

UW Computer Science and Engineering

Dfs (Microsoft's Distributed file system)

revised June 2004, June 2008 - Warren Jessop

Executive Summary

Under one drive, **O:**, on a Windows workstation you can access shared files, folders, and directories that reside on a wide variety of CSE file servers, both Windows and UNIX. This includes, for example, *all* Windows and UNIX home, project, web and course directories.

What is Microsoft Dfs?

Microsoft's Distributed File System, or Dfs (pronounced "doofus")¹, provides a stable, location-independent² naming scheme for all shared files you need to access when using Windows. A single "UNC"³ share, or a single drive mapped to such a share, is able to contain files and directories on any number of file servers. It works by looking up the actual locations of remote shares in a table called the Dfs map, in a way very similar to automount mapping in UNIX (see [UNIX Automount Mapping in CSE], page 10).

The UNC name for all Dfs files in CSE begins with `\\cseexec\cs`.⁴ By convention `\\cseexec\cs` will automatically be associated with drive letter **O:**. If this is not the case you can use one of the file manager tools to map `\\cseexec\cs` to **O:**, or you can enter the command `'net use O: \\cseexec\cs'`.

If you are using a computer that does not use CSE's name servers, for example your personal computer at home, you will need to map the fully qualified name, `\\cseexec.cs.washington.edu\cs`. *Important: you will also need to make sure your home computer is configured correctly*, as outlined in [Network Location Cannot Be Reached], page 3.

Microsoft's documentation on Dfs can be accessed at <http://support.microsoft.com/kb/812487>

¹ Microsoft Dfs should not be confused with Transarc's Distributed File System product called DFS (pronounced "dee eff ess", http://www.transarc.com/Library/whitepapers/efs_strategy.html) which has been around for many years. DFS is a component of DCE (the Distributed Computing Environment), promoted by the **Open Group**, previously known as the Open Software Foundation. To distinguish its distributed file system from the original, as well as to avoid treading on Transarc's trademark, Microsoft has chosen to spell Dfs with lower-case "f" and "s". In Transarc's DFS the main goal is to provide a universal naming scheme for all files.

² In location-independent file sharing the name of a file is independent of both the file's and the user's physical location, contributing to ease of file sharing and resource management.

³ One definition of "UNC (Universal Naming Convention) name" (from the Windows 2000 on-line Glossary) reads as follows: "A full Windows 2000 name of a resource on a network. It conforms to the `\\servername\sharename` syntax, where *servername* is the server's name and *sharename* is the name of the shared resource. UNC names of directories or files can also include the directory path under the share name with the following syntax: `\\servername\sharename\directory\filename`."

⁴ This used to be `\\ntdfs\cs`; see [Fault Tolerance], page 3 to know why it changed.

Dfs in CSE

Dfs High-Level Structure

The following table shows the topmost directory structure of Dfs directories in the department.

Dfs path prefix	Comment
0:\nt\projects	Research projects
\iprojects	Instructional projects
\runs	Research unsupported area
\iuns	Instructional unsupported area
\courses	NTFS instructional course directories
\dist-area	Software distribution area
\office	Research office area
\support	Research support area
\images	PC rebuild images
\grail	Grail (graphics) projects and data
\cygwin	Cygwin installation
\homes\ifaculty	Instructional PC faculty home dirs
\rfaculty	Research PC faculty home dirs
\istudents	Instructional PC student home dirs
\rstudents	Research PC student home dirs
\istaff	Instructional PC staff home dirs
\rstaff	Research PC staff home dirs
0:\unix\projects	Same as /projects, research UNIX
\nfs	Research NFS Gateway
\vole	Instructional TA NFS Gateway
\homes\gws	Same as /homes/gws, research UNIX
\iws	Same as /homes/iws, instructional UNIX
\sgi	Same as /homes/sgi, grail servers
\sys	Same as /homes/sys, system servers
0:\cse	Like /cse, research UNIX
0:\sources	Like /sources, research UNIX

Things to notice in this table:

