FluidFS and macOS - Best Practices Guide

Dell EMC FluidFS Network Attached Storage (NAS)

FluidFS System Engineering
February 2018

Acknowledgements

This white paper was produced by Bryan Lusk and the FluidFS Systems Engineering Team.

Authors:	Bryan Lusk and the FluidFS Systems Engineering team

Feedback

Please give us feedback on the quality and usefulness of this document by sending an email to:

FluidFS-System-Engineering@Dell.com

Revisions

Revision	Date	Description
A	February 2017	Initial Release
B	February 2018	Changed terminology from "Mac OS X" to "macOS" Cleaned up formatting Added section 3.7 – Time Machine Added clarification in section 2.1 about SMB Encryption

Table of contents

1 Preface

1.1 Audience

The audience for this document is intended to be systems, networking, storage or backup administrators who are responsible for the day-to-day management responsibilities of a FluidFS NAS solution and/or the macOS clients which are accessing FluidFS.

1.2 Purpose

The purpose of this document is to provide guidance to help the storage administrator or macOS administrator to best configure the FluidFS cluster and macOS clients to work together. This document is not intended to be a primer or FluidFS introductory resource for any of the subject matters involved, and it assumes at least introductory knowledge of many of the subjects covered in this document.

This document should be used in conjunction with other FluidFS resources as listed in Appendix B – Additional Resources.

1.3 Disclaimer

The information contained within this best practices document is intended to provide general recommendations only. Actual configurations in customer environments may need to vary due to individual circumstances, budget constraints, service level agreements, applicable industry-specific regulations, or other factors. Configurations should be tested before implementing them a production environment.

Furthermore, please note that starting with macOS 10.12 (Sierra), Apple has rebranded the Macintosh operating system from “OS X” to “macOS”. This brings macOS more in line with the names of the other Apple operating systems (such as iOS, watchOS, and tvOS). This document uses “macOS” when referencing macOS and/or Mac OS X.

2 MacOS and NAS Storage

2.1 Protocol Support of macOS

A macOS client is capable of accessing FluidFS using either the NFS or SMB protocol. FluidFS does not support the use of the Apple File Protocol (AFP). Typically, a storage administrator or IT department will have a preferred protocol (SMB or NFS) based on several factors, which will be discussed in the next section.

MacOS supports several different versions of the SMB and NFS protocol as detailed in the following table. FluidFS supports all of the protocol versions listed below.

macOS Version	SMB Versions Supported	NFS Versions Supported*
OS X 10.8 and previous	SMB 1.0	NFS 3, 4
OS X 10.9 (Mavericks)	SMB 1.0, 2.0, 2.1	NFS 3, 4
OS X 10.10 (Yosemite)	SMB 1.0, 2.0, 2.1, 3.0	NFS 3, 4
OS X 10.11 (El Capitan)	SMB 1.0, 2.0, 2.1, 3.0	NFS 3, 4
macOS 10.12 (Sierra)	SMB 1.0, 2.0, 2.1, 3.0	NFS 3, 4
macOS 10.13 (High Sierra)	SMB 1.0, 2.0, 2.1, 3.0	NFS 3, 4

*-macOS also supports NFS version 2, but FluidFS does not support it

SMB, SMB Signing, Encryption and Multichannel

Please note that the SMB 1.0 protocol is often referred to as the CIFS (Common Internet File System) protocol. Dell EMC recommends using the newest version of the SMB protocol possible (by using the newest version of macOS possible). Additionally, FluidFS supports SMB 3.0.2 and SMB 3.1.1, but those versions of the SMB protocol are not supported by macOS at this time.

The SMB 3.0 Protocol includes a feature called SMB Signing which uses cryptography to digitally sign SMB communications to prevent against man in the middle attacks. Apple started enabling SMB signing by default in macOS 10.11.5, which can provide some additional security benefits. However, this added security does come with a bit of a performance penalty. SMB signing can be disabled on macOS to provide a little bit of a performance benefit. This is detailed in section 3.6.

For customers who have high security environments, some will want even more security than is given by SMB Signing. For this requirement, FluidFS supports AES-based encryption for SMB 3.0 clients, which encrypts the entire SMB packet. In order to use SMB Encryption, enable it in the SMB share properties for FluidFS, and reconnect the SMB clients (macOS). In order to **require** encryption for all SMB clients connecting to FluidFS, the administrator can edit the SMB Protocol settings for the FluidFS cluster to require encryption for all SMB clients. By default macOS will not use encryption even if it is enabled on the SMB share, but if FluidFS is set to require encryption, macOS will use it.

NOTE: The use of SMB Encryption will introduce a noticeable performance penalty for SMB file operations. The performance penalty for using SMB Encryption is greater than the performance penalty to use SMB Signing.

At the time of writing this document, SMB Multichannel is officially supported by FluidFS (starting in v6). However, very little information is available from Apple (or publicly from the community) on how to implement SMB Multichannel on macOS clients. Specifically, very little is available on how to implement Receive Side Scaling (RSS), which is a requirement for SMB Multichannel clients with a single network interface. RSS is used to distribute the network load across more than one CPU core.. Also, in order to use SMB Multichannel, it is best to have multiple physical network interfaces. If multiple network interfaces are being used, RSS is not required. In addition to any client-side configuration for SMB Multichannel, the feature must also be enabled on FluidFS in the SMB Protocol settings (as it is disabled by default).

NFS

If the NFS protocol is the preferred choice, the Dell EMC recommendation at the time of writing this article is NFS3. NFS3 will give better performance than NFS4, better application compatibility than NFS4, and easier setup and use than NFS4.

FTP

FluidFS also supports the FTP protocol, which can be used by macOS clients to access files on FluidFS. However, this paper will focus on SMB and NFS, which are more commonly used for day to day file storage access than FTP.

2.2 How to choose whether to use NFS or SMB

Typically, an IT organization will have a general preference on whether they want to use the NFS or SMB protocol to access data stored on a NAS device. This preference usually arises from one of the following common reasons:

- **Familiarity**: What the IT organization is most comfortable with from a technical standpoint. I.E. If the IT Organization is a “Windows Shop” or a “UNIX Shop”. Mac users are all about user-friendly applications/OS. Windows shops typically prefer SMB, and UNIX shops typically prefer NFS
- **Permissions**: What types of permissions are on the data? If the NAS data has predominantly NTFS Access Control Lists (permissions) to delegate access, this data will be managed and accessed the best using the SMB Protocol. If the NAS data has predominantly POSIX permissions, or NFS4 Access Control Lists, then the data will be managed and accessed the best using the NFS protocol
- **Authentication**: Similar to the previous item, what type of authentication is in use? If Active Directory, that typically is paired with accessing NAS data using the SMB protocol. If using LDAP or NIS (or Active Directory-based LDAP), that typically is paired with accessing NAS data using the NFS protocol.
- **Performance**: Typically the NFS protocol will provide better performance than the SMB protocol. However, on FluidFS, performance is very good with either NFS or SMB.
- **Interoperability**: Will both Windows and Mac clients be accessing the data? If so, SMB will most likely be a better choice because SMB is native to Windows, and macOS can work with both SMB and NFS. SMB provides the benefit over NFS in that Alternate Data Streams can be used to store resource fork data, such that the Windows clients won’t see it. If NFS is in use, Windows clients would see the “dotbar” (._) files that the macOS clients would be using to store resource forks (metadata). However, Mac clients retrieve resource fork data slower over SMB than NFS, which could cause frustration if there are thousands of files or more in directories.

Taking into consideration ease of use, performance, and interoperability, the SMB protocol is the Dell EMC general recommendation on protocol to access FluidFS from macOS clients. That being said, there are certainly some customers who will prefer to use the NFS protocol, and that is supported as well.

2.3 How macOS stores metadata on NAS devices

MacOS clients behave differently than other SMB or NFS clients when it comes to how they store macOS-specific metadata on NAS devices. They use a special data type called a **resource fork** or **alternate data stream**, which is represented on NAS devices by a separate file.

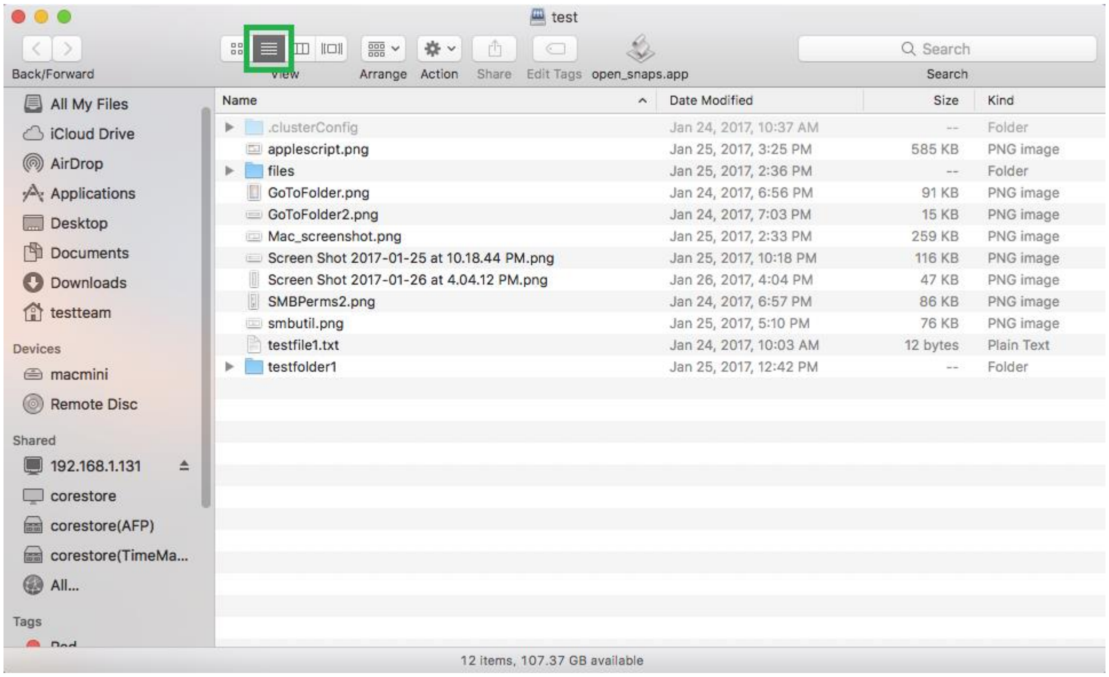
On an **SMB share**, the resource forks are more commonly referred to as **Alternate Data Streams** (or ADS for short). On a FluidFS SMB share which macOS clients are storing data on, there will typically be a hidden folder called ".AppleDouble" which stores all of the Alternate Data Streams/macOS metadata. When accessing FluidFS SMB shares, FluidFS will automatically hide this folder when browsing. However, if the same NAS volume/folder has an NFS export defined on it, when it's browsed using NFS, the .AppleDouble folders will be seen there. Inside that hidden .AppleDouble folder, there are *folders* that correspond to the *files* which have Alternate Data Streams. Inside these *folders*, which correspond to *files* that have ADS's, files exist that start with "com.apple.", which actually contain the macOS specific metadata. Please note that Windows SMB clients occasionally use Alternate Data Streams as well, for example, for files downloaded from the internet. Windows clients accessing SMB shares on FluidFS could also store Alternate Data Streams on files, and it would result in the same behavior noted above regarding the usage of the hidden .AppleDouble folder.

