

Automate Creation of Active Directory User Accounts

Larry Keyes Microdesign Consulting Inc.

I am currently configuring user accounts for a Windows 2003 server, and I want to automate adding accounts. Checking the [Windows Command-Line Administrator's Pocket Consultant](#) I find the following command syntax:

```
dsadd user UserDN -samid SAMName
[-upn UPN] [-fn FirstName] [-mi Initial]
[-ln LastName] [-display DisplayName]
[-empid EmployeeID] [-pwd {Password|*}]
[-desc Description] [-memberof Group;...]
[-office Office] [-tel PhoneNumber]
[-email Email] [-hometel HomePhoneNumber]
[-pager PagerNumber] [-mobile CellPhoneNumber]
[-fax FaxNumber] [-iptel IPPhoneNumber] [-webpg WebPage]
[-title Title] [-dept Department] [-company Company]
[-mgr Manager] [-hmdir HomeDirectory]
[-hmdrv DriveLetter:] [-profile ProfilePath]
[-loscr ScriptPath] [-mustchpwd {yes | no}]
[-canchpwd {yes | no}] [-reversiblepwd {yes | no}]
[-pwdneverexpires {yes | no}] [-acctexpires NumberOfDays]
[-disabled {yes | no}] [{-s Server | -d Domain}]
[-u UserName] [-p {Password | *}] [-q]
```

Even trying this to add a single account is pretty discouraging. The syntax is described on the [Microsoft XP web site](#).

Then I remember that I'm a database programmer, and that Active Directory is really only a database. And you would think that you could put all the users into a database file, and write a "report" which is really a .CMD file to add all of the users. In other words, what I want to end up with is a command file that looks like this:

```
ECHO Creating User: Stefan Andres
md \\mdwin2003\users\sandres
dsadd user "CN=Stefan Andres,CN=Users, DC=microdesign, &
DC=mxdesign, DC=net" -samid "sandres" -display "Stefan Andres" & -pwd
ztx98gwp -fn "Stefan" -ln "Andres" -canchpwd no -pwdneverexpires yes &
-desc "Teaching Staff" -office Columbia -hmdir \\mdwin2003\users\sandres &
-memberof "CN=CVABE, CN=users, DC=microdesign, DC=mxdesign, DC=net"
ECHO.
ECHO Creating User: Kendra Rome
md \\mdwin2003\users\krome
dsadd user "CN=Kendra Rome,CN=Users, DC=microdesign, &
DC=mxdesign, DC=net" -samid "krome" -display "Kendra Rome" -pwd xur24ebs &
-fn "Kendra" -ln "Rome"& -canchpwd no -pwdneverexpires yes & -desc "Teaching
Staff" -office Columbia -hmdir \\mdwin2003\users\krome &
-memberof "CN=CVABE, CN=users, DC=microdesign, DC=mxdesign, DC=net"

PAUSE
```

The batch file does two things. The MD command makes a home directory for the user in a shared folder called `\users`. No mystery there. The `dsadd` command is the all-singing all-dancing command to add the user's name and password, and any other information that you want to add to the Active Directory. These are domain accounts intended for staff using workstations attached to the server. Once the accounts are added through the CMD file, they appear in the Active Directory Users and Computers management console.

Walking through this version of `dsadd`, it shows that:

1. Kendra's login is *krome* Her display name, is *Kendra Rome*
2. Her password is added, with three lower-case letters, two digits and three more lower case letters. Note that this does *not* meet the default complexity password requirements set in Windows 2003. The default can be altered in the group policy snap-in.
3. The password cannot be changed and it does not expire.
4. I'm currently using the AD *description* field to hold the user's title.
5. The AD *office* field is set to the name of her office.
6. I set her home directory based on her name, to match the directory created in the first command.
7. Finally, I add her to a group *CVABE* that I've previously created. Users are also automatically added to the built-in *domain users* group.

Defining the database table

The database is a single table, which could be created in pretty much any program. I ended up starting the table in Excel, and then exported the excel spreadsheet to Visual FoxPro. However, as long as you can eventually use the table to write text output, you can use any program to hold the data. Even a comma delimited text file will do.

