



Education

DFS Over CIFS

Matthew Geddes

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced without modification
 - ◆ The SNIA must be acknowledged as source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the Author nor the Presenter is an attorney and nothing in this presentation is intended to be nor should be construed as legal advice or opinion. If you need legal advice or legal opinion please contact an attorney.
- The information presented herein represents the Author's personal opinion and current understanding of the issues involved. The Author, the Presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

➤ DFS Over CIFS

- ◆ This session is geared toward the managers of CIFS-based storage systems; especially those that may be looking for novel approaches to common problems.

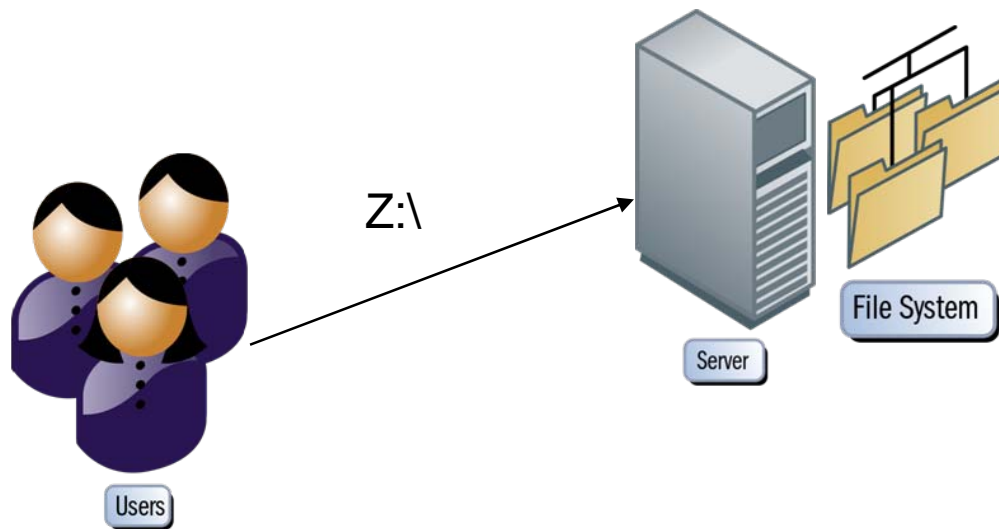
Outline

- Introduction
- Core DFS Overview
- Other DFS-Related Features
- Practical Uses Of DFS
- Conclusion
- Q&A

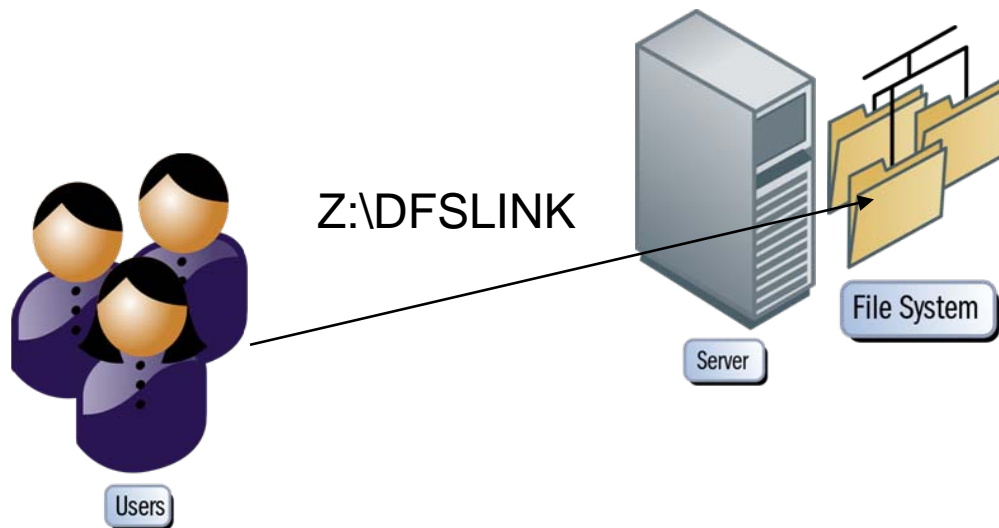
Introduction

- Distributed File System (DFS to its friends)
- An implementation of a global namespace for CIFS
- Can be considered similar to explorer shortcuts or symbolic links that can point to a remote CIFS host
- Works on top of CIFS transparently to the user

