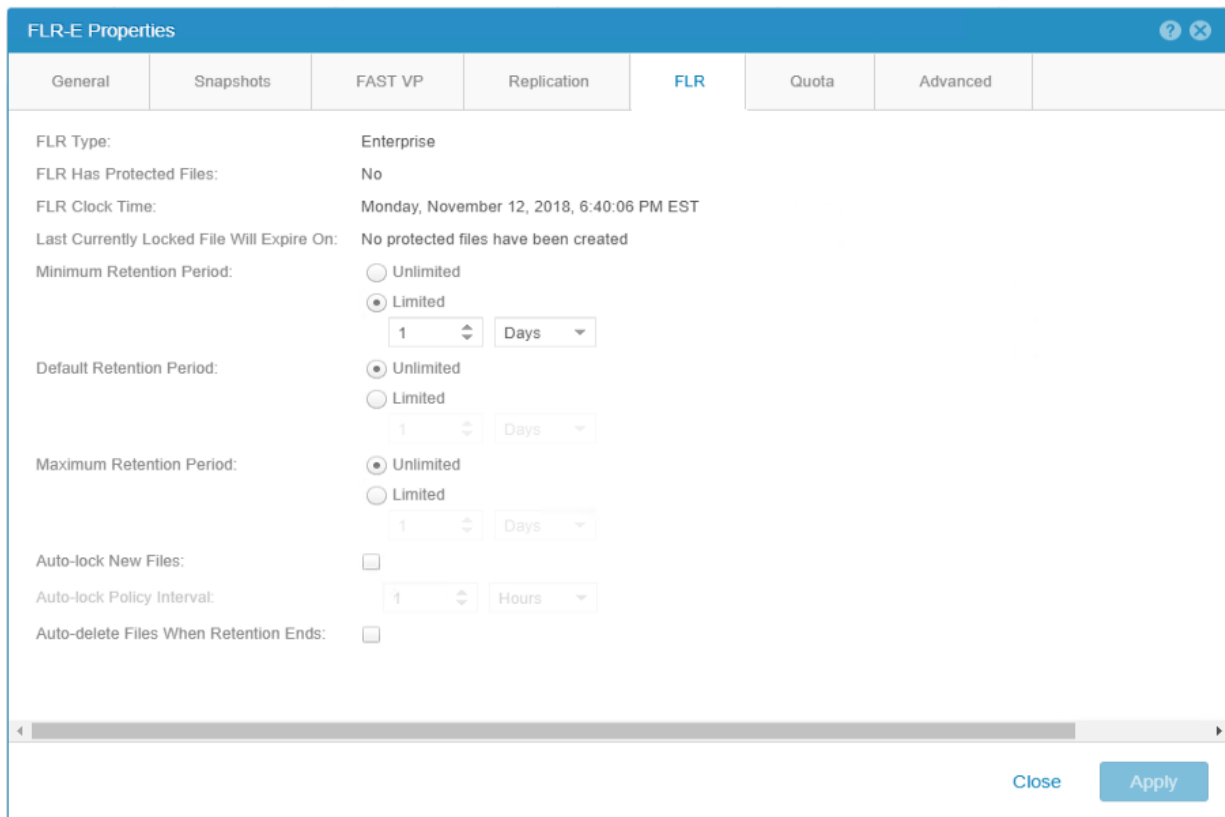


Figure 12 – FLR attributes and features from the file system’s properties

Under the **FLR** tab, the following file system’s attributes are shown:

- **FLR Type:** States the type of FLR that is enabled in the file system
- **FLR Has Protected Files:** States if the file system has FLR protected files

- **FLR Clock Time:** The nonmodifiable 24-hour software clock maintained by the file system
- **Last Currently Locked File Will Expire On:** Equal to the time when the last locked file on the file system will expire.

