Chaffey College
Chino Information Technology Center

CISNTWK-11

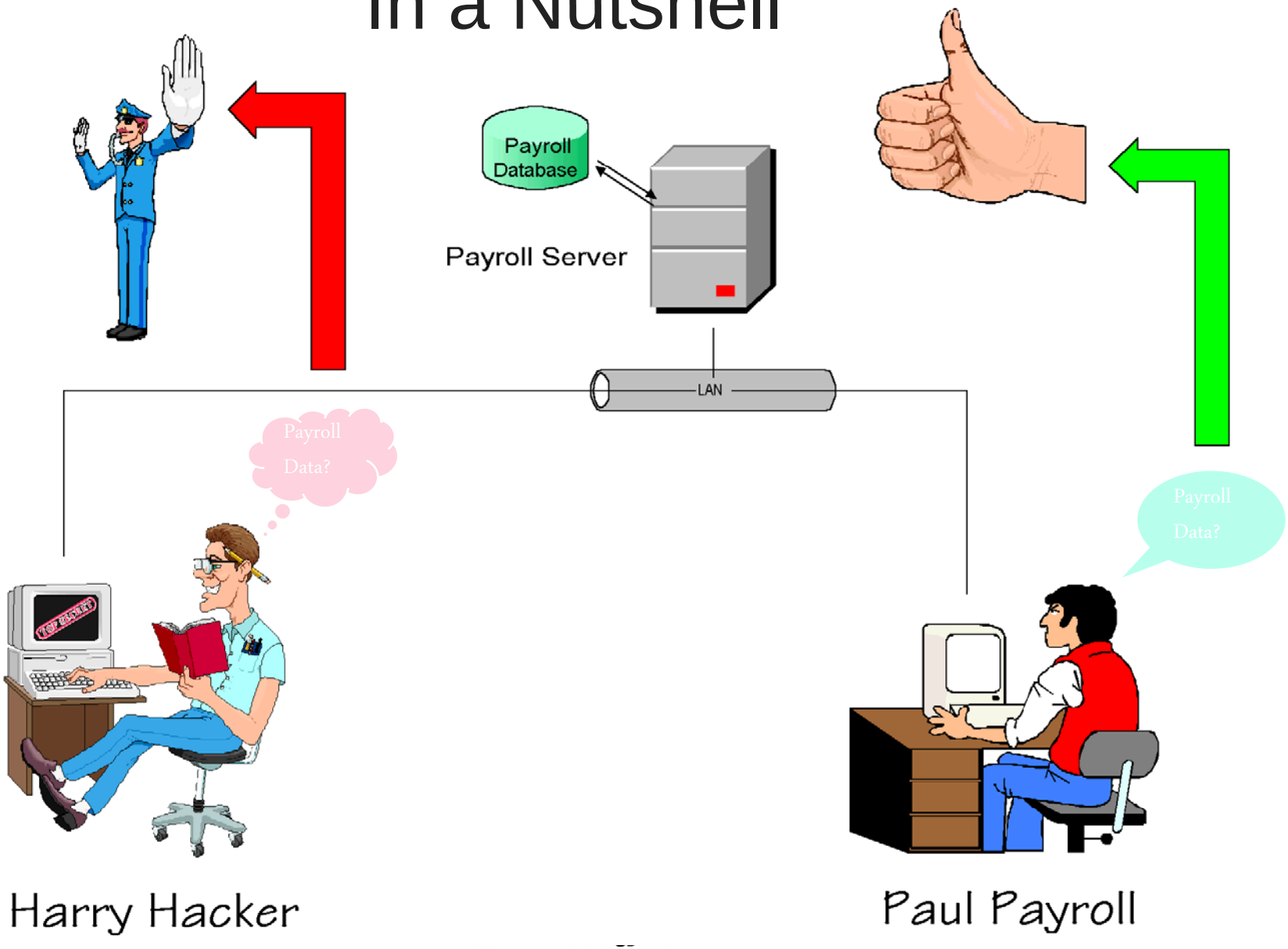
Microsoft Network Server

Chapter 5 Introduction

Permissions and Shares

Chaffey College
Chino Information Technology Center

In a Nutshell



Introduction

- A permission is a rule associated with an object, such as a
  - Share
  - Directory
  - File
- Objects have a security descriptor
  - Describes the security attributes for that object
- Access Control List (ACL)
  - Part of the security descriptor
  - Enumerates the protection (permissions) applied to that object

Terminology

- Access Control List (ACL)
  - A list of security protections that applies to an entire object
  - Enumerates the protection applied to an object
  - Made up of zero or more Access Control Entries
  - When describing access permissions, it is referred to as a “Discretionary ACL”
  - Each object contains an Owner Security ID (SID)

Terminology

- Access Control Entry (ACE)
  - An entry in an ACL
  - Each ACE contains the following pair of items:
    - a User SID or Group SID
    - a set of access rights that either allows or denies access

Terminology (continued)

- Common object types with permissions
  - NTFS Directories and files
  - Shares
  - Printers (print queuing)
  - Registry keys and values
  - Active Directory objects (Windows 2000 Server and later)
- Ownership

Terminology (continued)

- All objects have an owner
  - exactly one owner
- By default the owner is set to the creator of that object
  - for Windows NT 4 and later (except Windows XP)
    - the owner of the object will be set to “Administrators” for any object that is created by the Administrator account
- The owner can always change the permissions for any object he/she owns
- Ownership can be “transferred” by a user that has permissions to do so

Permissions for File Systems

- File Allocation Table (FAT[-32]) and Permissions
  - Minimal permissions supported
    - supports Share Permissions
      - only applicable for remote connections
  - No protection from local users
- Windows NT File System (NTFS) and Permissions
  - Supports Share Permissions
  - Supports Directory and File Permissions
  - Protection for both local users and remote users
  - Protection can be explicitly specified for every object
- Bottom line - **NTFS offers better protection than does FAT**

User Rights versus Permissions

- User Rights (also known as privileges)
  - Are authorization for a user to perform actions on the system
  - Apply to the system as a whole
- Permissions
  - Are rules associated with a particular object and regulate access to the object
- User Rights override permissions on an object, if applicable

Standard Permissions

- Windows 2000 (and later) uses the following “standard” permissions for directories and files

Permission	Code	NTFS directory (folder) permissions	NTFS file permissions
Read	R	Display (see) the elements of the directory	View or read the file contents
Read and Execute	X	Navigate through this directory, or make this the “current” directory (Includes “R” and “L” permissions)	Execute (run) an application (Includes “R” permission)
Write	W	Create new files and directories, modify attributes of directory	Change, append, or truncate the file contents; modify attributes of file
List Folder Contents	L	Display (see) the elements of the directory (folder)	*Not applicable*
Modify	M	Delete the directory (Includes “W” and “X” permissions)	Delete the file (Includes “W” and “X” permissions)
Full Control	F	Change permissions of directory Take ownership of the directory (Includes all NTFS permissions)	Change permissions of file Take ownership of the file (Includes all NTFS permissions)

Basic Permissions

- Windows 2000 (and later) uses the following “atomic” permissions for directories and files

Permission for Directory	Permission for File	F	M	X	L	R	W
Traverse Folder	Execute File	X	X	X	X		
List Folder	Read Data	X	X	X	X	X	
Read Attributes	Read Attributes	X	X	X	X	X	
Read Extended Attributes [1]	Read Extended Attributes [1]	X	X	X	X	X	
Create Files	Write Data	X	X				X
Create Folders	Append Data	X	X				X
Write Attributes	Write Attributes	X	X				X
Write Extended Attributes [1]	Write Extended Attributes [1]	X	X				X
Delete Subfolders and Files [2]	*Not applicable*	X					
Delete [3]	Delete	X	X				
Read Permissions	Read Permissions	X	X	X	X	X	X
Change Permissions	Change Permissions	X					
Take Ownership	Take Ownership	X					

Equivalent Standard Permissions Code

1

2

3

Implicit Groups

- The following table lists the Implicit Groups

  color indicates the Implicit Group is available with Windows 2000 (and later)

  - **Green** color indicates the Implicit Group is available with Windows Server 2008 R2 and Windows 7

Group Name	Active Directory Domain	Member Server or Workstation	Comment
Anonymous Logon	Yes	Yes	Anonymous users
Authenticated Users [1]	Yes	Yes	Users who have completed a valid logon
Batch	Yes	Yes	Users logged on from a batch queue facility
Console Logon	Yes	Yes	Users logged on to the console
Creator Group [2] [3]	Yes	Yes	The creator's primary group gets these permissions
Creator Owner [2]	Yes	Yes	The creator of the object gets these permissions
Dialup	Yes	Yes	Users logged on from a dial-up connection
Enterprise Domain Controllers	Yes	No	Domain Controllers with enterprise roles

Implicit Groups (continued)

- The following table lists the Implicit Groups
  - **Blue** color indicates the Implicit Group is available with Windows 2000 (and later)
  - **Orange** color indicates the Implicit Group is available with Windows Vista (and later)

Group Name	Active Directory Domain	Member Server or Workstation	Comment
Everyone	Yes	Yes	All users
Interactive	Yes	Yes	Users logged into the local computer
Network	Yes	Yes	Users connected remotely (using Shares)
Owner Rights	Yes	Yes	Restricts ability for owner to change the ACL
Proxy	Yes	No	Not used in Windows 2000 or Server 2003
Restricted code	Yes	No	Used by RunAs with the "Run this program with restricted access option" or “Protect my computer and data from unauthorized program activity”
Self	Yes	No	The object itself – allows the object to modify itself (available with Active Directory objects)

Implicit Groups (continued)

- The following table lists the Implicit Groups
  - **Blue** color indicates the Implicit Group is available with Windows 2000 (and later)
  - **Green** color indicates the Implicit Group is available with Windows Server 2008 R2 and Windows 7

Group Name	Active Directory Domain	Member Server or Workstation	Comment
Service	Yes	Yes	Service processes
System	Yes	Yes	The Operating System
Terminal Server User	No [1]	Yes	Users accessing system through Terminal Server
This Organization Certificate	No [1]	Yes	Used by Smart Card Authentication and Kerberos

Implicit Groups (continued)

- Windows Server 2003 (and later) includes the following Implicit Groups in a Domain Environment [1]
  - **Green** color indicates Implicit Group is supported on Windows 7, Windows Vista, Windows XP, and Windows Server 2008 outside of a Domain

Group Name	Comment
Digest Authentication	Session uses HTTP MD5 hash keys (Used by IIS Server)
Local Service	Special account used by Operating System (used by Services that only access the local computer)
Network Service	Special account used by Operating System (used by Services that need access to remote computers)
NTLM Authentication	Session uses Pre-Windows 2000 authentication
Other Organization	Session connected via a cross-forest trusts
Remote Interactive Logon	Session connected via Remote Desktop (RDP)
SChannel Authentication	Session uses a secure channel (Kerberos)
This Organization	Selected when “Other Organization” is not selected (user is local to the forest)

1

Implicit Groups (continued)

- Implicit groups are also known as “special identities” and “well-known security principals”
- The implicit group names can be used for assigning permissions (just like users and groups are used for assigning permissions)
- Membership in these implicit groups is automatic and based on the participating “role” of the user
  - The membership generally occurs when a user logs in or authenticates