On a FluidFS **NFS export** which macOS clients are storing data on, there are **resource forks**, which are represented by hidden files which have the filename of "._OriginalFile.txt", if the file written by the user was called "OriginalFile.txt". This hidden file which has the "._" prefix is the resource fork file, which is storing the macOS specific metadata, and it is located in the same folder as the file or folder it is representing.

The macOS Finder will not show these hidden files (SMB or NFS) since they are metadata. On a Windows client, if "Show Hidden Files" is enabled, hidden files which start with "." will be seen. On a Linux/UNIX NFS client, one will only see the hidden files which start with "." if they do a directory listing with the "-a" flag.

2.4 Finder Best Practices

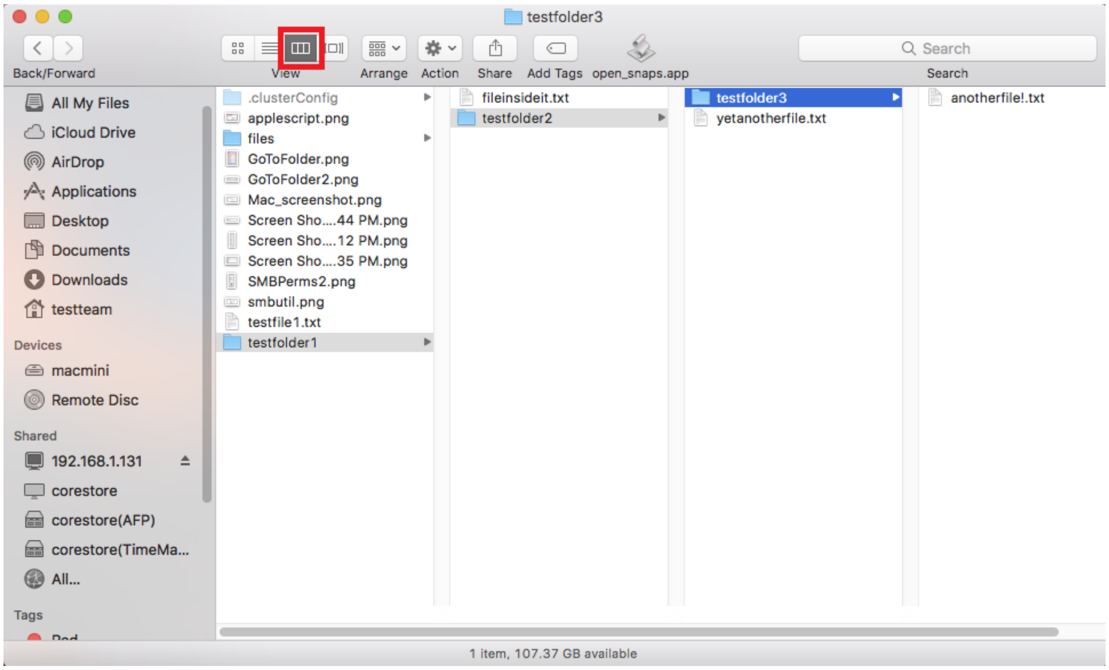
Dell EMC recommends to avoid the use of the “Column View” in finder. The “List View” method is highly preferred for performance reasons. Here is a screenshot of the List View (preferred, and the **icon** to select it):



Name	Date Modified	Size	Kind
▶ .clusterConfig	Jan 24, 2017, 10:37 AM	--	Folder
applescript.png	Jan 25, 2017, 3:25 PM	585 KB	PNG image
▶ files	Jan 25, 2017, 2:36 PM	--	Folder
GoToFolder.png	Jan 24, 2017, 6:56 PM	91 KB	PNG image
GoToFolder2.png	Jan 24, 2017, 7:03 PM	15 KB	PNG image
Mac_screenshot.png	Jan 25, 2017, 2:33 PM	259 KB	PNG image
Screen Shot 2017-01-25 at 10.18.44 PM.png	Jan 25, 2017, 10:18 PM	116 KB	PNG image
Screen Shot 2017-01-26 at 4.04.12 PM.png	Jan 26, 2017, 4:04 PM	47 KB	PNG image
SMBPerms2.png	Jan 24, 2017, 6:57 PM	86 KB	PNG image
smbutil.png	Jan 25, 2017, 5:10 PM	76 KB	PNG image
testfile1.txt	Jan 24, 2017, 10:03 AM	12 bytes	Plain Text
▶ testfolder1	Jan 25, 2017, 12:42 PM	--	Folder

12 items, 107.37 GB available

And here is a screenshot of the Column View (not preferred, and the **icon** to avoid selecting):



.clusterConfig
applescript.png
files
GoToFolder.png
GoToFolder2.png
Mac_screenshot.png
Screen Sho....44 PM.png
Screen Sho....12 PM.png
Screen Sho....35 PM.png
SMBPerms2.png
smbutil.png
testfile1.txt
testfolder1

fileinsideit.txt
testfolder2

testfolder3
yetanotherfile.txt

anotherfile!.txt

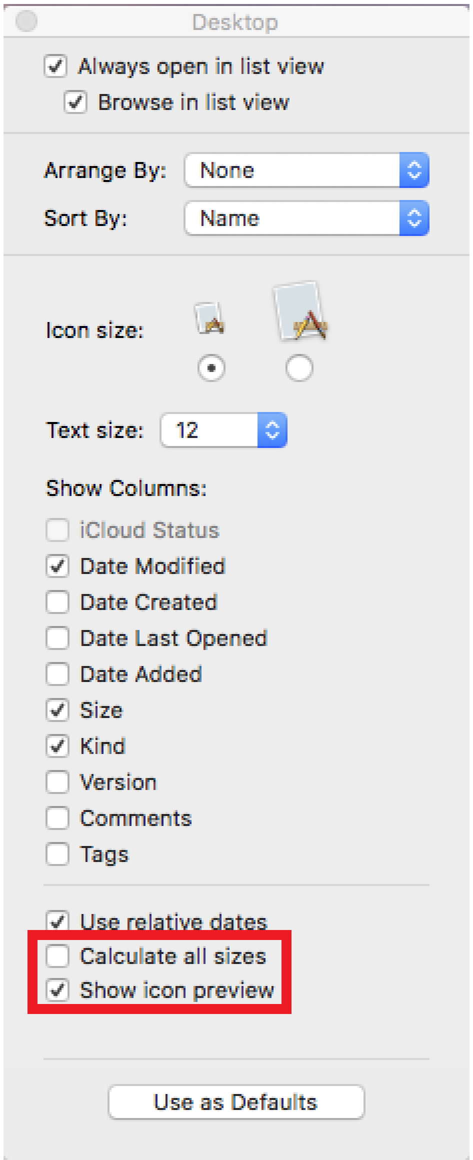
1 item, 107.37 GB available

The Column View will show several levels of the directory structure simultaneously. Each column shows a different level in the directory tree. This view is very popular and useful, but it can introduce a performance penalty. Finder will not display a directory listing until it's received the full directory list, as well as information from the resource forks for each file or folder. In Column View, Finder is working much harder to keep directories refreshed constantly, resulting in many more calls between macOS and FluidFS. Typically, the user really only wants to see the contents of one directory, but may suffer a delay of multiple minutes to retrieve a directory listing, because Finder is working on many directories instead of just the one the user cares about.

The Icon View doesn't have as many drawbacks as the Column View from a performance standpoint, but it does attempt to load icon previews (by default). This icon preview loading behavior can be disabled, which is detailed in the next paragraph, but doing that makes the Icon View lose its value. If the user favors the Icon View, keep in mind it does come at a small performance penalty because it needs to load the icon preview for each file.

Dell EMC recommends List View due to these factors. There are a couple more optimizations which can be done in Finder as well:

- Disable "Calculate All Sizes". When this feature is enabled, Finder will crawl/tree walk the directories which are accessed (and its subdirectories) to add up how much space on disk that directory (and its children) are consuming. Disabling this can give some performance benefits when browsing file data in Finder.
    - This is disabled by clicking "View" in the top bar, going to "Show View Options", and uncheck "Calculate all sizes". The user must be in "List View" to get this option. This is shown in the screenshot below.
- Disable "Show Icon Preview". Getting icon previews requires reading the icon file from the FluidFS NAS device. If the user doesn't care about this, it could reduce some of the overhead in Finder by disabling it.
    - This is disabled by clicking "View" in the top bar, going to "Show View Options", and uncheck "Show Icon Preview". This is shown in the screenshot below.



Desktop

☑ Always open in list view

☑ Browse in list view

Arrange By: None

Sort By: Name

Icon size: ● ○

Text size: 12

Show Columns:

□ iCloud Status

☑ Date Modified

□ Date Created

□ Date Last Opened

□ Date Added

☑ Size

☑ Kind

□ Version

□ Comments

□ Tags

☑ Use relative dates

□ Calculate all sizes

☑ Show icon preview

2.5 Multi-Protocol considerations

If an organization has both SMB and NFS clients accessing the same dataset, certain considerations must be taken into account. Please read over the document *FluidFS in a multiprotocol environment* to understand how FluidFS works in these types of situations, specifically in relation to mapping AD users to LDAP/NIS users. It is always the best and smoothest to use centralized user account management, meaning Active Directory for SMB users, and LDAP/NIS for NFS users. Alternatively, FluidFS local users can be utilized for either protocol.

In addition to the authentication related considerations, consider the differences in how macOS stores metadata between NFS and SMB. If the “dotbar files” that are created by macOS clients accessing NFS exports will be a problem for Windows users accessing FluidFS over SMB, one could configure the macOS clients to access FluidFS using SMB instead. However, it is important to note that performance will usually be better with NFS compared to SMB. Alternatively, macOS clients could continue accessing over NFS, and the FluidFS SMB share feature called “hide dot files” could be enabled, which would automatically make all files and folders on that SMB share that start with a dot hidden.

3 FluidFS and macOS – SMB Best Practices

3.1 How to access via SMB using Finder

The most common and easiest way to map an SMB share using Finder is by using the “Go” menu at the top of the screen, and select “Connect to Server”. The syntax for the “Server Address” field is as follows:

To mount an SMB share using SMB2 or SMB3 (the highest supported by that version of macOS), use this syntax:

```
smb://<DOMAIN>;<user>@<VIP>/<share>
```

For example, if we have the following:

- Active Directory domain name of ‘mycompany.com” with a NetBIOS name of MYCOMPANY
- User name of “John_Doe”
- FluidFS Virtual IP: 192.168.1.100 (could also use a hostname here)
- Share: test

The “Server Address” in the “Connect to Server” menu would be:

```
smb://MYCOMPANY;John_Doe@192.168.1.100/test
```

To mount an SMB share using SMB1, use this syntax below. The only difference is changing “smb://” for “cifs://”. Mounting using SMB1 will give slower performance, but can be useful for troubleshooting purposes, or for accessing snapshot data (which is detailed later in the document).

```
cifs://<DOMAIN>;<user>@<VIP>/<share>
```

After entering this information and clicking “Connect”, the user should be prompted for their password. After the user enters the correct password, the Finder will go directly to the root of the share that was specified.