There are also optional FLR Advanced features, which are disabled by default:

- **Auto-lock New Files:** If enabled, the system automatically locks files if they are not modified for a user-specified period of time. Automatically locked files use the default retention period. Files in append-only mode are also subject to automatic locking.
- **Auto-lock Policy Interval:** Specifies how long to wait after files are modified before they are automatically locked. This setting is only available if Automatic File Locking is enabled. The scan interval is a factor of the policy interval. It may take up $\frac{1}{2}$ of the time past the policy interval for the auto-lock to be triggered.

Default Value	Minimum Value	Maximum Value
1 Hour	1 Minute	366 Days

- **Auto-delete Files When Retention Ends:** Automatically deletes locked files after their retention date has expired. The auto-delete happens at 7-day intervals. The timer starts when auto-delete is enabled.

FILE SYSTEM DELETION

When an administrator attempts to delete a file system, Unisphere either prevents the administrator from deleting the file system operation or provides a confirmation dialog, depending on the FLR setting.

- **FLR-C** – The delete button is disabled and a tooltip message is given that states that the file system cannot be deleted since it is an FLR-C enabled file system and still contains protected files.
- **FLR-E** – The confirmation screen displays a warning if the file system has protected files or is in the process of scanning the file system. However, the administrator can ignore the warning and choose to delete the file system.

ENABLING FLR-C'S DATA INTEGRITY

The FLR-C write verification functionality is controlled by the NAS Server parameter `FLRCompliance.writeverify`. By default, it is set to 0, which is disabled.

To view the value for the write verification parameter, run the following command:

```
svc_nas <NAS_Server> -param -f FLRCompliance -i writeverify -v
```

To enable the write verify feature, run the following command:

```
svc_nas <NAS_Server> -param -f FLRCompliance -m writeverify -v 1
```

The following provides an example of enabling the write verify parameter:

```
svc_nas NAS2391 -param -f FLRCompliance -i writeverify -v
```

```
name                = writeverify
facility_name        = FLRCompliance
default_value        = 0
current_value        = 0
param_type           = NAS server
user_action           = none
change_effective     = immediate
range                = (0,1)
```

```
description = Set the writeVerify flag for FLR Compliance File Systems
```

```
detailed_description
```

```
If the writeverify parameter is enabled, all data write operations on all FLR Compliance file systems mounted on the NAS Server will be read back and verified to see whether the data has been written correctly. The system performance may degrade during this procedure due to the amount of work being performed.
```

HOW TO LOCK FILES

FLR provides the software infrastructure to protect files from deletion or modification by users and storage administrators. The procedure to lock a file requires the use of SMB or NFS protocols.

NFS ENVIRONMENT

Locking an NFS file requires setting the access time to the desired retention date and then changing the permission bits to read-only. Once successfully locked, the permission details show that the file has read-only access and the access date lists the retention date.

To specify the retention date, the `touch` command can be used to set the file's access time (`atime`). For example:

```
[root@VM test]# touch -at 202012312359 file
[root@VM test]# ls -lu --time-style=long-iso
-rw-r--r--. 1 root root 5 2020-12-31 23:59 file
```

To lock a file, the `chmod` command can be used to remove the file's write permissions. For example:

```
[root@VM test]# chmod -w file
[root@VM test]# ls -l
-r--r--r--. 1 root root 5 Sep  6 2018 file
```

The following are examples of the output when trying to alter or delete locked files.

- Trying to delete a locked file

```
[root@VM test]# rm -f file
rm: cannot remove `file': Permission denied
```

- Trying to move a locked file

```
[root@ VM test]# mv file file2
mv: cannot move `file' to `file2': Permission denied
```

- Trying to modify a locked file

```
[root@ VM test]# vi file
wq!
"file" E212: Can't open file for writing
```

- Tring to modify the attributes of a locked file

```
[root@ VM test]# chmod +w file
chmod: changing permissions of `file': Permission denied
```

WINDOWS ENVIRONMENT

Windows does not have a native UI/CLI to set retention periods and lock files. However, you can use the Windows API `SetFileTime` function. Users in an SMB environment should use the Dell EMC FLR Toolkit. Using the FLR Toolkit, you can administer and monitor files on FLR-enabled file systems. Install the Toolkit on a Windows client in the same domain as

the FLR file system you need to access. The FLR Toolkit requires DHSM to be enabled on the NAS Server. Do not check **Enforce HTTP Secure** when enabling DHSM on the NAS Server.

