

**NAME**

`nfsgid` – Prints a group ID map entry

**SYNOPSIS**

```
nfsgid -g map_file group [group ...]
nfsgid -m map_name [-d dump] [-s system] group [group ...]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsgid` command prints in a readable format the specified group entries from either the specified group ID map file, or the specified group ID map in the kernel or kernel dump. This command is useful mainly for diagnostic purposes.

The `nfsgid` command accepts the following options:

<code>-g <i>map_file</i></code>	Specifies the name of the group ID map file in which to find the entries to print.
<code>-m <i>map_name</i></code>	Specifies the name of the kernel group ID map in which to find the entries to print.
<code>-d <i>dump</i></code>	Reads the kernel dump file <i>dump</i> instead of <code>/dev/kmem</code> .
<code>-s <i>system</i></code>	Uses the UNICOS kernel binary file <i>system</i> instead of <code>/unicos</code> to obtain symbol information.

`group [group ...]` Specifies names of groups for which to find and print entries.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
<code>system, secadm, sysadm</code>	Allowed to use this command.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

The following command finds the group ID map entries from the kernel group ID map named `remote` for the local login names `local_group1` and `local_group2`, and it prints the contents of each entry if it is in the map.

```
nfsgid -m remote local_group1 local_group2
```

**FILES**

`/etc/gidmaps/nfsgid` Prints a group ID map entry

**SEE ALSO**

`nfsaddhost(8)`, `nfsaddmap(8)`, `nfsadduser(8)`, `nfsckhash(8)`, `nfsclear(8)`, `nfsidmap(8)`, `nfsidmem(8)`, `nfslist(8)`, `nfsmerge(8)`, `nfsrmhost(8)`, `nfsrmmmap(8)`, `nfsrmuser(8)`, `nfsuid(8)`  
`privtext(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011  
*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nfsidmap` – Reports or changes the state of NFS ID mapping in the kernel

**SYNOPSIS**

`/etc/uidmaps/nfsidmap [-d]`

`/etc/uidmaps/nfsidmap [-e]`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

If no options are specified, the `nfsidmap` command reports the state of network file system (NFS) ID mapping (enabled or disabled). Any user can use this command to determine the current state of NFS ID mapping.

Only an appropriately authorized user may change the state of NFS ID mapping. The `nfsidmap` command accepts following options:

`-d` Disables NFS ID mapping in the kernel. If mapping is already disabled, the disabled state is reported.

`-e` Enables NFS ID mapping in the kernel. If mapping is already enabled, the enabled state is reported.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
<code>system, secadm</code>	Allowed to change the state of NFS ID mapping.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to change the state of NFS ID mapping.

**SEE ALSO**

`nfsaddhost(8)`, `nfsaddmap(8)`, `nfsadduser(8)`, `nfsckhash(8)`, `nfsclear(8)`, `nfsgid(8)`, `nfsidmem(8)`, `nfslist(8)`, `nfsmerge(8)`, `nfsrmhost(8)`, `nfsrmmap(8)`, `nfsrmuser(8)`, `nfsuid(8)`  
`privtext(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011  
*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nfsidmem` – Reports the amount of kernel memory that NFS ID mapping is using

**SYNOPSIS**

```
/etc/uidmaps/nfsidmem [-g] [-i] [-u] [-d dump] [-s system]
/etc/uidmaps/nfsidmem [-v] [-d dump] [-s system]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsidmem` command reports the number of network file system (NFS) ID mapping domains, the number of user maps, and the number of group maps that have been inserted in the kernel. It also reports the number of bytes of kernel memory that have been allocated and the number of bytes of kernel memory that NFS ID mapping is actually using.

The `nfsidmem` command also reports any ID maps in the kernel that are not referenced by an ID mapping domain. Unreferenced maps generally indicate an administrative error: either the unreferenced maps should not have been added to the kernel, or an ID mapping domain that references these maps should be added.

The `nfsidmem` command accepts the following options:

- `-g` Gives detailed information on the kernel group ID maps. In addition to the totals described previously, the size of each group ID map is reported.
- `-i` Gives detailed information about memory usage of NFS ID mapping domains.
- `-u` Gives detailed information on the kernel user ID maps. In addition to the totals described previously, the size of each user ID map is reported.
- `-d dump` Reads the kernel dump file *dump* instead of `/dev/kmem`.
- `-s system` Uses the UNICOS kernel binary file *system* instead of `/unicos` to obtain symbol information.
- `-v` Gives more detailed information on each kernel ID map and the memory used by the NFS ID mapping domains. In addition to the totals described previously, the size of each ID map is reported. Using this option is the same as using the `-g`, `-i`, and `-u` options.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

Active Category	Action
system, secadm, sysadm	Allowed to use this command.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command.

**SEE ALSO**

`nfsaddhost(8)`, `nfsaddmap(8)`, `nfsadduser(8)`, `nfsckhash(8)`, `nfsclear(8)`, `nfsidmap(8)`, `nfslist(8)`, `nfsmerge(8)`, `nfsrmhost(8)`, `nfsrmmmap(8)`, `nfsrmuser(8)`, `nfsgid(8)`, `nfsuid(8)`  
`privtext(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011  
*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nfslist` - Lists all kernel ID mapping domains

**SYNOPSIS**

```
nfslist [-f] [-b] [-r] [-u] [-z] [-d dump] [-s system] [map_names]
nfslist [-f] [-b] [-r] [-d dump] [-s system] [-a address]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfslist` command lists all currently defined network file system (NFS) ID mapping domains in the kernel. For each ID mapping domain, `nfslist` prints the name of the map, mapping flags, the address range, and the address mask. It then prints out all addresses within the address range covered by the mask. For each matching address, the following information is printed: a byte-ordered decimal address, followed by a byte-ordered hexadecimal address, followed by the name from the `/etc/hosts` or `/etc/networks` file associated with that address. (See the `hosts(5)` and `networks(5)` man pages.) Map names are generally named for the remote administrative domain associated with the remote address.

The `nfslist` command accepts the following options:

- `-f` Lists all ID mapping domain information, even if NFS ID mapping is currently disabled. If ID mapping is turned off, the listing of any domains defined in the kernel is suppressed unless the `-f` option was specified.
- `-b` Lists the short form of the ID mapping domain information. The list of names for each address in the address range is not displayed.
- `-r` Displays the depth in the radix tree, the reference count, and the usage count of each ID mapping domain structure. This option is used primarily for problem diagnosis.
- `-u` Displays only ID mapping domains with nonzero usage counts.
- `-z` Displays only ID mapping domains with zero usage counts.
- `-d dump` Reads the kernel dump file *dump* instead of `/dev/kmem`.
- `-s system` Uses the UNICOS kernel binary file *system* instead of `/unicos` to obtain symbol information.
- map\_names* Specifies names of maps to be listed.
- `-a address` For the specified Internet address, lists the ID mapping domains that contain the address. The address can be a host name, a network name, or a byte-ordered Internet address. This option is mainly used for diagnosing ID mapping set-up problems.

The map flags are as follows:

- BAD\_ID, NOBODY, or MAP\_THRU** Indicates how a user ID not in the user map will be treated. It can be mapped to a bad ID (-1), mapped to nobody (-2), or mapped through without change.
- CLIENT** Indicates that the specified map is used to map IDs on the client side of the NFS protocol.
- CRAY** Indicates that all of the machines specified by the address range can map IDs in the same manner as that of the Cray Research system. Currently, they must be Cray Research machines. This is useful for correct access checking when mapping IDs.
- SERVER** Indicates that the specified map is used to map IDs on the server side of the NFS protocol.

If one or more map names are specified on the command line, only the ID mapping domains associated with those maps are printed. The special name MAP\_THRU is used to list domains for which IDs are the same as the current host.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

Active Category	Action
system, secadm, sysadm	Allowed to use this command.

If the PRIV\_SU configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

The following example shows sample output from `nfslist`. There are three ID mapping domains listed.

The first output references the MAP\_THRU map using the loopback address and maps IDs for both the NFS client and NFS server. IDs not found in the user map are automatically mapped to BAD\_ID (-1).

The second output references a map named `absurd` using an internet host address for a host known as `absurd`. IDs are only mapped on the NFS client and the host known as `absurd` appears to be another Cray machine using Cray style ID mapping. IDs not found in the user map are automatically mapped to NOBODY (-2).

The third output references a map named `peanut` using an internet host address for a host known as `peanut`. IDs are only mapped on the NFS server and the host known as `peanut` appears to be another Cray machine using Cray style ID mapping. IDs not found in the user map are automatically mapped to BAD\_ID (-1).

```
NFS ID Mapping is : ENABLED

NFS ID      NFS ID
Map         Mapping
Name       Flags                Address      Mask
MAP_THRU   CLIENT  SERVER  BAD_ID          7f.00.00.00 [ff.00.00.00]
```

## NFSLIST(8)

## NFSLIST(8)

				Addr(dec)	Addr(hex)	Host Name
				127.0.0.0	7f.00.00.00	loopback
absurd	CLIENT	NOBODY	CRAY	ff.aa.07.01	[ff.ff.ff.ff]	
				Addr(dec)	Addr(hex)	Host Name
				255.170.7.1	ff.aa.07.01	absurd
peanut	SERVER	BAD_ID	CRAY	ee.bb.70.01	[ff.ff.ff.ff]	
				Addr(dec)	Addr(hex)	Host Name
				238.187.112.1	ee.bb.70.01	peanut

## FILES

/etc/uidmaps/nfslist      Lists all kernel mapping domains

## SEE ALSO

nfsaddhost(8), nfsaddmap(8), nfsadduser(8), nfsckhash(8), nfsclear(8), nfsgid(8),  
nfsidmap(8), nfsidmem(8), nfsmerge(8), nfsrmhost(8), nfsrmmap(8), nfsrmuser(8),  
nfsuid(8)

privtext(1) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

hosts(5), intro(4), networks(5) in the *UNICOS File Formats and Special Files Reference Manual*,  
Cray Research publication SR-2014

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304



**NAME**

`nfsmerge` – Creates a user or group mapping between the local Cray Research administrative environment and a remote administrative environment

**SYNOPSIS**

```
nfsmerge [-l pef] [-e gef] [-u umf] [-g gmf] lpswd rpswd lgrp rgrp
nfsmerge [-L pef] [-e gef] [-u umf] [-g gmf] lpswd rpswd lgrp rgrp
nfsmerge [-L pef] [-E gef] [-u umf] [-g gmf] lpswd rpswd lgrp rgrp
nfsmerge [-l pef] [-E gef] [-u umf] [-g gmf] lpswd rpswd lgrp rgrp
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsmerge` command creates both a user ID mapping file and a group ID mapping file for use between the local Cray Research systems and remote administrative environments. network file system (NFS) ID mapping is necessary when users and/or groups do not have the same numerical IDs in two different administrative environments. The ID mapping files contain binary information suitable for use by the NFS routines in the UNICOS kernel.

To produce ID mapping files, the `nfsmerge` command uses the name fields from files in `passwd(5)` and `group(5)` formats, assuming that a common name in the `passwd` files of two administrations refers to the same user. Similarly, `nfsmerge` assumes that a common name in the `group` files of the two administrations refers to the same group. IDs that correspond to names that match exactly, or are specified as equivalent through the use of an exceptions file, are put into the map file. Exceptions files are explained in the following paragraph.

Some users and groups do not necessarily have common names between the local and remote administrations. For example, a user may not be able to have the same login name for the two administrations; someone else may already be using the name on one of the systems. Another example is that of two different users, each with an account in only one of the two administrations, and having the same login name. (The **EXAMPLES** section provides more explicit examples of these situations.) You can resolve these situations with the use of an exceptions file.

An *exceptions file* contains exceptions between names in the local and remote `passwd` or `group` files. For each entry in a login name exceptions file, there is a login name for a user in the local administration and an optional equivalent login name for the same user in the remote administration, separated by white space on a single line. If the name for the user in the remote administration is not present, the remote name is assumed to be the same as the one in the local administration. This is useful if you are forcing the use of exceptions by using either the `-L` or the `-E` option. Forcing the use of the exceptions file means that IDs for the names that appear in the exceptions file are the only ones put into the ID mapping file. A group name exceptions file has the same format and produces similar results for the group ID mapping.

The `nfsmerge` command produces ID mapping files in formats suitable for use by the `nfsaddmap(8)` command. The `-u` option allows you to specify a name for the user ID map file that is produced. The `-g` option allows you to specify a name for the group ID map file that is produced. If the names of the ID mapping files are not specified, `nfsmerge` generates a unique name, using the `mktemp(3C)` library routine. A template of `usrmap.XXXXXXX` is used for the user ID map file. A template of `grpmap.XXXXXXX` is used for the group ID map file. It is strongly recommended, however, that you specify on the command line a meaningful name for the ID mapping files.

The names of the local and remote password files, the names of the local and remote group files, the name of the exceptions file(s) (if being used), a list of user names for which IDs have been mapped, and a list of group names for which group IDs have been mapped are all written to standard output. Name exceptions are noted in the standard output by separating the local name from the remote name with an arrow symbol (`==>`).

See the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304, for more detailed information on configuring and administrating NFS with ID mapping.

The `nfsmerge` command accepts the following options and operands (if no options are specified, unique names for the user and group map files are chosen by the `mktemp(3C)` library routine):

- `-l pef` Specifies the login name exceptions file.
- `-L pef` Maps only the login names in the login name exceptions file. This option forces the use of the login name exceptions file.
- `-e gef` Specifies the group name exceptions file.
- `-E gef` Maps only the group names in the group name exceptions file. This option forces the use of the group name exceptions file.
- `-u umf` Specifies the name of the user ID map file.
- `-g gmf` Specifies the name of the group map file.

The following arguments must appear on the command line in the order specified:

- `lpswd` Specifies the name of a `passwd(5)` format file that contains user information for the local Cray Research system.
- `rpswd` Specifies the name of a `passwd(5)` format file that contains user information for the remote administrative environment.
- `lgrp` Specifies the name of a `group(5)` format file that contains group information for the local Cray Research system.
- `rgrp` Specifies the name of a `group(5)` format file that contains group information for the remote administrative environment.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
system, secadm, sysadm	Allowed to use this command.

If the PRIV\_SU configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

Example 1: The following command line creates a user ID map file named `u.local.remote` and a group ID map file named `g.local.remote` from the specified password and group files (the password and group files need not be sorted). Any name from the local password file, `passwd.local`, that appears in the exceptions file, `l.local.remote`, also is mapped if the remote name from the exceptions file is found in the remote password file, `passwd.remote`. All output created by this command is redirected into the `LOG.local.remote` file; that is, `LOG.local.remote` functions as a log file.

```
nfsmerge -l l.local.remote -u u.local.remote -e e.local.remote \
-g g.local.remote passwd.local passwd.remote \
group.local group.remote > LOG.local.remote
```

Example 2: Shows the contents of `l.local.remote` (the login name exceptions file in the previous command line):

```
jdoh      johnd
lvp       linus
lucy      lvp
bin       nobody
sys       nobody
dtm       dennis
fsmith    fred
```

The information in this file indicates that the IDs for the name `jdoh` on the local Cray Research system are mapped to the IDs for the name `johnd` on the remote system. This file also indicates that user `lvp` on the local Cray Research system is user `linus` on the remote system. However, user `lvp` on the remote system has a user name of `lucy` on the local Cray Research system.

**SEE ALSO**

nfsaddhost(8), nfsaddmap(8), nfsadduser(8), nfsckhash(8), nfsclear(8), nfsidmap(8),  
nfsidmem(8), nfslist(8), nfsuid(8), nfsrmuser(8), nfsrmhost(8), nfsrmmmap(8)

privtext(1) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

mktemp(3C) in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

group(5), intro(4), passwd(5) in the *UNICOS File Formats and Special Files Reference Manual*, Cray  
Research publication SR-2014

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nfsrmhost` – Removes an ID mapping domain host address or a range of host addresses

**SYNOPSIS**

```
nfsrmhost -l addr [-u addr] [-m mask]
nfsrmhost -u addr [-l addr] [-m mask]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsrmhost` command removes network file system (NFS) ID mapping domains from the kernel based on the range of network addresses associated with the domain. The range of network addresses should be equivalent to the addresses used when defined with the `nfsaddhost(8)` command. If the address specification does not exactly match a currently defined domain, an error is returned. Use the `nfslist(8)` command to list the ID mapping domains currently defined in the kernel.

You must specify either the `-l` or the `-u` option. `nfsrmhost` accepts the following options:

- `-l addr` Specifies a lower-bound network address that is associated with the specified map names. If you omit this option, the address becomes, by default, the upper-bound address specified by the `-u` option.
- `-u addr` Specifies an upper-bound network address that is associated with the specified map names. If you omit this option, the address becomes, by default, the lower-bound address specified by the `-l` option.
- `-m mask` Specifies a byte-ordered ASCII representation of a bit mask that is used in conjunction with the range of addresses specified on the command line to specify further the valid network addresses for this ID mapping domain.

A *network address* is a host name or host alias from the `/etc/hosts` file (see `hosts(5)`), a network name or network alias from the `/etc/networks` file (see `networks(5)`), or a byte-ordered ASCII representation of a network address, as described in `inet(3C)` in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
system, secadm, sysadm	Allowed to use this command.

If the PRIV\_SU configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

The following command lines remove two separate domains that were previously defined in the kernel by using the `nfsaddhost(8)` command (see the examples in the `nfsaddhost(8)` man page to see how those domains were added to the kernel):

```
nfsrmhost -l Cray-N-inet
nfsrmhost -l aaa-inet -u zzz-inet -m 0xffffffffe
```

**FILES**

`/etc/uidmaps/nfsrmhost` Remove NFS ID mapping domains

**SEE ALSO**

`nfsaddhost(8)`, `nfsaddmap(8)`, `nfsadduser(8)`, `nfscckhash(8)`, `nfsclear(8)`, `nfsgid(8)`, `nfsidmap(8)`, `nfsidmem(8)`, `nfslist(8)`, `nfsmerge(8)`, `nfsrmmmap(8)`, `nfsrmuser(8)`, `nfsstat(8)`, `nfsuid(8)`

`privtext(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`inet(3C)` in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

`hosts(5)`, `intro(4)`, `networks(5)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nfsrmmap` – Removes a previously defined user and/or group ID map

**SYNOPSIS**

```
nfsrmmap [-F] map_name
[map_name map_name ...]
nfsrmmap [-F] -M [map_name map_name ...]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsrmmap` command removes the specified user ID map and/or group ID map file(s), which were previously added to the kernel by using the `nfsaddmap(8)` command. You may remove the maps only if they are not being referenced by an ID mapping domain. If the specified map is not currently defined in the kernel, an error is returned.

This command accepts the following options and operand:

- F Forces the removal of a map even when entries are in a user map with Kerberos validated addresses. This effectively prevents those users from accessing and exporting file systems that require kerberos validated addresses.
- M Removes the special `mapthru` map if it is defined in the kernel. If secure UNICOS or kerberized network file system (NFS) is running, this special `mapthru` map may be required. This effectively prevents those users from accessing and exporting file systems that require Kerberos validated addresses. See the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304, for detailed information on this special ID map.
- map\_name* Specifies the name of the kernel ID map to delete. A list of these may exist on the command line.

By default, both the user and group maps that have the name *map\_name* are removed from the kernel.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
system, secadm, sysadm	Allowed to use this command.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

The following command line deletes from the kernel the user ID maps and group ID maps named `remote` and `other`:

```
nfsrmmap remote other
```

**FILES**

`/etc/uidmaps/nfsrmmap`      Removes a map

**SEE ALSO**

`nfsaddhost(8)`, `nfsaddmap(8)`, `nfsadduser(8)`, `nfsckhash(8)`, `nfsclear(8)`, `nfsgid(8)`,  
`nfsidmap(8)`, `nfsidmem(8)`, `nfslist(8)`, `nfsmerge(8)`, `nfsrmhost(8)`, `nfsrmuser(8)`,  
`nfsstat(8)`, `nfsuid(8)`

`privtext(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304



**NAME**

`nfsrmuser` – Removes a user ID map entry from a kernel ID map

**SYNOPSIS**

```
nfsrmuser [-F] map_name user1 [user2 user3 ...]
```

```
nfsrmuser [-F] -M user1 [user2 user3 ...]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsrmuser` command removes a user ID map entry from a kernel ID map.

This command accepts the following options and operands:

- F Forces the removal of the specified user ID map entries even if Kerberos validated addresses are associated with any of the entries. This effectively prevents those users from accessing and exporting file systems that require Kerberos-validated addresses.
- M Removes user ID map entries from the special `mapthru` map if it is defined in the kernel. This effectively prevents those users from accessing and exporting file systems that require Kerberos-validated addresses. If secure UNICOS or kerberized network file system (NFS) is running, this special `mapthru` map may be required. See the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304, for detailed information on this special ID map.

*map\_name* The kernel map from which to remove the entries.

*user1 [user2 user3 ...]*

The local login names of the user ID map entries will be removed from the specified kernel map.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
<code>system, secadm, sysadm</code>	Allowed to use this command.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

The following command gets the user ID (UID) for each user name specified on the command line (`local_user1` and `local_user2`), looks for the user entry in the kernel map named `map_name`, and removes the entry if it is found.

```
nfsrmuser remote local_user1 local_user2
```

**FILES**

`/etc/uidmaps/nfsrmuser` Removes a user ID map entry from a kernel ID map

**SEE ALSO**

`nfsaddhost(8)`, `nfsaddmap(8)`, `nfsadduser(8)`, `nfsckhash(8)`, `nfsclear(8)`, `nfsgid(8)`, `nfsidmap(8)`, `nfsidmem(8)`, `nfslist(8)`, `nfsmerge(8)`, `nfsrmhost(8)`, `nfsrmmap(8)`, `nfsuid(8)`, `udbgen(8)`

`privtext(1)`, `setucat(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nfsstart` – Starts the network file system (NFS)

**SYNOPSIS**

`/etc/nfsstart`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsstart` script starts the software necessary for the network file system (NFS) at system startup when executed by the `netstart(8)` script.

The `nfsstart` script performs the following functions:

- Starts NFS daemons (see `sdaemon(8)`). These typically include `nfsd(8)`, `mountd(8)`, and `automount(8)`.
- Initializes NFS user ID mapping by calling the `/etc/uidmaps/Set.domains` script if it exists.
- Calls the `/etc/mountnfs` script if it exists, to mount remote NFS file systems explicitly.

**SEE ALSO**

`automount(8)`, `mountd(8)`, `netstart(8)`, `nfsd(8)`, `sdaemon(8)`

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304

**NAME**

`nfsstat` – Displays NFS statistics

**SYNOPSIS**

```
/etc/nfsstat [-csnrt]
/etc/nfsstat [-mtx]
/etc/nfsstat [-a]
/etc/nfsstat [-C]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsstat` command displays statistical information about the UNICOS network file system (NFS) and Remote Procedure Call (RPC) interfaces to the kernel. You also can use it to reinitialize this information.

If you omit all options, `nfsstat` prints everything and reinitializes nothing, as if you specified the `-c`, `-n`, `-r`, and `-s` options. You must group multiple options that are specified with `nfsstat` after a single hyphen on the command line (as shown in the SYNOPSIS section).

The `nfsstat` command accepts the following options:

- `-c` Displays client information. Prints only the client-side NFS and RPC information. To print only client NFS or client RPC information, combine this option with the `-n` and `-r` options.
- `-s` Displays server information. Works like the `-c` option.
- `-n` Displays NFS information. Prints NFS information for both the client and the server. To print only client or server NFS information, combine this option with the `-c` and `-s` options.
- `-r` Displays RPC information. Works like the `-n` option.
- `-t` Displays timing information. Combine this option with any of the previous options or with the `-m` and `-x` options.
- `-m` Displays memory allocation statistics.
- `-x` Displays data transfer statistics.
- `-a` Displays all information.
- `-C` Clears all NFS statistics. Only root can use this option.

**FILES**

`/dev/kmem`      Kernel memory

**SEE ALSO**

`nfs(4P)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nfsuid` – Prints entries from a user ID map file, or a user ID map in the kernel or kernel dump

**SYNOPSIS**

```
nfsuid -u map_file user [user ...]
nfsuid -m map_name [-d dump] [-s system] user [user ...]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nfsuid` command prints, in a readable format, the specified user entries from either the specified user ID map file, or the specified kernel or kernel dump user ID map. This command is primarily useful for diagnostic purposes.

The `nfsuid` command accepts the following options and operands:

<code>-u <i>map_file</i></code>	Specifies the name of the user ID map file that contains the entries to print.
<code>-m <i>map_name</i></code>	Specifies the name of the kernel user ID map that contains the entries to print.
<code>-d <i>dump</i></code>	Reads the kernel dump file <i>dump</i> instead of <code>/dev/kmem</code> .
<code>-s <i>system</i></code>	Uses the UNICOS kernel binary file <i>system</i> instead of <code>/unicos</code> to obtain symbol information.
<code><i>user</i> [<i>user ...</i>]</code>	Specifies names of users for whom to find and print entries.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
<code>system, secadm, sysadm</code>	Allowed to use this command.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

The following command finds the user ID map entries from the kernel user ID map named `remote` for the local login names `local_user1` and `local_user2`, and it prints out the contents of each entry if it is in the map:

```
nfsuid -m remote local_user1 local_user2
```

**FILES**

/etc/uidmaps/nfsuid          Prints entries from a user ID map file

**SEE ALSO**

nfsaddhost(8), nfsaddmap(8), nfsadduser(8), nfsckhash(8), nfsclear(8), nfsgid(8),  
nfsidmap(8), nfsidmem(8), nfslist(8), nfsmerge(8), nfsrmhost(8), nfsrmmmap(8),  
nfsrmuser(8)

privtext(1) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nisaddcred` – Creates NIS+ credentials

**SYNOPSIS**

```
nisaddcred [-p principal] [-P nis_principal] [-l login_password] auth_type [domain_name]
nisaddcred -r [nis_principal]
nisaddcred -r [domain_name]
```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisaddcred` command creates security credentials for NIS+ principals. NIS+ credentials serve two purposes. The first is to provide authentication information to various services; the second is to map the authentication service name into an NIS+ principal name.

The `nisaddcred` command accepts the following options:

- `-p principal` Uses the principal name *principal* to fill the `auth_name` field for this entry. For LOCAL credentials, the name supplied with this option should be a string specifying a user ID (UID). For Data Encryption Standard (DES) credentials, the name should be a secure Remote Procedure Call (RPC) network name of the form *unix.id@domain*. If the `-p` option is not specified, the `auth_name` field is constructed from the UID of the current process and the name of the local domain.
- `-P nis_principal` Specifies the NIS+ principal name. Creates LOCAL credentials for users whose home domain is different than the local machine's default domain.  
  
When the `-P` option is not specified, `nisaddcred` constructs a principal name for the entry as follows. When it is not creating an entry of type LOCAL, `nisaddcred` calls `nis_local_principal`, which looks for an existing LOCAL entry for the effective UID of the current process in the `cred.org_dir` table and uses the associated principal name for the new entry. When creating an entry of authentication type LOCAL, `nisaddcred` constructs a default NIS+ principal name by taking the login name of the effective UID for its own process, and appending a dot (.) symbol to it followed by the local machine's default domain. If the caller is a super user, the machine name is used instead of the login name.



- l login\_password* Specifies a password to encrypt the secret key for the credential entry. This overrides the prompting for a password from the shell. This option is intended for administration scripts only. Prompting guarantees that no one can see your password on the command line by using `ps(1)`, and that you have not made any mistakes.
- The value of *login\_password* does not have to be the user's password, but it simplifies logging in if it is.
- r nis\_principal* Removes all credentials associated with the principal *nis\_principal* from the `cred.org_dir` table. If *nis\_principal* is not specified, the default is to remove credentials for the current user.
- r domain\_name* Removes a client or user from the system. If *domain\_name* is not specified, the operation is executed in the default NIS+ domain.
- When the `nisaddcred` command is run, these credentials are created and stored in a table named `cred.org_dir` in the default NIS+ domain. If *domain\_name* is specified, the entries are stored in the `cred.org_dir` table of the specified domain. The credentials of users must be stored in the same domain as their passwords.
- domain\_name* The name of a domain.

### NIS+ Principal Names

NIS+ principal names are used to specify clients that have access rights to NIS+ objects. See `nischmod(8)`, `nischown(8)`, `nis_objects(3N)`, and `nis_groups(3N)` for more information. Various other services can also implement access control based on these principal names.

The `cred.org_dir` table is organized as follows:

<i>cname</i>	<i>auth_type</i>	<i>auth_name</i>	<i>public_data</i>	<i>private_data</i>
fred.foo.com.	LOCAL	2990	10,102,44	
fred.foo.com.	DES	unix.2990@foo.com	098...819	3b8...ab2

The *cname* column in this table contains a canonical representation of the NIS+ principal name. By convention, this name is the login name of a user, or the host name of a machine, followed by a dot (.) followed by the fully qualified home domain of that principal. For users, the home domain is defined to be the domain where their DES credentials are kept. For hosts, their home domain is defined as the *domain\_name* returned by the `domainname(1)` command executed on that host.

### Auth\_type Entries

There are two types of *auth\_type* entries found in the `cred.org_dir` table: the authentication type LOCAL, and the authentication type DES. The authentication type should be specified on the command line, in uppercase or lowercase, as either `local` or `des`.

Entries of type `LOCAL` are used by the NIS+ service to determine the correspondence between fully qualified NIS+ principal names and users identified by UIDs in the domain containing the `cred.org_dir` table. This correspondence is required when associating requests made using the `AUTH_SYS` RPC authentication flavor to an NIS+ principal name. It is also required for mapping a UID in one domain to its fully qualified NIS+ principal name whose home domain may be elsewhere.

The principal's credentials for any authentication flavor are found the `cred.org_dir` table in the principal's home domain (extracted from the principal name). The same NIS+ principal may have `LOCAL` credential entries in more than one domain. Only users, and not machines, have `LOCAL` credentials. In their home domain, NIS+ users should have both `LOCAL` and `DES` types of credentials.

The `auth_name` associated with the `LOCAL` type entry is a UID that is valid for the principal in the domain containing the `cred.org_dir` table. This may differ from that in the principal's home domain.

The public information stored in the `public_data` column for a `LOCAL` entry type contains a list of group IDs (GIDs) for groups in which the user is a member. The GIDs also apply to the domain in which the table resides. There is no private data associated with this type. Neither a UID nor a principal name should appear more than once among the `LOCAL` entries in any one `cred.org_dir` table.

The `DES` authentication type is used for secure RPC authentication.

The authentication name associated with the `DES` `auth_type` is a secure RPC network name. A secure RPC network name has the form `unix.id@domain`, where `domain` must be the same as the domain of the principal. For principals that are users, the UNIX ID UID is the principal in the principal's home domain. For principals that are hosts, the UNIX ID is the host's name. In secure RPC, processes running under an effective UID of 0 (root) are identified with the host principal. Unlike `LOCAL`, there cannot be more than one `DES` credential entry for one NIS+ principal in the NIS+ namespace.

The public information in a `DES` authentication type entry is the public key for the principal. The private information in this entry is the private key of the principal encrypted by the principal's network password.

NIS+ users should have both types of credentials in their home domain. In addition, a principal must have a `LOCAL` entry in the `cred.org_dir` table of each domain from which the principal wants to make authenticated requests. An NIS+ client that makes a request from a domain in which it does not have a `LOCAL` entry will be unable to acquire `DES` credentials. An NIS+ service running at security level 2 considers such users unauthenticated and assigns them the name `nobody` for access rights.

This command is run only by NIS+ principals who are authorized to add or delete the entries in the `cred.org_dir` table.

If credentials are being added for the caller itself, `nisaddcred` automatically executes the `keylogin(1)` command for the caller.

## EXIT STATUS

The `nisaddcred` command exits with one of the following values:

Value	Description
0	Success.

1 Failure, an error occurred.

**SEE ALSO**

nischmod(8), nischown(8), nismatch(8), nistbladm(8)

chkey(1), domainname(1), keylogin(1), ps(1) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

nis\_groups(3N), nis\_local\_names(3N), nis\_objects(3N) in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nisaddent` – Creates NIS+ tables from their corresponding `/etc` files and NIS maps

**SYNOPSIS**

