



“ For over a decade Coolspirit have been supplying the UK’s top organisations with storage products and solutions so be assured we will meet your requirements head on.

It’s all about getting things right first time, quickly and simply! ”

**Damon Robertson**  
Coolspirit Ltd

#### Our address

24 The Bridge Business Centre  
Beresford Way  
Chesterfield  
S41 9FG

#### Get in touch

Call us on: 01246 454222  
Email us: [web@coolspirit.co.uk](mailto:web@coolspirit.co.uk)  
Find us: [View location map](#)  
Web: [www.coolspirit.co.uk](http://www.coolspirit.co.uk)

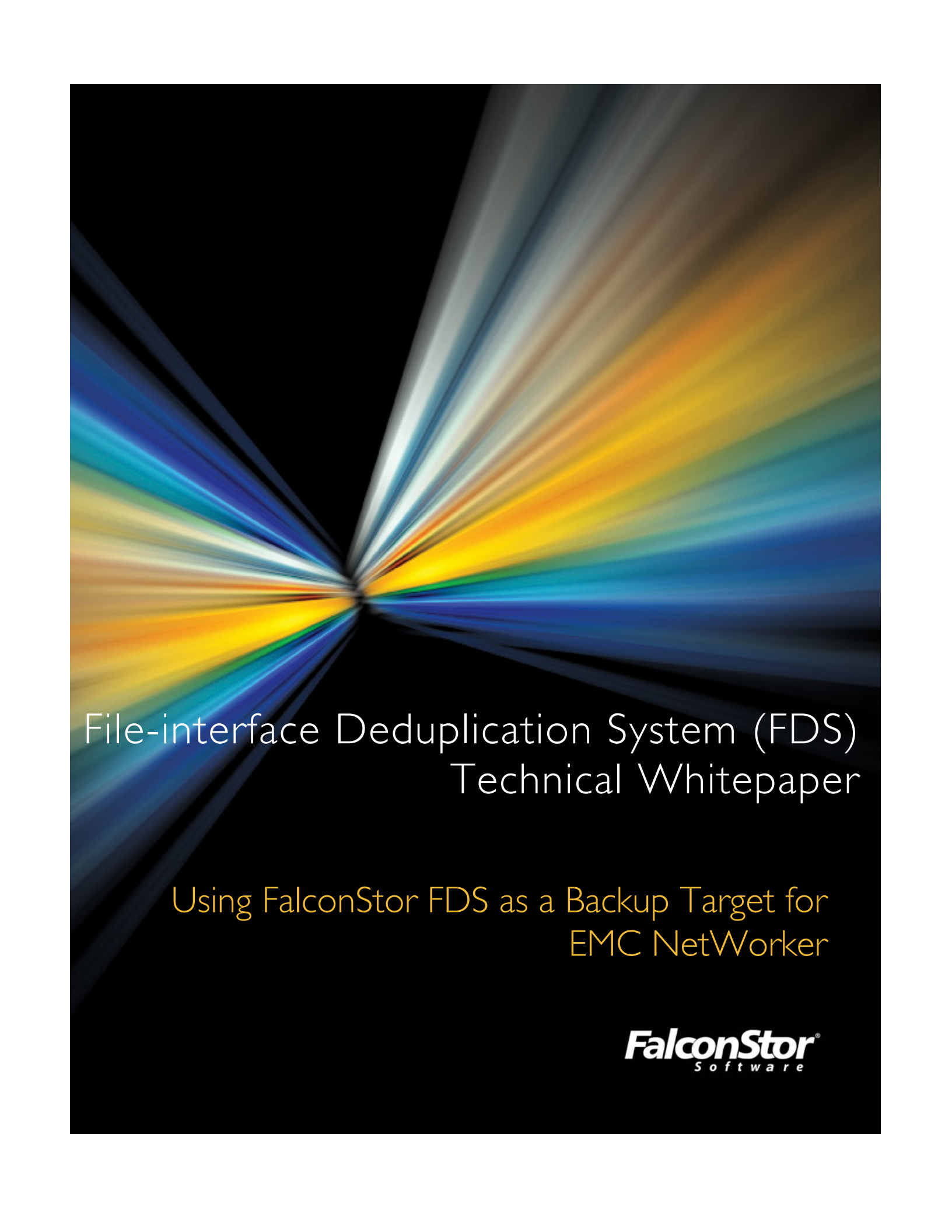
#### Office hours

mon - thurs 8:30am - 5:30pm  
fri 8:30am - 5pm  
sat - sun Closed

“ Boost your storage buying power...  
use ours! ”

