

# OPTIMIZING PRIMARY STORAGE THROUGH FILE ARCHIVING WITH EMC CLOUD TIERING APPLIANCE

## A Detailed Review

### Abstract

This paper describes the EMC Cloud Tiering Appliance. The Cloud Tiering Appliance (CTA) enables NAS data tiering, allowing administrators to move inactive data off of high-performance storage to less-expensive archival storage, thereby enabling more cost-effective use of file storage. The CTA also facilitates data migration, allowing the movement of data to new shares or exports, as would be required by hardware refreshes, for example.

August 2012

Copyright © 2012 EMC Corporation. All Rights Reserved.

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

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

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on [EMC.com](http://EMC.com).

VMware and VMware ESX are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other trademarks used herein are the property of their respective owners.

Part Number h19056

## Table of Contents

<b>Executive summary</b> .....	<b>4</b>
Business case .....	4
Solution overview.....	4
<b>Introduction</b> .....	<b>5</b>
Purpose.....	5
Scope.....	5
Audience.....	5
Terminology .....	5
<b>Archiving</b> .....	<b>6</b>
Overview .....	6
Hierarchical Storage Management (HSM) .....	6
FileMover .....	7
Archive Policies .....	8
Providing data to the CTA Archiver .....	8
Scheduler.....	9
Simulation.....	9
File recall.....	9
Recall with CTA-HA .....	10
Archival requirements for the source NAS server .....	10
Archival requirements for target repository server.....	11
Multi-tiered archive .....	12
CTA database .....	13
Stub scanner jobs .....	13
Orphans .....	13
Reporting .....	14
<b>Migration</b> .....	<b>14</b>
Overview .....	14
Migration source - NAS servers.....	15
Migration target - NAS servers .....	16
The migration process .....	16
Other interactions with VNX.....	17
Miscellaneous topics.....	18
<b>Summary</b> .....	<b>18</b>

## Executive summary

The EMC® Cloud Tiering Appliance can optimize primary NAS storage by automatically archiving inactive files to less-expensive secondary storage. The secondary storage might be lower cost disk, such as NL-SAS or SATA, on a NAS device. The secondary storage might also consist of public or private cloud platforms. When a file is archived, a small stub file is left behind, so that the file appears to the user as if it were in its original location. File tiering dramatically improves storage efficiency and shortens backup and restore time.

In addition to archiving data from primary to secondary storage, the Cloud Tiering Appliance can also permanently migrate files from a source to a destination without leaving a stub, as might be required, for example, for technology refreshes of NAS server hardware.

## Business case

From the early days of computing, computer storage has always existed in different tiers, or levels, separated by differences in cost, performance, availability/redundancy, etc. Today's flash storage, for example, outperforms storage made up of NL-SAS or SATA, but it comes with a higher price per gigabyte.

NAS data also exists in “tiers”. Not all data has the same value. As data ages, it is normally accessed less frequently. In order to make best use of a typical tiered NAS storage environment, a storage customer must find some way to ensure that the less-valuable, less frequently accessed data does not consume high-speed, expensive storage resources. The high performance storage should be reserved for the high-activity, important data, while the less active data should reside on cheaper storage.

Any process that ensures this tiering of storage should work automatically, so as not to add to the already significant activities and tasks of the storage administrator.

## Solution overview

EMC Cloud Tiering Appliance (CTA) employs the tried and true concept of Hierarchical Storage Management (“HSM”) that has been a staple of the mainframe world for decades. Hierarchical Storage Management involves the movement of a file from primary storage to lower-cost secondary storage, with a small stub pointer file left behind in the file's original location. This process of relocating data and stubbing is also known as “archiving” or “tiering”.

CTA acts as a “policy engine”, whereby it interacts with a VNX share or export, and identifies any files that fit certain predefined criteria (e.g. “has anyone accessed this file in six months?”). For these files, CTA initiates the movement of these files to a lower “repository” tier of storage (NAS, Centera, or cloud). A stub file is left behind on the VNX share, in the file's original location. When a client who is accessing the VNX share/export tries to read the stub, the original file is recalled from the repository tier and presented to the client. The result: the file appears, to the user, to be in its

original location, on high-performance VNX storage, when in fact it has been relocated to a lower tier. Instead of the entire file consuming space on the primary tier, the only space that is occupied is 8KB (the size of the stub).