```

nisaddent [-a] [-P] [-r] [-v] [-D defaults] [-t table] type [nisdomain]
nisaddent [-a] [-m] [-P] [-r] [-v] [-D defaults] -f file [-t table] type [nisdomain]
nisaddent [-a] [-P] [-r] [-v] [-D defaults] [-t table] -Y ypdomain [-Y map] type [nisdomain]
nisaddent -d [-A] [-M] [-t table] type [nisdomain]

```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisaddent` command creates NIS+ tables from their corresponding `/etc` files and NIS maps. This operation is customized for each of the standard tables that are used in the administration of UNICOS systems.

The `nisaddent` command accepts the following options and operands:

- a            Adds the *file* or *map* to the NIS+ table without deleting any existing entries. This option is the default. This mode propagates only additions and modifications, not deletions.
- A            Specifies that all the data within the table and all of the data in tables in the initial table's concatenation path be returned.
- d            Dumps the NIS+ table to the standard output in the appropriate format for the given *type*.
- f *file*      Specifies that *file* should be used as the source of input instead of the standard input.
- m            Merges the *file* or *map* with the NIS+ table. This is the most efficient way to bring an NIS+ table up to date with a *file* or NIS *map* when there are only a small number of changes. This option adds entries that are not already in the database, modifies entries that already exist (if changed), and deletes any entries that are not in the source. Use the `-m` option whenever the database is large and replicated, and the *map* being loaded differs only in a few entries. This option reduces the number of update messages that have to be sent to the replicas. Also see the `-r` option.
- M            Specifies that lookups should be sent only to the master server. This guarantees that the most up-to-date information is seen, but has the disadvantage that it may make the master server busy and therefore unavailable.
- P            Follows the concatenation path. Specifies that lookups should follow the concatenation path of a table if the initial search is unsuccessful.

- r** Replaces the *file* or *map* in the existing NIS+ table by deleting any existing entries and then adding the entries from the source (*/etc* files or NIS+ maps). This option has the same effect as the **-m** option. The use of this option is strongly discouraged, unless there are a large number of changes, due to its adverse impact on performance.
- t *table*** Specifies the NIS+ table to use for this operation. The *table* argument should be a relative name as compared to either your default domain or the *domainname* if it has been specified.
- v** Specifies verbose mode.
- y *ypdomain*** Uses the dbm files for the appropriate NIS map, from the NIS domain *ypdomain*, as the source of input. The files are expected to be on the local machine in the */etc/yp/ypdomain* directory. If the machine is not an NIS server, use *ypxfr(8)* to get a copy of the dbm files for the appropriate map.
- Y *map*** Uses the dbm files for *map* as the source of input.
- D *defaults*** Specifies to use a different set of *defaults* during this operation. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. Following are descriptions of the legal tokens.
- t*tl=*time***  
Sets the default time-to-live for objects that are created by this command. The value *time* is specified in the format as defined by the *nischttl(8)* command. The default is 12 hours.
- owner=*ownername***  
Specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the principal who is executing the command.
- group=*groupname***  
Specifies that the group *groupname* should be the group owner for the object that is created. The default value is *null*.
- access=*rights***  
Specifies the set of *access* rights that are to be granted for the given object. The value of *rights* is specified in the format as defined by the *nischmod(8)* command. The default is *----rmcdr----r---*.
- type*** Specifies the type of data being processed. Legal values for *type* are one of the following aliases:
- *bootparams*
  - *ethers*
  - *group*
  - *hosts*
  - *netid*

- netmasks
- networks
- passwd
- protocols
- publickey
- rpc
- services

*nisdomain* The name of the NIS+ domain.

By default, `nisaddent` reads from the standard input file and adds this data to the associated NIS+ table by using the type specified on the command line. An alternate NIS+ table may be specified by using the `-t` option. For a key-value type table, a table specification is required.

Note that the data type can be different than the table name. For example, the `automounter` tables have key-value as the table type.

Files may be processed by using the `-f` option, and NIS version 2 (YP) maps may be processed by using the `-y` option. The `-m` option is not available when reading data from standard input.

If a *ypdomain* argument is specified, the `nisaddent` command takes its input from the `dbm` files for the appropriate NIS map `group.byname`, `passwd.byname`, or `publickey.byname`. An alternate NIS map may be specified by using the `-Y` option. If key-value is specified for the *type* operand, a map specification is required. The map must be in the `/etc/yp/ypdomain` directory on the local machine. Note that *ypdomain* is case sensitive.

If *nisdomain* is specified, `nisaddent` operates on the NIS+ table in that NIS+ domain, otherwise the default domain is used.

In terms of performance, loading up the tables is fastest when you use the `-y` option, followed by the `-f files` option and the standard input.

## ENVIRONMENT VARIABLES

Following are environment variables that affect the defaults associated with a process.

**NIS\_DEFAULTS** This variable contains a default string that overrides the NIS+ standard defaults. If the `-D` switch is used, those values override both the `NIS_DEFAULTS` variable and the standard defaults.

**NIS\_PATH** If this variable is set and neither the *nisdomain* nor the table are fully qualified, each directory specified in `NIS_PATH` is searched until the table is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nisaddent` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure, an error occurred.

**SEE ALSO**

`nischmod(8)`, `nisdefaults(8)`, `nissetup(8)`, `nistbladm(8)`, `ypxfr(8)`

`passwd(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`hosts(5)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nis_cachemgr` – Maintains a cache of location information about NIS+ servers

**SYNOPSIS**

`nis_cachemgr [-d level] [-i] [-m maxsize] [-n] [-s size] [-v]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nis_cachemgr` daemon maintains a cache of the NIS+ directory objects. It is recommended that this daemon is run on all machines that are using NIS+.

The cache contains location information necessary to contact the NIS+ servers that serve the various directories in the namespace. This includes transport addresses, information needed to authenticate the server, and a time-to-live field that indicates how long the directory object can be cached. Although the cache helps to improve the performance of the clients that are traversing the NIS+ namespace, NIS+ requests are serviced whether or not `nis_cachemgr` is running.

The cache maintained by this program is shared by all of the processes that access NIS+ on a machine. At startup, `nis_cachemgr` initializes the cache from the coldstart file and preserves unexpired entries that already exist in the cache file. See `nisinit(8)` for more information. The cache survives machine reboots because the cache is maintained in a file that is memory mapped by all the processes.

The `nis_cachemgr` program is usually started from a system start-up script.

The `nis_cachemgr` command makes NIS+ requests under the NIS+ principal name of the host on which it runs. Before running `nis_cachemgr`, security credentials for the host should be added to the `cred.org_dir` table in the host's domain by using the `nisaddcred(8)` command. Data Encryption Standard (DES)-type credentials are required if the NIS+ service is operating at security level 2. See `nisd(8)` for more information. See the WARNINGS section for more information on security and performance.

The `nisshowcache(8)` command is used to look at cached objects.

The `nis_cachemgr` command accepts the following options:

- `-d level` Turns debugging mode on. In insecure mode, the `nis_cachemgr` daemon displays information on the terminal. The amount of information depends on the level (1 to 3) selected.
- `-i` Forces `nis_cachemgr` to ignore the previous cache file and reinitialize the cache from the coldstart file. By default, the cache manager initializes itself from both the coldstart file and the old cache file, thereby maintaining the entries in the cache across machine reboots.
- `-m maxsize` Determines the maximum cache size. When the limit is reached no more information can be cached. *maxsize* specifies the number of 4-kbyte chunks. The default is 8 \* 4 kbytes.



- n Executes `nis_cachemgr` in an insecure mode. By default, before adding a directory object to the shared cache, on the request of another process on the machine, it checks the encrypted signature on the request to make sure that the directory object is a valid one and is sent by an authorized server. In insecure mode, `nis_cachemgr` adds the directory object to the shared cache without making this check.
- s *size* Determines the starting size of the cache. *size* specifies the number of 4-kbyte chunks. The default and minimum is  $2 * 4\text{Kbytes}$
- v Sets verbose mode. In this mode, the `nis_cachemgr` program logs not only errors and warnings, but also additional status messages. The additional messages are logged using `syslog(3C)` with a priority of `LOGINFO`.

## WARNINGS

If the host principal does not have the proper security credentials in the `cred.org_dir` table for its domain, running this program without the `-n` insecure mode option may significantly degrade the performance of processes issuing NIS+ requests.

## MESSAGES

The `nis_cachemgr` daemon logs error messages and warnings using `syslog(3C)`. See `syslog(3C)` for more information.

## FILES

<code>/etc/nis/NIS_SHARED_DIRCACHE</code>	The shared cache file
<code>/etc/nis/COLD_START_FILE</code>	The coldstart file
<code>/etc/init.d/rpc</code>	Initialization scripts for NIS+

## SEE ALSO

`nisaddcred(8)`, `nisd(8)`, `nisinit(8)`, `nisshowcache(8)`  
`syslog(3C)` in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080  
`nisfiles(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`niscat` – Displays NIS+ tables and objects

**SYNOPSIS**

`niscat [-A] [-h] [-L] [-M] [-v] tablename...`

`niscat [-A] [-L] [-M] [-P] -o name...`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `niscat` command displays the contents of the NIS+ tables specified by *tablename*. The command also displays the internal representation of the NIS+ objects specified by *name*, and sends it to the standard output.

The `niscat` command accepts the following options and operands:

- A Displays the data within the table and all of the data in tables when the initial table's concatenation path is returned.
- h Displays a header line prior to displaying the table. The header consists of the hash (#) symbol followed by the name of each column. The column names are separated by the table separator character.
- L Follows links. If *tablename* or *name* specifies a LINK type object, the link is followed and the object or table named by the link is displayed.
- M Sends the request to only the master server of the specified data. This guarantees that the most current information is seen. This option increases the load on the master server and increases the possibility that the NIS+ server will be unavailable for updates.
- P Specifies that the request should follow the concatenation path of a table if the initial search is unsuccessful. This option is useful only when using an indexed name for the `-o name` option.
- v Displays columns containing binary data on the standard output. Without this option, binary data is displayed as the string `*BINARY*`.
- o *name*  
Displays the internal representation of the named NIS+ objects. If *name...* is an indexed name, each of the matching entry objects is displayed. See the `nismatch(8)` man page for more information. This option is used to display access rights and other attributes of individual columns.

*tablename*

The specified NIS+ table.

**NOTES**

Columns without values in the table are displayed by two adjacent table separator characters.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH`

If this variable is set and the NIS+ table name is not fully qualified, each directory specified is searched until the table is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `niscat` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure, an error occurred.

**SEE ALSO**

`nisdefaults(8)`, `nismatch(8)`, `nistbladm(8)`

`nis_objects(3N)`, `nis_tables(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nischgrp` – Changes the group owner of an NIS+ object

**SYNOPSIS**

`nischgrp [-A] [-f] [-L] [-P] group name...`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nischgrp` command changes the group owner of the NIS+ objects or entries specified by *name* to the specified NIS+ group. Entries are specified using indexed names. See `nismatch(8)` for more information. If *group* is not a fully qualified NIS+ group name, it will be resolved using the directory search path. See `nisdefaults(8)` for more information.

The only restriction on changing an object's group owner is that you must have modify permissions for the object.

This command fails if the master NIS+ server is not running.

The `nischgrp` command accepts the following options:

- `-A`     Modifies all entries in all tables in the concatenation path that match the search criterion specified in *name*. This option requires that you also use the `-P` switch.
  - `-f`     Forces the operation and fails silently if it does not succeed.
  - `-L`     Follows links and changes the group owner of the linked object or entries rather than the group owner of the link itself.
  - `-P`     Follows the concatenation path within a named table. The *name* is either an indexed name or the `-L` switch is specified and the named object is a link pointing to entries.
- group*    The NIS+ group name.
- name*     The NIS+ object name specified for a change.

**NOTES**

The NIS+ server checks the validity of the group name before making a change.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH`   If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nischgrp` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure, an error occurred.

**SEE ALSO**

`nischmod(8)`, `nischown(8)`, `nisdefaults(8)`, `nisgrpadm(8)`, `nismatch(8)`

`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nischmod` – Changes access rights on an NIS+ object

**SYNOPSIS**

`nischmod [-A] [-f] [-L] [-P] mode name...`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nischmod` command changes the access rights (mode) of the NIS+ objects or entries specified by *name* to *mode*. Entries are specified using indexed names. See `nismatch(8)` for more information. Only principals with modify access to an object may change its mode.

The `nischmod` command accepts the following options and operands:

- A     Modifies all entries in all tables in the concatenation path that match the search criteria specified in *name*. This option requires that you also use the `-P` option.
- f     Forces the operation and fails silently if it does not succeed.
- L     Follows links and changes the permission of the linked object or entries rather than the permission of the link itself.
- P     Follows the concatenation path within a named table. The *name* is either an indexed name or the `-L` switch is specified and the specified object is a link pointing to an entry.

*mode*   The access rights of NIS+ objects.

*name*   The name of an NIS+ object.

The variable *mode* has the following form:

*rights* [,*rights*]. . .

The variable *rights* has the following form:

[*who*] *op permission* [*op permission*]. . .

The *who* variable is a combination of the following values:

- n     Nobody's permissions
- o     Owner's permissions
- g     Group's permissions
- w     World's permissions
- a     All, or owg

If *who* is omitted, the default is *a*.

The *op* variable is one of the following values:

- + To grant the permission.
- To revoke the permission.
- = To set the permissions explicitly.

The *permission* variable is any combination of the following variables:

- r Read
- m Modify
- c Create
- d Destroy

## NOTES

Unlike the system `chmod(1)` command, this command does not accept an octal notation.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisdefaults(8)` for more information.

## EXIT STATUS

The `nischmod` command exits with one of the following values:

Value	Description
0	Success.
1	Failure, an error occurred.

## SEE ALSO

`nischgrp(8)`, `nischown(8)`, `nisdefaults(8)`, `nismatch(8)`

`chmod(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

`nis(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`nischown` – Changes the owner of an NIS+ object

**SYNOPSIS**

`nischown [-A] [-f] [-L] [-P] owner name...`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nischown` command is used to change the owner of the NIS+ objects or entries specified by the operands *name* and *owner*. Entries are specified using indexed names. See `nismatch(8)` for more information. If the operand *owner* is not a fully qualified NIS+ principal name, the default domain is appended to it. See `nisaddcred(8)` and `domainname(1)` for more information.

The only restriction on changing an object's owner is that you must have modify permissions for the object. If you are the current owner of an object and you change ownership, you may not be able to regain ownership unless you have modify access to the new object.

The `nischown` command accepts the following options and operands:

- A Modifies all entries in all tables in the concatenation path that match the search criteria specified in name. It also requires the -P option.
- f Forces the operation and fails silently if it does not succeed.
- L Follows links and changes the owner of the linked object or entries rather than the owner of the link itself.
- P Follows the concatenation path within a specified table. Use this option only when either *name* is an indexedname or the -L option is specified and the specified object is a link pointing to other entries.

*owner* The new owner of an NIS+ object.

*name* The name of an NIS+ object or an entry in an NIS+ table.

**NOTES**

The NIS+ server checks the validity of the name before making the modification.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisdefaults(8)` for more information.



**EXIT STATUS**

The `nischown` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure, an error occurred.

**SEE ALSO**

`nisaddcred(8)`, `nischgrp(8)`, `nischmod(8)`, `nischttl(8)`, `nisdefaults(8)`, `nismatch(8)`  
`domainname(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011  
`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nischttl` – Changes the time-to-live value of an NIS+ object

**SYNOPSIS**

`nischttl [-A] [-f] [-L] [-P] time name...`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nischttl` command is used to change the time-to-live value (*ttl*) of the NIS+ objects or entries specified by *name* to the value specified by *time*. Entries are specified by using indexed names. See `nismatch(8)` for more information.

The time-to-live value is used by object caches to expire objects within their cache. When an object is read into the cache, this value is added to the current time in seconds, yielding an expiration time. The object may be returned from the cache until the current time is earlier than the calculated expiration time. When the expiration time is reached, the object is flushed from the cache.

The time-to-live value may be specified in seconds or in days, hours, minutes, seconds format. The latter format uses a suffix letter of *d*, *h*, *m*, or *s* to identify the units of time.

The `nischttl` command accepts the following options and operands:

- `-A`     Modifies all tables in the concatenation path that match the search criteria specified in *name*. This option implies the `-P` switch.
- `-f`     Forces the operation and fails silently if it does not succeed.
- `-L`     Follows links and change the time-to-live of the linked object or entries rather than the time-to-live of the link itself.
- `-P`     Follows the concatenation path within a named table. The *name* is either an indexed name, or the `-L` switch is specified and the named object is a link pointing to entries.
- time*    Specified time value of NIS+ object.
- name*    Specified entry by *indexedname*.

**NOTES**

Setting a high *ttl* value allows objects to stay persistent in caches for a longer period of time and can improve performance. However, when an object changes, in the worst case, the number of seconds in this attribute must pass before that change is visible to all clients. Setting a *ttl* value of 0 means that the object should not be cached at all.

A high *ttl* value is a week, a low value is less than a minute. Password entries should have *ttl* values of about 12 hours. This allows easily one password change per day. Entries in the Remote Procedure Call (RPC) table can have *ttl* values of several weeks because this information is rarely changes.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

**NIS\_PATH** If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisdefaults(8)` for more information.

## EXIT STATUS

The `nischttl` command exits with one of the following values:

Value	Description
0	Success.
1	Failure, an error occurred.

## EXAMPLES

The following example shows how to represent the *ttl* of an object by using the seconds format and the days, hours, minutes, seconds format. The second example shows that the *ttl* of the object is set to 1 day and 12 hours.

```
example% nischttl 184000 object
example% nischttl 1d12h object
```

## SEE ALSO

`nischgrp(8)`, `nischmod(8)`, `nischown(8)`, `nisdefaults(8)`, `nismatch(8)`

`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nisctl` – Controls the operation of NIS+ servers

**SYNOPSIS**

`nisctl [-M] [-v 1] [-f flag] [-H hostname] domain`

`nisctl [-M] [-v 0] [-f flag] [-H hostname] domain`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisctl` command controls the operation of NIS+ servers. It is used to turn debugging on and off, to flush caches, and to print statistics.

The `nisctl` command accepts the following options:

- `-M`      Modifies the data segment space allocation for the NIS+ daemon `nisd` that resides on all servers that serve the local domain if no domain or server is specified. See the `-H` option. Modifies the data segment space allocation for the NIS+ daemon `nisd` on all servers that serve a specified domain if no server is specified. Modifies the data segment space allocation for the NIS+ daemon `nisd` that resides on a specific server that serves a specified domain if the `-H` option is used.
- `-v`      Turns debugging on (1) or off(0) on all the servers or just a specific server that serve(s) the local or a specified domain.
- `-f flag` Causes NIS+ servers to release cached information about the namespace. The `-f` option accepts the following *flag* arguments:
  - `-o`      Flushes all cached information about NIS+ object.
  - `-g`      Flushes all cached information about NIS+ groups.
  - `-d`      Flushes all cached information about NIS+ directory.
- `-H hostname`  
Specifies a server in an NIS+ domain.
- domain*   Specifies a particular domain in the namespace. If no domain is specified, the local domain is selected.

**EXIT STATUS**

The `nisctl` command exits with one of the following values:

Value	Description
0	Success.
1	Failure, an error occurred.

**NISCTL(8)**

**NISCTL(8)**

**SEE ALSO**

nisd(8)

**NAME**

`nisd` – NIS+ service daemons

**SYNOPSIS**

```
nisd [-A] [-C] [-D] [-F] [-f] [-h] [-r] [-v] [-Y] [-c seconds] [-d dictionary] [-L load]
[-S level]
```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisd` daemon is an Open Network Computing (ONC) Remote Procedure Call (RPC) service that implements the NIS+ service. This daemon must be running on all machines that serve a portion of the NIS+ namespace.

The `nisd` daemon is usually started from a system startup script.

The `nisd` daemon accepts the following options:

- A        Logs all the authentication related activities in verbose mode to `syslogd(8)` with `LOGINFO` priority.
- C        Opens a log file that contains all the messages displayed on the console.
- D        Turns debugging mode on.
- F        Forces the server to do a checkpoint of the database when it starts. Forced checkpoints may be required when the server is low on disk space. This option removes updates from the transaction log that have propagated to all of the replicas.
- f        Forces registration even if the program number is already in use.
- h        Prints all the `nisd` options.
- r        Tells the daemon that it is running as a root server. The primary difference between the root server and a nonroot server is that root servers are members of the same domain they serve, whereas nonroot servers are always members of a domain above the domain they serve.
- v        Sends a running verbose narration of what the daemon is doing to the `syslog` daemon with `LOGINFO` priority. See `syslogd(8)` for more information. This option is most useful for debugging problems with the service. (See also the `-A` option.)
- Y        Puts a server into NIS (YP) compatibility mode. When operating in this mode, the NIS+ server responds to NIS version 2 requests using the NIS version 2 protocol. Because the YP protocol is not authenticated, only those items that have read access to nobody (an unauthenticated request) are visible through the version 2 protocol. The server supports only the standard NIS version 2 maps in this mode.

- c *seconds* Sets the number of seconds between sending out updates to the replicas. By default this is set to 120 seconds or 2 minutes.
- d *dictionary* Specifies an alternate dictionary for the NIS+ database. The primary use of this option is for testing. The string is not interpreted, it is simply passed to the `db_initialize` function. See `nis_db(3N)` for more information.
- L *load* Specifies the load that the NIS+ service is allowed to place on the server. The load is specified in terms of the number of child processes that the server may spawn. The number of child processes must be at least 1 for the callback functions to work correctly. The default is 128.
- S *level* Sets the authentication security level of the service. By default, the daemon runs at security level 2.  
 The value of *level* is a number between 0 and 2:
  - 0 Security level 0 is designed to be used for testing and the initial setup of an NIS+ namespace. When running at level 0, the daemon does not enforce any access controls. Any client is allowed to perform any operation.
  - 1 At security level 1, the daemon accepts both `AUTH_SYS` and `AUTH_DES` credentials for authenticating clients and authorizing them to perform NIS+ operations. This is not a secure mode of operation since `AUTH_SYS` credentials are easily forged. `AUTH_SYS` authentication is not recommended on networks where there is the potential for untrusted user access.
  - 2 At security level 2, the daemon accepts only `AUTH_DES` credentials for authentication and authorization. This is the highest level of security currently provided by the NIS+ service. This is the default security level if the `-S` option is not used.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

`NETPATH` The transports that the NIS+ service uses can be limited by setting this environment variable. See `netconfig(4)` for more information.

## FILES

`/etc/nis/parent.object`

This file contains an external Data Representation (XDR)-encoded NIS+ object that describes the namespace above a root server. This parent namespace may be another NIS+ namespace or a foreign namespace such as one served by the domain name service. It is present only on servers that are serving the root of the namespace.

`/etc/nis/root.object`

This file contains an XDR-encoded NIS+ object that describes the root of the namespace. It is present only on servers that are serving the root of the namespace.

`/etc/nis/COLD_START_FILE`

This file contains the necessary information for the NIS+ client libraries to locate a server for the default domain.

`/etc/nis/NIS_SHARED_DIRCACHE`

This file contains a set of cached objects describing servers that serve other portions of the NIS+ namespace. It is created and maintained by the `nis_cachemgr(8)`.

`/etc/init.d/rpc`

This file is an initialization script for NIS+.

`/etc/nisl.log`

This file combines all the messages displayed on the console if the `-C` option is used.

## SEE ALSO

`nis_cachemgr(8)`, `nisinit(8)`, `nissetup(8)`, `syslogd(8)`

`nis_db(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

`netconfig(4)`, `nisfiles(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014



**NAME**

`nisdefaults` – Displays NIS+ default values

**SYNOPSIS**

`nisdefaults [-a] [-d] [-g] [-h] [-p] [-r] [-s] [-t] [-v]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisdefaults` command prints the default values that are returned by calls to the NIS+ local name functions. See `nis_local_names(3N)` for more information. With no options specified, all defaults are printed in verbose mode. When options are specified, options are displayed only in a terse mode suitable for shell scripts.

The `nisdefaults` command accepts the following options:

- a Prints all defaults in a terse format.
- d Prints the default domain name.
- g Prints the default group name.
- h Prints the default host name.
- p Prints the default principal name.
- r Prints the default access rights.
- s Prints the default directory search path.
- t Prints the default time-to-live value.
- v Prints the defaults in a verbose format. This prepends an identifying string to the output.

**ENVIRONMENT VARIABLES**

The following environment variables affect the defaults associated with a process.

**NIS\_DEFAULTS** This variable contains a defaults string that overrides the NIS+ standard defaults. The defaults string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. Following are descriptions of all the legal tokens.

`tttl=time`

Sets the default time-to-live for objects that are created. The value *time* is specified in the format as defined by the `nischttl(8)` command. The default value is 12 hours.

*owner=ownername*  
 Specifies that the NIS+ principal *ownername* should own created objects. The default for this value is the principal who is executing the command.

*group=groupname*  
 Specifies that the group *groupname* should be the group owner for created objects. The default is NULL.

*access=rights*  
 Specifies the set of access rights that are to be granted for created objects. The value of *rights* is specified in the format as defined by the `nischmod(8)` command. The default value is `----rmdir--r---`.

NIS\_GROUP This variable contains the name of the local NIS+ group. If the name is not fully qualified, the default domain is appended to it.

NIS\_PATH This variable overrides the default NIS+ directory search path. It contains an ordered list of directories separated by colon (:) symbols. The dollar sign (\$) symbol is treated specially. Directory names that end in the dollar sign have the default domain appended to them, and a dollar sign by itself is replaced by the list of directories between the default domain and the global root that are at least two levels deep. The default NIS+ directory search path is \$.

## SEE ALSO

`nischmod(8)`, `nischttl(8)`

`nis_local_names(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`niserror` – Displays NIS+ error messages

**SYNOPSIS**

`niserror error-num`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `niserror` command displays the NIS+ error message associated with the status value *error-num* in standard output. It is used to translate NIS+ error numbers that are returned into text messages for shell scripts.

The `niserror` command accepts the following operand:

*error-num* Specifies the status number attached to an error message.

**EXIT STATUS**

The `niserror` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure, an error occurred.

**SEE ALSO**

`nis_error(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nisgrpadm` – Administers NIS+ groups

**SYNOPSIS**