- Path prefixes with **\nt** refer to Windows (NTFS) file systems. Under **\nt**, where a distinction must be made between similar shares on different Windows domains, the convention is: names starting with “r” (research) refer to directories on Windows file servers in the CSERESEARCH domain; those starting with “i” (instructional) to directories on servers in the CSEPCLAB domain.
- Path prefixes with **\unix** refer to UNIX shares accessed via Samba. Each entry in the departmental **/homes** UNIX NFS automount maps is also found in the Dfs map. All entries in the **/projects** NFS map are as well.
- Path prefixes with **\cse** or **\sources** refer to “platform neutral” shares. Currently these shares are all on UNIX servers, but they may contain data pertinent to all platforms. Each entry under **\cse** and **\sources** has a corresponding entry in **/cse** or **/sources** UNIX automount maps.

Fault Tolerance

Microsoft has designed Dfs to be fault tolerant by allowing the Dfs map to be published to the “Active Directory” of a Windows domain. In CSE it is replicated among the domain controllers of “cseexec.cs.washington.edu”, our top-level domain. Thus the root, `\\cseexec\cs`, refers not to a host and share, but to a “volume object” in the active directory.

Until a few years ago a Dfs map in the active directory was restricted to about 1000 entries, so we used to run a standalone Dfs server reachable by the name `ntdfs`.

The standalone server still exists. Currently Dfs maps in cseexec’s Active Directory are replicated on server `ntdfs`, and reachable by `\\ntdfs\cs` or `\\ntdfs.cs.washington.edu\cs`.

Note: Previously Dfs files could also be accessed via `\\ntdfs.cseresearch.cs.washington.edu`; however, that is no longer the case.

Things to Keep in Mind About Dfs

Access Credentials

If you have not already given your CSERESearch or CSEPCLAB Windows credentials (e.g. you are using your personal laptop), you should be prompted for credentials¹ when you attempt to map `\\cseexec\cs` to drive **O:**. When entering credentials prefix your username with the CSE domain you want, e.g. `CSEPCLAB\jouser`, and enter the corresponding password.

Access Problems

If you are having problems accessing a Dfs path, you may be experiencing one of the errors described in this section.

Note: See [\[An Alternative to Mapping Files on Your Home System\]](#), page 10 for a way to finesse all of these access problems and yet still be able to easily transfer files between CSE systems and your home system.

Network Location Cannot Be Reached

If you see

```
...not accessible. The network location cannot be reached.
```

or perhaps

```
...not accessible. The network name cannot be found.
```

or

¹ It may happen that you are *not* prompted for credentials; in this case you need to click “Connect using a different user name” in the “Map Network Drive” window in order to enter your credentials.

`...not accessible. The specified network name is no longer available`

or other variants of the same idea, it usually means that networking has not been set up correctly on your Windows client.

In some cases you may have successfully mapped `\\cseexec.cs.washington.edu\cs` to a drive letter from a client outside the local CSE network, and now you are trying to drill down into a folder under that drive letter. You can usually fix this by reconfiguring your workstation by making sure that its “DNS suffixes” include “cs.washington.edu.” To do this follow the procedure in [[Adding cs.washington.edu to your DNS suffixes](#)], page 9. Note that after you do this you will need to disconnect and and re-map `\\cseexec.cs.washington.ed\cs`.

In rare cases these messages could also mean that one or more network services are down. Contact support@cs if you have questions.

Access Denied

When you see ‘`...not accessible. Access is denied`’ it means you do not have permission to get to the file or folder you are trying to access.

On Windows servers, each file and folder (directory) has read/write permissions assigned to individuals and/or groups. In the case of UNIX, each file has permissions assigned to its owner, one group, and the world (everyone). See http://www.cs.washington.edu/lab/support/web_write.html for a primer on UNIX file permissions.

Therefore, your access to a file or directory is based on your login name and group memberships, which in turn depend on your job and the projects you are working on.

Login names, group names, and group membership are “universal” among all CSE systems, whether Windows or UNIX.