Buy with confidence from  
Coolspirit your authorised  
FalconStor Partner

**FalconStor®**  
Software



# File-interface Deduplication System (FDS) Technical Whitepaper

Using FalconStor FDS as a Backup Target for  
EMC NetWorker

**FalconStor**<sup>®</sup>  
Software



# Contents

---

<b>Introduction .....</b>	<b>4</b>
<b>Abstract .....</b>	<b>4</b>
<b>Document Scope.....</b>	<b>4</b>
<b>Audience.....</b>	<b>4</b>
<b>Assumptions .....</b>	<b>4</b>
<b>Overview and Benefits.....</b>	<b>6</b>
<b>FalconStor FDS Benefits.....</b>	<b>7</b>
<b>FalconStor FDS Terminology .....</b>	<b>8</b>
<b>FalconStor FDS Architecture.....</b>	<b>9</b>
<b>EMC NetWorker .....</b>	<b>10</b>
<b>Overview .....</b>	<b>10</b>
<b>Benefits of FalconStor FDS and EMC NetWorker.....</b>	<b>10</b>
<b>EMC NetWorker Terminology .....</b>	<b>10</b>
<b>Typical NetWorker Architecture .....</b>	<b>12</b>
<b>Integrating FalconStor FDS with NetWorker .....</b>	<b>13</b>
<b>Methodology.....</b>	<b>13</b>
<b>FalconStor FDS and NetWorker Architecture .....</b>	<b>14</b>
<b>Configuration Guidelines – FalconStor FDS.....</b>	<b>15</b>
FalconStor FDS Appliance Types .....	15
Sizing.....	15
<b>Configuration Guidelines – EMC NetWorker.....</b>	<b>16</b>
File Type Device.....	16
Advanced File Type Device.....	16
Windows Clients .....	17
NFS Clients .....	18
<b>Conclusion.....</b>	<b>19</b>
<b>Appendix.....</b>	<b>20</b>
<b>Sources.....</b>	<b>20</b>
<b>Reference Documents .....</b>	<b>20</b>
<b>Related Documents .....</b>	<b>20</b>



# Introduction

---

## Abstract

The FalconStor® File-interface Deduplication System (FDS) is a block-level data deduplication tool that provides a space-efficient repository for data from EMC NetWorker. With FalconStor FDS, you can reduce your backup management costs by dramatically reducing your disk storage needs, reducing your dependency on tape, and reducing off-site tape storage costs by enabling you to achieve longer retention periods on disk and facilitating replication of your backups to meet off-site requirements.

## Document Scope

This document describes the basic concepts and integration guidelines for FalconStor FDS in an EMC NetWorker environment. The document is intended to provide an architectural overview of FalconStor FDS being used with EMC NetWorker and the benefits of a combined solution. The information in this document is presented in the form of guidelines. This document is not meant to be a technical Best Practices Guide.

## Audience

The audience for this document includes storage consultants, pre-sales specialists in charge of projects involving backup optimization concepts, and partners interested in FalconStor FDS. This document is especially beneficial for IT directors, storage administrators, backup administrators, data center managers, architects and others involved in the administration of backup architecture including EMC NetWorker. This document can also be valuable to IT staff in charge of disaster recovery (DR) projects.

## Assumptions

We assume that the reader is familiar with:

- EMC NetWorker
- Operating systems
- Network-attached storage and protocols (i.e. NFS, CIFS)
- LAN-based data protection
- Backup challenges
- Deduplication (refer to the FalconStor white paper *Demystifying Data Deduplication: Choosing the Best Solution* for an introduction to deduplication)
- Recovery Point and Recovery Time Objectives (defined below)
- Service Level Agreements and Objectives (defined below)

Term	Definition
<b>Recovery Point Objective (RPO)</b>	The maximum period of time for which a business is willing to accept data loss. For example, nightly backups have an RPO of 24 hours while synchronous replication can have an RPO of zero.
<b>Recovery Time Objective (RTO)</b>	The maximum amount of downtime a business is willing to accept.
<b>Service Level Agreement (SLA)</b>	A contract which records a common understanding about services, priorities, responsibilities, guarantees and warranties. Each area of service scope has the 'level of service' defined. Frequently used to represent the contracted RTO.
<b>Service Level Objective (SLO)</b>	SLOs are a key element of an SLA between a service provider and a customer. SLOs are agreed to as a means of measuring the performance of the service provider and are outlined as a way of avoiding disputes between the two parties based on misunderstanding. SLOs are specific measurable characteristics of the SLA, such as availability, throughput, frequency, response time, or quality.

The information in this document is presented in the form of guidelines.

We also assume that this may be the reader's first exposure to FalconStor FDS so we are including the basics of FalconStor FDS.



# Overview and Benefits

In today's business environment, many customers face increased challenges in protecting their vital data from loss, theft, corruption, and disaster. Traditional backup operations constantly reproduce data for protection and recovery purposes; therefore the amount of data keeps increasing and IT costs keep rising. Even though disk prices are lower each year and tape drive and SAN performance have increased, coping with exponential data growth remains a significant challenge for most organizations.

With the introduction of FalconStor FDS, it is now possible to control data growth resulting from producing multiple copies of the same backup data. FalconStor FDS is a block-level data deduplication tool that provides a space-efficient repository for data from:

- Third-party tape backup software, such as: EMC NetWorker, IBM Tivoli Storage Manager (TSM), Veritas NetBackup, Symantec Backup Exec, CA ARCserve, Arkeia Network Backup, and VMware Consolidated Backup.
- Database backup utilities, such as: Oracle RMAN and SQL-BackTrack.
- Archiving applications, such as: Mimosa™ Systems NearPoint™, Arkivio® auto-stor, CommVault DataArchiver™, FalconStor Capacity-on-Demand™, and Enigma Data Solutions' SmartMove.
- Any other mechanism for delivering data to a network share, such as FalconStor FileSafe™.