```

nisgrpadm -a [-s] group principal...
nisgrpadm -r [-s] group principal...
nisgrpadm -t [-s] group principal...
nisgrpadm [-s] group principal...
nisgrpadm -c [-M] [-s] group
nisgrpadm -d [-M] [-s] group
nisgrpadm -l [-M] [-s] group
nisgrpadm [-M] [-s] group

```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisgrpadm` command administers both NIS+ groups and group membership lists. The `nisgrpadm` command is used to create, destroy, or list NIS+ groups. The `nisgrpadm` command is also used to administer a group's membership list. It can add or delete principals to the group, or test principals for membership in the group.

The names of NIS+ groups are syntactically similar to names of NIS+ objects, but they occupy a separate namespace. A group named `a.b.c.d.` is represented by an NIS+ group object named `a.groups_dir.b.c.d.`; the functions described here all expect the name of the group, not the name of the corresponding group object. There are three types of group members:

- An explicit member is just an NIS+ principal name, such as `wickedwitch.west.oz.`
- An implicit domain member, such as `*.west.oz.`, means that all principals in the given domain belong to this member. No other forms of wildcards are allowed. Both `wickedwitch.*.oz.` and `wickedwitch.west.*.` are not valid. The principals in subdomains of the given domain are not included.
- A recursive group member, such as `@cowards.oz.`, refers to another group. All principals that belong to that group are considered to belong here.

Any type of group member may be made negative by prefixing it with a minus sign (-). A group may thus contain explicit, implicit, recursive, negative explicit, negative implicit, and negative recursive members.

A principal is considered to belong to a group if it belongs to at least one nonnegative group member of the group and belongs to no negative group members.

The `nisgrpadm` command accepts the following options and operands:

- a Adds the list of NIS+ principals specified to *group*. The *principal* name must be fully qualified.
- c Creates a group in the NIS+ namespace. The NIS+ group name must be fully qualified.
- d Destroys (removes) a group from the namespace.
- l Lists the membership list of the specified group. (See the `-M` option.)
- M Master server only. Sends the lookup to only the master server of the specified data. This guarantees that the most up-to-date information is seen, but has the disadvantage that it may make the master server busy and therefore unavailable. The `-M` option is applicable only when used with with the `-l` option.
- r Removes the list of principals specified from *group*. The principal name must be fully qualified.
- s Returns using the exit status of the command silently. This status can be translated into a text string by using the `niserror(8)` command.
- t Displays whether the principals specified are members in *group*.
- group* Specifies the group added or removed.
- principal* Specifies the principal added or removed.

## NOTES

Principal names must be fully qualified, but group names can be abbreviated on all operations except `create`.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ group name is not fully qualified, each directory specified is searched until the group is found. See `nisdefaults(8)` for more information.

## MESSAGES

- `NIS_SUCCESS` The `nisgrpadm` command returns 0 on success.
- `NIS_PERMISSION` You do not have the needed access right to change the group.
- `NIS_NOTFOUND` The specified group does not exist.
- `NIS_TRYAGAIN` The server for the group's domain is currently checkpointing or in a read-only state. Retry the command at a later date.
- `NIS_MODERROR` The group was modified by someone else during the execution of the command. Reissue the command and, optionally, check the group's membership list again.

**SEE ALSO**

nischgrp(8), nisdefaults(8)

nis\_error(3N), nis\_groups(3N) in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

nis(4) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nisinit` – Initializes an NIS+ client and server

**SYNOPSIS**

```

nisinit -r
nisinit -p Y parent_domain host...
nisinit -p D parent_domain host...
nisinit -p N parent_domain host...
nisinit -c -C coldstart
nisinit -c -H hostname
nisinit -c -B

```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisinit` command initializes a machine to be an NIS+ client or an NIS+ root server.

The `nisinit` command accepts the following options and operands:

- r Initializes a machine as an NIS+ root server. This option creates the file `/etc/nis/root.object` and initializes it to contain information about this machine.
- p Initializes an `/etc/nis/parent.object` file to make a domain part of the namespace above it. Only root servers can have parent objects. A parent object describes the namespace above the NIS+ root. If this is an isolated domain, this option should not be used. The *argument* to this option tells the command what type of name server is serving the domain above the NIS+ domain. This option takes a Y, D, or N argument. When clients attempt to resolve a name that is outside of the NIS+ namespace, this object is returned with the error `NIS_FOREIGNNNS`, indicating that a namespace boundary has been reached. It is up to the client to continue the name resolution process.

The parameter *parent\_domain* is the name of the parent domain in a syntax that is native to that type of domain. The list of host names that follow the domain parameter are the names of hosts that serve the parent domain. If there is more than one server for a parent domain, the first *host...* specified should be the master server for that domain.

- Y Specifies that the parent directory is an NIS version 2 domain.
- D Specifies that the parent directory is a Domain Naming Service (DNS).
- N Specifies that the parent directory is another NIS+ domain. This option is useful for connecting a pre-existing NIS+ subtree into the global namespace.

Currently, the NIS+ clients do not use the `-p` option. This parent object is currently not replicated on root replica servers.

- c Initializes the machine to be an NIS+ client. The three initialization options available are *broadcast*, *coldstart*, and *hostname*. The most secure option is to initialize a client from a trusted *coldstart* file. The second option is to initialize a client using a *hostname* that you specify as a trusted host. The third option is to initialize a client by *broadcast*. The third option is the least secure.

Using the `-c` option enables only the navigation of the NIS+ namespace from a specific client. To make NIS+ your name service, modify the `/etc/nsswitch.conf` file. See `nsswitch(4)` for more information.

- C *coldstart*

Uses the file *coldstart* as a prototype *coldstart* file when initializing an NIS+ client. This *coldstart* file may be copied from a machine that is already a client of the NIS+ namespace. For maximum security, an administrator can encrypt and encode the *coldstart* file and mail it to an administrator who is starting up a new machine. The new administrator can decode, decrypt, and use this file with the `nisinit` command to initialize the machine as an NIS+ client. If the *coldstart* file is from another client in the same domain, the `nisinit` command may be safely skipped and the file copied into the `/etc/nis` directory as `/etc/nis/COLD_START_FILE`.

- H *hostname*

Specifies that the host *hostname* should be contacted as a trusted NIS+ server. The `nisinit` command iterates over each transport in the `NETPATH` environment variable and attempts to contact `rpcbind(8)` on that machine. This *hostname* must be reachable from the client without the name service running. For Internal Protocol (IP) networks there must be an entry in `/etc/hosts` for the specified host when `nisinit` is invoked.

- B Specifies that the `nisinit` command should use an IP broadcast to locate an NIS+ server on the local subnetwork. Any machine that is running the NIS+ service may answer. No guarantees are made that the server that answers is a server of the organization's namespace. If this option is used, we recommend checking with your system administrator if the server and domain served are valid. The binding information can be dumped to standard output using the `nisshowcache(8)` command.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

**NETPATH** This environment variable may be set to the transports to try when contacting the NIS+ server. See `netconfig(4)` for more information.

## EXIT STATUS

The `nisinit` command exits with one of the following values:

Value	Description
0	Success.



- 1 Failure, an error occurred.

## FILES

`/etc/nis/COLD_START_FILE`

This file contains a list of servers, their transport addresses, and their secure RPC public keys that serve the machine's default domain.

`/etc/nis/hostname/root.object`

This file describes the root object of the NIS+ namespace. It is a standard XDR-encoded NIS+ directory object that can be modified by authorized clients using the `nis_modify(3N)` interface.

`/etc/nis/hostname/parent.object`

This file describes the namespace that is logically above the NIS+ namespace. The most common type of parent object is a DNS object. This object contains contact information for a server of that domain.

## SEE ALSO

`rpcbind(8)`, `nisshowcache(8)`

`uuencode(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`netconfig(4)`, `nisfiles(4)`, `nsswitch(4)`, `hosts(5)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nisl`n - Symbolically links NIS+ objects

**SYNOPSIS**

`nisl`n [-L] [-D *defaults*] *name linkname*

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisl`n command links an NIS+ object specified by *name* to an NIS+ name specified by *linkname*. If *name* is an indexed name, the link points to entries within an NIS+ table. See `nismatch(8)` for more information. Clients who want to look up information in the name service can use the `FOLLOW_LINKS` flag to force the client library to follow links to the name they point to. All NIS+ administration commands accept the `-L` switch, indicating that they should follow links. See `nisl_names(3N)` for a description of the `FOLLOW_LINKS` flag.

The `nisl`n command accepts the following options and operands:

- L            Specifies that this command should follow links. If *name* is itself a link, then this command will follow it to the linked object that it points to. The new link will point to that linked object rather than to *name*.
- D *defaults* Specifies to use a different set of defaults for the creation of the link object. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. The legal tokens are as follows:
  - `t``t``l``=`*time*  
Sets the default time-to-live for objects that are created by this command. The value *time* is specified in the format as defined by the `nischttl(8)` command. The default is 12 hours.
  - `o``w``n``e``r``=`*ownername*  
Specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the principal who is executing the command.
  - `g``r``o``u``p``=`*groupname*  
Specifies that the group *groupname* should be the group owner for the object that is created. The default value is NULL.
  - `a``c``c``e``s``s``=`*rights*  
Specifies the set of access rights that are to be granted for the given object. The value of *rights* is specified in the format as defined by the `nischmod(8)` command. The default value is `----r``m``c``d``r``--r``---`.
- name*        Specifies the name of an NIS+ object.

*linkname* A specified name that identifies a link between an NIS+ object and an NIS+ name.

## NOTES

When creating the link, `nisl` verifies that the linked object exists. Once created, the linked object may be deleted or replaced and the link will not be affected. If the linked object is deleted, the link is not valid and attempts to return the error message `NIS_LINKNAMEERROR` to the client. When the path attribute in tables specifies a link rather than another table, the link is followed if the flag `FOLLOW_LINKS` was present in the call to the `nisl` function. If the flag is not present, the link is ignored. See `nisl_tables(3N)` for more information. If the flag is present and the link is no longer valid, a warning is sent to the system logger and the link is ignored.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisl_defaults(8)` for more information.

## EXIT STATUS

The `nisl` command exits with one of the following values:

Value	Description
0	Success.
1	Failure, an error occurred.

## SEE ALSO

`nischmod(8)`, `nischttl(8)`, `nisl_defaults(8)`, `nismatch(8)`, `nism(8)`, `nislbladm(8)`  
`nisl_names(3N)`, `nisl_tables(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nislog` – Displays the contents of the NIS+ transaction log

**SYNOPSIS**

`nislog [-h num] [-v] [directory...]`

`nislog [-t num] [-v] [directory...]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nislog` command displays the contents of the NIS+ server transaction log to standard output. This command is used to track changes in the namespace. The `/etc/nis/hostname.log` file contains the transaction log maintained by the NIS+ server. The `hostname` variable is a string returned by `uname -n`. When updates occur, they are logged to this file and then propagated to replicas as log transactions. When the log is checkpointed, updates that have been propagated to the replicas are removed. The `nislog` command must be executed by root on an NIS+ server. It displays the log entries for that server only. If `directory` is not specified, the entire log is searched. If `directory` is specified, only log entries that correspond to the specified directories are displayed.

The `nislog` command accepts the following options and operands:

- `-h [num]` Displays *num* transactions from the "head" of the log. If the numeric parameter is omitted, it is assumed to be 1. If the numeric parameter is 0, only the log header is displayed.
- `-t [num]` Displays *num* transactions from the "tail" of the log. If the numeric parameter is omitted, it is assumed to be 1. If the numeric parameter is 0, only the log header is displayed.
- `-v` Specifies verbose mode.
- `directory` Specifies the log for a specified directory.

**FILES**

`/etc/nis/hostname.log` transaction log

**SEE ALSO**

`nisd(8)`

`uname(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR–2011

`nis(4)`, `nisfiles(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR–2014

**NAME**

`nisls` – Lists the contents of an NIS+ directory

**SYNOPSIS**

`nisls [-d] [-g] [-l] [-L] [-m] [-M] [-R] [name...]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisls` command lists the contents of the NIS+ directory that is specified by *name*. If *name* specifies an NIS+ object other than a directory, `nisls` simply echoes the name. If no *name* is specified, the first directory in the search path is listed. See `nisdefaults(8)` for more information.

The `nisls` command accepts the following options and operand:

- `-d` Treats NIS+ directories like other NIS+ objects, rather than listing their contents.
  - `-g` Displays the group owner instead of the owner when listing in long format.
  - `-l` Displays additional information in long format about the objects such as their type, creation time, owner, and access rights. The access rights are listed in the following order in long mode: nobody, owner, group owner, and world.
  - `-L` Specifies that links are to be followed. If the *name* variable points to a link, the link is followed to the linked object.
  - `-m` Displays modification time instead of creation time when listing in long format.
  - `-M` Returns information from only the master server of the specified object. This option guarantees that the most up-to-date information is seen, but has the disadvantage that it may make the master server busy and therefore unavailable.
  - `-R` Lists directories recursively by repeating the list for each subdirectory found in the process of listing each name.
- name* Specifies the name of the NIS+ directory you want to list.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nisls` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure, an error occurred.

**SEE ALSO**

`nisdefaults(8)`, `nisgrpadm(8)`, `nismatch(8)`, `nistbladm(8)`

`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nismatch`, `nisgrep` – Searches NIS+ tables

**SYNOPSIS**

```
nismatch [-A] [-c] [-h] [-M] [-o] [-P] [-v] key tablename
nismatch [-A] [-c] [-h] [-M] [-o] [-v] colname=key... tablename
nismatch [-A] [-c] [-h] [-M] [-o] [-P] [-v] indexedname
nisgrep [-A] [-c] [-h] [-M] [-o] [-v] key tablename
nisgrep [-A] [-c] [-h] [-M] [-o] [-v]
```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nismatch` and `nisgrep` commands are used to search NIS+ tables. The `nisgrep` command differs from the `nismatch` command by the ability to accept regular *keypat* expressions for the search criteria rather than simple text matches.

For both commands, the parameter *tablename* is the NIS+ name of the table to be searched. If only one *key* or *keypat* is specified, then it is applied and the first column is searched. Specific named columns can be searched by using the *colname=* syntax. When multiple columns are searched, only entries that match in all columns are returned. This is the equivalent of a logical join operation.

Because `nisgrep` uses a callback function, it is not constrained to searching only those columns that are specifically made searchable at the time of table creation. Thus, it is more flexible, but slower, than `nismatch`.

The `nismatch` command accepts an additional form of search criteria, *indexedname*, which is an NIS+ indexed name of the form: [*colname=value,...*],*tablename*

The `nismatch` and `nisgrep` commands accept the following options and operands:

- A Returns all the data within a table and all of the data in tables in the initial table's concatenation path.
- c Prints only a count of the number of entries that matched the search criteria.
- h Displays a header line before the matching entries that contains the names of the table's columns.
- M Sends the lookup to only the master server of the specified data. This guarantees that the most up-to-date information is seen, but has the disadvantage that the master server may be busy.
- o Displays the internal representation of the matching NIS+ objects.

- P Specifies that the lookup should follow the concatenation path of a table if the initial search is unsuccessful. This option is used only with the `nismatch` command.
- v Specifies verbose mode.
- key* Pattern of key strokes.
- tablename* Name of a specified table.
- colname=key* Specified column and key pattern to be searched.
- indexedname* Indexed name of the column name and table name.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

### NIS\_PATH

If this variable is set and the NIS+ table name is not fully qualified, each directory specified is searched until the table is found. See `nisdefaults(8)` for more information.

## EXIT STATUS

The `nismatch` command exits with one of the following values:

Value	Description
0	Returns 0 when it successfully matches entries.
1	Returns 1 when it successfully searches a table and no matches are found.
2	Returns 2 when an error condition occurs. An error message is also printed.

## MESSAGES

No memory	An attempt to allocate some memory for the search failed.
<i>tablename</i> is not a table	The object with the name <i>tablename</i> was not a table object.
Can't compile regular expression	The regular expression in <i>keypat</i> was malformed.
column not found: <i>colname</i>	The column named <i>colname</i> does not exist in the table named <i>tablename</i> .

## SEE ALSO

`niscat(8)`, `nisdefaults(8)`, `nisls(8)`, `nistbladm(8)`

`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169



**NAME**

nismkdir – Creates NIS+ subdirectories

**SYNOPSIS**

nismkdir [-D *defaults*] [-m *hostname*] *dirname*

nismkdir [-D *defaults*] [-s *hostname*] *dirname*

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nismkdir` command creates new NIS+ subdirectories within an existing domain. It has the option of creating replicated or nonreplicated directories.

The two primary aspects that are controlled when making a directory are its access rights and its degree of replication. Without options, this command will create a subdirectory with the same master and replicas as the parent directory.

The `nismkdir` command accepts the following options, tokens, and operands:

- D *defaults* Specifies to use a different set of defaults when creating new directories. The *defaults* string is a series of tokens separated by colons. These tokens represent the default values that will be used for the generic object properties. All of the legal tokens are as follows:
  - `ttl=time`  
Sets the default time-to-live for objects that are created by this command. The value of *time* is specified in the format as defined by the `nischttl(8)` command. The default value is 12 hours.
  - `owner=ownername`  
Specifies that the NIS+ principal *ownername* should own the created object. The default for this value is the principal who is executing the command.
  - `group=groupname`  
Specifies that the group *groupname* should be the group owner for the object that is created. The default value is NULL.
  - `access=rights`  
Specifies the set of access rights that are to be granted for the given object. The value of *rights* is specified in the format as defined by the `nischmod(8)` command. The default value is `----rmcdr--r---`.
- m *hostname* If the directory specified by *dirname* does not exist, then a new directory that is not replicated is created with host *hostname* as its master server. If the directory name specified by *dirname* does exist, then the host specified by *hostname* is made its master server.

*-s hostname* Specifies that the host *hostname* will be a replica for the existing directory that is specified by *dirname*.

*dirname* Specifies the fully qualified NIS+ name of the directory that will be created.

**NOTES**

A host that serves an NIS+ directory must be an NIS+ client in a directory above the one it is serving. Root NIS+ servers, which are both clients and servers of the same NIS+ directory, are exceptions to this rule.

When the host's default domain is different from the default domain on the client where the command is executed, the hostname supplied as an argument to the *-s* or *-m* options must be fully qualified.

**EXIT STATUS**

The `nismkdir` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure; an error occurred.

**ENVIRONMENT VARIABLES**

Following are environment variables that affect the defaults associated with a process.

`NIS_DEFAULTS` This variable contains a defaults string that overrides the NIS+ standard defaults. If the *-D* option is used those values override both the `NIS_DEFAULTS` variable and the standard defaults.

`NIS_PATH` If this variable is set and the NIS+ directory name is not fully qualified, each directory specified is searched until the directory is found. See `nisdefaults(8)` for more information.

**SEE ALSO**

`nischmod(8)`, `nischttl(8)`, `nisdefaults(8)`, `nisls(8)`, `nisrmdir(8)`

**NAME**

`nispaswd` – Changes NIS+ password information

**SYNOPSIS**

```
nispaswd [-g] [-h] [-s] [-D domainname] [username]
nispaswd -a
nispaswd [-D domainname] [-d [username]]
nispaswd [-l] [-f] [-n min] [-x max] [-w warn] [-D domainname] username
```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nispaswd` command changes a password, `gecos` field, home directory, or login shell associated with a user name in the NIS+ `passwd` table.

The command is also used to view or modify aging information associated with the user specified if the invoker has the right NIS+ privileges.

The `nispaswd` command accepts the following options:

- g       Changes the `gecos` field (`finger(1)`) information.
- h       Changes the home directory.
- s       Changes the login shell.
- D *domainname*  
      Consults the `passwd.org_dir` table in *domainname*. If this option is not specified, the default *domainname* returned by `nis_local_directory()` is used. This *domainname* is the same as that returned by `domainname(1)`.
- a       Shows the password attributes for all entries. This will show only the entries in the NIS+ `passwd` table in the local domain that the invoker is authorized to read.
- d *username*  
      Displays password attributes for the caller or the user specified if the invoker has the right privileges.
- l       Locks the password entry for the user specified by *username*. Subsequent attempts to log in by using `login(1)` and this NIS+ password entry would not be allowed.
- f       Forces users to change their password at the next login by expiring the password for *username*.

- n *min*** Sets the minimum field for *username*. The *min* field contains the minimum number of days between password changes for *username*. If *min* is greater than *max*, the user may not change the password. Always use this option with the **-x** option, unless *max* is set to **-1** (aging turned off). In that case, *min* need not be set.
- x *max*** Sets the maximum field for *username*. The *max* field contains the number of days that the password is valid for *username*. The aging for *username* will be turned off immediately if *max* is set to **-1**. If it is set to 0, users are forced to change their password at the next login session, and aging is turned off.
- w *warn*** Sets the warning field for *username*. The warning field contains the number of days before the password expires that users will be warned whenever they attempt to login.

Users may also use the **-n**, **-x**, and **-w** options to set or modify the minimum, maximum and warning attributes for their password information.

Users may also use the **-d** option to display password attributes for their login name. The format of the display is as follows:

```
username status mm/dd/yy min max warn
```

If password aging information is not present the format is as follows:

```
username status
```

The *username* specifies the login ID of the user.

The following variables are displayed:

<b>Variable</b>	<b>Description</b>
<i>status</i>	Displays the password status of <i>username</i> . PS stands for password exists or locked, LK stands for locked, and NP stands for no password.
<i>mm/dd/yy</i>	Displays the date the password was last changed for <i>username</i> . All password aging dates are determined using Greenwich Mean Time and, therefore, may differ by as much as a day in other time zones.
<i>min</i>	Displays the minimum number of days required between password changes for <i>username</i> .
<i>max</i>	Displays the maximum number of days the password is valid for <i>username</i> .
<i>warn</i>	Displays the number of days relative to <i>max</i> before the password expires and the user will be warned.

The `nispasswd` command uses Secure RPC to communicate with the NIS+ server, and never sends unencrypted passwords over the communication medium.

The `nispasswd` command does not read or modify the local password information stored in the `/etc/passwd` file.

When used to change a password, `nispasswd` prompts nonprivileged users for their old password. It then prompts for the new password twice to forestall typing mistakes. When the old password is entered, `nispasswd` checks to see if it has aged sufficiently. If aging is insufficient, `nispasswd` terminates.

The old password is used to decrypt the `username` secret key. If the password does not decrypt the secret key, `nispasswd` prompts for the old secure-RPC password. It uses this password to decrypt the secret key. If this fails, it gives the user one more chance, and if the user input fails again, `nispasswd` terminates. The old password also is used to ensure that the new password differs from the old by at least 3 characters.

Assuming that password aging is sufficient, a check is made to ensure that the new password meets specific construction requirements as described in the following. When the new password is entered a second time, the two copies of the new password are compared. If the two copies are not identical, the cycle of prompting for the new password is repeated twice. The new password is used to re-encrypt the user's secret key. Hence, it also becomes the user's secure-RPC password.

Passwords must meet the following requirements:

- Each password must have at least 6 characters. Only the first 8 characters are significant.
- Each password must contain at least 2 alphabetic characters and at least 1 numeric or special character. Alphabetic characters include all uppercase and lowercase letters.
- Each password must differ from the user's login `username` and any reverse or circular shift of that login `username`. For comparison purposes, an upper case letter and its corresponding lowercase letter are equivalent.
- New passwords must differ from the old by at least 3 characters. For comparison purposes, an uppercase letter and its corresponding lowercase letter are equivalent.

Network administrators own the NIS+ password table. They may change any password attributes if they establish their credentials before invoking `nispasswd`. The `nispasswd` command does not prompt privileged users for the old password and they are not forced to comply with password aging and password construction requirements. See `keylogin(1)` for more information.

## NOTES

The login program, file access display programs such as `ls -l` and network programs such as `rlogin(1B)` and `ftp(1B)` that require user passwords use the standard `getpwent(3C)` interface to get password information. These programs get the NIS+ password information, which is modified by `nispasswd`, only if the `passwd:` entry in the `/etc/nsswitch.conf` file refers to `nisplus`. See `nsswitch(4)` for more information.

## EXIT STATUS

The `nispasswd` command exits with one of the following values and messages:

Value	Messages
0	SUCCESS.
1	Permission denied.

- 2 Invalid combination of options.
- 3 Unexpected failure. NIS+ passwd table unchanged.
- 4 NIS+ passwd table missing.
- 5 NIS+ is busy. Try again later.
- 6 Invalid argument to option.
- 7 Aging is disabled.

**SEE ALSO**

nistbladm(8)

domainname(1), finger(1B), ftp(1B), keylogin(1), login(1), passwd(1), rlogin(1B) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

getpwent(3C) in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

nsswitch(4) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nispath` – Prints the search path of a specified NIS+ name

**SYNOPSIS**

`nispath [-v] name`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nispath` command displays the list of candidate names for the name specified in *name*. If this name is not fully qualified, `nispath` generates a list of names using the default NIS+ directory search path, or the environment variable `NIS_PATH` if it is set.

The `nispath` command accepts the following options:

`-v` Displays a more detailed message about the search path.

*name* The specified name for your search.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. This variable contains an ordered list of directories separated by colon ( : ) symbols.

**NAME**

`nisping` – Sends a ping to NIS+ servers

**SYNOPSIS**

```
nisping [-u] [-f] [-H hostname] [-r]
nisping [-u] [-f] [-H hostname] [directory]
nisping -C [-H hostname] [directory]
```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisping` command sends a ping to all replicas of an NIS+ directory. Once a replica receives a ping, it will check with the master server for the directory to get updates. Prior to pinging the replicas, this command attempts to determine the last update received by a replica and the last update logged by the master. If these two time stamps are the same, the ping is not sent. The `-f` option overrides this feature.

Usually, NIS+ replica servers get new information from the master NIS+ server within a short time. Therefore, there should not be any need to use the `nisping` command.

The `nisping` command with the `-C` option is used to send a checkpoint request to the servers. Upon receiving a checkpoint request, the servers commit all the updates from the table log files to the database files. If they are out of date, updates to replicas are sent. This option is necessary because the database log files for NIS+ are not automatically checkpointed. `nisping` should be used at frequent intervals to checkpoint the NIS+ database log files. It is recommended that the `nisping` command with the `-C` option be used at least once a day. This command may be added to a `crontab(1)` file. If the database log files are not checkpointed, their sizes will continue to grow.

The `nisping` command accepts the following options and operands:

- `-u` Displays the time of the last update; no servers are sent a ping.
- `-f` Forces a ping, even though the time stamps indicate there is no reason to do so. This option is useful for debugging.
- `-H hostname` Sends the ping, checks for an update time, or checkpoints only the host *hostname*.
- `-r` Retrieves status about the root object from the root servers, especially when new root replicas are added or deleted from the list. If used without the `-u` option, the `-r` option is used to send a ping request to the servers serving the root domain. When the replicas receive a ping, they update their root object if needed. The `-r` option can be used with all other options except with the `-C` option; the root object does not need be checkpointed.
- directory* Specifies the directory that is checkpointed or sent a ping.



- C Sends a request to checkpoint, rather than a ping, to each server. The servers schedule to commit all the transactions to stable storage.

## NOTES

If the server specified by the -H option does not serve the directory, then no ping is sent.

## ENVIRONMENT VARIABLES

Following is an environment variable that affects the defaults associated with a process.

**NIS\_PATH** If this variable is set and the NIS+ directory name is not fully qualified, each directory specified is searched until the directory is found.

## EXIT STATUS

The `nisping` command exits with one of the following values:

Value	Description
-1	No servers were contacted, or the server specified by the -H switch could not be contacted.
0	Success.
1	Some, but not all, servers were successfully contacted.

## SEE ALSO

`nislog(8)`

`crontab(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`nis_admin(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

`nisfiles(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nisrm` – Removes NIS+ objects from the namespace

**SYNOPSIS**

`nisrm [-i] name...`

`nisrm [-f] name...`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisrm` command removes specified NIS+ objects from the NIS+ namespace.

The `nisrm` command accepts the following options and operands:

- i Interactive mode. Removes NIS+ objects with request and confirmation interaction. Similar to the `rm(1)` command, the `nisrm` command asks for confirmation prior to removing an object. If the name specified by *name* is not a fully qualified name this option is forced on. This prevents the removal of unexpected objects.
- f Forces the removal of the NIS+ object, and if it fails for permission reasons, a `nischmod(8)` is attempted and the removal retried. If the command fails, it fails silently.

**NOTES**

This command does not remove directories or nonempty tables. See `nisrmdir(8)` and `nistbladm(8)` for more information.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

`NIS_PATH` If this variable is set and the NIS+ name is not fully qualified, each directory specified is searched until the object is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nisrm` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure; an error occurred.

**SEE ALSO**

nischmod(8), nisdefaults(8), nisrmdir(8), nistbladm(8)

rm(1) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

**NAME**

`nisrmdir` – Removes NIS+ directories

**SYNOPSIS**

`nisrmdir [-f] [-i] [-s hostname] dirname`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisrmdir` command deletes existing NIS+ directories. It can remove a directory outright, or simply remove replicas from serving a directory.

This command modifies the object that describes the directory specified by *dirname*, and then notifies each replica to remove the directory named *dirname*. If the notification of any of the affected replicas fails, the directory object is returned to its original state unless the `-f` option is present.

The `nisrmdir` command accepts the following options and operands:

- `-f` Forces the command to succeed, even though it may not be able to contact the affected replicas. This option is used when a replica is down and does not respond to the removal notification. When the replica is rebooted, it will read the updated directory object (it is no longer a replica for that directory), and stop responding to lookups on that directory. Cleanup of the files that held the now removed directory can be accomplished manually by removing the appropriate files in the `/etc/nis` directory. See `nisfiles(4)` for more information.
- `-i` Specifies interactive mode. Asks for confirmation prior to removing a directory specified by *dirname*.
- `-s hostname` Specifies that the host *hostname* should be removed as a replica for the directory specified *dirname*. If this option is not present all replicas and the master server for a directory are removed and the directory is removed from the namespace.
- dirname* Specifies the directory name.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

- `NIS_PATH` If this variable is set and the NIS+ directory name is not fully qualified, each directory specified is searched until the directory is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nisrmdir` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure; an error occurred.

**SEE ALSO**

`nisdefaults(8)`, `nisrm(8)`

`nisfiles(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nissetup` – Initializes an NIS+ domain

**SYNOPSIS**

`nissetup domain`

`nissetup [-Y] domain`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nissetup` command sets up an NIS+ domain to service clients that want to store system administration information in a domain named *domain*. This domain should already exist prior to executing this command. See `nismkdir(8)` and `nisinit(8)` for more information.

An NIS+ domain consists of an NIS+ directory and its subdirectories: `org_dir` and `groups_dir`. `org_dir` stores system administration information and `groups_dir` stores information for group access control. The `nissetup` command creates the subdirectories `org_dir` and `groups_dir` in the local domain. Both subdirectories will be replicated on the same servers as the parent domain. After the subdirectories are created, `nissetup` creates the default tables that NIS+ serves. These are `auto_master`, `auto_home`, `bootparams`, `cred`, `ethers`, `group`, `hosts`, `netmasks`, `networks`, `passwd`, `protocols`, `rpc`, `services`, `timezone`, and `netgroups`.

The `nissetup` command uses the `nistbladm(8)` command to create these tables. The command can be customized easily to add site-specific tables that should be created at setup time. This command is usually executed just once per domain.

The `nissetup` command accepts the following option and operand:

*domain* Specifies the name of an NIS+ domain.

`-Y domain` Specifies that the domain will be served as both an NIS+ domain as well as an NIS domain using the backward compatibility flag. This set up is less secure because all the system tables will be readable by unauthenticated clients.

**NOTES**

This command creates the default NIS+ tables without data. The `nisaddent(8)` command is used to populate NIS+ tables with data.

**SEE ALSO**

`nisaddent(8)`, `nisinit(8)`, `nismkdir(8)`, `nistbladm(8)`

**NAME**

`nisshowcache` – Prints the contents of the shared cache file

**SYNOPSIS**

`nisshowcache [-v]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisshowcache` command prints the contents of the per-machine NIS+ directory cache that is shared by all processes accessing NIS+ on the machine. By default, `nisshowcache` prints the directory names in the cache along with the cache header. The shared cache is maintained by `nis_cachemgr(8)`.

The `nisshowcache` command accepts the following option:

`-v` Prints the contents of each directory object, including information on the server name and its universal addresses in verbose mode.

**MESSAGES**

Error messages are sent to the `syslogd(8)` daemon.

**FILES**

`/etc/nis/NIS_SHARED_DIRCACHE`

**SEE ALSO**

`nis_cachemgr(8)`, `syslogd(8)`

`nisfiles(4)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nisstat` – Reports NIS+ server statistics

**SYNOPSIS**

`nisstat [-H host] [directory]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisstat` command queries an NIS+ server for various statistics on its operation. These statistics may vary between implementations and from release to release. Not all statistics are available from all servers. Requesting a statistic from a server that does not support that statistic is never fatal, it simply returns the message `unknown statistic`.

By default, statistics are fetched from the server(s) of the NIS+ directory for the default domain. If the `directory` variable is specified, servers for that directory are queried.

Supported statistics for this release are as follows:

*Operations*            This statistic returns results in the form:  
                           `OP=opname:C=calls:E=errors:T=micros`

Where *opname* is replaced by the Remote Procedure Call (RPC) procedure name or operation, *calls* is the number of calls to this procedure that have been made since the server started running, *errors* is the number of errors that have occurred while processing a call, and *micros* is the average time in microseconds to complete the last 16 calls.

*Directory Cache*      This statistic reports the number of *calls* to the internal directory object cache, the number of hits on that cache, the number of misses, and the hit rate percentage.

*Group Cache*            This statistic reports the number of *calls* to the internal NIS+ group object cache, the number of hits on that cache, the number of misses, and the hit rate percentage.

*Uptime*                 This statistic reports how long the service has been running.

The `nisstat` command accepts the following option and operands:

`-H host`      Queries a specified *host*.

*directory*      Specifies a directory. If *directory* is specified, servers for that directory are queried.



**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

**NIS\_PATH** If this variable is set, and the NIS+ directory name is not fully qualified, each directory specified is searched until the directory is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nisstat` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure; an error occurred.

**SEE ALSO**

`nisdefaults(8)`

**NAME**

nistbladm – Administers NIS+ tables

**SYNOPSIS**