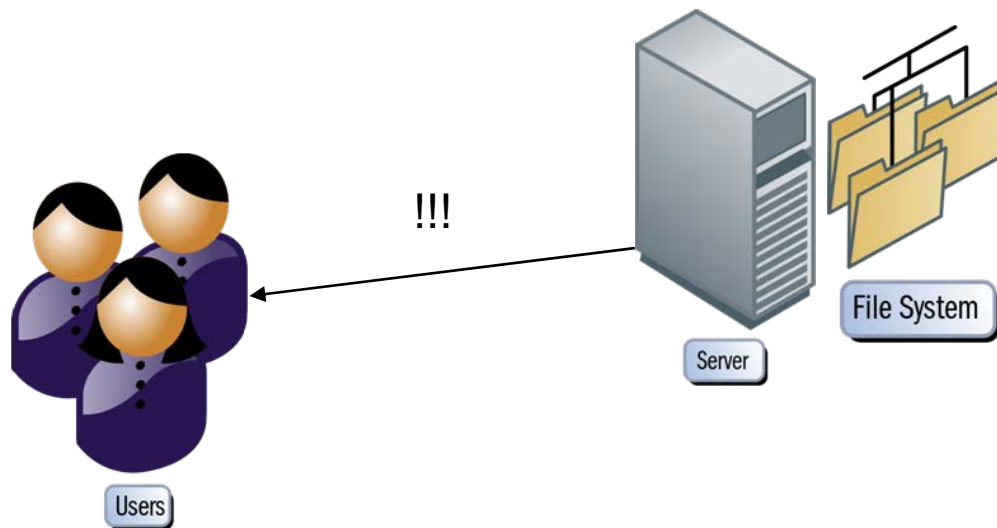
➤ Client traverses remote filesystem tree



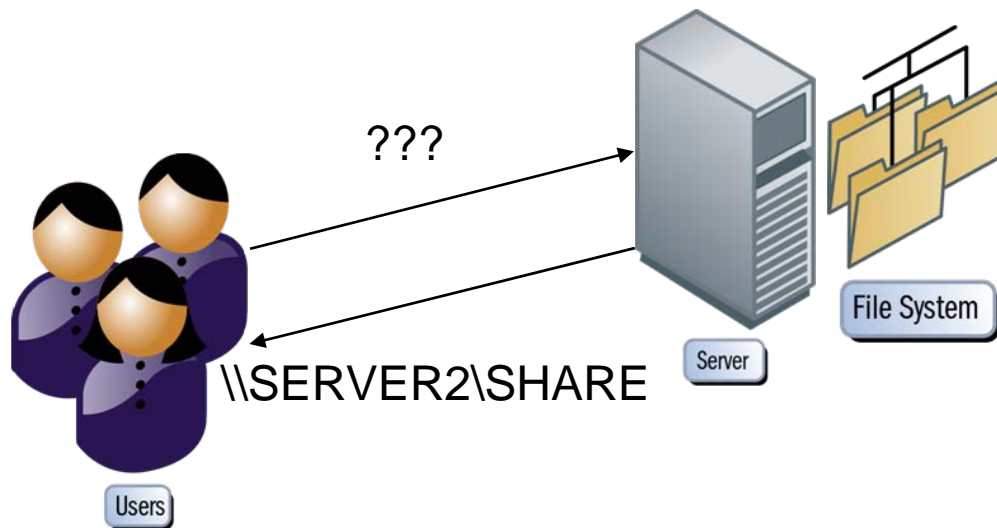
- Client requests a filesystem object that is a DFS link