Implicit Groups (continued)

  - An Administrator cannot “force” or override the membership of these implicit groups
- On Windows 2000 (and later), it is possible for an Administrator to assign these implicit groups ...
  - to a “local group” on a non-Domain computer (member Server or Professional)
  - in the “security” tab of Users, Groups, and Computers in the MMC “Active Directory Users and Computers”
- On Windows NT 4, the implicit group names are not visible in “User Manager” or “User Manager for Domains”

Permissions Usage

- Permissions are attached to Users and Groups
- Granted permissions for an object are cumulative
- Permissions can be explicitly denied
  - Permissions that are denied take precedence over permissions that are granted
- Standard permission sets are a shortcut to applying multiple basic permissions
  - Most (common) permissions can be specified with the standard permissions
  - Use basic permissions when “finer granularity” is required

Permissions Evaluation

- Users can access a directory or file only if they have been granted permission
- Permissions for objects are evaluated as follows:
  - If any applicable ACE entry assigns “Deny” or “No Access” to the user desiring access, the user is denied access to the object
  - Otherwise, the user is granted the sum of the permissions of all ACEs in the ACL which apply to him/her
  - If there are no entries that apply to the user, he/she is denied access

Permissions Evaluation

- Additional details
  - The owner of an object can always change (and look at) the permissions for that object
    - unless the “Owner Rights” implicit group has been defined on the ACL
  - The Administrator (or users belonging to the Administrators group) can always take ownership of an object
    - this allows them to change (and look at) the permissions for **all** objects
    - an “Administrator” cannot be blocked from access to objects

Inheritance of Permissions

- New objects inherit (obtain) permissions from their parent directory
  - Note that an object with an explicit ACL takes precedence over an inherited ACE
- ACLs on container objects (directories) can be configured to propagate to subordinate (child) directories and files
  - Administrators place ACLs on “key” directories, letting the permissions on these directories “flow” to files and subdirectories within these “key” directories
  - This simplifies administration of ACLs
  - This is the default behavior

Inheritance of Permissions

- A special (Implicit) Group named “Creator Owner” or “Creator Group” can be used in a directory ACE
  - It augments permissions for new objects created within that directory
- The difference between the standard permissions “*Read and Execute*” and “*List Folder Contents*” when applied to directories is
  - *Read and Execute* is inherited by both subdirectories and files
  - *List Folder Contents* is inherited by subdirectories only, but not files
    - this permission will be visible on subdirectories only

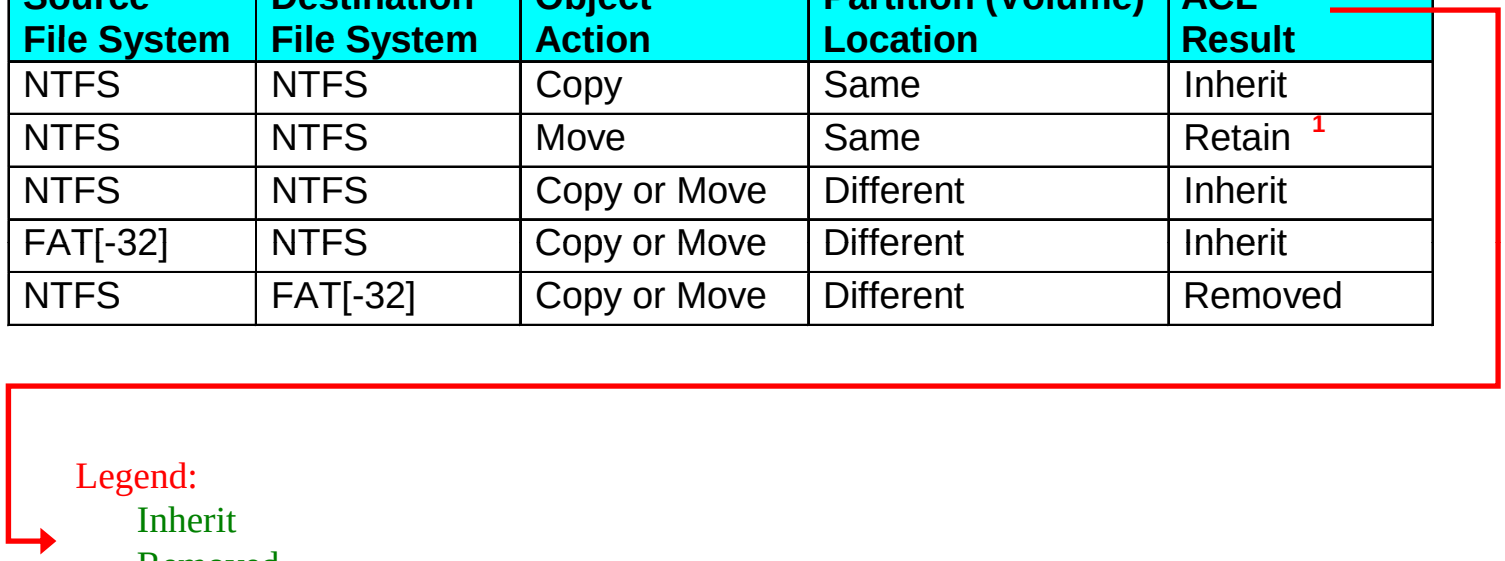
Inheritance of Permissions (continued)

- When an ACE is placed on a directory, the following scope of the entry's inheritance can be specified

ACE Apply Onto	Comments
This folder only	No inheritance occurs
This folder, subfolders, and files	All objects inherit this ACE **(this is the default)**
This folder and subfolders	Directories only – not files
This folder and files	Inheritance occurs only to files in this directory
Subfolders and files only	All objects contained in subdirectories inherit this ACE, but not to files in **this** directory
Subfolders only	Subdirectories only – not files
Files only	Files only – not subdirectories Subdirectories do not receive this ACE
Apply these permissions to object and/or containers within this container only	This limits inheritance only to those sub-objects that are immediately subordinate to this directory

Affect on ACLs With Copy/Move

Source File System	Destination File System	Object Action	Destination Partition (Volume) Location	ACL Result
NTFS	NTFS	Copy	Same	Inherit
NTFS	NTFS	Move	Same	Retain [1]
NTFS	NTFS	Copy or Move	Different	Inherit
FAT[-32]	NTFS	Copy or Move	Different	Inherit
NTFS	FAT[-32]	Copy or Move	Different	Removed



- The following can be used to insure that the ACL will remain intact if the object's source and destination file system is NTFS
  - Use the command "XCOPY /O" on Windows 2000 (and later)
  - Use the SCOPY utility from the NT 4 resource kit

Notes on Permissions

- The "*Full Control*" permission when applied to directories includes the basic permission called "*Delete Subfolders and Files*" (also known as "File Delete Child")
  - This permission permits a user to delete files in that directory, even if he/she does not have "delete" permission to the file(s)
  - This feature is necessary for POSIX compliance (for Unix interoperability)
- Deleting a User or Group does not automatically delete the corresponding references on an ACE
  - Adding the same User or Group back again will not resurrect the corresponding ACE entry(s)

Notes on Permissions

- Ownership is taken, not given
  - User A cannot change owner to User B; User B takes ownership from User A
  - This keeps everyone accountable
  - This is not absolutely true – see next slide

Permissions Evaluation Examples

- Assume the following permissions are established on an object
  - User Fred:Write
  - Group X: Read and Execute
  - Group Y: Delete (basic permission)
  - Group Z: Deny all access
- If User Fred is a member of Groups X and Y
  - User Fred's effective rights to this object: "**Read+Execute+Write+Delete**"
- If User Fred is a member of Groups X, Y, and Z
  - User Fred's effective rights to this object: "**No Access**"
- If User Mary is a member of Group W
  - User Mary's effective rights to this object: "**No Access**"
- If User Mary is a member of Groups X and Y
  - User Mary's effective rights to this object: "**Read+Execute+Delete**"

Permissions Evaluation Examples (continued)

- Assume the following permissions are established on an object, with the object located on a computer named Dilbert
  - Group X: Delete (basic permission)
  - Implicit Group “Everyone”: Read and Execute
  - Implicit Group “Interactive”: Write
  - Implicit Group “Network”: Deny all access
- If User Fred is a member of Group X, accessing the object from computer Dilbert (he’s logged in interactively on Dilbert)
  - User Fred’s effective rights to this object: “**Read+Execute+Write+Delete**”
- If User Mary is a member of Group W, accessing the object from computer Dilbert (she’s logged in interactively on Dilbert)
  - User Mary’s effective rights to this object: “**Read+Execute+Write**”
- If User Fred is a member of Group X, accessing the object from computer Wally (he’s logged in interactively on Wally)
  - User Fred’s effective rights to this object: “**No Access**”

Shares

- Shares
  - Are used to export a directory structure to remote users
  - Are required for all remote access, including Windows NT family
  - Apply to a directory
  - The share permissions apply to all subdirectories and files in the share
- The following permissions are supported for shares:
  - ***No Access*** [1] Access is denied to everything within the share
  - ***Read*** Allows users to read files and execute program files
    Allows users to view file and directory names and attributes
    Allows users to navigate through directories
  - ***Change*** Includes *Read* permission
    Users can write files, modify files, and change attributes on files and directories
    Users can create and delete files and directories
  - ***Full Control*** Includes *Change* permission
    Users can modify permissions on the share
    Users can take ownership of files and directories [2]

1

2

Shares (continued)

- To access a share
  - Use the “net use <DriveLetter:> \\ComputerName\ShareName” command
  - Use the Universal/Uniform Naming Convention (UNC) in the directory path:
  - Use “Network” to locate the computer and share on Windows 7 and Windows Vista
    - note that Network Discovery will need to be enabled for this to work
  - Use “My Network Places” to locate the computer and share on Windows 2003,
    Windows XP, and Windows 2000
  - Use “Network Neighborhood” to locate the computer and share on Windows NT 4

Shares (continued)

- A share that is located on a Windows Server has less restrictions than when the share is located on a non-server version of the NT family
  - On Windows 7 Professional/Vista Business/XP Professional/2000 Professional/NT Workstation, a maximum of 10 concurrent connections are supported to a share
  - On the Server product family, there are no concurrent limitations
- The actual access rights for a share object on an NTFS partition is the intersection (logical AND) of the share permissions and the NTFS permissions
  - Example: If the NTFS permission is *"Read"* and the share permission is
    "*Full Control",* then the effective rights will be *"Read"* (the most restrictive rights)

Built-in Shares

- The following shares exist on Windows NT family systems
  - Almost all of them are hidden shares
  - The Administrative Shares can be removed with a Policy (Registry setting)

Share Name	Share Location	Created on	Comments
ADMIN$	%SystemRoot%	All Systems	Administrative Share [1]
{DriveLetter}$ [2]	{DriveLetter}:\	All Systems	Administrative Share [1]
IPC$	*Not Applicable*	All Systems	Remote IPC (used for networking)
NETLOGON	%SystemRoot%\sysvol\sysvol\<Domain>\Scripts [3] %SystemRoot%\System32\Repl\Import\Scripts [4]	Domain Controllers	Logon Server Share
PRINT$ [5]	%SystemRoot%\System32\Spool\Drivers	Print Servers	Shared Printer Drivers
REPL$	%SystemRoot%\System32\Repl\Export	NT 4 Servers	For Directory Replication
SYSVOL	%SystemRoot%\syvol\sysvol	Active Directory Domain Controllers	Used by Active Directory