MacOS also has the ability to keep this share mounted forever, even after a reboot. To keep the share mounted, follow these steps:

1. Mount the share in Finder
2. In System Preferences, go to Users & Groups
3. Select the relevant user account and select Login Items
4. Drag the icon of the volume that was mounted from the desktop to the list of login items
5. Close System Preferences

3.2 SMB Authentication and Single Sign On (SSO)

When a user accesses any NAS Storage device using SMB, the SMB server (FluidFS) allows or denies access by comparing the SID (Security Identifier) of the accessing user against the Access Control List of the file or folder they are attempting to access. However, from the user's perspective, they are providing a user name and password. In the background, this username is translated into a Security Identifier, or SID, as well as group membership. SID's take the form of 'S-1-5-21-549688327-91903405-2500298261-1000'. Once the username is translated into SID and the relevant Group SID's for group membership, the SMB server will compare that to the files permissions, and grant or deny access.

The SMB server must have a repository to do lookups in, to translate the user names into SID's (and relevant group membership for that user too, along with the Group SID's). For SMB users, this repository is either Active Directory, or the FluidFS Local Users database.

Both FluidFS and macOS support Active Directory. Therefore, for macOS clients and FluidFS clusters that are joined to the same Active Directory forest, Single Sign On (SSO) is possible for SMB shares. The Directory Utility is used in macOS to join the system to Active Directory. Once the macOS system is joined to Active Directory, all that needs to be done is to log into macOS as an Active Directory user that has permission to access FluidFS shares. No authentication prompt should be given, since the macOS client and FluidFS are both joined to the same Active Directory domain, and exchange Kerberos tickets to prove their identity to each other.

Please note that in order to join an macOS client to Active Directory, the time on the macOS client and the Active Directory domain must be no more than 5 minutes apart. If they are more than 5 minutes apart, Kerberos will cease to function, which is needed for Active Directory. One good way to keep clients' time synchronized with Active Directory is to configure Network Time Protocol (NTP) on the macOS clients, and configure them to query the time from the Active Directory domain controllers themselves.

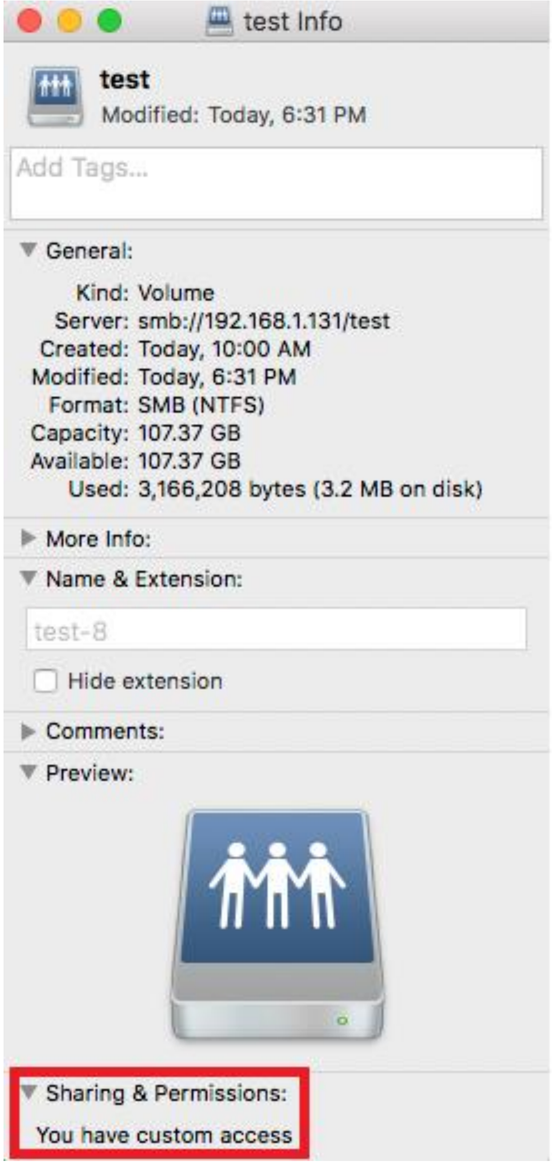
Additionally, FluidFS local users can be utilized to access SMB shares on FluidFS. Single Sign On won't work in this case, so the domain, username, and password must be specified when going through the "Connect to Server" dialog, as detailed in section 3.1.

3.3 Permissions for macOS SMB Clients

On a Windows system, when an SMB client views the permissions on a file or folder on an SMB share by right clicking and selecting Properties -> Security tab, they can see all of the users and groups who have access, and what level of access they have.

On a macOS system, when right clicking a file and selecting Get Info, and looking at the Sharing & Permissions section, will always result in macOS Finder giving a message of "You have custom access". This is due to macOS Finder not being able to natively display Windows Access Control Lists (ACL's) on an SMB server. This is true for other industry NAS devices as well. MacOS Finder supports POSIX ACL's. MacOS does not support NTFS/Windows ACL's, which is why it will always display "You have custom access", as is shown in the following screenshot.

Note: It is possible to view NTFS/Windows ACL's using Terminal on macOS (described later in this section)



test Info

test

Modified: Today, 6:31 PM

Add Tags...

▼ General:

Kind: Volume
Server: smb://192.168.1.131/test
Created: Today, 10:00 AM
Modified: Today, 6:31 PM
Format: SMB (NTFS)
Capacity: 107.37 GB
Available: 107.37 GB
Used: 3,166,208 bytes (3.2 MB on disk)

▶ More Info:

▼ Name & Extension:

test-8

□ Hide extension

▶ Comments:

▼ Preview:

▼ Sharing & Permissions:

You have custom access

It is important to note that even though permissions are not displayed properly on a macOS SMB client, they *are enforced properly* by FluidFS. That is because per the SMB protocol, the permissions enforcement is done by the SMB Server, which is FluidFS, and that function behaves the same regardless of what type of OS is behind the SMB client (macOS, Windows, Linux, etc.).

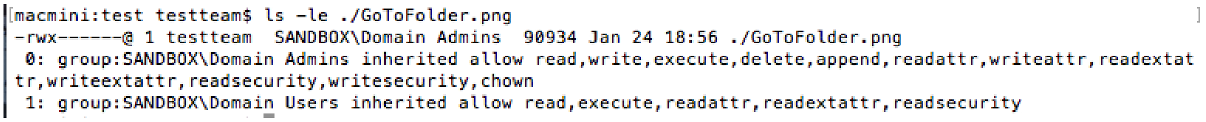
On SMB shares hosted on FluidFS, the Access Control List can be set on the root of a share through Dell Storage Manager or the FluidFS CLI, and all files and folders beneath that root level of the share can inherit permissions from the parent (root of the share). This could be used to control the permissions on an SMB share which is accessed purely only by macOS clients.

To create more granular Access Control Lists on subdirectories, a Windows system will need to be used to set the NTFS Access Control Lists.

Please note that since macOS Finder cannot view or modify Windows ACL’s over SMB, it is highly preferable to use inherited ACL’s to delegate access on SMB shares and folders that macOS clients are creating new files in.

For more tech savvy users, it is possible to view NTFS/Windows ACL’s using Terminal. One can enter the following command, and change out <*file_or_folder_name*> with the file or folder that is desired to view the Windows ACL on. This must be done on an macOS client that is joined to the same Active Directory domain as the FluidFS cluster.

```
ls –le <file_or_folder_name>
```



```
macmini:test testteam$ ls -le ./GoToFolder.png
-rwx------@ 1 testteam  SANDBOX\Domain Admins  90934 Jan 24 18:56 ./GoToFolder.png
 0: group:SANDBOX\Domain Admins inherited allow read,write,execute,delete,append,readattr,writeattr,readextat
tr,writeextattr,readsecurity,writesecurity,chown
 1: group:SANDBOX\Domain Users inherited allow read,execute,readattr,readextattr,readsecurity
```

For UNIX style NAS volumes with NFS exports on them, macOS clients can properly view and edit POSIX permissions on NFS exports. This is explained in the NFS Best Practices section.

3.4 Accessing Snapshots via SMB

FluidFS contains a feature that allows end users to access snapshot data directly, so they can self-restore files from snapshots in the case of accidental deletion, or saving changes they wish to revert.

The Dell EMC recommendation for accessing and restoring snapshot data, using the SMB protocol on files that have Windows NTFS ACL’s on them, is to use a Windows client instead of a macOS client. We recommend this because:

- **Ease of Use**: Windows has a built in facility for accessing snapshot data called “Previous Versions” which makes accessing and restoring files from snapshots very easy. On macOS it is a more manual procedure, which is described later on in this section.
- **Permissions**: If files are restored from a snapshot using a Windows system, it is much more likely that the NTFS ACL/permissions on it will be preserved. That is because if the restore is done from macOS, since macOS does not support Windows/NTFS ACL’s, the permissions may not be properly preserved (primarily if inheritance is not in use).

Alternatively, if the data has standard POSIX permissions on it, and it is being accessed over NFS, restoring files from snapshot data over NFS from macOS is much less problematic, and is detailed in section 4.4.

With those recommendation in mind, if the desire is still to access snapshot data from macOS over SMB, it is certainly possible. From a macOS system, accessing a FluidFS SMB share, snapshot data is accessed as follows.

Using Finder

To prepare, it is best to set Finder to show all files, including those that start with a dot (.) . In order to do this, open Terminal, and enter the following commands to force Finder to show all files

```
defaults write com.apple.finder AppleShowAllFiles TRUE

killall Finder
```

This effect can be disabled as follows:

```
defaults write com.apple.finder AppleShowAllFiles FALSE

killall Finder
```

For next steps, due to the special implementation of the SMB client on macOS, one must choose between two workarounds to implements on the SMB client in order to view snapshot data over SMB:

1. Use SMB1 to mount the SMB share. Please note that a Windows client can use any SMB version to access snapshot data.

   ```
   cifs://<DOMAIN>;<user>@<VIP>/<share>
   ```

   For example:

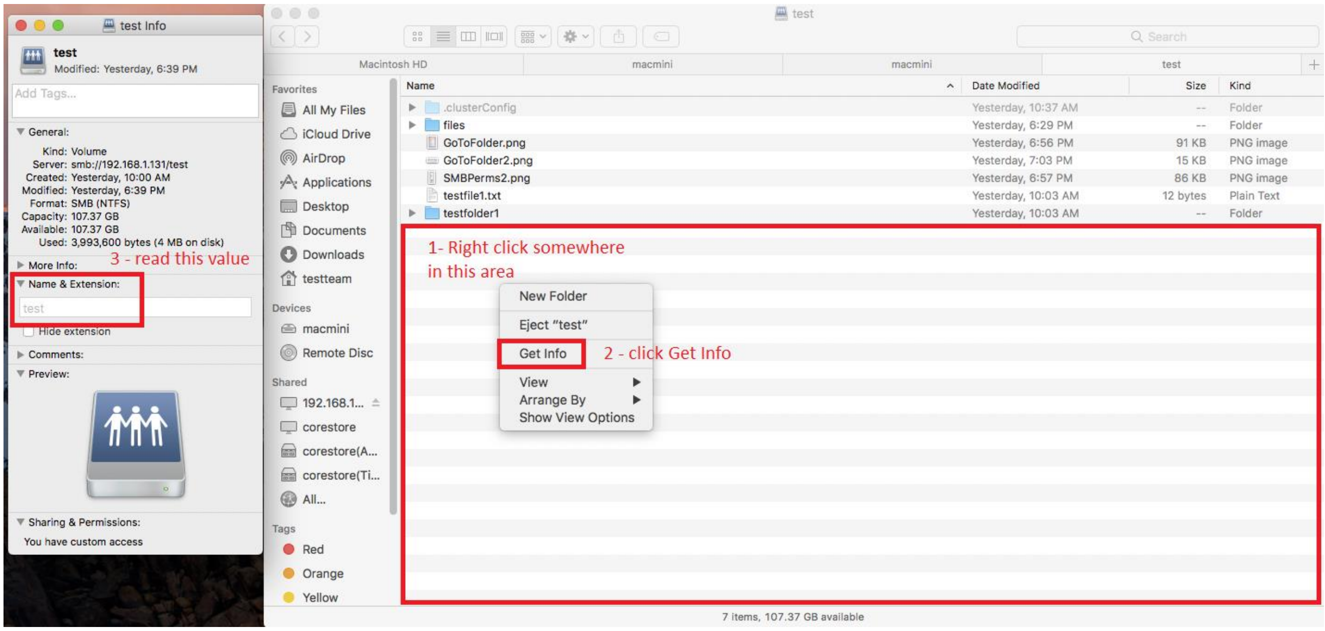
   ```
   cifs://MYCOMPANY;John_Doe@192.168.1.131/test
   ```

2. Use SMB2 or SMB3, but disable File ID's on the macOS client. macOS hides the .snapshots directory because the file ID's for the .snapshots folder, and its children, conflict with the file ID's for the non-snapshot versions of the files and folders.

