



Creating User Accounts in Windows 2000 with Perl

Background:

We will use code that is for the most part written by David N. Blank-Edelman in his excellent book, *Perl for System Administration*, on pages 84 and 85. I have modified it to make it more general, in that we can use it to easily create accounts from the data provided by the VTC GRS (Grade Reporting System).

There are two additional Perl modules required for this, however. We need to download and install them. I have downloaded one, `Win32::Lanman`. You will install the other, `Win32::Perms` yourself using the PPM (Perl Package Manager) program. You can read about PPM from the ActiveState ActivePerl documentation.

I downloaded `Win32::Lanman` from <http://cpan.org/modules/by-module/Win32/JHELBERG/lanman.1.0.10.0.zip>.

Procedure:

Installing the two Perl modules

Setting the PPM proxy (ONLY ON THE CAMPUS, not at home): Set the environment variable `HTTP_proxy` to the value `http://proxy.vtc.edu.hk:8080`, or what value actually works. To do this:

1. right click on "My Computer" (No, yours, not mine).
2. Click on Properties
3. Click on Advanced (yes, we are advanced in so many ways!)
4. Click on Environment Variables
5. Click on the **New...** button under System variables
6. in Variable Name, enter `HTTP_proxy`
7. in Variable Value. enter the proxy that works for you:
 - `http://hqproxy.vtc.edu.hk:8080`
8. Click on all appropriate **OK** buttons.

Installing Win32::Lanman

1. Download <http://nicku.org/ftp/perl/lanman.1.0.10.0.zip> and unzip it into a subdirectory of `D:\temp`, say, `D:\temp\lanman`.
2. Open a command prompt and change to `D:\temp\lanman` (I assume that you have installed ActiveState ActivePerl from <http://nicku.org/ftp/perl/ActivePerl-5.8.3.809-MSWin32-x86.msi>).

3. Type (at the `C:\temp\lanman>` prompt):

```
D:\temp\lanman>ppm
PPM - Programmer's Package Manager version 3.1.
Copyright (c) 2001 ActiveState SRL. All Rights Reserved.
```

```
Entering interactive shell. Using Term::ReadLine::Stub as readline
library.
```

```
Type 'help' to get started.
```

4. Next, add the current directory “.” as a PPM repository, called here, “CURRENTDIR”:

```
ppm> rep add CURRENTDIR .
```

Note; you can type “help rep” at the ppm> prompt.

5. See if PPM can find lanman:

```
ppm> search lanman
Searching in Active Repositories
  1. Win32-Lanman [1.0.10] Contains the most important calls from the
MS-Lanma~
```

6. Good, so let's install it:

```
ppm> install Win32-Lanman
=====
Install 'Win32-Lanman' version 1.0.10 in ActivePerl 5.8.3.809.
=====
Downloaded 229487 bytes.
Extracting 21/21: blib\html\site\lib\win32\lanman.html
Installing D:\Perl\site\lib\auto\win32\lanman\lanman.dll
Installing D:\Perl\site\lib\auto\win32\lanman\lanman.exp
Installing D:\Perl\site\lib\auto\win32\lanman\lanman.lib
Installing D:\Perl\html\site\lib\win32\lanman.html
Files found in blib\arch: installing files in blib\lib into architecture
depende
nt library tree
Installing D:\Perl\site\lib\win32\lanman.pm
Successfully installed Win32-Lanman version 1.0.10 in ActivePerl
5.8.3.809.
ppm>
```

7. Note that the documentation for the `Win32::Lanman` module is now installed with the documentation for ActiveState ActivePerl.

Installing Win32::Perms Here we are going to use the ppm program to download and install the package from Internet. There is one additional complication here: we need to install the package from David Roth's web site instead of from the default, ActiveState's web site. The first step is to add David Roth's repository to ppm; the second is to download and install Win32::Perms.

1. At the ppm> prompt in CMD.EXE, type at the ppm> prompt:

```
ppm> rep add ROTH http://www.roth.net/perl/packages/
Repositories:
[1] ActiveState PPM2 Repository
[2] ActiveState Package Repository
[3] ROTH

ppm> search perms
Searching in Active Repositories
  1. Win32-Perms [0.2002.~ The Win32::Perms extension for Win32 X86.
Manages ~
ppm> install Win32-Perms
=====
Install 'Win32-Perms' version 0.2002.06.05 in ActivePerl 5.8.3.809.
=====
Downloaded 97493 bytes.
Extracting 12/12: blib/lib/Win32/PERMS.PM
Installing D:\Perl\site\lib\auto\Win32\Perms\Perms.DLL
Files found in blib\arch: installing files in blib\lib into architecture
dependent library tree
Installing D:\Perl\site\lib\Win32\Perms.PM
Installing D:\Perl\site\lib\Win32\Perms\COPY.PL
Installing D:\Perl\site\lib\Win32\Perms\noinher.pl
Installing D:\Perl\site\lib\Win32\Perms\RECURSE.PL
Installing D:\Perl\site\lib\Win32\Perms\ShowPerm.pl
Installing D:\Perl\site\lib\Win32\Perms\TEST.PL
Installing D:\Perl\site\lib\Win32\Perms\test2.pl
Installing D:\Perl\site\lib\Win32\Perms\Win32-Perms.ppd
Installing D:\Perl\site\lib\Win32\Perms\History.txt
Installing D:\Perl\site\lib\Win32\Perms\README
Successfully installed Win32-Perms version 0.2002.06.05 in ActivePerl
5.8.3.809.

ppm> quit
```

