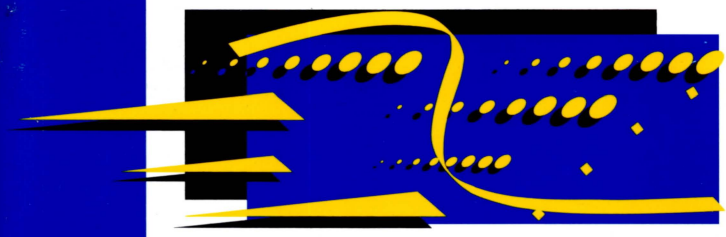

CONVEX

Using CONVEX Internet Services and NFS

Second Edition

Using CONVEX Internet Services and NFS Quick Reference

Order No. DSW-118
Document No. 710-023130-001

Released with CONVEX Internet Services and NFS V11.0.

Printed in the United States of America

Command summary

This reference describes how to use the CONVEX Internet Services and NFS utilities listed below. Utilities are in alphabetical order. Find the command name of the utility you wish to use, then flip to the page labeled with that command.

`chkey`	Change Secure NFS encryption key.
`domainname`	Show NIS domain of your system.
`finger`	Look up information about users.
`ftp`	Copy files to and from a remote system.
`hostname`	Print name of host system.
`keylogin`	Decrypt and store Secure NFS key.
`rcp`	Copy files to and from a remote system.
`rdiff`	Compare local or remote files.
`rdist`	Distribute files to remote systems.
`rex`	Execute commands remotely using NFS.
`rlogin`	Log in to a remote system.
`rsh`	Execute a command on a remote system.
`rup`	Show status of remote system.
`ruptime`	Show status of remote systems.
`rusers`	Show users of remote system.
`rwall`	Send message to users of remote system.
`rwho`	List users logged in to hosts on the local network.
`telnet`	Establish a session with a remote system.
`ypcat`	Display contents of NIS map.
`ypmatch`	Search for records in NIS map.
`yppasswd`	Change password on NIS system.
`ypwhich`	Show which host is NIS map server/master.

Some commands have more options than are listed here. For a complete technical reference, refer to the online ConvexOS man pages. References to ConvexOS man pages appear in the form rcp(1), where the name of the man page is followed by its section number.

Using command descriptions

This reference uses the following conventions:

- When instructed to *enter* a command, press the **RETURN** key after typing the command.
- When instructed to *type* a command, do *not* press the **RETURN** key after typing the command.
- The following typefaces have special meanings:

`Courier`	Signifies command names, system output, and error messages.
`Courier bold`	Type exactly as shown.
Italic	Substitute appropriate value.
KEYCAP	Press the indicated key. If keys are separated by a hyphen, press them simultaneously. For example, **CTRL-d** indicates that you press the **CTRL** key and the **d** key simultaneously.

- Brackets indicate an optional part of a command description. For example,

  `rwho` [*name...*]

  indicates that *name* is optional. Do not type the brackets.

- Horizontal ellipsis indicates that part of a syntax can be repeated. For example:

  `rwho` [*name...*]

  indicates that values for *name* can be repeated.

- In some examples, the ConvexOS prompt is shown as a percent sign (`%`). Do not type this symbol.

chkey

Enter a new NFS encryption key in the publickey(5) database.

```
chkey
```

The `chkey` command is used only with Secure NFS. `chkey` prompts for your login password, and uses it to encrypt a new encryption key to be stored in the publickey(5) database. A valid encryption key must already exist in the database; initial keys are set up by the superuser with newkey(8).

domainname

Find out the NIS domain of the host you are currently logged in to.

```
domainname
```

The domainname command is useful when your host is running NIS and you need to know where certain types of system information are coming from. The superuser can also use this command to set the domain.

Example

```
% domainname
nova
% rlogin aries
Last login: Wed Aug 19 16:15:35 from nova
% domainname
aries.stars.com
```

finger

Display information about users on your local host or on a remote host.

```
finger [options...] [[name][@host]...]
```

Options

`-f` Suppress headers in short or quick output formats.

`-h` Suppress printing of .project files.

`-i` Display quick list of users and include idle times.

`-l` Force long output format that includes content of users' .plan and .project files.

`-m` Match only on user name.

`-p` Suppress printing of .plan files.

`-q` Display quick list of users, similar to who(1) output.