```
nistbladm -a [-D defaults] colname=value... tablename
nistbladm -A [-D defaults] colname=value... tablename

nistbladm -a [-D defaults] indexedname
nistbladm -A [-D defaults] indexedname

nistbladm -c [-D defaults] [-p path] [-s sep] type colname=[flags][,access]... tablename
nistbladm -d tablename

nistbladm -m colname=value... indexedname
nistbladm -r [colname=value...] tablename
nistbladm -R [colname=value...] tablename

nistbladm -r indexedname
nistbladm -R indexedname

nistbladm -u [-p path] [-s sep] [-t type] [colname=access...] tablename
```

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The nistbladm command is used to administer NIS+ tables. There are five primary administration operations that nistbladm performs:

- Creates and deletes NIS+ tables
- Deletes NIS+ tables
- Adds entries to NIS+ tables
- Modifies NIS+ table entries
- Removes entries from NIS+ tables

Though NIS+ does not place restrictions on the size of tables or entries, the size of data has an impact on the performance and the disk space requirements of the NIS+ server. NIS+ is not designed to store huge data files. To avoid performance problems, it is recommended that pointer files to large data files be stored in NIS+. NIS+ is a naming service and not a directory service.

NIS+ design is optimized to support 10,000 objects with a total size of 10 megabytes. If your requirements exceed this limit, it is suggested that your domains be broken up into small segments, or the data stored in the tables be pointers to the actual data, instead of the data itself.

When creating tables, a table type, *type*, and a list of column definitions must be provided. *type* is a string, stored in the NIS+ table, which is used to verify the type of new entries that are being added to that table.

The syntax for column definitions is:

```
colname=[flags][,access]
```

For the variable *flags*, specify one or more of the following options:

- S Specifies that searches can be done on the column's values. See `nismatch(8)` for more information.
- I Specifies that searches are case insensitive and to search for uppercase and lowercase. This flag is used with the S flag).
- C Specifies that the column's values are encrypted.
- B Binary data. This flag is not used in combination with the S flag. If not set, the column's values are expected to be null-terminated ASCII strings.
- X External Data Representation (XDR) encoded data. This flag is used in combination with the B flag.

The variable *access* is specified in the format defined by the `nischmod(8)` command. When manipulating entries, `nistbladm` takes two forms of entry name. The first uses a series of space separated *colname=value* pairs that specify column values in the entry. The second is an NIS+ indexed name, *indexedname*, of the form:

```
[colname=value...],tablename
```

The `nistbladm` command accepts the following options and operands:

- a | -A Adds entries to an NIS+ table. The difference between the lowercase `-a` and the uppercase `-A` options is in the treatment of preexisting entries. The entry's contents are specified by the *colname=value* pairs on the command line. The values for all columns must be specified when adding entries to a table.  
Usually, NIS+ reports an error if an attempt is made to add an entry to a table that would overwrite an entry that already exists. This prevents multiple parties from adding duplicate entries and overwriting entries. If you wish to force the add, the uppercase `-A` option specifies that the entry is to be added, even if it already exists. This is analogous to a modify operation on the entry.
- c Creates a table named *tablename* in the namespace. The table that is created must have at least one column and at least one column must be searchable.
- d Destroys the table named *tablename*. The table must be empty. The table's contents can be deleted by using the `-R` option.
- m Modifies an entry in the table that is specified by *indexedname*. Since it is possible to modify the value in a column that would change *indexedname* for an entry, both the column value pair and *indexedname* are required. The user must specify the *indexedname* to look up the entry, modify it, and save it with the new value. The *indexedname* must uniquely identify a single entry.

- r** | **-R** Remove entries from a table. The entry is specified by either a series of *colname=value* pairs on the command line, or an *indexedname* that is specified as *entryname*. The difference between the lowercase **-r** and the uppercase **-R** option is in the treatment of nonunique entry specifications. Usually, the NIS+ server disallows an attempt to remove an entry when the search criterion specified for that entry resolves to more than one entry in the table. However, it is sometimes desirable to remove more than one entry, or all of the entries from a table. In this case, using the uppercase **-R** option forces the NIS+ server to remove all entries matching the passed search criterion. If that criterion is NULL and no column values are specified, all entries in the table are removed.
- u** Updates attributes of a table. This allows the concatenation path (**-p**), separation character (**-s**), column access rights, and table type string (**-t**) of a table to be changed. Neither the number of columns, nor the columns that are searchable may be changed.
- D defaults** Specifies a different set of defaults to be used during this operation. The *default* string is a series of tokens separated by colons. These tokens represent the default values to be used for the generic object properties. Following are descriptions of all the legal tokens.
- t***tl**=*time*  
Sets the default time-to-live for objects that are created by this command. The value *time* is specified in the format as defined by the `nischttl(8)` command. The default value is 12 hours.
- owner*=*ownername*  
Specifies that the NIS+ principal *ownername* should own the created object. The default value is the principal who is executing the command.
- group*=*groupname*  
Specifies that the group *groupname* should be the group owner for the object that is created. The default value is NULL.
- access*=*rights*  
Specifies the set of access rights that are to be granted for the given object. The value of *rights* is specified in the format as defined by the `nischmod(8)` command. The default value is `----rmcdr--r---`.
- p path** Specifies the table search path to use when creating or updating a table. When an `nis_list` function is invoked, the user can specify the flag `FOLLOW_PATH` to tell the client library to continue searching tables in the table's path if the search criteria used does not yield any entries. The *path* argument consists of an ordered list of table names, separated by colons (:). The names in *path* must be fully qualified.
- s sep** Specifies the table's separator character. The separator character is used by `niscat(8)` when displaying tables on the standard output. Its purpose is to separate column data when the table is in ASCII form. The default value is a space.
- t type** Specifies the table's type string.
- colname=value* The column entry value.

*indexedname* A specified entry name.  
*tablename* The name of an NIS+ table.

## WARNINGS

The symbols open bracket ([]) and close bracket (]) are interpreted by the shell. When typing entry names in the form of NIS+ indexed names, the name must be protected by using single quotes. It is possible to specify a set of defaults such that you cannot read or modify the table object later.

## ENVIRONMENT VARIABLES

Following are environment variables that affect the defaults associated with a process.

**NIS\_DEFAULTS** This variable contains a defaults string that overrides the NIS+ standard defaults. If the `-D` option is used, those values override both the `NIS_DEFAULTS` variable and the standard defaults.

**NIS\_PATH** If this variable is set and the NIS+ table name is not fully qualified, each directory specified is searched until the table is found. See `nisdefaults(8)` for more information.

## EXIT STATUS

The `nistbladm` command exits with one of the following values:

Value	Description
0	Success.
1	Failure; an error occurred.

## SEE ALSO

`niscat(8)`, `nischmod(8)`, `nischown(8)`, `nischttl(8)`, `nisdefaults(8)`, `nismatch(8)`, `nissetup(8)`

`nis(4)` in the *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

`nistest` – Returns the state of the NIS+ namespace

**SYNOPSIS**

`nistest [-A] [-t type] [-a rights]`

`nistest [-L] [-t type] [-a rights]`

`nistest [-M] [-t type] [-a rights]`

`nistest [-P] [-t type] [-a rights]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nistest` command returns the state of the NIS+ namespace.

The `nistest` command accepts the following options:

- A Specifies that the data within the table and in all the tables in the first table's concatenation path is returned. This option is only valid when using indexed names or following links.
- L Specifies that if the object or the tablename component of an *indexedname* specifies a LINK type object, the link is followed when this option is present.
- M Specifies that the lookup should be sent only to the master server of the specified data. Although this guarantees that the most up-to-date information is seen it may make the master server busy and therefore unavailable.
- P Specifies that the lookup should follow the concatenation path of a table if the initial search is unsuccessful. This option is only valid when using indexed names or following links.
- t *type* Tests the type of object. The value of *type* can be one of the following:
  - G Return true if the object is a group object.
  - D Return true if the object is a directory object.
  - L Return true if the object is a table object.
  - T Return true if the object is a link object.
  - E Return true if the object is an entry object.
  - P Return true if the object is a private object.
- a *rights* Verifies that the current process has the desired or required access rights on the specified object or entries. The access rights are specified in the same way as the `nischmod(8)` command. See `nischmod(8)` for more information.

**ENVIRONMENT VARIABLES**

Following is an environment variable that affects the defaults associated with a process.

**NIS\_PATH** If this variable is set, and the NIS+ directory name is not fully qualified, each directory specified is searched until the directory is found. See `nisdefaults(8)` for more information.

**EXIT STATUS**

The `nistest` command exits with one of the following values:

<b>Value</b>	<b>Description</b>
0	Success.
1	Failure due to object not present, not of specified type, and/or no such access.
2	Failure due to illegal usage.

**SEE ALSO**

`nischmod(8)`, `nisdefaults(8)`, `nismatch(8)`

**NAME**

`nisupdkeys` – Updates the public keys in an NIS+ directory object

**SYNOPSIS**

`nisupdkeys [-a] [-H host] [directory]`

`nisupdkeys [-C] [-H host] [directory]`

**IMPLEMENTATION**

Cray Research systems licensed for ONC+™ and UNICOS 8.3 or later

**DESCRIPTION**

The `nisupdkeys` command updates the public keys in an NIS+ directory object. When the public key for an NIS+ server is changed, the new key must be propagated to all directory objects that reference that server.

The `nisupdkeys` command reads a directory object and attempts to get the public key for each server of that directory. These keys are placed in the directory object and the object is then modified to reflect the new keys.

If *directory* is specified, the directory object for that directory is updated. If *directory* is not specified, the directory object for the default domain is updated.

Before the `nisupdkeys` command is executed, make sure that the new address/public key has been propagated to all replicas.

The `nisupdkeys` command accepts the following options and operands:

- `-a` Updates the universal addresses of the NIS+ servers in the directory object. Currently, this works for the Transmission Control Protocol, Internet Protocol (TCP/IP) family of transports. This option should be used when the IP address of the server is changed. The server's new address is resolved using `gethost(3C)`. The `/etc/nsswitch.conf` file must point to the correct source for the host's entry.
  - `-C` Specifies to clear rather than set the public key. Communication with a server that has no public key does not require the use of Secure RPC.
  - `-H host` Limits key changes to the server specified by *host*. If the host name is not a fully qualified NIS+ name, then it is assumed to be a host in the default domain. If the specified host does not serve the directory, no action is taken.
- directory* Specified NIS+ directory object.

**NOTES**

The user executing this command must have modify access to the directory object. The existing directory object is displayed by using the `niscat(8)` command with the `-o` option.



This command does not update the directory objects stored in the `COLD_START_FILE` file on NIS+ clients.

**SEE ALSO**

`nisaddcred(8)`, `niscat(8)`

`chkey(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`gethost(3C)` in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

`nis_objects(3N)` in *ONC+ Technology for the UNICOS Operating System*, Cray Research publication SG-2169

**NAME**

npecho – Utility to loop data to the IOS model E low-speed network driver

**SYNOPSIS**

```
/etc/npecho [-D] [-n count] [-t] [-z device]
```

**IMPLEMENTATION**

Cray PVP systems with I/O subsystem model E

**DESCRIPTION**

The npecho utility sends "echo" N-packets to the IOS-E communications driver, receives the replies, and prints timing information if desired. The npecho utility accepts the following options:

- D        Debug flag. Prints debugging statements.
- n *count* Number of echo requests to make. Default is 1.
- t        Prints the average time for an echo packet to be returned.
- z *device* Device special file for npecho to communicate with the np driver. The default is /dev/comm/0030/lp00.

**NOTES**

In a PAL-only system where this command is installed with a privilege assignment list (PAL), a user who runs this command must have one of the following categories active to open and manipulate the network driver: *system*, *secadm*, *sysadm*. In a PRIV\_SU + PAL or PRIV\_SU-only system, the super user can also open and manipulate the network driver.

**SEE ALSO**

np(4) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`nqsresp` – Attempts to open the named pipe that `nqsdaemon` is reading

**SYNOPSIS**

`/usr/air/bin/nqsresp`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nqsresp` command attempts to open the named pipe that the `nqsdaemon` process reads. If the open function fails, either the local Network Queuing System (NQS) daemon is not running or the local NQS daemon is attempting to shutdown and has not completed. In the shutdown state, group and world write access on the pipe are disabled; however, `nqsresp` does not recognize this as a temporary situation and indicates only that it was unable to open the pipe.

**FILES**

`/usr/spool/nqs/private/requests/FIFO`      The `nqsdaemon` pipe

**SEE ALSO**

*UNICOS Resource Administration*, Cray Research publication SG–2302

**NAME**

`nschedv` – Sets memory scheduling parameters

**SYNOPSIS**

```
/etc/nschedv [-B thrash_blocks] [-C compress_interval] [-c cpuhog] [-d] [-f fit_boost]
[-G guarantee_in] [-g guarantee_out] [-H hog_max_mem] [-h memhog] [-i interactive_preferred]
[-K constant_in] [-k constant_out] [-L big_proc] [-M mfactor_in] [-m mfactor_out] [-N nfactor_in]
[-n nfactor_out] [-P pfactor_in] [-p pfactor_out] [-R thrash_interval] [-r cpu_factor] [-T tfactor_in]
[-t tfactor_out] [-V max_runs] [-v] [-X MPX_scheduling] [-x max_outage] [-Y interactive_interval]
[-y small_proc]
```

**IMPLEMENTATION**

Cray PVP systems

**DESCRIPTION**

The `nschedv` command sets and displays memory scheduling parameters using the `schedv(2)` system call. Only an appropriately authorized user can set parameters.

The `nschedv` command accepts the following options:

- B *thrash\_blocks* Specifies the number of blocks as the swap threshold. If this value is exceeded within a specified time period, the scheduler attempts to limit swapping to `SWAPDEV`. You must use `-B` with `-R`, which sets the interval in which *thrash\_blocks* blocks can be swapped out. Swapping caused by user memory requests is not counted towards the total limit set by `-B`. The default for this option is 0, which specifies no limit (no swap threshold is enforced).
- C *compress\_interval* Sets the compression interval, which is the minimum time between memory repacks caused by memory fragmentation. (Memory repacks are also done to move processes into low core as the result of a `plock` system call, or when real-time processes are swapped in; the *compress\_int* parameter has no effect in these cases.) Never set `-C` to 0; this could cause severe performance problems in some circumstances by allowing the swapper to repack memory continuously. The default for `-C` is in the range from 4 to 30, based on the number of megawords of memory.
- c *cpuhog* Specifies threshold, in CPU seconds, at which a process becomes a hog process and therefore comes under hog restrictions. System processes (such as `init`), processes spawned by root, and plocked processes are never considered CPU hogs. The `-h` option (*memhog*) defines the size of a hog process.
- d Displays all parameters. This option should be used in conjunction with other options. If no options are specified with `nschedv`, this is the default behavior.

- f *fit\_boost* Specifies a negative floating-point value that is added to the priority of processes being considered for swap-out if they are equal to or greater in size than the process attempting to swap in. The default for this option is `-2.0`.
- G *guarantee\_in* Specifies a comma-separated pair of floating-point values. The first value is the minimum time a process is guaranteed residency in memory. The second value is a size-proportional value that is calculated based on the normalized size of the process and the `-G` parameter value. The normalized memory size is a value between 0 and 1, based on the size of the process as a percentage of *hog\_max\_mem* (if used) or of the memory available to user processes if *hog\_max\_mem* is 0. The *guarantee\_in* value is multiplied by the normalized memory size and added to the guaranteed residency time for the process. (For sample values of `-G`, see the EXAMPLES section.)
- g *guarantee\_out* Specifies a comma-separated pair of floating-point values. The first value is the minimum time a process is guaranteed residency on SWAPDEV. The second value is a size-proportional value that is calculated based on the normalized size of the process and the `-g` parameter value. The normalized memory size is a value between 0 and 1, based on the size of the process as a percentage of *hog\_max\_mem* (if used) or of the memory available to user processes if *hog\_max\_mem* is 0. The *guarantee\_out* value is multiplied by the normalized memory size and added to the guaranteed residency time for the process. (For sample values of `-g`, see the EXAMPLES section.)
- H *hog\_max\_mem* Specifies an integer value that is the total memory in 512-word blocks (also called *clicks*) that is available for processes greater than or equal in size to *memhog*, or that have used more CPU time than *cpuhog*. Typical values for *hog\_memory* are 60% to 90% of a machine's memory available to user processes, which reserves 10% to 40% for processes that are less than the *memhog* and *cpuhog* definitions. If 0, the hog feature is disabled, which specifies that no space is available for hog processes.
- h *memhog* Specifies the size in clicks at which a process becomes a *hog* process; a process of this size and greater comes under the hog restrictions. System processes (such as *init*) and plocked processes are never considered memory hogs.
- i *interactive\_preferred*

If *interactive\_preferred* is 0, this option is disabled. If *interactive\_preferred* is 1, it indicates that nonhog interactive processes should be queued for swap-in before noninteractive processes of the same category. For example, hog interactive processes are queued before hog batch processes, but not before nonhog batch processes. The following category values are defined and used by the scheduler:

  - 0 Interactive kernel process
  - 1 Batch kernel process
  - 2 Interactive realtime process
  - 3 Batch realtime process
  - 4 Interactive root process

- 5 Batch root process
- 6 Interactive nonhog process
- 7 Batch nonhog process
- 8 Interactive hog process
- 9 Batch hog process
- K *constant\_in* Specifies a constant floating-point value that is added to the priority of an incore process being considered for swap-out.
- k *constant\_out* Specifies a constant floating-point value that is added to the priority of a process on SWAPDEV being considered for swap-in.
- L *big\_proc* Specifies the size, in clicks, at which a process is locked into core. If a process is greater than or equal to *bigproc* in size, it never swaps from memory to the SWAPDEV, unless it makes a memory request or is suspended. If this parameter is set to values less than or equal to one half of memory available to user processes (if *hog\_memory* is set to memory available to user processes), the system may lock until one job completes. A value of 0 disables this behavior.
- M *mfactor\_in* Specifies a positive floating-point value that defines a process' incore priority based on the memory size of that process. The calculation of incore priority, which must be a positive floating-point value, is based on *hog\_memory*, if set (with the -H option), or the memory available to user processes if *hog\_memory* is 0. The system derives this portion of the incore priority using the following equation:
- $$\text{float}(\text{process size})/\text{float}(\text{hog\_memory})*\text{mfactor\_in}$$
- m *mfactor\_out* Specifies a negative floating-point value that defines the swap priority (the order in which processes move onto the swap device (SWAPDEV)), based on the memory size of the process. The calculation of swap priority, which must be a negative floating point value, is based on *hog\_memory*, if set (with the -H option), or on the memory available to user processes if *hog\_memory* is set to 0. The system derives this portion of the swap priority using the following equation:
- $$\text{float}(\text{process size})/\text{float}(\text{hog\_memory})*\text{mfactor\_out}$$
- N *nfactor\_in* Specifies a positive floating-point value that provides a process that swaps into memory a priority based on its nice value. The system derives this portion of the incore priority using the following equation (*p\_nice* is the nice value of the process):
- $$(1.0 - (\text{float}(p\_nice)/\text{float}(\text{NZERO}*2))) * \text{nfactor\_in}$$
- n *nfactor\_out* Specifies a negative floating-point value that provides a process that swaps to SWAPDEV a priority based on its nice value. The system derives this portion of the swap priority using the following equation (*p\_nice* is the nice value of the process):

$$(\text{float}(p\_nice) / \text{float}(\text{NZERO} * 2)) * nfactor\_out$$

- P *pfactor\_in* Specifies a positive floating-point value that is part of the priority calculation when a process swaps into memory. Each process is given a percentage of this value as the process relates to the maximum priority on the system plus 1.0. The system derives the process priority using *pfactor\_in* and the priority of the process swapping into memory (*p\_upri*) in the following equation:

$$(1.0 - (\text{float}(p\_upri) / \text{float}(\text{MAXPRI} + 1))) * pfactor\_in$$

- P *pfactor\_out* Specifies a negative floating point value that is part of the priority calculation when a process swaps onto SWAPDEV. Each process is given a percentage of this value as the process relates to the maximum priority on the system plus 1.0. The system derives the process priority using *pfactor\_out* and the priority of the process swapping into memory (*p\_upri*) in the following equation:

$$(\text{float}(p\_upri) / \text{float}(\text{MAXPRI} + 1)) * pfactor\_out$$

- R *thrash\_interval* Specifies an integer value that is the interval of time, in seconds, in which *thrash\_blocks* (set with -B) blocks can be swapped to SWAPDEV. The default is 0, which disables this option.
- r *cpu\_factor* Specifies an integer value that is the number of running processes that the scheduler attempts to retain in memory. When the scheduler has this number of processes in memory under certain conditions, it does not swap any more processes out of memory. If the value is set too low for your site, idle CPU cycles may result and large processes (that take a large percentage of memory) may swap into memory. This parameter must never be set to 0.
- T *tfactor\_in* Specifies a floating-point value less than or equal to 0 that is added to the priority of an incore process each second. If both *tfactor\_in* and *tfactor\_out* (set with -t) are set to 0, no priority aging or growth occurs.
- t *tfactor\_out* Specifies a floating-point value greater than or equal to 0 that is added to the priority of a process on SWAPDEV each second. If both *tfactor\_in* and *tfactor\_out* (set with -t) are set to 0, no priority aging or growth occurs.
- V *max\_runs* Specifies the maximum number of times the scheduler is allowed to run per second. The default is 0, which allows the scheduler to run an unlimited number of times per second.

`-v` Verification option. Computes and displays swap-in and swap-out priorities for all combinations of priority-based parameters using current `nschedv` settings. The combinations are determined by several tables of values in the `nschedv` program (`priotab`, `upriotab`, `nictab`, and `siztab`); if your site has source, you may change these tables to suit your system's environment. You can use the `-v` option to check the range of priorities for the current settings, as well as to compare the priorities of competing processes.

`-x MPX_scheduling`

Toggles multiplexed scheduling (MPX); not enabled by default. The value 1 enables MPX scheduling, and the value 0 disables it. This feature performs asynchronous swap-out operations and overlaps swap I/Os. This can reduce the overall time that processes wait to swap in, and should improve interactive response time on systems that have a mix of large (usually batch) and small (usually interactive) processes. To enable the MPX scheduling feature, the swap device must be configured with more than one partition; for best performance, the path to each partition should be independent of the others. To improve interactive response, the `-x` option should be used with interactive-preferred scheduling, specified by the `-i` option.

`-x max_outage` Specifies an integer value used to express the maximum swapped-out time of a process. A 0 value disables this option.

`[-Y interactive_interval]`

Sets the interactive interval, in seconds, which is the maximum time that a small process (*small\_proc*) will not be considered for a swap-out operation if there has been no intervening input from a terminal. You must use the `-Y` option with the `-y` option, which sets the size of a small process.

`[-y small_proc]` Specifies the size, in clicks, of a small process. If a process is smaller than *small\_proc*, it will not be swapped out for at least *interactive\_interval* seconds (set with the `-Y` option). For each interaction from a terminal, the time interval for a process is reset.

Swap priorities are used for establishing both the order in which processes are swapped in and the order in which processes are considered for swapping out to make room for a process to swap in.

The swap priorities are established by the addition of process variables that are multiplied against weighting factors established by `nschedv`. The higher the numeric value of the priority, the higher an out-of-memory process is queued for swap-in; the lower the value, the higher an in-memory process is queued for swap-out.

The priority of a swapped-out process being considered for swap-in is computed as follows:

$$\text{constant\_out} + \text{"normalized-size"} * m\text{factor\_out} + \text{"normalized-nice"} * n\text{factor\_out} + \text{"normalized-priority"} * p\text{factor\_out} + \text{"time-swapped"} * t\text{factor\_out}$$

The priority of an incore process being considered for swap-out is computed as follows:

$$\text{constant\_in} + \text{"normalized-size"} * m\text{factor\_in} + \text{"normalized-nice"} * n\text{factor\_in} + \text{"normalized-priority"} * p\text{factor\_in} + \text{"time-swapped"} * t\text{factor\_in}$$



If the process is greater than or equal in size to the process attempting to swap in, *fit\_boost* (see the *-f* option) is added to the calculated priority.

Default values for all of the *nschedv* parameters are calculated at boot time.

*nschedv* does not check parameter settings beyond the checking provided by *schedv(2)*.

## NOTES

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

Active Category	Action
system, secadm, sysadm	Allowed to set parameters.

If the *PRIV\_SU* configuration option is enabled, the super user is allowed to set parameters.

## EXAMPLES

Examples of *-G* option usage are as follows:

- G 60 : size proportional guaranteed residence time of 60 seconds.
- G 10,0 : constant guaranteed residence time of 10 seconds.
- G 5,60 : constant of 5 seconds plus size proportional of up to 60 seconds.

Examples of *-g* option usage are as follows:

- g 60 : size proportional guaranteed residence time of 60 seconds.
- g 10,0 : constant guaranteed residence time of 10 seconds.
- g 5,60 : constant of 5 seconds plus size proportional of up to 60 seconds.

## SEE ALSO

*limit(1)* in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

*limit(2)*, *schedv(2)* in the *UNICOS System Calls Reference Manual*, Cray Research publication SR-2012

**NAME**

ntalkd – Listens at ports and logs debugging information

**SYNOPSIS**

/etc/ntalkd [-d]

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ntalkd` program is the remote server used by the UNIX `talk(1B)` command. It listens at the port indicated in the `talk` service description (see `services(5)`). The actual conversation occurs on a TCP connection that is established by negotiation between the two machines involved.

The `ntalkd` program accepts the following option:

`-d` Through `syslog(3C)`, `ntalkd` logs general debugging information concerning `talk` requests and responses handled by `ntalkd`.

**BUGS**

The protocol is architecture-dependent; the current implementation works between Sun computer systems and Cray Research systems, but not between DEC VAX computer systems and Cray Research systems.

**SEE ALSO**

`talk(1B)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

`syslog(3C)` in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

`services(5)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`ntp` – Queries a network time protocol (NTP) clock

**SYNOPSIS**

```
/etc/ntp [-v] [-s] [-f] [hosts]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ntp` command sends an NTP packet to the NTP daemon that runs on each of the specified hosts. The daemon fills in fields of the NTP packet per RFC 1059 and sends the packet back. `ntp` then formats and prints the result on the standard output.

