

## Command-Line NetDB

*There is no "undo," there is no "oops," there is no "are you sure?"*

From any public Unix systems machine (corn, cardinal) the command is:

```
/usr/pubsw/sbin/netdb
```

You may want to put `/usr/pubsw/sbin/` into your path.

If you just type the command, you get the help screen:

```
corn:~> netdb
```

Usage:

```
netdb node admin --add admin,... --remove admin,... INPUT
netdb node alias --add alias,... --remove alias,... name
netdb node comment [ --set comment | --clear ] INPUT
netdb node custom --add name[=value],... --remove name[=value],... INPUT
netdb node delete [ --keep_mx ] [--force ] INPUT
netdb node department --set department INPUT
netdb node expiration [ --set date | --clear ] INPUT
netdb node info INPUT
netdb node ip_address --remove old_ip [ --add new_ip ] node
netdb node group --add group,... --remove group,... INPUT
netdb node location --set building:room INPUT
netdb node model --set make:model INPUT
netdb node name --remove old_name [ --add new_name ] node
netdb node os --add os,... --remove os,... INPUT
netdb node receive_mail_for --add mailname[:preference],...
    --remove mailname,... name
netdb node state --set state INPUT
netdb node user --add user,... --remove user,... INPUT
```

where INPUT is either a list of one or more nodes  
separated by spaces or the '--input file' option.

```
netdb node clone --template node --name name
    [ --location building:room | :room ]
    [ --hardware|hw hardware address [ --dhcp [ --roam ] ] ]
    [ --ip ip address[+] | none ]
    [ --model make:model ] [ --os os,... ]
    [ --user user,... ] [ --admin admin,... ]
    [ --comment comment ]
```

```
netdb node interface --add none --ip ip address[+] node
netdb node interface --add hardware address
    [ --dhcp [ --options option=value,... ] [ --roam ] ]
    [ --ip ip address[+] ] node
```

```
netdb node interface --modify (hardware address | IP address)
    [ --hardware|hw hardware address | none ]
    [ --dhcp[=(on|off)] ] [ --options option=value,... ]
    [ --roam[=(on|off)] ] [ --ip ip address[+] ] node
```

```
netdb node interface --remove (hardware address | IP address),... node
```

```
netdb user active_flag [ --set | --clear ] INPUT..
```

(We'll not discuss the "netdb user..." commands, since those are for creating new NetDB accounts. They can be used to look up NetDB users, which we'll show)

A more detailed help system ("man" based) is available from netdb --help:

```
corn:~> netdb --help
NETDB(1)                User Contributed Perl Documentation                NETDB(1)

NAME
    netdb - Create, Modify or Delete NetDB Records

SYNOPSIS
    netdb node admin --add admin, ... --remove admin, ...
        [ --input file | node ... ]

    netdb node alias --add alias, ... --remove alias, ... name

    netdb node comment [ --set comment | --clear ]
        [ --input file | node ... ]
```

(etc.)

## Things to keep in mind:

- *Spaces, commas, quotes:*

Command-line NetDB's biggest "gotcha" is the incorrect use of spaces and/or commas, which is a function of how the Unix shell works. A space means a change has been done to the command sequence. Anything without a modifier in front of it is telling NetDB that you're listing a node. I.e.:

```
netdb node alias --add alias1 alias2 host
```

Will attempt to add the alias "alias1" to the host "alias2" and again to the host "host" and that will result in an error, since you can't add the same alias to two hosts. The correct syntax is:

```
netdb node alias --add alias1,alias2 host
```

Which will add two aliases (alias1 and alias2) to the record called "host." In the incorrect example, putting a space after alias1 without putting any modifier (--remove, for example) told NetDB that you're done with commands and are starting to list the hosts to which you wish to apply your commands.

If your input has spaces (comment fields, departments, locations, etc.), use quotes.

If your input has commas (such as for the SU Firewall custom field), use quotes and backslash before the comma, i.e.:

```
netdb node custom --add "SU Firewall"="enable\,ssh=w\,vnc=w" host-for-world-ssh-and-vnc
```

- *"--add" and "--remove" vs. "--set" and "--clear"*

If a field's data can't be added to or subtracted from (comment, date, etc.), it'll use the "--set" or "--clear" modifier to the command. If a field's data can have more or less information without modifying the existing information (adding or subtracting aliases, users, administrators, etc.) then use the "--add" or "--remove" modifier.