Accessing Administrative Shares

- Users that are members of the Administrators Group cannot (by default) connect to Administrative shares on a system running Windows 7 or
  Windows Vista if User Account Control is enabled on the remote system
  - When the system has not been joined to a Domain
    - this feature is known as “remote restrictions”
  - This behavior is to prevent malicious users/software from bypassing the User Account Control (UAC) elevation prompt by simply connecting to an administrative share on that system (known as a loopback attack)
    - C$, D$, Admin$, etc.
  - Note that (by default) the Administrator user account is exempt from this behavior
    - also note that the Administrator user account (by default) is disabled on Windows 7 and Windows Vista

Accessing Shares as an “Administrator”

- The “Administrators” Group does not “flow through” (by default) when connecting to shares on a system running Windows 7 or Windows Vista if User Account Control is enabled on the remote system
  - When the system has not been joined to a Domain
    - this feature is known as “remote restrictions”
  - Instead, the “Users” Group is used (substituted) in place of the “Administrators” Group on the “remote” system
    - this will happen even if the user is not a member of the “Users” Group
  - This behavior is to prevent malicious users/software from bypassing the User Account Control (UAC) elevation prompt by simply connecting to a share with “administrative” access (known as a loopback attack)
  - Note that (by default) the Administrator user account is exempt from this behavior
    - also note that the Administrator user account (by default) is disabled on Windows 7 and Windows Vista

Guidelines For Using Permissions

- Whenever possible, assign permissions to Groups instead of Users
- Apply NTFS permissions before sharing a directory
- Share permissions can be more relaxed when NTFS permissions are used
- For Server, create other file systems for sharing
  - Avoid using the “system partition” or “boot partition” for this purpose
- For applications, it is best to grant Read and Execute to the executables
- It is generally best to separate applications from data
- For shared data directories - use the following general scheme:
  - Assign the Authenticated Users group “Read” and “Create Files” basic permissions
  - Assign the CREATOR OWNER “Full Control” permission
  - Create the directory using an Administrator account

Guidelines For Using Permissions (continued)

- Permissions (by default) are created too permissive on Windows NT and Windows 2000
  - New File Systems formatted with NTFS may be too permissive
    - EVERYONE has “Full Control” at the root of the drive
  - New Shares may be too permissive
    - EVERYONE has “Full Control”

Guidelines For Using Permissions (continued)

- Permissions (by default) are created more restrictive on Windows XP and Windows Server 2003
  - New File Systems formatted with NTFS are more restrictive
    - EVERYONE has “List Folder Contents” only at the root of the drive
    - the USERS built-in group has the following permissions on the drive
      - Read & Execute
      - List Folder Contents
      - Read
      - Create Directories
        - » create files and subdirectories in the directories that a user creates
  - New Shares are more restrictive
    - EVERYONE has “Read”

Guidelines For Using Permissions (continued)

- Permissions (by default) are created more restrictive on Windows Vista (and later)
  - New File Systems formatted with NTFS are more restrictive
    - the ADMINISTRATORS built-in group has “Full Control” permissions on the drive
    - AUTHENTICATED USERS has Modify permission on the drive
    - the USERS built-in group has “Read & Execute” permissions on the drive
  - New Shares are more restrictive
    - EVERYONE has “Read”

Managing Permissions

- Using the Graphical User Interface (GUI)
  - From Windows Explorer, right click on the object, then select (depending upon your version of Windows)
    - Properties -> Security (for NTFS file and directory permissions)
    - Sharing and Security -> Security (for NTFS file and directory permissions)
  - NT 4 Server includes an Administrative Wizard called "Managing File and Folder Access"
    - Start Menu -> Programs -> Administrative Tools -> Administrative Wizards

Managing Permissions

- On Windows XP Professional computers that are not joined to a Domain
  - Access to the Security tab (NTFS permissions) is disabled
  - You can enable access to the Security tab using the following steps
    - "Control Panel -> Folder Options -> View"
      - uncheck "**Use simple file sharing [Recommended]**"
  - Note that this default behavior is intended to make Windows XP Professional behave like Windows XP Home Edition

Managing Permissions (continued)

- The following command line tools can be used to manage permissions
  - Note that the Resource Kit and Support Tools from Windows Server 2003 can be used on Windows Server 2008, Windows 7, and Windows Vista

Name of Tool	Description	Included with	2008 / Win 7 / Vista	2003 / XP	2000	NT
CACLS	Change ACLs	Windows	X	X	X	X
FIXACLS [1]	Reset ACLs for the System/Boot Partition to their default	Resource Kit				X
ICACLS	Improved CACLS utility	Windows	X			
PERMS	Display access permissions for a specified user	Resource Kit		X	X	X
SHOWACCS	Show Access Control Lists	Support Tools		X	X	
SHOWACLS	Display Access Control Lists	Resource Kit		X	X	X
SUBINACL	An advanced ACL tool for managing object security	Resource Kit		X	X	X
XCACLS	An “enhanced” CACLS utility	Support Tools		X		
XCACLS	An “enhanced” CACLS utility	Resource Kit			X	X

Managing Shares

- Using the Graphical User Interface (GUI)
  - From Windows Explorer, right click on the directory, then select (depending upon your version of Windows)
    - Properties -> Sharing
    - Sharing and Security -> Sharing
    - Sharing...
    - Share…

Managing Shares

- From Windows 2000 (and later), use the “Computer Management” MMC
  - Start Menu -> Programs -> Administrative Tools -> Computer Management
  - expand “Shared Folders” within the “System Tools” tree
- From a Windows NT Server, use “Server Manager”
  - Start Menu -> Programs ->Administrative Tools -> Server Manager
  - select “Computer -> Shared Directories...”
  - “Server Manager” is also available on Windows 2000 Server
    - the program is named “SRVMGR” [1]

Managing Shares (continued)

- On Windows XP Professional computers that are not joined to a Domain
  - Access to the Sharing tab limits the file sharing options
    - you cannot control the specific permissions on shares
    - you cannot control specific users or groups accessing the shares
  - You can enable the full file sharing capabilities by using the following steps
    - "Control Panel -> Folder Options -> View"
      - uncheck "**Use simple file sharing [Recommended]**"
  - Note that this default behavior is intended to make Windows XP Professional behave like Windows XP Home Edition

Managing Shares (continued)

- On Windows 7, Windows Vista, and Windows Server 2008
  - By default, only a “simple” file sharing capability exists
    - the terminology (and permissions) is completely different than what is discussed in these slides
      - and from earlier versions of Windows
  - You can enable the “traditional” file sharing capability by using the following steps
    - “Control Panel -> Folder Options -> View”
      - uncheck “**Use Sharing Wizard [Recommended]**”
  - Note that this “feature” is even enabled (by default) on Windows Server 2008 Domain Controllers

Managing Shares (continued)

- Using the command line
  - Use the “NET SHARE” command
    - can be used to list, create, modify, and remove shares
  - The Windows 2003, Windows XP, Windows 2000, and Windows NT Resource kits include an additional tool [1]
    - **PERMCOPY** Copy ACLs between shares

Chaffey College
Chino Information Technology Center

CISNTWK-11

Microsoft Network Server

Chapter 5

Configuring, Managing, and Troubleshooting Resource Access

Chaffey College
Chino Information Technology Center

COURSE TECHNOLOGY
CENGAGE Learning™

Chapter 5
Configuring, Managing, and Troubleshooting Resource Access

Chaffey College
Chino Information Technology Center

Objectives

- Set up security for folders and files
- Configure shared folders and shared folder security
- Install and set up the Distributed File System
- Configure disk quotas
- Implement UNIX compatibility

Managing Folder and File Security

- Creating accounts and groups are the initial steps for sharing resources
  - The next steps are to create access control lists (ACLs) to secure these objects and then to set them up for sharing
- **Discretionary ACL (DACL)**
  - An ACL that is configured by a server administrator or owner of an object
- **System control ACL (SACL)**
  - Contains information used to audit the access to an object

Configuring Folder and File Attributes

- Attributes are stored as header information with each folder and file
  - Along with other characteristics including volume label, designation as a subfolder, date of creation, and time of creation
- Two basic attributes remain in NTFS that are still compatible with FAT
  - Read-only and hidden
- The advanced attributes are archive, index, compress, and encrypt

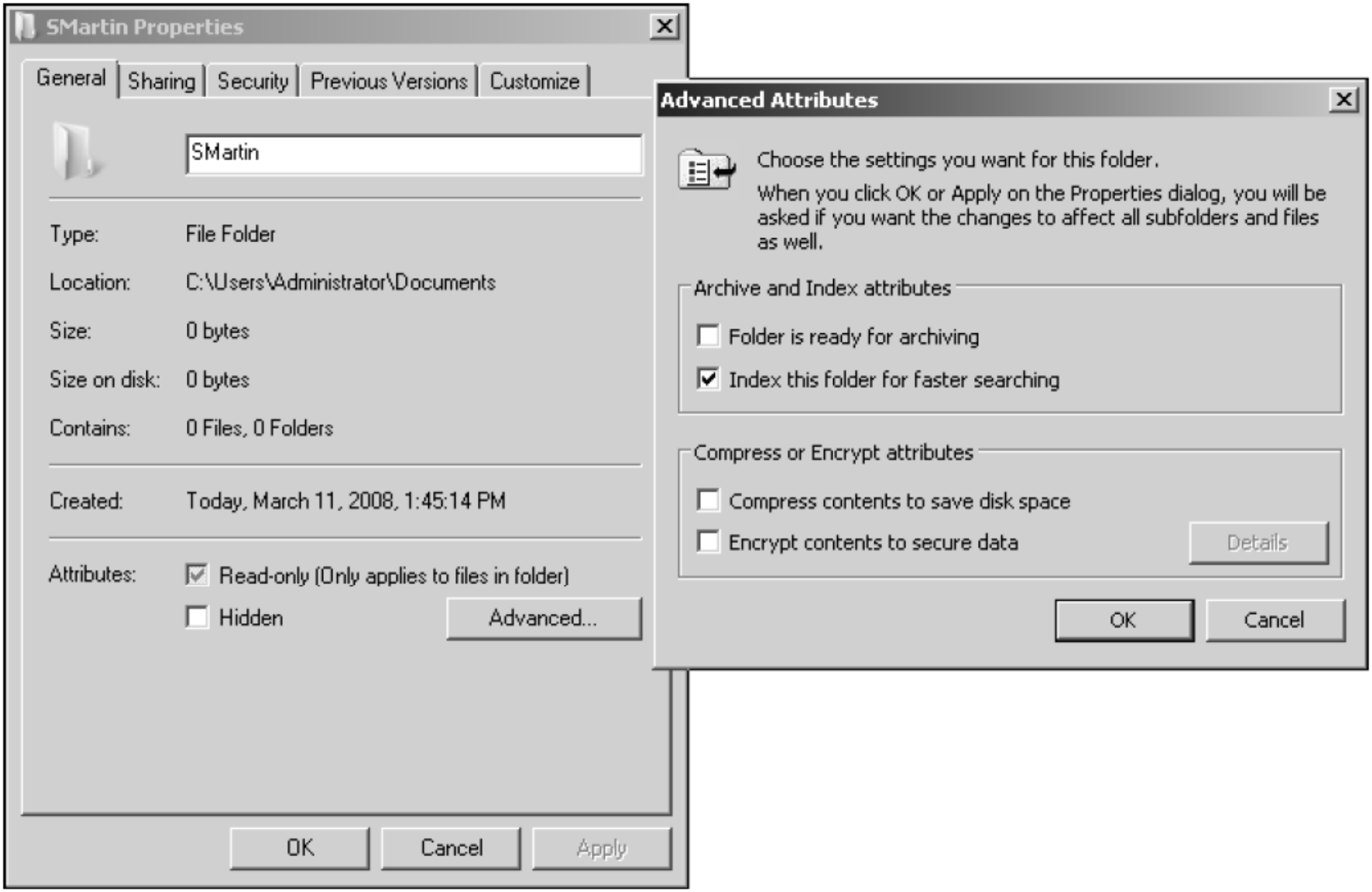


Figure 5-1 Attributes of a folder on an NTFS formatted disk

Configuring Folder and File Attributes (continued)

