How the Folder Redirection Extension Works

In this section

Folder Redirection, a client-side extension (CSE) of Group Policy, enables an administrator of an organization to set policies that redirect users' folders to a different location, such as a shared network folder. When used in combination with Roaming User profiles and offline files, Folder Redirection enables administrators to centrally manage and protect user data and settings including policy-based management of user desktops.

Folder Redirection provides the greatest benefit when used with other components of IntelliMirror in an Active Directory environment. Folders can be redirected only from client computers running Windows 2000 or Windows XP.

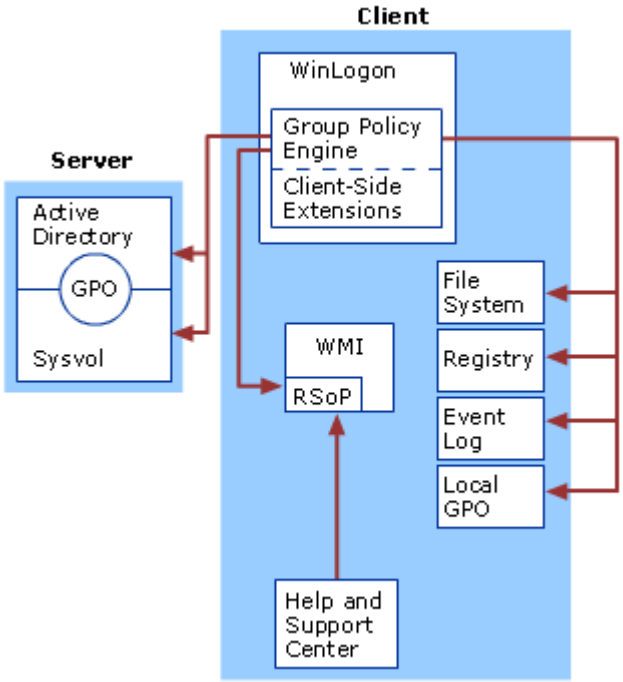
Back to Top

Folder Redirection Architecture

The Folder Redirection client-side extension consists of a dynamic-linked library (DLL), fdeploy.dll, a redirection information file, {25537BA6-77A8-11D2-9B6C-0000F8080861}.ini, which includes information about Folder Redirection options specific to each user profile on the computer, and {1C08E84D-F112-4252-978B-EC82A225CC20}.ini which stores information about previous locations used for Folder Redirection.. Fdeploy.dll is loaded during the logon process by the Group Policy engine. The Folder Redirection policies are passed to fdeploy.dll, which then examines the policy settings and redirects user folders based on those settings.

The first step in Folder Redirection is to get a list of all Folder Redirection policies and evaluate them. The policies are handed down to the fdeploy.dll by the Group Policy engine through an API call into fdeploy.dll. The policies come in the form of two linked lists. The first list is a list of added policies. This refers to a list of Folder Redirection policies that are still applicable to the user. The second list is a list of deleted policies that refers to policies that used to be applicable to the user but are no longer applicable. The changes that result from Folder Redirection are then updated with the shell. The client-side cache, also referred to as offline files, is also updated to reflect the files moved by Folder Redirection.

Folder Redirection Client-side Extension Architecture



The following table describes the components that interact with Folder Redirection extension.