   To disable File ID's on the macOS client, add the following line to the */etc/nsmb.conf* file on the macOS client:

   ```
   file_ids_off=yes
   ```

Next, the user must determine the mount point of the share. This is done by right clicking on the white space for the share and click "Get Info" to see the "Name & Extension" field.



test Info

test
Modified: Yesterday, 6:39 PM

General:

Kind: Volume
Server: smb://192.168.1.131/test
Created: Yesterday, 10:00 AM
Modified: Yesterday, 6:39 PM
Format: SMB (NTFS)
Capacity: 107.37 GB
Available: 107.37 GB
Used: 3,993,600 bytes (4 MB on disk)

Name & Extension: test

Sharing & Permissions:
You have custom access

Name	Date Modified	Size	Kind
.clusterConfig	Yesterday, 10:37 AM	--	Folder
files	Yesterday, 6:29 PM	--	Folder
GoToFolder.png	Yesterday, 6:56 PM	91 KB	PNG image
GoToFolder2.png	Yesterday, 7:03 PM	15 KB	PNG image
SMBPerms2.png	Yesterday, 6:57 PM	86 KB	PNG image
testfile1.txt	Yesterday, 10:03 AM	12 bytes	Plain Text
testfolder1	Yesterday, 10:03 AM	--	Folder

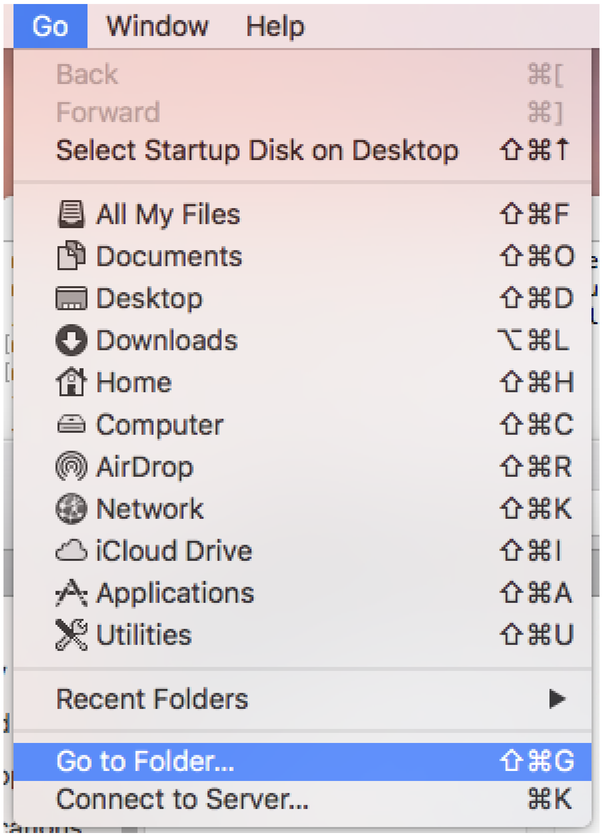
7 items, 107.37 GB available

1- Right click somewhere in this area

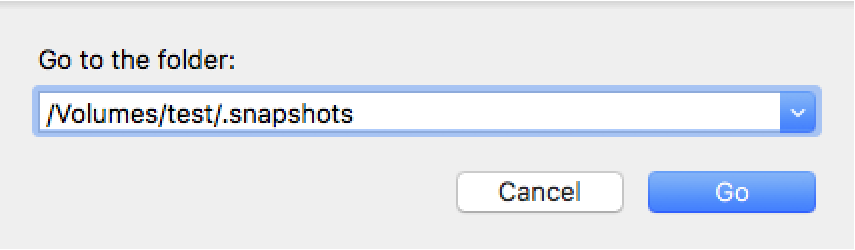
2 - click Get Info

3 - read this value

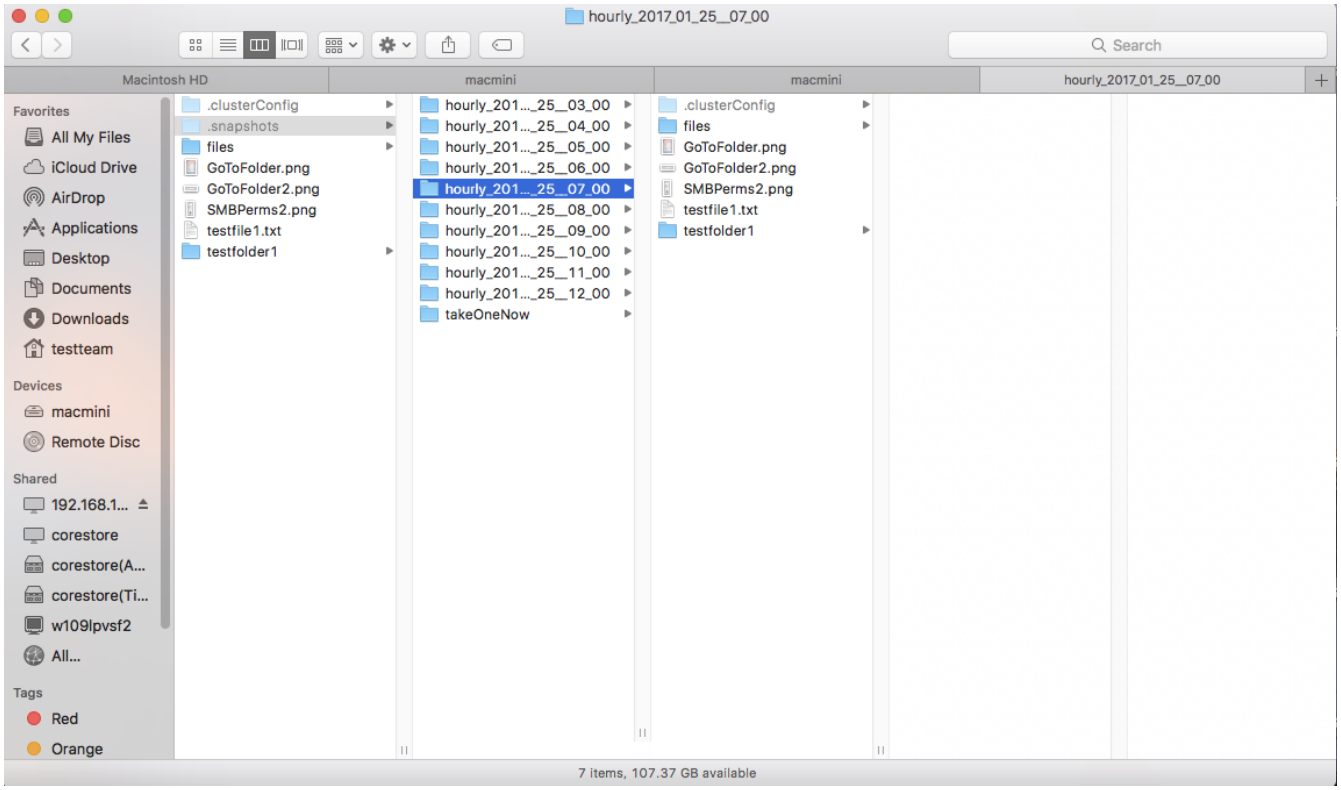
Next, the user must utilize the Finder “Go To Folder” command, which is triggered by Shift+Command+G. It can also be found under to “Go” drop down menu at the top of the screen.



In the “Go to Folder…” dialog, enter the folder using the syntax seen below. It will always be /Volumes/<value_from_GetInfo>/.snapshots. The “<value_from_GetInfo>” is what was read from the “Name & Extension” step.

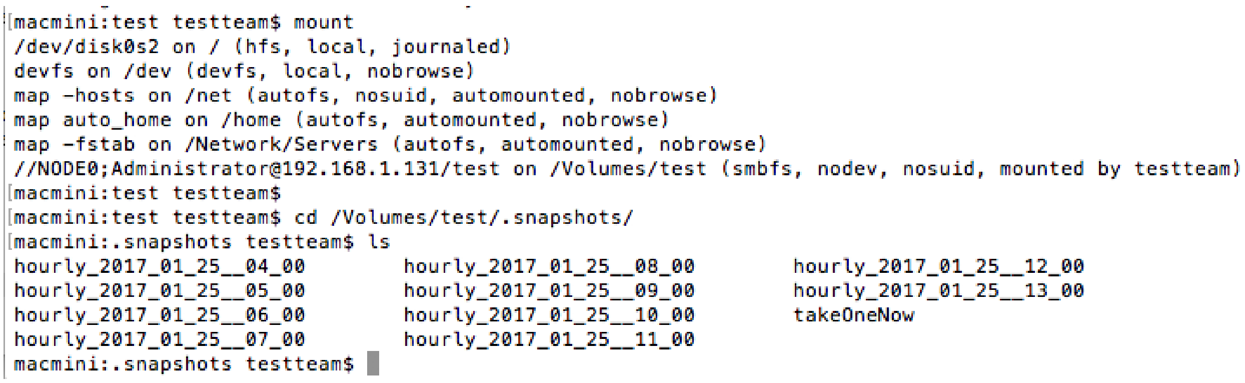

Go to the folder: /Volumes/test/.snapshots

After clicking “Go”, one should be able to browse the .snapshots folder as is shown in the next screenshot.