If the storage administrator decides to move the data in the share or export to another location (for example, to replace an old Celerra with a new VNX), CTA can help. Included in CTA is a “file migration” feature, which will allow the relocation of data from one share/export to another, even if the source data is multiprotocol, or contains CTA stubs.

When used for archiving/tiering, CTA will automatically move inactive data to lower tiers of storage, allowing more efficient, less wasteful use of the most expensive, highest-performing NAS storage.

When used for file migration, CTA enables more efficient, easier relocation of NAS data within a NAS server, across NAS servers, and even across NAS servers from different vendors.

## Introduction

### Purpose

This white paper provides an understanding of what CTA does, how it functions, and what are the business problems it helps solve.

### Scope

CTA features and a technical overview of the product are presented in this paper. In addition, information on how to manage CTA and implement the solution in a VNX NAS environment is provided.

### Audience

This white paper is intended for people who have a basic understanding of VNX Unified Storage or, at a minimum, general NAS storage concepts.

### Terminology

**Archive Repository** – Storage that represents a lower tier of storage than the NAS storage that the CIFS or NFS clients access. The repository is the target of a file archival process. In an archive operation, data is moved from the primary tier (or source tier) to the repository. A stub file that points to the actual file in the repository is placed on the primary tier. A repository tier may be a NAS share/export, an EMC Centera, an EMC Atmos cloud, or the Amazon S3 cloud.

**File Archival** – A primary function of CTA, file archiving involves the scanning of a NAS file server for files fitting a particular set of rules (e.g. files that are older than 1 year), and the movement of these files to a lower tier of storage. The file is replaced on the NAS server with a stub file that points to the real file, which now resides on the archive repository.

**File Migration** – The movement of files from one export/share to another. The most common use case for this is for the replacement of a NAS server for, say, a newer model.

**File Mover** – Software that is included on VNX data movers. FileMover (or “dhsm”) enables stubbing and recall on archived files, and provides an API that the CTA uses for both archive and migration. FileMover must be enabled on any Celerra or VNX file system/export/share that will be used for file archiving.

**File Tiering** – See File Archival.

**FPolicy** – Software that is included on NetApp filers. FPolicy provides a similar function to FileMover on the VNX. The CTA uses the FPolicy interface in order to archive files from NetApp servers.

**Orphan file** – When a file has been archived, there is a stub on the source NAS server that points to it. If the stub is deleted, the archived file IS NOT automatically deleted as well. Instead, it becomes an “orphan” (i.e. a repository file that has no stub pointing to it). It will remain in the repository, as an orphan, until it is specifically deleted (usually by running an “orphan delete” job with CTA).

**Policy** – A set of rules (for migrations) or a set of rules and one or more repository destinations (for archiving/tiering). An archiving policy, for example, might say “if a file has not been accessed in 1 year, send it to my company’s private Atmos cloud ‘atm01’. If a file hasn’t been accessed in 2 years, then send it to the public Amazon S3 cloud.”

**Primary Storage** – The storage tier that CIFS and NFS clients mount on the VNX.

**Source Tier** – See “Primary Storage”

## Archiving

### Overview

Cloud Tiering Appliance provides two primary functions: Archiving (“Tiering”) and Migration. Archiving, the movement of inactive data to a lower tier of storage (leaving behind a stub file), will be discussed in this section.

### Hierarchical Storage Management (HSM)

In the days of mainframes, a few decades ago, there was a natural tiering of storage that came about due to the great expense associated with disk storage and the existence of relatively inexpensive tape storage. A mainframe user’s disk storage was limited, so the admin would often (manually) move less important data to tape. The admin was responsible for remembering where (on what tape volume) the data was kept, so it could be found when needed.

Mainframe developers began to consider solutions to this problem. How do you move the data to cheaper, external storage, but eliminate the need for the admin to

remember where it was placed, and eliminate the requirement to manually run recall jobs to recover the data when it was needed?

The solution was Hierarchical Storage Management (HSM). The concept is simple—the system would scan the data, find files that had not been accessed in some period of time, and automatically move them to a lower tier of storage (generally, tape in those days). In place of the file, a small stub would be left behind that contained an internal pointer to the file on tape. The user would see the stub as if it were their actual file, but when they tried to access it, they would be told to wait while the system automatically retrieved the file from tape and restored it to the user's disk, in its original location.