With FalconStor FDS, you can reduce your disk storage needs dramatically, allowing you to maintain far more data on disk while incurring minimal additional storage costs. FalconStor FDS can also function as a nearline data repository for project archives, storing older files, etc.

FalconStor FDS supports many-to-one data replication, providing a cost-effective disaster recovery solution. Only deduplicated data is sent over the WAN, providing bandwidth savings. Smaller offices and remote sites can eliminate tape backup entirely using the FDS repository. Data restore is quick and efficient from native-format files rather than from tape-backup formats.

FalconStor FDS uses standard network protocols such as Common Internet File System (CIFS) or Network File System (NFS) to present a simple, network-based file share as the target for backed up data. Connection to FalconStor FDS is a simple matter of mapping to a share, making it compatible with any application that uses an IP network to store data.

Each FDS file share holds incoming data, acting as a "disk" for disk-to-disk (D2D) backup. Based on user policy, deduplication occurs at a scheduled time or on an as-needed basis.

During deduplication, the system analyzes blocks of data and determines whether the data is unique or has already been copied to the FDS repository (virtualized disks that hold deduplicated data). The process then passes only single instances of unique data to the FDS repository and replaces each deduplicated file with a small file (called a *stub* file), whose function is to point to the repository and is used to retrieve stored data.

Even though the user interface is file-based, deduplication is done at the block-level, not at a file level. Block-level deduplication examines small sub-blocks, making it far more effective at reducing storage consumption than file-based deduplication.

Because it uses network-based file shares for backed up data, restoring data is faster and easier with FalconStor FDS. The administrator has direct access to all files without the need for a restore job. Even after deduplication occurs, pointers (*stub* files) on the share point to the full file in the repository. Restoring is as simple as copying the necessary files from a share back to the appropriate location.

## FalconStor FDS Benefits

**Easy deployment:** FalconStor FDS is qualified to seamlessly work with EMC NetWorker by presenting a file interface or a CIFS or NFS network share. This ease of integration allows for a seamless deployment into the existing storage infrastructure and doesn't require any changes to current backup and archiving processes.

**High-performance backup:** FalconStor FDS was built with performance in mind. Its post processing and concurrent block-level deduplication technology is optimized to ingest backup data without affecting backup speed. Its concurrent processing options allow for the deduplication process to take place in the background while its file interface maintains the high performance characteristics needed to meet the backup window.

**Flexible deduplication:** Data deduplication is policy driven; the deduplication processes can be set by the user to start immediately after the backup or can be scheduled to occur at a set time on a regular basis. This flexibility allows the end user to accommodate different operations on non-duplicated data such as data copies, restore operations or other operations such as data mining or database testing.

**High-performance restores:** FalconStor FDS is optimized to enable high-performance data access for both non-duplicated data as well as deduplicated data. This allows for quick backup data restore processes when needed. Data is striped across the deduplication repository to maximize read operation performance. In addition, the data deduplication repository has direct block-level access with no file system overhead resulting in no performance degradation during read operations.

**Flexible, scalable architecture:** FalconStor FDS can scale from a small footprint deployment up to petabytes of logical storage capacity. Its physical managed capacity can scale from 1 TB up to 64 TB of deduplication repository in a single node.

**Multi-site Disaster Recovery:** FalconStor FDS offers global deduplication capability for quick and cost effective disaster recovery deployments. Connecting remote offices via FalconStor FDS appliances allows organizations to eliminate tape shipments between sites and ensures data is readily available online when needed. FalconStor FDS is enabled with intelligent global data replication technology; unique data is only sent once from the remote sites to the main data center. This WAN-optimization method allows for cost effective data replication and significant bandwidth savings; up to 97% reduction in production bandwidth usage.

## FalconStor FDS Terminology

The primary components of the FalconStor FDS solution are the FDS appliance, FDS clients, and the console. These components all sit on the same network segment, the *storage network*. The terminology and concepts used in FalconStor FDS are described here. For additional information, refer to the *FalconStor FDS User Guide*.

Component	Definition
<b>Appliance</b>	An industry-standard server that provides all data deduplication functions. The appliance can function as a standalone appliance with internal storage or it can function as a gateway to storage on an existing network. FDS storage is used to store the original data as well as the unique data blocks and the indexes to the data. The FDS appliance can be attached to physical SCSI and/or Fibre Channel storage devices.
<b>Clients</b>	FDS clients are NetWorker storage nodes that use an FDS share to store data. Storage resources appear to client operating systems (Windows, Linux, Solaris, etc.) as network-attached devices.
<b>Shares</b>	The logical entities presented to FDS clients via the IP network. Clients access FDS shares using either the NFS or CIFS network protocol.
<b>Console</b>	The administrative tool that allows you to create shares, configure deduplication, and monitor resources and deduplication. This Java application can be run on any Windows machine or Linux platform that supports the Java 1.5 Runtime Environment (JRE).
<b>Physical Resources</b>	The actual physical LUNs used to create logical resources, as seen by the RAID controller/storage HBA within the FDS appliance. Clients do not have access to physical resources.
<b>Logical Resources</b>	<p>These are all of the logical/virtual resources defined on the FDS server.</p> <p>Logical resources consist of sets of storage blocks from one or more physical hard disk drives. This allows the creation of logical resources that contain a portion of a larger physical disk device or an aggregation of multiple physical disk devices. For example, the logical resources listed below can all be created from a single large physical device.</p> <ul style="list-style-type: none"> <li>▪ FDS Resources: The staging area for the files.</li> <li>▪ Repository Resources: Virtualized disks configured as storage (data disks, index disks, and folder disks) for deduplicated data. <ul style="list-style-type: none"> <li>♦ Data repository disks: Where the unique data blocks are stored.</li> <li>♦ Repository Index and Folders: Where the metadata data is stored.</li> </ul> </li> </ul>