Folder Redirection Components

Component	Description
Fdeploy.dll	Client-side extension DLL that processes policy settings for Folder Redirection.
{25537BA6-77A8-11D2-9B6C-0000F8080861}.ini	Locally cached redirection information file specific to each user profile.
{1C08E84D-F112-4252-978B-EC82A225CC20}.ini	A file indicating the status of the previous locations used for redirected folders. This file is used to detect server location changes and user name changes.
Server (domain controller)	In an Active Directory forest, the domain controller is a server that contains a writable copy of the Active Directory database, participates in Active Directory replication, and controls access to network resources.
Active Directory	Active Directory, the Windows-based directory service, stores information about objects on a network and makes this information available to users and network administrators. Administrators link Group Policy objects (GPOs) to Active Directory containers such as sites, domain, and organizational units (OUs) that include user and computer objects. In this way, policy settings can be targeted to users and computers throughout the organization.
Sysvol	The Sysvol is a set of folders containing important domain information that is stored in the file system rather than in the directory. The Sysvol folder is, by default, stored in a subfolder of systemroot folder (%\systemroot\sysvol\sysvol) and is automatically created when a server is promoted to a domain controller. The Sysvol contains the largest part of a GPO: the Group Policy template, which includes Administrative Template-based policy settings, security settings, script files, and information regarding applications that are available for software installation. It is replicated through the File Replication Service (FRS) between all domain controllers in a domain.
Group Policy object (GPO)	A GPO is a collection of Group Policy settings, stored at the domain level as a virtual object consisting of a Group Policy container and a Group Policy template. The Group Policy container, which contains information about the properties of a GPO, is stored in Active Directory on each domain controller in the domain. The Group Policy template contains the data in a GPO and is stored in the Sysvol in the /Policies subdirectory. GPOs affect users and computers that are contained in sites, domains, and OUs.
Local Group Policy object	The Local Group Policy object (Local GPO) is stored on each individual computer, in the hidden **Windows\System32\GroupPolicy** directory. Each computer running Windows 2000, Windows XP Professional, Windows XP 64-Bit Edition, Windows XP Media Center Edition, or Windows Server 2003 has exactly one Local GPO, regardless of whether the computers are part of an Active Directory environment. Local GPOs are always processed, but are the least influential GPOs in an Active Directory environment, because GPOs based on Active Directory have precedence.
Winlogon	A component of the Windows operating system that provides interactive logon support, Winlogon is the service in which the Group Policy engine runs.
Group Policy engine	The Group Policy engine is the framework that handles common functionalities across registry-based settings and client-side extensions (CSEs).
Client-side extensions	CSEs run within dynamic-link libraries (DLLs) and are responsible for implementing Group Policy at the client computer. The CSEs are loaded on an as-needed basis when a client computer is processing policy.
File system	The NTFS file system on client computers.
Registry	A database repository for information about a computer's configuration, the registry contains information that Windows continually references during operation, such as: • Profiles for each user. • The programs installed on the computer and the types of documents that each can create. • Property settings for folders and program icons. • The hardware on the system. • Which ports are being used.

	The registry is organized hierarchically as a tree, and it is made up of keys and their subkeys, hives, and entries. Registry settings can be controlled through Group Policy, specifically, Administrative Templates (.adm files). Windows Server 2003 comes with a predefined set of Administrative Template files, which are implemented as text files (with an .adm extension), that define the registry settings that can be configured in a GPO. These .adm files are stored in two locations by default: inside GPOs in the Sysvol folder and in the **Windows\inf** directory on the local computer.
Event log	The Event log is a service, located in Event Viewer, that records events in the system, security, and application logs.
Help and Support Center	The Help and Support Center is a component on each computer that provides HTML reports on the policy settings currently in effect on the computer.
Resultant Set of Policy (RSoP) infrastructure	All Group Policy processing information is collected and stored in a Common Information Model Object Management (CIMOM) database on the local computer. This information, such as the list, content, and logging of processing details for each GPO, can then be accessed by tools using Windows Management Instrumentation (WMI).
WMI	WMI is a management infrastructure that supports monitoring and controlling of system resources through a common set of interfaces and provides a logically organized, consistent model of Windows operation, configuration, and status. WMI makes data about a target computer available for administrative use. Such data can include hardware and software inventory, settings, and configuration information. For example, WMI exposes hardware configuration data such as CPU, memory, disk space, and manufacturer, as well as software configuration data from the registry, drivers, file system, Active Directory, the Windows Installer service, networking configuration, and application data. WMI filtering in Windows Server 2003 enables you to create queries based on this data. These queries (WMI filters) determine which users and computers receive all of the policy configured in the GPO where you create the filter.
SMB	Server Message Block (SMB) protocol is the primary method of file and print sharing. Folder Redirection client-side extension use SMB to access the Sysvol as well as back up and retrieve files to a remote file system. The client computer also uses SMB to read the Sysvol on the domain controller.

The following table shows the location of the files used by Folder Redirection.