Hierarchical Storage Management is the basis for the Cloud Tiering Appliance's archive function. The concept is straightforward, robust, and time-tested. Of course, fewer and fewer people use tape as a secondary storage tier these days. CTA therefore supports archiving only to disk storage (cloud, CAS, or NAS). But the existence of storage tiers in nearly all customer environments today make the HSM tiering solution as important today as it was decades ago in the mainframe era.

## FileMover

Included with every VNX (and Celerra) that provides file services (NFS, CIFS) is FileMover. FileMover is software that, among other things, allows HSM-style archiving. On VNX, the primary FileMover command is “dshm”, which shows FileMover's HSM roots.

At its most basic level, FileMover is used to intercept client access to data so that an action might be taken before the client is presented the data they are trying to access. An external system might interact with FileMover to say “if the client tries to access a file with these characteristics (a, b, c, etc.), then take action X.” CTA works with FileMover in this way. The CTA administrator will set up a set of rules to present to FileMover. CTA might say to FileMover on the VNX, “I'll scan the files in this share. When we find a file that is more than 60 days old, we'll move it to the archive and replace it with a stub. We will turn on the CIFS offline bit on the stub.” Once the stub is in place, FileMover monitors the stub. When a client tries to access the stub (read or write), FileMover will intercept that access and effect a recall of the data, using information contained in the stub. The data will be presented to the client that made the request.

The FileMover API must be enabled and configured for each VNX or Celerra that is to be used as an archive source. This is done during CTA configuration setup.

NetApp has an API, called “FPolicy”, that performs a function similar to FileMover.

The existence of these APIs and functions (FileMover, FPolicy) is prerequisite to being able to archive from a NAS system using CTA. The absence of these functions from other platforms (e.g. Linux, Windows, etc.) is why, currently, CTA can only be used to archive data from VNX, Celerra, and NetApp.

## Archive Policies

A “policy”, in archiving/tiering context, is simply a set of rules and one or more destinations. In its simplest form, a policy might say “if this file has not been accessed in six months, send it to the Atmos cloud, and replace it with a stub”. The one-rule, one-destination is quite common, and many CTA users will use this type of policy on all of their data.

However, CTA rules are very flexible. You can create much more complex rules, involving archiving to multiple tiers. For example, you might say, in a single policy:

“If any file has not been accessed in more than one year, AND is larger than 1 megabyte in size, send it to my Isilon, UNLESS it’s a pdf file, in which case don’t archive it. THEN, when these files have reached two years of not being accessed, move them from the Isilon and send them to the Atmos cloud, and update the stub file to point to their new location.”

Policy rules can be based on access time (“atime”), modify time (“mtime”), inode change time (“ctime”), file size, file name (a regular expression that looks for certain characters in the name), and directory name. The rules can say “if xxx, then archive”, or they can say “if xxx, then don’t archive”. You can use combinations of “ands” (multiple statements in a rule) and “ors” (multiple rules).

A policy does not contain a share name. Policies are created independent of the data they will be used against. So, you can use one policy against shares A, B, and C, but run another policy against share D. Or, you can apply several different policies against a single share (although this is not a typical way to run policies).

## Providing data to the CTA Archiver

In most cases, administrators will simply point a CTA archive policy at a file system, CIFS share, or NFS export. CTA will scan all of the files in the share and apply the policy rules to each file, one at a time. If there are multiple rules in the file, the CTA will keep applying the rules, one at a time, until a rule evaluates to “true”. It will then take the action associated with the rule (e.g. “archive”, or “don’t archive”) and move on to the next file.

There is another way to provide files to a CTA policy. Instead of telling CTA to scan all of the files in a share, a list of filenames can be provided to CTA, and CTA will scan only the files in that list, applying a policy to the files one at a time. This feature, called “file ingest”, was designed primarily for third-party vendors who might have software products that scan file systems in their own way, but want to use the CTA as their archival engine. File ingest won’t be discussed in detail in this paper. This paper will assume that CTA policies are run against file systems/shares/exports that are scanned.