`-s` [*name...*]
Force short output format.

Interpreting output

For each user logged in to your local host, `finger` displays:

- Login name
- Full name
- tty number
- Idle time
- Time at which user logged in
- Office location (if known)
- Phone number (if known)

Specifying user and host names

`finger` *name*
Display information about the specified user on the local host only. *name* can be account name, first name, or last name.

`finger` *name@host*
Display information about the named user on the specified remote host.

`finger` *@host*
Display information about all users logged in to the specified remote host.

`[[`*name*`][@`*host*`]]` can be repeated to specify a list of user and/or host names.

ftp

Transfer files to and from a remote host.

`ftp` [*options...*] *host*

Begin an `ftp` session in command mode, or, if *host* is supplied, connect to the specified remote *host*.

Options

`-d` Enable debug mode.
`-g` Disable file name globbing.
`-i` Disable interactive prompting during multiple file transfers.
`-n` Disable autologin upon initial connection.
`-u` Transfer files via UltraNet.
`-v` Enable verbose mode.

Obtaining information about `ftp` commands

`help`, `?`, or `remotehelp` [*command*]

Display information about all or specific commands.

Command summary

In command mode, indicated by the `ftp>` prompt, `ftp` accepts and executes the following commands:

`!` Invoke interactive shell on local host.

`append` *local_file* [*remote_file*]
Append local file to remote file.

`ascii` Set file transfer type to ASCII, the default.

`bell` Sound a bell after each transfer.

`binary` Set file transfer type to binary.

`bye/quit` Exit `ftp` and return to shell.

`case` Toggle remote host file name case mapping.

`cd` *rem_dir* Change remote working directory.

`cdup` Change remote directory to its parent.

`close` End connection; return to command mode.

`cr` Toggle carriage return stripping.

`delete` *rfile* Delete specified remote file.

`dir` *remote_dir* [*local_file*]
List contents of remote directory.

`form` *format* Set file transfer form to *format*.

`get` *remote_file* [*local_file*]
Copy remote file to local file.

`glob` Toggle file name expansion.

`hash` Toggle hash-sign printing.

`lcd` *dir* Change local working directory.

`ls` [*remote_dir*] [*local_file*]
List contents of remote directory.

`mdel` *remote_file...*
Delete one or more remote files.

`mdir` *remote_dir...* [*local_file*]
List contents of remote directories.

`mget` *remote_file...*
Copy remote files to local working directory.

`mkdir` *rdir* Create remote directory.

`mls` *rdir...* Print contents of remote directories.

`mode` [*name*] Set file transfer mode to *name*.

`modtime` *remote_file*
Show last modification time of remote file.

`mput` *local_file...*
Copy local files to remote working directory.

`nlist` [*remote_dir*] [*local_file*]
Print contents of remote directory.

`nmap` [*inpattern outpattern*]
Set/unset file name mapping mechanism.

`ntrans` [*inchars* [*outchars*]]
Set/unset file name character translation.

`open` *host* [*port*]
Connect to specified host FTP server.

`prompt` Toggle interactive prompting.

`proxy` *cmd* Run `ftp` command on secondary connection.

`put` *local_file* [*remote_file*]
Copy a local file to a remote destination.

`pwd` Print name of remote working directory.

`rstatus` [*remote_file*]
Show status of remote host or file.

`rename` [*from*] [*to*]
Rename file on remote machine.

`rmdir` *dir* Delete remote directory.

`size` *remote_file*
Return size of *rfile* on remote host.

`status` Show current status of `ftp`.

`system` Show type of remote operating system.

`type` [*t*] Set transfer `type` to *t*. Default is `ASCII`.

`user` *user_name* [*password*] [*account*]
Identify yourself to the remote FTP server.

`verbose` Toggle verbose mode. Default is ON.

`window` [*number* or *number*M]
Set the tcp window size used by `ftp` to *number* kbytes or megabytes.

hostname

Find out the name of the host you are currently logged in to.

```
hostname [-s]
```

Options

`-s` Trim any domain information from the host name.

The `hostname` command is useful for keeping track of where you are currently logged in when you log in from one host to another in a series.

