sgi

StorHouse/CIFS Installation and Configuration Guide

Publication Number
007-6328-001

Release 2.0

November 19, 2013

StorHouse®

Publication Number: 007-6328-001

Contents

Welcome

Welcome to the *StorHouse/CIFS Installation and Configuration Guide*. StorHouse/CIFS is a Linux-based StorHouse component that provides industrial-strength SMB/CIFS 2.X support and Active Directory integration so that Microsoft Windows users can seamlessly access files and folders within StorHouse/RFS.

StorHouse/CIFS enables you to join your Linux-based StorHouse/RFS server to an Active Directory domain, thereby making the StorHouse/RFS mount point (`/RFS` folder in Linux) available to Windows users as a standard Windows file share. This capability is in addition to the native NFS support that a StorHouse/RFS Linux platform already provides

Portions of the technology for StorHouse/CIFS are the result of a partnership between SGI and Likewise Software – a leader in the area of enterprise-class authentication, security, and information access solutions targeted for use in mixed network environments. While Likewise Software makes certain portions of its software available under Open Source licensing terms, StorHouse/CIFS is made available to SGI customers under commercial licensing terms. As such, the use of StorHouse/CIFS is governed by the SGI End User License and Services Agreement (EULASA) for your StorHouse solution, and the product is fully supported by the SGI customer support organization.

Purpose of This Manual

This manual explains how to install and configure the StorHouse/CIFS software on a Linux-based StorHouse/RFS server. It also discusses how to set default permissions on files/directories added to StorHouse/RFS through StorHouse/CIFS

Audience

This document is for StorHouse/RFS system administrators who are responsible for installing, configuring, and maintaining StorHouse/RFS. It assumes that the reader is familiar with StorHouse/RFS concepts and operation, knowledgeable about installing software in Windows/UNIX environments, and familiar with Windows and UNIX directory and file permissions.

Contents

This manual contains two chapters and three appendices:

- Chapter 1, “Installing and Configuring the StorHouse/CIFS Software,” introduces StorHouse/CIFS and explains how to install and configure the StorHouse/CIFS agent on a Linux-based StorHouse/RFS server.
- Chapter 2, “StorHouse/CIFS and Active Directory/UNIX Network Information Service Interoperability,” explains how to configure the system for StorHouse/CIFS and Active Directory/UNIX network information service interoperability.
- Appendix A, “Setting Default Permissions on Files/Directories Added to RFS through StorHouse/CIFS,” explains how to set default permissions on files/directories added to StorHouse/RFS through StorHouse/CIFS.
- Appendix B, “UNIX to Windows Security Rights Mapping,” communicates important information about how StorHouse/CIFS brokers UNIX (POSIX) access rights to Windows client systems using the StorHouse/RFS virtual file system on Linux.
- Appendix C, “More Information about StorHouse/CIFS Permissions,” provides two tables with additional information about StorHouse/CIFS permissions.

C H A P T E R 1

Installing and Configuring the StorHouse/CIFS Software

This chapter explains how to install the StorHouse/CIFS software and configure it to work with a Linux-based StorHouse/RFS system.

About the StorHouse/CIFS Software

StorHouse/CIFS software is an agent that runs on Linux platforms supported by SGI for StorHouse/RFS. It enables StorHouse/RFS system administrators to join these StorHouse/RFS servers to Microsoft Active Directory and to authenticate users with their Active Directory credentials. StorHouse/RFS uses StorHouse/CIFS so that Windows users can directly access StorHouse data stored within a Linux-based StorHouse/RFS system. For this to work, you must install the StorHouse/CIFS software and configure it to work with StorHouse/RFS.

Installing and Configuring the StorHouse/CIFS Software

SGI distributes the StorHouse/CIFS software electronically through its HTTP website. Contact your SGI account representative for information about how to access this site and obtain the StorHouse/CIFS software.

Basic Steps

Here are the basic installation and configuration steps:

- First, install the StorHouse/CIFS software on a Linux-based StorHouse/RFS server that is running a version of Linux supported by StorHouse.
- Then, create a share on the StorHouse/RFS Linux file system.
- Next, configure StorHouse/CIFS to join an Active Directory domain.
- Continue by redirecting the StorHouse/CIFS share to refer to the actual StorHouse/RFS staging directory so that clients accessing the system using the CIFS/SMB protocol from Windows machines will see the full set of StorHouse-hosted files.
- Once you have redirected the StorHouse/CIFS share, set the proper permissions on the StorHouse/RFS file share.
- Finally, map the StorHouse/RFS directory from Linux to a Windows drive letter.

After completing these tasks, your Linux-based StorHouse/RFS server will be joined to an Active Directory domain, thereby making the StorHouse/RFS mount point (`/RFS` folder in Linux) available to both Windows and NFS users.

Installation Procedure

Use the following procedure to install and configure the StorHouse/CIFS software.

To successfully run StorHouse/CIFS, the SELinux setting must be set to Permissive or Disabled. In addition, these ports listed in Table 1 must be opened on the local Firewall.

Table 1: Ports to Open

Port	Protocol	Use
3268	TCP	Global Catalog search
464	UDP/TCP	Machine password changes (usually after 30 days)
445	TCP	SMB over TCP
389	UDP/TCP	LDAP
139	TCP	NetBIOS session (SMB)
137	UDP	NetBIOS name service
123	UDP	NTP
88	UDP/TCP	Kerberos 5
53	UDP/TCP	DNS

To install the StorHouse/CIFS software on a Linux StorHouse/RFS server

1. Copy the file, `StorHouse-CIFS-6.1.0-0.x86_64.rpm`, to the root directory of the Linux system where StorHouse/RFS is installed.

   Note: StorHouse/CIFS and StorHouse/RFS do not currently support extended file attributes. Therefore, files copied to StorHouse/RFS via the StorHouse/CIFS interface will have their extended attributes removed before the file is stored on the system.

2. Using the `root` account, type the following command to install the `.rpm` file:

   ```
   rpm -ivh StorHouse-CIFS-6.1.0-0.x86_64.rpm
   ```

The resulting display is:

```
[root@vm-jh-cent15 ~]# rpm -ivh StorHouse-CIFS-6.1.0-0.x86_64.rpm
Preparing...                ########################################### [100%]
   1:likewise-open-cifs     ########################################### [100%]
SUCCESS
SUCCESS
[  OK  ] lwsmd: [  OK  ]
Waiting for lwreg startup.ok
Installing settings from /opt/likewise/share/config/accounts.reg...
Installing settings from /opt/likewise/share/config/dcerpcd.reg...
Installing settings from /opt/likewise/share/config/eventlogd.reg...
Installing settings from /opt/likewise/share/config/lsassd.reg...
Installing settings from /opt/likewise/share/config/lwiod.reg...
Installing settings from /opt/likewise/share/config/lwreg.reg...
Installing settings from /opt/likewise/share/config/netlogond.reg...
Installing settings from /opt/likewise/share/config/npfs.reg...
Installing settings from /opt/likewise/share/config/privileges.reg...
Installing settings from /opt/likewise/share/config/pvfs.reg...
Installing settings from /opt/likewise/share/config/rdr.reg...
Installing settings from /opt/likewise/share/config/srv.reg...
Installing settings from /opt/likewise/share/config/srvsvcd.reg...
Reloading Likewise Service Manager configuration[  OK  ]
Starting service dependency: dcerpc
Starting service dependency: lwio
Starting service dependency: pvfs
Starting service dependency: npfs
Starting service dependency: netlogon
Starting service dependency: rdr
Starting service dependency: lsass
Starting service dependency: srv
Starting service: srvsvc
[root@vm-jh-cent15 ~]#
```

StorHouse/CIFS runs a number of services that you must register with the StorHouse/CIFS server. You install these services using .reg files that reside in the configuration directory after installation. To install the complete list of services, you must start the registration daemon, and then pass all `.reg` files to the registry shell command line tool.