- Archive attribute
  - Indicates that the folder or file needs to be backed up because it is new or changed
  - File server backup systems can be set to detect files with the archive attribute to ensure those files are backed up
- Index attribute vs. Windows Search Service
  - The NTFS index attribute is used to index the folder and file contents so that file properties can be quickly searched in Windows Server 2008
    - Through the Indexing Service

Configuring Folder and File Attributes (continued)

- Index attribute vs. Windows Search Service (continued)
  - Windows Server 2008 offers a newer, faster search service called the Windows Search Service
  - To use the Windows Search Service, you must install the File Services role via Server Manager
- Multimaster replication
  - Each DC is equal to every other DC in that it contains the full range of information that composes Active Directory
- Active Directory is built to make replication efficient

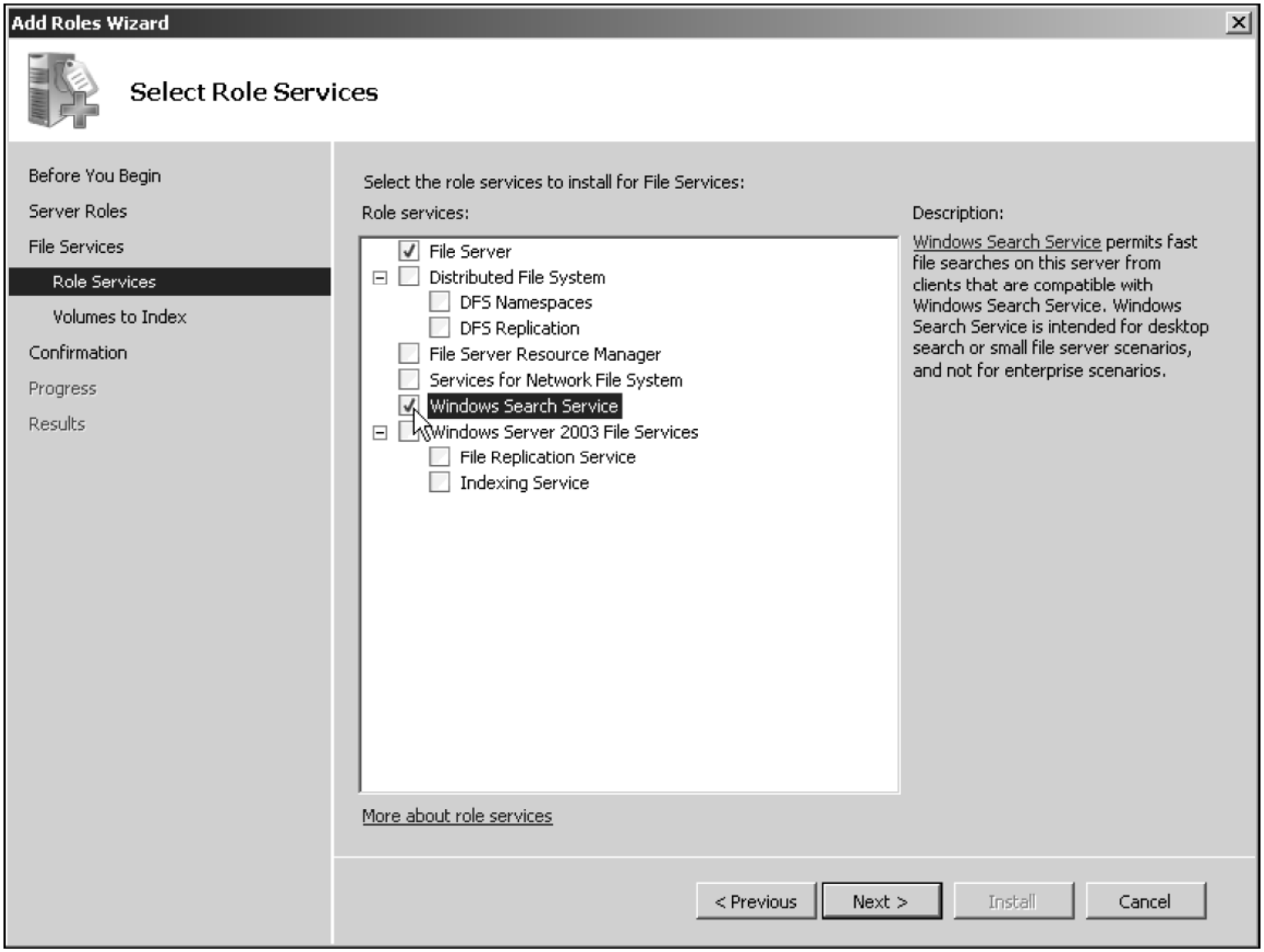


Figure 5-2 Installing the Windows Search Service with the File Services role

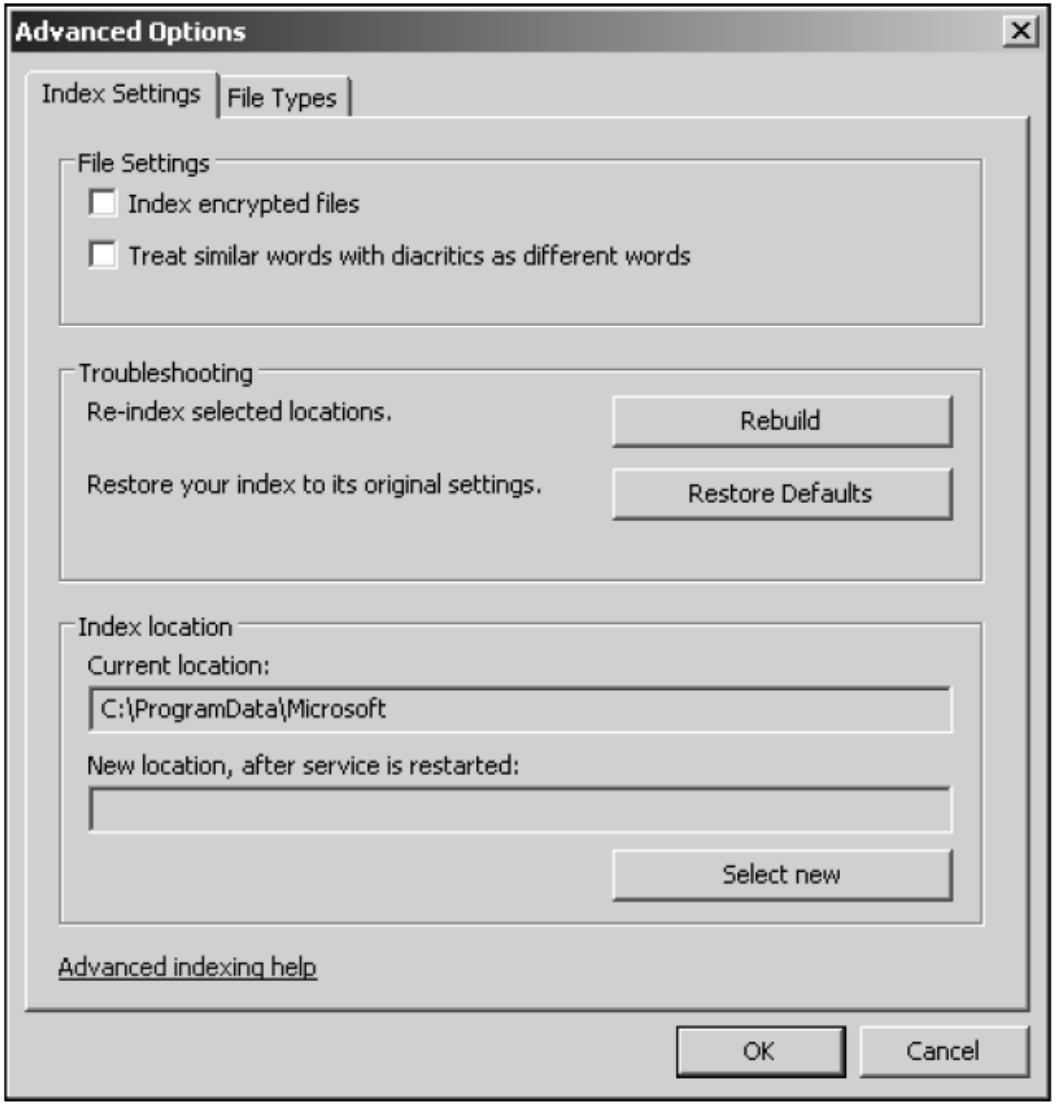


Figure 5-3 Configuring advanced indexing options

Configuring Folder and File Attributes (continued)

- Compress attribute
  - A folder and its contents can be stored on the disk in compressed format
  - Compression saves space and you can work on compressed files in the same way as on uncompressed files
  - Compressed files increase CPU overhead to open the files and to copy them

Configuring Folder and File Attributes (continued)

- Encrypt attribute
  - Protects folders and files so that only the user who encrypts the folder or file is able to read it
  - An encrypted folder or file uses the Microsoft **Encrypting File System (EFS)**
    - Which sets up a unique, private encryption key associated with the user account that encrypted the folder or file
  - EFS uses both symmetric and asymmetric encryption techniques

Configuring Folder and File Attributes (continued)

- Encrypt attribute (continued)
  - When you move an encrypted file to another folder on the same computer, that file remains encrypted, even if you rename it

Configuring Folder and File Attributes (continued)

- Activity 5-1: Encrypting Files
  - Time Required: Approximately 10 minutes
  - Objective: Encrypt files in a folder

Configuring Folder and File Permissions

- **Permissions**
  - Control access to an object, such as a folder or file
- When you configure a folder so that a domain local group has access to only read the contents of that folder
  - You are configuring permissions
- At the same time, you are configuring that folder's discretionary access control list (DACL) of security descriptors