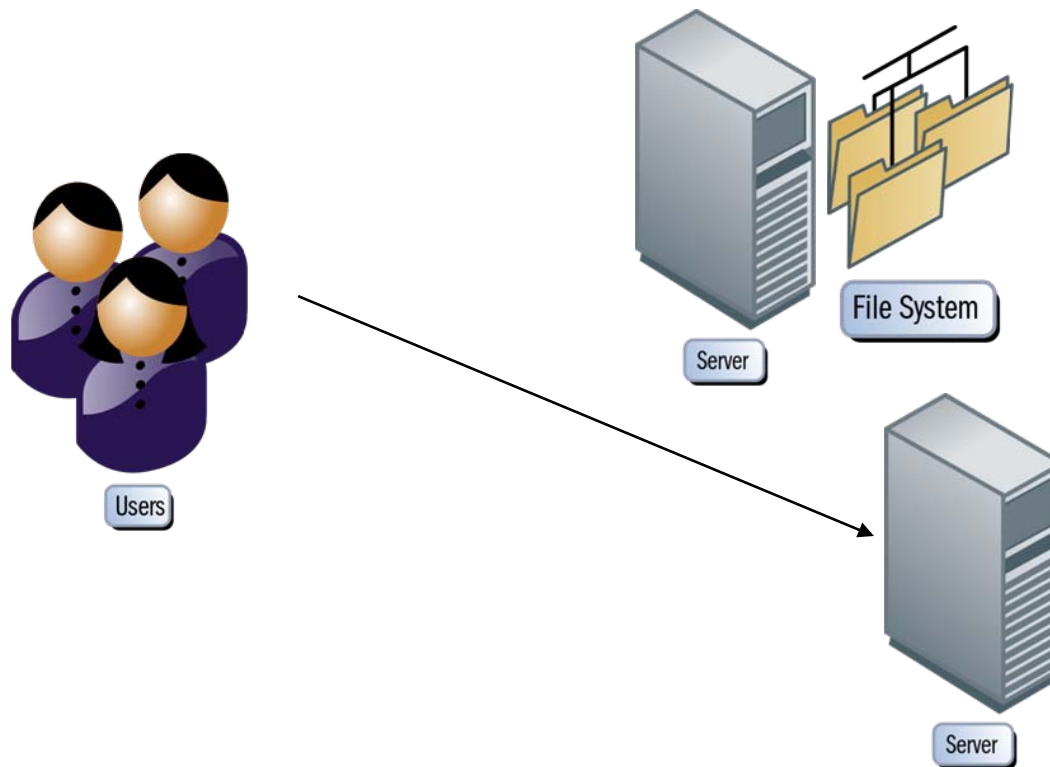
- Server replies with
`NT_STATUS_PATH_NOT_COVERED`



- Client requests a DFS referral for the object



- Client establishes a new session to the target of the DFS referral



Core DFS Overview

- The client contacts the target, never the first server
- The target may be one or more UNC paths
- Multiple paths allows for crude load balancing and fault tolerance
- The client caches the DFS redirect information for an administrator-determined amount of time or until rebooted