3. Type the following command to start the registration daemon:

```
/etc/init.d/lwregd start
```

```
[root@vm-jh-cent6 ~]# /etc/init.d/lwregd start
Starting lwregd:                                           [  OK  ]
[root@vm-jh-cent6 ~]#
```

Type the following command to register the default StorHouse/CIFS-Likewise scripts:

```
for i in /opt/likewise/share/config/*.reg; do
/opt/likewise/bin/lwregshell upgrade $i; done
```

There is no display when the registration command executes successfully.

```
[root@vm-jh-cent6 ~]# for i in /opt/likewise/share/config/*.reg; do /opt/likewise/bin/lwregshell upgrade $i; done
[root@vm-jh-cent6 ~]#
```

4. Type the following command to stop the StorHouse/CIFS-Likewise registration daemon:

```
/etc/init.d/lwregd stop
```

```
[root@vm-jh-cent6 ~]# /etc/init.d/lwregd stop
Stopping lwregd:                                           [  OK  ]
[root@vm-jh-cent6 ~]#
```

5. Type the following command to start the StorHouse/CIFS-Likewise service manager:

```
/etc/init.d/lwsmd start
```

```
[root@vm-jh-cent6 ~]# /etc/init.d/lwsmd start
Starting lwsmd:                                            [  OK  ]
[root@vm-jh-cent6 ~]#
```

6. Type the following commands to start all StorHouse/CIFS-Likewise services:

```
/opt/likewise/bin/lwsm start srv
/opt/likewise/bin/lwsm start srvsvc
```

```
[root@vm-jh-cent6 ~]# /opt/likewise/bin/lwsm start srv
Starting service dependency: lwio
Starting service dependency: pvfs
Starting service dependency: npfs
Starting service dependency: netlogon
Starting service dependency: rdr
Starting service dependency: lsass
Starting service: srv
[root@vm-jh-cent6 ~]#
```

```
[root@vm-jh-cent8 ~]# /opt/likewise/bin/lwsm start srvsvc
Starting service dependency: dcerpc
Starting service: srvsvc
[root@vm-jh-cent8 ~]#
```

7. Type the following command to confirm that all StorHouse/CIFS-Likewise services necessary to support CIFS are running and active:

   ```
   /opt/likewise/bin/lwsm list
   ```

```
[root@vm-jh-cent8 ~]# /opt/likewise/bin/lwsm list
lwreg       running (standalone: 32740)
dcerpc      running (standalone: 331)
dfs         stopped
eventlog    stopped
lsass       running (standalone: 32765)
lwio        running (standalone: 32756)
netlogon    running (standalone: 32762)
npfs        running (io: 32756)
pvfs        running (io: 32756)
rdr         running (io: 32756)
srv         running (io: 32756)
srvsvc      running (standalone: 343)
[root@vm-jh-cent8 ~]#
```

8. Type the following command to enable the nsswitch for StorHouse/CIFS-Likewise. This command will expose the default `/lwcifs` share on the file system. You may need to create the /lwcifs directory manually on your Linux system after this command is executed (if this is not the first time you have installed StorHouse/CIFS on the system).

   Note: When accessing this from the Windows network, the share name is `C$`. Therefore, the share path would be `\\<RFS_server_name> \c$`.

   ```
    /opt/likewise/bin/domainjoin-cli configure --enable
   nsswitch
   ```

```
[root@vm-jh-cent6 ~]# /opt/likewise/bin/domainjoin-cli configure --enable nsswitch
SUCCESS
[root@vm-jh-cent6 ~]#
```

9. Join the intended Windows domain using the FQDN of the domain and a user account that has rights to add computers to the domain. Type the following command to prompt for the password of the defined user:

   ```
   /opt/likewise/bin/domainjoin-cli join <domain> <username>
   ```

   Note that `<domain>` is the FQDN of the target domain, and `<username>` is a user with rights to add machines to this domain. You do not need to preface the username with the domain name.

   ```
   [root@vm-jh-cent6 ~]# /opt/likewise/bin/domainjoin-cli join MyCompany.com
   AdminUserAccount
   Joining to AD Domain:  MyCompany.com
   With Computer DNS Name: vm-jh-cent6.MyCompany.com

   AdminUserAccount MYCOMPANY.COM's password:
   SUCCESS
   [root@vm-jh-cent6 ~]#
   ```

   To link the now-exposed CIFS file share to point to StorHouse/RFS, you must first create a symbolic link in the `/lwcifs/` folder to point to the StorHouse/RFS virtual directory:

   Type the following command to change your current working directory to the existing `lwcifs` folder.

   ```
   cd /lwcifs
   ```

   ```
   [root@vm-jh-cent8 ~]# cd /lwcifs
   [root@vm-jh-cent8 lwcifs]# pwd
   /lwcifs
   [root@vm-jh-cent8 lwcifs]#
   ```

10. Next, type the following command to create a new symbolic link inside the /lwcifs directory (targeting the StorHouse/RFS mount point):

    ```
    ln –s /RFS RFS
    ```

    The previous command example uses the name `RFS` for the new symbolic link, however, you may use another suitable name for your symbolic link if so

desired. Further steps in this document will refer to the `RFS` name for consistency.

```
[root@vm-jh-cent8 lwcifs]# ln -s /RFS RFS
[root@vm-jh-cent8 lwcifs]#
```

11. Type the following command to change permissions on the link, thus making the new /lwcifs/RFS symbolic link available to all users:

    `chown –h <username>:<groupname> /lwcifs/RFS`

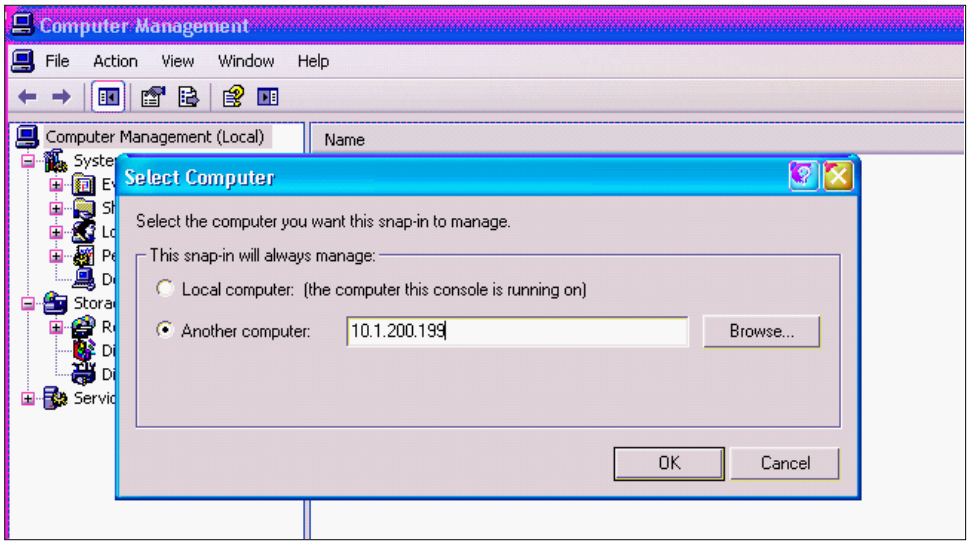
    - In this example, the username and group name represent an existing Active Directory user account and group identifier to be associated with `/lwcifs/RFS`.
    - The username format should be `DomainName\\UserName`.
    - The group name format should be `DomainName\\GroupName`.

```
[root@vm-jh-cent6 /]# chown –h
MyCompany\\AdminUserAccount:MyCompany\\LikewiseGroup /lwcifs/RFS
```

    You may also wish to modify the **RWX** permissions on the link to meet your individual system requirements. Unless you install IDMU (ID Management for UNIX) (see Appendix A), there will be no User ID or Group ID mapping between Windows users and Linux users.

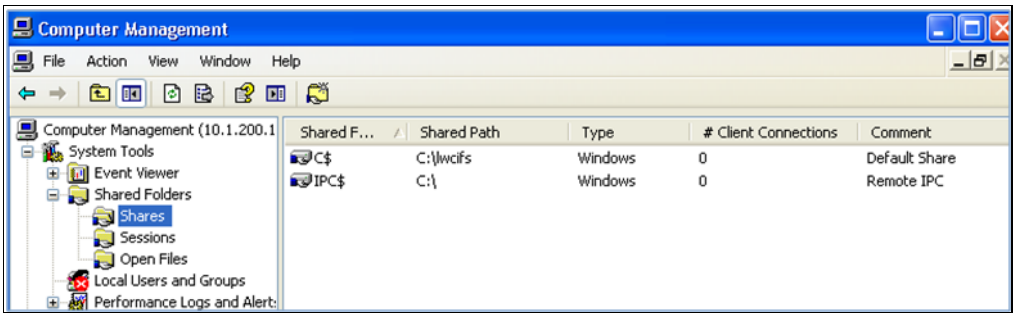
12. Now create a new share accessible from Windows client machines on your Active Directory domain. *You must perform this step using Domain Administrator privileges.* From a Windows client system that is a member of the same domain as your StorHouse/CIFS system, right-click on **My Computer** and select **Manage** to open the **Computer Management** dialog.

    a) Right-click on **Computer Management (Local)** in the left-hand column and select **Connect to another machine**. Then enter the IP address or hostname of the StorHouse/CIFS system, and click **OK**.