The default output shows the delay, offset, and remote date and time in `ctime(3C)` format.

You can use options to reset the time of the local system clock.

The `ntp` command accepts the following options:

- `-v` Verbose output. Shows the full contents of received NTP packets, plus calculated offset, displacement, and so on.
- `-s` Sets system time-of-day clock. Happens only if time offset is less than compiled-in constant `WAYTOBIG` (currently 1000 seconds). If remote host is unsynchronized, does not happen. Only root is allowed to use this option.
- `-f` Forces the setting of the system clock regardless of offset. Must be used with `-s` option. If the remote system is unsynchronized, the clock is not reset.

*hosts* Specifies host names.

**NTP Results**

The default output for each *host* appears as follows:

```
128.8.10.1: delay:1.845207 offset:-0.358460 Mon Mar 20 08:05:44 1989
```

The verbose output for each *host* appears as follows:

```

Packet from: [128.8.10.1]
Leap 0, version 1, mode Server, poll 6, precision -10 stratum 1 (WWVB)
Synch Distance is 0000.1999 0.099991
Synch Dispersion is 0000.0000 0.000000
Reference Timestamp is a7bea6c3.88b40000 Tue Mar 7 14:06:43 1989
Originate Timestamp is a7bea6d7.d7e6e652 Tue Mar 7 14:07:03 1989
Receive Timestamp is a7bea6d7.cf1a0000 Tue Mar 7 14:07:03 1989
Transmit Timestamp is a7bea6d8.0ccc0000 Tue Mar 7 14:07:04 1989
Input Timestamp is a7bea6d8.1a77e5ea Tue Mar 7 14:07:04 1989
umd1: delay:0.019028 offset:-0.043890 Tue Mar 7 14:07:04 1989

```

The various fields are interpreted as follows:

Packet from: *[address]*

The address from which this NTP packet was received.

Leap indicator: *n*

The leap second indicator. If a leap second will be added or subtracted at the new year, *n* is nonzero.

version: The version of NTP used by this implementation. The version currently ported to UNICOS is version 1.

mode: May be Server, Client, Broadcast, Symmetric Active, Symmetric Passive, Unspecified, or Reserved. The value depends on the relationship between the local host and the remote host, as specified in the configuration file.

poll = *n* The desired poll rate of the peer (in log base 2 seconds).

precision =  $2^{\text{exponent}}$  (dec)

The claimed precision of the clock (in log base 2 seconds).

stratum: *n (source)*

The stratum of the clock in the NTP hierarchy, along with the source of the clock, which is either the name of a reference standard (such as WWVB or GOES) or the Internet address of the clock from which this clock is derived. Servers with physically attached reference clocks are usually stratum 1. Servers that synchronize with stratum *n* servers are, themselves, stratum (*n*+1) servers.

Synch Distance is *hex-fixedpoint double*

The minimum total round-trip path delay between each server and the primary reference source to which it is synchronized.

Synch Dispersion is *hex-fixedpoint double*

A measure of the variability of the measured round-trip packet delays between each server and the primary reference source to which it is synchronized.

The next five time stamps are given as NTP fixed-point values, in both hexadecimal and `ctime(3C)` format. These are set either by this NTP process, or by the server you are querying.

Reference Timestamp is *hex-timestamp ctime string*

The last time the server clock was adjusted (remote time).

Originate Timestamp is *hex-timestamp ctime string*

The time at which the NTP request was transmitted by you to the server (local time).

Receive Timestamp is *hex-timestamp ctime string*

The time at which the NTP request was received at the server (remote time).

Transmit Timestamp is *hex-timestamp ctime string*

The time at which the NTP response was transmitted by the server (remote time).

Input Timestamp is *hex-timestamp ctime string*

The time at which the NTP response was received by you (local time).

*hostname: delay:time offset:time*

The summary of the results of the query, giving the host name of the responding clock (from the command line), the round-trip delay, and the offset between the two clocks (assuming symmetric round-trip times).

## BUGS

Using `ntp` with the local host shows inaccurate results.

The `ntp` command uses version 1 of the NTP protocol. Version 2, as defined by RFC 1119, may be supported in a future UNICOS release.

## SEE ALSO

`ntpd(8)`, `ntpd(8)`, `ntptime(8)`

`ctime(3C)` in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

RFC 1059, *Network Time Protocol (Version 1) Specification and Implementation*, Revised 15 April 1988, David L. Mills, University of Delaware.

**NAME**

`ntpd` – Performs NTP time synchronization daemon function

**SYNOPSIS**

```
/etc/ntpd [-a threshold] [-n] [-c config-file] [-d] [-D level] [-l] [-r time] [-s]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ntpd` command is the network time synchronization daemon, and it is usually invoked at boot time from the `/etc/netstart` file. It implements a version of the network time protocol (NTP) that is described in RFC 1059. It synchronizes the host's time with a set of distributed time servers, which each have varying accuracy and reliability. Multiple time server masters can exist, but no requirement exists to elect a single master.

The `ntpd` command uses the `adjtime(2)` and `settimeofday(2)` system calls to adjust the clock of the host by small amounts to keep the clock synchronized.

When `ntpd` is started on the machine, it reads configuration information from `/etc/ntp.conf`, which contains information about other NTP time servers and host-specific information. Configuration information is listed one entry per line, with fields separated by white space. Lines that begin with a `#` character are treated as comments. The following is a sample configuration file:

```
#
#           Local clock parameters
#
#           Precision of the local clock to the nearest power of 2
#           ex.
#           60-HZ    = 2**-6
#           100-HZ   = 2**-7
#           1000-HZ  = 2**-10
precision -10
#
peer      foo.umd.edu
peer      192.5.39.94
peer      bar.arpa
server    bogon.umd.edu
```

Two major types of information are specified in the configuration file: local host information and remote time server specification.

The local host information describes the intrinsic properties of the local host's timekeeping machinery. The keyword in this group is `precision`, which takes a number that describes the resolution of the local clock as a power of 2. On Cray Research systems, optimal performance is achieved with a `precision` value of `-10`.

The `driftfile` keyword specifies the name of the file from which the drift compensation register is loaded at initialization time and into which updated values are written. The drift compensation value describes your host's clock intrinsic drift. By default, the `/etc/ntp.drift` file is used.

The time server specifications currently supported are `peer`, `server`, and `passive`. Each keyword takes either a standard Internet dot address or a host name. Each host that is specified with any one of the three keywords is eligible for synchronization; random hosts that set up a peer relationship are not eligible. The `peer` and `server` keywords create an active polling situation; for `peer`, the NTP packets are sent in symmetric-active mode; using `server` causes the packets to be sent in client mode. When accessibility is lost with a configured host in either of these two cases, the daemon continues to poll to reacquire that host. However, a host that is specified with the `passive` keyword does not continue to be polled. If that host begins to poll, it is eligible to be synchronized but is not polled if accessibility is lost.

You must specify most of the configured peers by using the `server` keyword; this minimizes resource usage on the remote NTP server. If your host is serving as a redistribution point for a cluster of hosts, you must set up `peer` relationships with higher-quality clocks (lower stratum) and other equal stratum clocks. Thus, when you are not redistributing time to others, you do not have to configure any `peers` in your NTP configuration, because `server` specifications are used more appropriately.

The `ntpd` command accepts the following options:

- `-a threshold` Sets the threshold that limits the amount `ntpd` changes the system clock. It is used as an ultimate sanity check to prevent your system time from being changed often. By default, the threshold is 1000 seconds. To defeat the sanity check, specify `threshold` in units of seconds or use the string `any`.
- `-n` Specifies that `ntpd` must lock itself into UNICOS memory and not allow itself to be swapped out.
- `-c config-file` Specifies the location of the `ntpd` configuration file. By default, `/etc/ntp.conf` is used.
- `-d` Bumps the debug level by 1. You can specify this option more than once to increment the debug level by 1 each time. If `ntpd` is not compiled with `DEBUG` defined, this option has no effect.
- `-D level` Sets the debug level to the specified value. If `ntpd` is not compiled with `DEBUG` defined, this option has no effect.
- `-l` Logs a message each time the logical clock is changed. Do not specify this option unless you wanted to gather statistical information to analyze the logical clock behavior. When you specify the `-l` option, `ntpd` logs a message approximately every 2 minutes.

- `-r time`           Generates a detailed NTP report every *time* hours and writes the report to the system log file.
- `-s`                Specifies that the local clock must never be adjusted.

## NOTES

Choose your NTP peers and servers carefully; for assistance, send mail to `ntp@TRANTOR.UMD.EDU`.

## BUGS

To prevent a major time warp when running the UNICOS operating system, you must run the `ntp(8)` command with the `-s` and `-f` options before `ntpd` is started. This causes the local time to be the network time during system startup, before UNICOS user processes begin. Cray Research recommends that you use the `ntpstart.sh` script when you start NTP.

Note: To set the time, `root` must invoke the `ntp -s -f` command.

Although the current implementation of `ntpd` uses the NTP version 1 protocol, the NTP protocols are designed to be upwardly compatible. Thus, your Cray Research `ntpd` that is running the version 1 protocol can synchronize to a foreign host that is running NTP version 2 software. However, an NTP version 1 time server does not respond to NTP version 2 requests. Thus, this particular `ntpd` does not synchronize version 2 clients unless the clients disguise their requests as being from version 1 hosts. However, most version 2 implementations allow configuration parameters to be set to allow version 2 clients to disguise their requests to prespecified servers.

## FILES

- `/etc/ntp.conf`      NTP daemon configuration file
- `/etc/ntp.drift`    NTP daemon drift history file

## SEE ALSO

`ntp(8)`, `ntpd(8)`, `ntptime(8)`

`gettimeofday(2)` in the *UNICOS System Calls Reference Manual*, Cray Research publication SR-2012

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

RFC 1059, *Network Time Protocol (Version 1) Specification and Implementation*, Revised 15 April 1988, David L. Mills, University of Delaware. This implementation is based primarily on the version of `ntpd` written by Louis A. Mamakos and Michael G. Petry of The University of Maryland Computer Science Center.



**NAME**

`ntpdc` – Monitors operation of network time protocol (NTP) daemons

**SYNOPSIS**

`/etc/ntpdc [-n] [-v] hosts...`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ntpdc` command sends an `INFO_QUERY` packet to an NTP daemon that runs on the specified hosts. Each daemon responds with information about each of its peers, which `ntpdc` formats on the standard output.

The `ntpdc` command accepts the following options:

- `-n` Disables the printing of the name of the responding host and its peers, printing only the Internet addresses. The default is to print the names.
- `-v` Generates a verbose report. The default is a terse, table-style report.
- `hosts` Specifies hosts on which the NTP daemon is running.

**Terse Report**

A typical terse report looks as follows:

(rem)Address	(lcl)	Strat	Poll	Reach	Delay	Offset	Disp
-umdl	128.8.10.14	1	64	266	3.0	-65.0	0.0
*DCN1.ARPA	128.8.10.14	1	256	332	155.0	-4.0	0.0
128.8.251.92	128.8.10.14	2	64	367	16.0	-61.0	0.0
idunno.Princeto	128.8.10.14	3	64	252	60.0	-53.0	0.0
leo	128.8.10.14	2	64	275	4.0	-273.0	1536.2

The fields in this report are interpreted as follows:

- `-` or `*` The `-` character indicates a preconfigured peer (mentioned in `/etc/ntp.conf`). The `*` character identifies the preconfigured peer (if any) currently being used for synchronization.
- (rem) Address The remote host name or Internet address of a peer.
- (lcl) The local host as specified as an argument to `ntpdc`.
- Strat The stratum level of the peer (as perceived by the local host). Hosts with physically attached reference clocks are usually of stratum 1. Servers that synchronize with stratum  $n$  servers are stratum  $(n+1)$  servers.

Poll	Current polling interval in seconds for this peer.
Reach	Octal value of a shift register that indicates the responses received from the previous eight polls to this peer (see RFC 1059).
Delay	Round-trip delay (in milliseconds) for this peer, as of the latest poll.
Offset	The difference between the local time and the times reported by each host (in milliseconds).
Disp	Current value of the packet dispersion (see RFC 1059) for this peer.

### Verbose Reports

When you use the `-v` option, a series of verbose reports is presented. A typical report looks as follows:

```
Neighbor address 128.4.0.6 port:123 local address 192.35.201.47
Reach: 0376 stratum: 1, precision: -10
dispersion: 0.014000 flags: 0 leap: 0
Reference clock ID: WWV timestamp: a7c2832e.6f9d0000
hpoll: 10, ppoll: 10, timer: 1024 sent: 266 received: 192
Delay(ms) 1144.00 1296.00 1118.00 1115.00 1225.00 1129.00 1086.00 1087.00
Offset(ms) 19.00 92.00 -17.00 12.00 41.00 4.00 -1.00 -14.00

delay: 1086.000000 offset: -1.000000 dsp 0.014000
```

The fields in this report are interpreted as follows:

`Neighbor address` The address and port number of this neighbor, followed by the local address.

`Reach: nn` Reachability in response to last eight polls (octal value of shift register).

`stratum: n` Stratum level. Servers with physically attached reference clocks are usually stratum 1. Servers that synchronize with stratum *n* servers are stratum (*n+1*) servers.

`precision: nn` The precision of this clock (in log base 2 seconds). For example, a clock derived from the power line frequency (60 Hz) has a precision of 1/60 second (about  $2^{-6}$ ) and would be indicated by a precision of `-6`.

`dispersion: nn` A measure of the variation of packet round-trip times to the specified host.

`flags: nn` The flags field is a bit mask, whose bits are defined as follows:

```
#define PEER_FL_CONFIG      0x0001 /* peer is in configuration file */
#define PEER_FL_AUTHENABLE  0x0002 /* peer supports authentication */
#define PEER_FL_SANE       0x0100 /* a "sane" peer */
#define PEER_FL_CANDIDATE  0x0200 /* a candidate peer */
#define PEER_FL_SYNC       0x1000 /* peer can be sync'd to */
#define PEER_FL_BCAST     0x2000 /* broadcast peer */
#define PEER_FL_REFCLOCK   0x4000 /* peer has a local reference clock */
#define PEER_FL_SELECTED   0x8000 /* the "best" peer available */
```

<code>leap: <i>flag</i></code>	The leap second indicator. If a leap second will be added or subtracted at the new year, value is nonzero.
<code>Reference clock ID: [<i>string</i>]</code>	Specifies the type of clock used by the reference server. Examples of common types are Geosynchronous Orbit Environmental Satellite (GOES), Atomic Clock (ATOM), often Cesium-133 based, and WWV, WWVB, WWVL, or WWVH. The last four types represent radio clocks, which are tuned to one of the stations that broadcast time information. The letters are the actual FCC call letters assigned to those stations.
<code>timestamp: <i>nn</i></code>	Fixed-point representation of the time stamp sent by the specified server.
<code>hpoll: <i>time</i></code>	The log base 2 of the number of seconds proposed between polls to this host.
<code>ppoll: <i>time</i></code>	The log base 2 of the number of seconds currently between polls to this host.
<code>timer: <i>nn</i></code>	The number of seconds currently between polls to this host.
<code>sent: <i>nn</i></code>	The number of NTP packets sent to this neighbor.
<code>received: <i>nn</i></code>	The number of NTP packets received from this neighbor.
<code>Delay and Offset</code>	The round-trip delay and clock offset for the last eight NTP packet exchanges. If fewer than eight valid samples exist, the delay field will be 0.
<code>delay: <i>avg-delay</i> offset: <i>avg-offset</i> dsp <i>dispersion</i></code>	Average delay, offset, and dispersion calculated from the preceding samples.

**SEE ALSO**

`ntp(8)`, `ntpd(8)`, `ntptime(8)`

RFC 1059, *Network Time Protocol Specification and Implementation (Version 1)*, Dave Mills, University of Delaware.

**NAME**

`ntpstart` – Starts the network time protocol (NTP) daemon

**SYNOPSIS**

`/etc/ntpstart [ntp_arguments]`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ntpstart` script sets the system time and starts the necessary software for the network time protocol (NTP) at system startup. It is primarily intended to be used as an entry in the `/etc/config/daemons` file that the `sdaemon(8)` script uses.

The `ntpstart` script accepts the following argument:

`ntp_arguments` Specifies argument list to be passed to the `ntpd` daemon.

The `ntpstart` script performs the following functions:

- Scans its argument list (`ntp_arguments`) for any `-c` arguments, which indicate specification of an alternate NTP configuration file for `ntpd`.
- Calls the `ntp(8)` program with any peers or servers that are listed in the NTP configuration file to set the system time.
- Calls the `ntpd(8)` daemon itself and passes its argument list to the daemon.

**SEE ALSO**

`ntp(8)`, `ntpd(8)`, `sdaemon(8)`

*UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG-2304

**NAME**

`ntptime` – Displays offset between local time and real time

**SYNOPSIS**

`/etc/ntptime [-v]`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ntptime` command sends an `INFO_QUERY` packet to all network time protocol (NTP) daemons that run on the hosts found in the `ntp.conf` configuration file. Each daemon responds with information about its time, which `ntptime` formats on the standard output.

The default output specifies only the amount of time the local clock varies from the time reported by the best server.

The `ntptime` command accepts the following option:

`-v` Generates a verbose report, specifying the offsets from all hosts in the configuration file.

**EXAMPLES**

A typical terse output appears as follows:

```
Your clock is running 0.002 seconds slow.
```

When you use `-v` option, a more verbose report is presented. A typical report is as follows:

```
[ *130.126.174.40  8/8  (0.008888/-0.002000) ]
[  128.4.0.1      5/4  (7.507498/0.024999) ]
[  128.8.10.1     4/3  (15.505128/0.013001) ]
Your clock is running 0.002 seconds fast
```

The verbose lines have the following form:

```
[ *Host          sent/recv (disp/offset) ]
```

The fields in the output line have the following meanings:

- `*` Indicates the host used for the actual offset calculation.
- `Host` Host name or host Internet address obtained from the configuration file.
- `sent` Number of packets sent to this host.
- `recv` Number of packets received from this host.

*disp* Dispersion, a measure of the variance of the packet round-trip times to this host. Higher numbers indicate greater variance in round-trip times.

*offset* Difference between the local time and the reference time offered by this host.

**SEE ALSO**

`ntp(8)`, `ntpd(8)`, `ntpd(8)`

RFC 1059, *Network Time Protocol Specification and Implementation (Version 1)*, David Mills, University of Delaware.

**NAME**

nu – Manages user login accounts

**SYNOPSIS**

```
/etc/nu -a [-p path] [-c file] [-D] [-n] [-h]
/etc/nu -m [-p path] [-c file] [-D] [-n] [-h]
/etc/nu -d [-p path] [-c file] [-D]
/etc/nu -g [-p path] [-c file] [-D] [-h]
/etc/nu [-p path] [-c file] [-D] -k users
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `nu` command helps a UNICOS system administrator create, modify, delete, and destroy user accounts. Although everything accomplished by `nu` can be done manually by editing files and issuing shell commands or running `udbgen(8)`, `nu` can direct you through the details correctly by checking file locking, typographical errors, and so on. Alternatively, all the functionality of `nu` is available through a graphical user interface (GUI) by using the `xadmin(8)` command.

The `nu` command accepts the following options (the `-a`, `-m`, `-d`, `-g`, and `-k` options are mutually exclusive):

- a        Adds new accounts. The program prompts you for the login ID, password, name, and other information about each new user. It then creates the account, creates its directories, initializes the contents of the directories, and makes an entry in the log file.
- m        Modifies existing accounts. The program asks for account names (login IDs) and instructions for the changes that you want to make to those accounts, until you indicate that you are finished.
- d        Disables accounts (login IDs) while still leaving an entry in `/etc/udb` (see `udb(5)`) for the deleted account. This prevents the specific user ID from being reused, and it lets you access the accounting data after deleting an account. The program repeatedly asks you for the names of accounts to be deleted and the amount of deleting you want to do to those accounts.  
WARNING: When you use this option, user files are deleted.
- g        Displays and allows editing of the global tape name map and the user database (UDB) default fields. Each field to be changed is selected and its new value assigned. When all of the changes have been made, the new information is written to the UDB. All records created after the changes to the default table will assume the new values. See `udbgen(8)` for more information.

- k *users*    Deletes from the system almost all information that pertains to the specified user IDs, *users*. Specifically, *nu* removes the entry from */etc/udb*, deletes the login directory and all of its contents, and deletes the mailbox. It does not remove that user from any mailing lists in */usr/lib/aliases*. The *-k* option is not interactive. It must be the last option on the command line. You should provide the complete list of users to be deleted as arguments to the option on the command line; you must separate items in this list with spaces.  
WARNING: When you use this option, user files are deleted.
- p *path*    Specifies the path name to access the *nu* and *udb* files. This option allows private or test versions of these files to be maintained. The path name must end with a directory name. The default path name is */etc*.
- c *file*    Specifies the configuration file that *nu* will use. If you specify the *-p* option, the default file is *path/nu.cf60*; otherwise, it is */etc/nu.cf60*.
- D           Causes *nu* to run in debug mode. Debug mode helps you remove bugs from your shell scripts before you run them as *root*. You do not have to be logged in as *root* to run *nu* in debug mode. Typically, this option is used with the *-p* option.
- n           Specifies that the user entry data will be processed and displayed in the new format.
- h           Specifies that section headers will be output when possible. Typically, this option is used with the *-n* option.

### Configuration File

When *nu* is started, it reads configuration commands from the *nu.cf60* file, which contains information needed to create new accounts on your machine. By default, this file is */etc/nu.cf60*. However, if you specify the *-p* option, this file is *path/nu.cf60*; if you use the *-c* option, this file is specified in the *-c* option.

To update the */etc/nu.cf60* file, use the UNICOS Installation Menu System (installation tool). The system administrator must use the installation tool to make site-specific modifications to *nu*. In particular, the GroupHome declarations must be updated to reflect the file systems that hold the login directories for members of different groups. You also should change the default values to match the UDB defaults. If they do not agree, these defaults will be used in preference to those in the UDB, and messages indicating this will be issued whenever *nu* is initiated.

When *nu* is required to create a new directory, it runs a shell script specified in *nu.cf60*. Similarly, when it is required to initialize the files in a newly created directory, it runs another shell script, the name of which it finds in *nu.cf60*. To change configuration and/or customize, you can edit these shell scripts to conform to local practices. If you do edit the scripts, remember that *nu* runs as *root* and that the shell scripts contain possibly irreversible statements, such as *rm -rf \**. Caution is advised.



The `nu.cf60` configuration file is a text file that contains a series of statements, one statement per line. A semicolon that is not inside a quoted string causes the rest of that line to be treated as a comment. Each line in the file that is nonblank, after stripping comments, is treated as an assignment statement. Each statement assigns a value to one variable. All integers are nonnegative integers, and all strings must be delimited with double quotation marks (`"`). You may use the words `none` and `unlimited` without quotation marks in place of a numeric value in the context of resource limits. These special values are translated into the appropriate internal form to specify that none of a resource or an unlimited amount of a resource may be used. No quoting or doubling convention exists for a double-quotation character inside a string. The configuration variables are case-sensitive.

The configuration file contains two types of information:

- System information
- User default information, which is organized in the following categories:
  - General user information
  - User security information
  - User general limits
  - User file limits
  - User MPP limits
  - User tape limits

### System (administration) Information

The system variables in the `nu.cf60` configuration file include the following:

Variable	Description
<code>CreateDir</code>	<p>A string variable that identifies the shell script that is run whenever <code>nu</code> must create a new directory. This shell script must be executable. It is called with the following arguments:</p> <ol style="list-style-type: none"> <li>1 The integer user ID of the desired user.</li> <li>2 The integer group ID of the desired user.</li> <li>3 The name of the user's actual home directory.</li> <li>4 Same function as argument 3 (included for historical reasons).</li> <li>5 An integer that is nonzero if it is possible to overwrite an existing directory with the same name as argument 4.</li> <li>6 An integer that is nonzero if <code>nu</code> is running in debug mode.</li> <li>7 The integer account ID of the desired user.</li> </ol> <p>The standard value for <code>CreateDir</code> is <code>/etc/nulib/nul.sh</code>.</p>

`CreateFiles` A string variable that identifies the shell script that is run when `nu` must initialize a directory (newly created or otherwise) with some standard files. The system administrator must create the `/usr/skel/[a-z]*` directory and it must be initialized with files appropriate to the particular site. The `nu2.sh` script copies `/usr/skel/[a-z]*` into a new login directory. This shell script must be executable. It is called with the following arguments:

- 1 The name of the login directory to be initialized.
- 2 The integer user ID of the user.
- 3 The integer group ID of the user.
- 4 An integer that is nonzero if setup for the Cray station will be performed.
- 5 The account name of the desired user.
- 6 An integer that is nonzero if `nu` is running in debug mode.

The standard value for `CreateFiles` is `/etc/nulib/nu2.sh`.

`DeleteAccts` A string variable that identifies the shell script that is run whenever `nu` must delete a user's account that was created in some earlier session with `nu`. Deleting accounts involves changing the password so that the user cannot log in, deleting all of the user's files and directories, and deleting the user's mailbox. This shell script must be executable. It is called with the following arguments:

- 1 The login ID of the account to be deleted.
- 2 The login directory for that account.
- 3 The name given in `/etc/udb` for the login directory (which might possibly be a symbolic link to item 2 and therefore it must be specified separately).
- 4 An integer that is nonzero if `nu` is running in debug mode.

The standard value for `DeleteAccts` is `/etc/nulib/nu4.sh`.

`DestroyAccts` A string variable that identifies the shell script that is run whenever `nu` must destroy a user's account that was created in some earlier session with `nu`. Destroying accounts involves removing the user from the user database, deleting all of the user's files and directories, and deleting the user's mailbox. For a less drastic account removal action, see `DeleteAccts`.

This shell script must be executable. It is called with the following arguments:

- 1 The login ID of the account to be deleted.
- 2 The login directory for that account.
- 3 The name given in `/etc/udb` for the login directory (which might be a symbolic link to item 2 and therefore it must be specified separately).
- 4 An integer that is nonzero only if `nu` is running in debug mode.

	The standard value for <code>DestroyAccts</code> is <code>/etc/nulib/nu3.sh</code> .
<code>Logfile</code>	A string variable that specifies the file in which all nu transactions are logged. The standard value of <code>Logfile</code> is <code>/usr/adm/nu.log</code> .
<code>MaxGroups</code>	An integer variable that defines the maximum number of <code>GroupHome</code> symbols contained in the configuration file. Released as 100.
<code>MaxGroupAdm</code>	An integer variable that defines the maximum number of <code>GroupAdm</code> symbols contained in the configuration file. Released as 100.
<code>MaxNameLength</code>	An integer variable that gives the maximum number of characters permitted in a login name; for unmodified UNICOS systems, it should be set to 8. It must never exceed 15; more than 8 is not recommended because many standard UNIX command output formats expect up to 8 characters.
<code>MaxSymbols</code>	An integer variable that defines the maximum number of general symbols contained in the configuration file. Released as 100.
<code>WantCSsetup</code>	An integer variable that should be set to 1, if you want nu to set up for the Cray station. nu just passes the value of <code>WantCSsetup</code> through to the shell script specified in <code>CreateFiles</code> , which does the actual initialization. The standard value is 1.

**User Default Information**

The user default variables in the `nu.cf60` configuration file are organized by category.

The general user information category includes the following:

<b>Variable</b>	<b>Description</b>
<code>DefaultAcids</code>	Default account IDs. Typical value is 0.
<code>DefaultAge</code>	Default password aging. Four fields are provided to define the aging characteristics that the UNICOS operating system maintains. The fields are written from left to right, separated by commas. You may omit fields to the right if you do not need default values for those fields.

```
DefaultAge = "[flags [maxage [minage [age]]]" "prompt" ; comment
```

These fields have the following meanings:

<b>Field [Values]</b>	<b>Meaning</b>
<i>flags</i> [superuser, force, none]	superuser means only an administrator changes passwords. force means the user must change the password on the next login. none means neither of these is enabled.
<i>maxage</i> [0w through 64w]	Maximum age of the password. An optional suffix to specify the unit may be provided. w means weeks, d means days, and s means seconds. The default unit is weeks.
<i>minage</i> [0w through 64w]	Minimum age of the password. An optional suffix to specify the unit may be provided. w means weeks, d means days, and s means seconds. The default unit is weeks.
<i>age</i> [0w < value]	Initial age of the password. The value 0 means the present date and time. An optional suffix to specify the unit may be provided. w means weeks, d means days, and s means seconds. The default unit is weeks. No specific upper bound to the age is imposed, although an age that would translate to a date more than 64 weeks in the past may not be honored and makes no sense.
<i>prompt</i> [on or off]	You must turn this default either on or off. If it is off, the specified default is used; you are not prompted for these values when you are adding new users. If it is on, you are prompted for information.