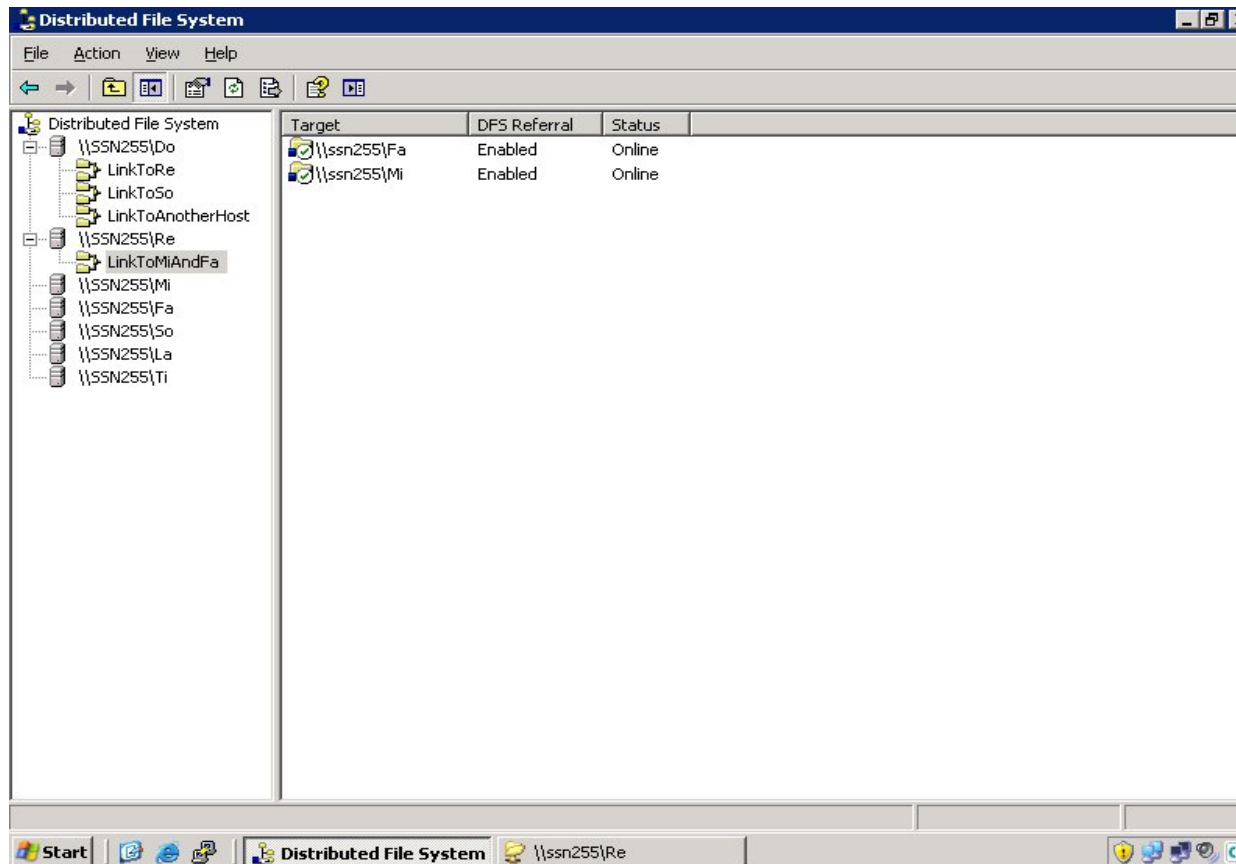
Other DFS-Related Features

- Management protocol used by MMC and other tools
- Active Directory DFS roots
- File Replication Service (FRS) and other replication
- Proxy redirects