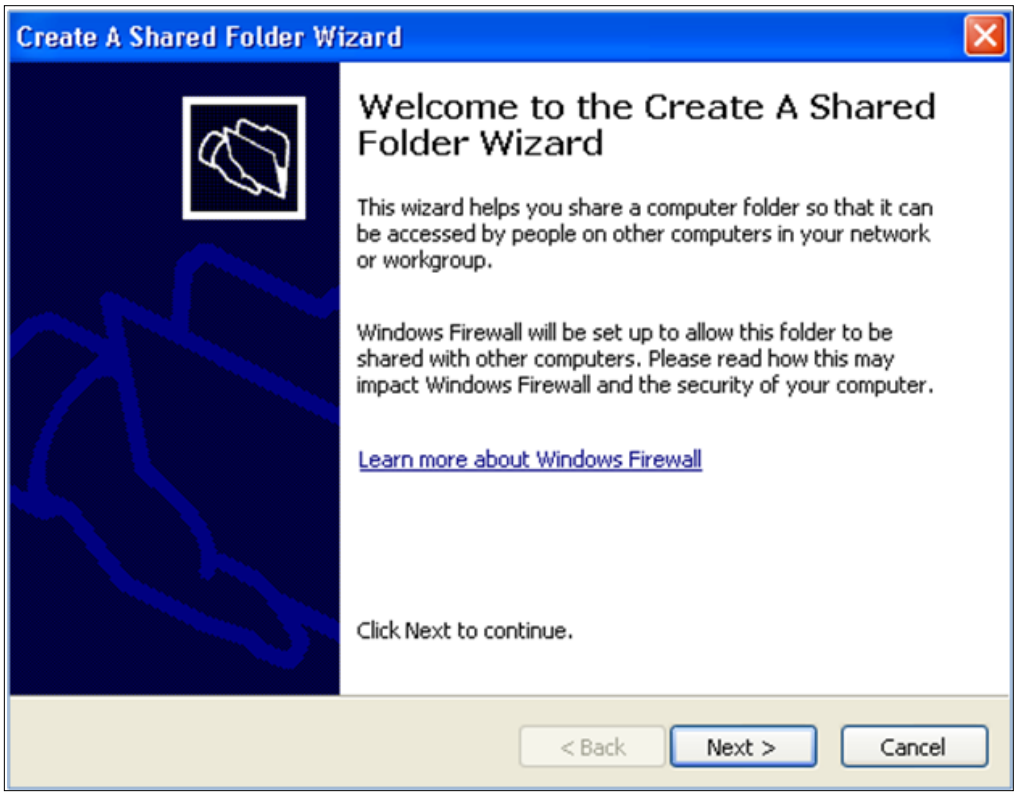


b) If this fails to connect, recheck the status of the srvsvc daemon on the StorHouse/CIFS system. (See steps 6 and 7 for information on starting the srvsvc daemon.)

c) When you have established a connection, open the **System Tools → Shared Folders → Shares** view. At this point, you may receive an **Access Denied** error message. You can safely ignore this alert. It is caused because the Windows computer management system is attempting to connect to certain services on the CIFS system, and these services exist only on Windows systems (for example, the Event Viewer).



d) Right-click on the **Shares** folder selection in the left-hand tree, and select **New File Share**, which opens the Create a Shared Folder Wizard. Click **Next** to start the wizard.



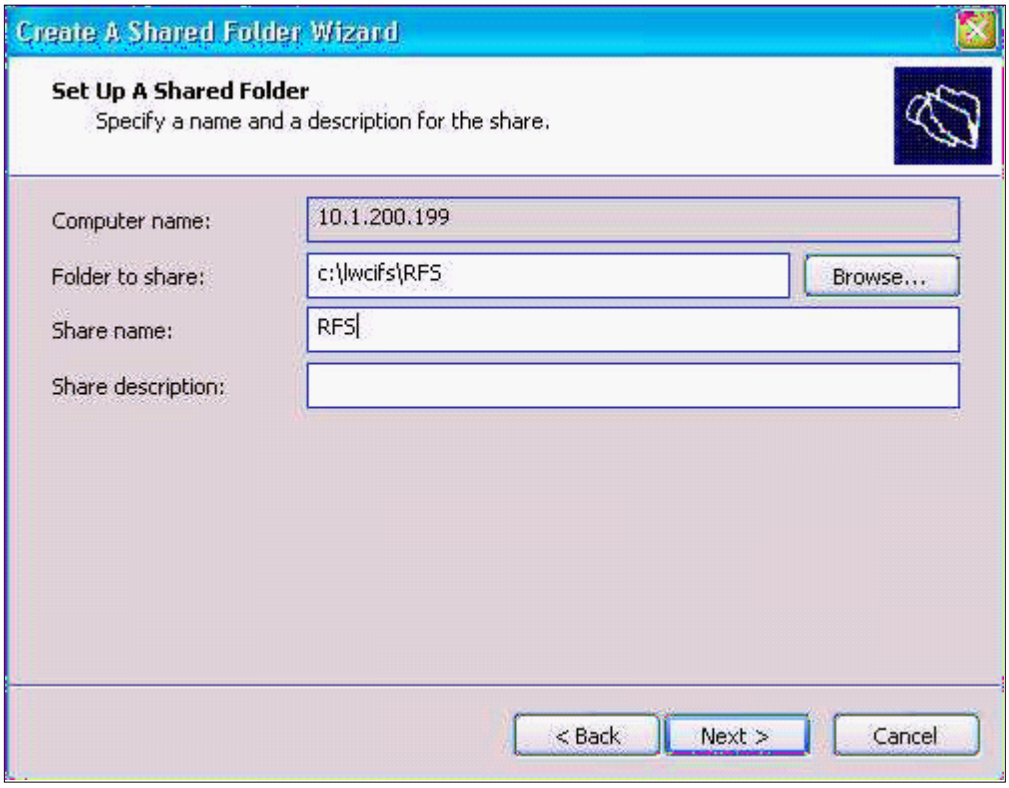
e) Note that steps e and f depend on the operating system of the client making the connection.

On the **Set up a Shared Folder** screen, in the **Folder to Share** text box, enter the path to the share that now exists on the Linux system (for example, `/lwcifs/RFS`).

You must enter the path to the share in the format:

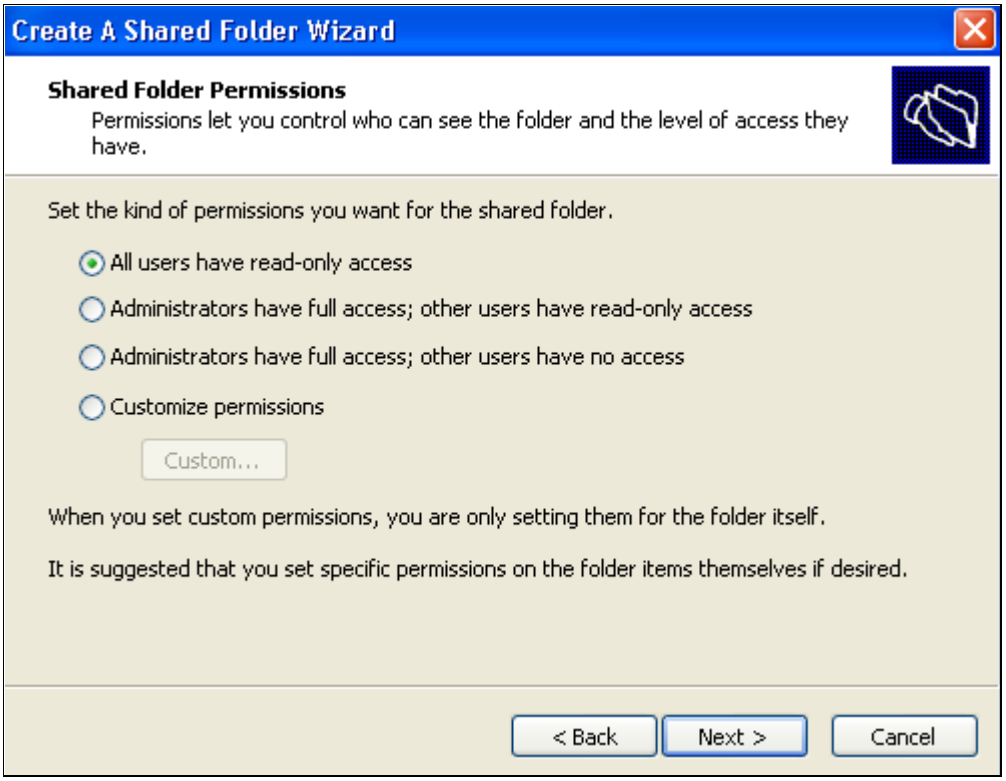
`C:\lwcifs\<Name_of_link_to_RFS> (e.g., C:\lwcifs\RFS)`

Click **Next**.