## Scheduler

The scheduler is used to apply a policy to a share for the purpose of archiving data from that share. For example, a batch job can be defined that says, “at 2:00am on Saturday, scan share01 with policy A. For every file in share01, apply policy A, and decide whether or not to archive that file. “ The job that runs policy A against share01 is scheduled by the CTA administrator (using the GUI or command line). It is most common for an administrator to schedule the job to run weekly, every other week, or monthly. The first time an archival job is run the archival policy will often select at least half of the data to be archived. (It is very common to find that more than half of an organization’s NAS data hasn’t been accessed in over six months). So, the first archival job may run for awhile and move a lot of data. If the job is run, say, weekly, then future jobs will move incremental amounts of data (e.g. the next week, the job will only archive data that reached 6 months of age in the past week).

## Simulation

CTA is able to simulate archive jobs. In other words, a CTA administrator can schedule an archive job with a policy, but run it in simulation mode. The CTA will scan the source share and apply the policy rules against each file but not take any action. Instead, CTA will keep track of the number of files and amount of data it would have archived and at the end of the simulation display a report. Simulation is a good way to test the effectiveness of a policy and to make tweaks in the policy rules before actually running a real archival job.

## File recall

When a file has been archived to a repository and a stub is placed on the source NAS share, the expectation of the NAS client is that the stub will look and behave like the original file. When the user clicks on the stub file, the user should quickly be provided with that original file. This process is called “file recall”.

The stub file contains all of the information needed to find the actual file, which is somewhere in an archival repository. The VNX can identify a stub file because an offline bit on the file was set when the stub was created during archival. FileMover interacts with CTA so that when a user attempts to read a stub file, it will intercept that read request and begin the process of recalling the actual file. FileMover will read the stub file and directly recall the data, using CIFS or NFS, if the repository is on a CIFS or NFS share. If the repository is a CAS or cloud (Centera, Atmos, or Amazon), then VNX will send the recall request to CTA, which will effect the recall and pass the recalled file on to the VNX. (NetApp must always employ CTA to recall files, no matter what type of repository the file has been archived to.)

When the VNX presents the recalled file to the requesting user, it does one of two things. It either will simply provide the file to the user, but leave the stub in place (“passthrough recall”), or it will write the file back to its original location and delete the stub (“full recall”). These two recall styles are set on a file system basis by

running a FileMover command on the VNX control station. (On a NetApp filer, passthrough or full recall is set filer-by-filer, rather than file system-by-file system as it is on the VNX.)

## Recall with CTA-HA

Archive and migration are batch jobs that, if they fail, don't cause loss of access to data. If an archive or migration job fails, the administrator merely fixes the problem and re-runs the job. For this reason, the complexity of a High Availability (HA) configuration is not necessary or justified to ensure continuous archiving. However, recall *is* considered mission-critical because it affects a customer's ability to access data. For this reason, in configurations where the CTA is in the recall path (archive from VNX/Celerra to Centera, Atmos, or Amazon, or all archival from NetApp), an HA configuration is required.

The CTA-HA physical appliance (or the CTA/VE-HA virtual appliance) is a system that can be paired with one or more CTA or CTA/VE systems when an HA configuration is required for continuous recall. The CTA-HA system is a recall-only version of the CTA. It is configured so that all data recalls that cannot be done directly by the source NAS server (see above) can be performed by either recall host: the CTA or its CTA-HA partner. By creating a DNS hostname that maps to *both* IP addresses (the CTA and CTA-HA), and by telling the VNX to find the CTA by using that hostname, the VNX will be presented with both recall hosts alternately, in a round-robin fashion. This balances the recall load, and in the event of a failure of one recall host, the other will perform all recalls until the failed host is brought back online. This configuration also allows one of the recall hosts to be brought down for maintenance (e.g. OS upgrades) while the other continues to perform recalls.

CTA-HA also performs keystore replication for encryption keys that are generated on CTA when the encryption feature is selected during archive to Atmos or Amazon clouds (see below).

## Archival requirements for the source NAS server

CTA in its present form can archive data only from CIFS or NFS shares on VNX, Celerra, and NetApp NAS servers.

FileMover (VNX/Celerra) and FPolicy (NetApp) both provide archive services and both require an offline bit to be set on stubs when they are created, so that they can be identified as stub files. The CIFS protocol contains support for offline bits, but NFS does not. For this reason, a CTA administrator who wishes to archive data from an NFS export on a NetApp volume must also create a CIFS share at the same level or higher on the volume, even if the export is only to be used for NFS access. This allows FPolicy to create and read the offline bits that are necessary for archiving. This is not necessary if the source NAS system is Celerra or VNX, because these platforms are able to handle offline bit creation internally.