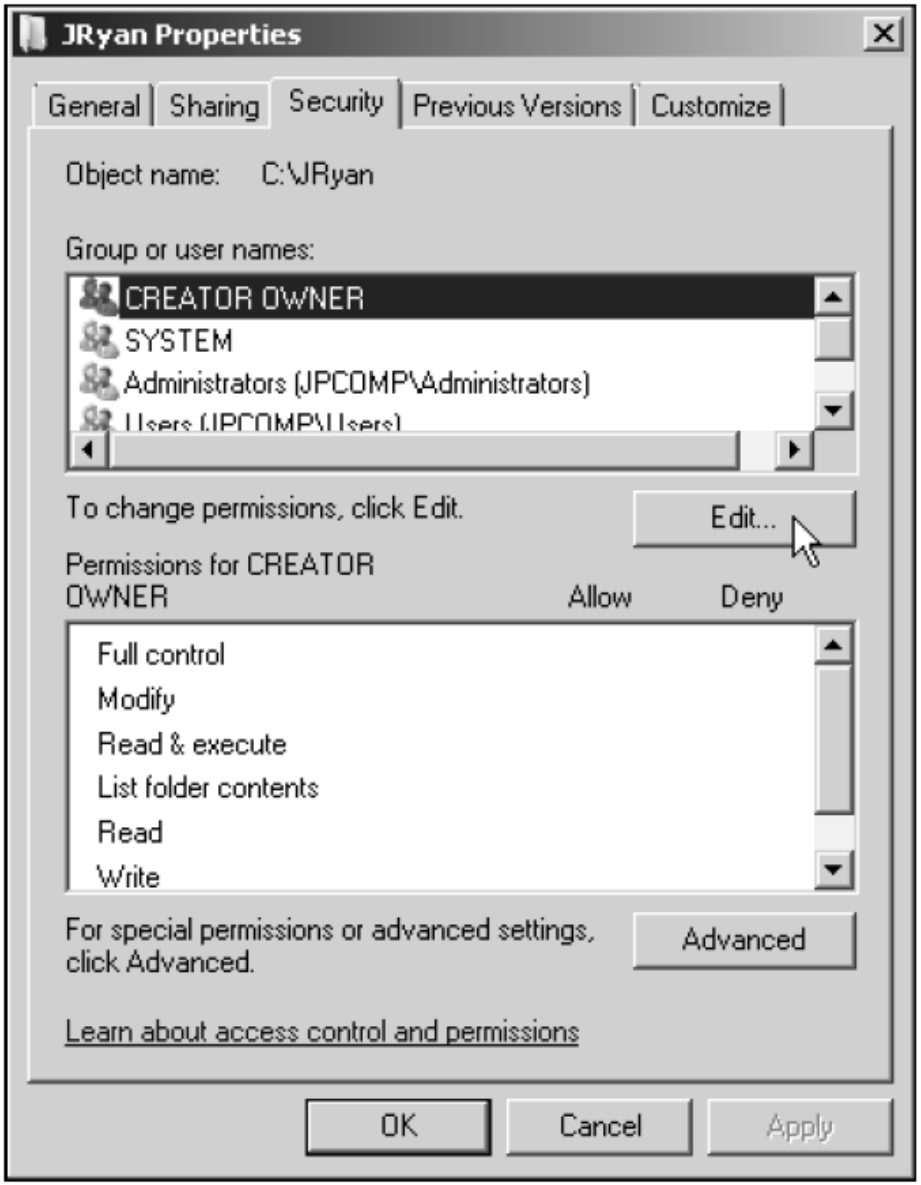


Figure 5-4 Configuring folder permissions

Configuring Folder and File Permissions (continued)

Table 5-1 NTFS folder and file permissions

Permission	Description	Applies to
Full control	Can read, add, delete, execute, and modify files plus change permissions and attributes, and take ownership	Folders and files
Modify	Can read, add, delete, execute, and modify files; cannot delete subfolders and their file contents, change permissions, or take ownership	Folders and files
Read & execute	Implies the capabilities of both List folder contents and Read (traverse folders, view file contents, view attributes and permissions, and execute files)	Folders and files
List folder contents	Can list (traverse) files in the folder or switch to a subfolder, view folder attributes and permissions, and execute files, but cannot view file contents	Folders only
Read	Can view file contents, view folder attributes and permissions, but cannot traverse folders or execute files	Folders and files
Write	Can create files, write data to files, append data to files, create folders, delete files (but not subfolders and their files), and modify folder and file attributes	Folders and files
Special permissions	Special permissions apply (see Table 5-2)	Folders and files

Configuring Folder and File Permissions (continued)

- Activity 5-2: Configuring Folder Permissions
  - Time Required: Approximately 10 minutes
  - Objective: Configure permissions on a folder so that users can modify its contents

Configuring Folder and File Permissions (continued)

- Activity 5-3: Removing Inherited Permissions
  - Time Required: Approximately 10 minutes
  - Objective: Remove inherited permissions on a folder

Configuring Folder and File Permissions (continued)

- If you need to customize permissions
  - You have the option to set up special permissions for a particular group or user

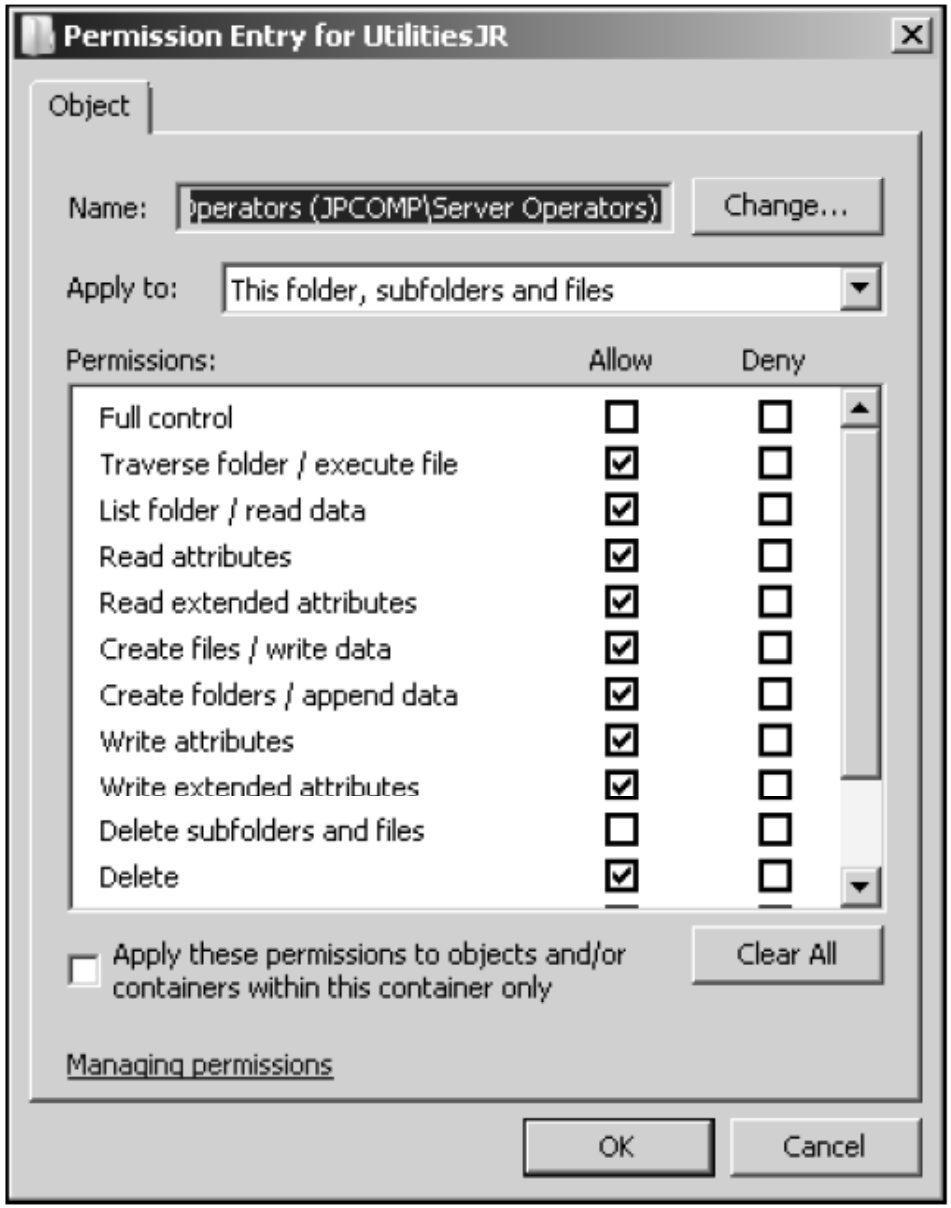


Figure 5-6 Special permissions

Table 5-2 NTFS folder and file special permissions

Permission	Description	Applies to
Full control	Can read, add, delete, execute, and modify files, plus change permissions and attributes, and take ownership	Folders and files
Traverse folder/execute file	Can list the contents of a folder and run program files in that folder; keep in mind that all users are automatically granted this permission via the Everyone and Users groups, unless it is removed or denied by you	Folders and files
List folder / read data	Can list the contents of folders and subfolders and read the contents of files	Folders and files
Read attributes	Can view folder and file attributes (read-only and hidden)	Folders and files
Read extended attributes	Enables the viewing of extended attributes (archive, index, compress, and encrypt)	Folders and files
Create files / write data	Can add new files to a folder and modify, append to, or write over file contents	Folders and files
Create folders / append data	Can add new folders and add new data at the end of files, but otherwise cannot delete, write over, or modify data	Folders and files
Write attributes	Can add or remove the read-only and hidden attributes	Folders and files
Write extended attributes	Can add or remove the archive, index, compress, and encrypt attributes	Folders and files
Delete subfolders and files	Can delete subfolders and files (the following Delete permission is not required)	Folders and files
Delete	Can delete the specific subfolder or file to which this permission is attached	Folders and files
Read permissions	Can view the permissions (ACL information) associated with a folder or file (but does not imply you can change them)	Folders and files
Change permissions	Can change the permissions associated with a folder or file	Folders and files
Take ownership	Can take ownership of the folder or file (read permissions and change permissions automatically accompany this permission)	Folders and files

Configuring Folder and File Permissions (continued)

- Activity 5-4: Configuring Special Permissions
  - Time Required: Approximately 15 minutes
  - Objective: Configure special permissions for a folder to grant a group expanded access

Configuring Folder and File Auditing

- **Auditing**
  - Enables you to track activity on a folder or file
- Windows Server 2008 NTFS folders and files
  - Enable you to audit a combination of any or all of the activities listed as special permissions in Table 5-2

Configuring Folder and File Auditing (continued)

- Activity 5-5: Auditing a Folder
  - Time Required: Approximately 10 minutes
  - Objective: Configure auditing on a folder to monitor how it is accessed and who is making changes to the folder

Configuring Folder and File Ownership

- With permissions and auditing set up, you might want to verify the ownership of a folder
- Folders are first owned by the account that creates them
- Folder owners have the ability to change permissions for the folders they create
- Ownership can be transferred only by having the Take ownership special permission
  - Or Full control permission (which includes Take ownership)