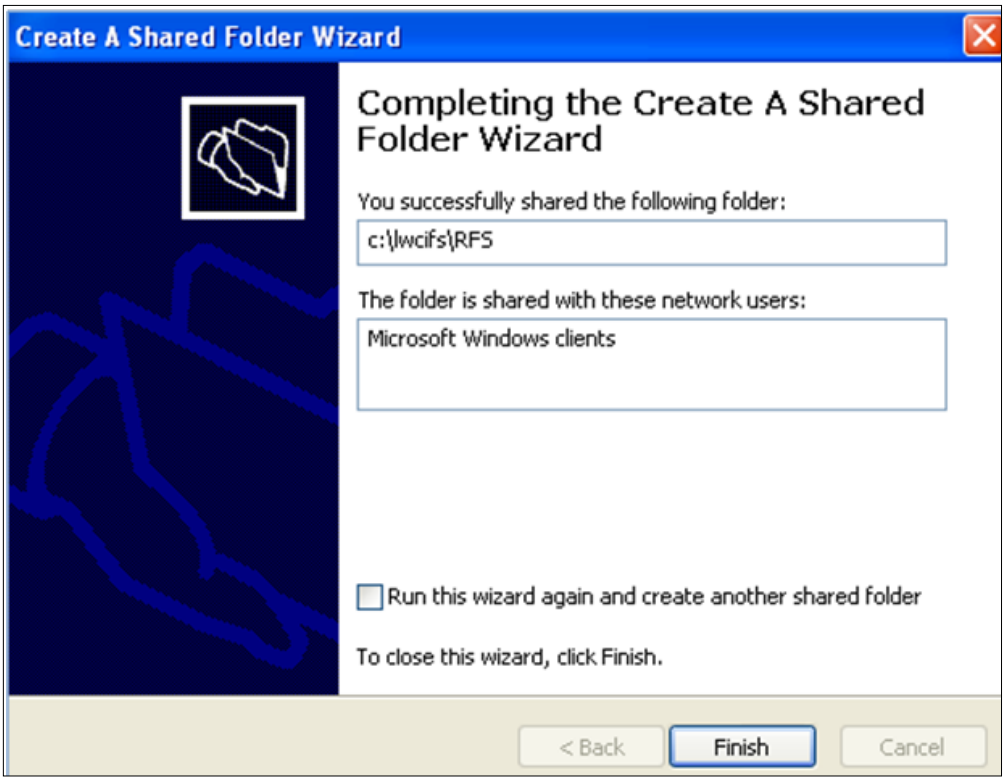


f) On the next wizard screen, assign a Share Name and an optional Description for your new CIFS share. The Share Name in the example is RFS, which is the name all Windows client(s) will use to map network drives or access StorHouse/RFS through the CIFS interface. No change is necessary for other fields such as the Offline Setting value. Click **Next** to advance the wizard.

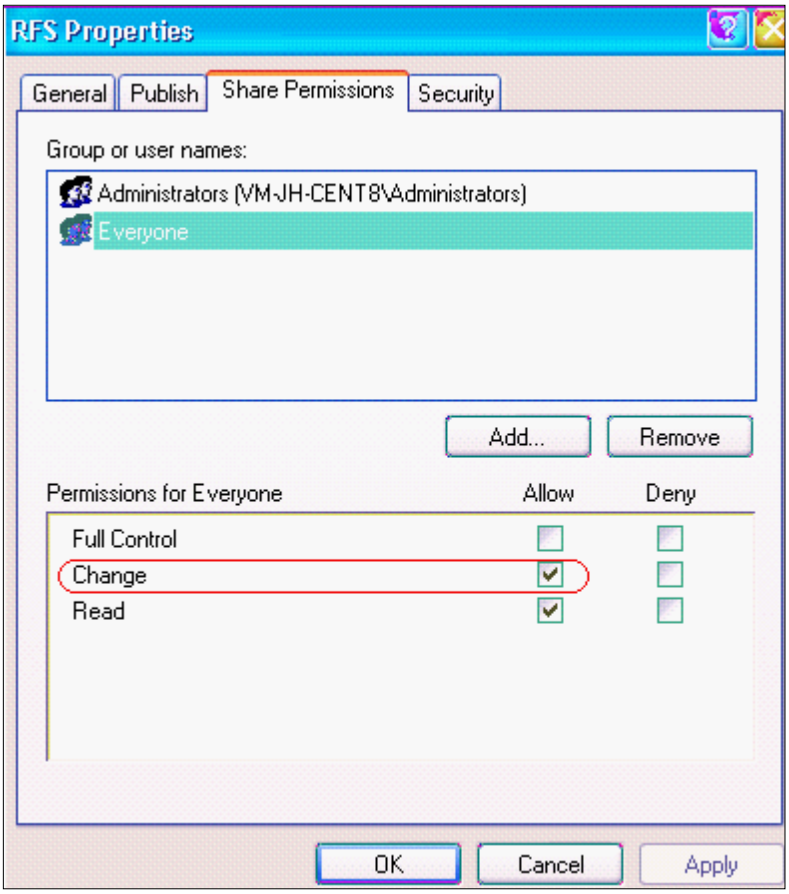
g) On the **Shared Folder Permissions** screen, select the appropriate security level for the new share. Note that the new share itself is read-only by default. You can change this setting to allow administrators and other users full control over the share. Share permissions on the Windows side must match the share permissions established on the Linux side. (Refer to step 11 for information on setting ownership and permissions access on the Linux/CIFS share). Once you have established the necessary share permissions, click **Next** .



h) On the last wizard screen, confirm that your new share was created correctly and there are no errors or warnings reported in the confirmation dialog. Then click **Finish**.



i) Once you have created the share, you must enable users to add new files. Therefore, change the Share permissions to give the Everyone group (or some other defined AD group) the **Change** right. Within the **Shares** window, right-click the newly created StorHouse/RFS share and go to the **Share Permissions** tab. Highlight the **Everyone** group, and select the **Allow** checkbox for the Change permission.



13. From the command prompt on a Windows client, map the newly created StorHouse/CIFS share from Linux to a Windows drive letter using the credentials for the StorHouse/CIFS-Likewise user:

```
 net use Z: \\<LinuxHostName_or_IP>\<name_of_share>
/USER:domain\newusername /PERSISTENT:YES
```

```
C:\Documents and Settings\jhowley_admin>net use Z: \\10.1.200.199\MyRFSShare /US
ER:teqa\jhowley_admin /PERSISTENT:YES
The command completed successfully.
```

At this point, Windows users on this client can now access their `Z:\` drive and browse data in the StorHouse/RFS directory through StorHouse/CIFS.

Uninstalling StorHouse/CIFS

To uninstall StorHouse/CIFS, run the following command:

```
rpm –ev [package_name]
```

Now you are ready to configure your system for StorHouse/CIFS and Active Directory/UNIX network information service interoperability.

C H A P T E R 2

StorHouse/CIFS and Active Directory/UNIX Network Information Service Interoperability

This chapter explains how to configure the system for StorHouse/CIFS and Active Directory/UNIX network information service interoperability.

Introduction

To allow a centralized location for managing user and group accounts across both Windows Active Directory and Linux/UNIX platforms, you can install the Identity Management for UNIX (IdMU) component to a Microsoft Active Directory domain controller running Microsoft Windows Server 2003 R2 and above. The IdMU component enables the Windows Active Directory domain controller to function as a Network Information Service (NIS) server. It also enables Windows users and groups to have associated UNIX attributes that are known to the Active Directory domain. In other words, the user ID (UID), group ID (GID), user home directory, shell settings, and NIS group membership for users can be stored in Active Directory and therefore honored by the StorHouse/CIFS authority service (*lsassd*). In this case, the Windows Server platform hosting IdMU acts as both an Active Directory domain controller and an NIS server.