- *node vs. INPUT*

You can apply most NetDB commands to a file saved in your AFS space, using --input [file] instead of just node. The file needs to have each node listed by name only (no aliases or IP numbers) each on its own line. You can also apply a command to multiple nodes just by listing them all at the end of the command separated by spaces.

If you wanted to clear the comment on the hosts host1, host2 and host3, you could do so by:

```
netdb node comment --clear host1 host2 host3
```

Or you could have a file in your afs space called "clearhosts" that looks like the following:

```
host1
host2
host3
```

and then issue the command:

```
netdb node comment --clear --input clearhosts
```

As you might be able to guess, sometimes it's easier to have the one file, especially if you want to make multiple changes to the same list. You can only issue one netdb command at a time. If, for example, you wanted to change the administrator, group and department for the same list of nodes, put the list in a file and apply the 3 different commands to it.

## Common Tasks:

- *Look up a record:*

```
netdb node info [node name, alias, sub-interface name]
```

This will get you the complete NetDB record, which includes fields hidden in whois or stanfordwhat (<http://stanfordwhat.stanford.edu>), such as the room number and custom fields. Whois is faster, but don't forget the "." as shorthand for .stanford.edu (i.e.: "whois host" will attempt to look up a person with the sunet ID of host, while "whois host." will look up a NetDB device called host). If it's not in the .stanford.edu domain, you'll need to specify the correct domain (.sunet etc.) or you'll get a cryptic java error.

You can't use the "info" search for IP number, but you can search whois or stanfordwhat by IP number (or use host or dig and then "info" for the name).

- *List choices:*

```
netdb list [groups, locations, models, oses, departments, states]
```

Any can be singular (netdb list group) instead.

Just list all the existing options for those fields. You can use the "\*" wildcard, but you'll have to quote (single or double quote) the whole string, and it is case sensitive, so you may have better luck dropping the first character:

```
corn:~> netdb list group "*network*"
MedIRT-Networking
Networking
Networking-NB
Networking-fws
networking-test-lab
```

If there are spaces, you'll need to use quotes even if you're not using a wildcard:

```
corn:~> netdb list department "Stanford Daily"
Stanford Daily

corn:~> netdb list department Stanford Daily
ERROR: You can pass only one filter to the list command
```

- *Add or delete aliases:*

```
netdb node alias --add [new alias, or multiple aliases separated by commas
with no spaces] --remove [alias to be removed, remove multiple ones separated
by commas with no spaces] [node]

netdb node alias --add alias3,alias4 --remove alias1 host
```

Will add two new aliases to the above sample, and remove one, leaving "alias2" alone for the node "host."

- *Changing administrators, with or without admin teams:*

```
netdb node admin --add [must be the sunet ID, no names, or admin teams with a
colon] --remove [administrator] INPUT
```

If you want to use an admin team, identify with a colon. To change the node `governorstanford`'s administrator from the user `lelandstanford` to the admin team "University Founders" the command would be:

```
netdb node admin --add "University Founders": --remove lelandstanford
governorstanford
```

The colon for the quoted admin team can be inside or outside the quotes.

- *Set department, comment, other text that may have spaces:*

```
netdb list department (to get your department, if you don't know it)
netdb node department -set "Department With Spaces" node
```

If the department name doesn't have spaces (somewhat rare) you can skip the quotes.

Some fields can be empty, for those, you may use the `--clear` modifier:

```
netdb node comment --clear node
```

## *The scariest command in command-line NetDB:*

```
netdb node delete INPUT
```

You can delete one record, multiple records by separating by spaces or a huge swath of nodes by referencing a file on your afs space.

There is **no** "undo" for the delete command, there is **no** "are you sure?" This is a large shotgun pointed at your feet, loaded for bear. Be careful. Although the NetDB database is backed up regularly, it's backed up as a whole database, so rolling it back rolls back all the changes made by every user. Simply put, you're not important enough to qualify for a roll-back because of the mistakes you may have made with the "delete" command.