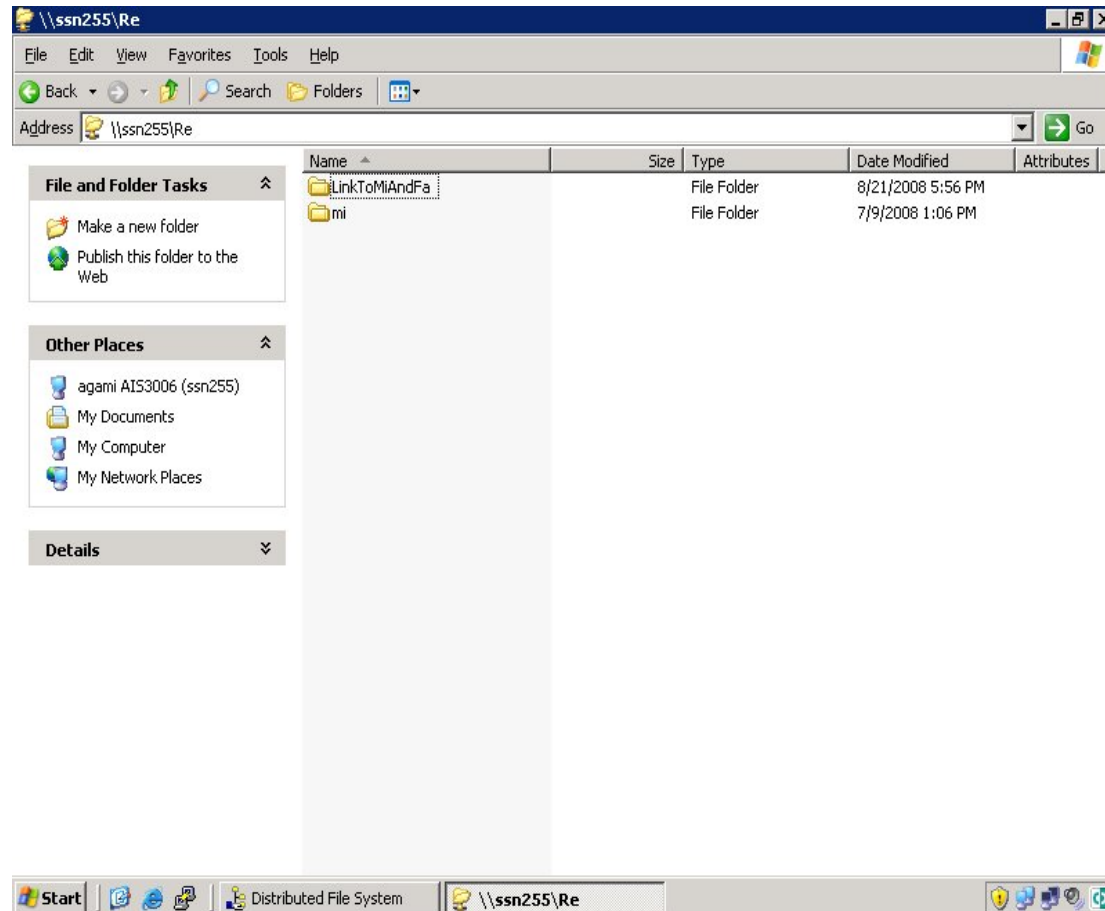
Management Protocol

- Allows remote DFS administration
- Used by MMC snapins and DFSCMD.EXE
- Allows creation and modification of:
 - ◆ DFS roots – shares containing DFS redirects
 - ◆ DFS links/redirects and their targets

➤ Management using MMC snapin



➤ Users' view of DFS redirect on DFS root



Active Directory DFS roots

- It is also possible to have DFS links stored in Active Directory, instead of on the storage devices
- Needs client Active Directory support
- Allows redundancy at the DFS root level more easily than standalone DFS

- Despite claims to the contrary, DFS does not handle replication
- It is often deployed with other replication mechanisms, such as FRS or the newer DFS-R

Proxy redirects

- It is also possible to create DFS redirects at the share level
- Users connecting to such a DFS root share will immediately be redirected to the target
- This can aid in the migration of data from one system to another before users' share mappings are updated
- This may also aid fault tolerance if multiple targets are specified

Practical Uses Of DFS

- Cost-effective hierarchical storage across multiple storage devices
- Assisting in transparent data migration