Overall, this configuration follows the convention for using Active Directory as a Network Information Service (NIS) instance. For additional information about this configuration option, refer to:

http://www.ietf.org/rfc/rfc2307.txt

The IdMU installation process adds a UNIX Attributes tab on each Active Directory User properties dialogue. By default, the NIS Domain dropdown is set to "<none>" for each user (meaning that this Active Directory user does not yet have a correlating NIS account). To configure Active Directory users to also be available to Linux/UNIX client systems as a NIS user, you must select the "NIS Domain" dropdown and choose the NIS domain associated with the Active Directory domain. Once the "NIS Domain" value is set for an individual user, the UID, login shell, home directory, and primary group name/GID will be populated automatically and can be adjusted as necessary.

Figure 2-1 illustrates the UNIX Attributes tab on the user properties dialog.

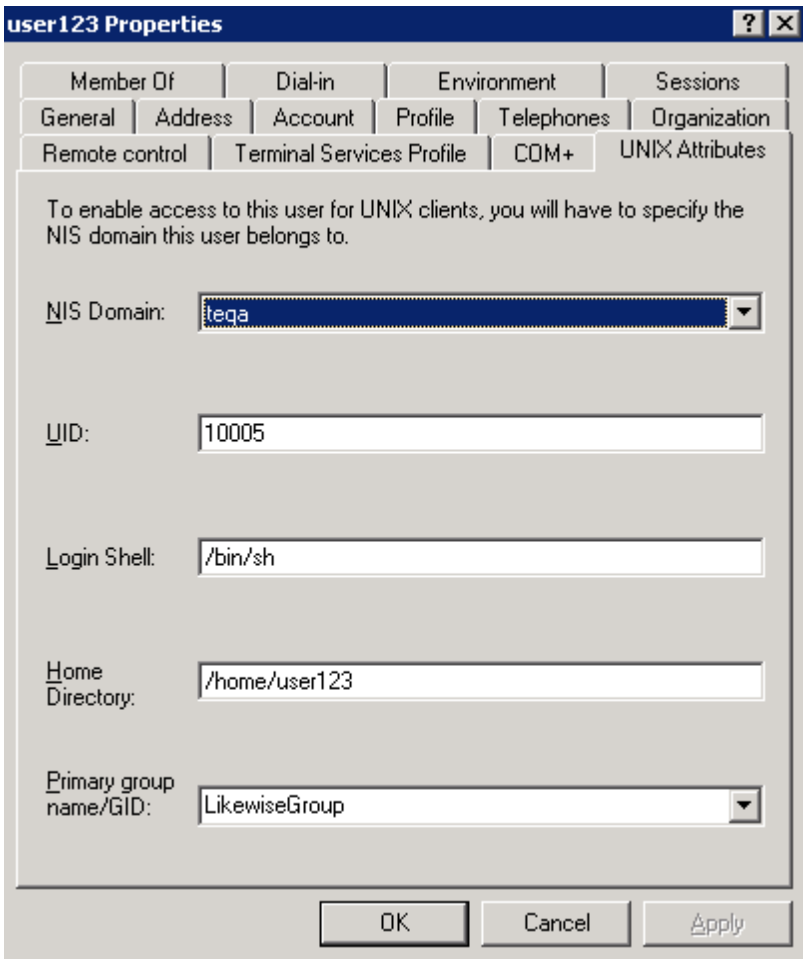


Figure 2-1: UNIX Attributes Tab on User Properties Dialog

Similarly, the installation process adds a UNIX Attributes tab to each Active Directory Group properties dialogue box as illustrated in Figure 2-1.

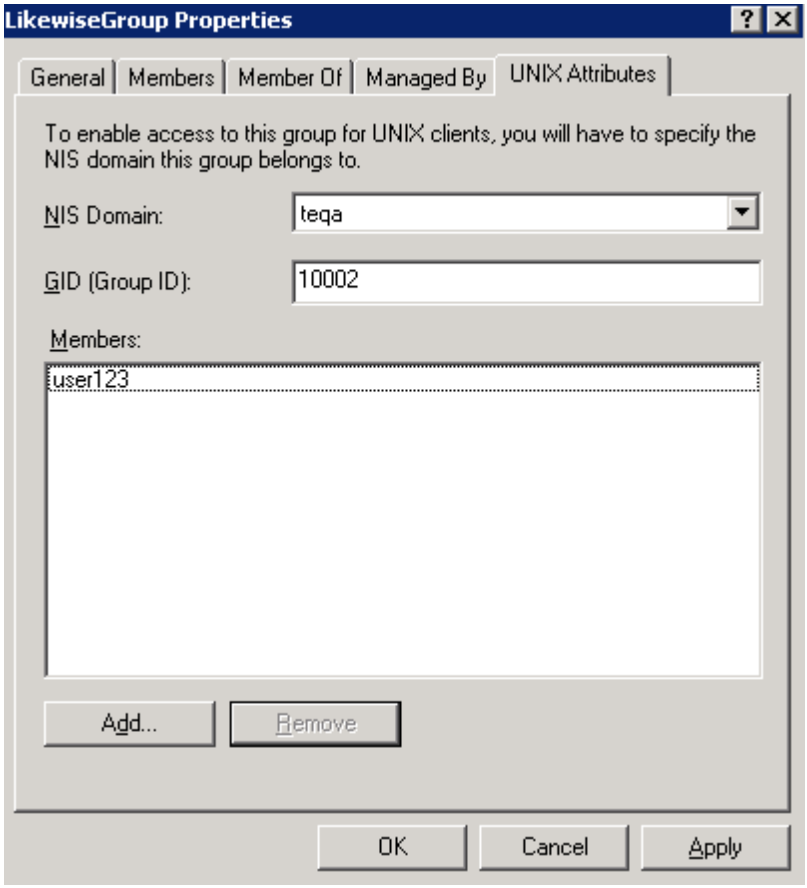


Figure 2-2: UNIX Attributes Tab on Group Properties Dialog

Note: For consistency in applying permissions on a Linux/UNIX disk drive, SGI recommends that the Windows Active Directory User Primary Group Property match the UNIX attribute Primary Group Name/GID. This matching ensures that the rights assigned to new folders and files remain the same whether the end-user accesses the StorHouse/RFS share through a Windows client (via StorHouse/CIFS) or through a UNIX client (via NFS). Figure 2-3 illustrates how to use the same naming convention.

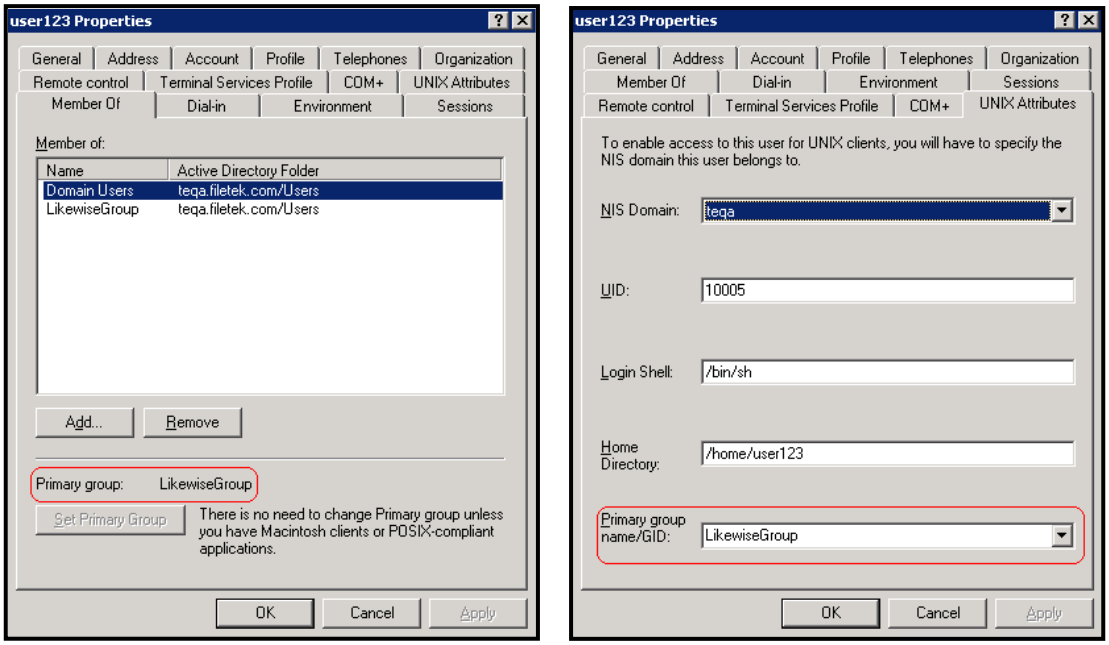


Figure 2-3: Using the Same Naming Convention

Configuration

Once you have assigned Active Directory Users and Groups Linux/UNIX attributes, you must tell StorHouse/CIFS to look for these additional properties. To ensure that StorHouse/CIFS is aware of the Active Directory-to-NIS-user/group mapping, you must change the StorHouse/CIFS registry.