Example:

```
% hostname
nova
% rlogin aries
Last login: Wed Aug 19 16:15:35 from nova
% hostname
aries
% rlogin gemini
Last login: Tue Aug 18 12:25:50 from cosmo
% hostname
gemini
% rlogin cosmo
Last login: Wed Aug 19 09:00:10 from aries
% hostname
cosmo
% logout
Connection closed.
% hostname
gemini
% logout
Connection closed.
% hostname
aries
% logout
Connection closed.
% hostname
nova
%
```

keylogin

Decrypt your NFS encryption key.

```
keylogin
```

The `keylogin` command is used only with Secure NFS. `keylogin` prompts for your login password, and uses it to decrypt your secret key stored in the publickey(5) database. Once decrypted, your key is stored by the local key server process keyserv(8C) to be used by any secure network services, such as NFS.

Normally, the `login` command does this work when you log in to the system, but running `keylogin` may be necessary if you did not enter a password when you logged in.

rcp

Copy files between hosts that support BSD networking.

`rcp` [*options...*] [[*user*@]*host* :]*source...* [[*user*@] *host* :]*dest*

user	Account name to use on the remote host if it differs from the name you use locally.
host	Name of remote host copied from or to.
source	Local or remote file or directory to copy from. Can be repeated to specify a list of files.
dest	Local or remote file or directory to copy to.

Remote relative path names are interpreted relative to your home directory on the remote host.

Options

`-r` Copy files recursively. If any of the source files are directories, `rcp` copies each subtree. In this case, *dest* must be a directory.

Example: `rcp -r local_tree aries:remote_dir`

`-p` Preserve the source file's access mode and modification times. By default, owner and access mode of *dest* are preserved if *dest* already exists; otherwise, *source*'s mode is modified by the umask on the destination host, and ownership of *dest* is set to your user name on the destination host.

Using metacharacters

To have metacharacters interpreted remotely, rather than locally, enclose remote path names in double (" ") or single (' ') quotes, or precede metacharacters with a backslash (\).

Examples:

```
rcp aries:/fortran_dir/\*.f fort_dir
```

Copies all files ending in `.f` from fortran_dir on remote host aries into the local directory, fort_dir.

```
rcp aries:"devel/*.c" my_c_progs
```

Copies all files ending in `.c` from the devel directory (relative to your remote home directory) into the local directory, my_c_progs.

Accessing remote systems

To use `rcp`, you must have a .rhosts or /etc/hosts.equiv file entry on any remote host copied from or to. The .rhosts and hosts.equiv files must be writable only by the owner of the file.

rdiff

Compare files on the local or remote hosts.

`rdiff` [*diff_options...*] [*host*:]*file* [*host*:]*file*

diff_options Passed directly to the `diff` utility. Refer to the diff(1) man page for a complete list of options.

host Name of the host on which a remote file resides.

file Local or remote file name or fully-qualified path name. Remote relative path names are interpreted relative to your home directory on the remote host. Unlike `diff`, `rdiff` cannot be used to compare directories.

Examples

`rdiff /etc/group aries:/etc/group`

Compares the /etc/group file on the local host with the /etc/group file on remote host aries.

`rdiff saki:/etc/passwd sushi:/etc/passwd`

Compares password files on remote hosts saki and sushi.

`rdiff hamlet:.login .login`

Compares your .login file on remote host hamlet with your .login file on the local host.

rdist

Distribute identical copies of files to multiple hosts.

`rdist` [*options...*] [`-c` *name...* [*login*@]*host*[:*dest*]]

Options

`-b` Perform a binary comparison and update files if they differ, rather than comparing dates and sizes.

`-c` *name*
Interpret remaining arguments as a small *distfile* equivalent to:

(*name...*) `->` [*login*@]*host*
`install` [*dest*];

`-d` *var=value*
Set the variable named *var* to *value*. Used to define or override variable definitions in *distfile*.

`-f` *distfile*
Name of file containing `rdist` commands. If no `-f` option is present, `rdist` searches for a file named distfile, then for Distfile.

`-h` Copy the file a link points to rather than the link itself.

`-i` Ignore unresolved links.

`-m` *host...*
Limit the number of hosts that are updated by specifying a subset of hosts listed in *distfile*.

`-n` Print *distfile* commands without executing them. Used for debugging *distfile*.

`-q` Suppress printing of files as they are modified.

`-R` Remove any files that exist on the remote host that do not exist in the master directory.

`-t` *timeout*
Number of minutes in which the remote host must respond before `rdist` breaks the connection.

`-u` Update files only; do not install files.