Hierarchical storage with DFS

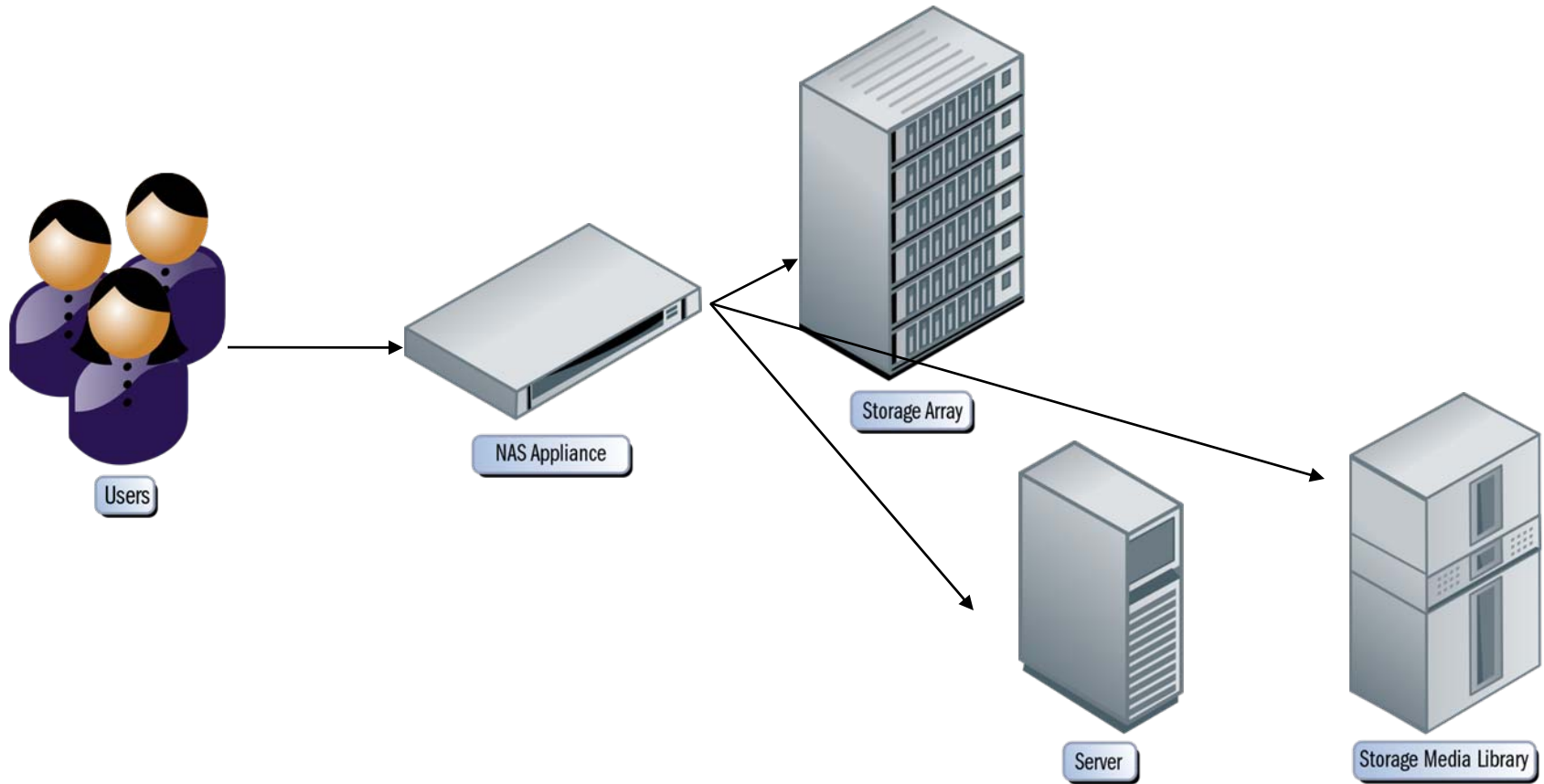
- Users see a single CIFS share
- Data may in fact be spread over multiple CIFS hosts
- DFS used to tie all data sets together into the single CIFS namespace seen by the users

Hierarchical storage with DFS

- Data might be distributed across a range of hosts like this:
 - ◆ *Infrequently* accessed data stored on lower-cost hardware. This hardware may even keep disks spun down when not being accessed
 - ◆ *More frequently* accessed data stored on reasonably-performing, but somewhat more expensive storage
 - ◆ *Performance critical* data might be stored on a host utilising expensive, but high-performance solid state storage, such as flash