Using Terminal

For more technically savvy users, viewing snapshot data can be accomplished using Terminal as well:



```
[macmini:test testteam$ mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
map -fstab on /Network/Servers (autofs, automounted, nobrowse)
//NODE0;Administrator@192.168.1.131/test on /Volumes/test (smbfs, nodev, nosuid, mounted by testteam)
[macmini:test testteam$
[macmini:test testteam$ cd /Volumes/test/.snapshots/
[macmini:.snapshots testteam$ ls
hourly_2017_01_25__04_00        hourly_2017_01_25__08_00        hourly_2017_01_25__12_00
hourly_2017_01_25__05_00        hourly_2017_01_25__09_00        hourly_2017_01_25__13_00
hourly_2017_01_25__06_00        hourly_2017_01_25__10_00        takeOneNow
hourly_2017_01_25__07_00        hourly_2017_01_25__11_00
macmini:.snapshots testteam$ 
```

Using Applescript

Additionally, an AppleScript can be created that will take the user directly to the .snapshots folder. This AppleScript is detailed in section 4.5

3.5 SMB Change Notifications

MacOS has a utility called Finder which is used for file browsing. Finder relies upon Change Notifications from SMB shares to trigger it to update (I.E. refresh) its current view on a folder. For example, let's say User #1 is viewing a folder on an SMB share, and User #2 writes a new file or changes an existing file. With Change Notifications enabled, FluidFS will notify User #1 that a certain file or folder has changed, which causes Finder on User #1's system to refresh its view automatically.

Note: The SMB Change Notifications feature is available starting in FluidFS v6

The Advanced Change Notification feature on FluidFS is disabled by default. A very basic set of Change Notifications are always enabled on FluidFS, but the Advanced Change Notifications should be enabled for the best behavior with Finder. It is a setting that can be set on a NAS Volume via the CLI, Dell Storage Manager, or any of the API's (such as PowerShell or REST).

The Dell EMC recommendation for SMB shares which will be accessed by macOS clients is to enable Advanced Change Notifications on the SMB shares. Only the "Advanced Change Notify" is necessary for macOS; there is no need to enable the "Watch Tree", also known as Recursive, or "Notify on File Changes in Whole Directory Tree". The "Watch Tree" functionality may degrade performance, and since it provides no added benefit for macOS, unless its needed by another application (such as IIS), Dell EMC recommends to leave "Watch Tree" disabled.

For customers who are not running FluidFS v6 or later, or cannot implement the SMB Change Notifications feature for any reason, it is possible to create an AppleScript to make Finder refresh a directory. The Finder utility does not include a built in "Refresh" button. To compensate for this, we can make our own. First an AppleScript will be created, as follows:

```
tell application "Finder" to tell front window to update every item
```

Go to File -> Save … and save it as an Application, so the extension will be ".app". Hold the COMMAND key, and drag this .app file onto the toolbar in Finder, and whenever it is clicked, it will refresh the current view.

3.6 SMB Client Configuration

The SMB Protocol includes a feature called SMB Signing which uses cryptography to digitally sign SMB communications to prevent against man in the middle attacks. Apple started enabling SMB signing by default in macOS 10.11.5, which can provide some additional security benefits. However, this added security does come with a bit of a performance penalty. SMB signing can be disabled on macOS to provide a performance benefit.

This is accomplished by adding the following line to the /etc/nsmb.conf file:

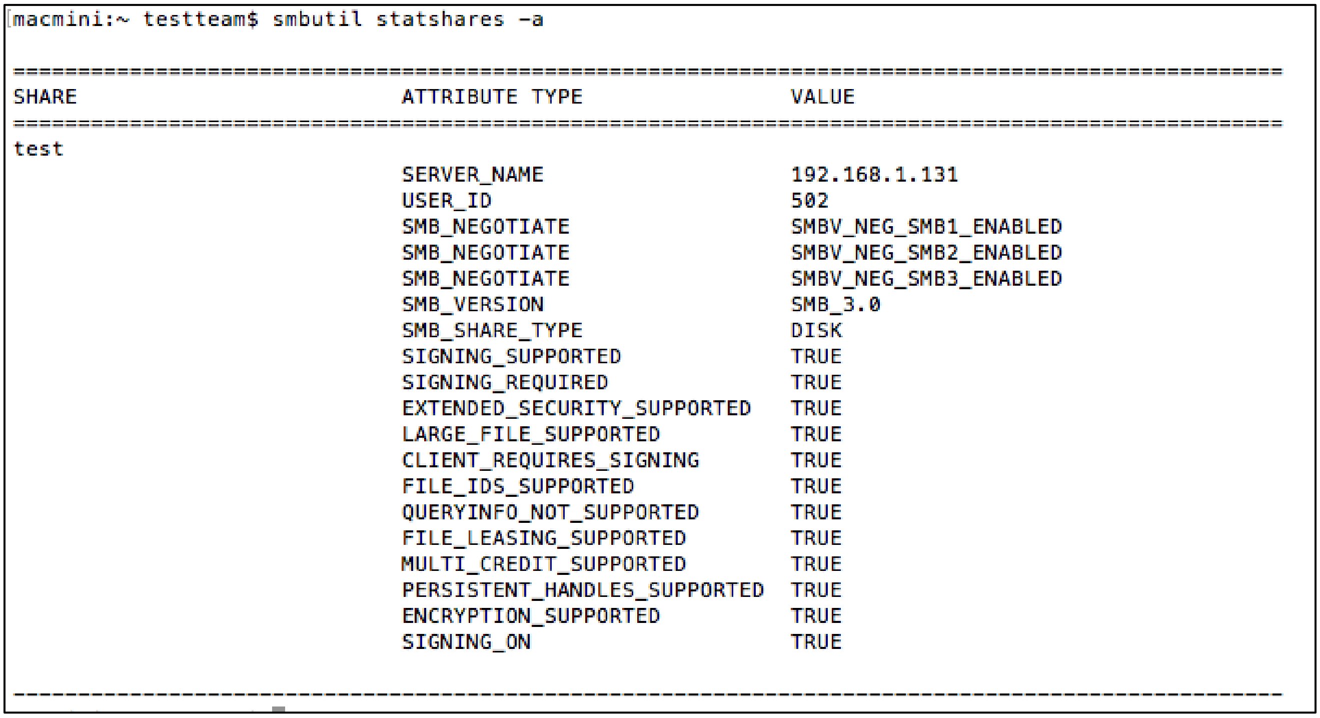
```
[default]

signing_required=no
```

After making this addition/change to the /etc/nsmb.conf file, restart the macOS client, and SMB signing should be disabled. This can be checked on the macOS client by entering the following command at the Terminal:

```
smbutil statshares -a
```

Running this command will show a variety of information about the SMB session, such as SMB_VERSION, and SIGNING_ON. The best performance will be achieved when using a higher SMB version (like SMB 3.0) and when SIGNING_ON is FALSE.

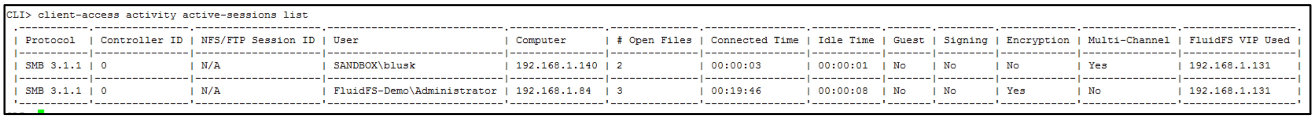


```
macmini:~ testteam$ smbutil statshares -a

==================================================================================================
SHARE                         ATTRIBUTE TYPE                VALUE
==================================================================================================
test
                              SERVER_NAME                   192.168.1.131
                              USER_ID                       502
                              SMB_NEGOTIATE                 SMBV_NEG_SMB1_ENABLED
                              SMB_NEGOTIATE                 SMBV_NEG_SMB2_ENABLED
                              SMB_NEGOTIATE                 SMBV_NEG_SMB3_ENABLED
                              SMB_VERSION                   SMB_3.0
                              SMB_SHARE_TYPE                DISK
                              SIGNING_SUPPORTED             TRUE
                              SIGNING_REQUIRED              TRUE
                              EXTENDED_SECURITY_SUPPORTED   TRUE
                              LARGE_FILE_SUPPORTED          TRUE
                              CLIENT_REQUIRES_SIGNING       TRUE
                              FILE_IDS_SUPPORTED            TRUE
                              QUERYINFO_NOT_SUPPORTED       TRUE
                              FILE_LEASING_SUPPORTED        TRUE
                              MULTI_CREDIT_SUPPORTED        TRUE
                              PERSISTENT_HANDLES_SUPPORTED  TRUE
                              ENCRYPTION_SUPPORTED          TRUE
                              SIGNING_ON                    TRUE

--------------------------------------------------------------------------------------------------
```

This information can be obtained from the FluidFS Command Line Interface as well.



```
CLI> client-access activity active-sessions list
 .-----------.---------------.--------------------.----------------------------.---------------.--------------.----------------.-----------.-------.---------.------------.---------------.------------------.
 | Protocol  | Controller ID | NFS/FTP Session ID | User                       | Computer      | # Open Files | Connected Time | Idle Time | Guest | Signing | Encryption | Multi-Channel | FluidFS VIP Used |
 |-----------|---------------|--------------------|----------------------------|---------------|--------------|----------------|-----------|-------|---------|------------|---------------|------------------|
 | SMB 3.1.1 | 0             | N/A                | SANDBOX\blusk              | 192.168.1.140 | 2            | 00:00:03       | 00:00:01  | No    | No      | No         | Yes           | 192.168.1.131    |
 |-----------|---------------|--------------------|----------------------------|---------------|--------------|----------------|-----------|-------|---------|------------|---------------|------------------|
 | SMB 3.1.1 | 0             | N/A                | FluidFS-Demo\Administrator | 192.168.1.84  | 3            | 00:19:46       | 00:00:08  | No    | No      | Yes        | No            | 192.168.1.131    |
 '-----------'---------------'--------------------'----------------------------'---------------'--------------'----------------'-----------'-------'---------'------------'---------------'------------------'
```

3.7 Time Machine

Starting in Sierra (macOS 10.12), the integrated macOS backup client called “Time Machine” can utilize an SMB share as a place to store the data that it backs up (macOS systems). Previously, the Time Machine software could only use external disks. FluidFS supports using SMB shares hosted on it as a destination for the macOS Time Machine backup data.

Note: In order to see the FluidFS SMB share when clicking “Select Disk…” in Time Machine, the SMB share must first be accessed using Finder → Go → Connect to Server…

3.8 Firewall Settings

For a full list of TCP and UDP ports that FluidFS utilizes, see the *FluidFS Support Matrix*.

The SMB protocol uses only a few ports, which are detailed in the following tables.

SMB Services Provided By FluidFS to macOS SMB Clients

Port Number	Protocol	Function/Service
111	TCP and UDP	RPC Portmapper
445	TCP and UDP	SMB

Services Accessed By FluidFS which SMB may require

Port Number	Protocol	Function/Service
53	TCP	Domain Name Service (DNS)
88	TCP and UDP	Kerberos
111	TCP and UDP	RPC Portmapper
123	UDP	Network Time Protocol (NTP)
135	TCP	Active Directory
138	UDP	NetBIOS datagram service
139	TCP	NetBIOS session service
389 and 636	TCP and UDP	LDAP
464	TCP and UDP	Kerberos
543-544	TCP	Kerberos login and shell
749	TCP and UDP	Kerberos administration
3268-3269	TCP	LDAP Global Catalog

4 FluidFS and macOS – NFS Best Practices

4.1 NFS Secure Port settings

FluidFS defaults all NFS exports to require the usage of a secure TCP port (low number) for mounting. This is because traditionally speaking, restricting NFS to privileged ports (lower than 1023) is considered a security measure to ensure the NFS client is approved by the administrator/OS owner. This is due to the fact a program can only use a privileged port if it's run by the root user (or sudo). Of course, this is only a basic start when it comes to securing an enterprise environment, and it may or may not be required by every customer.

