System Administration Made Easy(er)

User and Group Manipulation

As we all know the IT industry is plagued with a notoriously high turnover rate. According to BNA, Inc. out of Washington, 1-2% of the workforce, per month, left their jobs in 1999. The number of new employees, contractors, interns, moving in, out, and around today's' organizations is enough to give any administrator cause to purchase stock in the companies producing Excedrin, Tylenol, or any of the other pain reducing drugs. MKS actually ran an ad a while back whereby the MKS Toolkit product was depicted on the side of an aspirin bottle, a very successful campaign if I remember correctly. And I haven't even mentioned the migraines that are brought on through mergers and acquisitions.

Solving this problem, that's for Human Resources. Dealing with the rapid turnover, a System Administrators task. Finding more efficient and automated ways to manage these user and group accounts would save enough time to allow System Administrators to concentrate on the really tough issues facing IT.

This article will demonstrate ways in which you can automate the task of user administration using the MKS Toolkit system administration products. In the Additional Resources section below you will find links to the man pages for the primary commands used as well as to additional scripts that are available from the MKS Toolkit Resource Kit.

Let us start by adding a new user to the system the Windows way, by opening (at least on Windows 2000) the Users and Passwords applet from the Control Panel. Follow through the Add New User Wizard by adding the username and domain after clicking the Add button. Next you select the level of access or the group you would like to put this user in and click the Finish button to complete the operation.

Using the command line tools for adding a single user is just as simple. There are two commands that come in to play, userinfo and member. They both accept a number of options, but to duplicate the actions above you would simply type the following*:

```
[C:/] userinfo -S //YourServer -a jdoe
[C:/] member -S //YourServer -u jdoe -a Accounting
```

* All examples assume that you are running in an MKS KornShell environment.

Adding one user or group to a system is easy and whether you point and click your way through it or type it on the command line the task is going to take about the same amount of time. The power of a scriptable solution is shown when you have to perform many of these simple operations over and over.

Given a list of users you can easily modify the above example to automatically add two, four, ten, fifty user accounts to your server. Here is the source we will use for this example:

```
[C:/] cat names.txt
Doug Akers
Ed Matthews
Alan Brown
Caterina Mclean
Brooke Lanier
Heather Jacobs
Wendy Mades
```

To simplify further, and because of the powerful text processing available in the KornShell environment, the usernames for both login and home directory can be generated from the full name of each individual.

```
[C:/] cat adduser.ksh
#!/mks/mksnt/ksh

# This line strips the space character from the
# default field separator list.
IFS=${IFS/ /}

# Process every name in the file passed to the
# script as an argument on the command line.
for i in `cat $1`
do
  # Lowercase the variable used for usernames.
  typeset -l uname

  # Create the username from the full name of the
  # given user.
  uname="$(echo $i | cut -b1)${i/(*) (*)/\2}"

  # Add the user to the system with various
  # extended options.
  userinfo -S //YourServer -a \
       -f FullName:"$i" \
       -f HomeDir:Q:/Homes/$uname \
       -f Password:$uname \
       -f PasswoerExpired:Yes \
       $uname

  # Print the username to stdout
  echo $uname
done
```

As you can see from the above listing, you can set additional user defaults with the userinfo command. FullName, Password, Comment, HomeDirDrive, HomeDir, Profile, LogonScript, AccountDisable, Lockout, and PasswordExpired are just a few of the parameters which can be set during a userinfo operation. The output from the above command would look like this:

```
[C:/] adduser names.txt
dakers
ematthews
abrown
cmclean
blanier
hjacobs
wmades
```

But what about adding those users to a group? Did that on purpose actually, I wanted to show you how you can link various commands together to perform slightly more complex operations. First, let's create a test group that we can use instead of one of our live groups.

```
[C:/] groupinfo -S //YourServer -a TESTING
```

Then we could then run the following in place of simply running the adduser script:

```
[C:/] member -S //YourServer -L -g TESTING -a \
> `adduser names.txt`
```

Since the adduser script prints out the usernames that it adds to the system we can take advantage of that with the member command. By placing the invocation of adduser inside the `` (back tick) characters that script will be run before the member command is processed. In essence, what the member command will see is the following:

```
[C:/] member -S //YourServer -L -g TESTING -a dakers \
> ematthews abrown cmclean blanier hjacobs wmades
```

