ABSTRACT
Do you ever issue x commands on the server, which run successfully, only to find the same commands fail on your local PC? Do you cringe at the sight of “Permission Denied”? Did you re-boot and absent-mindedly forget to re-map that network drive? If you answered ‘Yes’ to any of these questions, then this article is for you. We present the net.exe command, which has been a part of DOS since Windows 95. You can – from within SAS – issue x commands to synchronize time between Client and Host, check the names of available and open “shares,” or even manage and control a remote system, mounting and unmounting its network drives. Your SAS program is no longer a passive, independent entity!

Skill Level: Beginning to Intermediate familiarity with Base/SAS, FILENAME and LIBNAME statements.

INTRODUCTION
The x command, or SAS Macro function %sysexec(), is a very powerful statement. It allows pass-through execution down to the operating system level. When you work exclusively on a single PC the x command is self explanatory. However, in today’s modern world communication is key and everything – and everyone – is networked. We discuss two of the most common Windows’ networking topologies, describe what is a “UNC” path reference, and outline the simple steps necessary for file management anywhere within your network. Oh, one more thing: Contrary to popular belief, MS-DOS path names really are case sensitive.

NETWORK TOPOLOGY
There are two common networking configurations. The DOMAIN configuration is used in almost every corporate intranet, while small laboratories and home users most often set up the WORKGROUP style for connecting PC’s together. Both types have their own advantages and disadvantages, described below.

STEPS TO AUTHENTICATED REMOTE CONNECTIONS
THE DOMAIN

1 AUTH Client sends username and password credentials to Domain Controller, asking for permission to access Host.
2 REQ Domain Controller forwards request to appropriate host, awaiting its response.
3 RESP target Host responds to Domain Controller either accepting or denying Client's access request. In the case of denied requests, Domain Controller increments the 'wrong password' counter.
4 GRANT Domain Controller grants desired access to Client.
5 ... communication between Client and Host proceeds just as if Host were an integral, internal part of Client.

Information technologists prefer Domains because clients and hosts are managed centrally, easier to administrate and provide synonymous, transparent access to all hosts. In other words, one user ID fits all. However, because an entire corporate intranet is centrally controlled, the Domain configuration has zero tolerance for defects and maintenance issues: If the Domain Controller goes down, every single client and host becomes unreachable. To alleviate this worry you’ll often see two or more identically configured hosts playing the role of Domain Controller. Backup Domain Controller (BDC) takes over should the Primary (PDC) happen to fail.
THE WORKGROUP

1 AUTH Client sends username and password credentials directly to Host, asking for permission to access it.
2 GRANT Host grants desired access to Client.
3 ... communication between Client and Host proceeds just as if Host were an integral, internal part of Client.

In the Workgroup configuration hosts are managed individually, and more cumbersome to administrate users. To add or remove a user, you must “log in” to each and every host and individually change user policy settings. For the application programmer, however, it is much easier to develop within the Workgroup, and in general a Workgroup configuration is less prone to system-wide failure. One unreachable host doesn't affect any other host.

UNC PATH REFERENCES

The UNC or Universal Naming Convention definition for a path in a filesystem has three pieces:
\HOST name \SHARE (volume) name \file-system PATH

Each piece is separated with backslashes, and the whole address is preceded by a double backslash.
Think about a SHARE as a logical or virtual path; it is a convenient renaming of long directory names, and also serves to “hide” higher level folders from view (see appendix for some administration tips). Unlike drive letters (C:, D:, etc), UNC paths are LINUX-compliant, too!
Even local drives have UNC equivalents. Windows defines C$, D$, etc. as special administrative or hidden shares, invisible to the casual net view command query. The dollar sign suffix indicates that a share is hidden. These shares are always mounted and available at the local level to \localhost.