The tool has multiple functionalities, including:

1. **FLR Explorer** – A UI that can be used to set retention periods, lock files, run queries, and generate reports. Figure 13 shows an example of a report from FLR Explorer.

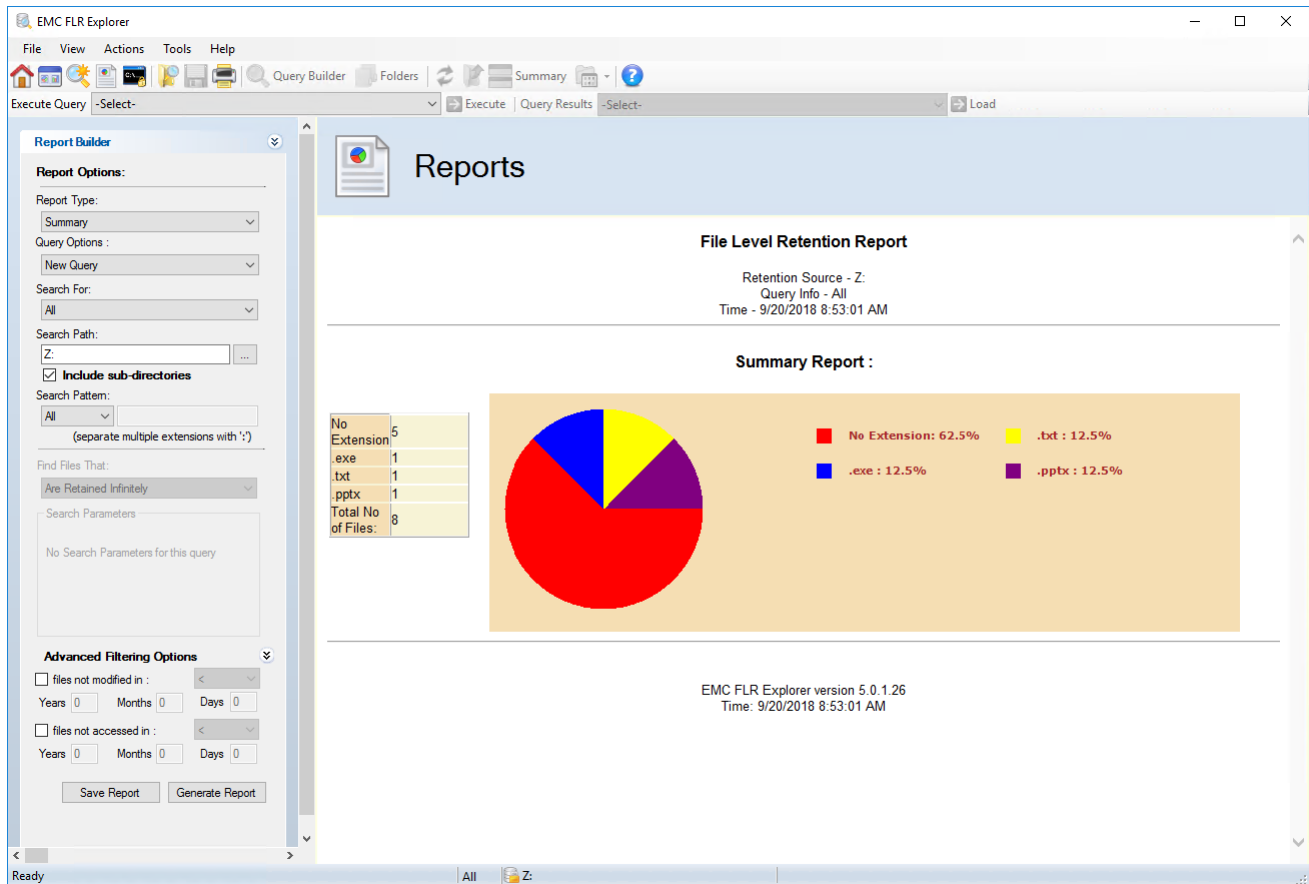


Figure 13 – FLR Explorer example

2. **FLRApply** – Provides CLI options for setting retention periods and locking files. The following are the options available as part of the FLRApply tool:

```
flrapply [/?] |
    <drive:path> [/S] [/EXCLUDEAPPEND] [/AT <YYYYMMDDHHMM>] [/NOW]
    [/RY <years>] [/RM <months>] [/RD <days>] [/RH <hours>]
    [/I <pattern>] [/E <pattern>] [/NOTIFY <number of seconds>]
    [/TC <number of threads>]
```

EXAMPLES:

```
>flrapply Z:\testdir\testfile /AT 201010101000
    Set the retention time October 10 2010 10:00AM on testfile.
>flrapply Z:\testdir\testfile1 /RY 2 /RM 3
    Extend the retention time of "testfile1" by
    2 years and 3 months.
>flrapply Z:\testdir /S /AT 201608101010 /I *.txt;*.doc /NOTIFY 1
    Set the retention time August 10 2016 10:10AM on all .txt and
    .doc files in the directory Z:\testdir and its subdirectories
```

the optimum thread count will be used and Report the status of set operation every second.

```
>flrapply Z:\testdir /RY 3 /RD 20 /E *.img;*.png;*.jpg
```

Extend the retention date of all the files, ignoring .img, .png, .jpg files in directory Z:\testdir by 3 years 20 days using optimum thread count and default notification period.

```
>flrapply Z:\testdir /S /AT 203012101000 /E *.img;*.png;*.jpg /TC 9
```

Set the retention time December 10 2030 10:00AM on all the files ignoring .img, .png, .jpg files in Z:\testdir directory and its subdirectory using 9 threads and default notification intervals.

3. **FLR Monitor Service** – A service that monitors folders in FLR-enabled file systems and takes action on them based on a user-configured policy
 - a. FLR can automatically lock and delete files with expired retention periods. Unlike auto-lock and auto-delete on the FLR Toolkit, auto-lock and auto-delete on the storage system apply to both SMB and NFS files. Figure 13 shows an example of the FLR Monitor Service.