MacOS, as well as some other OS'es, default to *not* using secure TCP ports when mounting NFS exports. Meaning, they try to use TCP ports greater than 1023 to mount. This results in clashing default settings between the macOS NFS client (to use ports > 1023) , and the FluidFS NFS server (to require ports < 1023).

There are two ways around this problem:

1. Disable the "Require Secure Port" option on the NFS export hosted on FluidFS. This is accomplished via the FluidFS Command Line Interface, or the appropriate graphical interface such as Dell Storage Manager.
2. Configure the macOS client to use a reserved port using the "resvport" mount option. This can be done in /etc/nfs.conf, specified in Disk Utility, or specified in terminal in the mount command.

4.2 How to access via NFS using Finder

The most common and easiest way to map an NFS export using Finder is by using the "Go" menu at the top of the screen, and select "Connect to Server". The syntax for the "Server Address" field is as follows:

To mount an NFS export, use this syntax:

```
nfs://<VIP>:/<nas_volume_name>/<exported_path_on_volume>
```

For example:

```
nfs://192.168.1.131:/test/exported_folder_on_volume/
```

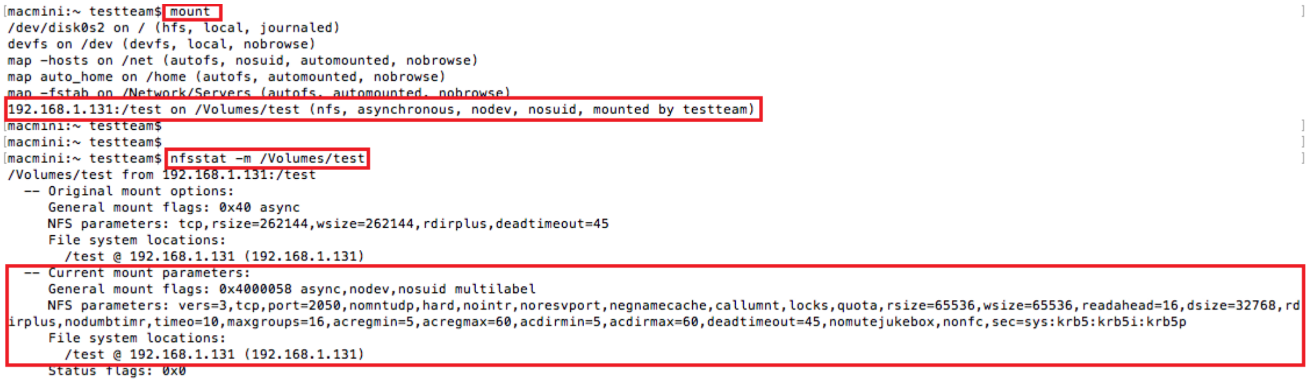
After mounting, one can check the status of the NFS mount from the Terminal by issuing the following 2 commands:

```
mount
```

and

```
nfsstat –m <mount_point>
```

The expected output of these two commands is shown in this screenshot:



```
macmini:~ testteam$ mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
map -fstab on /Network/Servers (autofs, automounted, nobrowse)
192.168.1.131:/test on /Volumes/test (nfs, asynchronous, nodev, nosuid, mounted by testteam)
macmini:~ testteam$
macmini:~ testteam$
macmini:~ testteam$ nfsstat -m /Volumes/test
/Volumes/test from 192.168.1.131:/test
  -- Original mount options:
     General mount flags: 0x40 async
     NFS parameters: tcp,rsize=262144,wsize=262144,rdirplus,deadtimeout=45
     File system locations:
       /test @ 192.168.1.131 (192.168.1.131)
  -- Current mount parameters:
     General mount flags: 0x4000058 async,nodev,nosuid multilabel
     NFS parameters: vers=3,tcp,port=2050,nomntudp,hard,nointr,noresvport,negnamecache,callumnt,locks,quota,rsize=65536,wsize=65536,readahead=16,dsize=32768,rd
irplus,nodumbtimr,timeo=10,maxgroups=16,acregmin=5,acregmax=60,acdirmin=5,acdirmax=60,deadtimeout=45,nomutejukebox,nonfc,sec=sys:krb5:krb5i:krb5p
     File system locations:
       /test @ 192.168.1.131 (192.168.1.131)
     Status flags: 0x0
```

MacOS also has the ability to keep this share mounted forever, even after a reboot. To keep the share mounted, follow these steps:

1. Mount the share in Finder
2. In System Preferences, go to Users & Groups
3. Select the relevant user account and select Login Items
4. Drag the icon of the volume that was mounted, from the desktop, to the list of login items
5. Close System Preferences

Mounting using NFS 4

To access FluidFS using NFS version 4 from macOS clients, the best way is to edit the /etc/nfs.conf file on macOS, and add “vers=4” to the nfs.client.mount.options. After this change is made unmount and mount the NFS exports, and when they are mounted back, NFS version 4 should be in use. This can be verified using “nfsstat –m”.

There is not any performance benefit to using NFS version 4. In fact, NFS version 4 is slower than NFS version 3 for most workloads. NFS version 4 has security enhancements when compared with NFS version 3, and also it is “firewall friendly” in that it only uses TCP port 2049.

NOTE: At the time of writing this document, some compatibility issues exist between macOS 10.12 and FluidFS v6 when using NFS4. Dell EMC highly recommends the NFS3 protocol over the NFS4 protocol at this time.

4.3 NFS Authentication

When a user accesses any NAS Storage device using NFS, the NFS server (FluidFS) allows or denies access by comparing the UID (user identifier) and GID (group identifier) of the accessing user against the UNIX/POSIX permissions of the file or folder they are attempting to access. However, from the user’s perspective, they are providing a user name. In the background, this username is translated into a UID (and the appropriate group membership with GID’s). Once the username is translated into UID and the relevant GID’s for group membership, the NFS server will compare that to the files permissions, and grant or deny access.

The NFS server must have a repository to do lookups in, to translate the user names into UID’s and their relevant group membership. For NFS users, this repository is either LDAP or NIS.

FluidFS and macOS both support LDAP and NIS. The administrator can bind an macOS client to LDAP or NIS using the "Directory Utility" in macOS. FluidFS and macOS both support Microsoft Active Directory-based LDAP (RFC2307) as well.

Note: FluidFS does not support Open Directory at the time of writing this document.

If using LDAP to provide UID's and GID's, Dell EMC recommends using Microsoft Active Directory-based LDAP (RFC2307). FluidFS does not support Open Directory, and any other LDAP servers (such as OpenLDAP) can be difficult to configure the LDAP client for macOS on. It is possible to bind FluidFS and macOS to almost any popular industry LDAP server, but that is outside the scope of this document.

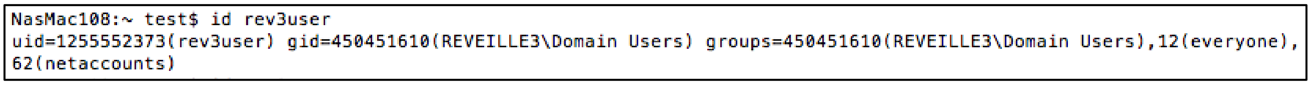
Network Information Service (NIS) is also another good option as a repository for UNIX user information (UID, GID, etc…) and FluidFS and macOS are both fully compatible with NIS.

As a side note, macOS is a UNIX-based operating system. It natively uses UID's and GID's to identify users and groups (unlike Windows, which natively uses SID's to identify users and groups).

<u>Configuring macOS to properly use Active Directory-based LDAP (RFC2307)</u>

If Active Directory-based LDAP (RFC2307) is in use, the macOS client must be configured to read the correct values from Active Directory (AD) for uidNumber and gidNumber. If the macOS client is not properly configured, the values returned from AD for UID and GID will not be correct, which will result in files and folders being written to FluidFS with the wrong UID and GID. The literal values for UID and GID will be very large numbers (1 billion or greater) which is out of the typical range for a UID or GID (typically they are some number >=0 but <= 65,534.

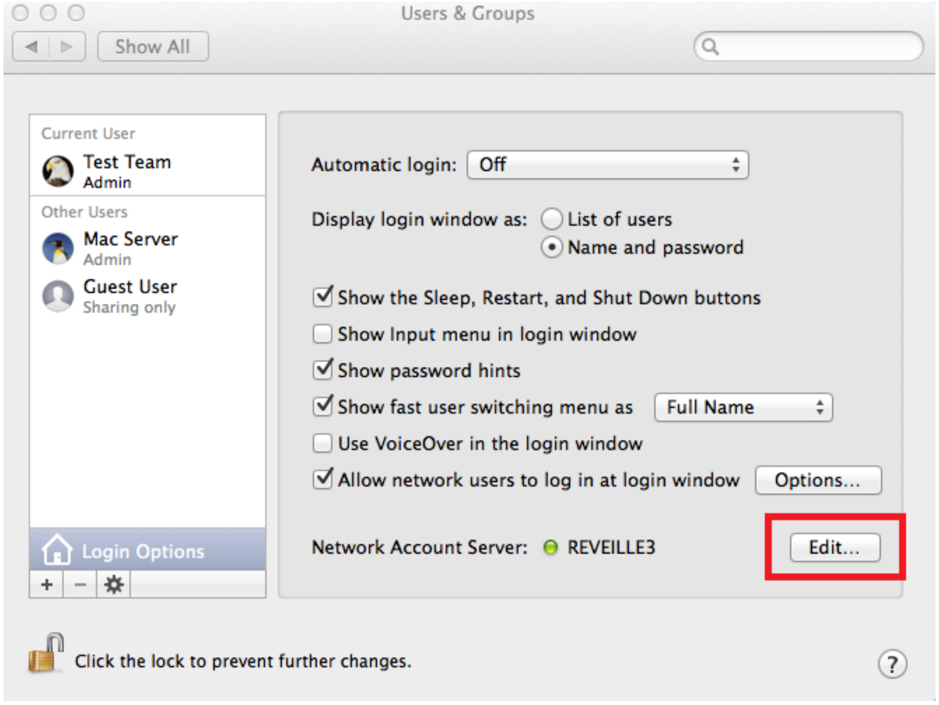
If a macOS client is not properly configured with AD-based LDAP, running "id" at the terminal will show a very large UID and GID, like what is shown in this screenshot:



```
NasMac108:~ test$ id rev3user
uid=1255552373(rev3user) gid=450451610(REVEILLE3\Domain Users) groups=450451610(REVEILLE3\Domain Users),12(everyone),
62(netaccounts)
```

In order to properly configure the macOS client, follow these steps:

1. Open Directory Utility in macOS



Automatic login: Off

Display login window as: ○ List of users ● Name and password

☑ Show the Sleep, Restart, and Shut Down buttons

□ Show Input menu in login window

☑ Show password hints

☑ Show fast user switching menu as Full Name

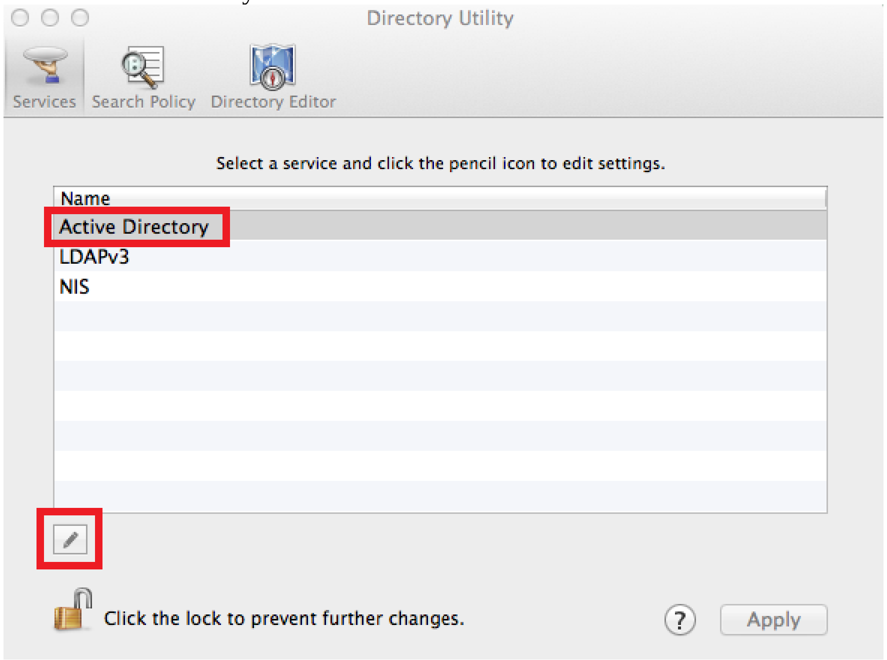
□ Use VoiceOver in the login window

☑ Allow network users to log in at login window

Network Account Server: ● REVEILLE3

Click the lock to prevent further changes.

2. Select Active Directory and click the little Pen icon

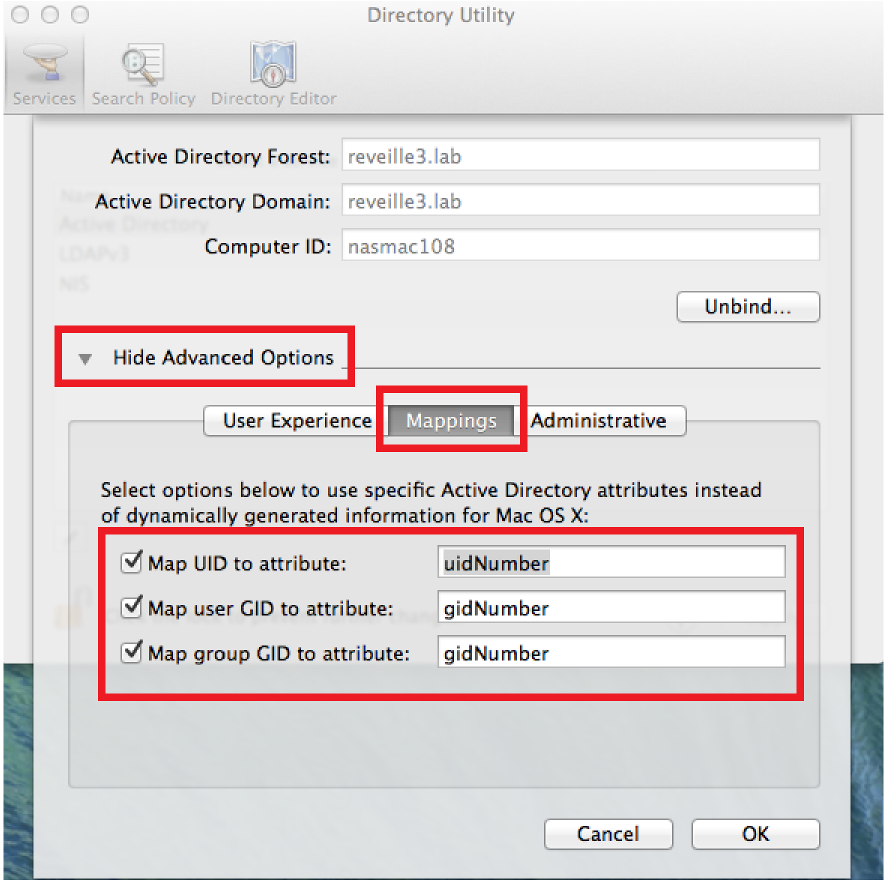


Select a service and click the pencil icon to edit settings.

Name
Active Directory
LDAPv3
NIS

Click the lock to prevent further changes.

3. Click “Show Advanced Options”, then the “Mappings” tab



Active Directory Forest: reveille3.lab

Active Directory Domain: reveille3.lab

Computer ID: nasmac108

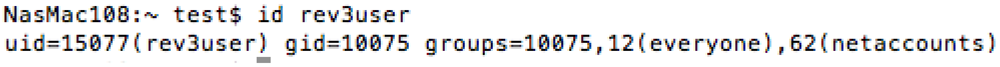
Select options below to use specific Active Directory attributes instead of dynamically generated information for Mac OS X:

☑ Map UID to attribute: uidNumber

☑ Map user GID to attribute: gidNumber

☑ Map group GID to attribute: gidNumber

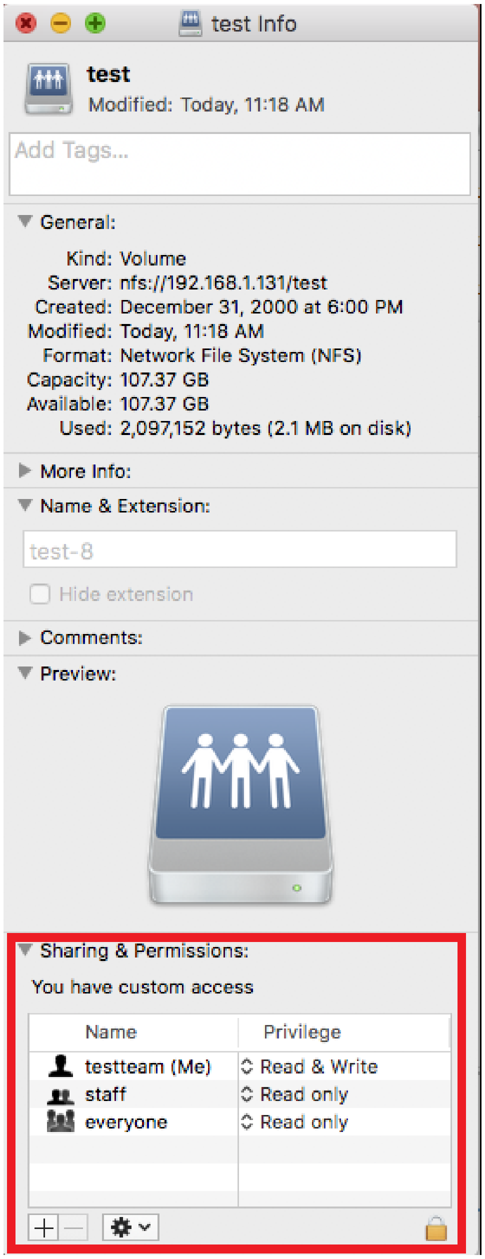
4. Select all 3 options and enter the values show in the screenshot above.
5. After doing this Click OK
6. Go back to the OSX Terminal and enter “id <username>” again. Instead of getting the output seen before as detailed in the “Symptoms” section above, the output should look like this. This is the correct output of the UID’s and GID’s:



```
NasMac108:~ test$ id rev3user
uid=15077(rev3user) gid=10075 groups=10075,12(everyone),62(netaccounts)
```

4.4 Permissions for macOS NFS Clients

MacOS supports POSIX-based permissions, which is what FluidFS defaults to on UNIX style NAS volumes. Typically, a UNIX style NAS volume is what would be used to host an NFS export. Standard UNIX tools such as chmod and chown function from the macOS Terminal, and Finder can be used to view and modify permissions in the Get Info dialog.



test Info

test
Modified: Today, 11:18 AM

Add Tags...

▼ General:

Kind: Volume
Server: nfs://192.168.1.131/test
Created: December 31, 2000 at 6:00 PM
Modified: Today, 11:18 AM
Format: Network File System (NFS)
Capacity: 107.37 GB
Available: 107.37 GB
Used: 2,097,152 bytes (2.1 MB on disk)

▶ More Info:

▼ Name & Extension:

test-8

□ Hide extension

▶ Comments:

▼ Preview:

▼ Sharing & Permissions:

You have custom access

Name	Privilege
testteam (Me)	Read & Write
staff	Read only
everyone	Read only

The user name and group name will only be displayed if the UID and GID map to some UID and GID the macOS client is aware of. If macOS is not able to map the UID and GID to an actual name, it will usually display it as “_unknown”.

Note: FluidFS supports NFSv4 Access Control Lists (ACL's) but macOS does not.

4.5 Accessing Snapshots via NFS

Accessing snapshots from macOS clients over the NFS protocol is less problematic than with the SMB protocol. This is because macOS natively supports POSIX permissions, so the permissions are much more likely to be preserved when a file is copied out of a snapshot than when performing the same operation using the SMB protocol (using a macOS client).

Using Finder

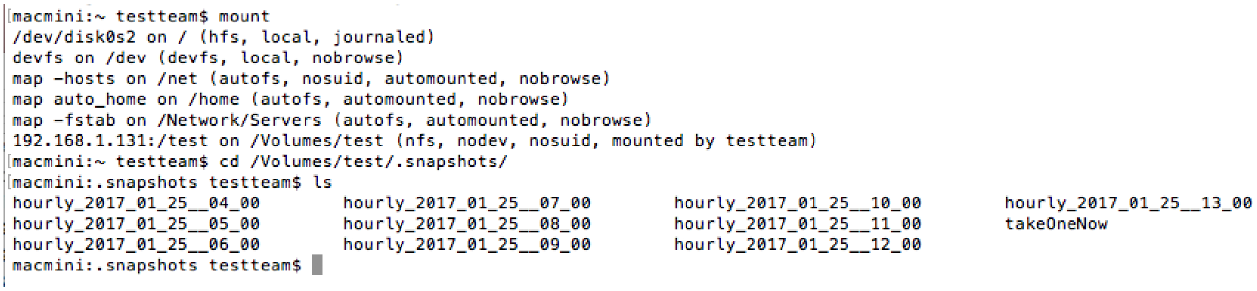
Accessing snapshot data over NFS is accomplished very similarly as is described for SMB clients in Section 3.4. The only exception is: since the data access method is using the NFS protocol, the syntax to be used when in the "Connect to Server" dialog would be:

```
nfs://<VIP>:/<nas_volume_name>/<exported_path_on_volume>
```

Then once the export is mounted, click "Go To Folder" and access the .snapshots directory on /Volumes/<mount_point>/.snapshots.