## FalconStor FDS Architecture

The typical FalconStor FDS architecture is made of three major components. Use Figure 1 as a reference.

- The first component, the front-end, communicates to the clients (i.e. database and backup servers) via CIFS and/or NFS.
- The second component, the FDS appliance itself, performs deduplication functions and handles replication between FDS appliances.
- The third component, the back-end, is the disk used by the FDS appliance to store the deduplicated backup data.

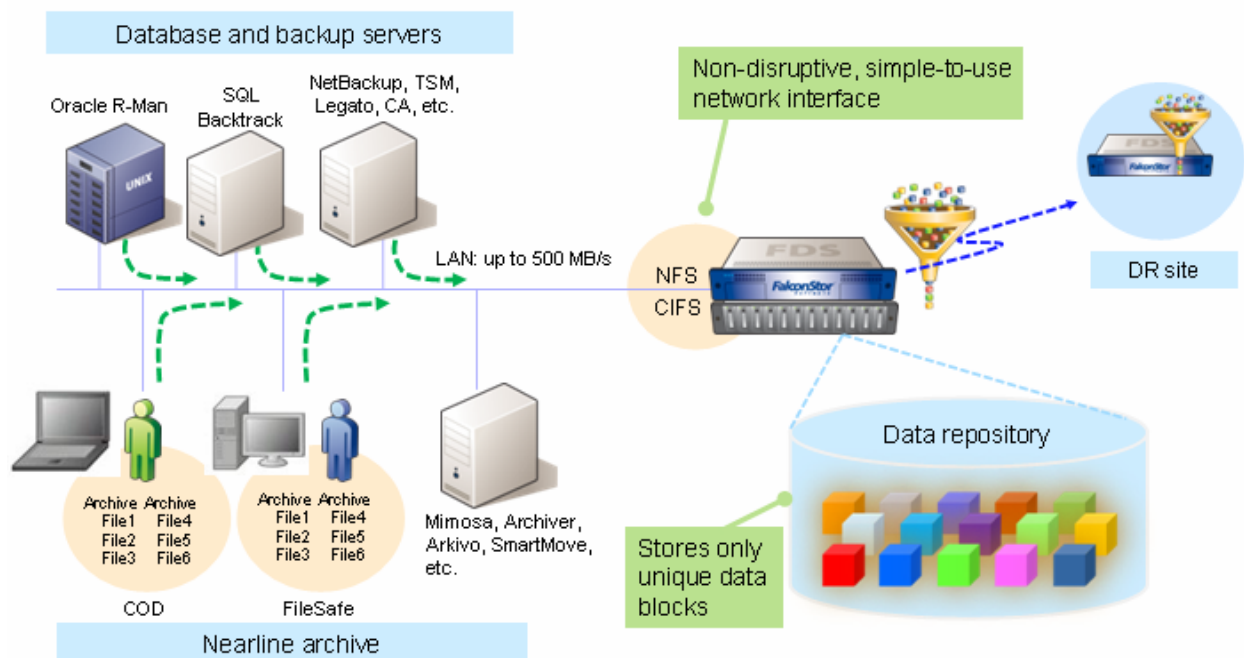
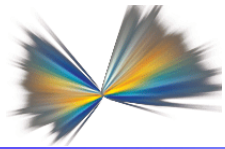


Figure 1. Typical FalconStor FDS Architecture

Depending on the FalconStor FDS licensing and packaging, the storage can be embedded or not; if so, the second and third components are integrated together. The following configurations are available:

- **Virtual Appliances:** Small footprint, ideal for small environments without demanding performance requirements, such as remote offices.
- **Physical Appliances:** Provide easy-to-deploy and easy-to-manage self-contained deduplication repository.
- **Gateway Appliances:** Integrate with existing storage infrastructure to provide storage capacity optimization over existing resources.



# EMC NetWorker

## Overview

EMC NetWorker is a high-performance data protection application. Its architecture is designed for large and complex distributed computing environments. NetWorker provides scalable storage servers (backup server and storage nodes) that can be configured for network backup, recovery, archiving, and file staging services.

## Benefits of FalconStor FDS and EMC NetWorker

FalconStor FDS provides an alternative storage solution for NetWorker customers who are faced with storage expansion associated with huge amounts of backup and archive data. By acting as a space-efficient repository for NetWorker, FalconStor FDS is also the ideal solution to centralize networked backups to a single point of administration.

