



**Copy or Restore Data with fs\_ddcopy**

## **Copyright**

© 2020 Panzura, Inc.

All rights reserved. The information in this manual is subject to change without notice.

Originated in the USA. Any other trademarks appearing in this guide are the property of their respective companies.

Updated: Tuesday, October 13, 2020

## **Contact Us**

695 Campbell Technology Parkway

Suite 225

Campbell, CA 95008

[support@panzura.com](mailto:support@panzura.com)

1-855-PANZURA (1-855-726-9872)

+1 (408) 578-8888

[www.panzura.com](http://www.panzura.com)

## Table of Contents

<b>Introduction</b> .....	<b>4</b>
<b>Pre-requisites</b> .....	<b>4</b>
<b>Best Practices</b> .....	<b>4</b>
<b>Supported Arguments</b> .....	<b>5</b>
<b>Examples</b> .....	<b>6</b>
Example 1: Basic Operation.....	6
Example 2: Overwrite.....	7
<b>Summary</b> .....	<b>8</b>

# Introduction

User or project directories can be restored by using `fs_ddcopy`, a recursive file system copy command that runs on the filer. This method does not copy the restored files from the cloud into the local filer's cache. Instead, `fs_ddcopy` restores the metadata for the files.

For Windows clients, Windows Previous Version (WPV) provides the same result as `fs_ddcopy`, and is managed by individual users, so that the administrator does not need to be involved in most small user restorations. However, WPV is single-threaded and performance issues can arise when copying many files or large files. In contrast, since `fs_ddcopy` deals only with the restored file metadata instead of copying the files themselves, it is a much faster operation. Using `fs_ddcopy` can be 5-10 times faster on a single large file and can be another 10 times faster with 10 threads running over a large number of files.

## Pre-requisites

Panzura filers must be running PZOS version 7.1.9.3.15511 or newer. Destination filer's snapshots should be in sync (specifically with the source filer).

Currently, `fs_ddcopy` operations should only be initiated by Panzura Support via the filer support shell.

For large copy operations and operations where snapshots are involved, take care to turn off snapshots prior to running `fs_ddcopy`. Navigate to WebUI > Configuration > Snapshot Settings > Filer Snapshot Settings and turn "Enable Scheduled Snapshots" slider to the OFF position. After `fs_ddcopy` operations are complete, restore the original "Enable Scheduled Snapshots" state.

## Best Practices

- `fs_ddcopy` should always be run from the destination filer and, preferably, to an empty directory.
- The source and destination path **MUST** be expressed in the POSIX format (for example: `"/cloudfs/filer/dir/"`) instead of the UNC format (for example: `"\\filer\cloudfs\filer\dir\"`).

- If the source and destination path have spaces or special characters, then surround the parameters with quotes:

```
fs_ddcopy -t 10 -S "/cloudfs/server_nyc/.snapshot/monthly.2/users/wayne mayson/" -D "/cloudfs/server_nyc/users/wayne mayson new/"
```

- Increasing the number of threads can help with the performance of the fs\_ddcopy process. The following command uses 10 threads to copy from the source to destination.

```
fs_ddcopy -t 10 -S /cloudfs/<srcfiler>/<srcpath> -D /cloudfs/<dstfiler>/<dstpath>
```

- Panzura recommends that large scale copies using fs\_ddcopy be performed in low demand time periods. Contents should be copied to an empty or new subdirectory within the share created by an Administrator user, this ensures the right to access the new directory after the copy is complete.
- If user snapshot schedule was turned off prior to using fs\_ddcopy, make sure to restore the original “Enable Scheduled Snapshots” state after fs\_ddcopy operations are complete,

## Supported Arguments

The following options are supported for fs\_ddcopy.

```
Usage: fs_ddcopy [-s <nNsStT>] [-NL] [-g fileglob] [-S srcpath] [-D dstpath]
[-t num_threads] [-T timeout_secs] [-R retries] [-fhvV] [-d serialize_fmt] [-m log_mask]
[-o option_file]
```