### **Using the "Clone" command to make new records from templates.**

If you expect to need to create many new records that are largely similar, you may find it quicker to use the "clone" command than to use the web interface. The "clone" command behaves like the "use as a template" web option in that you can use either a regular node or a template node as the source. Here's the whole option set for the clone command:

```
netdb node clone --template node --name name
[ --location building:room | :room ]
[ --hardware|hw hardware address [ --dhcp [ --roam ] ] ]
[ --ip ip address[+] | none ]
[ --model make:model ] [ --os os,... ]
[ --user user,... ] [ --admin admin,... ]
[ --comment comment ]
```

If you simply use the most basic form specifying a template node and a new name, NetDB will apply the new name to the new node, pick a free IP number from the same subnet as the template node (and tell you what IP number it picked), and will carry over the type (Advanced or not), state, location, make, model, OS, user, administrator and comment field. It will carry over the labels for custom fields, but not the contents. If the template has multiple interfaces or multiple IP addresses for any one interface, the clone command will not be able to automatically pick a free IP address and will generate an error. In that case, you'll need to use the --ip modifier to specify an IP number, or at least specify a "seed" subnet. The "+" option tells NetDB to pick a free IP number from the same subnet as the specified number, just in case the specified number is taken.

For example, if an existing node ("old-node") has two IP numbers, 172.31.20.5 and 172.31.21.5 and you want to make a new record for a new host ("new-node") with the ethernet address of 0000.abba.cafe on the 172.24.31.0/24 subnet, but you don't care what number it picks, set up for DHCP but not roaming, keeping the user, administrator, make, model, OS and location, the command is:

```
netdb node clone --template old-node --name new-node --hardware 0000.abba.cafe
--dhcp --ip 172.31.21.5+
```

Since 172.31.21.5 is already taken, but you've used the "+," NetDB will pick a new IP number on that subnet.

## Creating and working with files for the --input option.

Command-line NetDB has no complex search functions, just the simple "netdb node info" command to show you one whole record. You'll still need to use the web form to do your complex searches, using the "Full Search ...Node" option. Once you've done your search, re-do the search but have NetDB only list the names, by changing the "Display" column on the right to "-" for everything except Node. You may not want to search for aliases, which is checked by default.

You'll have an output that just lists your record names. At the bottom, rarely noticed, is the link to "Generate result (of the last search) in CSV form." You'll have a plain text output starting with "Name:Node" and then listing all your nodes.

Select every item except "Name:Node" and paste it into your text editor of choice, then save the file to your afs space. If you don't have a Unix text editor of choice, pico is fairly easy to use. You can give the file any name you'd like, and there's no need for an extension. If you mount afs on your desktop, you can use a desktop text editor (note pad, text edit, BBedit, etc.) and save the file to your afs space.

You don't have to save the list to a file if you don't want, you can always apply a command to a huge list of nodes separated by spaces. Take the list from your search (after "Name:Node" of course), paste into your text editor of choice, replace every carriage return/line feed with a space and just append that list at the end of your command.

If your search results in:

```
host1
host2
...
host33
host34
```

You can save that list into your afs space (let's call it hostlist) straight from the NetDB CSV form which will have each on a separate line as desired, or convert it in your favorite text editor to:  
host1 host2 ... host33 host34

Then the command, for example, to add a new Admin Team called SuperTeam to all 34 records would be:

```
netdb node admin --add SuperTeam: --input hostlist
```

or

```
netdb node admin --add SuperTeam: host1 host2 ... host33 host34
```

Both will work the same way.

## Moving buildings and subnets, changing a large number of nodes.

One of the most tedious tasks that many of our users find is changing a whole building's worth of records as a department moves to a new building and new subnet.

- *Cleaning up unused IP numbers*

The first step should be to use Kent Reuber's "oldnodelist" script to help you clean up unused nodes. The IP Unused report from SUNet reports, <https://www.stanford.edu/group/networking/dist/sunet.reports/ipunused.list>, can be used to help you clean up and delete old nodes, but Kent's script is a little less brute force. The script is found at

```
~reuber/public/oldnodelist
```

and just typing the command gets the help screen. The most common use is to clean up anything not seen in a year, for which you'd use something like

```
~reuber/public/oldnodelist 365 171.64.20.0/24 > cleanpinehall
```