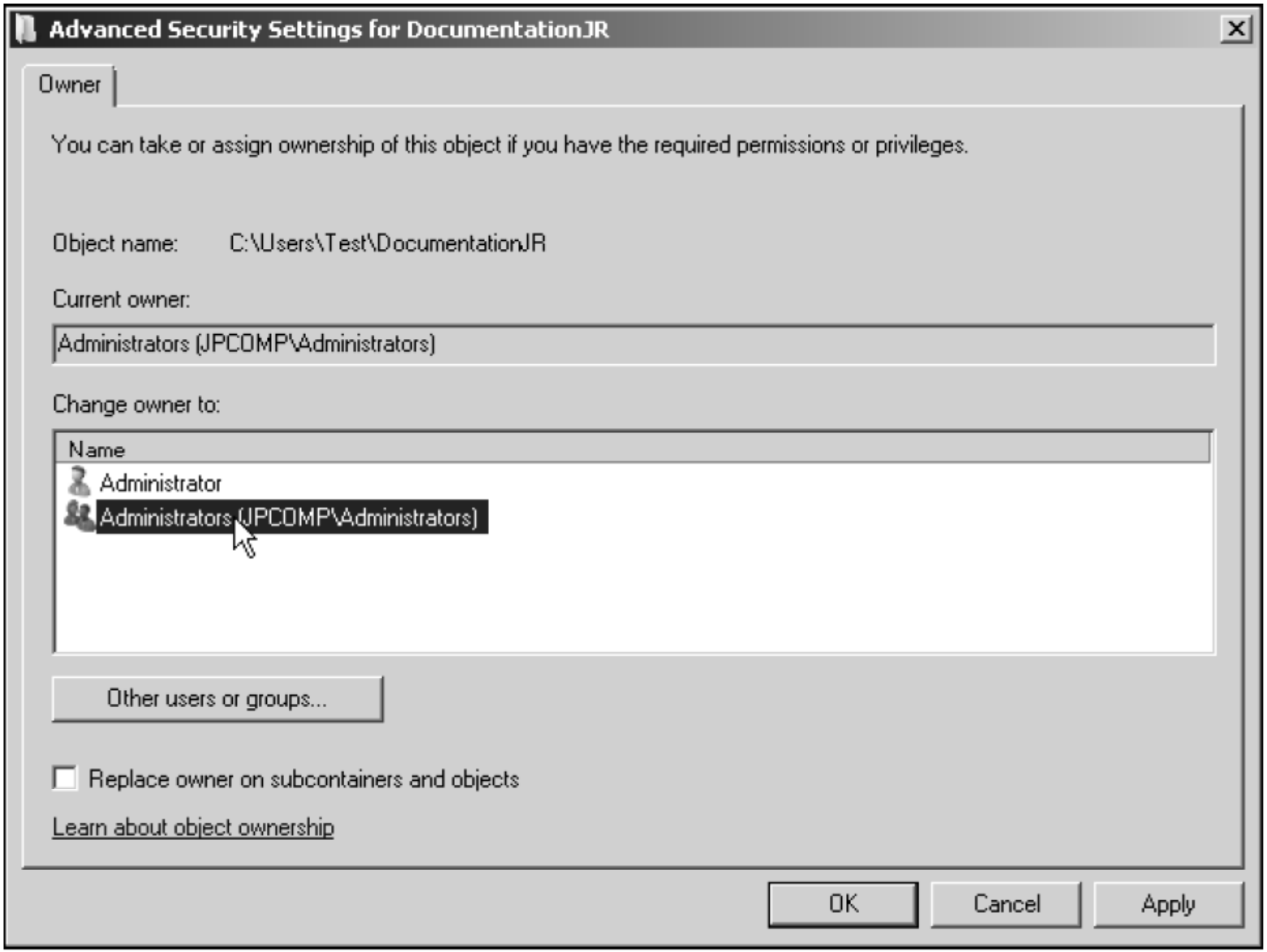


Figure 5-9 Taking ownership of a folder

Configuring Shared Folders and Shared Folder Permissions

- A folder can be set up as a shared folder for users to access over the network
- Configuring a shared folder is changed in Windows Server 2008 from previous versions
  - To help make the person offering the shared folder more aware of security options
- The first step for sharing a folder over the network is to turn on file sharing

Configuring Shared Folders and Shared Folder Permissions (continued)

- Activity 5-6: Enabling Sharing a Folder
  - Time Required: Approximately 5 minutes
  - Objective: Turn on file sharing and public folder sharing

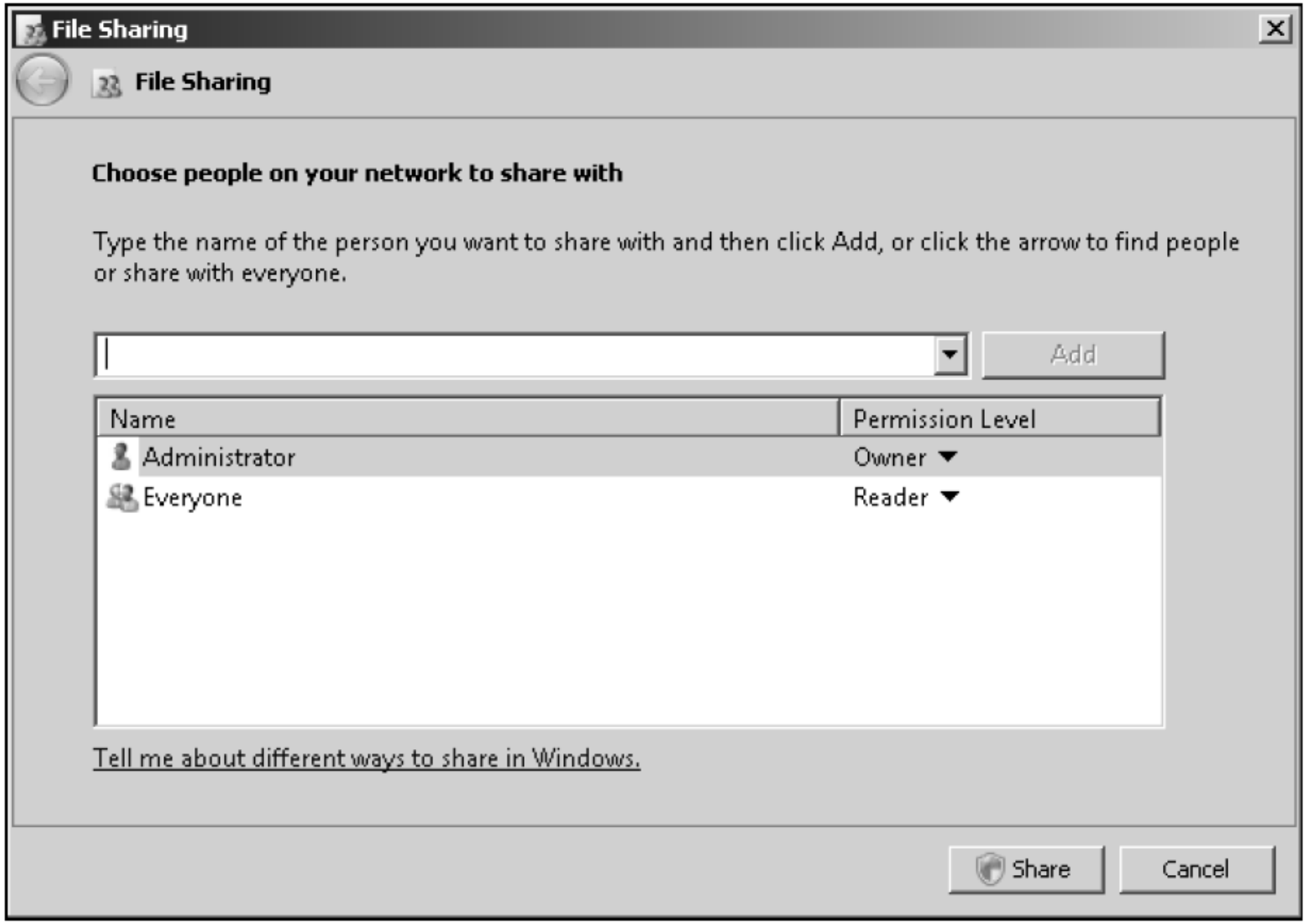


Figure 5-10 File Sharing dialog box

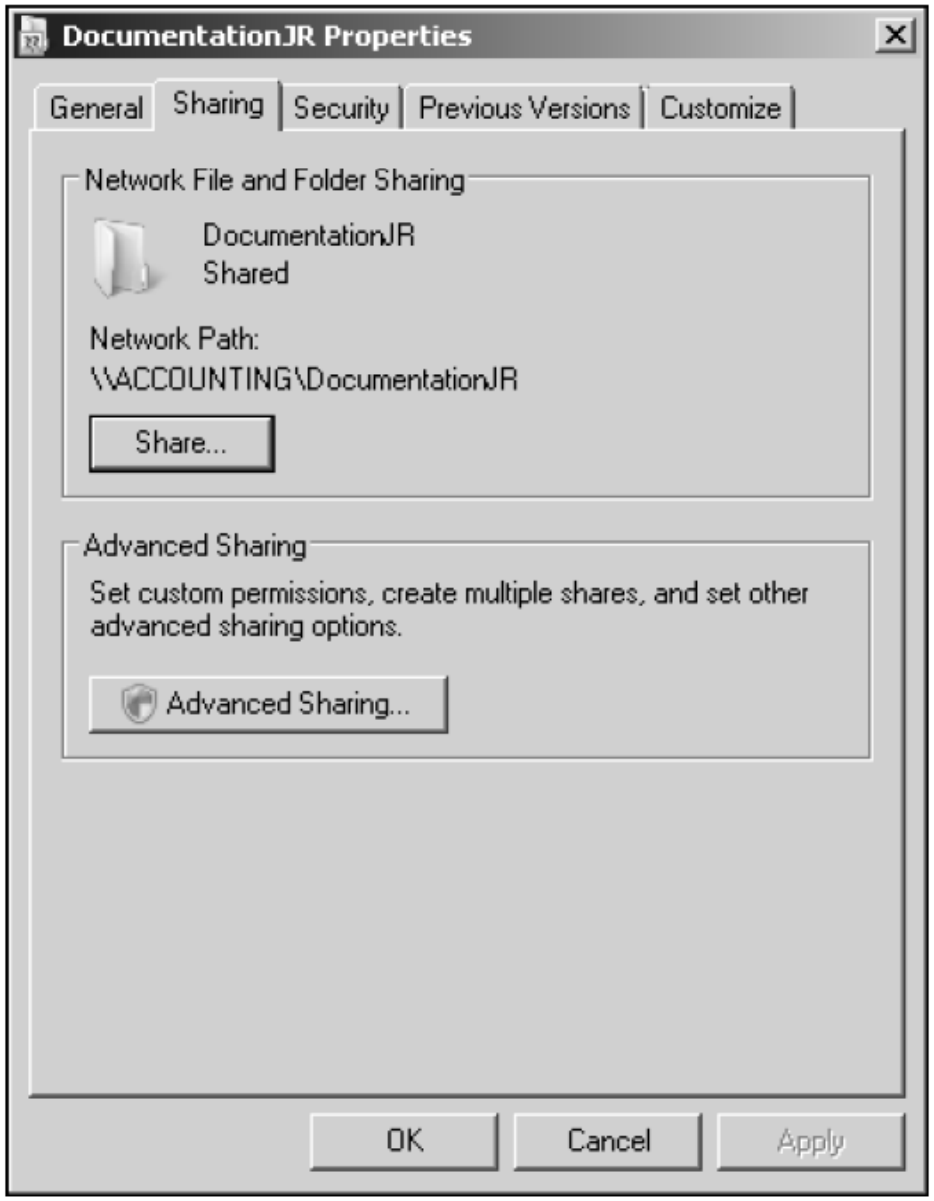


Figure 5-11 Sharing tab

Configuring Shared Folders and Shared Folder Permissions (continued)

