support bulletin volume 2, issue 3 fall 1992

making friends with niutil

by Dave Cottle

niutil—you've seen it in the *support bulletin*. Maybe you've read the man page or even tried a command or two. But how confident are you that you could use this utility effectively? niutil can be a powerful tool in an administrator's repertoire, but it can also be a dangerous one. In this article, we'll show you how to use niutil to manipulate a NetInfo database.

why use niutil?

Despite its reputation, niutil is fairly simple. Its options allow you to create, examine, and destroy NetInfo directories or properties. Many of the things you can do with niutil you can also do with NetInfoManager. Because it's easy to make mistakes on a command line, it's usually safer to use NetInfoManager when making modifications to NetInfo.

Sometimes, though, niutil is the only choice-for example, if you're logged in over a modem without access to the graphic interface.

a look at the database

The two options for examining a NetInfo database are list, for directories, and read, for properties. With each, you must indicate the domain to read and the path of the directory. Because you're only examining the NetInfo database, there's no danger here.

list

You use the list option to examine the subdirectories of a given NetInfo directory. The following command looks at the subdirectories of the /machines directory in the parent domain (..):

```
niutil -list .. /machines
```

The output of this command includes the number of each subdirectory and, by default, the value or values of the name property. In this case, each subdirectory represents a host entry stored in the parent domain.

```
35        earth_dialin
33        earth_remote
2         earth mailhost
34        pluto
37        venus
```

If you want to examine the values of a different property, simply include the name of the property in the command:

```
niutil -list .. /machines serves
```

The output lists the values of the serves property instead of the name property. Notice that directory 35 (earth_dialin) has no value; it doesn't have a serves property.

```
35
33        earth_remote/local
2         earth/local ./network
34        mars/local
37        venus/local
```

read

The read option displays the properties associated with a given NetInfo directory. The following command shows the properties stored in the host entry for earth in the parent domain:

```
niutil -read .. /machines/earth
```

The output includes each property key and the associated value(s).

Notice that this directory has a host alias (mailhost) stored as the second value of the name property.

```
name: earth mailhost
ip_address: 192.82.163.1
serves: earth/local ./network ../super
```

If you can't access a domain with a relative name (".", "..", or "/"), you can use tagged domain notation with the t option. Remember that tagged domain notation is in the form host/tag, where host is the name or Internet address of the computer serving the domain, and tag is the tag of the NetInfo database. The following command lists the properties stored in the directory /users/root (the root user account) in the domain tagged super served from the host earth:

```
niutil -read -t earth/super /users/root
```

The output of this command contains all the properties for a user account:

```
name: root
uid: 0
gid: 1
realname: Operator
```

```
home: /
shell: /bin/csh
_writers_passwd: root
passwd: Vbg0X7ZIQSEIU
```

The most familiar way to specify a NetInfo directory is with the value of the name property, and so far, the examples have all used this method. However, you can also identify a directory by the value of some other property. For example, the following command displays the properties associated with the subdirectory under /users that has a uid property with a value of 0 in the root domain:

```
niutil -read / /users/uid=0
```

Because the root account has user ID 0, the output of this command is the same as if you had indicated /users/root.

What if the value you specify is ambiguous-if more than one directory has a property with the indicated value? The following command looks in the current domain for a user account that has the default group ID 1. Group ID 1 is the group wheel, and the users root, daemon, and me have it as their default group. Any guesses as to the result?

```
niutil -read . /users/gid=1
```

The read option displays only the properties for a single directory. The first directory that matches the specified path is displayed, as determined by the directory number. In this case, root has the lowest directory number, and its directory is the first one found. Remember, the list option displays directory numbers as well as, by default, the value of the name property.

Finally, if you know the directory number, you don't need to use a path. For example, the output of niutil with the list option used earlier shows us that /machines/earth in the parent domain is directory number 2. The following commands are equivalent:

```
niutil -read .. /machines/earth
niutil -read .. 2
```

the creative spirit

With niutil, you also have options to create directories (with create) and properties (with createprop). Be very careful with createprop because it will overwrite an existing property that has the same key. When using create, be careful what you type so you create only directories you intend to create.

create

The following command creates a new directory named admins under the /aliases directory in the parent domain:

```
niutil -create .. /aliases/admins
niutil -read .. /aliases/admins
name: admins
```

As you can see from this output, you now have a new directory with a single property, name, assigned the value admins. A mail alias without any members, however, isn't very useful.

createprop

The next command creates a new property and assigns it several values:

```
niutil -createprop .. /aliases/admins members root
 celia george mary
niutil -read .. /aliases/admins
```

The output from the second command shows that the new property members has been created and assigned the values root, celia, george, and mary.

```
name: admins
members: root celia george mary
```

If you use the createprop option to set the value(s) of an existing property, that property is overwritten. For example, the following command overwrites the name property so this mail alias can be used with either lowercase or uppercase letters:

```
niutil -createprop .. /aliases/admins name admins
 ADMINS
niutil -read .. /aliases/admins
name: admins ADMINS
members: root celia george mary
```

If you want to modify a domain and you're not logged into the server as root, you need to use the p option. When you use this option, you're prompted for the root password of the domain you're modifying:

```
niutil -create -t -p earth/super /aliases/engineers
Password:
```

destructive tendencies

The final pair of options is used to destroy directories or properties: destroy for directories and destroyprop for properties. Be very careful

with both of these-you don't get a chance to change your mind.

destroy

Sometimes you may need to eliminate an existing directory. For example, let's say you make a mistake while entering a command to create a new mail alias directory:

```
niutil -create .. /aliases/admisn
```

To destroy the offending directory so you can start over, enter the following:

```
niutil -destroy .. /aliases/admisn
```

Of course, you can also fix this situation by using createprop to overwrite the name property.

destroyprop

Sometimes you need to destroy only a specific property. For example, let's say you made the following mistake when trying to create the members property in a new mail alias:

```
niutil -createprop .. /aliases/admins root celia
```

```
george mary
```

As the output below shows, you've created a new property named root instead of a property named members. Oops.

```
niutil -read .. /aliases/admins
name: admins
root: celia george mary
```

You correct the situation by destroying the mistake and starting over:

```
niutil -destroyprop .. /aliases/admins root
```

some last thoughts

As you've seen, niutil is a nifty little program. For an overview of its options, take a look at table 1. With a little forethought and caution, you'll find that niutil can be a great help, especially when you're troubleshooting. By the way, are you friends yet?

table 1: niutil options

option	**function**
list	Display the subdirectories of a given NetInfo directory.

read	Display the properties and values associated with a given NetInfo directory.
create	Create a NetInfo directory.
createprop	Create a property.
destroy	Destroy a NetInfo directory.
destroyprop	Destroy a property.
t	Identify the domain with tagged domain notation.
p	Propt for the root password of the domain.