which will create a file in your afs space called "cleanpinehall" (you'll probably want to use a more appropriate name) that has text something like this:

```
# Host: 171.64.20.7
# -----
#
# 171.64.20.7 registered interfaces:
#   000d.937e.8ee7
#   000a.95c9.06b2
#   171.64.20.7
# 171.64.20.7: no listed interfaces in use.
#
/usr/pubsw/sbin/netdb node delete `~reuber/public/ip2host 171.64.20.7`
# -----
# Host: 171.64.20.13
# -----
#
# 171.64.20.13 registered interfaces:
#   001d.7286.f569
#   171.64.20.13
#   001f.3b16.52c1
#   171.66.254.210
# 171.64.20.13: other interfaces in use:
#
#   IP Address & Type   NetDB Entry   Hardware   First   Last   Times
#   Address & Type   NetDB Entry   Address   NIC Type   Seen   Seen   Seen
#   -----
#   10.31.4.7          d ---         001f3b:1652c1 ---    05/01/10 05/05/10   27
#   10.33.145.14       d Wireless-Quad14 001f3b:1652c1 ---    05/19/10 05/26/10    4
#   10.33.146.200      d Wireless-Quad14 001f3b:1652c1 ---    12/17/09 05/01/10   85
#   10.33.147.65       d Wireless-Quad14 001f3b:1652c1 ---    05/05/10 05/06/10    4
#   10.33.147.119      d Wireless-Quad14 001f3b:1652c1 ---    06/02/10 06/03/10    4
#   10.33.147.218     d Wireless-Quad14 001f3b:1652c1 ---    06/03/10 09/08/10  177
#
#
/usr/pubsw/sbin/netdb node ip_address --remove 171.64.20.13 `~reuber/public/ip2host
171.64.20.13`
```

Now that you've learned about command-line NetDB, you'll understand what the non-commented lines do. The commented text explains what the script learned and then the netdb command follows. The commands from this script either delete a whole node or delete an IP number from a node where the hardware address(es) have been active on other IP ranges. You'll want to look over the file and delete any lines that you disagree with, then you can either copy and paste each netdb command, or, if you're sure that the file is clean and will do all you want, execute the command



```
sh [filename]
```

to get the Unix shell to read through each line (ignoring all the commented lines) and executing each line as a command, one at a time.

- *Search with web NetDB to get your hosts into multiple .csv files*

Now that your unused nodes are cleaned up, using the web form, do your search to find all the hosts that will be moving. Hopefully, somewhere you have a spreadsheet with the information on the rooms into which each of these hosts will be moved. Let's assume your hosts are in the 10.20.30.0/24 IP range and that the location information is generally inaccurate. Let's also assume that every host in the 10.20.30.0/24 IP range is moving to a new location with the IP range of 10.100.100.0/24, and that you want to clean up the department and administrator information (using an admin team) while you handle the rest of the moves. Let's assume that all the hosts are using DHCP. Finally, let's assume that all the hosts will be offline and in boxes for a day or two so you can delete the old IP numbers while adding new ones while all the devices are boxed up.

In web NetDB, do a full search for host, enter 10.20.30.0/24, exclude the node types: "IP Connectivity Provider," "Template" and "Router." Display Node Name and IP Address. After you have that search, display in CSV form and save it. Re-do the search but now display only the Administrator field, and save that CSV file. Re-do the search a third time listing the node names and the user field and save that as a CSV file.

The CSV for the first display will look something like:

```
Name:Node,Interface#,Address
larry-desktop,1,10.20.30.10
moe-desktop,1,10.20.30.14
curly-desktop,1,10.20.30.15
shemp-desktop,1|2,10.20.30.40|10.20.30.41
```

As you can see, shemp-desktop is problematic. For some reason, it has two interfaces both on the same IP range. For those cases, you'll probably want to look at them and fix them one at a time through the web interface, because they may be servers. Delete those lines from your CSV file (which you're probably editing in Excel or your favorite spreadsheet program) and make note of them so you can fix them later.

The CSV for the second display of the same search will just be a strange looking long list of administrators by name, or Admin Teams with a semicolon after them. The purpose of that search is to see all the incorrect administrators and get a list. You'll need to figure out the SUNet ID's of all the regular people. Let's assume our hosts have one or more administrators who have the SUNet ID's of "larrystooge" "moestooge" and "curlystooge" and some are in the Admin Team "Stooges," while you really want them all to be in the Admin Team "Three Stooges."

From the third CSV file that lists the node names and users, you can probably figure out the new rooms that each host will go to based on the user name.