Field Name	Type	Width	Comment
Common	Character	25	Common Name, i.e. "Mary Smith"
Login	Character	16	Login Name, i.e. msmith
Password	Character	12	User's password, i.e., abc99xyz
First	Character	12	User's first name
Last	Character	12	User's last name
Canchpwd	Character	3	Yes or No (written out, not a boolean field)
Passexpire	Character	3	Yes or No (written out, not a boolean field)
Descrip	Character	25	User "description"
Office	Character	25	User's location
Homedir	Character	16	Same as the login name

In the spreadsheet or a FoxPro Browse, the data itself will look something like this:

Common	Login	Password	First	Last	Canchgpwd	Passexpire	Descrip	Office	Homedir
Stefan Andres	sandres	gwp98zts	Stefan	Andres	no	no	Teaching Staff	Montpelier	sandres
Kendra Rome	krome	xur22abx	Kendra	Rome	no	no	Teaching Staff	Montpelier	krome

The database field names and length are arbitrary. They match one-to-one a subset of the fields in the Active directory dialog boxes. If I had wanted to fill more of the AD fields, I would create corresponding fields in my user table.

To create the command file that will add these users, I used the Visual FoxPro TextMerge function.

```
* MakeUserAdd.prg
* Creates a batch command file for adding users to MS Active Directory
* Requires an existing database file called 'users' which holds the AD data
and parameters
* The program creates a user home directory based on the login name. Assumes
that there is an existing share called "Users"
* LK - Microdesign Consulting Inc. April 11, 2005
* lkeyes@mxdesign.net
```

```
SELECT USERS
* Suppress messages
SET SAFETY OFF
CLEAR
SET TEXTMERGE TO users.cmd
SET TEXTMERGE ON
SCAN
\ECHO.
\ECHO Creating User: <<ALLTRIM(First)+' '+ALLTRIM>Last>>
\md \mdwin2003\users\<<homedir>>
\
\dsadd user "CN=<<ALLTRIM(Common)>>,CN=Users, DC=microdesign, DC=mxdesign,
DC=net"
\\ -samid "<<ALLTRIM(Login)>>"
\\ -display "<<ALLTRIM(First)+' '+ALLTRIM>Last>>"
\\ -pwd <<ALLTRIM>Password>>
\\ -fn "<<ALLTRIM(First)>>"
\\ -ln "<<ALLTRIM>Last>>"
\\ -canchpwd no
\\ -pwdneverexpires yes
\\ -desc "<<ALLTRIM(descrip)>>"
\\ -office <<ALLTRIM(office)>>
\\ -hmdir \mdwin2003\users\<<ALLTRIM(homedir)>>
\\ -memberof "CN=CVABE, CN=users, DC=microdesign, DC=mxdesign, DC=net"
ENDSCAN
\PAUSE
SET TEXTMERGE OFF
SET TEXTMERGE TO
```

The textmerge program creates an ASCII file called *user.cmd*. The user.cmd file is the file that is actually run at the command line which creates the users.

Once the `user.cmd` is created, it is run on the server's command line.

Enhancements

One thing I noticed was when I ran the `user.cmd` program, the first time, I received several errors. So I made a hard-coded "removal" program which removed all of my new users. This consists of a single line for each user, using the `dsrm` command:

```
dsrm "CN=Stefan Nichols,CN=Users, DC=microdesign, DC=mxdesign, DC=net" &  
-noprompt
```

The `dsrm` command is used to remove any directory object. As long as it has the fully qualified name of the object, then it can find it within in the directory. The `-noprompt` suppresses a message asking "do you really want to delete this object?"

This `delusers.cmd` program would be a candidate for another textmerge program.

Another possible enhancement might be a front-end for the user database which makes data entry easier and could enforce naming conventions.

Alternatives

The database table can be built in anything that can output ASCII text. The TextMerge program could also be written in Perl, or even SQL. Eventually, with the ability to query the Active Directory with `dsquery` you have the beginnings of a complete ad user management system. Finally, you could eliminate the whole command file interface by creating a similar program that wrote to the Active Directory objects in Visual Basic, C# or Access.

Larry Keyes is a principal at Microdesign Consulting Inc. He specializes in strategic information systems planning and development for non-profit, government and healthcare. www.mxdesign.net and www.techfornonprofits.com