Folder Redirection Component Locations

Component	Location
Fdeploy.dll	%windir%\system32
{25537BA6-77A8-11D2-9B6C-0000F8080861}.ini	%USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\File Deployment\
Fdeploy.log	%windir%\debug\usermode\

Locally Cached Redirection Information File

Whenever a folder is redirected, fdeploy.dll saves the details of the redirection in a local file. This data is used later by fdeploy.dll to determine how to process redirected folders if a user's personal name (UPN) or security group membership changes. The file is also used with roaming user profiles.

The redirection information file is stored in the %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\File Deployment\ folder.

The file has one section per folder currently redirected through Folder Redirection policy. If there is no policy applicable for a particular folder, then that section is removed from this file. For each folder, the following information is maintained:

- **User Name.** The user name of the user when the redirection was performed. This information is used to handle UPN changes.
- **Path.** The path to which the folder is redirected.

- **Flags.** The flags describing the Folder Redirection settings.
- **Group.** The string representation of the SID for the security group that was responsible for that particular path to get chosen as the redirection destination.
- **GPO.** The GUID of the GPO that was responsible for redirection.

An example cached redirection file:

```
[Desktop]
Username=username
Path=%USERPROFILE%\Desktop
Flags=31
Group=S-1-5-21-397955417-626881126-188441444-3038017
GPO={AFC36721-F1D5-46BA-981C-FC33BDE293D7}
```

Back to Top

Folder Redirection Processes and Interactions

Folder Redirection client-side extension interacts with the Group Policy engine as well as the shell.

Folder Redirection processing contains five steps:

1. Determine which user folders to redirect based on changes to Group Policy settings at time of logon.
2. Determine the target location specified for redirection and confirm the user has access rights to that location.
3. If the target folder does not exist, the folder is created and the appropriate access control list (ACL) rights are set.
4. If the folder exists, access rights and folder ownership are checked.
5. If desired, the files contained within specified folders are moved to the new location, which also deletes them from the source folder if the source folders are local.

If the policy for Folder Redirection does not specify that the user be granted exclusive access to the redirected location, fdeploy.dll confirms only that the destination folder exists. If the target folder does not exist, it is created. However, if the policy requires that the user be given exclusive access to the destination folder, fdeploy.dll does one of the following, depending on whether the destination folder already exists or not.

Important

- With Windows XP SP1 the user must be the owner of the folder on the server. If not, the ownership check fails, and redirection fails unless a policy is set to ignore this. This was added in SP1.
- If the destination already exists, fdeploy.dll checks whether the user is the owner of the folder. If the folder is owned by the user, fdeploy.dll proceeds; otherwise it fails and aborts the redirection of that folder.
- If the destination does not exist, fdeploy.dll creates the folder and then sets ACLs on it so that only the user and the local system account have full access to it, but no other users have any access to it. Fdeploy.dll also ensures that the folder does not inherit any Access Control Entries (ACE) from its parent. Fdeploy.dll then sets the user as the owner of the folder. This is done because if the user belongs to the local administrators group on the destination server, the local administrators group becomes the owner of the group by default. By setting the user as the owner of the folder you can assure that any quota accounting for that user is done correctly. This also ensures that any subsequent redirections to the same location from other computers do not fail due to the ownership check performed on pre-existing folders described previously. If configuration of either ownership or Access Control Lists (ACL) fails, the process is aborted and redirection fails for the folder specified. If not, redirection proceeds and the files are copied to the redirected location.

Once this is done, fdeploy.dll deletes the files from the source. However, there is an exception to this. If the source is a network location and the destination is a local location, the files are not deleted from the source. This ensures that any subsequent redirection operations from other computers also get a copy of the files on the local computer as required by the policy.

Folder Redirection Interaction with the Shell

There are two main points of interaction with the shell. The first point is when fdeploy.dll determines the current location of a folder. For this, it uses the **SHGetFolderPath** API from shell32.dll. The second point of interaction is when fdeploy.dll needs to notify the shell about the new location of a folder after it applies redirection settings. For this, it uses the **SHSetFolderPath** API from shell32.dll.