- **Share permissions** for an object
  - Differ from the NTFS access permissions set through the Security tab
- The NTFS and share permissions are cumulative
  - With the exception of permissions that are denied
- Share permissions:
  - Reader
  - Contributor
  - Co-owner
  - Owner

Configuring Shared Folders and Shared Folder Permissions (continued)

- You can cache a folder to make the contents of a shared folder available offline
  - Any offline files that have been modified can be synchronized with the network versions of the files
- A folder can be cached in three ways:
  - Only the files and programs that users specify will be available offline
  - All files and programs that users open from the share will be automatically available offline
  - Files or programs from the share will not be available offline

Configuring Shared Folders and Shared Folder Permissions (continued)

- Activity 5-7: Configuring a Shared Folder
  - Time Required: Approximately 15 minutes
  - Objective: Configure a shared folder, share permissions, and offline access

Publishing a Shared Folder in Active Directory

- To **publish** an object
  - Means to make it available for users to access when they view Active Directory contents
    - Makes it easier to find when a user searches for that object
- **Directory Service Client (DSClient)**
  - Allows earlier Windows-based operating systems to search Active Directory
- When you publish an object, you can publish it to be shared for domain-wide access or to be shared and managed through an organizational unit (OU)

Publishing a Shared Folder in Active Directory (continued)

- Activity 5-8: Publishing a Shared Folder
  - Time Required: Approximately 5 minutes
  - Objective: Publish a shared folder in Active Directory

Troubleshooting a Security Conflict

- Windows Server 2008 offers the Effective Permissions tab in the properties of a folder or file
  - As a tool to help troubleshoot permissions conflicts
- Using the Effective Permissions tab, you can view the effective permissions assigned to a user or group
- Take into account what happens when a folder or files in a folder are copied or moved
  - A newly created file inherits the permissions already set up in a folder

Troubleshooting a Security Conflict (continued)

- Take into account what happens when a folder or files in a folder are copied or moved (continued)
  - A file that is copied from one folder to another on the same volume inherits the permissions of the folder to which it is copied
  - A file or folder that is moved from one folder to another on the same volume takes with it the permissions it had in the original folder
  - A file or folder that is moved or copied to a folder on a different volume inherits the permissions of the folder to which it is moved or copied

Troubleshooting a Security Conflict (continued)

- Take into account what happens when a folder or files in a folder are copied or moved (continued)
  - A file or folder that is moved or copied from an NTFS volume to a folder in a FAT volume is not protected by NTFS permissions
    - But it does inherit share permissions if they are assigned to the FAT folder
  - A file or folder that is moved or copied from a FAT volume to a folder in an NTFS volume inherits the permissions already assigned in the NTFS folder

Troubleshooting a Security Conflict (continued)

- Activity 5-9: Troubleshooting Permissions
  - Time Required: Approximately 10 minutes
  - Objective: View the effective permissions on a folder

Implementing a Distributed File System

- **Distributed File System (DFS)**
  - Enables you to simplify access to the shared folders on a network by setting up folders to appear as though they are accessed from only one place
  - DFS also makes managing folder access easier for server administrators
- If DFS is used in a domain, then shared folder contents can be replicated to one or more DCs or member servers

Implementing a Distributed File System (continued)

- DFS advantages:
  - Shared folders can be set up so that they appear in one hierarchy of folders
    - Enabling users to save time when searching for information
  - NTFS access permissions fully apply to DFS on NTFS-formatted volumes
  - Fault tolerance is an option by replicating shared folders on multiple servers
  - Access to shared folders can be distributed across many servers (**load balancing**)

Implementing a Distributed File System (continued)

- DFS advantages: (continued)
  - Access is improved to resources for Web-based Internet and intranet sites
  - Vital shared folders on multiple computers can be backed up from one set of master folders
- DFS reduces the number of calls to server administrators asking where to find a particular resource
- Another advantage of DFS in a domain is that folders can be replicated automatically or manually through Microsoft File Replication Service

DFS Models

- **Stand-alone DFS model**
  - No Active Directory implementation is available to help manage the shared folders
  - This model provides only a single or flat level share
- **Domain-based DFS model**
  - Takes full advantage of Active Directory and is available only to servers and workstations that are members of a domain
  - Enables a deep, root-based, hierarchical arrangement of shared folders that is published in Active Directory

DFS Topology

- **DFS topology**
  - The hierarchical structure of DFS in the domain-based model
- **Namespace root**
  - A main container (top-level folder) in Active Directory that holds links to shared folders that can be accessed from the root
- Namespace server
  - The server that maintains the namespace root
- After the namespace root is created, it is populated by shared folders for users to access

DFS Topology (continued)

- Folders are established in a level hierarchy and appear to be in one server location
  - Although they can be on many servers
- **Replication group**
  - A set of shared folders that is replicated or copied to one or more servers in a domain

Installing DFS

- DFS is installed as a service within the File Services role
- If the File Services role is already installed, but you don't see the DFS Management tool on the Administrative Tools menu
  - This means you didn't install Distributed File System when you installed the File Services role

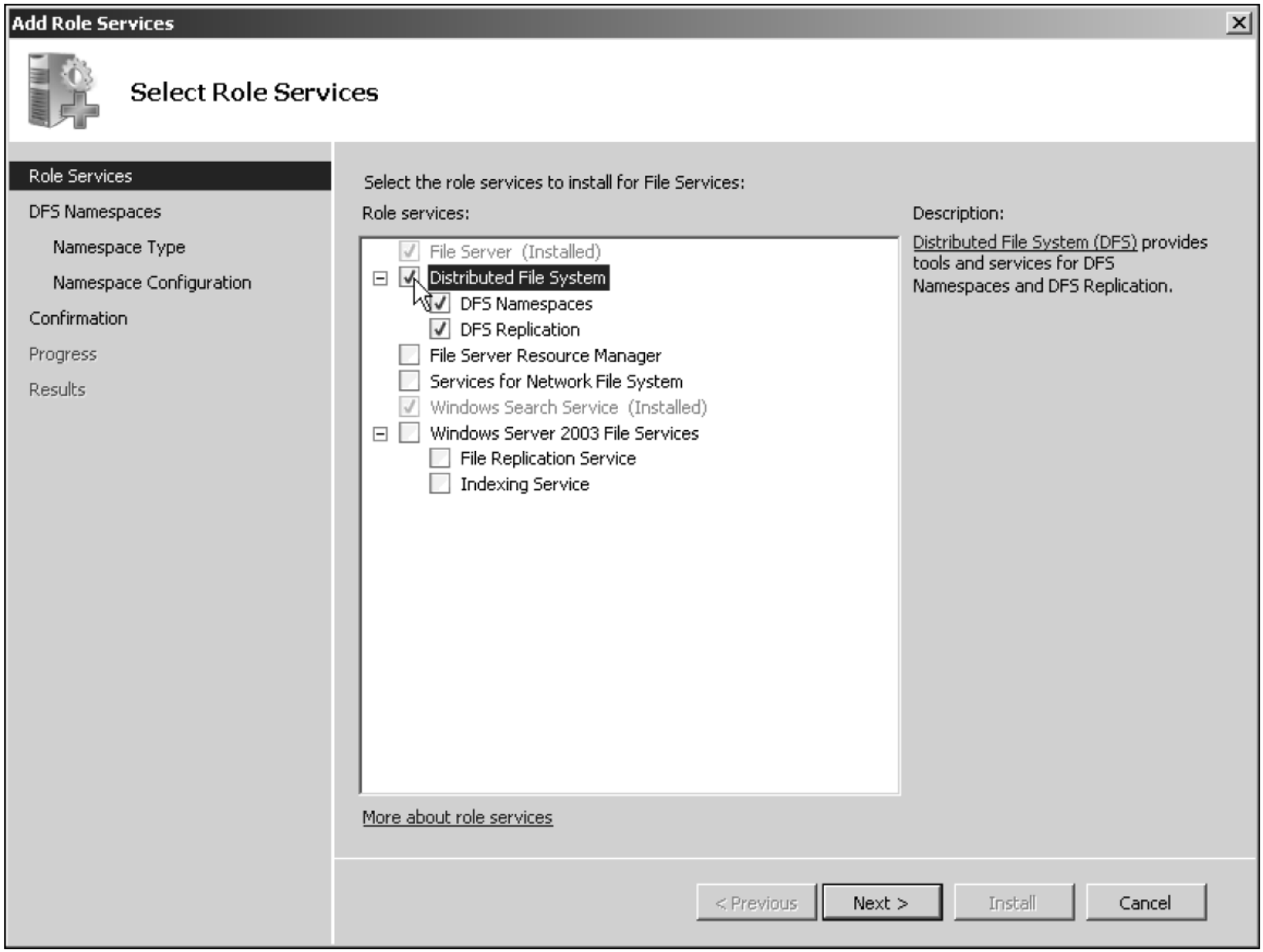


Figure 5-14 Selecting to install DFS

Installing DFS (continued)

- Activity 5-10: Creating a Namespace Root
  - Time Required: Approximately 10 minutes
  - Objective: Configure a namespace root

Managing a Domain-Based Namespace Root System

- Creating a folder in a namespace
  - A folder is simply a shared folder that you add to (or link to) the namespace root
  - **Folder target**
    - A path in the Universal Naming Convention (UNC) format, such as to a shared folder or to a different DFS path
  - **Universal Naming Convention (UNC)**
    - A naming convention that designates network servers, computers, and shared resources
  - Clients who access the namespace can see a list of folder targets ordered in a hierarchy

Managing a Domain-Based Namespace Root System (continued)

- Activity 5-11: Adding a Folder and Folder Target in DFS
  - Time Required: Approximately 5 minutes
  - Objective: Add a folder in DFS

Managing a Domain-Based Namespace Root System (continued)

- Delegating Management
  - Delegating management simply involves right-clicking the namespace and clicking Delegate Management Permissions
- Tuning a Namespace
  - Tuning options:
    - Configure the order for referrals
    - Configure cache duration for a namespace
    - Configure cache duration for a folder
    - Configure namespace polling
    - Configure folder targets as enabled or disabled