`-v` Verify that files are up-to-date on all hosts.

`-w` Append the whole file name to the name of the destination directory.

`-y` Do not update files younger than the master copy.

distfile formats

distfile entries have one of three formats:

variable_name = *name_list*
Assign *variable_name* to a list of files, directories, or hosts on which `rdist` will operate.

[*label*:] *source_list* `->` *dest_list command_list*
Copy files in *source_list* to hosts in *dest_list*.

[*label*:] *source_list* :: *time_stamp_file command_list*
List files in *source_list* that are newer than *time_stamp_file*.

where:

label	Optional identifier for later reference.
source_list *dest_list* *name_list*	Single item, or list enclosed in parentheses.
command_list	Optional list of `rdist` commands that add to or modify the function of *distfile* entries.

Variables to be expanded begin with `$` followed by a name enclosed in braces.

Example:
```
FILES = (.login .cshrc .mailrc.)
HOSTS = (eeny meany miney moe)
dots: ${FILES} -> ${HOSTS}
     install -i;
```

`rdist` commands

`install` [*options...*] *opt_dest_name*;
Copy each file in *source_list* to each host in *dest_list*. Optionally, rename *source_list* files to *opt_dest_name*.

`notify` [*name_list*];
Mail list of updated files to names in *name_list*.

`except` [*name_list*];
Update all files in *source_list* except those listed in *name_list*.

`except_pat` [*pattern_list*];
Update all files in *source_list* except files with names containing strings specified in *pattern_list*.

`special` [*name_list*] "*string*";
Execute shell commands listed in string after each file in *name_list* is updated or installed. *string* is enclosed in quotes. Separate multiple commands with semicolons.

Using metacharacters

To have metacharacters interpreted remotely, rather than locally, precede them with a backslash (\).

rex

Execute commands remotely using NFS.

`rex` [*options...*] *host command* [*arguments...*]

host Remote host on which to execute command.

command Command to execute on the remote host.

arguments... Arguments to the host command.

`rex` is similar to `rsh`, except that `rex` preserves your working directory and `rsh` does not. To use rex, the working file system must be mounted or exported to the remote system.

Options

`-d` Run command in debug mode.

`-i` Run command in interactive mode. Uses remote echoing and special character processing and reproduces terminal modes and window size changes. This option must be used with programs that expect to talk to a terminal.

`-n` Causes a remotely-executed program to get an end-of-file indication when it reads from standard input. Use the `-n` option when using `rex` to execute a command that does not require input.

Error messages

`unknown host`
: Host name not found

`cannot connect to server`
: Host down or not running the server

`can't find .`
: Problem finding the working directory

`can't locate mount point`
: Problem finding current file system

Other error messages may be passed back from the server.

rlogin

Log in to a remote host.

`rlogin` *rhost* [*options...*]

rhost Name of remote host to log in to.

Options

`-e`*char*
Use *char* (instead of ~) as the escape character to control the remote session. *char* must be a single character. There is no space between `-e` and *char*.

`-l` *username*
Log in under the specified *username* instead of your local account name. Your local user name and host name must be listed in *username*'s .rhosts file.

`-8` Transfer 8-bit bytes. Default is 7 bits.

Controlling the remote session

`logout`	Log out of the current remote host.
`~.`	Terminate all `rlogin` processes and return control to the local shell.
`~`**CTRL-d**	Terminate all `rlogin` processes and return control to the local shell.
`~`**CTRL-z**	Suspend the current remote session. The suspend character (**CTRL-z**) depends on which shell you use. The suspend character must be the first character on the command line.
`fg`	Resume a suspended remote session.

Accessing remote hosts

To access a remote host via `rlogin`, you must have an account on the remote host. To log in without having to enter your remote account name and password, one of the following conditions must be met:

1. Your account name on the remote host is the same as your account name on the local host, and the name of your local host is listed in the /etc/hosts.equiv file on the remote host.
2. You have a .rhosts file in your home directory on the remote host that lists your local host name and local account name. Your .rhosts file must be writable only by you.

If neither of the above conditions is met, the remote host will prompt you for your account name and password.

Error messages

`Unknown host`
: The remote host is not defined in the /etc/hosts file.

`Connection refused`
: The remote host returned a message refusing the connection.

`Network unreachable`
: No route to the remote host exists in the routing tables.