Use the following procedure to change the StorHouse/CIFS registery.

To change the StorHouse/CIFS registery to make StorHouse/CIFS aware of the Active Directory-to-NIS-user/group mapping

1. On the Linux system running StorHouse/CIFS, from an available command prompt, type the following command as root, and press **Enter**.

   ```
   /opt/likewise/bin/lwregshell
   ```

```
[root@vm-jh-cent10 /]# /opt/likewise/bin/lwregshell

\>
```

This command puts you in the StorHouse/CIFS registry shell.

2. At the \> prompt, type the following command to change registry directories to the necessary location and press **Enter**:

   ```
   cd HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory
   ```

3. To add the required registry value, type the following command, and press **Enter**:

   ```
   add_value CellSupport REG_SZ "rfc2307"
   ```

4. Next, type **ls** to confirm that the value was added properly. Then press **Enter**.

```
HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory> add_value CellSupport REG_SZ "rfc2307"

HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory> ls

[\Services\lsass\Parameters\Providers\ActiveDirectory\]
+   "AssumeDefaultDomain"                   REG_DWORD       0x00000000 (0)
+   "CellSupport"                           REG_SZ          "rfc2307"
+   "SyncSystemTime"                        REG_DWORD       0x00000001 (1)
+   "UserDomainPrefix"                      REG_SZ          ""
```

StorHouse/CIFS registry values are written (and stored) in an internal database and are not otherwise viewable on the file system. Note the following:

- CellSupport is the value name.
- REG_SZ is the value type (in this case, a literal string).
- The value of the string is "rfc2307" (in quotes).

Now that the support for RFC2307 schema attributes (between Active Directory and NIS) is enabled, you must exit the registry shell and restart StorHouse/CIFS.

5. To exit the registry shell, type **quit** at the registry shell prompt to return to the file system.

```
HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory> quit
[root@vm-jh-cent10 /]#
```

6. Next, type the following command at the regular shell prompt , and press **Enter**:

   ```
   /opt/likewise/bin/lw-refresh-configuration
   ```

7. To complete the reconfiguration, restart StorHouse/CIF by executing the following command:

   ```
   /opt/likewise/bin/lwsm restart lwio
   ```

```
[root@vm-jh-cent10 /]# /opt/likewise/bin/lwsm restart lwio
Stopping service reverse dependency: srvsvc
Stopping service reverse dependency: srv
Stopping service reverse dependency: lsass
Stopping service reverse dependency: npfs
Stopping service reverse dependency: pvfs
Stopping service reverse dependency: rdr
Stopping service: lwio
Starting service: lwio
Starting service reverse dependency: rdr
Starting service reverse dependency: pvfs
Starting service reverse dependency: npfs
Starting service reverse dependency: lsass
Starting service reverse dependency: srv
Starting service reverse dependency: srvsvc
[root@vm-jh-cent10 /]#
```

Other useful StorHouse/CIFS commands

Table 2-1 lists some useful StorHouse/CIFS commands.

Table 2-1: Useful StorHouse/CIFS Commands

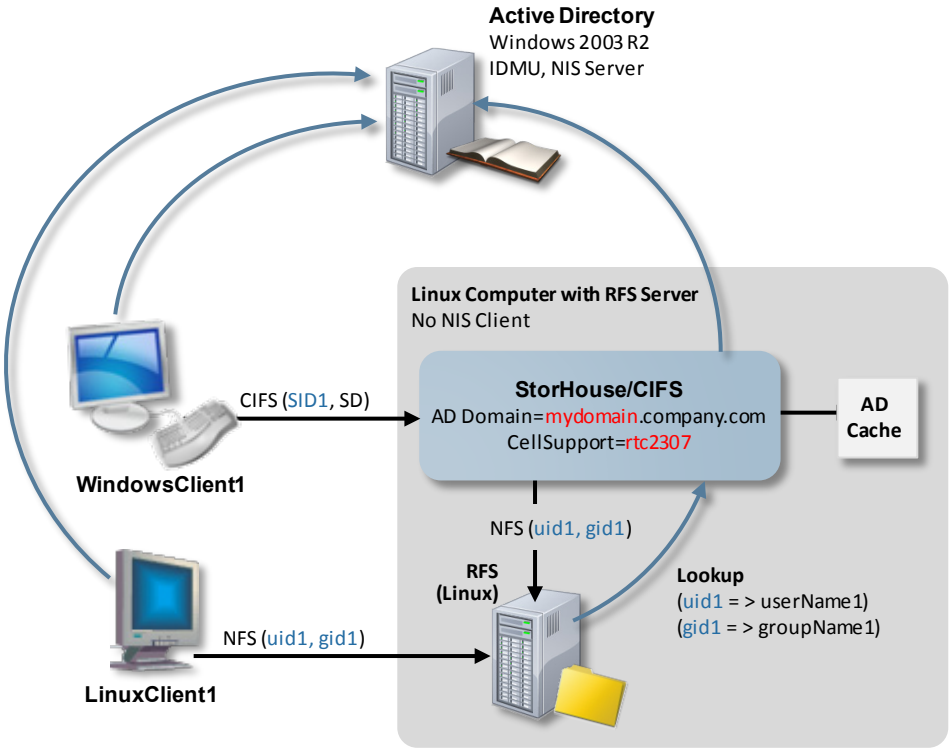
Command	Description
/opt/likewise/bin/lw-ad-cache –delete-all	Purges the Active Directory User and Group cache within StorHouse/CIFS. This command is useful when there have been changes to AD or NIS attributes. Clearing the cache means that StorHouse/CIFS will re-read user and group information when requested from the Active Directory/NIS server (which contains all recent changes).
/opt/likewise/bin/lw-ad-cache --enum-users	Lists all AD/NIS users in cache and the defined attributes for each entry.
/opt/likewise/bin/lw-ad-cache --enum-groups	Lists all AD/NIS groups in cache and the defined attributes for each entry.
id <domain>\\<username>	Performs a user lookup for the stated username and returns the UID, GID (primary group), and group memberships that the domain user is associated with.

Considerations

NIS tracks users based upon their individual User ID (UID) and Group ID (GID). When a Windows user connects to a CIFS share hosted on UNIX, Active Directory manages the credentials. When a UNIX client connects to the same CIFS share, those credentials are passed through StorHouse/CIFS into Active Directory.

The consistency and integrity of the UID/GID assignments depend on the domain administrator. This is an important factor to keep in mind within multiple domain forest configurations. Any user or group assigned a UID or GID 0 (i.e., root) will be

ignored and logged as an error. When RFC2307 mode is enabled, users and groups are only assigned UNIX attributes based on the values stored in Active Directory. Such users and groups are referred to as *provisioned*. Domain user and group objects lacking RFC2307 schema attribute values are referred to as *unprovisioned*. Both *provisioned* and *unprovisioned* users may access files and directories on the StorHouse/CIFS server (as determined by the file system object security descriptors), but only provisioned users can create new file system objects.



Active Directory to NIS User Mapping via StorHouse/CIFS and RFS
Using RFS 2307 schema attributes on Windows 2003 R2 Active Directory domain

Figure 2-1: Active Directory to NIS User Mapping via StorHouse/CIFS and RFS

Chapter 2 – StorHouse/CIFS and Active Directory/UNIX Network Information Service Interoperability

APPENDIX A

Setting Default Permissions on Files/Directories Added to RFS through StorHouse/CIFS

This chapter explains how set default permissions on files/directories added to StorHouse/RFS through StorHouse/CIFS.