Examples:

The following example shows how to set the default to force changes at login with a maximum age of 21 days and a minimum age of 2 days. An initial password is unnecessary because force makes the password appear to be expired. The default is on; thus, the administrator is given the opportunity to change the values.

```
DefaultAge "force, 3w, 2d" "on"
```

You also can force the password to be changed by setting the initial age older than the maximum age:

```
DefaultAge "none, 3w, 2d, 22d" "on"
or
DefaultAge "3w, 2d, 22d" "on"
```

If the flag is not wanted, it is correct either to specify none or to omit it because, for the default, no flags are predefined.

DefaultAl	Default Data Migration Facility (DMF) archive limit (in disk blocks). Released as 0.
DefaultAm	Default Data Migration Facility (DMF) media type. Typical value is 0.
DefaultAs	Default allocation shares. Typical value is 100.
DefaultDr	Default login root string. Typical value is /.
DefaultGids	Default group IDs. Typical value is 0.
DefaultPm	Default permission name string. Typical value is none.
DefaultRg	Default resource group ID. Typical value is 0.
DefaultPb	Default permbits. Typical value is none.
DefaultHome	A string variable that is set to the file system or top-level directory that will be used to hold the login directory for accounts in groups not explicitly set up to have their login directories elsewhere. When you are creating a new account, nu asks you for the group number of the group in which you would like the account held. If that group number is mentioned in a GroupHome declaration, the home directory for the group is the one specified in that GroupHome declaration. If the group number is not mentioned in a GroupHome declaration, login accounts created in that group will have their login directories put into DefaultHome. There is no standard value for DefaultHome.
GroupHome	A pseudo variable that is a name by which the configuration code can load entries into a directory location table. In particular, if you provide two GroupHome declarations, they are both processed; if you provide two of any other declaration, only the latest one has any effect. For example, a typical set of GroupHome declarations might look like the following:

```
GroupHome= 10 "/usr"
GroupHome= 20 "/mnt"
GroupHome= 25 "/usr/cis"
GroupHome= 31 "/usr/guest"
```

The GroupHome declarations serve as default login directory location information for new accounts. You can put any account anywhere you want; however, the GroupHome information makes the defaults appear in the correct places.

Consequently, to create a new account, press the <RETURN> key to accept the defaults. In the preceding example, the declarations would cause group 10 to default to /usr (for example, /usr/smith or /usr/jones). Group 31 would default to /usr/guest (for example, /usr/guest/smith or /usr/guest/jones). If the login group is not mentioned in a GroupHome declaration, the DefaultHome variable is used.

DefaultShell A string variable that is set to the name of the shell file to be used by default. The standard value for DefaultShell is /bin/csh.

The user security information category includes the following:

Variable	Description
DefaultDl	Default default security level name or number. Typical value is 0.
DefaultMx	Default maximum security level name or number. Typical value is 0.
DefaultMn	Default minimum security level name or number. Typical value is 0.
DefaultCm	Default valid compartment name string. Typical value is none.
DefaultDc	Default active compartment name string. Typical value is none.
DefaultDcat	Default default category name string. Typical value is none.
DefaultVcat	Default valid category name string. Typical value is none.
DefaultDl_class	Default default integrity class. A typical value is 0. The use of integrity class values is not supported.
DefaultMx_class	Default maximum integrity class. A typical value is 0. The use of integrity class values is not supported.

Each of the following defaults sets a limit for the user. These limits have the format:

*variable = value "queue" "prompt"*

They can be any of the following:

Variable	Meaning
<i>variable</i>	The variable name, DefaultAge.
<i>value</i>	The default value for the variable: none, unlimited, or an integer number.
<i>queue</i>	The run mode for the session: batch or integer.
<i>prompt</i>	The state of the prompt (the same as DefaultAge): on or off.

The general user limits category also includes the following variables:

Variable	Description
DefaultCj	Default CPU time per session. Released as unlimited.
DefaultCp	Default CPU time per process. Released as unlimited.
DefaultMj	Default memory per-session limit. Released as unlimited.
DefaultMp	Default memory per-process limit. Released as unlimited.
DefaultNc	Default session nice value adjustment. Released as none.
DefaultSc	Default per-session socket buffer limit. Released as unlimited.

DefaultSj            Default per-session shared memory segments. Released as none.  
 DefaultSm            Default per-session shared memory size limit. Released as none.

The file limits category includes the following variables:

<b>Variable</b>	<b>Description</b>
DefaultFd	Default per-process open file size limit. Released as unlimited.
DefaultJl	Default per-session file size limit. Released as unlimited.
DefaultJs	Default per-session SDS limit. Released as none.
DefaultPc	Default per-process core file size limit (512-word blocks). Released as unlimited.
DefaultPs	Default per-process SDS limit. Released as none.
DefaultUl	Default per-process file size limit. Released as unlimited.

The MPP limits category includes the following variables:

<b>Variable</b>	<b>Description</b>
DefaultMb	Default per-session MPP barrier limit. Released as none.
DefaultMt	Default per-session MPP time limit. Released as none.
DefaultPe	Default per-session MPP processing element (PE) limit. Released as none.
DefaultPt	Default per-process MPP time limit. Released as none.

The tape limits category includes the following variable:

<b>Variable</b>	<b>Description</b>
DefaultTp	Default per-session tape limit. Released as none. You must define this default for each of the eight tape types; also, each of the tape types must specify a default for both batch and inter and must be either on or off. (See EXAMPLES.)

**NOTES**

The nu command traps interrupt signals (such as <CONTROL-c> and does not end if you try to stop it in the middle of a critical section. Critical sections are primarily the updates of the UDB. A list of changes is recorded in a log file (usually /usr/adm/nu.log).

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
system, secadm	Can modify all UDB fields.
sysadm	Can modify all UDB fields, except sensitive fields.

If the PRIV\_SU configuration option is enabled, the administrator running as the super user may modify all UDB fields.

**BUGS**

The extensive use of shell scripts for doing sensitive duties such as deleting and changing accounts means that someone may be able to make nu fail in destructive ways without having access to the source code.

**EXAMPLES**

The following example shows how to select a user's password if the machine-generated password feature is enabled:

```
New password is: lempamdo
Re-enter new password or (CR) to get another:
```

The following are examples of the Default.xx directives:

General user limits category

```
DefaultCj = unlimited      "batch" "off" ; default CPU time per session
DefaultCj = unlimited      "inter" "off" ; default CPU time per session
DefaultCp = unlimited      "batch" "off" ; default CPU time per process
DefaultCp = unlimited      "inter" "off" ; default CPU time per process
DefaultMj = unlimited      "batch" "off" ; default memory per-session limit
DefaultMj = unlimited      "inter" "off" ; default memory per-session limit
DefaultMp = unlimited      "batch" "off" ; default memory per-process limit
DefaultMp = unlimited      "inter" "off" ; default memory per-process limit
DefaultNc = none           "batch" "on" ; per-session nice value
DefaultNc = none           "inter" "on" ; per-session nice value
DefaultSj = none           "batch" "on" ; shared memory limit per session
DefaultSj = none           "inter" "on" ; shared memory limit per session
DefaultSm = none           "batch" "on" ; shared memory segments per session
DefaultSm = none           "batch" "on" ; shared memory segments per session
```

User file limits category

```
DefaultFd = unlimited      "batch" "on" ; default per-process open file limit
DefaultFd = unlimited      "inter" "on" ; default per-process open file limit
DefaultJl = unlimited      "batch" "on" ; default per-session file size limit
DefaultJl = unlimited      "inter" "on" ; default per-session file size limit
DefaultUl = unlimited      "batch" "off" ; default file size limit
DefaultUl = unlimited      "inter" "off" ; default file size limit
DefaultJs = none           "batch" "off" ; default per-session SDS limit
DefaultJs = none           "inter" "off" ; default per-session SDS limit
DefaultPs = none           "batch" "off" ; default per-process SDS limit
DefaultPs = none           "inter" "off" ; default per-process SDS limit
DefaultPc = unlimited      "batch" "on" ; default per-process core file size limit
DefaultPc = unlimited      "inter" "on" ; default per-process core file size limit
```



## User MPP limits category

```

DefaultPe = none      "batch" "on" ; default per-session MPP PE limit
DefaultPe = none      "inter" "on" ; default per-session MPP PE limit
DefaultMt = none      "batch" "on" ; default per-session MPP time limit
DefaultMt = none      "inter" "on" ; default per-session MPP time limit
DefaultMb = none      "batch" "on" ; default per-session MPP barrier limit
DefaultMb = none      "inter" "on" ; default per-session MPP barrier limit
DefaultPt = none      "batch" "on" ; default per-process MPP time limit
DefaultPt = none      "inter" "on" ; default per-process MPP time limit

```

## User tape limits category

```

DefaultTp = none "type0" "batch" "on" ; default per-session tape limit
DefaultTp = none "type0" "inter" "off" ; default per-session tape limit
DefaultTp = none "type1" "batch" "off" ; default per-session tape limit
DefaultTp = none "type1" "inter" "off" ; default per-session tape limit
DefaultTp = none "type2" "batch" "off" ; default per-session tape limit
DefaultTp = none "type2" "inter" "off" ; default per-session tape limit
DefaultTp = none "type3" "batch" "off" ; default per-session tape limit
DefaultTp = none "type3" "inter" "off" ; default per-session tape limit
DefaultTp = none "type4" "batch" "off" ; default per-session tape limit
DefaultTp = none "type4" "inter" "off" ; default per-session tape limit
DefaultTp = none "type5" "batch" "off" ; default per-session tape limit
DefaultTp = none "type5" "inter" "off" ; default per-session tape limit
DefaultTp = none "type6" "batch" "off" ; default per-session tape limit

```

You can replace the tape names (such as "type0") by the names defined in the global tape map after it has been defined in the UDB.

## FILES

The default directory for the following files is /etc; to change it, use the -p or -c option.

/etc/acid	Account file
/etc/group	System group file
/usr/include/udb.h	User database structure
/etc/nu.cf60	Configuration file
/etc/nulib/*.sh	Shell scripts to perform the work
/etc/nulib/nu.cf60	Release default configuration file
/etc/udb	User validation file that contains user control limits
/etc/udb.public	Public version of the user database with security sensitive information removed

**SEE ALSO**

udbgen(8), xadmin(8)

spset(1), udbsee(1) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

getgrent(3C), getpwent(3C) in the *UNICOS System Libraries Reference Manual*, Cray Research publication SR-2080

group(5), udb(5) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

*General UNICOS System Administration*, Cray Research publication SG-2301

**NAME**

nulladm – Creates an empty file owned by adm

**SYNOPSIS**

*/usr/lib/acct/nulladm file*

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The nulladm command creates a file with mode 664 and ensures that the owner and group are adm. nulladm is included as part of many accounting scripts, and it rarely is invoked alone.

**NOTES**

Sites may allow users in group adm to run Cray Research system accounting (CSA), if they have the permission bit `acct` set in their user database (UDB) entries. However, such users cannot run accounting after a super user has done so, because the group ID and permissions of the files will change. In this case, execute the `csaperm(8)` command to reset group IDs and permissions so that nonsuper users can run accounting again.

**SEE ALSO**

`acct(8)`, `acctsh(8)`, `csa(8)`

*UNICOS Resource Administration*, Cray Research publication SG–2302

**NAME**

`nwmstart` – Initializes networking media

**SYNOPSIS**

`/etc/nwmstart`

**IMPLEMENTATION**

Cray Research systems with I/O subsystem model E

**DESCRIPTION**

The `nwmstart` script initializes the underlying networking media necessary for communication over networking interfaces, according to information contained in a set of tabular configuration files. On IOS model E systems, this consists of executing `fdload(8)` to initialize the FDDI channels that are used for networking.

`nwmstart` is intended to be used primarily by the `netstart(8)` command during system startup.

**File Format**

The `nwmstart` configuration file for the IOS model E systems is `/etc/config/fddi_config`. It consists of a series of lines with the following format:

```
interface ordinal action
```

The elements in the entry have the following meanings:

*interface ordinal*

Ordinal of the FDDI interface channel being referenced. This ordinal should correspond to the `fddev` ordinal in the `param` file for the same channel.

*action*

Indication of whether to initialize the channel. An *action* value of YES or ON (case is insignificant) indicates that `nwmstart` will initialize FDDI device number *interface ordinal* on the appropriate attached IOS system by executing `fdload` with appropriate arguments.

**EXAMPLES**

The following example `/etc/config/fddi_config` file initializes one FDDI channel on the attached IOS system:

```
#
# Example fddi_config file to initialize one FDDI channel
#
0 YES
1 NO
```

**FILES**

`/etc/config/fddi_config` FDDI channel configuration file

**SEE ALSO**

`fddiload(8)`, `netstart(8)`

**NAME**

olcfpt, olcm, olcrit, olcsvc, olibuf, olsbt – Online diagnostic CPU confidence tests

**SYNOPSIS**

```

/etc/diag/olcfpt [chkpnt mode] [cpu clist] [cputime h:m:s] [+/-getseed]
[getseed file] [help] [maxerr n] [maxp n] [+/-modes] [+/-waitded] [+/-parcel]
[time h:m:s] [+/-verbose] [disable ilist] [enable ilist] [+/-isolate] [isop n]
[numins n] [+/-repeat] [seed n] [vl n] [+/-vload] [+/-fpbits] [+/-fprand]
[+/-random] [+/-fpadd] [+/-fpmult] [+/-fprecip] [+/-scalar] [+/-vector]

/etc/diag/olcm [chkpnt mode] [cpu clist] [cputime h:m:s] [+/-getseed] [getseed file]
[help] [maxerr n] [maxp n] [+/-modes] [+/-waitded] [+/-parcel] [time h:m:s]
[+/-verbose] [+/-repeat] [section slist] [seed n] [stride n] [words n]

/etc/diag/olcrit [chkpnt mode] [cpu clist] [cputime h:m:s] [+/-getseed]
[getseed file] [help] [maxerr n] [maxp n] [+/-modes] [+/-waitded] [+/-parcel]
[time h:m:s] [+/-verbose] [+/-ranclus] [cluster n] [disable ilist] [+/-eis]
[enable ilist] [+/-hwbmm] [+/-isolate] [isop n] [numins n] [+/-repeat] [seed n] [vl n]
[+/-vload] [+/-bits] [+/-onezero] [+/-random] [+/-address] [+/-ci] [+/-cm]
[+/-fpadd] [+/-fpmult] [+/-fprecip] [+/-int] [+/-jump] [+/-logical] [+/-pop]
[+/-scalar] [+/-shift] [+/-shr] [+/-vector]

/etc/diag/olcsvc [chkpnt mode] [cpu clist] [cputime h:m:s] [+/-getseed]
[getseed file] [help] [maxerr n] [maxp n] [+/-modes] [+/-waitded] [+/-parcel]
[time h:m:s] [+/-verbose] [disable ilist] [enable ilist] [+/-eis] [+/-isolate] [isop n]
[numpar n] [+/-repeat] [seed n] [+/-sgci] [vl n] [+/-vuse vrlist] [+/-onezero]
[+/-random] [+/-slide] [+/-cm] [+/-fpadd] [+/-fpmult] [+/-fprecip] [+/-int]
[+/-logical] [+/-pop] [+/-shift] [+/-lz]

/etc/diag/olibuf [chkpnt mode] [cpu clist] [cputime h:m:s] [+/-getseed]
[getseed file] [help] [maxerr n] [maxp n] [+/-modes] [+/-waitded] [+/-parcel]
[time h:m:s] [+/-verbose] [+/-repeat] [seed n] [+/-bits] [+/-onezero] [+/-ozpar]
[+/-random] [+/-solid]

/etc/diag/olsbt [chkpnt mode] [cpu clist] [cputime h:m:s] [+/-getseed] [getseed file]
[help] [maxerr n] [maxp n] [+/-modes] [+/-waitded] [+/-parcel] [time h:m:s]
[+/-verbose] [+/-anycpus] [cluster n] [numins n] [+/-repeat] [seed n] [+/-bits]
[+/-onezero] [+/-random]

```

**IMPLEMENTATION**

Cray PVP systems. The olsbt test does not run on CRAY T90 systems.

**DESCRIPTION**

Online diagnostic CPU confidence tests provide a comprehensive performance check of the system hardware. The CPU confidence tests are as follows:

`olcftp` Comprehensive floating-point test  
`olcm` Central memory test  
`olcrit` Comprehensive random instruction test  
`olcsvc` Comprehensive scalar and vector comparison test  
`olibuf` Instruction buffer test  
`olsbt` Semaphore, shared B and shared T register test

Each confidence test's command synopsis includes monitor options, which are listed first in the synopsis, and test-specific command options. The monitor options are common to all confidence tests. You can enter the test options in any order. If an option is omitted, the default value is used. The following options provide different methods of specifying the starting seed value (specify only one for each test executed):

`+/-getseed`, `getseed file`, or `seed n`. Test results are sent to `stdout` by default or to a file when UNICOS output redirection is specified on the command line.

**Monitor Options**

The monitor options are as follows:

`chkpnt mode`

Indicates whether restart files are to be generated. *mode* is one of the following arguments:

`first` Generates a restart file for the first failure detected (default)  
`all` Generates a restart file for each failure detected, including failures detected during error isolation  
`none` Does not generate restart files

Restart file names follow the format:

*pid-master\_cpu\_digit diagnostic\_name*

For example, if `olcrit` is run in CPU `c` with a process ID of 87865, the restart file will be named `87865-2olcrit`

`cpu clist` Selects the CPUs to be tested. Enter *clist* in the following format:

*x,x,...,x*

*x* can be a, b, c, ..., and so on, or 0, 1, 2, ..., and so on, up to the maximum number of CPUs. The range of CPU selection using characters is from a through p. Digits must be used to select CPUs above 16. The first CPU selected is the master CPU. The default is `cpu a`. (For `olsbt`, the default is `cpu a,b` on multiple-CPU systems and `cpu a` on single-CPU systems.) The test output always specifies the CPU(s) as a digit.

The recommended usage of this option is `cpu x`, where *x* is a single CPU designator. Specifying more than one CPU at a time with this option causes increased wall-clock completion times. The increase in time could be substantial on heavily loaded systems. The exception to this recommendation is the `olsbt` test, which will complete fastest when two CPUs are selected. The `olsbt` test does not run efficiently on single-CPU systems. It is recommended that `olcrit` be used to test the shared B and T register instructions and the set and clear semaphore instructions.

The `olcm` test cannot be executed in multiple-CPU mode (the `cpu list` option can contain only a single CPU designator for *list*; for example, `cpu b`).

`cputime h:m:s`

Sets the execution time in CPU time. The time is specified in hours (*h*), minutes (*m*), and seconds (*s*); minutes and seconds; or just seconds.

Generally, actual execution time is within 1 second of the specified CPU time. If `cputime` is allowed to default or is set to 0, the test uses the `maxp` value. However, if set to a value other than 0, `cputime` overrides `maxp`.

`+/-getseed`

Enables or disables the option that reads the file `test.seed` to obtain a starting seed. If the test terminates because the maximum pass or error limit is reached, the seed from the last pass is saved in the file `test.seed`. If there are any problems reading the seed from this file, the test uses the default seed (O'33). If you select `+getseed`, do not select `seed n`. The default is `-getseed`.

`getseed file`

Gets a starting seed from *file*. *file* can contain a dump from a previous failure or a single seed value. If allowed to default, the test uses the seed value specified by `+getseed` or `seed n`.

`help` Generates an online help display that contains a command synopsis.

`maxerr n`

Sets the maximum number of errors (in octal). The default for *n* is O'1.

`maxp n` Sets the maximum number of passes (in octal). The default for *n* is O'1000. If `cputime` or `time` is set to a value other than 0, the specified option overrides `maxp`.



`+/-modes`

Enables or disables the option that tests exchange package mode bits. Only the BDM, AVL, FPE, ORE, and IUM mode bits are tested. Before the CPU test is run in the selected CPU the mode bits are tested. If the mode bits do not set and clear properly, the test will abort and report the failing mode bits. The mode bit test will not run when more than one CPU is selected. The default is `-modes`.

`+/-waitded`

Enables or disables the wait for a dedicated CPU. When this option is enabled, the diagnostic waits for the dedicated CPU to be freed before running. When this option is disabled, the diagnostic does not attempt to run in the dedicated CPU; instead, it exits normally and prints an informative message that the CPU was dedicated. The default is `-waitded`. See the `cpu(8)` man page for more information on dedicating a CPU.

`+/-parcel`

Enables or disables the option that forces dumped data to parcel format. `+parcel` forces data that would otherwise be in word format (64 bits in octal, with leading 0's) to parcel format (four groups of 16 bits in octal). Parcel format displays 2 words (8 parcels) per line. Word format displays 4 words per line. The default is `-parcel`.

`time h:m:s`

Sets the test execution time in elapsed (wall-clock) time. The time is specified in hours (*h*), minutes (*m*), and seconds (*s*); minutes and seconds; or just seconds.

Generally, actual execution time is within 1 second of the specified elapsed time. If `time` is allowed to default or is set to 0, the test uses the `maxp` value. However, if set to a value other than 0, `time` overrides `maxp`.

`+/-verbose`

Enables or disables the generation of informational messages. `+verbose` causes a line of output to be generated after each pass of the test. The default is `-verbose`.

### **olcftp Test-specific Options**

The test-specific command options for `olcftp` are as follows:

`disable ilist`

Deselects specific instructions. Enter *ilist* in the following format:

*n,n,...,n*

*n* is the octal value in the *gh* field of the specific instruction. This option overrides the `enable ilist` option and any selected (+) or deselected (-) instruction options.

`enable ilist`

Selects specific instructions. Enter *ilist* in the following format:

*n,n,...,n*

$n$  is the octal value in the  $gh$  field of the specific instruction. This option overrides any selected (+) or deselected (-) instruction options. When a test is run with default values for the +/- instruction options, and the enable *ilist* option is selected, only the instructions specified by the enable *ilist* option are run.

`+/-isolate`

Enables or disables the error isolation option. The default is `+isolate`.

`isop  $n$`  Sets the isolation pass limit to  $n$  (octal). During isolation, the test repeatedly executes the suspected failing sequence. If the sequence fails, the loop terminates and the test attempts to isolate the sequence further. If the sequence does not fail, the loop terminates after  $n$  passes, and the test assumes that the error is not in the tested sequence. The default for  $n$  is `O'1000`.

`numins  $n$`

Sets the number of instructions to be generated.  $n$  can be any octal value within the range 1 through 20. The default for  $n$  is `O'20`.

`+/-repeat`

Enables or disables the option that repeats the first pass until the test terminates. This option is useful for recreating an error and is usually used with one of the following options: `seed  $n$` , `+getseed`, or `getseed file`. The default is `-repeat` (the program generates new instructions and data after each pass).

`seed  $n$`  Sets the random seed to  $n$ .  $n$  can be any 64-bit octal value. If  $n$  is 0, the test reads the real-time clock and uses the value for the initial seed. The default for  $n$  is `O'33`. If this option is selected, do not select `+getseed` or `getseed file`.

`v1  $n$`  Sets the vector length to  $n$ .  $n$  can be any octal value in the range 0 through 100 ( $n$  can be up to `O'200` for the CRAY C90 series and CRAY T90 series). If  $n$  is set to 0, a random value is used to initialize the test. The default for  $n$  is `O'100` (`O'200` for the CRAY C90 series and CRAY T90 series).

`+/-vload`

Selects or deselects vector instructions for the instruction buffer and, in the case of `-vload`, does not allow loading or saving of the vector registers. `-vload` overrides vector instructions selected by `+vector` and enable *ilist*. The default is `+vload`.

`+/-fpbits`, `+/-fprand`, `+/-random`

Selects or deselects specific data patterns. If allowed to default, all data patterns are run. If the `v1` option is 0, the vector length register is initialized with 6 bits of random data. The data patterns are as follows:

<code>fpbits</code>	Random number of consecutive 1-bits in the coefficient. Exponent data depends on the floating-point instruction.
<code>fprand</code>	Random bit generation in the coefficient. Exponent data depends on the floating-point instruction.
<code>random</code>	Random bit generation in a word.

`+/-fpadd, +/-fpmult, +/-fprecip, +/-scalar, +/-vector`

Selects or deselects specific instruction groups for the following options:

<code>fpadd</code>	Floating-point addition
<code>fpmult</code>	Floating-point multiply
<code>fprecip</code>	Floating-point reciprocal
<code>scalar</code>	Scalar instruction (destination)
<code>vector</code>	Vector instruction (destination)

If allowed to default, all instruction groups are run. The groups are as follows:

<code>fpadd</code>	062, 063, 170 through 173
<code>fpmult</code>	064 through 067, 160 through 167
<code>fprecip</code>	070, 174
<code>scalar</code>	062, 063, 064 through 067, 070
<code>vector</code>	160 through 167, 170 through 174

### **o1cm Test-specific Options**

The test-specific command options for `o1cm` are as follows:

`+/-repeat`

Enables or disables the option that repeats the first pass until the test terminates. This option is useful for recreating an error and is usually used with one of the following options: `seed n`, `+getseed`, or `getseed file`. The default is `-repeat` (the program generates new instructions and data after each pass). This option is valid only for test section 6.

`section slist`

Selects the test sections to be executed. Enter *slist* in the following format:

*n,n,...n*

*n* can be any of the following test sections, entered in any order (if allowed to default, all test sections are executed):

- 1 Central memory storage and scalar path test.
- 2 Central memory storage and T-register path test.
- 3 Central memory storage and B-register path test.
- 4 Central memory storage and vector register path test. Simplified algorithm for quick memory address testing.
- 5 Central memory storage and vector register path test using both vector logical units.
- 6 Central memory random-data test.
- 7 Central memory conflict test.

`seed n` Sets the random seed to *n*. *n* can be any 64-bit octal value. If *n* is 0, the test reads the real-time clock and uses the value for the initial seed. The default for *n* is O'33. If this option is selected, do not select `+getseed` or `getseed file`.

`stride n` Indicates the stride used to go through memory. *n* is a positive value in the range O'1 through the maximum allowable memory. `stride n` must be less than or equal to `words n`. This option is applicable only to test section 1. The default for *n* is O'1.

`words n` Sets the number of words to be tested in central memory. *n* is a value in the range O'1000 through the maximum allowable memory. All values are rounded down to the nearest O'1000 words. The default for *n* is O'3000.

#### **olcrit Test-specific Options**

The test-specific command options for `olcrit` are as follows:

`+/-ranclus`

Enables or disables random cluster selection. The default is `-ranclus`.

`cluster n`

Selects a specific cluster. The value for *n* can range from 2 through the maximum number of clusters (the maximum number of clusters available is the value `numclstr` returned by the `target(1)` command). The default for *n* is 0, which selects a cluster number randomly. This option must be used to recreate a failure. If `cluster n` is selected, do not select the `+ranclus` option.

`disable ilist`

Deselects specific instructions. Enter *ilist* in the following format:

*n,n,...n*

*n* is the octal value in the *gh* field or *ghijk* field of the specific instruction. If the *gh* field does not specify a unique instruction, use the *ijk* field to deselect a specific instruction. This option overrides the `enable ilist` option and any selected (+) or deselected (-) instruction options.

`+/-eis` This option applies only to the CRAY C90 series and CRAY T90 series. Enables or disables the 005400 CRAY C90 series and CRAY T90 series system instructions.

`enable ilist`

Selects specific instructions. Enter *ilist* in the following format:

*n,n,...,n*

*n* is the octal value in the *gh* field or *ghijk* field of the specific instruction. If the *gh* field does not specify a unique instruction, use the *ijk* field to deselect a specific instruction. This option overrides any selected (+) or deselected (-) instruction options. When a test is run with default values for the +/- instruction options, and the `enable ilist` option is selected, only the instructions specified by the `enable ilist` option are run.

When using the `enable` option to select any of the following instructions, numins *n* should be greater than 1: 34 through 37, 56, 57, 76, 77, 100 through 130, 150 through 153, 176, 177.

`+/-hwbmm`

This option applies only to Cray PVP systems with bit matrix multiply (BMM) hardware. Enables or disables testing of BMM hardware. The default is that the test determines whether or not BMM hardware is present using the result from the `sysconf(2)` system call and performs testing as appropriate.

`+/-isolate`

Enables or disables the error isolation option. The default is `+isolate`.

`isop n` Sets the isolation pass limit to *n* (octal). During isolation, the test repeatedly executes the suspected failing sequence. If the sequence fails, the loop terminates and the test attempts to isolate the sequence further. If the sequence does not fail, the loop terminates after *n* passes, and the test assumes that the error is not in the tested sequence. The default for *n* is O'1000.

`numins n`

Sets the number of instructions to be generated. For `olcrit`, *n* can be any octal value within the range 1 through 2000. The default for *n* is O'200.

`+/-repeat`

Enables or disables the option that repeats the first pass until the test terminates. This option is useful for recreating an error and is usually used with one of the following options: `seed n`, `+getseed`, or `getseed file` (or `+cluster` together with `cluster n`). The default is `-repeat` (the program generates new instructions and data after each pass).

`seed n` Sets the random seed to *n*. *n* can be any 64-bit octal value. If *n* is 0, the test reads the real-time clock and uses the value for the initial seed. The default for *n* is O'33. If this option is selected, do not select `+getseed` or `getseed file`.

`v1 n` Sets the vector length to *n*. *n* can be any octal value in the range 0 through 100 (*n* can be up to O'200 for the CRAY C90 series and CRAY T90 series). If *n* is set to 0, a random value is used to initialize the test. The default for *n* is O'0.

`+/-vload`

Selects or deselects vector instructions for the instruction buffer and, in the case of `-vload`, does not allow loading or saving of the vector registers. `-vload` overrides vector instructions selected by `+vector` and enable *ilist*. The default is `+vload`.