A WORD ABOUT SCOPE

All network resources in the Windows' operating system are scope-global to all users. Suppose John and Mary are both logged into the Host. John issues a map drive command in his server program, so that G: maps to some other host's c:\user\john\sas folder. Almost simultaneously, Mary's server program maps the same drive letter G: to c:\user\mary\sas\dev. However, Mary is smart and, prior to her mapping command, she tests if the desired drive letter is already mapped; if so, her code un-maps any previous connection. What happens? John’s code obviously either (a) fails because it can't find required input files, or (b) writes output where it doesn't belong, or (c) both. Even worse, John’s code may overwrite files of Mary’s which have the same name.

This is a major and unresolvable artifact of Windows. Suffice it to say, the only solution is to use UNC path references and avoid absolute drive letter mappings altogether. It is also good practice for John and Mary to schedule their jobs far enough apart in time so they don't overlap, or better yet, if they both reference the same Host and share, for John and Mary to “team up” and commonalize their remote Host file handling task portions.

CREDENTIAL CONFLICT

Jeff and Melissa both use UNC path references in their Host server programs to access remote files. They already know about the dangers of absolute drive letter mapping which, by the way, is not portable among UNIX, MVS or other mainframe operating systems.
Melissa's user ID was given read/write access by the I.T. Staff while Jeff is only a conventional user with read-only permission. Jeff’s program runs first. He authenticates a share and path to the company web server, to pick up some data. (the net.exe command is described in a later section)

Jeff (Read-only access) Melissa (Read/Write access)
1. net use \web\a\b /user:jeff j
2. copy \web\a\b\c.gif c:\tmp
3. net use \web\a\b /user:melissa m
   --ERROR!--
Melissa's code starts, and at step 3 she happens to authenticate the same web server share and path, and her code fails with the following error message:

```
This set of credentials conflicts with an existing set of credentials
```

Why? Again, all network resources in the Window's operating system are scope-global to all users. Jeff's credentials to `\web\...\...` and those of Melissa don't have equivalent file permission policies. There are two ways to avoid this scenario. First, carefully schedule all jobs touching `\web` to happen synchronously, and non-overlapping in time. This is the 'best practice' we recommend (see below).

Alternately, one can authenticate to the dotted-decimal UNC path while another uses the name-resolved version, because Host address conventions are stored separately within Windows' SMB (Server Message Block) – the agent which manages authentication, credentials and policies.

### Jeff

```plaintext
1. net use \web\a\b /user:jeff j
2. copy \web\a\b\c.gif c:\tmp
3. net use \1.2.3.4\a\b /user:melissa m
4. copy c:\charts\c.gif \1.2.3.4\a\b
   --OK--
5. net use \web\a\b /d
6. net use \1.2.3.4\a\b /d
```

You can learn the dotted-decimal address of any host simply by "ping"ing it:

```
ping web
```

shows ->

```
1.2.3.4
```

Tip: In order to verify network integrity, it is a good idea to do a name server look-up of any host you reference. Nslookup the host name to determine its dotted-decimal address, then nslookup that address to confirm that the host name is the same. If you get conflicting or different information, there may be an out-of-date Domain Name System (DNS) server, or the dotted-decimal address might be aliased on purpose.

### THE CASE OF THE WRONG CASE

Jeff successfully authenticates a host and mounts one of its network resources. He later copies a file to the host using its UNC path equivalent. To his confusion, Jeff receives the dreaded `Permission Denied` message from the host. Why?

Elementary, my dear Watson. Carefully observe Jeff's `%sysexec()` or "X" commands:

```plaintext
x "net use \biometric\patprof\peopledata /user:jeff j";
x "copy c:\123-45-6789.doc \biometric\PatProf\PeopleData";
```

Look at the share name. PatProf and patprof are different. So are PeopleData and peopledata. The Windows' SMB stores authenticated UNC references in a case-sensitive manner. It remains an unsolved mystery why references mounted to a drive letter, such as files at the local drive level (i.e., C:), are transparent to case. The solution? Use UNC paths and be careful with path name case!

```plaintext
x "net use G: \biometric\patprof /user:jeff j";
x "copy c:\123-45-6789.doc g:\peopledata";
x "copy c:\123-45-6789.doc g:\PeopleData";
```