2. Note that the documentation for the Win32::Perms module is the file D:\Perl\site\lib\Win32\Perms\README

The code to create user accounts

This code (from David N. Blank-Edleman) implements adds a user:

```
use strict;
#*
```

```

** basic local user account creation routine for NT/2000
**
use Win32::Lanman; # for account creation
use Win32::Perms; # to set the permissions on the home directory

# our $homeNTdirs = "\\servername\home"; # home directory root dir
our $homeNTdirs = "D:\home"; # home directory root dir

# returns false if successful, true and an error message if there is a
# problem. Rather un-Perl like, more like C.
sub CreateNTAccount
{
    my ( $account, $password, $fullname, $group ) = @_;

    # create this account on the local machine
    # (i.e. empty first parameter)
    my $result
        = Win32::Lanman::NetUserAdd(
            "",
            {
                name           => $account,
                password       => $password,
                home_dir       => "$homeNTdirs\\$account",
                full_name      => $fullname,
                # expires in 60 days from now:
                acct_expires  => 3600 * 24 * 60 + time()
            }
        );
    return Win32::Lanman::GetLastError() unless $result;

    my @info;
    # add to appropriate LOCAL group (first get the SID of the account)
    die "SID lookup error: ".Win32::Lanman::GetLastError()."\n" unless
        Win32::Lanman::LsaLookupNames( "", [ $account ], \@info );
    $result = Win32::Lanman::NetLocalGroupAddMember( "", $group,
        @{$info[0]}{sid} );
    return Win32::Lanman::GetLastError() unless $result;

    # create home directory
    mkdir $homeNTdirs, 0777 unless -d $homeNTdirs;
    mkdir "$homeNTdirs\\$account", 0777 or return "Unable to make homedir:$!";

    # now set the ACL and owner of the directory
    my $acl = new Win32::Perms( "$homeNTdirs\\$account" );
    $acl->Owner( $account );

    # we give the user full control of the directory and all of the
    # files that will be created within it (hence the two separate calls)
    $acl->Allow( $account, FULL, DIRECTORY|CONTAINER_INHERIT_ACE );
    $acl->Allow( $account, FULL,

```

```

                                FILE|OBJECT_INHERIT_ACE|INHERIT_ONLY_ACE);
$result = $acl->Set();
$acl->Close();

return $result ? "" : $result;
}

```

...and this code (also from David) implements delete a user:

```

**
** basic account deletion routine for NT/2000
**

use Win32::Lanman;    # for account deletion
use File::Path;      # for recursive directory deletion

sub DeleteNTAccount
{
    my $account = shift;
    my @info;
    # remove user from LOCAL groups only. If we wanted to also
    # remove from global groups we could remove the word "Local" from
    # the two Win32::Lanman::NetUser* calls *e.g. NetUserGetGroups)
    die "SID lookup error: ".Win32::Lanman::GetLastError()."\n" unless
        ( Win32::Lanman::LsaLookupNames( "", [ $account ], \@info ) );
    my @groups;
    Win32::Lanman::NetUserGetLocalGroups( "", $account, '', \@groups );
    foreach my $group ( @groups )
    {
        print "Removing user from local group ".$group->{name}."...";
        print( Win32::Lanman::NetLocalGroupDelMember( "",
                                                    $group->{name},
                                                    ${$info[0]}{sid} )
              ? "succeeded\n" : "FAILED\n"
              );
    }

    # delete this account on the local machine
    # (i.e. empty first parameter)
    my $result = Win32::Lanman::NetUserDel( "", $account );

    return Win32::Lanman::GetLastError() if $result;

    # delete the home directory and its contents
    $result = rmtree( "$homeNTdirs\\$account", 0, 1 );

    # rmtree returns the number of items deleted,
    # so if we deleted more than 0, it is likely that we succeeded
    return $result;
}

```

And here is a bit of code to test it with:

```
my $result
    = CreateNTAccount( "test", "testpasswd", "The Test User", "Users" );
unless ( $result )
{
    print "good, it worked\n";
}
else
{
    print "Cannot create test account: $result";
}
```

Okay, these subroutines use some features of the Perl programming language that we didn't cover in lectures. But we never aimed to cover every feature of Perl in just five weeks. For example, `_${info[0]}{sid}` is a hash element in an array element that we refer to by a reference. And no, I won't explain that here; you can find that out for yourself, or ask me out of class hours.

Now Create Those Thousands of Accounts on the Windows server

Use these two subroutines and the file <http://nicku.org/snm/lab/regular-expressions/artificial-student-data-2003.txt> to generate accounts for all those thousands of students.