Backing up to FalconStor FDS as a target for EMC NetWorker provides several benefits when compared to backing up to basic disk (with no deduplication).

- The backup storage footprint is reduced. This means less disk is required, so more space is freed up in your data center for other uses. In addition, since there is less spinning disk on the floor, power and cooling costs associated with backup are also reduced.
- Replication is possible. Basic disks may not have had any replication technology available, or at least feasible. FalconStor FDS deduplication greatly reduces replication bandwidth requirements by only replicating unique data that does not already exist at the target FDS appliance. And, because FalconStor FDS does not limit you to the type of disk you must use, the brand of storage you use with FalconStor FDS at your production site can be different than the brand that you select for the replicated data. Facilitating replication offers additional cost savings. Depending upon a customer's retention policies and storage capacity allocated to FalconStor FDS, backing up to tape and/or cloning to tape may be avoided altogether, along with offsite storage and transportation charges.

## EMC NetWorker Terminology

Component	Definition
<b>NetWorker Server</b>	This server contains the NetWorker configuration, client file index, and media database. All NetWorker operations, including backups, are governed by this server.
<b>Client</b>	A computer whose data can be backed up by one or more NetWorker servers. The NetWorker Server is also a client.
<b>Storage Node</b>	A NetWorker client that can back up its own data as well as data of other clients to a backup device (i.e. tape drive, File Type Device). The NetWorker Server is also a storage node.
<b>Dedicated Storage</b>	A NetWorker client that can back up only its own data to a backup

Component	Definition
<b>Node</b>	device.
<b>Data Zone</b>	A data zone is comprised of exactly one NetWorker Server and all of its associated storage nodes and clients.
<b>File Type Device</b>	This type of backup device is a disk backup destination which can either be local (for Microsoft Windows, UNIX, or Linux storage nodes) or NFS (for UNIX or Linux storage nodes only). <i>As a result, only UNIX/Linux storage nodes can use File Type Devices that reside on FalconStor FDS.</i> The NetWorker File Type Device is a legacy device based on an optical device interface. Streaming and recovery capabilities are limited. No enhancements and few fixes are planned for the File Type Device. It is no longer a best practice to use this type of device as a backup target for anything other than demo and test purposes.
<b>Advanced File Type Device (AFTD)</b>	This type of backup device is a disk backup destination which can either be local (for Windows, UNIX, or Linux storage nodes), CIFS (Windows only), or NFS (for UNIX or Linux storage nodes only). <i>Windows, UNIX, and Linux storage nodes can use Advanced File Type Devices that reside on FalconStor FDS.</i>
<b>Saveset</b>	A set of files or locations backed up onto a device.
<b>Group</b>	One or more clients with the same backup schedule start time.
<b>Cloning</b>	Copying backup volumes or savesets from one set of media to another. Clones function identically as the original backup volume.
<b>Staging</b>	Moving data from one storage type to another. Staging also later removes the data from its original location.
<b>Retention Policy</b>	Determines how long backups are retained on the given media. Savesets can have a different retention policy than a clone.
<b>Browsing Policy</b>	Determines how long file entries remain in the Client File Index. This facilitates recovery of individual files via the NetWorker Recovery GUI.
<b>Resource Database</b>	This database retains all the NetWorker configuration information. It resides on the NetWorker server.
<b>Client File Index Database</b>	This database contains entries for each object that is backed up. It resides on the NetWorker server.
<b>Media Database</b>	This database tracks the life cycle and location of the savesets as well as volumes. It resides on the NetWorker server.
<b>nsrim</b>	Automatically manages the server media database. It is invoked by <b>nsrmdbd</b> when it starts up at the end of the <b>savegrp</b> program. It removes both aborted and expired save sets once every 24 hours after a savegroup is completed (if Volume Recycle is set to Auto). It ( <b>nsrim</b> ) is also called by <b>nsrd</b> when a user removes the oldest backup cycle.

## Typical NetWorker Architecture

In the NetWorker data zone shown below in figure 2, there is a NetWorker server and a NetWorker storage node, which can:

- Dynamically share tape drives from the tape library
- Back up NetWorker client data over the LAN to the tape drives on the SAN
- Back up data found on the SAN storage to the tape drives

In a NetWorker backup to tape environment, FULL backups are normally performed weekly and incremental backups are typically performed daily. Following the completion of the backups (or the backup cloning process), the backup tape media is transported to an offsite vault.