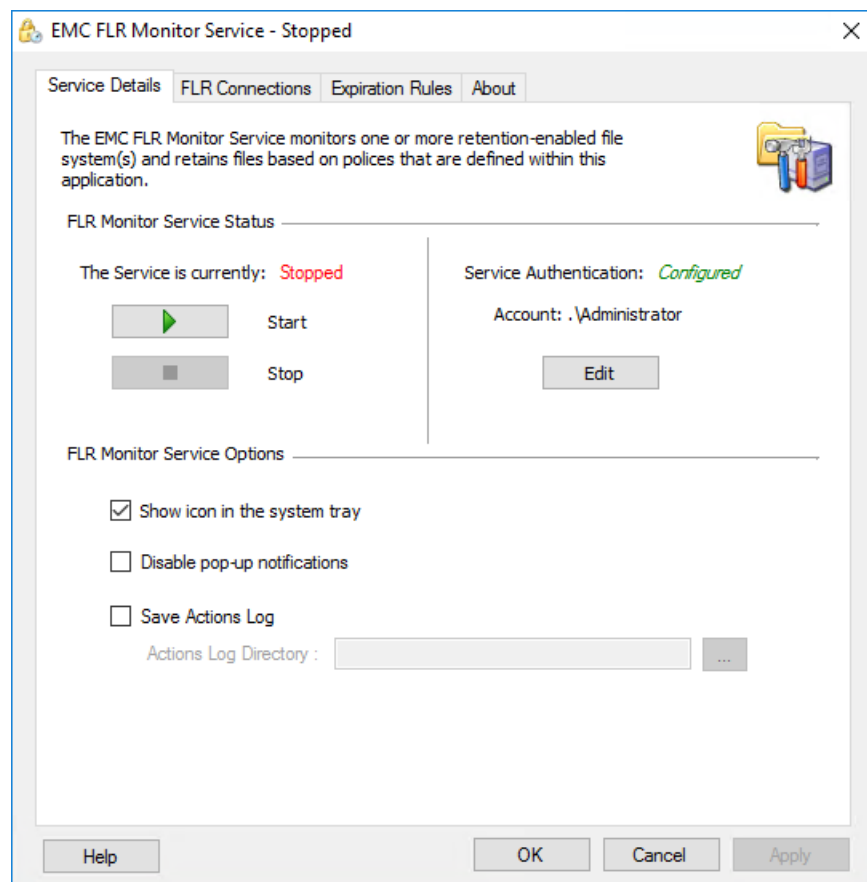


Figure 14 – FLR Monitor Service example

2. **Windows Explorer Enhancements** – The user can manually lock files with the desired retention date by using the FLR Attributes tab on the File Properties screen. The FLR Attributes tab comes with the installation of the FLR Toolkit. Figure 15 shows an example of the FLR Attributes tab under a file's properties.

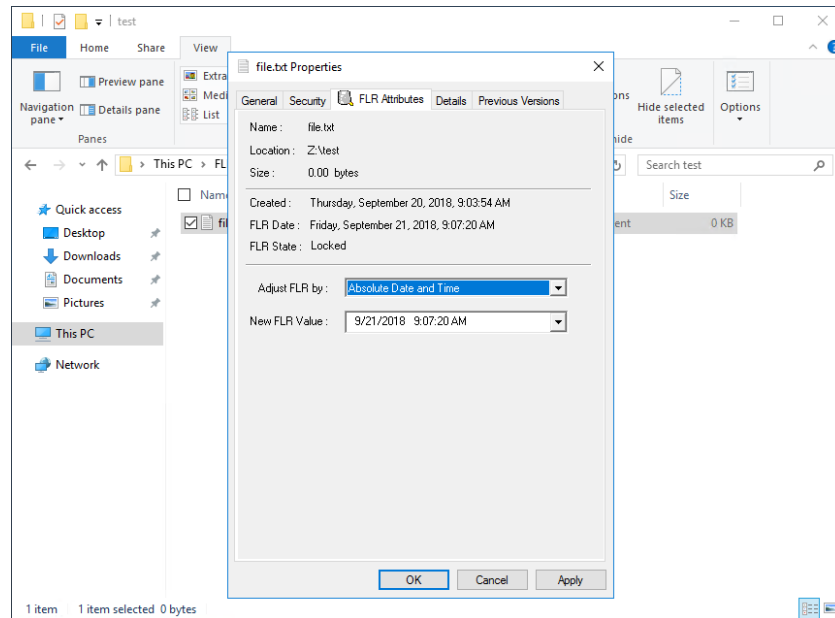


Figure 15 – FLR Attributes tab example

HOW TO CREATE APPEND-ONLY FILES

To create an Append-Only file:

1. Create an empty file
2. Remove write permissions (read-only)
3. Reapply write permissions
4. Write to the end of the file

To lock an append-only file later, set the retention date and lock it like a normal file.

The following is an example of the workflow to create an append-only file:

```
[root@VM dir]# touch append-only

[root@VM dir]# chmod -w append-only

[root@VM dir]# chmod +w append-only

[root@VM dir]# echo abc >> append-only

[root@VM dir]# cat append-only

abc

[root@VM dir]# echo 123 >> append-only

[root@VM dir]# cat append-only

abc

123

[root@VM dir]# echo xyz > append-only
-bash: append-only: Permission denied

[root@RHEL-VM1 dir]# rm -f append-only

rm: cannot remove `append-only': Permission denied
```

INTEROPERABILITY

ANTIVIRUS SCANNING

You can run an antivirus scan on FLR-enabled file systems by using Common Anti-Virus Agent (CAVA). When an infected locked file is identified, the resident AV engine records the infection and its location in the log file of the resident scan engine. Administrators cannot repair or remove an infected file. They can delete the file only after its retention date has passed. However, administrators can change the file's permission to restrict read access, ensuring that the file is unavailable to users. CAVA's scan-on-first-read functionality does not detect a virus in a locked file. A subsequent scan of the file system is necessary to detect viruses on locked files.