CTA communicates with VNX and Celerra using the dhsm (i.e. "FileMover") API. There are a number of setup steps required when first configuring a VNX or Celerra to work as an archiving source with CTA. These are outlined in the *Cloud Tiering Appliance*

*and Cloud Tiering Appliance/VE Getting Started Guide.* Before an archive of data from a VNX or Celerra can be performed, a “connection” that links a file system to one or more repositories must be configured. These connections are created automatically by CTA and FileMover when needed (provided the VNX/Celerra was set up properly during initial configuration). An advantage of using connections is that in the event all of the archived data must be recalled from a repository archive, it is only necessary to delete the connection by running a command on the VNX or Celerra control station. This will (optionally) trigger a recall of all stubbed data on that file system from that connection’s repository. NetApp has no concept of a connection, and there is no way to trigger a full recall of archived data to a NetApp share except by putting the filer into Full-Recall mode and reading each stub file to trigger recalls.

CTA and CTA-HA must have full control of the source shares. If the source consists of NFS exports, these exports must be exported root and read/write to the IP address of the CTA and CTA-HA(s) defined to the source server.

For archiving from CIFS shares, the source server must belong to a domain, and it must be defined to the CTA using a username from that domain. This username must be in the local administrator’s group of the CIFS server that will be associated with the source share.

### Archival requirements for target repository server

The CTA is able to archive to three different kinds of repositories:

- NAS (CIFS or NFS)—VNX, Celerra, VNXe, Data Domain, Isilon, Windows, NetApp
- CAS —Centera
- Cloud —Atmos, Amazon S3

The requirements for configuring each are slightly different.

- For NAS repositories, the requirements are similar to the requirements for CIFS and NFS that were outlined for the source servers (CIFS domain user in the local admin group of the CIFS server, NFS exports to the CTA and CTA-HA IPs with root and r/w permissions).
- For Centera, PEA file (or “anonymous”) is used.
- For the cloud tiers, tenant user (Atmos) and bucket user (Amazon S3).

One repository can be used as an archive target for multiple CTAs, and one CTA can archive to multiple repositories. A CTA repository migration job can be run that will take all of the archived data in a repository and move it to a different repository, updating the stubs to point to the new location along the way.

CTA repositories are not meant to be accessible by anything other than the CTA. It is possible to mount the NAS share that serves as the repository, but the layout of archived data is proprietary and any changes that one might make to the repository could result in archived data being rendered unrecallable. It is considered a best-practice to make repositories visible and available only to the CTA and CTA-HA.

CTA allows compression and encryption when archiving to either of the two cloud repository tiers (Atmos or Amazon S3).

Compression is enabled when a policy is created by selecting a compression style (fast vs. strong) and checking a box.

Encryption is also enabled by selecting a checkbox, but some administrative work must be performed before encryption can be employed:

1. Set up a keystore replication between the CTA and a CTA-HA machine.
2. Generate a key through the CTA GUI. The key will be placed in the keystore and replicated to the CTA-HA, and it will be applied to every archival that takes place under control of the policy or policies that have the “encrypted” checkbox selected. If a new key is generated later, it will be replicated and used for all new encrypted archives, and the old keys will remain in the keystore to be used during recalls of files that were originally encrypted by those old keys.

It is essential that the keystore be preserved. The replication will be sufficient for normal outages, but backups of the CTA should be performed any time a new key is generated, so that keys are never lost.

### Multi-tiered archive

A use case for the CTA might be something like this:

Archive my NAS files to my private cloud storage if no one has accessed these files in six months or to public cloud (e.g. Amazon S3 or Atmos-based public cloud) if the files have not been accessed in one year. Additionally, if the six-month old files on the private cloud eventually reach one year of time without being accessed, move them from the private to the public cloud and update the stub files to point to them in their new location.

This kind of archiving scenario is possible through the use of the CTA’s multi-tiered archiving feature. By creating a policy with the type “multi-tiered”, and by creating several rules in this policy, each with a different repository, an archive scheme such as the one in the example above can be created.

