

Unix/Linux interoperability components in Windows (SUA, IdMU, NFS, etc.)

Tech Note #1 in the "Interop Components in Windows" series

By Rodney Ruddock (Interop Systems)

The split-up of Windows Services for UNIX into individual components in Windows Server and Windows Vista has left users confused as to how to set up the ultimate Unix/Linux interoperability environment.

Whither Services for Unix (SFU)?

Way back in 1999 when Microsoft acquired Interix from Softway Systems many people predicted that Interix would get buried by the Windows people. After all, Interix is a Unix system to be run on the Windows OS! Conspiracy theorists were having a heyday.

It took a few years before Services for Unix (SFU) version 3.0 was released with Interix, NFS client, NFS server, User Name Management (NIS) and NFS Gateway to provide a comprehensive Unix solution on the Windows OS. SFU 3.5 was the next release with even more functionality. For many people the 3.5 release was very exciting because SFU became freely available for everyone. Clearly Interix and SFU were not getting buried. Microsoft was not only keeping SFU going, but was improving and expanding it. The additional Unix functionality was also getting more and more popular with IT departments working in a mixed OS environment.

Then Microsoft announced that most of the components of SFU were being integrated into the Windows OS releases. Not only is it all free, but available as part of the base installation of Windows. This started with the release of Windows Server 2003/R2 and is now included with Windows Server 2008. Of course it meant that SFU would have no more releases as that would be redundant. SFU 3.5 is still available as a free download (for Windows XP, Windows 2000 and Windows Server 2003 pre-R2 users) until 2009 with support continuing until the year 2011.

It's not so much that SFU has become end-of-life, but that it has been allowed to advance to the next level. By being part of the Windows OS release everything that was SFU should now be taken even more seriously. With Vista, the components can be found in the Ultimate and Enterprise versions.

Being integrated into the Windows OS release meant some name changes for some of these features. NFS client and server are still NFS client and server. User Name Mapping has been renamed Identity Management. Interix now goes by the moniker SUA (Subsystem for Unix-based Applications) but the system will continue to identify itself as "Interix" for clear compatibility with programs, scripts and makefiles.

Installing SUA, NFS, etc. on Windows OS

To install any of these components/features on Windows Server or Vista is very simple. No extra CDs or DVDs are required. From the Control Panel, start the "Program and Features" application (formerly known as "Add/Remove Programs"). Then select from the left panel "Turn Windows features on or off." After a few seconds a popup window will list all of the currently active features. Scroll through the list to make sure the features you are interested in having active are check marked. Some of the features are only available on the Server versions. Here's the list as it appears on Vista:

- Services for NFS to install NFS client and matching administrative tools
- Subsystem for Unix-based Applications for SUA/Interix

Here's the list as it appears on Windows Server 2008 and Windows Server 2003 R2. For SUA the selection is the same as with Vista. For the other components you need to select "File and Print Services" to get the next submenu:

- NFS Admin
- NFS Client
- NFS Server
- Server for NFS authentication
- Identity Management for Unix (IdMU)

Once these features have been installed you will need to reboot the machine. After the reboot you will have some configuration to do that will vary depending on which features you chose. If you chose SUA then you will need to download the Utilities and SDK package to get the shells, utilities, libraries and manual pages. This can be done easily from the Start Menu under All Programs, Subsystem for Unix-based Applications.

When you install NFS Server it is always recommended that you install Identity Management (formerly known as User Name Mapping; UNM) so that Windows IDs can be mapped to Unix IDs and the reverse.

What is Interix/SUA?

Interix is the Unix-like system that runs on the Windows OS. It runs as a peer system to the Win32 environment that most people simply call Windows. Interix has the same access to the NT kernel, the file systems, networking and security systems that Win32 does.

The development of Interix began in the early 1990s with Softway Systems under the product name OpenNT. Softway developed the Interix system and utilities with the goal of meeting the Unix specification and defacto standards. While Microsoft had originally developed a strictly POSIX compliant system for NT 3.0, it lacked utilities and many supporting features needed beyond POSIX to make a Unix system. Examples of de facto standards that are incorporated into Interix are Berkeley sockets and pseudo-terminals. Other standards included the X/Open standards (e.g. XCU) and matching test suites.