`Connection timed out`
: The remote host did not respond to the request for a connection. This message usually indicates that the remote host is down.

rsh

Execute a command on a remote host.

`rsh` *host* [*options...*] [*command*]

host Remote host on which to execute command.

command Command to execute on the remote host. You cannot use `rsh` to run interactive commands such as `emacs` or `vi`. To run interactive commands, use `rlogin`.

If you omit *command*, `rsh` will use `rlogin` to log you in to the remote host.

`rsh` is similar to `rex`, except that `rsh` does not export your working directory like `rex` does. To use `rsh`, you must have a .rhosts or /etc/hosts.equiv file entry on the remote host.

Options

`-l` *username*
Account name to use on the remote host if it differs from your local account name. Your local user name and host name must be listed in *username*'s .rhosts file.

Example: `rsh vega -l jones`

`-n` Redirect input of `rsh` on the remote host from /dev/null. Use the `-n` option when using `rsh` to execute a command that requires no input.

Using metacharacters

To have metacharacters interpreted remotely, rather than locally, enclose them in double quotes (" ") or precede them with a backslash (\).

Example:

`rsh vega cat fileA >> fileB`
Appends remote fileA on host vega to local fileB.

`rsh vega cat fileA ">>" fileB`
Appends remote fileA on host vega to remote fileB on host vega.

rup

Check the status of hosts on your local network or subnet.

```
rup [ -o ] [ -h | -l | -t ] [ hostname ... ]
```

hostname List the status of the named host only. *hostname* can be repeated to specify a list of hosts.

The `rup` command is similar to the `ruptime` command, but uses RPC. `rup` works across subnets if host names are explicitly given. It does not maintain a list of hosts and must send out status requests when run.

Options

`-h` Sort the display alphabetically by host name.

`-l` Sort the display by load average.

`-t` Sort the display by up time.

`-o` Use a larger timeout and do an extra broadcast for old versions of the rstatd daemon.

Interpreting output

For each host on your local network or subnet, or for each host specified on the command line, rup displays

- Host name
- Status (up or down) at the current time
- Uptime in the form: days, hours:minutes
- Load averages over the last 1, 5, and 15 minutes

By default, output is unsorted, with entries displayed as broadcast responses arrive from the hosts.

Examples

```
% rup aries
aries up 1 day, 13:36, load average: 4.32, 4.17, 4.34
% rup
vega up 1 day, 1:52, load average: 0.02, 0.00, 0.00
astra up 6 days, 3:19, load average: 0.90, 1.39, 1.25
gemini up 8 days, 8:18, load average: 0.62, 0.86, 0.77
```

ruptime

Check the status of hosts on your local network, or on your local subnet if your network is divided into subnets.

```
ruptime [options...] [hostname...]
```

hostname List status of the named host only. *hostname* can be repeated to specify a list of hosts.

`ruptime` displays information only for hosts running the `rwho` daemon.

Options

`-a` Include all users, regardless of their idle times.

`-l` Sort entries in order of decreasing load averages.

`-r` Reverse the sort order.

`-t` Sort entries in order of decreasing uptime or downtime.

`-u` Sort entries in order of decreasing number of users.

Interpreting output

For each host on your local network or subnet, or for each host specified on the command line, `ruptime` displays:

- Host name
- Status (`up` or `down`)—Machines for which no status has been received for 11 or more minutes are considered down.
- Uptime or downtime in the form: days+hours:minutes
- Number of users logged in to the host who have been idle for less than an hour
- Load averages over the last 1, 5, and 15 minutes

By default, output is sorted alphabetically by host name.

rusers

Find out who is logged in to hosts on your network or subnet.

```
rusers [-a][-l][-o][-h|-i|-u] [ hostname ...]
```

hostname List users on the named host only. *hostname* can be repeated to specify a list of hosts.

The `rusers` command is similar to the `rwho` command, but uses RPC. `rusers` works across subnets if host names are explicitly given.

Options

`-a` Give a report for a machine even if no users are logged on.

`-h` Sort alphabetically by host name.

`-i` Sort by idle time.

`-l` Give a longer listing in the style of who(1).

`-u` Sort by number of users.

`-o` Use a larger timeout as well as doing an extra broadcast for very old versions of the `rusersd` daemon.