Using Terminal

As with SMB, accessing snapshot data can also be accomplished from the Terminal:



```
macmini:~ testteam$ mount
/dev/disk0s2 on / (hfs, local, journaled)
devfs on /dev (devfs, local, nobrowse)
map -hosts on /net (autofs, nosuid, automounted, nobrowse)
map auto_home on /home (autofs, automounted, nobrowse)
map -fstab on /Network/Servers (autofs, automounted, nobrowse)
192.168.1.131:/test on /Volumes/test (nfs, nodev, nosuid, mounted by testteam)
macmini:~ testteam$ cd /Volumes/test/.snapshots/
macmini:.snapshots testteam$ ls
hourly_2017_01_25__04_00        hourly_2017_01_25__07_00        hourly_2017_01_25__10_00        hourly_2017_01_25__13_00
hourly_2017_01_25__05_00        hourly_2017_01_25__08_00        hourly_2017_01_25__11_00        takeOneNow
hourly_2017_01_25__06_00        hourly_2017_01_25__09_00        hourly_2017_01_25__12_00
macmini:.snapshots testteam$ 
```

Using Applescript

Accessing snapshots via NFS on macOS also gives us the ability to use an AppleScript to provide a direct link to the .snapshots folder. Use the "Script Editor" app, and write a script of this form. The path in the "open" line will need to be replaced to match the mount point for the NFS export.

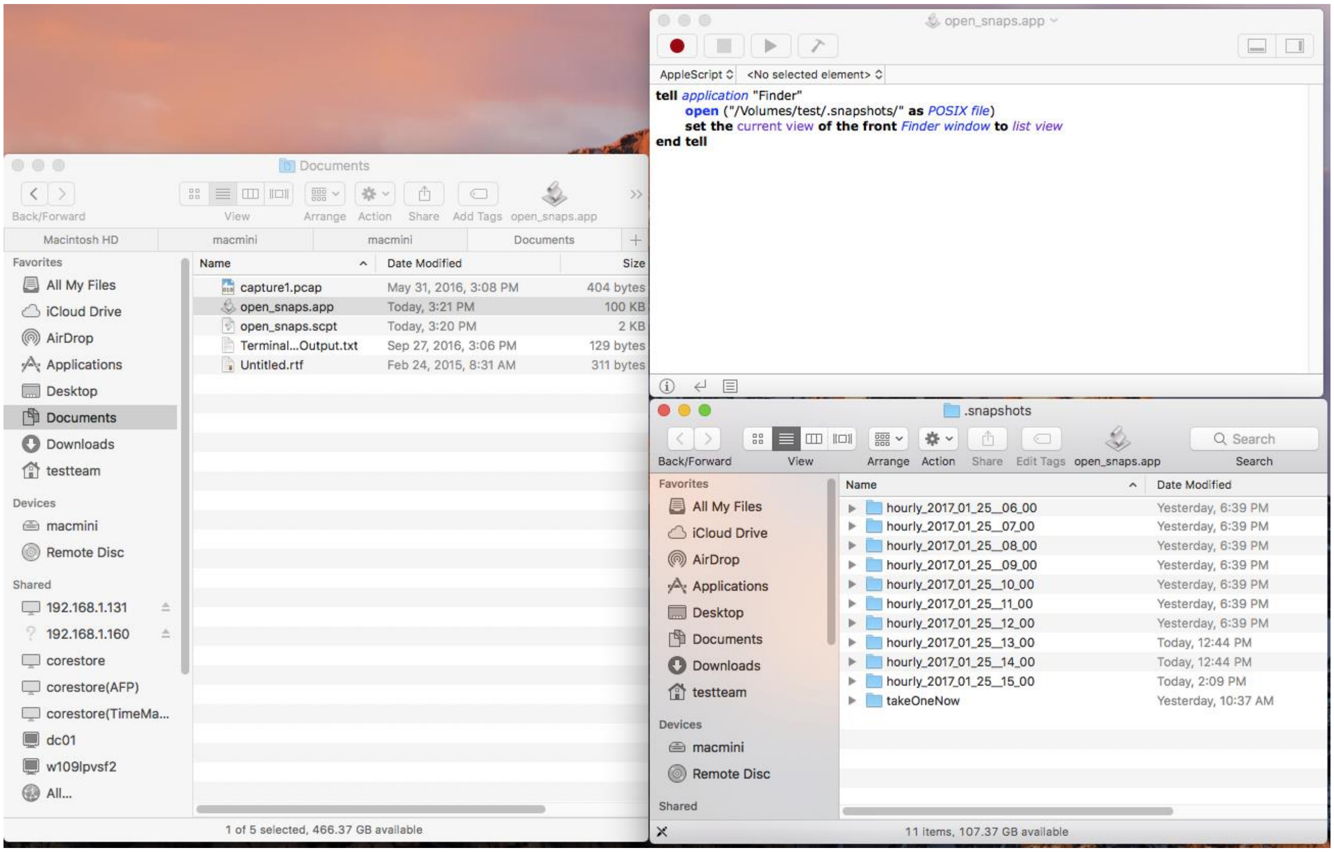
```
tell application "Finder"

    open ("/Volumes/test/.snapshots/" as POSIX file)

    set the current view of the front Finder window to list view

end tell
```

Click File -> Save … and in the “File Format” drop down, select “Application”. Now there is something which can be double clicked to open .snapshots. One can also hold down the COMMAND key, and drag this .app file onto the Finder toolbar to create a shortcut for it.



```
tell application "Finder"
    open ("/Volumes/test/.snapshots/" as POSIX file)
    set the current view of the front Finder window to list view
end tell
```

Name	Date Modified	Size
capture1.pcap	May 31, 2016, 3:08 PM	404 bytes
open_snaps.app	Today, 3:21 PM	100 KB
open_snaps.scpt	Today, 3:20 PM	2 KB
Terminal...Output.txt	Sep 27, 2016, 3:06 PM	129 bytes
Untitled.rtf	Feb 24, 2015, 8:31 AM	311 bytes

Name	Date Modified
hourly_2017_01_25__06_00	Yesterday, 6:39 PM
hourly_2017_01_25__07_00	Yesterday, 6:39 PM
hourly_2017_01_25__08_00	Yesterday, 6:39 PM
hourly_2017_01_25__09_00	Yesterday, 6:39 PM
hourly_2017_01_25__10_00	Yesterday, 6:39 PM
hourly_2017_01_25__11_00	Yesterday, 6:39 PM
hourly_2017_01_25__12_00	Yesterday, 6:39 PM
hourly_2017_01_25__13_00	Today, 12:44 PM
hourly_2017_01_25__14_00	Today, 12:44 PM
hourly_2017_01_25__15_00	Today, 2:09 PM
takeOneNow	Yesterday, 10:37 AM

4.6 NFS Client Configuration

The NFS settings on a macOS client are in the */etc/nfs.conf* file. MacOS describes all available mount options in the *man_nfs* man page. There is a separate man page for nfs.conf.

Dell EMC recommends the following mount options for NFS. The following lines should be added to /etc/nfs.conf on the macOS client.

```
nfs.client.mount.options=tcp,rw,async,rdirplus,rwsize=65536

nfs.client.allow_async=1
```

This provides

- (tcp) - NFS3 over TCP
- (rw) - Read/write access

- (async) - Writes to the FluidFS NAS can be acknowledged without that write actually having been flushed to persistent disk. This is OK since FluidFS cache is mirrored.
- (rdirplus) - More efficient metadata retrieval using READDIRPLUS. A nice metadata performance benefit can be achieved with READDIRPLUS. This call combines the LOOKUP, ACCESS, GETATTR, and READDIR calls into one single call. The result of this can be Finder displaying directory listings faster.
- (rwsize=65536) - Data is read and written to FluidFS in 64k request sizes (the FluidFS default for NFS 3). Note that the maximum supported by FluidFS is 256k, but the maximum supported by macOS is 64k.

NFS Locking

The "nolock" mount option can be added if the mount is private or if advisory locks are not in use. Locking on NFS 3 is advisory only, meaning the applications must be written/configured to honor advisory locks. Using "nolock" can give some performance benefits in Finder as well. When the Finder gets a directory listing, it will get the initial list of files and folder from FluidFS, and then it attempts to get icon previews for the objects in the folder (unless this has been disabled). As the Finder works to get icon previews, it tries to take an exclusive lock on the object, read it to generate the preview, release the lock, and then do the same with the next object in the folder. Sometimes, a deadlock can occur in the Finder if other clients are manipulating files in the same folder, and have locks. The deadlock comes from the Finder attempting to get a lock, and FluidFS denies it because another user has a lock already on that file. The Finder will stop going through the folder listing until the other user who holds the lock on that one object releases the lock. For these reasons, it could be advantageous to disable locking. In addition to avoiding the deadlock situation, disabling file clocking reduces the total amount of network calls as well to FluidFS, which will help Finder list folder contents faster.

Reserved Port/Secure Port

Optionally, as mentioned in section 4.1, the "resvport" mount option could be added here. This forces the macOS NFS client to use ports 1023 and lower, which would make it compliant with the FluidFS default NFS export setting to require TCP ports of 1023 or lower.

NFS version 4 (NFS4)

As is covered in section 4.2, macOS clients can mount using NFS version 4 by adding "vers=4" to the nfs.client.mount.options in /etc/nfs.conf.

NOTE: At the time of writing this document, some compatibility issues exist between macOS 10.12 and FluidFS v6 when using NFS4. Dell EMC highly recommends the NFS3 protocol over the NFS4 protocol at this time.

4.7 Firewall Settings

For a full list of TCP and UDP ports that FluidFS utilizes, see the *FluidFS Support Matrix*.

NFS version 4 is considered "Firewall Friendly" because it utilizes only one TCP port – 2049. NFS version 3 however, uses more ports, as is detailed in the following table:

NFS Services Provided By FluidFS to macOS NFS Clients

Port Number	Protocol	Function/Service
111	TCP and UDP	RPC Portmapper
2049-2057	TCP and UDP	NFS daemon
4000-4007	TCP and UDP	NFS stat daemon
4050-4057	TCP and UDP	Network Lock Manager (NLM)
5001-5008	TCP and UDP	NFS mount daemon
5051-5058	TCP and UDP	NFS quota daemon

Services Accessed By FluidFS which NFS may require

Port Number	Protocol	Function/Service
88	TCP and UDP	Kerberos (only if using Kerberos)
111	TCP and UDP	RPC Portmapper
389 or 636	TCP and UDP	LDAP / Secure LDAP (over TLS) – (only if using LDAP)
464	TCP and UDP	Kerberos (only if using Kerberos)
543-544	TCP	Kerberos login and shell (only if using Kerberos)
RPC/portmapper dependent	UDP	Network Information Service (NIS) – (only if using NIS)
749	TCP and UDP	Kerberos administration (only if using Kerberos)
3268-3269	TCP	LDAP Global Catalog – (only if using LDAP)

A Appendix: Summary of Best Practices per protocol

Below is a summary of best practices, per protocol, taken from sections of this document.

SMB

- Enable Change Notifications
- Disable SMB signing for better performance
- Use List View in Finder
- Disable “Calculate all sizes” for Finder
- Disable “Show Icon Preview” for Finder
- Configure permissions on SMB shares to use inherited ACL’s

NFS

- Disable “Require Secure Port” on FluidFS, or set the macOS client to use resvport using /etc/nfs.conf mount option “resvport”
- If using NFS3, be sure all of the necessary firewall ports are open
- Configure the /etc/nfs.conf with the proper mount options. Readdirplus will provide performance benefits for finder.
- Set correct permissions on the NFS exports

B Appendix: Additional resources

All FluidFS Technical Collateral on Dell TechCenter