Consider the following rules that might make up a multi-tiered archiving policy:

```
if atime > 1 year  archive to Amazon S3 cloud
if atime > 6 months  archive to private Atmos cloud
```

These rules will effect the archive described in the scenario above. Note, however, that the order of the rules is important. When multiple rules exist in a policy, they will be applied one at a time, and when the first rule evaluates as “true”, the action (archive, or don’t archive) will be applied. The subsequent rules will never be applied, and the policy will move on to the next file. In the example above, if the order of rules were reversed and the 6 months rule were evaluated first, then the 1 year old rule would never be applied, because any file older than 1 year is also older than 6

months. *All* data older than 6 months would be archived to the private Atmos cloud, which is not what was intended.

## CTA database

When a file is archived, the stub on the source NAS tier has a pointer to the file's location in the repository. However, the file in the repository has no backwards pointer. The repository files have no idea where they came from. This is one of the primary reasons for including a database with the CTA. Each time a file is archived, an entry is made in the CTA database that records the file's original location (which is now the location of the stub), and the file's location in the repository. The database is NOT required for recalls as the stub contains all of the information necessary in order to locate the file in the repository. However, because the database contains entries for every archived file, it can be used for statistical reporting and for orphan management (described below). It is important that the database be protected, so an administrator can set up, in the scheduler, a job that regularly backs up the CTA database. In the event of the loss of a CTA, a new CTA can be built and the most recent database backup can be imported into the CTA so that no statistical or orphan information is lost.

## Stub scanner jobs

When an archive job is scheduled against a source share, a “stub scanner” job is automatically scheduled by the CTA to run monthly against that source share. The stub scanner job is a utility that reads all of the stubs in a share and compares them to the entries in the CTA database. Its purpose is to keep the database as up-to-date as possible, in the event that stubs are moved to different locations or orphans are formed. If a stub has been moved to a different directory, for example, and the database has not yet been updated with the change by a stub-scanner job, the stub will still point to the archived file and recalls will still occur. It is not critical that the database be exactly in synch with the location of stubs at all times, but the closer the database matches the stub and repository file locations on the system, the more efficiently the repository storage will be utilized. It is possible to run stub scanner jobs more frequently than the default of 30 days, but it is generally not necessary.

## Orphans

Users will often delete files, of course. When the files they delete actually happen to be stubs, the stubs are deleted from the source storage, but the actual files in the repository become “orphans”, and *are not* automatically deleted. This is for a very good reason—generally a storage administrator will choose to back up stubs when backing up CTA-archived shares. Many backup products (e.g. NDMP-based) will back up stubs by default. Given that the repositories are protected (replication or normal backup of repository data that is static and changes only when archive or certain maintenance jobs are run) and that stubs are very small, a NAS server that employs CTA can benefit greatly from much faster backups and smaller backup windows. However, it is important to consider that those backups may need to be restored at some point, and if this happens, stub files need to be pointing at something. If CTA

had been designed to delete archived files whenever stubs were deleted, then it's possible that restoring files from backup would result in stubs that point to nothing.

In order to delete orphan files and recover space on the repository, it is necessary to run an "orphan\_delete" job periodically. The CTA administrator can specify how old the orphan files need to be before they're deleted. As a best practice, orphans should not be deleted until there's no chance the stubs that point to them will be restored from backups. For example, backups are kept for for six months, then orphan deletion jobs should only delete orphans that have been orphans for at least eight months.

The CTA database and the stub scanner play important roles in the management of orphan files. Every time the stub scanner sees a stub, it records a "last seen" time in the database. If the stub is deleted, the stub scanner sees that, and it knows that the file that the stub pointed to is now an orphan. Because it has a "last seen" time, it knows how long the file has been an orphan. The orphan delete jobs can use this orphan age when determining which orphans to delete.

If CTA's database were ever lost, there would be no way to determine the location and age of orphan files in the repository. It would be possible to recover the space consumed by those unknown orphans, but only with a bit of effort. It's far easier to make sure the database is regularly backed up.

## Reporting

The CTA's reporting capabilities are generally limited to the files it actually archives or migrates. Reports can be generated that show the size, number of files archived, and breakdown, by file types, of archived files, but CTA will not provide a detailed profile of the data in the file system. By running archive simulations, an administrator can obtain information on file ages (e.g. run a simulation that selects files with access times greater than 2 years to get an idea of how actively users are accessing the data). For detailed breakdowns of the files in the file systems, there are other tools that can be deployed, which can sometimes be useful in determining how to configure CTA archive policies

## Migration

### Overview

The Cloud Tiering Appliance provides two primary functions: Archive ("Tiering") and Migration. Migration, the movement of files from one share or export to another, will be discussed in this section.