DATA REDUCTION

You can enable Data Reduction, including Advanced Deduplication, on an FLR-C or FLR-E file system without compromising the protection offered to the data on the file system.

FILE TIERING WITH CLOUD TIERING APPLIANCE (CTA)

Archiving an FLR-enabled file system to the cloud is not supported. It is not possible to use a file system with FLR enabled as primary source in CTA. However, Unity FLR-enabled file systems can be used as a destination repository when tiering from Celerra, VNX, or NetApp systems. Refer to the *Dell EMC Unity: Cloud Tiering Appliance (CTA)* on Dell EMC Online Support for more details.

NDMP BACKUP

FLR supports NDMP backup and restore, but it does not preserve the lock status. It is possible to back up protected files from an FLR file system and restore them to a non-FLR file system. You can also back up from a non-FLR file system and restore them to an FLR file system.

NDMP backups include retention period and permissions, but not lock status. After restore:

- All read-only files on the file system are locked with their respective retention period, even if they were not previously locked
- Append-only files are no longer locked since they have write permissions
 - It is recommended to lock these files and create a new append-only file
- Administrators should ensure that files were restored to a FLR-enabled file system

When you restore an expired locked file to an FLR file system, the file is locked with the default retention period. If the default retention period is set to an infinite retention period, an FLR-E file system treats the infinite retention period as "soft", which means that you can decrease the retention period. If the expired file is restored to an FLR-C file system, the infinite retention period is "hard", which means that you cannot decrease the retention period, and you cannot delete the file.

SNAPSHOTS

FLR-C

A snapshot of an FLR-C file system inherits the associated FLR attributes.

- FLR-C only supports read-only snapshots. Read-Write snapshots are not supported by FLR-C file systems.
- FLR-C does not support snapshot restores

FLR-E

FLR-E file systems support snapshots and do not conform to the SEC compliance requirements. Read-Write snapshots are also supported in FLR-E file systems. If you restore a snapshot to an FLR-E file system, note that restoring the snapshot could overwrite protected files.

REPLICATION

If the source file system is FLR-enabled, the destination configuration must be FLR-enabled as well.

- Both source and destination systems must be running Dell EMC Unity OE 4.5 or later.
- If provisioning the destination file system manually using UEMCLI, ensure the FLR type matches.
- Any changes to FLR settings on the source resource are replicated to the destination resource.

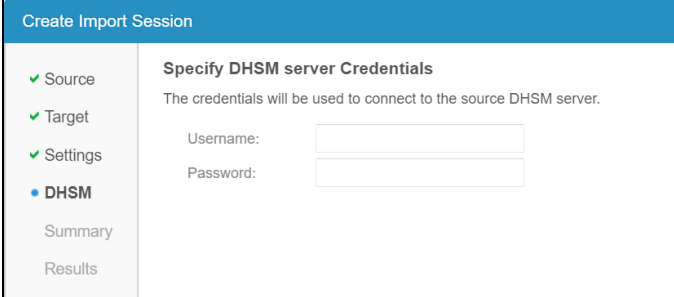
DELL EMC UNITY NATIVE FILE IMPORT

FLR is supported when using Native File Import to migrate from VNX to Unity.

- If the source file system is FLR-enabled, the destination is provisioned with the same FLR settings
- No ability to reconfigure FLR settings during import

Retention periods and lock status are migrated.