The resulting environment of functionality meant that when you logged into an Interix system by, for example, telnet, the shell and utilities gave the same feel as running on a BSD based system. With over 350 utilities plus the SDK for developers covering X-Windows, Motif, OpenGL, libc, etc. Interix became another Unix-like platform that Unix software could be ported to.

A big win is that it's on the same machine that's running Win32. This means that on one machine Win32 and Unix applications can run side-by-side. Cut and paste from an X-Window to a Word document with ease. Collect input from a Win32 program and analyze it with a set of Unix tools – or the reverse. One of the early examples of inter-system communication used the Unix 'tides' program to generate information placed into an Excel spreadsheet.

For administrators this gives them the ability to have a common base of tools and functionality available across a heterogeneous computing environment. Connecting by telnet or ssh to and from all machines is good. Even better is being able to run the same scripts on Interix as on Solaris or Linux to complete a task. No more getting bogged down in a GUI or wasting time repeating tasks that can be automated in scripts.

What is NFS?

NFS is the Network File System developed by Sun Microsystems in the 1980s for sharing disk drives across a network. It was developed for use with Unix systems. It also became a de facto standard at many sites for providing network disks to PCs as well. NFS has an excellent track record for network drive access as well as management of these resources.

Meanwhile on Windows, SMB evolved (now called CIFS) to be a de facto standard for Windows machines. It has some shortcomings that have never been addressed. The advent of SAMBA (on Unix) was an attempt to bridge SMB shared disks to Unix machines. However, SAMBA cannot overcome the shortcomings of SMB. A couple of these shortcomings include the lack of case-sensitivity in file and directory names, and inconsistent file information. These shortcomings make it difficult to use SMB in a mixed OS environment. On the other hand, NFS has shown itself to work better in this mixed OS environment.

Coupled with the use of NFS on Windows is User Name Mapping (UNM). This allows the mapping of Windows Security IDs to Unix IDs both to and from Windows. By doing this the Windows system communicates with the Unix system in the same, common language for security access to the files. The mapping of the IDs can be done in a simple manner and also in more complex arrangements with multiple systems. Without UNM the access to the disks must be done in an anonymous mode that gives a reduced security for the shared disks.

Starting with Windows Server 2003/R2 the UNM became part of Active Directory rather than as a stand-alone service.

Identity Management for Unix (IdMU)

Identity Management for Unix (IdMU) is a merging of what was previously known as Server for NIS (Network Information System) and User Name Mapping (UNM). Starting with Windows Server 2003/R2 this is integrated with Active Directory (AD). This allows for the control of NIS domains from a Windows machine. IdMU does not allow for a Windows machine or an AD domain to be placed under the control of a Unix NIS server though. Included with IdMU is Password Synchronization to allow AD and NIS to coordinate password updates for more uniform user control in a heterogeneous environment.

With IdMU the network of machines with a heterogeneous mix of operating systems can have a single list of users that have the same password to access all of the machines. It also means that users can change their passwords on any of the machines and have the result reflected in all of the other machines on network. The resulting ease for users and system administrators makes with this a win-win tool.

The identity mapping capabilities provide the SID to UID mapping for NFS communications.

For more detailed information there is a separate Tech Note written about installing and configuring IdMU.

Setting up a Working Environment with SUA/Interix

With SUA/Interix installed as one of the additional Windows OS features the next order of business is to install the commands and utilities for command line usage. At the same time, optionally, the Software Development Kit (SDK) can be installed if you will be doing software development. From the Start menu, under All Programs then under Subsystem for Unix-based Applications you can find the link that will automatically retrieve the commands and SDK from the Microsoft web site.

When doing the installation there is the option of working with the default installation or using a custom installation. The custom installation gives you the best selection for what to install or not install without a lot of complication. For strictly command line usage the installation of the "Base Utilities" and "GNU utilities" together gives the widest set of programs. The "SVR5 Utilities" are based on the Unix System V utilities and are not as commonly used unless the site has a strong System V emphasis already. The Base and GNU utilities provide the programs with options that most Unix and Linux users already will be familiar with.

The SDK, similar to the utilities, has two selections: "Base SDK" and "GNU SDK." The Base SDK is a must for developers to install because this provides all of the needed include files, SDK manual pages, static libraries and

utilities for building applications. The Base SDK includes wrapper scripts (c89 & cc) to work with Microsoft Visual Studio's C/C++ compiler (MSVC). It's recommended to have MSVC installed before installing the SDK so that the SDK sees MSVC and does some automatic configuration. The GNU SDK will install the gcc compiler suite (which includes g++) and shared libraries. With gcc you will be able to create Unix-style shared libraries. Included are the X11 libraries used for creating Unix GUI applications.