Interpreting output

For each user logged in to a host on your local network or subnet, `rusers` displays

- Host name
- List of users

Examples

```
% rusers
aries joe_bob joe_user jim_dand
vega joe_bob elliemae
gemini jane granny jed jethro
% rusers -l gemini
jane gemini:ttypb May 1 14:41 27 (cosmo)
granny gemini:ttypc May 1 15:05 2 (cosmo)
jed gemini:console May 1 10:38 56
jethro gemini:ttype May 1 10:38 56 (nova)
```

rwall

Send a broadcast message to all users on one or more remote hosts.

```
rwall host ... [-n netgroup ...] [-h host ...]
```

With the -n option, the message can be sent to a netgroup (see netgroup(5)).

Example

```
% rwall aries
There will be a test of the fire alarm in
Building A on Wednesday at 7:00 A.M.
^D
```

rwho

Find out who is logged in to hosts on your local network, or on your local subnet if your network is divided into subnets.

```
rwho [options...] [name...]
```

name Display information about the named user only. *name* can be repeated to specify a list of users.

`rwho` displays information only for hosts running the `rwho` daemon.

Options

`-a` Include all users, regardless of their idle times.

`-i` Sort output first by user name, then by idle time, then by host name, then by tty number.

`-m` Display idle time in minutes.

Interpreting output

For each user who is logged in to a host on your local network or subnet and who has been idle for less than an hour, or for each user specified on the command line, `rwho` displays:

- Account name
- Name of the host the user is logged in to
- tty number
- Date and time the user logged in
- Idle time

By default, output is sorted first by user name, then by host name, then by tty number.

telnet

Communicate with a remote host via the TELNET protocol.

`telnet` [*options...*] [*host* [*port*]]

Options

`-d` Set debug toggle to TRUE.

`-n` *tracefile* Open *tracefile* for recording trace information.

Execution modes

`telnet` runs in two modes:

command mode At the `telnet>` prompt, `telnet` executes the commands listed below.

connected mode `telnet` is transparent, allowing you to use the remote host as if it were local.

If *host* or *port* is supplied, `telnet` performs an `open` command to the specified remote *host* or connects to the `telnet` server at *port*, and enters connected mode. If *host* and *port* are omitted, `telnet` enters command mode.

To invoke command mode while in connected mode, type the `telnet` escape character, which is initially set to **CTRL-]**.

Obtaining information about `telnet` commands

`?` [*command*] Display list of `telnet` commands or a description of the specified command.

Command summary

`close` Disconnect and return to command mode.

`display` [*argument...*]
Display all or selected `set` and `toggle` values.

`mode` *type* Set user input mode to the specified *type*:

`character`	Character-at-a-time mode
`line`	TELNET LINEMODE
`isig(-isig)`	Enable (disable) TRAPSIG mode
`edit(-edit)`	Enable (disable) EDIT mode
`?`	Get help information for `mode`

`open` *host* [*port*]
Connect to *host* through *port* or default port.

`quit` Close `telnet` session and exit.

`send` *args* Send specified *args* to remote host. *args* can be:

`abort`	ABORT processes sequence
`escape`	Current `telnet` escape character
`ga`	Go Ahead sequence
`getstatus`	TELNET status request
`ip`	Interrupt Process sequence
`?`	Get help information for `send`

`set` *var value* Set variable *var* to specified *value* or TRUE.

`unset` *vars...* Set variable *var* to FALSE (disabled). *vars* can be:

`echo`	Character to enable local echoing
`eof`	Character to cause an EOF
`erase`	Character to erase a character
`escape`	`telnet` escape character
`tracefile`	File to receive trace information
`worderase`	Character used to erase a word
`?`	Display `set` (`unset`) commands

`toggle` *args...* Toggle `telnet` *args* (TRUE / FALSE): *args* can be:

`autoflush`	Output flushing
`autosync`	Send characters in urgent mode
`binary`	Send and receive binary data
`inbinary`	Receive binary data
`outbinary`	Send binary data
`crlf`	Send carriage return as <CR><LF>
`crmod`	Map received carriage returns
`?`	Get help information for `toggle`

`z` Used with csh(1), suspends `telnet`.

`!` [*command*] Execute *command* on local host.

`status` Show current `telnet` status.

ypcat

Examine the contents of an NIS map.

```
ypcat [ -k ] [ -t ] [ -d domainname ] mname
ypcat -x
```

Use the `ypcat` command when your host is running NIS to determine what system information your current host is using. The first form displays the contents of an NIS map. The second form displays the map name translation table.