Generally speaking, file migration involves the copying of data from one share or export on one system to a share or export on another. The most common use of migration techniques is the replacement of servers. As new server technology becomes available, it is important for administrators to find ways to easily move data from the old servers to the new with minimum disruption to the NAS client users.

CTA can be used to perform multi-protocol, incremental, stub-aware, cross-vendor migrations, and can greatly reduce the effort and complexity of implementing new NAS technology, even from a new vendor.

A CTA migration is a batch job that moves data from a source NAS share to a target. The target share must be of sufficient size to hold the data that will be migrated from the source share, but otherwise can be of any size. It does not need to be the same size or layout as the source share.

CTA migrations can be performed on CIFS, NFS, or multi-protocol CIFS/NFS file systems. The supported source platforms are: VNX, Celerra, NetApp, Windows. The supported target platforms are: VNX, VNXe, Celerra, Isilon. Data from any source can be migrated to any target (except Windows source, which can be migrated only to VNX).

### Migration source - NAS servers

CTA migration uses NDMP as its file transfer engine when migrating from VNX, Celerra, and NetApp. It uses EMCOPY when migrating from Windows.

The NDMP-style migrations are policy-based. The CTA administrator first creates a policy, similar to the archive policies described above. For example, if you want to migrate all of the data in share1 to newshare1, but you don't want to include .pdf files that are more than 3 years old, you can create a rule to omit these files from the migration. If you don't wish to filter the migration, but wish to simply copy everything, you simply create a trivial policy with a rule that says, for example,

“if size >= 0 bytes, then migrate”

The EMCOPY-based migration (Windows to VNX) is not policy-based. When migrating from Windows, all data will be copied.

CTA will use the FileMover (“dhsm”) API when migrating from VNX or Celerra. CTA will create snapshots on the source file system by making API calls. The snapshots allow users to continue to access and write to the source share while the migration is taking place. Likewise, snapshots will be created on the NetApp source when migrations take place from NetApp filers.

CTA requires the same kind of access permissions on the source and target shares/exports as in the archive configurations. For CIFS migrations, a domain user in the local admin group on the CIFS server must be defined to the CTA when the CIFS server is defined to the CTA, and the source and target share must be exported root and r/w to the CTA when NFS migrations are performed. For multi-protocol migrations, both of these must be done. (The CTA-HA has no part in the migration process.)

An NDMP user must exist on the source server for the NDMP-style migrations. The userid and password for this user is provided to the CTA when the server is configured on the CTA.

---

NOTE: Celerra and VNX allow you to specify a password for this NDMP user. On NetApp, you create a user with a password, but NetApp will generate another, different password that you need to supply when you configure the NetApp to the CTA. This is explained in the *Cloud Tiering Appliance and Cloud Tiering Appliance/VE Getting Started Guide*.

---

When migrating from Windows, a small agent must be installed (basically, a wrapper for EMCopy) on the Windows server. Only Windows 2003 and 2008 are currently supported as migration source Windows platforms.

### Migration target - NAS servers

The target shares/exports must be created on the target servers ahead of time; CTA will not create them automatically. If you are migrating to the root level in these shares, the shares must be empty (except for file system files such as .lost+found, .etc, etc.). However, you don't need to migrate to the root level. You can migrate to any empty directory in the share. CTA needs the same credentials/permissions on these file systems as on the source. For CIFS migration targets a domain user in the local admin group on the CIFS server must be defined to the CTA when the target CIFS server is defined to the CTA. For NFS, the target share must be exported root and r/w to the CTA. Both of these are required for multi-protocol migrations.

The target share/export must match the source (CIFS, NFS, or Mixed). Additionally, an NDMP user must be configured (see above) for the target servers in NDMP-style migrations.

### The migration process

Migrations run as scheduled batch jobs. A migration task is scheduled on the CTA Schedule page (or through the CLI). A source share is specified first, then a policy (required for NDMP-style migrations, but not allowed for Windows-to-VNX EMCopy-style migrations) and a target share/directory (which must be empty).