`+/-bits, +/-onezero, +/-random`

Selects or deselects specific data patterns. The default is all data patterns are run. The data patterns are as follows:

<code>bits</code>	Random number of consecutive 1-bits in a word
<code>onezero</code>	Random selection of all 1's or all 0's in a word
<code>random</code>	Random bit generation in a word

`+/-address, +/-ci, +/-cm, +/-fpadd, +/-fpmult, +/-fprecip, +/-int, +/-jump, +/-logical, +/-pop, +/-scalar, +/-shift, +/-shr, +/-vector`

Selects or deselects specific instruction groups. The default is all instruction groups are selected.

### **olcsvc Test-specific Options**

The test-specific command options for `olcsvc` are as follows:

`disable ilist`

Deselects specific instructions. Enter *ilist* in the following format:

*n,n,...,n*

*n* is the octal value in the *gh* field of the specific instruction. This option overrides the `enable ilist` option and any selected (+) or deselected (-) instruction options.

`+/-eis` This option applies only to the CRAY C90 series and CRAY T90 series. Enables or disables the 005400 CRAY C90 series and CRAY T90 series system instructions.

`enable ilist`

Selects specific instructions. Enter *ilist* in the following format:

*n,n,...,n*

*n* is the octal value in the *gh* field of the specific instruction. This option overrides any selected (+) or deselected (-) instruction options. When a test is run with default values for the `+/-` instruction options, and the `enable ilist` option is selected, only the instructions specified by the `enable ilist` option are run.

`+/-isolate`

Enables or disables the error isolation option. The default is `+isolate`.

`isop n` Sets the isolation pass limit to *n* (octal). During isolation, the test repeatedly executes the suspected failing sequence. If the sequence fails, the loop terminates and the test attempts to isolate the sequence further. If the sequence does not fail, the loop terminates after *n* passes, and the test assumes that the error is not in the tested sequence. The default for *n* is O'1000.

- `numpar n`  
Sets the minimum number of parcels of vector instructions to be generated on each pass. *n* can be any octal value in the range 1 through 200. The default for *n* is O'100.
- `+/-repeat`  
Enables or disables the option that repeats the first pass until the test terminates. This option is useful for recreating an error and is usually used with one of the following options: `seed n`, `+getseed`, or `getseed file`. The default is `-repeat` (the program generates new instructions and data after each pass).
- `seed n`  
Sets the random seed to *n*. *n* can be any 64-bit octal value. If *n* is 0, the test reads the real-time clock and uses the value for the initial seed. The default for *n* is O'33. If this option is selected, do not select `+getseed` or `getseed file`.
- `+/-sgci`  
Enables or disables testing of the scatter/gather/compressed index hardware. When allowed to default, the test determines the type of hardware configuration and sets the default value accordingly.
- `v1 n`  
Sets the vector length to *n*. *n* can be any octal value in the range 0 through 100 (*n* can be up to O'200 for the CRAY C90 series and CRAY T90 series). If *n* is set to 0, a random value is used to initialize the test. The default for *n* is O'0.
- `vuse vrlist`  
Lets you specify a vector register to be used in place of those usually generated. This option provides a mechanism for allowing the same instruction and data to be used with different registers and can help you determine whether a failure is related to a register or functional unit.  
*vrlist* is a comma-separated list of up to eight vector register designators. The position of the element determines which vector register is being reassigned.
- `+/-onezero, +/-random, +/-slide`  
Selects or deselects specific data patterns. The default is all data patterns are run. The data patterns are as follows:
- |                      |  |
|----------------------|--|
| <code>onezero</code> | Random selection of all 1's or all 0's in a word.  |
| <code>random</code>  | Random bit generation in a word.   |
| <code>slide</code>   | Random number of consecutive 1's (0's) that slide in either direction through a field of 0's (1's). Consecutive words contain the sliding pattern. |
- `+/-cm, +/-fpadd, +/-fpmult, +/-fprecip, +/-int, +/-logical, +/-pop, +/-shift, +/-lz`  
Selects or deselects specific instruction groups. `lz` applies only to the CRAY C90 series and CRAY T90 series. The default is all instruction groups are run.

**olibuf Test-specific Options**

The test-specific command options for `olibuf` are as follows:

`+/-repeat`

Enables or disables the option that repeats the first pass until the test terminates. This option is useful for recreating an error and is usually used with one of the following options: `seed n`, `+getseed`, or `getseed file`. The default is `-repeat` (the program generates new instructions and data after each pass).

`seed n` Sets the random seed to *n*. *n* can be any 64-bit octal value. If *n* is 0, the test reads the real-time clock and uses the value for the initial seed. The default for *n* is `O'33`. If this option is selected, do not select `+getseed` or `getseed file`.

`+/-bits`, `+/-onezero`, `+/-ozpar`, `+/-random`, `+/-solid`

Selects or deselects specific data patterns. The default is all data patterns are run. The data patterns are as follows:

<code>bits</code>	Random number of consecutive 1-bits are run through the test area
<code>onezero</code>	Random patterns of all 1's or all 0's are run through the test area
<code>ozpar</code>	Random selection of all 1's or all 0's in each parcel are run through the test area
<code>random</code>	Random bit patterns are run through the test area
<code>solid</code>	A random pattern of either all 1's or all 0's is run through each instruction buffer area of the test area with one complement pattern

**olsbt Test-specific Options**

The test-specific command options for `olsbt` are as follows:

`+/-anycpus`

Overrides the `olsbt` restriction on the number of CPUs that can be selected. The default is `-anycpus` (from 2 to 4 CPUs can be selected). It is recommended that the `+anycpus` option be used only on lightly loaded dedicated systems or Cray Research STCO systems.

`cluster n`

Selects a specific cluster. The value for *n* can range from 2 through the maximum number of clusters (the maximum number of clusters available is the value `numclstr` returned by the `target(1)` command). The default for *n* is 0, which selects a cluster number randomly. This option must be used to recreate a failure.

`numins n`

Sets the number of instructions to be generated. *n* can be any octal value within the range 1 through 20. The default for *n* is `O'20`.

`+/-repeat`

Enables or disables the option that repeats the first pass until the test terminates. This option is useful for recreating an error and is usually used with one of the following options: `seed n`, `+getseed`, or `getseed file` (or `cluster n` for `olsbt`). The default is `-repeat` (the program generates new instructions and data after each pass).

`seed n` Sets the random seed to *n*. *n* can be any 64-bit octal value. If *n* is 0, the test reads the real-time clock and uses the value for the initial seed. The default for *n* is `O'33`. If this option is selected, do not select `+getseed` or `getseed file`.



`+/-bits, +/-onezero, +/-random`

Selects or deselects specific data patterns. The default is all data patterns are run. The data patterns are as follows:

<code>bits</code>	Random number of consecutive 1-bits in a word
<code>onezero</code>	Random selection of all 1's or all 0's in a word
<code>random</code>	Random bit generation in a word

## EXAMPLES

The following example executes `olcrit` in CPU c. (To run a sequence of diagnostic tests, use the `runsequence` utility.) See the *Online Maintenance Tools Guide for Cray PVP Systems*, Cray Research publication SD-1012, for information on `runsequence`.

```
olcrit cpu c +getseed cputime 3 > critC.fail
```

The following example executes `olcsvc` in CPU a. This example would be used if `olcsvc` had failed previously without completing instruction isolation. The `getseed` option obtains the seed from the previous `olcsvc` failure, which is stored in `csvcA.fail`. The `+repeat` option will repeat the first pass until the test terminates. The `isop` option tries to isolate a failing instruction sequence up to 10,000 times. The ampersand (&) causes the entire command to execute in the background so that another prompt is immediately displayed and you can continue to use the system.

```
olcsvc cpu a cputime 2:00 getseed csvcA.fail +repeat isop 10000 > csvcA.iso &
```

The following examples illustrate the use of the `+/-waitded` option. In the first example, assume CPU b is dedicated. Because the `+waitded` option is enabled, `olcrit` waits until the dedicated CPU is freed (for however long it takes) and then runs in the CPU.

```
voyager$ olcrit cpu b +waitded
olcrit cpu b +waitded
olcrit started in cpu 1 on Tue Sep 17 17:47:23 1996
CRAY TS MODE
Bit matrix multiply detected.
olcrit reached maximum pass limit with 1000 passes and 0 errors on Tue Sep 17 17:47:25 1996
```

In this example, CPU b is dedicated, but `+waitded` is not specified.

```
voyager$ olcrit cpu b
olcrit cpu b
olcrit not run, dedicated cpu 1 Tue Sep 17 17:46:51 1996
```

**SEE ALSO**

`oldmon(8)` for information on running the online confidence tests in a down CPU environment under the `oldmon` down CPU monitor

`sysconf(2)` for information on the `sysconf` system call, which the `olcrit(8)` test uses to retrieve system implementation information

`target(1)` for information on the maximum number of clusters available (the value `numclstr`) for use with the `olcrit` or `olsbt cluster n` option in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

*Online Maintenance Tools Guide for Cray PVP Systems*, Cray Research publication SD-1012 (This document contains information private to Cray Research, Inc. It can be distributed to non-CRI personnel only with approval of the appropriate Cray manager.)

*OLNET Online Diagnostic Network Communications Program Maintenance Manual for UNICOS*, Cray Research publication SD-1021 (This document contains information private to Cray Research, Inc. It can be distributed to non-CRI personnel only with approval of the appropriate Cray manager.)

*Disk Drive Offload User Guide (Version 2.0)*, publication CDM-1028-000 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)

**NAME**

`oldmon` – Initiates, controls, and monitors the down CPU diagnostic tests

**SYNOPSIS**

```
/etc/oldmon [-d cpulist [-b batchfile]] [-q] [-u cpulist]
```

**IMPLEMENTATION**

Cray PVP systems (except CRAY T90 series and CRAY J90 series)

**DESCRIPTION**

The `oldmon` down CPU diagnostic monitor initiates, controls, and monitors the down CPU diagnostic tests. Two types of down CPU tests run under `oldmon`: confidence tests and maintenance tests. The down CPU confidence tests are online confidence tests that have been converted to run under `oldmon` (offline). The down CPU maintenance tests are derived from the offline diagnostic release (except for the CRAY C90 series).

By default, `oldmon` provides an interactive display that lets you load and execute diagnostic tests and examine the results. The diagnostic tests that `oldmon` can execute are in the `/ce/oldmon` directory.

The following options can be entered in any order.

`-d cpulist` Provides a list of CPUs to be downed. The list of CPUs has the following format:

*n, n, . . . , n*

*n* is a value in one of the following ranges:

*0, 1, 2, . . . , n*

*a, b, c, . . . , x*

By default, no CPUs are downed.

`-b batchfile` Provides the name of the batch input file. This file contains a list of diagnostic tests (and associated test options) to be executed. The tests will be executed in the first CPU specified in the `-d cpulist` option; therefore, the `-d` option is required with the `-b` option. If *batchfile* is `'-'`, `stdin` will be read as the batch input. In batch mode, the interactive interface is not entered.

Batch input has the following format:

*test [param[=]val] . . . [+/-option] . . . [>[>]outfile]*

*test* is the name of the diagnostic test to be executed. For a list of available tests, see the `/ce/oldmon` directory. *param* is a parameter to be set with the corresponding *val*, which can be `cputime`, `maxp`, `time`, or a diagnostic DIB entry. (For a list of diagnostic DIB entries, see the help display for the online version of the offline test you want to run. For example, enter `olcrit help` to obtain the DIB entries for `offcrit`.)

If a test fails when executing in batch mode, the entire memory image will be saved to *outfile*. This file can be loaded into `oldmon` interactively to examine the results at a later time.

- q Exits `oldmon` after processing the command line. This option is useful for using `oldmon` to bring CPUs up or down without entering the interactive user interface.
- u *cpulist* Provides a list of CPUs that have been returned to normal system operations. The list of CPUs has the following format:

*n, n, . . . , n*

*n* is a value in one of the following ranges:

*0, 1, 2, . . . , n*

*a, b, c, . . . , x*

**EXAMPLES**

Example 1: Shows how to start the interactive user interface.

`/etc/oldmon`

Example 2: Shows two different ways to bring down CPU *c* and start the interactive user interface.

`/etc/oldmon -d c`

`/etc/oldmon -d 2`

Example 3: Brings down CPUs *a* and *c* without entering the interactive user interface.

`/etc/oldmon -d a,c -q`

Example 4: Executes the diagnostic tests in the `batch` file as batch input to CPU *d*.

`/etc/oldmon -d d -b batch`

Example 5: Again executes the diagnostic tests in the `batch` file as batch input to CPU *d*. When execution is complete, CPU *d* will be returned to normal system operations.

`/etc/oldmon -d 3 -b batch -u 3`

**EXIT STATUS**

When executing the interactive user interface, the exit code is undefined. In batch mode or with the `-q` option, the following exit codes are defined:

- 0 All diagnostic tests executed successfully.
- 1 An option on the command line or in the batch input file was not valid.
- 2 A diagnostic test in the batch input file failed.
- 3 A diagnostic test in the batch input file terminated abnormally. The termination could be caused by an operand range error, error exit, register parity error, and so on.

**SEE ALSO**

*Online Maintenance Tools Guide for Cray PVP Systems*, Cray Research publication SD-1012. (This document contains information private to Cray Research, Inc. It can be distributed to non-CRI personnel only with approval of the appropriate Cray manager.)

**NAME**

`oldt` – Provides online confidence testing of Cray Research disk drives

**SYNOPSIS**

```
/etc/diag/oldt -d devicename | -f filename [-D dkname] [-F fsize] [-n] [-P fs_prct]
[-b bsize] [-S] [-m maxpass] [-p patterns] [-u userpat] [-s seed] [-t tests] [-C hh:mm:ss]
[-T hh:mm:ss] [-a] [-v] [-q] [-h]
```

**IMPLEMENTATION**

Cray PVP systems

**DESCRIPTION**

The `oldt` command tests Cray Research disk drives by first creating a new file or using an existing user-specified file (`-f filename`) in a specified directory and then performing reads and writes to that file, which resides on the specified device (`-d devicename`). If a miscompare is detected, `oldt` stops execution and displays error information including expected, actual, and difference for the word in error and 20 words before and after the error. Also, the physical location on the device is displayed.

- `-d devicename` Defines the name of the device to be tested (required, unless `-f filename` is used). For example, `0235.4` on an IOS model E (IOS-E) or `ESDI_100` on the CRAY EL series. You can find device names on an IOS-E system or the CRAY EL series by using the `/etc/pddstat` command. Specifying `-d sds` will test the secondary data segment (SDS) area of the SSD, but no actual file is created. The `sds` argument uses the `ssbreak(2)`, `sswrite(2)`, and `ssread(2)` system calls. If SSD is a defined device, it can also be tested, as long as an SSD-resident file system is mounted. By default, the file name created is `/path/oldt_devicename`. If the `-d` option is used with the `-f` option, you can specify the name for the file. (If you also specify a device with the `-d` option, specify only the file name with the `-f filename` option; do not specify a full path name). (You must specify the `-S` option along with the `-d` option to save the file.)
- `-f filename` Specifies the name of the existing file to use for all reads and writes (required, unless `-d devicename` is used). This file will reside on the device to be tested. If this option is not specified, `oldt` creates a file with the default name `/path/oldt_devicename`. When using the `-f` option, specify the full path name (for example, `/tmp/oldt_0235.4`). The first time you run `oldt` to a disk, use the `-d devicename` option. The test creates a file with the above-mentioned type name and tells you where the file was created. If you also specify a device with the `-d devicename` option, specify only the file name with the `-f filename` option (do not specify a full path name). (You must specify the `-S` option with the `-f` option to save the file.)

- D *dskname* Defines the file system name where you want the work file created. */dev/dsk/name* can be found by issuing a `/etc/dmmap -p device` command for the device to which you want to run `oldt`. The `-D` option must be used with the `-d devicename` option. This option lets you create the work file on a particular area of the disk. `oldt` will attempt to create a file only in the file system selected with `-D`. (Do not specify the `/dev/dsk` path for *dskname*.)
- F *fsize* Defines the file size of the work file. *fsize* can be a decimal block value between 1 and the maximum file size. The default is 1 sector. Do not specify `-F` with `-P`.
- n Indicates that the `-F fsize` value should be in decimal tracks rather than decimal blocks. Do not specify `-n` with `-P`.
- P *fs\_prcnt* Defines the percentage of the file system on the selected device to create the work file. The maximum value is 98, which causes the work file being created to be the same size as 98% of the available free space of the partition on the selected device. Do not specify `-P` with `-F` or `-n`.
- b *bsize* Defines the size of the buffer to use for reads and writes to the work file. Specify the value of *bsize* in sectors. The default is 1 sector. The maximum value that may be specified is the number of sectors in 1 track. The larger the value of *bsize*, the more memory will be allocated by `oldt` (3 buffers of *bsize* are allocated during execution).
- S Saves the created work file. By default, the file is removed at test completion. If the device specified by the `-d devicename` option is `sds`, the `-S` option is not valid.
- m *maxpass* Defines the maximum number of passes to run the test. The default is 10 passes. If you specify `-m 0`, `oldt` creates a work file, but it does not run any tests. This feature is useful if you want to create a file on a particular disk (for example, `oldt -d 0235.4 -m 0 -S`).
- p *patterns* Defines the test patterns to run. Available patterns are as follows: `zeros`, `ones`, `chkbrd`, `chkbrdc`, `alt2`, `alt2c`, `alt4`, `alt4c`, and `rdm`. The default is all patterns are run. *patterns* is a comma-separated list (for example, `-p zeros,alt2,alt2c`). The pattern descriptions are as follows:
 

<code>zeros</code>	<code>00000000000000000000000000000000</code>
<code>ones</code>	<code>01777777777777777777777777777777</code>
<code>chkbrd</code>	<code>01252525252525252525252525252525</code>
<code>chkbrdc</code>	<code>005252525252525252525252525252525</code>
<code>alt2</code>	<code>01777777777777777777777777777777</code>
<code>alt2c</code>	<code>00000000000000000000000000000000</code>
<code>alt4</code>	<code>01777770000000000000000000000000</code>
<code>alt4c</code>	<code>00000007777777777777777777777777</code>

- rdm           Set from the real-time clock value or seed (-s)
- u *userpat*   Specifies a user-defined octal data pattern (up to 64 bits) to run.
- s *seed*       Specifies a user-defined decimal seed used for generating random data patterns and read/write tests. The default is the real-time clock value.
- t *tests*      Defines the test sections to run. Available tests are as follows: *sw*, *sr*, *rw*, *rr*, *ss*, *rs*. The default is all test sections are run. (The seek tests are not performed if the device specified by the -d *devicename* option is *sds*.) *tests* is a comma-separated list (for example, -t *sw, sr*). The test descriptions are as follows:
  - sw*           Sequential write test
  - sr*           Sequential read test
  - rw*           Random write test
  - rr*           Random read test
  - ss*           Sequential seek test
  - rs*           Random seek test
- C *hh:mm:ss*   Defines the amount of CPU time the test will run. If this option is used in conjunction with -T, the first option to be satisfied will stop the test. The -m *maxpass* value will be overridden.
- T *hh:mm:ss*   Defines the amount of wall-clock time the test will run. If this option is used in conjunction with -C, the first option to be satisfied will stop the test. The -m *maxpass* value will be overridden.
- a            Performs asynchronous reads and writes rather than the default synchronous reads and writes.
- v            (Verbose) Echoes all file creation information, device information, and a pass counter to `stdout` and `stderr` as the test is running. The default is to echo only error information.
- q            Performs a quick test. If you use the -q option, you need only specify the device (-d) or the file (-f) with -q (for example, `oldt -d 0235.4 -q`). When -q is specified, only the *sw* and *sr* tests, the *ones* pattern, and 1 pass are run.
- h            Displays the help screen.

**NOTES**

If this command is installed with the default privilege assignment list (PAL), you must have an active *secadm*, *sysadm*, or *diagadm* category to use this command.



oldt depends on a character special device node existing in the /dev/ddd or dev/ssdd (SSD) directory in order to work.

## EXAMPLES

Example 1: Creates a work file on device 0235.4 in the file system /dev/dsk/tmp\_40 (-D tmp\_40) with the file name of myfile (-f myfile) and a default size of 1 sector. The test runs for 100 passes (-m 100) of the default tests and patterns (all), deletes the file (no -S), and then exits.

```
oldt -d 0235.4 -D tmp_40 -f myfile -m 100
```

Example 2: Creates a work file with a size of 100 blocks (-F 100) on device 0233.2, runs the default (10) passes of the default tests and patterns (all), saves the work file (-S), and then exits. All information is displayed during test execution (-v).

```
oldt -d 0233.2 -F 100 -S -v
```

Example 3: Uses the work file created in the previous example (-f /tmp/oldt\_0233.2), increases the file size to 100 tracks (-F 100 -n), and runs the sequential write and read tests (-t sw,sr) using only the one/zero, one/zero-complement and user-selected patterns (-p chkbrd,chkbrdc -u 1234). After execution is completed, the work file is deleted (no -S).

```
oldt -f /tmp/oldt_0233.2 -F 100 -n -t sw,sr -p chkbrd,chkbrdc -u 1234
```

Example 4: Creates a work file named test1 (-f test1) on device 2214.6. test1 has a size of 20 blocks (-F 20), uses the default values for all other options, and saves the file at test completion (-S).

```
oldt -d 2214.6 -f test1 -F 20 -S
```

Example 5: Creates a work file with the default size of 1 sector on device 3226.0, runs the sw and sr tests using the ones pattern for 1 pass only, deletes the work file, and then exits.

```
oldt -d 3226 -q
```

Example 6: Tests the SDS space of the SSD. It allocates 2 units of 128 blocks each (-F 200), runs all tests and patterns for 30 passes (-m 30), and then exits.

```
oldt -d sds -F 200 -m 30 -v
```

## SEE ALSO

olhpa(8) for information on the system error log formatter

*Online Maintenance Tools Guide for Cray PVP Systems*, Cray Research publication SD-1012. (This document contains information private to Cray Research, Inc. It can be distributed to non-CRI personnel only with approval of the appropriate Cray manager.)

**NAME**

`olhpa` – Formats UNICOS hardware errors logged by `errdemon(8)`

**SYNOPSIS**

```
/etc/diag/olhpa [-h] [-l] [-q] [-g [d] [m] n] [-cdnimptv] [-C arg] [-D arg] [-N arg]
[-I arg] [-M arg] [-P arg] [-T arg] [-V arg] [-s start] [-e end] [-r] [-F file] [-A] [-z]
[errfile(s)]
```

**IMPLEMENTATION**

All Model E based Cray Research systems

**DESCRIPTION**

The `olhpa` command formats hardware errors generated by UNICOS device drivers and logged by `errdemon(8)`. Error types supported are local disk, network disk, tape, SSD, central memory, register parity, IOS communication (both COMM and HIPPI errors), and Cray MPP local memory. (SSD, register parity, and Cray MPP reports are not supported on the CRAY EL series.)

NOTE: `olhpa` is not supported on GigaRing based Cray Research systems. Instead, error information can be viewed with `errpt(8)` and the `sysmon(8)` GUI interface on the Cray system workstation (SWS). For more information, see the `errpt(8)` and `sysmon(8)` man pages and the *SWS-ION Administration and Operations Guide*, publication SG-2204.

Memory location on the CRAY J90 series is reported as three values (for example, 01/01/05) corresponding to CRAY J90s with a 2X2, 4X4, and 8X8 backplane (respectively). The correct memory location value should be selected based on the hardware configuration of the machine.

Errors are grouped by hardware type and are displayed in the format specified. Possible formats include quick, regular, long, custom, and graph. The quick report provides a summary of errors for each hardware type. The regular report displays several key fields of the error record on one line. The long report displays all available error fields and is formatted in a paragraph form. The custom report prints all user-specified fields on the same line. The graph report displays a bar graph of errors in intervals specified by hours, days, or minutes.

By default, `olhpa` reads error records from `/usr/adm/errfile`. You can override this by specifying error file(s) on the command line. By default, `olhpa` uses the current execution time for the ending time and 30 days minus the ending time for the start time. Only errors within the given time interval are displayed.

The `olhpa` command accepts the following options:

- h      Displays the command synopsis.
- l      Displays a long version of an error report, which contains the information as it was found in `errpt(8)`. A long report created with the `-l` option should look just like an `errpt` report.
- q      Displays only the summary information.

- g [d] [m] *n*  
Displays a bar graph in which *n* is the interval in hours or (optionally) days or minutes (d or m).
- c Displays a report of all register parity errors in the specified *errfile(s)* and between the specified times. This option is not valid for the CRAY EL series.
- d Displays a report of all local disk errors in the specified *errfile(s)* and between the specified times.
- n Displays a report of all network disk errors in the specified *errfile(s)* and between the specified times.
- i Displays a report of all IOS communication errors in the specified *errfile(s)* and between the specified times.
- m Displays a report of all memory errors in the specified *errfile(s)* and between the specified times.
- p This option is available at Cray MPP sites only. Calls the `mpphpa(8)` command, which displays a report of all Cray MPP errors in the specified *errfile(s)* and between the specified times. This option is not valid for the CRAY EL series.
- t Displays a report of all tape errors in the specified *errfile(s)* and between the specified times.
- v Displays a report of all SSD errors in the specified *errfile(s)* and between the specified times. This option is not valid for the CRAY EL series.
- C *arg*, -D *arg*, -N *arg*, -I *arg*, -M *arg*, -P *arg*, -T *arg*, -V *arg*  
Displays register parity, local disk, network disk, IOS communication, memory, Cray MPP local memory, tape, and SSD errors, respectively. (SSD, register parity, and Cray MPP local memory reports are not supported on the CRAY EL series.) The -P option is available at Cray MPP sites only. This option calls the `mpphpa(8)` command, which displays a report of all Cray MPP errors in the specified *errfile(s)* and between the specified times. The required argument can be one of the following:
 

S , <i>field</i> [!]= <i>value</i> [ , <i>field</i> [!]= <i>value</i> ]	Searches.
P[ , +] , <i>field</i> [ , <i>field</i> ]	Prints fields.
H	Displays an associated help menu.
- s *start* *start* can be one of the following: *hh:mm* , *MM/DD/YY* , *hh:mm* , or *MM/DD/YY*.
- e *end* *end* can be one of the following: *hh:mm* , *MM/DD/YY* , *hh:mm* , or *MM/DD/YY*.
- r Displays the error records in raw mode.
- F *file* Stores all binary records that are displayed to the file specified by *file*. *file* is in the standard *errfile* format and can be read by `olhpa` or `errpt(8)`.
- A Abbreviates the IOS-E disk summary to display only recovered, corrected, unrecovered, and recovery off disk error totals.
- z Suppresses printing of header and summary information if the graph (-g) or quick (-q) option is not specified.

*errfile(s)* Specifies the *errfile(s)* to be read. *errfile* can be any file that was produced by `errdemon(8)` and read by `errpt(8)`.

## NOTES

If this command is installed with the default privilege assignment list (PAL), you must have an active `secadm`, `sysadm`, or `diagadm` category to use this command.

## EXAMPLES

Example 1: Displays in the regular format all tape errors in the `/usr/adm/errfile` error file from the past 30 days.

```
olhpa -t
```

Example 2: Displays in the long report format all disk, tape, and IOS communication errors from a starting date of 2/3/93 until the present date and time. All error records are read from the `/usr/adm/errfile1` error file.

```
olhpa -s 2/3/93 -ldti /usr/adm/errfile1
```

Example 3: Displays for the past 30 days and in the long format all disk errors for which the device type is DD-49. Error records are read from all files in the format `/usr/adm/errfile*` (that is, `errfile1`, `errfile2`, and so on).

```
olhpa -lDS,dt=dd49 /usr/adm/errfile*
```

Example 4: Displays for the past 30 days and in the regular format all disk errors for which the device type is `dd60` and the cylinder value is not 1572. Error records are read from the `/usr/adm/errfile` error file.

```
olhpa -DS,dt=dd60,cyl!=1572
```

## SEE ALSO

`errdemon(8)` for information on the UNICOS error-logging daemon  
`errpt(8)` for information on the `errpt` command, which also processes data logged by `errdemon`

UNICOS MAX:

`mpphpa(8)` for information on formatting hardware errors logged by `mppd(8)`, the Cray MPP daemon

*Online Maintenance Tools Guide for Cray PVP Systems*, Cray Research publication SD-1012. (This document contains information private to Cray Research, Inc. It can be distributed to non-CRI personnel only with approval of the appropriate Cray manager.)