If an FLR file system is being imported, DHSM must be configured on the VNX. Enter the DHSM credentials in the Import wizard, as shown in Figure 16.



The screenshot shows a web-based wizard titled "Create Import Session". On the left is a vertical navigation pane with the following items: "Source" (checked), "Target" (checked), "Settings" (checked), "DHSM" (selected with a blue dot), "Summary", and "Results". The main content area is titled "Specify DHSM server Credentials" and includes the instruction "The credentials will be used to connect to the source DHSM server." Below this are two input fields: "Username:" and "Password:".

Figure 16 – DHSM credentials in the Import Session wizard

DESIGN CONSIDERATIONS

This is a recap of the FLR concepts and the things we need to keep in mind.

The following are design considerations when planning on using FLR:

- Non-FLR, FLR-C, and FLR-E file systems can exist on the same system.
- Select the required FLR setting (Off, Enterprise, or Compliance) when you create the file system. You cannot change the setting after the file system is created.
- You can enable Data Reduction and Advanced Deduplication on an FLR-enabled file system without affecting the integrity of the data.
- FLR-E is intended for self-regulated archiving. FLR-C is intended to assist companies that must comply with regulations such as the U.S. SEC ruling 17a-4(f).
- CAVA's scan-on-first-read functionality does not detect a virus in a locked file. A subsequent scan of the file system is necessary to detect viruses on locked files.
- You can use a file system with FLR enabled as a destination NAS repository for CTA. However, you cannot use an FLR-enabled file system as primary storage for CTA. Refer to the *Dell EMC Unity: Cloud Tiering Appliance (CTA)* on Dell EMC Online Support for more details.
- NDMP:
 - Although FLR supports all backup functionality, the lock status is not preserved in the NDMP backup
 - Upon restore, all read-only files on the FS are locked with their respective retention period
 - Even if they were not previously locked
 - Append-only files are no longer locked after restore
 - The Unity administrator must ensure that the file system is restored to a Unity file system with FLR enabled.
- You cannot create a writable snapshot of an FLR-C file system. However, you can create a writable snapshot of an FLR-E file system.
- You cannot change a FLR enabled file system from an FLR-E file system to an FLR-C file system, or vice versa.
- Most file copy tools have options to preserve the timestamps of a file being copied.
- When copying files into an FLR file system with automatic locking enabled, it is possible for the files to be locked almost immediately after being copied into the FLR file system. The files immediately meet the criteria for being locked automatically due to modified time being preserved.
- For example, assume the lock policy interval is set to 30 minutes. File A was modified 15 minutes ago. File B was modified 60 minutes ago. File A is not locked when copied into the FLR file system because its modification time

is less than the lock policy interval. However, File B is locked when copied into the FLR file system because its modification time is more than the lock policy interval. This example does not account for the file system scan interval, which may add a delay before the file is locked.

CONCLUSION

Dell EMC Unity storage systems offer a cost-effective solution when files require file-level retention to ensure data integrity for the remainder of their lifecycle. FLR is included with all Dell EMC Unity systems at no additional cost. FLR provides a robust software infrastructure that protects locked files from accidental or malicious attempts of modification or deletion. FLR is suitable if you are looking to self-regulate your record-keeping practices or if you are required to meet strict compliance rules of U.S. SEC Rule 17a-4 (f) for digital data storage. FLR is simple to manage and use and allows you to take advantage of the Unity storage efficiency features to reduce the storage footprint.

REFERENCES

The following references can be found on Dell EMC Online Support:

- Dell EMC Unity: Cloud Tiering Appliance (CTA)
- Dell EMC Unity: Data Reduction
- Dell EMC Unity: Introduction to the Platform
- Dell EMC Unity: Unisphere Overview
- Dell EMC Unity: NAS Capabilities
- Dell EMC Unity: Snapshots and Thin Clones
- Dell EMC UnityVSA
- Unisphere Command Line Interface User Guide
- Unisphere Management REST API Programmer's Guide
- Unisphere Management REST API Reference Guide