In a multi-server environment synchronizing the users and groups can be a tedious challenge that can also be automated with scripting. Take the following example script which takes all the members of a group on one machine and adds them to an identical group on another.

```
[C:/] syncgroup TESTING YourServer TargetServer
[C:/] cat syncgroup.ksh
#!/mks/mksnt/ksh

# usage: syncgroup group source-server target-server

# This line strips the space character from the
# default field separator list.
IFS=${IFS/ /}

# Create the group on the target server.
groupinfo -S //$3 -a $1

# Query the source server for usernames and add
# them to the new group on the target server.
for user in $(groupinfo -S //$2 -m $1)
do
  member -S //$3 -g $1 -a $user
done
```

A similar script can be created to synchronize users across machines. When run against the source server the output (either full output or brief output if you use the -b option) can easily be parsed with awk or Perl and reformatted into correct input lines for userinfo to add to the target server.

Each of the commands used here can also be used to delete or modify existing users and groups. Using the -d (delete) or -u (update) options to userinfo or groupinfo you can easily clean up and manage your entire user environment. The following script will clean up what the previous examples have created.

```
[C:/] cat clean.ksh
#!/mks/mksnt/ksh

# This line strips the space character from the
# default field separator list.
IFS=${IFS/ /}

# Remove the user accounts found in the specified
# group on YourServer.
for user in $(groupinfo -S //YourServer -m TESTING)
do
  userinfo -S //YourServer -d $user
done

# Remove the group YourServer.
groupinfo -S //YourServer -d TESTING

# Remove the group from the TargetServer.
groupinfo -S //TargetServer -d TESTING
```

These same tools that allow you to add, delete, and manipulate users and groups can also allow you to create custom reports. And in order to effectively manage the users and groups on your system you must first understand who those users are. Take the usrrpt.ksh script found in the Scripts section of the MKS Toolkit Resource Kit. This script allows you to create custom records for each user in your system, reporting only the data that you want or need to see, in the order that you need to see it. Because it is well formatted data, you can then pipe this output to other scripts which perform some magic (automatic email, setting permissions, clean out home directories, etc.) or even import it into Excel for those fancy management summaries :)

Ok, so there's not a whole lot to manipulating users and groups within a system. And, granted, it's not the most glamorous or sexy thing you can do with the command line scripting and the tools and utilities that are available to you. But even the mundane can be automated and all but swept into obscurity allowing you to concentrate on those really cool System Administration tasks.

About MKS Toolkit

There are several products in the MKS Toolkit product line aimed at different kinds of people, performing different kinds of tasks. All MKS Toolkit products are unified by our goal of making your use of Windows more efficient and more enjoyable. The products fall into two broad categories—those for system administrators and those for software developers.

There are three main components included in the MKS Toolkit product line, and each of the individual products may contain some or all of each group. This document primarily refers to the commands and utilities component; a subset of which is included in all MKS Toolkit products.

MKS Toolkit Commands and Utilities

There are over 400 individual commands and utilities that are offered as part of the MKS Toolkit product line. These range from command shells like the KornShell, C Shell, and **bash** to invaluable UNIX commands like **grep**, **make**, **awk**, **sed**, **vi**, and **find**, to system connectivity utilities and services like Secure Shell, **rsh**, and **telnetd**. MKS Toolkit products also include many Win32 specific commands to deal with Windows specific constructs and issues. This includes commands like **registry**, **shortcut**, **winctrl**, **desktop**, and **domain**.

MKS Toolkit UNIX APIs

Including more than 2700 APIs, this component of the MKS Toolkit product line allows you to port all manner of UNIX and Linux applications to Windows, while maintaining a single source code baseline.

MKS N*u*TCRACKER Platform

The N*u*TCRACKER Platform is a full UNIX runtime compatibility environment for windows platforms, that provides services to each of the MKS Toolkit products and to any application or script migrated from UNIX or Linux to Windows with either MKS Toolkit for Professional Developers or MKS Toolkit for Enterprise Developers.

Additional Reference

For more information on the individual MKS Toolkit products, please visit our web site at http://www.mkssoftware.com.

- MKS Toolkit Products: http://www.mkssoftware.com/products
- MKS Toolkit Reference Pages: http://www.mkssoftware.com/docs (man pages)
- MKS Toolkit Command List: http://www.mkssoftware.com/products/tk/commands.asp