

# System Administration Made Easy(er)

## User and Group Manipulation

By Doug Akers, Product Manager, MKS Toolkit Product Family

As we all know the IT industry is plagued with a notoriously high turnover rate. 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 is 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 MKS 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 ([www.mkssoftware.com/reskit](http://www.mkssoftware.com/reskit)). 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.

### Additional Resources:

- [userinfo](#) — man page
- [groupinfo](#) — man page
- [member](#) — man page
- The following scripts can be found in the \$ROOTDIR/samples/sh/admin directory after installing the MKS Toolkit product:
  - addgrp, adduser, gaddgrp, gadduser, grmgrp, grmuser, and usrrpt

## Other Scriptable MKS Toolkit Solutions

Picture a world where at the touch of a button you could instantly set up user accounts, group associations, system permissions, end-user desktops, manage services, network mounts, the registry, and a myriad of other System Administration tasks.

The following paragraphs and links will expose a number of scriptable solutions that will help you manage your systems more effectively. They'll talk about how you can use the MKS Toolkit products, along with native Windows tools, to perform the tasks I listed above and a few dozen more.

### Monitoring Your System

Do you have a web site that must be available 24x7. Do you want to be notified when a server's disk utilization exceeds a certain threshold? Then MKS Toolkit scripting can solve your problems. A script which runs on a timer using either `tasksched` or Windows Task Scheduler can monitor vital functions and send e-mail or page you when there is trouble. In fact, that's almost exactly what MKS AlertCentre does. MKS AlertCentre is a comprehensive system and application monitoring solution that is built with the scripting components of MKS Toolkit for System Administrators. Find out more about MKS AlertCentre from our web site at [www.alertcentre.com](http://www.alertcentre.com).

### System Backup

Are you struggling with Windows backup and the problems associated with single tape drive access, drive letter mappings, and scripting a single instance GUI tool? The MKS Toolkit `tar` utility run from a scheduler such as `TKSched` or Windows Task Scheduler is the answer to your problems. MKS Toolkit for System Administrators will even backup and restore Windows security information. If you have a requirement to share tapes with UNIX backup/restore technology then MKS Toolkit `tar` is the answer to your problems. We all know `tar`—we've used it so many times that our fingers type "`tar cvf`" and "`tar xvf`" whenever we want to move entire directory trees from one machine to another. The *MKS Toolkit Backup and Tape Handling Solutions Guide* ([www.mksoftware.com/docs/sg/#bsg](http://www.mksoftware.com/docs/sg/#bsg)) will give you a complete overview of the tape and backup functionality within MKS Toolkit products.

### Automating Installations

Deploying applications has to be one of the most time consuming processes that falls within the System Administration realm. Any process or tool that can help push updates and new applications to your users desktops can only be "a good thing™."

Did you know you could automate the installation of the MKS Toolkit products themselves? Check out the Installation FAQ on the MKS Technical Support web site ([www.mkssoftware.com/support](http://www.mkssoftware.com/support)).

## Password Synchronization

Many of you have both UNIX and Windows machines to administer. Your users must access both kinds of machines, but want only one password and do not want to have to change their passwords two or more times. How important is Password Synchronization in a mixed environment and how can you ensure that passwords changed on either UNIX or Windows are propagated back to the other systems?

## Simple Network Management Protocol (SNMP)

You have a managed network and have added a number of Windows boxes and you need them all to play well with your Network Management software. Windows comes with built in SNMP protocol support and provide some counters to permit the workstation (or server) to be considered a managed node. MKS Toolkit contains a full suite of SNMP command line utilities to help you better manage your Windows machine in this environment.

## Windows Management Instrumentation (WMI)

WMI <http://msdn.microsoft.com/library/backgrnd/html/wmicim.htm> as with any COM interface can be accessed using MKS Perl. Perl is particularly well suited as a scripting language for COM. The *MKS Toolkit Evaluation Guide* has a sample of calling WMI to retrieve network adapter settings on Windows.

## Setting and Implementing Corporate Standards

You have hundreds of existing users and you want to add an icon to all of their desktops; or you want to map a drive for every user logging into all machines on the network. The built in Windows GUI tools will allow you to make these kinds of modifications, but you have to visit each machine serially in order to affect the necessary changes. How do you make bulk edits to remote registries or user profiles? The power of MKS Toolkit scripting can be used to solve large numbers of one-time or repetitive tasks.

## Controlling Services

Have you ever wanted to start a service on a remote machine from the command line or a script? `NET` does not control remote machines. Why try to remember all the `NET` command options when all you need is the `service` command within MKS Toolkit for System Administrators to fully manage all services locally or on a remote machine? In one script you can enumerate all the services, `grep` (text search) for the one you want, stop it, update the registry, and restart your services.