- g, Include all files that match glob.
- D, Write files to this path.
- S, Read files from this path.
- N, No work performed.
- L, list attrs/xattrs
- r, Do not recursively walk the file system
- s, Sort by:
  - n - name
  - N - name reversed
  - s - size
  - S - size reversed
  - t - time
  - T - time reversed

-t, For recursive walks, use multiple threads  
-T, Set individual timeout (Default=0 is no timeout).  
-R, Set retries per attempt (Default=2 is two retries).  
-d, Dump internal state of the system on completion.  
decimal value or:  
Format: J - JSON. X - XML. B - BEF. T - Text. C - CSV.  
Quantity: r - recurse, a - about (more detail), p - Pretty, d - Debug  
Examples: Jra - JSON, recurse, about

X - XML, top level only

-f, force overwrite of all existing files (Starting in Freedom v8.0, default behavior does not overwrite existing files.)  
-h, Display this help summary.  
-n, Print information on what would have been done, but do not execute.  
-m, Set logging mask.  
-o, Take options from file [option\_file].  
-v, Enable verbosity. Add more 'v's to increase verbosity.  
-V, Print version information.

## Examples

### Example 1: Basic Operation

This example shows how to copy to a new location while temporarily maintaining the existing destination. All operations in this example are performed on the filer named “server\_nyc”.

Source:

```
/cloudfs/server_nyc/.snapshot/monthly.2/users/wayne_m/
```

Destination (newly created):

```
/cloudfs/server_nyc/users/wayne_m_new/
```

1. First, an administrator creates an empty directory in the users share called “wayne\_m\_new”.
2. The fs\_ddcopy command is then launched from destination filer “server\_nyc”. In this case, the source and destination filer are the same.

```
fs_ddcopy -t 10 -S /cloudfs/server_nyc/.snapshot/monthly.2/users/wayne_m/  
-D /cloudfs/server_nyc/users/wayne_m_new/
```

3. After the copy is complete, you can rename as follows:

Source: /cloudfs/server\_nyc/users/wayne\_m/

Renamed Source: /cloudfs/server\_nyc/users/wayne\_m\_old/

Destination: /cloudfs/server\_nyc/users/wayne\_m\_new/

Renamed Destination: /cloudfs/server\_nyc/users/wayne\_m/

4. You can now review any content in wayne\_m\_old that is not in the new wayne\_m to verify that the correct content is pulled forward, and then remove the older content.

## Example 2: Overwrite

You can use fs\_ddcopy to directly overwrite a user directory. In this example, the Source and Destination folders are:

Source:

```
/cloudfs/server_nyc/.snapshot/monthly.2/users/wayne_m/
```

Destination (existing):

```
/cloudfs/server_nyc/users/wayne_m/
```

1. The fs\_ddcopy command is then launched from destination filer "server\_nyc". In this case, the source and destination filer are the same. Take note of the '-f' switch, which forces overwrite of existing files:

```
fs_ddcopy -f -t 10 -S  
/cloudfs/server_nyc/.snapshot/monthly.2/users/wayne_m/ -D  
/cloudfs/server_nyc/users/wayne_m/
```

2. The restore operation is additive and behaves the same as a typical Windows file server when overwriting. Any files that are in the destination directory before the operation and not overwritten continue to be in the destination directory.

Note: If there is any chance that a virus has infected any of the files in the destination directory, do not use fs\_ddcopy to overwrite.

## Summary

fs\_ddcopy can be used to copy files & directories between different locations in the CloudFS very quickly. Since data is already in the cloud, and tracked in the CloudFS via metadata, the fs\_ddcopy process can then skip copying the data, and copy only the metadata. Therefore, content is moved much more quickly than conventional methods (for example: rsync, cp, or robocopy).

Through fs\_ddcopy, Panzura Freedom filers produces the same result (a one-to-one match of the data & metadata) as other processes in a much more performant manner.