Folder Moved or Removed / Location Unavailable

When you see

`...not accessible. The folder was moved or removed.`

or perhaps

`...refers to a location that is unavailable...`

it means a folder you are trying to access does not exist, or it can mean that a file server is not (or is no longer) sharing that folder.

Network Password Is Not Correct

When you see

```
...not accessible. The specified network password is not correct
```

or perhaps

```
...not accessible. Login failure: unkown user name or bad password.
```

it means you do not have the correct Windows or UNIX credentials. Both Windows and UNIX servers attempt to verify your Windows credentials with a Windows domain controller.

Furthermore, you must also have an account on each file server that Dfs attempts to access. See [You Need a UNIX Account to Access UNIX File Systems], page 7 for further info about UNIX file servers.

Network Path Not Found

When you see

```
...not accessible. Windows cannot find the network path.
```

it most likely means that the remote file server you are trying to reach is down or cannot be reached (e.g. your ISP is blocking the file sharing port²), although it might also mean that your Windows client configuration is bollixed.

If your ISP is blocking the port, there is a way around this problem, but at present it works for windows file servers only. Note that if you can map `\\cseexec.cs.washington.edu\cs` to a drive letter, that is proof that your ISP is *not* blocking the file sharing port. If you cannot do such a mapping, see www/lab/sw/nt-inegration/remotearchive/winremote.html for directions on how to “tunnel in” to a Windows file server³ and get around the fact that your ISP is blocking a port.

Not Authorized to Log In / Format Invalid

When you see

```
...not accessible.  
The account is not authorized to log in from this station
```

or perhaps

² Port 445 is used for Windows file sharing. If your ISP confirms that they are blocking this port, your only recourse is to complain to them

³ Note that `cseexec.cs.washington.edu` is *not* a file server: you must know the exact names of the Windows file servers where your folders reside and map *directly* to those servers.

...not accessible.

The format of the specified network name is invalid

it usually means there is something wrong with one of the file or domain servers. Contact support@cs to have the situation corrected.

Your SMB Client Software May Not Be Dfs-Aware

This is not a problem with all Windows versions since Windows 98. But it can be (and *is*) a problem with other types of clients, e.g. the stuff that comes with Mac OS or Linux. Just because you can access remote files on Windows servers and UNIX (via Samba) does not mean you will be able to drill down into `\\cseexec\cs` beyond the so-called “junction points” that are derived from the Dfs map.

For example, you may be able to get to `\\cseexec\cs\unix\homes\iws` just fine, but not to your home directory one level down. In some cases third party software is available to make your system Dfs-aware, e.g. <http://www.thursby.com/products/dave.html> for MacOS users.

Mapping Arbitrary Directories to a Drive Letter

You can map any directory under the Dfs root to a drive letter.⁴ For example, `jouser`'s `barb`⁵ home directory UNC path via Dfs is `\\cseexec\cs\unix\homes\gws\jouser`. She will find it very convenient to map this to a drive letter using one of the file explorer tools or, for example, a command:

```
net use J: \\cseexec\cs\unix\homes\gws\jouser
```

Following this example, `jouser` could then access her `barb` home directory by just clicking on the **J:** drive.

Is this better than mapping **J:** to `\\barb\jouser`? YES. Why? Because `jouser`'s so-called “`barb` home directory” is most likely *not on barb*—and neither is yours.

By mapping your home directory with Dfs you ensure that you deal directly with the UNIX server your home directory resides on; by mapping via `barb` you slow things down, because the always overloaded `barb` is then forced to act as an intermediary between your PC and your UNIX home directory.

Finding the File Server for a Dfs Path

If you need to know what file server a particular directory is on, use one of the following procedures:

⁴ You can even do this with NT 4.0, but only for Dfs (`\\cseexec\cs`) paths.

⁵ `barb` is the departmental main computer

on Windows systems:

Use Windows Explorer as follows: right-click on a directory and choose “Properties”; then choose the “Dfs” tab. This will show you the actual location of the directory.

on UNIX systems:**Windows home directories:**

Use the **windows-home** command:

```
/cse/lab/bin/windows-home domain
```

where *domain* is **i** for instructional (CSEPCLAB) systems or **r** for research (CSERESEARCH) systems. For example, to locate your research Windows home directory share:

```
/cse/lab/bin/windows-home r
```

other directories:

Consult `/cse/lab/maps/dfs.csv`, a comma-separated spreadsheet file containing the complete Dfs map.