In addition, there is one other point of interaction which is specific to redirection of the My Documents folder. In the Windows shell, the properties dialog box for the My Documents folder has a **Target** tab. Settings on the **Target** page enable a user to change the location of this folder. If the folder is redirected through policy, this functionality is not desirable and should be disabled. Therefore, fdeploy.dll needs to set the policy which disables this functionality. This is done by creating a DWORD value called **DisablePersonalDirChange** under the key HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer.

Folder Redirection and Environment Variables

Environment variables can be used to define the path to the redirected location. Supported environment variables include %USERNAME% and %USERPROFILE%.

These are the only environment variables supported because other variables are not defined when the Folder Redirection extension is loaded by Winlogon. Note that fdeploy.dll does not do any sort of validation to ensure that only certain allowed variables are present in the paths. It merely expands the paths and if any variables are not defined, they stay as they are. Homedir redirection is an exception. Redirection to the home directory of the user is only allowed for the My Documents folder; fdeploy.dll does perform validation to ensure that other folders cannot be redirected to that location.

Folder Redirection and Mapped Drives

Because Folder Redirection is processed early in the logon process, drives mapped with logon scripts (including the homedrive for folders other than My Documents), the Folder Redirection client side extension is not able to redirect to these locations. At the time that redirection takes place, the drives do not exist hence redirection fails.

Using Offline Files Settings

Using offline files settings on a shared network folder where user data is stored is especially useful for users of portable computers. It is recommended that you use Folder Redirection in conjunction with offline files.

The offline files feature of Windows, also referred to as client-side caching, is designed to make network access of files more efficient by keeping a local copy of the files in the offline files cache. It also facilitates access to those files when the user is not connected to the network. Since Folder Redirection moves a user's files to and from network locations, it is important for it to move files within the local offline files cache to provide a seamless and transparent experience to the user. In most cases when you use Folder Redirection, you will combine it with offline files so that users can access cached copies of the redirected folders when disconnected from the network. The following table provides some recommended settings for redirecting folders.

Note

- All redirected folders are pinned by default.

Recommended Configuration for Offline Files

Redirected Folder	Recommended Offline Folder Settings
My Documents	Auto-caching for documents or manual caching for documents (if you want users to have to manually make files and folders available offline).
My Pictures	Auto-caching for documents or manual caching for documents (if you want users to have to manually make files and folders available offline).
Application Data	Auto-caching for programs.
Desktop	Auto-caching for programs if the desktop is read-only.

Folder Redirection and Software Installation Policies

When logon optimization is enabled, a user might need to log on to a computer twice before Folder Redirection policies and software installation policies are applied. This is because application of these types of policies requires the synchronous policy application. During a policy refresh (which is asynchronous), the system sets a flag that indicates that the application of Folder Redirection or a software installation policy is required. The flag forces synchronous application of the policy at the user's next logon.

Because background refresh is the default behavior in Windows XP, Folder Redirection and Software Installation might require as many as three logons to apply changes.

This behavior exists because Folder Redirection and Software Installation cannot apply during an asynchronous or background application of policy. Folder Redirection can be applied only when processed synchronously.

Here is a sample scenario showing how polices are applied:

- An administrator deploys a software package to User A.
- User A logs on fast and receives a background (asynchronous) application of policy.
- Because the policy application was asynchronous, the software that was set to be installed cannot be installed at this time. Instead the computer is tagged, indicating that software needs to be installed.
- The next time the user logs on, the computer instead logs on the user synchronously to allow the software package to be installed. (This is the same behavior as Windows 2000). This results in one extra logon for the software to be installed.
- In the case of Advanced Folder Redirection, because policy is evaluated based on security group membership three logons will be required: the first logon to update the cached user object (and security group membership), the second logon for policy to detect the change in security group membership and require a foreground policy application, and the third logon to actually apply Folder Redirection policy in the foreground.

Note

- When a client running Windows XP logs onto a Windows 2000 or Windows Server 2003 Active Directory, all Software Installation policy settings for Windows 2000 clients will be applied and work successfully on the Windows XP client.

Back to Top

Related Information

The following resources contain additional information that is relevant to this section.

- Core Group Policy Technical Reference in the Group Policy collection.
- Group Policy Tools in the Group Policy collection.