About Registry Key Values

Windows permissions are expressed as octal numbers. For example, the octal number 700 represents the default permissions assigned to a new file or folder created through StorHouse/CIFS. This number indicates **rwx** permissions (read-write-execute for owner and no permissions for group or other). To alter these defaults, you must add two keys to the StorHouse/CIFS registery:

- CreateDirectoryMode to indicate the permissions for directories.
- CreateFileMode to indicate the permissions for files.

StorHouse/CIFS stores the registry values for directory and file creation privileges in decimal rather than octal format. So before you can add or change a registry entry, you must know the octal value of the default permissions you want to use. Then you must convert this octal number to its decimal equivalent. You can use an online converter tool or a scientific calculator to perform the conversion task.

For example, the octal value 755 indicates **rwxr-xr-x** permissions (read-write-execute for the owner and read-execute for both group and other). The decimal

equivalent of octal 755 is 493. To set permissions corresponding to 755, you would use the decimal number 493 as the value of the CreateDirectoryMode or CreateFileMode registry entry. Similarly, the octal number 644 indicates **rw-r-r** permissions (read-write for the owner and read for both group and other). The decimal equivalent of octal 644 is 420.

Changing Permissions

You use the `add_value` command to set the initial values for the CreateDirectoryMode and CreateFileMode registry keys according to the following procedure.

To alter default directory and file permissions

1. From an available command prompt on the Linux system running StorHouse/CIFS, type the following command as root, and press **Enter**:

   ```
   /opt/likewise/bin/lwregshell
   ```

   ```
   [root@vm-jh-cent10 /]# /opt/likewise/bin/lwregshell

   \>
   ```

   Once you press Enter, you are in the StorHouse/CIFS registry shell.

2. At the command prompt, type the following to change registry directories to the necessary location, and press **Enter**.

   ```
   cd HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs
   ```

   ```
   \> cd HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs

   HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs>
   ```

3. At the command prompt, type the following to add the CreateDirectoryMode registry key and its associated directory-level permissions. Then press **Enter**. Be sure to enter the value of the REG_DWORD keyword in decimal format.

   ```
   add_value CreateDirectoryMode REG_DWORD <decimal_number>
   ```

```
HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> add_value CreateDirectoryMode REG_DWORD 493

HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs>
```

   In this example, the REG_DWORD value is decimal 493, which is the equivalent of octal 755.

4. At the command prompt, type the following to add the CreateFileMode registry key and its associated file-level permissions. Then press **Enter**. Be sure to enter the value of the REG_DWORD keyword in decimal format.

   ```
   add_value CreateFileMode REG_DWORD <decimal_number>
   ```

```
HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> add_value CreateFileMode REG_DWORD 420

HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs>
```

5. To confirm the changes, type **ls**, and press **Enter**.

```
HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> ls

[\Services\lwio\parameters\Drivers\pvfs\]
+   "CreateDirectoryMode" REG_DWORD       0x000001ed (493)
+   "CreateFileMode"      REG_DWORD       0x000001a4 (420)
+   "Path"                REG_SZ          "/opt/likewise/lib64/libpvfs.sys.so"
+   "PathCacheSize"       REG_DWORD       0x00002800 (10240)


HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs>
```

   The preceding screen illustrates that the permissions for default directory creation have been set to the decimal value 493, which translates to the octal value 755. This octal value reflects a Linux permission level of **rwxr-xr-x**. The permissions for default file creation are now set to the decimal value 420.

6. Once you have set your registry values to the correct equivalents for directory and file permissions, type **Quit**, and press **Enter** to return to the file system:

```
HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> quit
[root@vm-jh-cent10 /]#
```

7. After exiting the StorHouse/CIFS registry shell, enter the following command at the regular shell prompt, and press **Enter**.

   ```
   /opt/likewise/bin/lw-refresh-configuration
   ```

8. Type the following command to restart StorHouse/CIFS, and press **Enter**.

   ```
   /opt/likewise/bin/lwsm restart lwio
   ```

```
[root@vm-jh-cent10 /]# /opt/likewise/bin/lwsm restart lwio
Stopping service reverse dependency: srvsvc
Stopping service reverse dependency: srv
Stopping service reverse dependency: lsass
Stopping service reverse dependency: npfs
Stopping service reverse dependency: pvfs
Stopping service reverse dependency: rdr
Stopping service: lwio
Starting service: lwio
Starting service reverse dependency: rdr
Starting service reverse dependency: pvfs
Starting service reverse dependency: npfs
Starting service reverse dependency: lsass
Starting service reverse dependency: srv
Starting service reverse dependency: srvsvc
[root@vm-jh-cent10 /]#
```

After completing step 8, you have restarted StorHouse/CIFS with updated permission settings for creating directories and files through the StorHouse/CIFS share. Directories and files created in or moved to the StorHouse/CIFS share inherit the permissions defined by the octal equivalent for the CreateDirectoryMode and CreateFileMode registry entries.

Changing Permissions Specified by CreateDirectoryMode and CreateFileMode

Use the `set_value` command to change existing permissions expressed by the CreateDirectoryMode and CreateFileMode registry keys. When using `set_value`, you can omit the REG_DWORD keyword from your declaration.

Here are two examples:

- `set_value CreateDirectoryMode 511`
- `set_value CreateFileMode 416`

```
HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> set_value CreateDirectoryMode 511

HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> set_value CreateFileMode 416

HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs>
```

Once you have set your registry values to the correct equivalents for directory and file permissions, type **quit** to return to the file system.

```
HKEY_THIS_MACHINE\Services\lwio\parameters\Drivers\pvfs> quit
[root@vm-jh-cent10 /]#
```

Then continue with step 7 and step 8 in the procedure entitled “To alter directory and file permissions.”

Appendix A – Setting Default Permissions on Files/Directories Added to RFS through StorHouse/CIFS

A P P E N D I X B

UNIX to Windows Security Rights Mapping

This appendix communicates important information about how StorHouse/CIFS brokers UNIX (POSIX) access rights to Windows client systems using the StorHouse/RFS virtual file system on Linux.

In UNIX (NFS) file systems, you can set only three permissions attributes on any file or directory. Windows (NTFS) file systems have a much more granular approach to access rights and security on directories and files. Figure B-1 illustrates the available Windows and Linux rights.

Available Windows Rights
Full Control
Traverse Folder / Execute File
List Folder / Read Data
Read Attributes
Read Extended Attributes
Create Files / Write Data
Create Folders / Append Data
Write Attributes
Write Extended Attributes
Delete
Read Permissions
Change Permissions
Take Ownership
Delete Subfolders and Files*

Available Linux Rights
Read
Write
Execute

* Directories Only

Figure B-1: Available Windows and Linux Rights

Because StorHouse/RFS on Linux is an NFS file system, the security attributes available for directories and files within StorHouse/RFS are restricted to read-write-execute combinations. Additionally, read-write-execute access is restricted to the owner-group-other model, which allows for access rights defined for one user (owner), one security group, and then all others that have access to the directory or file. Windows (NTFS) file systems do not share this additional property. In other words, there are no limits to the number of individual users or groups that can share access rights to directories and files in NTFS.

While StorHouse/CIFS allows for CIFS-styled access to the StorHouse/RFS virtual file system, this fact does not change the underlying security structure that StorHouse/RFS must maintain within UNIX. The security information on each directory and file within StorHouse/RFS must be kept according to NFS guidelines. However, doing so requires a proper mapping framework so that when permissions are viewed and/or changed through StorHouse/CIFS, the read-write-execute for owner-group-other model is preserved.

Table B-1 shows the correlation between Windows and UNIX (Linux) security attributes. In other words, it shows the permissions selections in the Windows Advanced Security Settings dialog that correspond to the acceptable read-write-execute combination for user (owner), group, or other objects in UNIX.