When the job begins, CTA first creates a snapshot on the source. It then copies all of the data, per policy (except for Windows migrations), on the source to the target. A migration can begin anywhere in the source share (at the top level, in a subdirectory, etc.) and migrate to any *empty* directory on the target.

After the first pass completes, the target will have a copy of the data from the snapshot on the source (except for any files that may have been filtered through a policy). However, the source share will, at this point, probably be different from the snapshot if users have continued updating the source while the migration was going on. So, a second pass can be run to pick up those incremental changes. CTA will create another snapshot and compare the second to the first to pick up changes (new files, deleted files, metadata changes, etc.). Multiple passes can be run, but at some point, the administrator will need to stop updates to the source (by taking it offline or by putting it into read-only mode to the clients) to make a final pass, thus ensuring that the target is identical to the source. At this point, clients can be cut over to the target share and the migration is complete. All CIFS and/or NFS ACLs/permissions will

have been copied to the target. After the clients have been cut over, the source share can be deleted and its storage space recovered on the source NAS system.

The CTA migration tasks can be throttled by specifying a maximum MB/sec rate during task scheduling so that migration activity doesn't consume too much network bandwidth. The administrator can also, if needed, use a provided tool to create a SID mapping of local to domain SIDs ahead of time (using a SID translation tool). The mapping will be applied at migration time, so that the SIDs on the target will be correct.

Finally, migration tasks can be run in continuous mode. CTA has the ability to run incremental migrations continuously until a file-moved threshold is reached, e.g. "Keep running incrementals until fewer than 1000 files are moved in one incremental run. At this point, stop and notify the administrator". More often, however, customers choose to perform incremental runs during scheduled off hour periods over the course of several evenings, with the final locked cutover coming during off hours as well. It depends, of course, on NAS usage patterns and activity windows.

Along with the fact that migration can be quite asymmetrical (between file systems of different size, different disk layouts, crossing different platforms, even from different vendors), CTA migration is also unique in that it provides profile filtering (except for Windows sources) and multi-protocol migration (for example, from NetApp "multi" to VNX "multi", which are not exactly alike). It also handles migration of data that includes CTA stubs. If the source and target both support CTA stubs, the migration job will copy the stubs without recalling the archived data, and the stubs will still function properly after migration. If the source is VNX or Celerra and the target is a platform that does not support stubs (i.e. Isilon or VNXe), the actual files, and not the stubs, will be migrated. If the source is a NetApp and the target is Isilon or VNXe, the archived data must first be recalled, due to limitations in the NetApp FPolicy and NDMP implementations.

### Other interactions with VNX

CTA supports VNX File-Level Retention (FLR). When scheduling an archive to a VNX (or Celerra or Centera), the CTA administrator can specify retention times on the archived data. Optionally, retention times (managed by FileMover) can be set on the stub files. CTA does not support FLR-enabled source file systems. Stub retention time on the source is managed by FileMover.

---

NOTE: Because of the way NDMP works, the preservation of access times during a migration was previously not possible. NDMP would always set access times equal to modification times when moving the data. However, a new feature in VNX 7.1 now allows, by default, the preservation of access times during a file migration, when VNX 7.1 is the migration source.

---

Deduplication on the repository shares of a VNX can add a great deal to the efficiency of repository storage. Deduplication is not recommended for source shares. Stub files are unique and will not benefit from de-duplication.

### Miscellaneous topics

There are a number of other types of scheduled tasks that the CTA can perform, in addition to those mentioned previously.

- “Delete stub with policy” is the only task type that can delete stubs, the repository file they point to, and remove the reference to them from CTA’s database. As such, it is a task type that should be used with great caution.
- The “Multi-tier stub with policy” task type is similar to the “Multi-tier archive” task type described previously, except that it only scans stubs. It can be useful for moving some archived data from a repository to another, without moving all of the archived data. Like the “delete stub with policy” task, it scans only stubs and not regular files.

### Summary

As we have seen, the Cloud Tiering Appliance (CTA) enables NAS data tiering, allowing administrators to move inactive data off of high-performance storage to less-expensive archival storage, thereby enabling more cost-effective use of file storage. CTA also facilitates data migration, allowing the movement of data to new shares or exports, on the same or different servers, even from different vendors.