You Need a UNIX Account to Access UNIX File Systems

In order to access a UNIX file from Windows you need to have an account on a UNIX server that can access that file. This is no different from Windows, where you also need to be authenticated in the domain of a Windows server to access its files.

So, for example, if you want to access **O:\cse\www** the remote host, in this case **www**, needs to see an entry for you in its `passwd` file. Note that this does not necessarily give you login privileges to the remote UNIX host, as you could have a “nologin” entry that only allows remote access of files. Actually **www** is not a good example, since all faculty, staff, and grads have `passwd` entries on the web server. However, this is not the case for all UNIX file servers, e.g. the Grail laboratory’s research servers, on which you probably do not have an account.

Note: To find out the hosts where you have login accounts, use this link:

<http://www.cs.washington.edu/htbin-post/unrestricted/lab/myresources.cgi>

Setting Permissions on UNIX Files and Folders from Windows

Of course the easiest way to change permissions on UNIX files is to log into a UNIX system and use the **chmod** command (for which there’s a man page), but it certainly can be done using the Windows Explorer tool as well.

For files, go to the “Security” window by right-clicking on the file, choosing “Properties” and then clicking on the “Security” tab. You will see three entries under Name: “Everyone”, a group name, and a user name, displayed as follows:

```
<multiple profile icon> Everyone
```

<single profile icon> *username(hostname\username)*
 <multiple profile icon> *groupname(hostname\groupname)*

If you choose one of the three you will see the file's permissions for the highlighted **Everyone**, *username*, or *groupname* line.

To change permissions, clear all the boxes and click "Read" and/or "Write". If you want execute permission in addition to read, click "Read&Execute". If you *only* want execute permission, you'll need to click on "Advanced" and choose only the "Traverse Folder/Execute" box; the procedure to follow is the same as that outlined below for setting folder permissions.

Be sure NOT to click any boxes in the "Deny" column.

For folders (directories) the process is a bit more complicated, since (because of bugs in the Windows explorer tools, even on XP) none of the permissions are displayed in the main Security window. So, you'll need to click on "Advanced..." in all cases.

On the "Permissions" tab of the Advanced window you will see six entries. You will only be concerned with the three entries that have "This folder only" in the "Apply to" column. These three entries will be as described above: "Everyone", a group name, and a user name. Choose the one you want to change and click "View/Edit". Yet another window will pop up! (This is Windows, after all.) This window shows the exact permission settings for the folder. To set permissions:

1. Click "Clear all".
2. Click any combination of the following choices in the Allow column to achieve the permissions you want:
 - Click "List Folder/Read Data" if you want the corresponding UNIX "r" permission bit set.
 - Click "Create Files/Write Data" if you want the corresponding UNIX "w" permission bit set.
 - Click "Traverse Folder/Execute File" if you want the corresponding UNIX "x" permission bit set.
3. Click OK (the window disappears), then click Apply on the Advanced window.
4. Verify that the permissions are what you want. You'll note that Windows will want to set other Allow bits besides the ones you checked. That is normal. Also note that if you want no permissions, the Windowsese for this is "Take Ownership" under the Permission column. Again this is normal.
5. Click OK and OK to make the remaining properties windows go away.

As for files, in no case touch anything in the Deny column.

The NFS Gateways

There are two additional (hidden) directories under **O:\unix** (or **\\cseexec\cs\unix**), namely **nfs** and **vole**.

A path name such as **O:\unix\nfs\nfspath** enlists the aid of a research NFS "gateway" UNIX host, where *nfspath* is a UNIX NFS path (with /'s replaced by \'). Similarly, **O:\unix\vole** uses an instructional NFS gateway.