If you run into any difficulty installing SUA, you can refer to the second Tech Note titled *Installing SUA Commands, Utilities and Libraries* where step-by-step screenshots are included.

For both users and developers additional programs and libraries from the F/OSS world are desired such as bash, OpenSSH and an X-server. These items don't normally ship with SUA/Interix. However, they can easily be obtained from third-parties at several different levels (free, low-cost and premium) depending on the application. One of the key sites for getting these third-party programs is the 15,000-member SUA Community website where many applications have been prepared in ready-to-go binary packages (for more details visit the site www.interopsystems.com/community). This site is operated independently by Interop Systems, with a financial contribution from Microsoft.

For the best results in application operations and security some key choices at installation should be made. During the installation phase you should choose, when asked, to have case sensitive pathnames turned on and the SetUid ability also turned on. For case-sensitive pathnames there are some key things to note:

1. Turning this setting to "on" will only affect SUA/Interix. It will not affect Win32.
2. All Unix applications expect a case-sensitive environment. Best behavior from your Unix applications happens with case-sensitivity on.

The SetUid ability is the special ability on Unix systems to have certain applications run with the authority and privileges of a specific user – usually the owner of the application. This is a very powerful ability. And with great ability comes the responsibility of being careful to not create a security problem. The tremendous, positive results of having this ability easily outweigh the responsibilities in the vast majority of situations. SetUid allows for the Administrator (and only the Administrator, not members of the Administrators group) to get certain special tasks done quickly and easily at the local machine or through a remote connection such as telnet or ssh. Also, it allows for the automation of many tasks in the background on behalf of the users. An example of this is setting up automated tasks with the cron utility.

The overall footprint of a full installation plus several dozen third party applications is relatively small by today's terms. A full install includes all user applications and SDK programs, files and libraries is less than a full CD. That's a tiny amount of disk space these days. So there's little worry about filling a disk if you want to install everything. Installing everything provides pretty much the default installation on most Unix-like systems.

One of the best and highly recommended things to do is to explicitly set a home directory in the user database for each user. This provides the highest amount of security for users for their files and for applications that need ensured trust when running. When users do not have an explicit directory in the user database (Active Directory is an example) then spoofing and Trojan Horses can be easily done. There are also applications that need to know that the user information they get is 100% trustworthy. A key example of this required trust are the SSH applications. Public keys must be stored on an individual user basis – sharing is not a secure option. Individual home directories, with correct file permissions, provide the ability to store this information. Obtaining the location of the users' home directories from the trusted user database where the information is stored is critical to security. If the chain of trust cannot be assured then trust cannot be ensured and the security-base applications refuse to work.

Of course the mark of good administration is one that automates as much as possible. If it's a large task or a task that gets repeated more than once then scripting the job is the way to go. The setting of user home directories in the user database can be automated. Adding new users from a list can be automated. The scripts can be done in a variety of scripting languages such as shell, Perl, Tcl, PHP and Python. The available scripting choices make it easy to get a script done because a new language does not need to be learned. Having familiar tools available on SUA/Interix that work and behave as on all other platforms is central to making things easy for IT shops in a heterogeneous computer environment.

Setting up NFS Server

Having NFS working with or without SUA/Interix installed is a plus in a heterogeneous network of machines. This common file access method is fast for the users and easy to setup by the administrator. There are two halves to the NFS setup. There is the server-side, where the disk lives, and the client-side, where the access requests will come from.

The server-side of NFS can come from a Unix machine or from Windows machine providing NFS. To NFS clients on the network either of these types of NFS servers will appear virtually indistinguishable. If it provides NFS services then the NFS client will be happy. For a Unix machine providing NFS services, refer to that machine's NFS server documentation. For the Windows NFS server the vast majority of installations can happen with straight-forward instructions. Actually it's prudent to mention that this information is available with the on-line help too. As mentioned earlier the NFS server component will be installed from the list of Windows Features via the Program and Features on the Control Panel.

From a Windows machine hosting NFS Server the simplest way to share a disk is to start Windows Explorer. Then choose the disk or directory that you want to share by NFS Server. Right click your selection to open the Properties panel. Then choose the NFS Server tab. Now enter a name that the NFS share will be known by for any NFS clients and apply this setting. You may optionally want to allow anonymous access to this share.