### OBJECT-ORIENTED NETWORKING

Separation and encapsulation are the principles of good O-O design. You may wish to separate the tasks of `Authentication` and resource `Mount`. Special share IPC$ allows you to do just this. You can't mount it to a drive letter, nor can you call up its directory since it's not a real resource. The Inter-Process Communication (IPC$) share is simply a placeholder. It lets you take care of your login credentials once, without worrying about which share or folder you need to access. We strongly recommend the following "best practice" for networking:

```plaintext
x "net use \...\ipc$ /user:mydomain\melissa m";  * Authenticate *
x "copy \...\a\x.txt \...\b";  * move x.txt from share A to share B *
x "del \...\logs\myprog.txt";  * delete myprog.txt from the logs *
x "net use \...\ipc$ /d";  * de-authenticate *
```

Tip: Since access to various shares can be set individually by the network administrator, it makes good sense to authenticate the Domain Controller, rather than the individual hosts.
**DOS COMMANDS**

When working with network authentications it is critically important that the `X` command be synchronized; it is unpredictable just how long it will take to reach a host or Domain Controller to verify user identity. We strongly recommend placing the following options immediately prior to any transaction:

```
options xsync noxwait;
```

**AUTHENTICATION**

Everyone knows how to map drives. From the Windows' Explorer you select the Tools menu, then “Map Network Drive...” UNIX people use the command `mount`. The drive letter is the mount point and the remote network location is the mount target:

```
G: -> \biometric\patprof
```

To map or mount a drive from within your batch program, you can’t use a visual point-and-click window. Instead, there is a powerful DOS system command called `net.exe` which does the mapping. It is quite simple to use, and can even be used with the `X` command in SAS:

```
x "net use g: \biometric\patprof";
```

To un-map, you delete the network resource using

```
x "net use g: /d";
```

Don’t worry, you’re not actually deleting the content, only the link between you and the host. As you have seen above, drive letter mount points and direct UNC mappings are distinctly different network entities. Thus, it is an error to map `G:` and then use the direct UNC equivalent `net use \biometric\patprof /d` to remove it; in this case you must remove the drive letter mapping with `net use g: /d`.

To see just what connections have been made you can use the `net use` command by itself, with no further arguments; you’ll see a nice little table of information. Once in a while it is a good idea to check network mappings and pipe the output of this command into a file, using something like

```
x "net use >> netmaps.txt";
```

to see if there are any unwanted mapped locations or other network resource “leaks.”

```
net use \host\ipc$ /user:ddd\uuu auth user U of domain D

net use \host\share\dir\Dir\DIR /user:ddd\uuu ppp auth & mount a share

net use \host\share /user:uuu ppp same, for a workgroup user U

net use \1.2.3.4\share /user:uuu ppp auth & mount a workgroup share

net use \1.2.3.4\share /d disconnect net resource
```

**USING UNC PATHS WITH LIBNAME AND FILENAME**

Once we have established ourselves with the target Host (e.g., \biometric), we can do many useful things.

```
x "net use \biometric\patprof /user:ddd\uuu ppp";
```

Your SAS program can grab data directly from a remote host’s UNC folder path:

```
libname mylib "\biometric\patprof\peopledata";
```

and create output files directly to remote hosts

```
filename myfile "\biometric\patprof\output\test.gif";
```

If the network connection is de-authenticated or disconnected, strange and unpredictable SAS behavior will result when trying to further access the `libname` or `filename` resources.