Hierarchical storage with DFS

➤ Hierarchical storage example

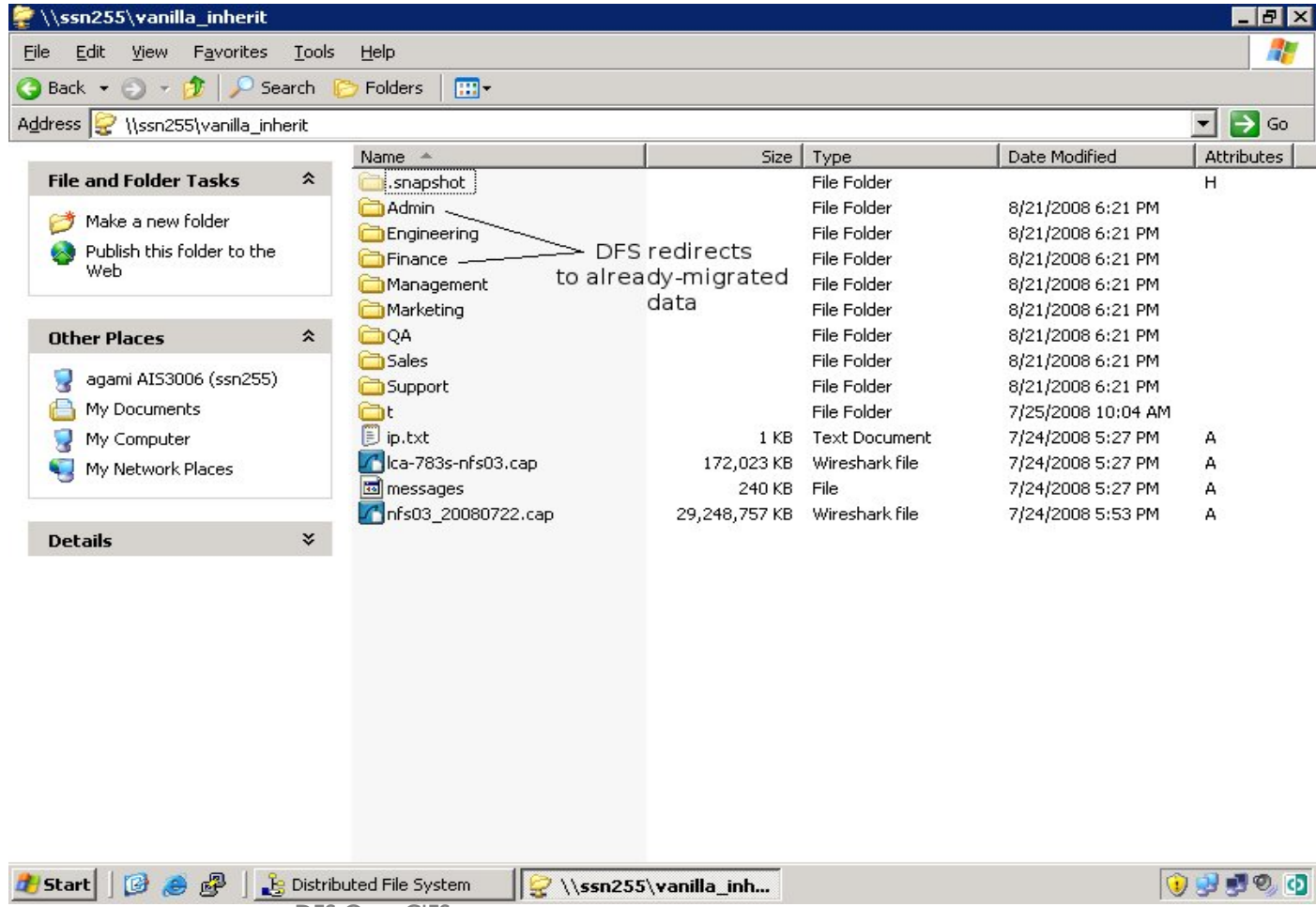


DFS To Aid Migration

- Data migration takes time and often requires significant downtime
- Data may be migrated a little at a time by:
 - ◆ Blocking access to a subset of the data by users
 - ◆ Migrating the inaccessible data
 - ◆ Replacing the source copy with a DFS redirect to the migrated copy
 - ◆ Allow access by users
 - ◆ Lather, Rinse, Repeat...
- Downtime is minimal

DFS To Aid Migration

➤ Migration may be done in stages using DFS



Conclusion

- DFS has been around long enough to be stable in most implementations
- DFS is a useful tool and, combined with some lateral thinking, can solve a number of common problems

- Please send any questions or comments on this presentation to SNIA: trackfilemgmt@snia.org

**Many thanks to the following individuals
for their contributions to this tutorial.**

- SNIA Education Committee

**Name of contributor here
Name of contributor here
Name of contributor here
Name of contributor here
Name of contributor here
Name of contributor here**

**Name of contributor here
Name of contributor here
Name of contributor here
Name of contributor here
Name of contributor here
Name of contributor here**