For users to access this NFS share two additional actions must happen. One action, described in the next section, is the NFS client must mount the share. Action number two is the user's identity must be mapped. As mentioned earlier this identity mapping used to be known as Username Mapping (UNM). Starting with Windows Server 2003/R2 this mapping is part of Active Directory. A later section outlines the name mapping.

Setting up NFS Client

Installing the NFS Client side is, without surprise, done through the Windows Features list via the Program and Features application launched from the Control Panel. Scroll through the features list and make sure NFS Client is checked. This installs an NFS Client driver on the machine. This client driver handles the communication to the NFS server and also handles presenting the NFS disk to the local system like it is just another disk.

Before mounting an NFS disk to a client it is recommended that you configure the identity mapping for the client-side. The NFS client-side can use the older User Name Mapping (provided with SFU) or use AD (starting with Windows Server 2003/R2 and now with Windows Server 2008). A special administrative panel is used for the mapping configuration. From the Control Panel you select Administrative Tools and then select "Services of Network File System (NFS)". A new panel will open from which you select "Client for NFS" and open the properties for it (by right clicking the mouse). From here you choose to use AD or UNM or both for the mapping.

Mounting, or connecting, an NFS disk from an NFS Server is done in the same manner that a disk from a Windows machine is mounted. From a Windows Explorer Tool menu select Map Network Drive. As normal, you can enter the

details directly if you know them or you can browse with a GUI to get the right NFS disk to be mounted. And without any more fanfare the job of getting an NFS disk mounted is done.

Alternatively, if you wish, you can mount NFS disks using command line utilities. More information on this is provided with the on-line help.

Ideally the user using the NFS disk is accessing the data on that disk as that user. That happens with user information being transmitted between the NFS client and the NFS server. Between machines the identity of some of the users may vary somewhat resulting in the NFS server not recognizing the user. In this case it is typical that access to the NFS disk will be provided, but the user will be making each access request as an anonymous user. This makes it difficult for users wanting to write data or read data that has even a mild amount of permission restriction. The solution for this is to have the identity mapping configured (what used to be called User Name Mapping). This maps user identities on the NFS Client machine to appropriate user identities for the NFS Server machine resulting in the expected rights for data access. If you do decide that you want to have user access as an anonymous user you must specifically set this as allowable with the NFS Server since, for security reasons, this access mode is typically not allowed by default.

Name Mapping Outlined

Name Mapping allows for the mapping of a user identity in one domain or on one machine to another user identity that is recognized on another machine. This provides a key element with NFS disk support particularly critical in a mixed OS environment.

Windows OS machines use SIDs (Security Identifiers) for user and group identities. Unix machines use UIDs (User IDs) and GIDs for user and group identities respectively. NFS Server and NFS client communicate using UIDs and GIDs. This is based on the original design of NFS working with Unix machines. Thus sending SIDs from an NFS Client to an NFS Server is not going to work with the protocol. The mapping of a SID to a UID or GID is performed in a manner that is seamless to NFS and to applications.

Starting with Windows Server 2003/R2 and now with Windows Server 2008 an LDAP store (such as Active Directory) that is RFC 2307 compliant can now be consulted to get the mapping to UIDs and GIDs. Previously the mapping was done with UNM. The NFS clients can use either or both mapping providers as described in the section above "Setting Up NFS Client."

Configuring the mapping on AD requires the administrator to add a specific mapping identity for each user. This is done through the Identity Management feature which also handles the NIS integration. Through AD a user is chosen and the Properties panel is opened at the "UNIX Attributes" panel. In the UID box new user ID is entered. This value is what will be provided to the NFS client when asking for a SID to be mapped to a UID. The NFS communication will use this UID value. The NFS Server in turn will get this UID mapped to a SID when it receives the NFS request. The reverse happens when the server sends responses to the NFS client.

The component UNM is no longer available after SFU 3.5 because the ability of NFS to map the identities is now being provided by any RFC2307 compliant LDAP such as Active Directory. In fact you can do this with Windows 2003 Server R2 too – and Windows Server 2008. But for Windows XP and Windows 2003 Server pre-R2 the UNM services must be used. The NFS client with Vista is flexible enough to use one or both methods

Feel free to post a question in the [SUA Community](#) forums.