**WHAT’S THE TIME OVER THERE?**

During program validation a scheduled batch job may start earlier or later than expected if Client and Host timestamps are not synchronized. One may easily check the time of a remote host using the `net time` command, shown below. For best results, it is typical to place a syncing step in a daily task of each Host, so that each Host sync’s itself to some central common clock such as the Primary Domain Controller, NIST standard, and so forth.

```
net time \host

net time \host /set /y
```

**AVAILABLE (OPEN) SHARES**

If we aren’t sure of a particular share name on the Host, we can ask the Host to show us all of its available open shares using the `net view` command. Also displayed might be comments and descriptions about the share — very useful information indeed!

```
net view \\host
```
CONCLUSION
Together with the \texttt{x} command, we have described a method and general practice for managing remote Hosts from within any SAS program. Authenticate what you need, when you need it, and never longer than necessary.

REFERENCES
“How the Bad Password Count Is Incremented in Windows NT,” (May 13, 2003), Microsoft Knowledge Base, 219898.
“How to create and delete hidden or administrative shares on client computers,” (March 21, 2006), Microsoft Knowledge Base, 314984.


ACKNOWLEDGMENTS
John, Mary, Jeff and Melissa for use of their names.

TRADEMARK CITATION
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the authors at:

Na Li
Senior Project Analyst
Pharmaceuticals, Inc.
995 East Arques Avenue
Sunnyvale, CA 94086
Office: 408-990-7293
Email: nli@pcyc.com

Paul D Sherman
Electrical Engineer
335 Elam Village Lane, Apt. 424
San Jose, CA 95134
Home: 408-383-0471
Email: sherman@idiom.com

Web site: http://www.idiom.com/~sherman/paul/pubs
APPENDIX

AUTHENTICATION
net use
net use \\host\ipc$ /user:d\u
net use \\host\share\dir\Dir\DIR /user:d\u p
net use \\host\share /user:u p
net use \\1.2.3.4\share /user:uu pp
net use \\1.2.3.4\share /d

AVAILABLE (OPEN) SHARES
net view \\host

WHAT'S THE TIME OVER THERE?
net time \\host
net time \\host /set /y

NETBIOS TRANSACTION STATUS
nbtstat -s

PORT ACTIVITY – TCP & UDP
netstat -a

GENERAL UTILITIES
nslookup 1.2.3.4
ping host.subnet.com
tracert host.subnet.com

WHAT JOBS ARE RUNNING OVER THERE?
at \\host 4:35am /every:m,t,w,th,f,s,su “c:\work\cron\mytask.cmd”
at \\host 13:20 “atask.cmd”
at \\host 22 /d

options nosource;
/*==============================================================*/
/* Network Authentication                                        */
/* WARNING! DOS is global, session persistent and case- */
/* sensitive to share (volume) names. */
/* 1.0  2004-11-12  pds  initial cut */
/*================================================================*/
%macro netauth(mode);
%local domain uid pwd;
%let domain=dddddd;
%let uid=uuuuuu;
%let pwd=pppppp;
%if %trim(%upcase(&mode.)) eq CONNECT %then %do;
options xsync noxwait;
x "net use \\host.subnet.com\share\dir\Dir /user:&domain.\&uid. &pwd."
%end; %else %if %trim(%upcase(&mode.)) eq DISCONNECT %then %do;
options xsync noxwait;
x "net use \\host.subnet.com\share\dir\Dir /d"
%end; %else %do;
%put ERROR: (invalid mode: &mode. Expected CONNECT or DISCONNECT);
%end;
%mend;
/*================================================================*/
/* Remote Mirroring                                              */
/* 1.0  2004-11-12  pds initial cut */
/*================================================================*/
%macro mirror(args);
%let rootpath = "c:\tmp\data\sasout";
%let src = "/rrootpath.\stuff";
%let tgt = "\\host.subnet.com\share\dir\Dir\stuff"
options xsync noxwait;
%netauth(CONNECT);
*x "xcopy /s /v /c /y &src.\*.* &tgt."
%netauth(DISCONNECT);
%mend;
/*================================================================*/
/* 1.0  2004-11-12  pds initial cut */
/*================================================================*/
%options nosource;
/*==============================================================*/
/* Remote Mirroring                                              */
/* 1.0  2004-11-12  pds initial cut */
/*================================================================*/