- *Convert the .csv files into NetDB commands with excel and a text editor*

Looking at the first CSV file, here's what it'll look like in Excel:

larry-desktop	1	10.20.30.10
moe-desktop	1	10.20.30.14
curly-desktop	1	10.20.30.15

Delete the contents of the 2nd column. Cut the contents of the 1st column and paste it two columns to the right of the IP number list. You'll now have this:

	10.20.30.10		larry-desktop
	10.20.30.14		moe-desktop
	10.20.30.15		curly-desktop

Which looks a little odd. Make all the cells "text" so that Excel doesn't freak out about "--" or other strange characters. In the first cell of the first column, enter "netdb node ip\_address --remove" and fill it down to all the rows. In the first cell of the third column, enter "--add 10.100.100.10" and fill it down such that Excel makes the next cell ".11" and ".12" etc. It'll now look like this:

netdb node ip_address --remove	10.20.30.10	--add 10.100.100.10	larry-desktop
netdb node ip_address --remove	10.20.30.14	--add 10.100.100.11	moe-desktop
netdb node ip_address --remove	10.20.30.15	--add 10.100.100.12	curly-desktop

Select all the contents and paste into your fairly smart text editor of choice. Make sure it's a text editor that doesn't like to add unexpected line feed/carriage return characters. Do a search and replace for all tabs (\t is a common way to reference the tab) and replace them all with spaces and you'll now have a plain text file like this:

```
netdb node ip_address --remove 10.20.30.10 --add 10.100.100.10 larry-desktop
netdb node ip_address --remove 10.20.30.14 --add 10.100.100.11 moe-desktop
netdb node ip_address --remove 10.20.30.15 --add 10.100.100.12 curly-desktop
```

Save that file into your afs space and when you're ready, execute the "sh" command and NetDB will go through that whole file, line by line, and change all your IP addresses.

Go back to your Excel file with the NetDB commands and delete the 2nd and 3rd columns. Change the contents of the 1st column to be:

```
netdb node admin --remove larrystooge,moestooge,curlystooge,stooges: --add
"Three Stooges":
```

And fill that all the way down. Copy and paste into your text editor, replace the tabs with spaces, and you'll have another file to save that will let you fix the administrators for all the hosts. It will look like:

```
netdb node admin --remove larrystooge,moestooge,curlystooge,stooges: --add
"Three Stooges": larry-desktop
netdb node admin --remove larrystooge,moestooge,curlystooge,stooges: --add
"Three Stooges": moe-desktop
netdb node admin --remove larrystooge,moestooge,curlystooge,stooges: --add
"Three Stooges": curly-desktop
```

You could also just have all the node names in a file, each on their own line, and execute the command:

```
netdb node admin --remove larrystooge,moestooge,curlystooge,stooges: --add
"Three Stooges": --input [filename]
```

Since most hosts won't have all four unwanted administrators, NetDB will send you lots of errors, but it will still make all the changes that are appropriate and just move on to the next command.

You could do the same thing with the department, but remember that departments with spaces need quotes, and that the department is a "--set" field, so there's no need to remove the old department:

```
netdb node department --set "Metro Goldwyn Mayer" larry-desktop (etc.)
```

Finally, to change the location. Your CSV file from the search where you listed the node names and users will look something like this:

```
Name:Node,Node User
larry-desktop,Larry Fine
moe-desktop,Moe Howard
curly-desktop,Curly Howard
shemp-desktop,Shemp Howard
```

Ideally, you have a file somewhere that lists the users and their destinations. Let's say that Larry is moving to room 101 of your new location ("MGM Studios") and Moe and Curly are moving to room 102 and Shemp is moving to room 105. Using find and replace and following some of the above steps, you'll end up with something like this:

```
netdb node location --set "MGM Studios":101 larry-desktop
netdb node location --set "MGM Studios":102 moe-desktop
netdb node location --set "MGM Studios":102 curly-desktop
netdb node location --set "MGM Studios":105 shemp-desktop
```

As you can see, since each host may have a different room, it's not possible to use the --input option to make the same change to all of them. The location field does support the --input option, if you want to specify the correct room, you'll need to use a different command for each node.

## **The netdb user command to look up users.**

Unless you have the ability to create new users, the only thing you'll need to do with the netdb user command is to look up a user's NetDB account to see what access they have.

```
netdb user info [sunset ID only]
```