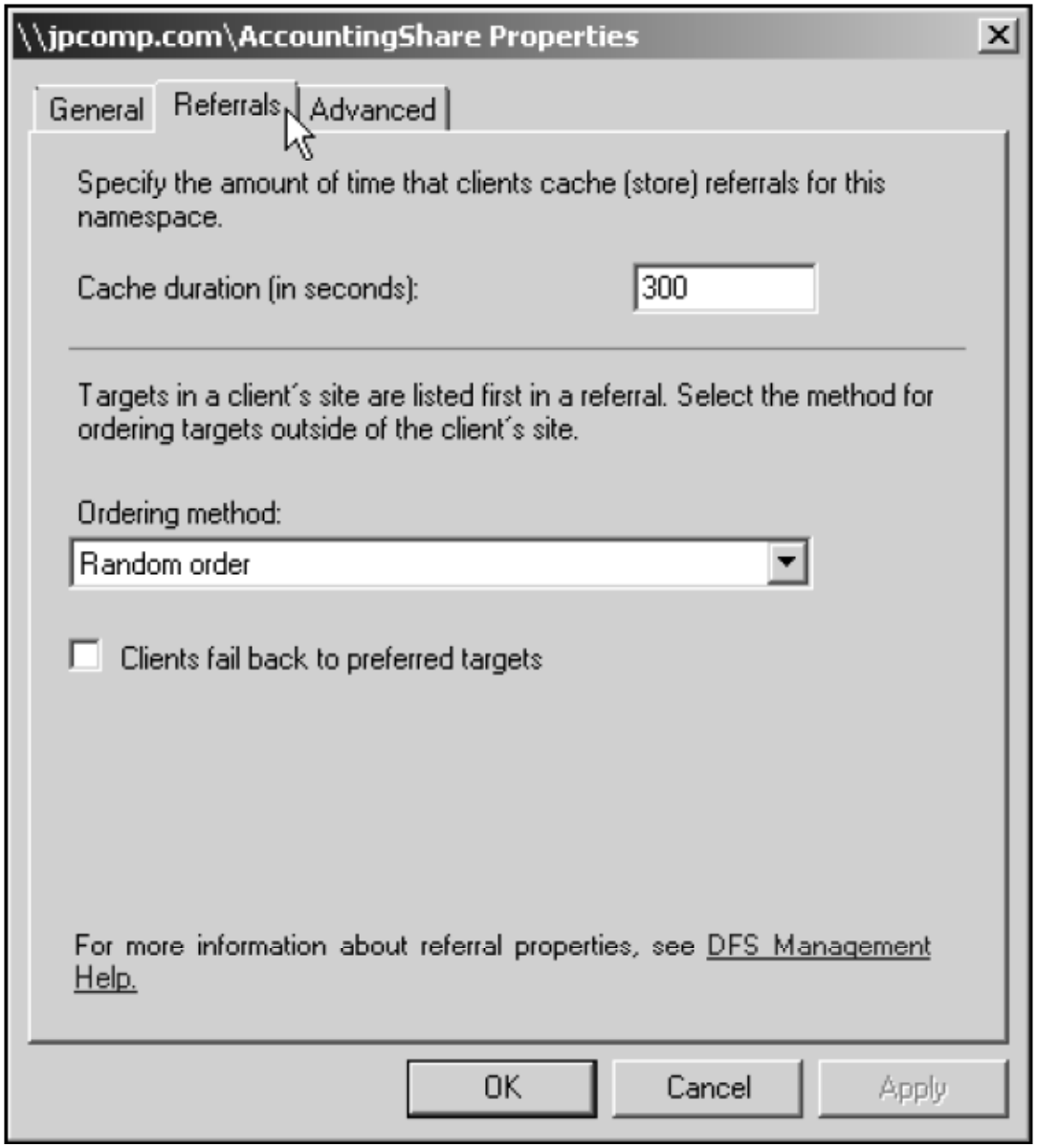


Figure 5-17 Referrals tab

Managing a Domain-Based Namespace Root System (continued)

- Deleting a namespace root
  - You can delete the namespace root via the DFS Management tool by clicking the namespace root and clicking Delete
- Using DFS Replication
  - To configure replication, you first must have defined two or more folder targets
  - You need to decide which server is to be the primary group member
    - The primary group member should be the server containing shared folders and files that are most current

Managing a Domain-Based Namespace Root System (continued)

- Windows Server 2008 includes some important improvements to DFS replication:
  - Enables faster and more reliable recovery of changes to folders in DFS when a server crashes or goes down unexpectedly, such as during a power loss
  - Replication is faster for all sizes of files
  - DFS replication is more efficient over LANs and WANs to help reduce its overhead on networks

Configuring Disk Quotas

- Disk quotas advantages:
  - Preventing users from filling the disk capacity
  - Encouraging users to help manage disk space
  - Tracking disk capacity needs on a per-user basis for future planning
  - Providing server administrators with information about when users are nearing or have reached their quota limits
- Disk quotas can be set on any local or shared volume

Configuring Disk Quotas (continued)

- You can establish disk quotas by volume or user
- Disk quota management parameters
  - Enable quota management
  - Deny disk space to users exceeding quota limit
  - Do not limit disk usage
  - Limit disk space to
  - Set warning level to
  - Log event when a user exceeds their quota limit
  - Log event when the user exceeds their warning level

Configuring Disk Quotas (continued)

- Activity 5-12: Configuring Disk Quotas
  - Time Required: Approximately 10 minutes
  - Objective: Enable disk quotas and then set a disk quota for a specific group of users

Using UNIX Interoperability in Windows Server 2008

- **Subsystem for UNIX-based Applications (SUA)**
  - Provides interoperability between Windows Server 2008 and UNIX and Linux systems
- SUA allows you to:
  - Run UNIX/Linux applications with few or no changes to the program source code
  - Run UNIX/Linux scripts
  - Use popular UNIX/Linux shells
  - Run most UNIX/Linux commands
  - Run the popular vi UNIX/Linux editor

Using UNIX Interoperability in Windows Server 2008 (continued)

- Most UNIX/Linux applications can be moved over to Windows Server 2008 SUA with only minor program code modifications
  - All applications must be recompiled in SUA
- Scripts can be moved over to Windows Server 2008 SUA and run with no or few modifications
- SUA can be set up to run in “mixed mode”
  - UNIX/Linux processes can link to Windows dynamic-link library (DLL) files

Using UNIX Interoperability in Windows Server 2008 (continued)

- **Server for Network Information Services**
  - Network Information Services (NIS) provides a naming system for shared resources on a UNIX/Linux network
  - Through the NIS server, a user can access shared resources, such as a shared partition containing shared files
  - Server for NIS also ensures the synchronization of account passwords

Using UNIX Interoperability in Windows Server 2008 (continued)

- Windows Server 2008 offers several important new features for SUA:
  - More transparent ability for UNIX/Linux applications to connect to Oracle and SQL Server databases
  - Inclusion of true 64-bit libraries for support of 64-bit applications and utilities for high-performance response
  - New utilities to support both the major UNIX versions: BSD UNIX and SVR-5 UNIX
  - Ability for application developers to use Microsoft Visual Studio for designing UNIX/Linux applications

Using UNIX Interoperability in Windows Server 2008 (continued)

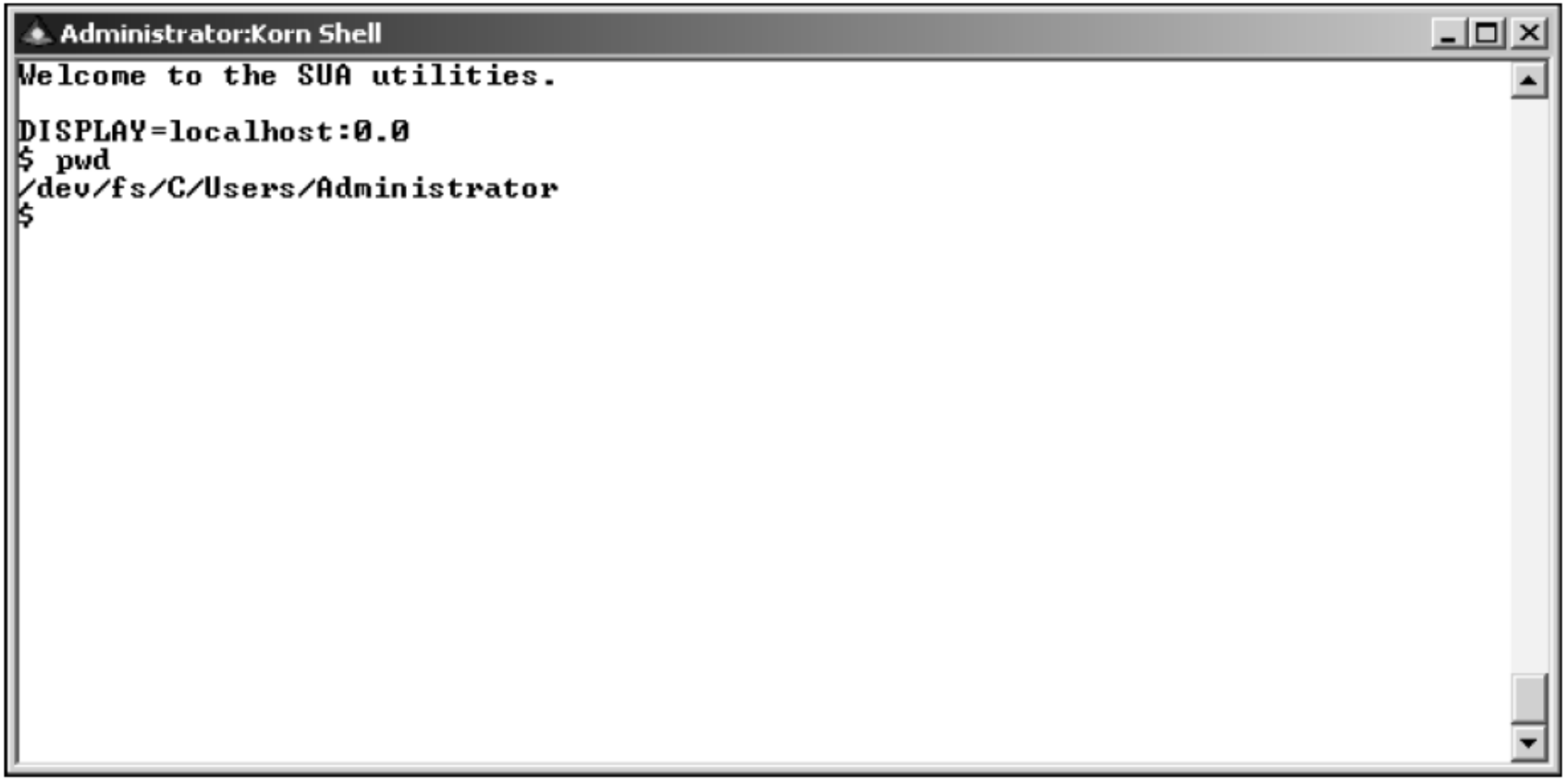


Figure 5-19 Window for using the Korn shell

Summary

- Windows Server 2008 uses discretionary access control lists for managing access to resources
- NTFS uses folder and file attributes for one level of security
- When you use the encrypt attribute, this employs the Microsoft Encrypting File System to protect files and folders
- Permissions provide another level of security for files and folders

Summary (continued)

- Special permissions provide the option to further customize security at a more granular level than basic permissions
- Folder and file auditing enable you to track who has accessed resources
- Folder and file owners have Full control permissions, including the ability to change permissions
- Folders can be shared for users to access over a network, and shared folder security is configured through share permissions

Summary (continued)

- Use the Effective Permissions capability to troubleshoot a security conflict
- The Distributed File System (DFS) enables you to set up shared folders
- Use disk quotas to manage the resources put on a server disk volume
- If you have a network that uses a combination of Windows Servers and UNIX/Linux computers, you can install the Subsystem for UNIX-based Applications