To maintain a viable **rwx** combination for owner, group, or other class, permissions set through the Windows Security panel must use the following schema:

rwx = read, write, and execute

rw- = read and write (no execute)

r-x = read and execute (no write)

r-- = read only (no write/execute)

-w- = write only (no read/execute)

--x = execute only (no read/write)

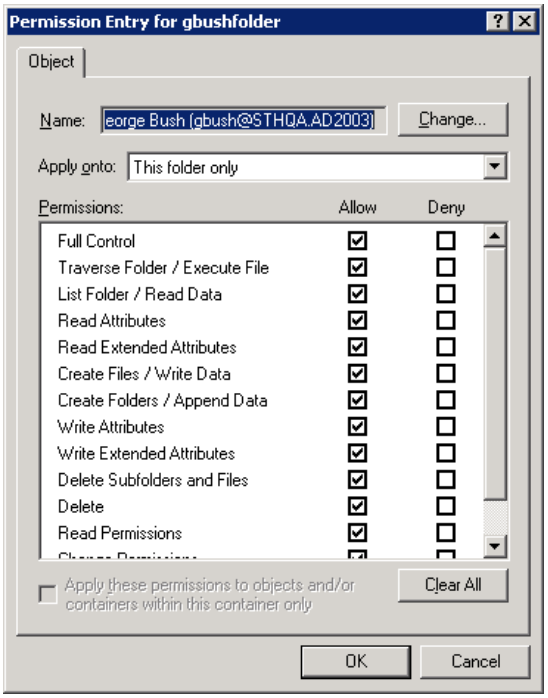
-wx = write and execute

Table B-1: Correlation between Windows and UNIX (Linux) Security Attributes

Windows	Linux Permissions						
	rwx	rw-	r-x	r--	-wx	-w-	--x
Full Control	y						
Traverse Folder / Execute File	y		y		y		y
List Folder / Read Data	y	y	y	y			
Read Attributes	y	y	y	y	y		y
Read Extended Attributes	y	y	y	y			
Create Files / Write Data	y	y			y	y	
Create Folders / Append Data	y	y			y	y	
Write Attributes	y	y			y	y	
Write Extended Attributes	y	y			y	y	
Delete Subfolders and Files*	y	y			y	y	
Delete	y	y			y	y	
Read Permissions	y	y	y	y			
Change Permissions	y						
Take Ownership	y						

*Available for directory permissions only

For Windows, SGI assumes that you make permissions changes through the Windows Advanced Security Settings dialog for any StorHouse/RFS directory or file accessed through StorHouse/CIFS:



Setting the permissions to **Full Control** for the user (owner) of a directory or file correlates to full **rwx** permissions for the owner in NFS. Conversely, when setting the permissions for the owner in NFS to **rwx**, viewing the permissions for the owner through Windows Advanced Security Settings shows that this user has **Full Control** with all associated rights granted. You can set permissions in Windows (through StorHouse/CIFS) as long as there is a read-write-execute combination that correlates to the security selections made in the Windows user interface.

Note that you access the UNIX **other** object in Windows as **Everyone**. Permissions granted to **other** through UNIX will be associated with the **Everyone** object in Windows and vice-versa.

If you attempt a mix of permissions that does not fit into an acceptable read-write-execute combination through Windows, StorHouse/CIFS will not allow the action. Instead, it will force the permissions set to the closest applicable **rwx** combination. Furthermore, StorHouse/CIFS disallows any attempt to define permissions for more than one individual user or group through Windows. In this case, your permission settings will be reset to the last good configuration with a single owner and group. If you need to change the owner or group for a directory or file in StorHouse/RFS, you must do so through NFS.

Only the owner/user of a directory or file can change permissions through StorHouse/CIFS, even if members of the defined group with appropriate rights through NFS are accessing the object through Windows. SGI is considering adding

the ability to change permissions through StorHouse/CIFS as a member of a permitted group (but not the owner) for a future StorHouse/CIFS release. Currently, this feature is available only through action on the Linux/NFS side.

APPENDIX C

More Information about StorHouse/CIFS Permissions

This appendix provides more information about StorHouse/CIFS Permissions in tabular format.

Table C-1: Windows Rights Based on CIFS Access Permissions

Windows	SCIFS Code Permission
Full Control	FILE_ALL_ACCESS
Traverse Folder / Execute File	FILE_EXECUTE(file)/FILE_TRAVERSE(directory)
List Folder / Read Data	FILE_READ_DATA(file, pipe)/FILE_LISTDIRECTORY(directory)
Read Attributes	FILE_READ_ATTRIBUTES (file, pipe, directory)
Read Extended Attributes	FILE_READ_EA(file, directory)
Create Files / Write Data	FILE_WRITE_DATA (file, pipe)/FILE_ADD_FILE(directory)
Create Folders / Append Data	FILE_APPEND_DATA(file)/FILE_ADD_SUBDIRECTORY(directory)/ FILE_CREAT_PIPE_INSTANCE(pipe)
Write Attributes	FILE_WRITE_ATTRIBUTES(file, pipe, directory)
Write Extended Attributes	FILE_WRITE_EA(file, directory)
Delete	DELETE
Read Permissions	READ_CONTROL
Change Permissions	WRITE_DAC

Table C-2: Map POSIX Mode to Windows Access Rights for Files

PvfsSecurityAccessMapFromPosix

File Rights in ACL

OwnerSID

Posix Mode						Windows Access Rights
S_ISUID 4000	S_ISGID 2000	S_ISVTX 1000	S_IRUSR 0400	S_IWUSR 0200	S_IXUSR 0100	
						+ READ_CONTROL + WRITE_DAC + WRITE_OWNER
			x			+ FILE_GENERIC_READ
				x		+ FILE_GENERIC_WRITE + DELETE
					x	+ FILE_GENERIC_EXECUTE
			x	x	x	FILE_ALL_ACCESS & ~(WRITE_OWNER|WRITE_DAC)

GroupSID

Posix Mode						Windows Access Rights
S_ISUID 4000	S_ISGID 2000	S_ISVTX 1000	S_IRGRP 0040	S_IWGRP 0020	S_IXGRP 0010	
			x			+ FILE_GENERIC_READ
				x		+ FILE_GENERIC_WRITE + DELETE
					x	+ FILE_GENERIC_EXECUTE
	x					+ WRITE_DAC
			x	x	x	FILE_ALL_ACCESS & ~(WRITE_OWNER|WRITE_DAC)

EveryoneSID (S-1-1-0)

Posix Mode						Windows Access Rights
S_ISUID 4000	S_ISGID 2000	S_ISVTX 1000	S_IROTH 0004	S_IWOTH 0002	S_IXOTH 0001	
			x			+ FILE_GENERIC_READ
				x		+ FILE_GENERIC_WRITE + DELETE
					x	+ FILE_GENERIC_EXECUTE
			x	x	x	FILE_ALL_ACCESS & ~(WRITE_OWNER|WRITE_DAC)

Table C-3: Map POSIX Mode to Windows Access Rights for Directories

PvfsSecurityAccessMapFromPosix

Directory Rights in ACL

OwnerSID

Posix Mode						Windows Access Rights
S_ISUID 4000	S_ISGID 2000	S_ISVTX 1000	S_IRUSR 0400	S_IWUSR 0200	S_IXUSR 0100	
						+ READ_CONTROL + WRITE_DAC + WRITE_OWNER
			x			+ FILE_GENERIC_READ + FILE_LIST_DIRECTORY
				x		+ FILE_GENERIC_WRITE + DELETE + FILE_DELETE_CHILD + FILE_ADD_SUBDIRECTORY
					x	+ FILE_GENERIC_EXECUTE, FILE_TRAVERSE
			x	x	x	FILE_ALL_ACCESS & ~(WRITE_OWNER|WRITE_DAC)

GroupSID

Posix Mode						Windows Access Rights
S_ISUID 4000	S_ISGID 2000	S_ISVTX 1000	S_IRGRP 0040	S_IWGRP 0020	S_IXGRP 0010	
			x			+ FILE_GENERIC_READ + FILE_LIST_DIRECTORY
				x		+ FILE_GENERIC_WRITE + DELETE + FILE_DELETE_CHILD + FILE_ADD_SUBDIRECTORY
					x	+ FILE_GENERIC_EXECUTE + FILE_TRAVERSE
	x					+ WRITE_DAC
			x	x	x	FILE_ALL_ACCESS & ~(WRITE_OWNER|WRITE_DAC)

EveryoneSID (S-1-1-0)

Posix Mode						Windows Access Rights
S_ISUID 4000	S_ISGID 2000	S_ISVTX 1000	S_IROTH 0004	S_IWOTH 0002	S_IXOTH 0001	
			x			+ FILE_GENERIC_READ, FILE_LIST_DIRECTORY
				x		+ FILE_GENERIC_WRITE + DELETE + FILE_DELETE_CHILD + FILE_ADD_SUBDIRECTORY
					x	+ FILE_GENERIC_EXECUTE + FILE_TRAVERSE
			x	x	x	FILE_ALL_ACCESS & ~(WRITE_OWNER|WRITE_DAC)