Options

-k	Print the key before each value in the map.
-d *domain*	Print information about maps in a domain other than the default domain.
-t	Inhibit nickname translations on map names.
-x	Display the map name translation table.

Example

```
% ypcat rpc
.
.
.
rexd 100017 rex
.
.
.
% ypcat -k rpc
.
.
.
100017 rexd 100017 rex
.
.
.
% ypcat -x
Use "passwd" for map "passwd.byname"
Use "group" for map "group.byname"
Use "networks" for map "networks.byaddr"
Use "hosts" for map "hosts.byaddr"
Use "protocols" for map "protocols.bynumber"
Use "services" for map "services.byname"
Use "aliases" for map "mail.aliases"
Use "ethers" for map "ethers.byname"
Use "rpc" for map "rpc.bynumber"
Use "pwrestrict" for map "pwrestrict.byname"
% ypcat -t rpc.bynumber
.
.
.
rexd 100017 rex
.
.
.
```

ypmatch

Search NIS maps for particular entries.

```
ypmatch [ -d domain ] [ -k ] [ -t ] key . . . mname
ypmatch [ -x ]
```

Use the `ypmatch` command when your host is running NIS. The second form of the command is only used to display the map name translation table.

Options

-k	Print the key before each value in the map.
-d *domain*	Print information about maps in a domain other than the default domain.
-t	Inhibit nickname translations on map names.
-x	Display the map name translation table.

Example

```
% ypmatch rex rpc
rexd 100017 rex
% ypmatch -k rex rpc
rex: rexd 100017 rex
% ypmatch -x
Use "passwd" for map "passwd.byname"
Use "group" for map "group.byname"
Use "networks" for map "networks.byaddr"
Use "hosts" for map "hosts.byname"
Use "protocols" for map "protocols.bynumber"
Use "services" for map "services.byname"
Use "aliases" for map "mail.aliases"
Use "ethers" for map "ethers.byname"
Use "rpc" for map "rpc.byname"
Use "pwrestrict" for map "pwrestrict.byname"
% ypmatch -k -t rex rpc.byname
```

yppasswd

Change your password on a system running NIS.

```
yppasswd [ name ]
```

Use the `yppasswd` command rather than `passwd` to change your password if your system is running NIS. The password is changed across the entire NIS domain.

If no name is supplied, your own password is changed.

Example

```
% yppasswd
Changing yp password for joe_user
Old yp password:
New password:
Retype new password:
Rebuilding passwd database...
yellow pages passwd changed on vega
secret key reencrypted for unix.100@vega on vega
```

ypwhich

Determine the NIS server to which your host is bound, or find out which host has the master copy of an NIS map.

```
ypwhich [ -d domain ] [ -V1 | -V2 ] [ hostname ]
ypwhich [ -d domain ] [ -t ] -m [ mname ]
ypwhich -x
```

Use the `ypwhich` command when your host is running NIS and you need to know where certain types of system information are coming from. The first form inquires about the currently bound server providing NIS information to the default domain. The second form inquires about the location of one or all master NIS maps. The third form displays the map name translation table.

Options

-d *domain*	Print information about a domain other than the default domain.
-V1	Show server providing service via the old version of the NIS protocol.
-V2	Show server providing service via the current version of the NIS protocol.
-m [*mname*]	Print the hostname where the named master NIS can be found, or show masters for all maps if no argument is given.
-t	Inhibit nickname translations on map names.
-x	Display the map name translation table.

Example

```
% ypwhich
aries
% ypwhich -d stars
gemini
% ypwhich gemini
aries
% ypwhich -d stars gemini
gemini
% ypwhich -x
Use "passwd" for map "passwd.byname"
Use "group" for map "group.byname"
Use "networks" for map "networks.byaddr"
Use "hosts" for map "hosts.byaddr"
Use "protocols" for map "protocols.bynumber"
Use "services" for map "services.byname"
Use "aliases" for map "mail.aliases"
Use "ethers" for map "ethers.byname"
Use "rpc" for map "rpc.bynumber"
Use "pwrestrict" for map "pwrestrict.byname"
% ypwhich -m passwd
stars
% ypwhich -t -m passwd.byname
stars
```

ORDER NUMBER
DSW-118

DOCUMENT NUMBER
710-023130-001