O:\unix\nfs should only be used for the rare situation in which an NFS file server you need to get to is not also a Samba server, or in case you do not have an account on a UNIX NFS server but you can access files on that server via NFS.⁶ Use **O:\unix\nfs** only if you have to, because the price you pay is that all data from and to a file's location must make an additional network hop via the NFS gateway host.

The other gateway machine,

vole, is used exclusively by undergrad TAs who need access to their course web files, found under directory **O:\unix\vole\cse\www\education\courses**. Undergrads must use a gateway because they do not have accounts on the web server, **www**.

Feedback

Send problems, questions and comments to support@cs.

Appendix

Adding 'cs.washington.edu' to your DNS suffixes

In Windows XP, here is what you need to do (the procedure is similar in W2000, but there may be a few minor differences):

1. Get the "local area connection status" window up by one of the following methods:
 - start -> settings -> network connections -> local area connection
 - start -> settings -> control panel -> network connections -> local area connection
 - start -> control panel -> network connections -> local area connection

(There may be more ways - I only have two XP systems for which there are 3 ways!)
2. From the "local area connection status" window, get the DNS tab of the "advanced TCP/IP settings" window up as follows:
properties -> internet protocol (TCP/IP) -> properties -> advanced... -> DNS
3. In the "advanced TCP/IP settings" window there will be an "append these DNS suffixes (in order)" radio button. Click that button and use the "add..." button to add
cs.washington.edu
to the list.
4. Hit "OK" as many times as is necessary (or "close" if there is no "OK" button) to get out of this fine mess.

⁶ An NFS server does not require the user to have a local UNIX password entry in order to access its files.

UNIX Automount Mapping in CSE

UNIX users in CSE are already familiar with the concept of Dfs, since it is exactly the way NFS automounting works. In NFS automounting, the actual locations (*host:/directory*) are listed in a table called an automount map in which each location has a corresponding key. For example, here are a few entries from the **/projects** map on the research UNIX systems:

Key	Maps to
ai2	fury:/fury13/ai2
robotics	dark:/dark10/robotics
grail20	doppio:/doppio1/projects

/projects can be called the “root” of the map. The automount path prefix is formed by combining root and key values: *root/key*. Thus any reference to **/projects/ai2/...** on any host results in an actual access to **/fury13/ai2/...**, located on host **fury**. NFS automounting on the research UNIX systems is set up with *multiple* roots: **/projects**, **/sources**, **/cse**, **/homes/gws**, etc.

An Alternative to Mapping Files on Your Home System

You need to transfer files to/from your home system, but you cannot or do not wish to go through the fru-fra⁷ of getting your system to map remote drives. What can you do? Answer: Use remote desktop to log in to a CSE system (either your desktop system in Allen or one of the time-sharing terminal servers), and use the “Redirection of Local Drives” feature. All is described in

<http://www.cs.washington.edu/lab/sw/uwcsehydra.html#rdpclient>.

⁷ Dealing with your port-blocking ISP ([[Network Path Not Found](#)], page 5) or reconfiguring your home system ([[Network Location Cannot Be Reached](#)], page 3).

Table of Contents

Executive Summary	1
What is Microsoft Dfs?.....	1
Dfs in CSE	2
Dfs High-Level Structure	2
Fault Tolerance	3
Things to Keep in Mind About Dfs.....	3
Access Credentials	3
Access Problems	3
Network Location Cannot Be Reached	3
Access Denied	4
Folder Moved or Removed / Location Unavailable	4
Network Password Is Not Correct	5
Network Path Not Found	5
Not Authorized to Log In / Format Invalid.....	5
Your SMB Client Software May Not Be Dfs-Aware	6
Mapping Arbitrary Directories to a Drive Letter	6
Finding the File Server for a Dfs Path	6
You Need a UNIX Account to Access UNIX File Systems	7
Setting Permissions on UNIX Files and Folders from Windows	7
The NFS Gateways	8
Feedback	9
Appendix	9
Adding ‘cs.washington.edu’ to your DNS suffixes	9
UNIX Automount Mapping in CSE.....	10
An Alternative to Mapping Files on Your Home System	10