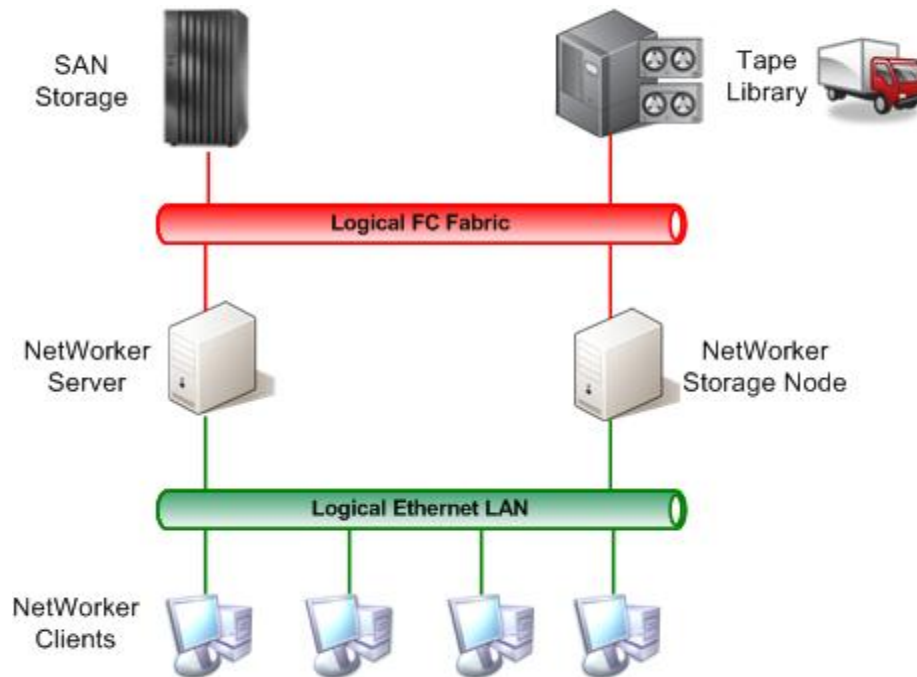


Figure 2. EMC NetWorker Data Zone



# ***Integrating FalconStor FDS with NetWorker***

---

## **Methodology**

Backing up data with NetWorker to an FDS appliance follows the same rules and procedures as standard backup to disk (i.e. CIFS or NFS shares). Given that fact, it is very important for FalconStor FDS users to understand the way NetWorker must be configured in order to back up to disk.

EMC NetWorker 7.5 provides two device types when using the DiskBackup Option, the ***File Type Device*** and the ***Advanced File Type Device (AFTD)***. The File Type Device provides for disk backups to local paths for Microsoft Windows while it provides for local paths and NFS mounts for UNIX/Linux. On the other hand, the AFTD provides for local and network paths for both Windows, via CIFS, and UNIX/Linux, via NFS. In the context of FalconStor FDS we will only be concerned with File Type Devices and AFTD's using network paths since FalconStor FDS will provide an NFS or CIFS share that a NetWorker storage node can back up to. Note that this implies that FalconStor FDS can only be used with Advanced File Type Devices on Windows.

## FalconStor FDS and NetWorker Architecture

In the NetWorker data zone diagram below in figure 3, an FDS appliance has been added. The FDS appliance is attached to a LAN (possibly a dedicated backup LAN) and may come integrated with its own local storage or provisioned with backend SAN storage. You will notice that there are no changes to the existing NetWorker data zone required. Since the data path to FalconStor FDS is via NFS or CIFS, it is most appropriate for LAN-based backups. For large SAN-based backups, FalconStor VTL with SIR (Single Instance Repository) should be considered.

In this data zone, the NetWorker server and/or storage node would have AFTDs defined in order to back up to the FDS appliance. The path to the FDS network shares could be via either NFS or CIFS depending on the respective operating systems of the NetWorker server and storage node using the FDS shares.

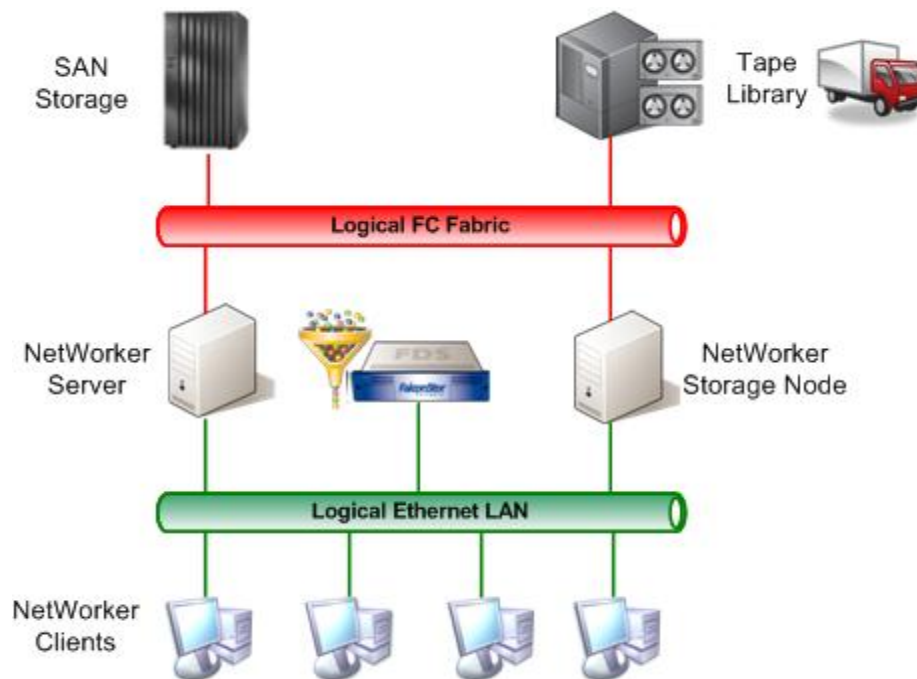


Figure 3. EMC NetWorker Data Zone with FalconStor FDS

An ideal use case for FalconStor FDS would be a customer having these requirements:

- Predominantly LAN-based backup environment
- Need to reduce/eliminate media costs
- Need to reduce/eliminate media transportation costs
- Need to reduce/eliminate offsite storage costs
- Need to improve speed of recovery from offsite media which are currently missing SLAs or SLOs
- Need to improve security when transporting media

In the NetWorker data zone shown in figure 4, backups remain on the FDS appliance for their entire retention period and are replicated to a remote FDS appliance in order to meet offsite vaulting requirements.

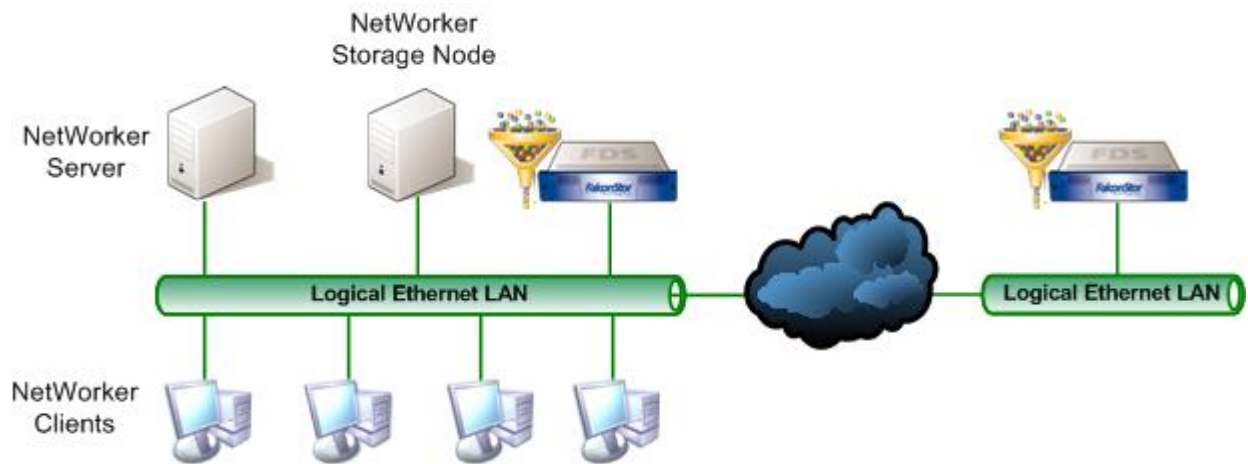


Figure 4. EMC NetWorker Data Zone with FalconStor FDS replicated remotely

## Configuration Guidelines – FalconStor FDS

### FalconStor FDS Appliance Types

When configuring FalconStor FDS for the first time, different scenarios are possible:

- The FDS appliance is an all-in-one appliance. Virtual and physical resources are already defined. The only entities that must be configured are the FDS shares.
- The FDS appliance is used as a gateway. In this scenario, the physical resources and the virtual resources must be configured.

### Sizing

The ratio of data stored (after deduplication and compression) wholly depends on different factors, such as:

- Backup methods being used and backup policies
- Retention policies
- Rate of data change
- Type of data (structured, unstructured, images, etc.)

The sizing of a FalconStor FDS solution must take into account these factors. As NetWorker is not aware of the deduplication, some general rules must be followed.

Rule #1: Propose an assessment of the backup solution so that FalconStor FDS sizing can be done in collaboration with the backup team that is in place at the customer site. A conversation about the factors listed above should be undertaken with the people in charge of the backup environment. This is a prerequisite that will significantly influence the success of the FalconStor FDS implementation.

Rule #2: Plan for initial disk design. The size of FDS physical resources must be calculated by taking into consideration the size of the initial backup cycle and by anticipating a possible deduplication ratio. This should help the FalconStor FDS administrator to determine (in the first phase) the amount of data that can be retained on disk.

As everybody knows, the benefits of data deduplication are realized over time. Therefore, FalconStor recommends reconsidering the FalconStor FDS sizing once a significant subset of production backup data is sent to a NetWorker storage node being used with a FalconStor FDS system.

By adopting an incremental approach, the FalconStor FDS administrator will be better able to forecast the space needed to sustain data growth. This method requires ongoing measurement that should help the FalconStor FDS administrator grow the system over time.

## Configuration Guidelines – EMC NetWorker

As mentioned earlier, EMC NetWorker provides two device types to facilitate disk backups: File Type Devices and AFTDs. Retention policies and staging policies, which move the *savesets* off of these disk devices, must be implemented to prevent the corresponding file systems from getting full.

### ***File Type Device***

(Refer to the *EMC NetWorker 7.5 Administration Guide* for the details on how to configure a File Type Device.)

It is important to note the impact of the Volume Default Capacity attribute. For File Type Devices, the Volume Default Capacity is a hard limit to the amount of data that can be written to the device. This setting is used to restrict the size of the File Type Device in order to avoid filling up the file system. Unlike AFTD's, File Type Devices cannot dynamically expand to use additional disk space. If additional space is made available to the file system, the Volume Default Capacity attribute must be manually updated and will take effect the next time the File Type Device is relabeled. Essentially, the File Type Device simulates a single tape cartridge. It does not put to use the advantages of writing to disk. For this reason, it is recommended that only AFTDs be used with FalconStor FDS.

### ***Advanced File Type Device***

(Refer to the *EMC NetWorker 7.5 Administration Guide* for the details on how to configure an AFTD (*adv\_file*).

One major difference between the AFTD and the File Type Device is that the AFTD ignores the Volume Default Capacity attribute and allows for dynamic expansion of a device. When an AFTD runs out of disk space:

- The current backup is suspended.
- The Recover *adv\_file* Space notification is fired.
- By default: **nsrim** is executed to delete expired *savesets* from the AFTD.

At this point, if enough space has been cleared, the backup is allowed to continue. Otherwise, if after a 10-minute waiting period measured from the time the **nsrim** was requested, there is still not enough space, then:

- Filesystem Full – Waiting for **adv\_file** Space notification is fired.
- By default: An email is sent to the root user on UNIX/Linux systems and logs a message to the Media log on Windows systems.

At this point, the backup stops until more space has been made available. The default actions described above are specific to the AFTD and can be customized to include commands to extend the file system, for example.

The fact that the AFTD supports dynamic expansion plays right into the underlying strengths of FalconStor FDS. FalconStor FDS also supports dynamic expansion of the storage that is provisioned to the NFS and CIFS shares. This allows backup administrators to add capacity to the AFTD should the device fill up during a backup by simply assigning additional physical resources to the FDS resource on which the share resides.

Note that the FDS resource is simply a staging area used during backups. The deduplication process will reclaim space from the FDS resource. For EMC NetWorker AFTDs, the deduplication process may run concurrently with the backup since deduplication can begin as soon as a file is closed. The AFTD device is composed of many directories and files. There is no need to wait until the entire backup is done in order to begin deduplication.

### **Windows Clients**

In order to configure an AFTD for a Windows storage node working with FalconStor FDS, the authentication mode must be set. The authentication mode applies to the entire appliance.

There are two authentication modes:

**Share Mode** – This authentication mode is used for Windows Workgroup environments. The Workgroup name of the device should be set to the Workgroup name of the NetWorker server or storage node that is backing up to it. Shared mode provides the ability to define two passwords: one for read-only access and a second for read/write access. The passwords are defined per share. The username is limited to “guest” for all shares. When defining an AFTD using this mode, the “guest” username and read/write password must be entered into the AFTD properties for the remote user.

- **Domain Mode** – This authentication mode is used for Windows Active Directory environments. In production environments, this mode will be more commonly used than Share Mode. When configuring this mode, the administrator will be asked to provide the domain controller (and optionally the backup domain controller) that the FDS users will be authenticated against. For more details on configuring Domain Mode authentication, refer to the *FalconStor FDS Configuration Guide*.

### ***NFS Clients***

In order to configure a File Type Device or AFTD for a UNIX/Linux storage node working with FalconStor FDS, you must first define the storage nodes, by either DNS name or IP address that will have NFS access to the FDS appliance. This is performed via the FDS console. In addition, you can specify by NFS share whether the given client has access at all and whether that access is READ-ONLY or READ/WRITE. Typically, storage nodes will have READ/WRITE access.



# Conclusion

---

Adapting EMC NetWorker to the FalconStor FDS landscape to take full advantage of its deduplication benefits by implementing AFTDs is simple and can be implemented with little or no disruption to your backup operations. FalconStor FDS is the ideal backup-to-disk target for your EMC NetWorker backup environment because it:

- Will reduce your backup storage requirements by deduplicating the backup data
- Will reduce/eliminate your tape expenses by extending your retention on disk policies
- Will reduce/eliminate your off-site tape management costs if replication between a source and target FDS appliance is implemented
- Will enhance the security of your backups by avoiding tape transportation
- Will take full advantage of the benefits afforded by the AFTD including the dynamic expansion of the backup volume

FalconStor FDS is a scalable solution that can work with heterogeneous hardware connected to SMB, SME, and enterprise solutions, overwhelmingly confirming that FalconStor FDS is a totally open solution that can be adapted to different environments, including EMC NetWorker.



# Appendix

---

## Sources

[http://en.wikipedia.org/wiki/Service\\_level\\_agreement](http://en.wikipedia.org/wiki/Service_level_agreement)

[http://en.wikipedia.org/wiki/Service\\_level\\_objective](http://en.wikipedia.org/wiki/Service_level_objective)

## Reference Documents

EMC NetWorker Release 7.5 - Release Notes

EMC NetWorker Release 7.5 - Administration Guide

## Related Documents

- FalconStor File-interface Deduplication System (FDS) Best Practices: FDS as a Backup Target for EMC NetWorker
- FalconStor FDS User Guide
- FalconStor FDS Configuration Guide
- [FalconStor Demystifying Data Deduplication: Choosing the Best Solution White Paper](#)