*CRAY T3D Diagnostic Reference Manual*, publication CDM-0601-000 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)

**NAME**

`olnet` – Detects and isolates problems in networks associated with Cray Research systems

**SYNOPSIS**

```
/etc/diag/olnet [-a input-channel-timeout] [-b input-short-burst-flag] [-c input-channel-state-flag]
[-d input-drop-connection-flag] [-e input-send-i-field-flag] [-f input-i-field-mask]
[-g input-software-timeout] [-h input-header-usage-flag] [-i input-i-field-value] [-l message-length]
[-m messages-per-pass] [-n] [-p pass-count] [-r remote-address] [-A output-channel-timeout]
[-B output-short-burst-flag] [-C output-channel-state-flag] [-D output-drop-connection-flag]
[-E output-send-i-field-flag] [-F output-i-field-mask] [-G output-software-timeout]
[-H output-header-usage-flag] [-I output-i-field-value] [-L] [-M test-mode] [-N errorfile]
[-P pattern-type] [-R messages/ack-ratio] [-S SSD-address] [-T tracefile] test-name input-device-name
[output-device-name]

/etc/diag/olnet [comma-separated list of OLNET menu choices]
```

**IMPLEMENTATION**

All Cray Research Model E systems except CRAY J90 series and CRAY EL series

**DESCRIPTION**

The OLNET online diagnostic network communications program provides diagnostic testing for the following systems and hardware associated with Cray Research systems:

System/OS	Hardware	Test Configuration(s)
UNICOS IOS-E	Cray LOSP (CCA1)	Cable loopback
	IBM FEI-1 (standard and data-streaming)	End-to-end (OLNET active/passive) with IBM MVS and VM Data-streaming FEI internal and external loopback
	FEI-3 (VME)	Cray cable loopback End-to-end (OLNET active/passive)
	HIPPI	Cable loopback Software (PS32) loopback Hardware loopback through Ultra device End-to-end (OLNET active/passive)
	NSC A series adapters	Remote adapter loopback End-to-end (OLNET active/passive)

System/OS	Hardware	Test Configuration(s)
	NSC N (DX) series HYPERchannel or FDDI adapters	Local adapter loopback Remote adapter loopback End-to-end (OLNET active/passive) Remote adapter statistics
	FDDI (FCA-1)	End-to-end (OLNET active/passive) Ring loopback IOS software echo
	FDR-4	Local FDR loopback Remote FDR loopback Address loopback Data error injection Address/block length error injection Control error injection Synchronization error injection Framing pulse error injection Block length error injection High bias error injection Low lamp error injection SSD loopback
Cray MPP	n/a	Software loopback Write to MPP Read from MPP Type statistics IOG statistics IOG echo
IBM/MVS IBM/VM	FEI-1 (standard and data- streaming)  NSC A series adapters	External FEI loopback End-to-end (OLNET active/passive)  Remote adapter loopback End-to-end (OLNET active/passive)
	NSC N (DX) series HYPERchannel or FDDI adapters	Local adapter loopback Remote adapter loopback End-to-end (OLNET active/passive)
Sun Workstation	NSC A or NSC N (DX) series HYPERchannel or FDDI adapters	Local adapter loopback Remote adapter loopback End-to-end (OLNET active/passive)

System/OS	Hardware	Test Configuration(s)
SUN	VME	End-to-end (OLNET active/passive)
OWS/MWS Workstation	FEI-3	Cable loopback

For a detailed description of supported systems and OLNET test configurations, see the "Supported Configurations" section in the *OLNET Online Diagnostic Network Communications Program Maintenance Manual for UNICOS*, Cray Research publication SD-1021.

For information on accessing and building non-CRI source files, see the "OLNET Build Procedures" section in the *OLNET Online Diagnostic Network Communications Program Maintenance Manual for UNICOS*, Cray Research publication SD-1021.

The OLNET executables are located as follows:

System	Directory/Location
UNICOS	/ce/bin
Sun/UNIX OWS/MWS	/home/ows/cri/smarte/bin/olnet, where <i>ows</i> is the name of the OWS, for example, /home/ows1601/cri/smarte/bin/olnet
Sun/UNIX Workstation	Determined by installer
IBM/MVS IBM/VM	Determined by installer

## OPTIONS

The `olnet` program accepts the following options:

`-a input-channel-timeout`

Specifies the channel time-out value for the input HIPPI channel. HIPPI test only.

`-b input-short-burst-flag`

Specifies whether the short burst flag should be turned on. Valid values are YES or NO. HIPPI test only. IOS model E only.

`-c input-channel-state-flag`

Specifies the channel state of the input HIPPI channel. Valid values are UP or DOWN. HIPPI test only.

- d *input-drop-connection-flag*  
Specifies if connections should be dropped between data transfers for the input HIPPI channel. If enabled, the connection will be broken down and rebuilt for each data transfer. If disabled, the connection will stay active until the channel is closed. Valid values are ENABLED or DISABLED. HIPPI test only.
- e *input-send-i-field-flag*  
Valid values are ENABLED or DISABLED. HIPPI test only.
- f *input-i-field-mask*  
The value should be a 32-bit hexadecimal number. HIPPI test only.
- g *input-software-timeout*  
Specifies the software time-out value for the input HIPPI channel. HIPPI test only.
- h *input-header-usage-flag*  
Valid values are ENABLED or DISABLED. HIPPI test only.
- i *input-i-field-value*  
Specifies the value to be used for the I-field for the input HIPPI channel. The value should be a 32-bit hexadecimal number. HIPPI test only.
- l *message-length*  
Specifies the length of data to be sent or received. The *message-length* value is test-dependent but can be set to RN (random). The default is 100.
- m *messages-per-pass*  
Specifies how many messages are to be sent for each pass of the chosen test. The *messages-per-pass* value can range from 1 through 1000000. The default is 10.
- n  
Specifies that `olnet` should continue on error.
- p *pass-count*  
Specifies how many passes are to be performed of the chosen test. The *pass-count* value can range from 1 through 1000000. The default is 1.
- r *remote-address*  
Specifies the source address (the machine on which OLNET is currently running). The *remote-address* value is dependent on *test-name* and is valid only for the following tests:
  - FDT The value should be either the 48-bit IEEE Ethernet-style address or a machine name from the `/etc/ethers` file. The default is 0:0:0:0:0:0.
  - HI The value should be a decimal number for the remote logical path.
  - NT The value should be the 16-bit hexadecimal HYPERchannel adapter address. The default is 0000.
  - VT The value should be the 8-bit hexadecimal path number. The default is 0000.
- A *output-channel-timeout*  
Specifies the channel time-out value for the output HIPPI channel. HIPPI test only.



- B *output-short-burst-flag*  
Specifies whether the short burst flag should be turned on for the output HIPPI channel. Valid values are YES or NO. HIPPI test only. IOS model E only.
- C *output-channel-state-flag*  
Specifies the channel state of the input HIPPI channel. Valid values are UP or DOWN. HIPPI test only.
- D *output-drop-connection-flag*  
Specifies if connections should be dropped between data transfers for the output HIPPI channel. If enabled, the connection will be broken down and rebuilt for each data transfer. If disabled, the connection will stay active until the channel is closed. Valid values are ENABLED or DISABLED. HIPPI test only.
- E *output-send-i-field-flag*  
Valid values are ENABLED or DISABLED. HIPPI test only.
- F *output-i-field-mask*  
The value should be a 32-bit hexadecimal number. HIPPI test only.
- G *output-software-timeout*  
Specifies the software time-out value for the output HIPPI channel. HIPPI test only.
- H *output-header-usage-flag*  
Valid values are ENABLED or DISABLED. HIPPI test only.
- I *output-i-field-value*  
Specifies the value to be used for the I-field for the output HIPPI channel data transfer. The value should be a 32-bit hexadecimal number. HIPPI test only.
- L  
Specifies that OLNET should be locked into memory.
- M *test-mode*  
Specifies the test mode to use. Valid arguments are as follows:
  - AA Asynchronous active mode
  - AEI Address/block length error injection; FDR test only
  - AL Address loopback; FDR test only
  - AM Active mode
  - AP Asynchronous passive mode
  - CC Control cable test; VME test only
  - CL LOSP cable loopback; NSC, VME, FEI, and FEI-station tests only
  - BLEI Block length error injection; FDR test only
  - CEI Control error injection; FDR test only
  - DC Disable disconnect interrupt; VME test only

DEI Data error injection; FDR test only  
 DISEDB DS FEI external double-byte loopback; FEI, and FEI-station tests only  
 DISESB DS FEI external single-byte loopback; FEI, and FEI-station tests only  
 DISIDB DS FEI internal double-byte loopback; FEI, and FEI-station tests only  
 DISISB DS FEI internal single-byte loopback; FEI, and FEI-station tests only  
 DL Display driver statistics; VME and NSC tests only  
 ECHO IOS software loopback; FDDI test only  
 FPEI Framing pulse error injection; FDR test only  
 FY fymc test; VME test only  
 HBFI High bias error injection; FDR test only  
 HM HYPERchannel map; NSC test only  
 IO I/O master clear; VME test only  
 IOGE IOG echo; MPP test only  
 IS Read MPP IOG statistics; MPP test only  
 LBK Loopback; FDDI, HIPPI, FDR, and MPP tests only  
 LL Local loopback; NSC and FDR tests only  
 LLEI Low lamp error injection; FDR test only  
 MC Master clear; VME test only  
 MPPR Read from the MPP; MPP test only  
 MPPW Write to the MPP; MPP test only  
 PL Program loop  
 PM Passive mode  
 RL Remote loopback; NSC and FDR tests only  
 RP Read single adapter profile; NSC test only  
 RPM Map adapter profiles; NSC test only  
 RS Remote statistics; NSC test only  
 S Local statistics; NSC test only  
 SE IOS software echo; VME test only  
 SEI Synchronization error injection; FDR test only  
 VR Dump VME registers; VME test only  
 WE WIN echo; VME test only

XM Xmapping routine; NSC test only  
 YS Read MPP YPE statistics; MPP test only

The default is AM.

**-N *errorfile***

Specifies the name of the file to which `olnet` will write error output.

**-P *pattern-type***

Specifies the data pattern type to use for the chosen test. Valid arguments are as follows:

AD Address  
 AO All ones  
 AP All patterns  
 AZ All zeros  
 BT Bits  
 RN Random  
 SO Sliding ones  
 SZ Sliding zeros

The default is AD. The *pattern-type* value is not case-sensitive.

**-R *messages/ack-ratio***

When `-M test-mode` is set to AA or AP only, specifies the ratio of messages sent to acknowledgments received. The *messages/ack-ratio* value can be of the form *mm:1*, *mm:0*, or *mm:RN*. The default is 3:1.

**-S *SSD-address***

Specifies the SSD address to be used with the address loopback and SSD loopback tests. The default is 0.

**-T *tracefile***

Specifies the name of the file to be created that will contain OLNET trace information. The default is not to create a trace file.

***test-name***

Specifies the kind of device being tested. Valid values are as follows:

FDR4 FDR-4 test  
 FDT FDDI test  
 FT FEI test  
 HI HIPPI test  
 MPP Cray MPP test  
 NT NSC test

VT VME test

The *test-name* value is not case-sensitive.

*input-device-name*

Specifies the name of the character-special file to be used for the test. If *output-device-name* is used, *input-device-name* specifies the name of the character-special file to be used for input (HI only). The *input-device-name* value is case-sensitive.

*output-device-name*

If specified, *output-device-name* specifies the name of the character-special file to be used for output (HI only). The *output-device-name* value is case-sensitive.

## NOTES

If this command is installed with the default privilege assignment list (PAL), you must have an active *secadm*, *sysadm*, or *diagadm* category to use this command. If the *PRIV\_SU* configuration option is enabled, you must be root to use this command.

## SPECIAL NOTES ON THE FDR-4 TESTS

The FDR-4 test suite was designed to diagnose the components of the shared file system configuration. Those components are: the local and remote FDR-4 fiber optic VHISP channel adapters, the MPX-24 VHISP channel multiplexer, and one or more SSDs.

When an SSD is configured into a system, it is sliced up in much the same way as a traditional disk device. Nodes in the */dev/pdd* directory can access the entire SSD address range and are referred to as *control slices*. Nodes in the */dev/ssdd* directory are configured to access only specific ranges of SSD addresses and are referred to as *noncontrol slices*.

Kernel support for diagnostic access to the SSD VHISP channels makes no attempt to differentiate a diagnostic request from a traditional SSD I/O operation. The only validation the kernel does is to make sure the request does not attempt to access addresses outside the valid range for the slice. However, the error injection tests are set up by turning on some high-order bits of the SSD address register. The outcome is that, depending on the size of slice chosen, the kernel may not allow some of the error injection tests to be executed. For that reason, the error injection tests are restricted to control slices that will always have a large enough address range to accommodate SSD addresses set by OLNET.

With the exception of the SSD loopback test, the hardware forces all transfers to be four SSD blocks in size. An SSD block is sixty-four 64-bit words. For consistency, the SSD loopback test also uses this size. Also, even though the address loopback test allows you to specify a block length value, the transfer will be for four SSD blocks.

Following is a brief description of each test that OLNET is capable of performing:

Local FDR loopback

Available to either kind of slice. Data is written to, and then read from, the local FDR module and compared.

**Remote FDR loopback**

Available to either kind of slice. Data is written to, and then read from, the remote FDR module and compared.

**Address loopback**

Available to either kind of slice. Data is read from the MPX module and compared. The data returned is a combination of the SSD address and block length you specify. By specifying various SSD addresses and block lengths, you can test all bit combinations for those two fields.

**Address/block length error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the maintenance workstation (MWS).

**Data error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

**Control error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

**Synchronization error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

**Framing pulse error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

**Block length error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

**High bias error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

**Low lamp error injection**

Available only when using a control slice. You can choose which fiber the error will be injected into as well as whether the error is injected at the local or remote FDR. The error will be detected by the MWS.

SSD loopback

Available only when using a noncontrol slice. Data is written to, then read from, the SSD address specified. This is potentially the most dangerous test because SSD memory is directly manipulated. Because of the risk of corrupting customer data, this test is not made available when using a control slice. Typically, a noncontrol slice is configured for diagnostic use.

EXAMPLES

OLNET is intended to be used as an interactive menu-driven utility, but it can also be executed from the command line.

To initiate OLNET execution, enter /ce/bin/olnet from the command line. The main menu is displayed as follows (this example shows a menu from a Cray PVP system; the menu may vary, depending on system type):

```

OLNET A.1 MAIN MENU
-----

YOUR SYSTEM:  NAME = sn1703c, NODE NAME = cool, RELEASE = 8.1.0ac,
              VERSION = d81.13, MACHINE = CRAY Y-MP

COMMAND      DESCRIPTION
-----
FDT - Call the FDDI test.
FT - Call the FEI test.
HI - Call the HIPPI test.
NT - Call the NSC test.
VT - Call the VME test.
FDR4 - Call the FDR-4 test.
MPP - Call the MPP test.
HELP - Get HELP information about this menu.
PLK - Enable/disable PROGRAM LOCK.  Lock is: DISABLED
QT - Quit OLNET.

```

OLNET IS A PROPRIETARY PRODUCT OF CRAY RESEARCH INC.

Enter a command:

In a typical OLNET menu, the leftmost column contains a command, followed by a brief command description, followed by an associated value (if applicable).

From the preceding menu, you can select the NSC test by entering the NT command, followed by pressing the <RETURN> key. The NSC menu is displayed as follows:

```

***** NSC TEST INITIAL MENU *****

NSC INITIAL MENU COMMANDS          Current Value(if applicable)
-----
DV - Device path -----> undefined

DPM - NSC Device Path Menu.

HELP - Get HELP information about this menu.

TMM - Select the OLNET NSC Test Mode Menu.

RT - Return to the OLNET Main menu.

```

Enter a command:

You are now ready to execute the OLNET NSC test.

In the following example, OLNET is executed without using the built-in menu system.

The configuration of machine A is as follows: the device /dev/comm/n00-32/1p04 is an NSC HYPERchannel adapter with HYPERchannel address BF00. The logical path is 04, by convention. Machine A will be the passive mode side.

The configuration of machine B is as follows: the device /dev/comm/n00-32/1p0a is an NSC HYPERchannel adapter with HYPERchannel address 6A00. The logical path is 0a, by convention. Machine B will be the active mode side.

Command line for machine A:

```
olnet -p 100 -l 1024 -M pm 6A0A nt /dev/comm/n00-32/1p04
```

Command line for machine B:

```
olnet -p 100 -l 1024 -M am BF04 nt /dev/comm/n00-32/1p0a
```

For detailed information on test execution and command descriptions, see the *OLNET Online Diagnostic Network Communications Program Maintenance Manual for UNICOS*, Cray Research publication SD-1021.

**SEE ALSO**

fddi(4), hippi(4), hy(4), np(4), vme(4) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

*OLNET Online Diagnostic Network Communications Program Maintenance Manual for UNICOS*, Cray Research publication SD-1021. (This document contains information private to Cray Research, Inc. It can be distributed to non-CRI personnel only with approval of the appropriate Cray manager.)

*FDR-4/MPX-24 Engineering Note*, publication PRN-0917 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)



**NAME**

`oper`, `screen` – Invokes the operator display manager

**SYNOPSIS**

`/usr/lib/msg/oper [-r rate] [command [arguments]]`

`/etc/screen [-r rate] [command [arguments]]`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

`oper` and `screen` are functionally identical, except where noted.

The `oper` command displays informative messages and action messages and runs other commands as refresh displays. Display commands, such as `who(1)`, `ps(1)`, `infd(8)`, and `msgd(8)`, are defined in a configuration file, as are commands such as `vi(1)` and `ex(1)`, which require full control of the screen.

The screen is divided into two parts. The bottom three lines are used for input and for running non-display commands. The remainder of the screen is used for refresh displays. When in `oper`, you can use the top line of the display to notify the operator of something that requires attention.

The valid command-line options and arguments are as follows:

`-r rate` Refresh rate in seconds. The default is 10 seconds.

`command [arguments]`

Name of a *command* and any *arguments* to the *command*.

The following commands affect the `oper` screen:

`ref time` Sets the refresh rate for the refresh display to a specified length of *time* (in seconds). The minimum refresh rate is 1 second.

`redraw` Redraws the screen.

`snap file` Sends a copy of the screen to *file*.

`exit`

`ex` Exits from `oper`.

`?` Help.

`-` Displays the previous page.

`+`

`space bar` Displays the next page.

The following commands display informative and action messages when in `oper`:

- `infd`            Displays informative messages. See `infd(8)` and `msgd(8)` for more information.
- `msgd`            Displays action messages, such as tape mount messages. See `msgd(8)` for more information.

The following command lets you respond to an action message when in `oper`:

- `rep msg_number [reply_string]`  
 Allows an operator to respond to an action message, such as a tape mount message. See `rep(8)` for more information.

## NOTES

If this command is installed with a privilege assignment list (PAL), a user who is assigned the following privilege text upon execution of this command is allowed to perform the action shown:

Privilege Text	Action
<code>both</code>	Messages are not subject to security label restrictions.

If this command is installed with a PAL, a user with one of the following active categories is allowed to perform the action shown:

Active Category	Action
<code>system, secadm, sysadm, sysops</code>	Allowed to use this command to view all messages.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command to view all messages.

## FILES

<code>\$HOME/.operrc</code>	Optional configuration file
<code>/usr/lib/oper.rc</code>	Default configuration file

## SEE ALSO

`infd(8)`, `msgd(8)`, `msgdaemon(8)`, `msgdstop(8)`, `rep(8)`

`msgi(1)`, `msgr(1)`, `privtext(1)` in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011

*Tape Subsystem User's Guide*, Cray Research publication SG-2051

**NAME**

`ospf_monitor` – Monitors OSPF gateways

**SYNOPSIS**

`ospf_monitor mon_db_file`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ospf_monitor` command queries open shortest path first (OSPF) routers. The `ospf_monitor` command operates in interactive mode. It allows the user to query the various OSPF routers to provide detailed information on input/output (I/O) statistics, error logs, link-state databases, autonomous system (AS) external databases, the OSPF routing table, configured OSPF interfaces, and OSPF neighbors.

The `ospf_monitor` command accepts the following option:

`mon_db_file` Specifies the complete path name of a database composed of records configuring destinations for `ospf_monitor` remote commands. Each destination record is a single-line entry that lists the destination Internet Protocol (IP) address, the destination host name, and an OSPF authentication key (if authentication is activated by the destination). Since authentication keys may be present in the destination records, it is recommended that general access to this database be restricted.

See RFC 1583 (OSPF Specification, version 2) for details about OSPF database and packet formats.

**Commands**

When entering interactive mode, `ospf_monitor` presents the following prompt:

```
[ # ] dest command params >
```

You can enter any of the `ospf_monitor` interactive commands at this prompt. The command line length must be less than 200 characters. Interactive commands can be interrupted at any time using a keyboard interrupt. `ospf_monitor` uses both local and remote commands.

**Local commands**

The following list describes the `ospf_monitor` local commands:

- ? Displays all local commands and their functions.
- ?R Displays all remote commands and their functions.
- d Displays all configured destinations. This command displays *dest\_index*, the IP address, and the host name of all potential `ospf_monitor` command destinations configured in *mon\_db\_file*.
- h Displays the command history buffer showing the last 30 interactive commands.
- x Exits the `ospf_monitor` program.

@ *remote\_command*

Sends *remote\_command* to the same (previous) destination.

@*dest\_index remote\_command*

Sends *remote\_command* to the configured destination *dest\_index*.

F *filename* Sends all `ospf_monitor` output to *filename*.

S Sends all `ospf_monitor` output to `stdout`.

### Remote commands

The following list describes the `ospf_monitor` remote commands:

a *area\_id type ls\_id adv\_rtr*

Displays link-state advertisement. The *area\_id* parameter specifies the OSPF area for which the query is directed. The *ls\_id* parameter specifies the IP address of the link-state advertisement, either network or host, depending on the type of request. The *adv\_rtr* parameter specifies the router ID of the router which originated this link-state advertisement. The *type* parameter specifies the type of advertisement to request and can have one of the following values:

- 1 Requests the router links advertisements. They describe the collected states of the router's interfaces. For this type of request, set the *ls\_id* field to the router ID of the originating router.
- 2 Requests the network links advertisements. They describe the set of routers attached to the network. For this type of request, set the *ls\_id* field to the IP interface address of the network's designated router.
- 3 Requests the summary link advertisements describing routes to networks. They describe inter-area routes, and enable the condensing of routing information at area borders. For this type of request, set the *ls\_id* field to the IP address of the destination network.
- 4 Requests the summary link advertisements describing routes to AS boundary routers. They describe inter-area routes, and enable the condensing of routing information at area borders. For this type of request, set the *ls\_id* field to the router ID of the described AS boundary router.
- 5 Requests the AS external (ASE) link advertisements. They describe routes to destinations external to the AS. For this type of request, set the *ls\_id* field to the IP address of the destination network.

c Displays a cumulative log. This log includes I/O statistics for monitor request, hello, data base description, link-state request, link-state update, and link-state acknowledgment packets. Area statistics describe the total number of routing neighbors and number of active OSPF interfaces. Routing table statistics are summarized and reported as the number of intra-area routes, inter-area routes, and AS external database entries.

e Displays cumulative errors. This log reports the various error conditions that can occur between OSPF routing neighbors and shows the number of occurrences for each.

- l *retrans* Displays the link-state database (except for ASE link advertisements). This table describes the routers and networks making up the AS. If *retrans* is specified and not 0, the retransmit list of neighbors present in the link-state database is displayed.
- A *retrans* Displays the AS external database entries. This table reports the advertising router, forwarding address, age, length, sequence number, type, and metric for each AS external route. If *retrans* is specified and not 0, the retransmit list of neighbors present in the ASE database is displayed.
- o Displays the OSPF routing table. This table reports the AS border routes, area border routes, summary AS border routes, and networks currently managed using OSPF.
- I Displays all interfaces. This report shows all interfaces configured for OSPF. Information reported includes the area, interface IP address, interface type, interface state, cost, priority, and the IP address of the designated router (DR) and backup designated router (BDR) for the network.
- N Displays all OSPF routing neighbors. Information reported includes the area, local interface address, router ID, neighbor IP address, state, and mode.

## SEE ALSO

gated(8), gdc(8)

gated-config(5) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

RFC 1583 (OSPF Specification, version 2)

## COPYRIGHT INFORMATION

This package and associated documentation is Copyright (c) 1990,1991,1992,1993,1994 Cornell University, all rights reserved. This software contains code that is Copyright (c) 1988 Regents of the University of California, all rights reserved. This package contains code that is Copyright (c) 1989, 1990, 1991 The University of Maryland, College Park, Maryland, all rights reserved.

GateD is maintained and developed by Cornell University and its collaborators.

**NAME**

`pac` – Collects and displays printer/plotter accounting information

**SYNOPSIS**

```
/etc/pac [-c] [-m] [-Pprinter] [-pprice] [-r] [-s] [names]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `pac` command reads the printer/plotter accounting files, accumulates the number of pages or feet (for raster devices) of paper consumed by each user, and prints out how much each user consumed in pages or feet and dollars. By default, statistics are printed for every user who has used any paper.

The `pac` command accepts the following options:

- `-c`           Sorts the output by cost. By default, the output is sorted alphabetically by name.
- `-m`           Ignores the host name in the accounting file. This lets a user on multiple machines have all of his or her printing charges grouped together.
- `-Pprinter`   Causes accounting to be done for the specified *printer*. If you omit this option, accounting is done for the default printer (site-dependent) or for the printer that is the `PRINTER` environment variable value.
- `-pprice`      Causes the value *price* to be used for the cost in dollars, rather than the default value of 0.02 or the price specified in `/etc/printcap`.
- `-r`           Reverses the sorting order.
- `-s`           Causes the accounting information to be summarized on the summary accounting file. This summarization is necessary because on a busy system, the accounting file can grow by several lines per day.
- names*       Limits reported statistics to the user names specified.

**FILES**

```
/etc/printcap   Printer capability database
/usr/adm/?acct   Raw accounting files
/usr/adm/?_sum   Summary accounting files
```

**SEE ALSO**

`printcap(5)` in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`panic` – Sets or clears the system user panic flag

**SYNOPSIS**

```
/etc/panic set
/etc/panic clear
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `panic` command sets or clears the system user panic flag according to what you specify on the command line. If you specify `set`, a `upanic(2)` system call from any user program crashes the system.

Only an appropriately authorized user can use this command.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
<code>system, secadm</code>	Allowed to use this command.
<code>sysadm, sysops</code>	Allowed to use this command. Shell redirected I/O is subject to security label restrictions.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to use this command.

Specify either `set` or `clear` on the command line for `panic` to work. Do not specify both.

**SEE ALSO**

`upanic(2)` in the *UNICOS System Calls Reference Manual*, Cray Research publication SR–2012

**NAME**

`pcache` – Controls device cache allocation, deallocation and attributes

**SYNOPSIS**

```
/etc/pcache [-D high,[low]] [-d] [-h hashlen] [-m members] [-n count] [-s size] [-t type]
[-T max,[min]] [-w] device
/etc/pcache [-f infile]
/etc/pcache [-z]
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `pcache` command controls device cache allocations and/or parameters for a device.

Space for device cache is allocated from main memory. The size of the memory pool is controlled through the boot config file or with the `pcinit(8)` command. Cache may be assigned to a device, or to a set of members comprising the device by using the `-m` option. If cache is being assigned to members of a device, each specified member will receive the amount of cache specified by the `-n` and `-s` options.

Cache may be assigned to any of the following device types: `dev_pdd`, `dev_sdd`, `dev_mdd`, `dev_xdd`, `dev_hdd`. Logical devices (`dev_ldd`) cannot have cache, due to the architecture of the UNICOS/mk operating system; however, a `dev_ldd` (filesystem) device can be specified on the `pcache` command line. In this case, the cache will be allocated to each member of the logical device (or each partition of the filesystem), as if the `-m all` option had been specified.

Cache is specified as a number of units where each unit consists of a specified number of 4096-byte blocks. The number of blocks in a cache unit must be a multiple of the largest I/O unit (disk sector) of the physical devices on which the partition (or partitions) reside.

Old cache units are periodically flushed according to an interval specified in the boot config file. The default interval is 120 seconds. This can cause large backlogs of I/O requests when this sync interval expires, so it is recommended that trickle-sync (`-T` option) be used to balance the the load of flusing old units.

The `pcache` command accepts the following options:

`-D high,[low]` Specifies threshold values for dirty units in cache. The *high* value specifies the maximum number of dirty blocks that may be in the cache of a slice at any one time. If the number of dirty units exceeds *high*, new requests to dirty units will sleep until the number falls below the threshold. When the number of dirty units in cache exceeds *low*, the system automatically starts flushing the oldest dirty units. If *low* is not specified, it defaults to the same value as *high*. If *high* is larger than the number of cache units allocated, it is set to the number of cache units. If *low* is greater than *high*, *low* is set to *high*. To disable the threshold parameters, set *high* to 0. The default is 0.



- `-d` Enables debugging output.
- `-f infile` Read cache allocation command lines from *infile*. Each line in *infile* is a pcache command line. Lines beginning with # are ignored. This option cannot be mixed with any other options, and cannot be nested.
- `-h hashlen` Specifies the size of the cache unit header hash array. The minimum (and default) value is 64. The `-n` option must be used with this option.
- `-m members` Acts on specified members of the named device. *Members* is a comma-separated list of tokens in one of the following formats:
- name* The device name of the member.
  - number* A member number - slice 0 corresponds to the first member.
  - range* A numeric member range having the format *min:max*. The range is inclusive.
  - primary* If the named device is a formatted filesystem, *primary* refers to all members corresponding to primary partitions. Invalid if the device is not a filesystem.
  - secondary* If the named device is a formatted filesystem, *secondary* refers to all members corresponding to secondary partitions. Invalid if the device is not a filesystem.
  - all* Alias for all members. The *all* token is case-insensitive.
- The following examples illustrate use of the `-m` option:
- `-m 0` Indicates the first member of the device.
  - `-m 0,1` Indicates the first and second members of the device.
  - `-m 0:2` Indicates members 0 through 2 (inclusive).
  - `-m all` Indicates all members.
  - `-m primary`  
Indicates all members corresponding to primary partitions.
  - `-m secondary`  
Indicates all members corresponding to secondary partitions.
  - `-m primary, secondary`  
In the case of file systems, functionally equivalent to `-m all`.
  - `-m 0:1,3` Indicates members 0, 1, and 3 (omitting 2).
- `-n count` Number of cache units to allocate. A count of 0 indicates that cache should be deallocated. The `-s` option must also be used if count is greater than 0.
- `-s size` Size in 4096-byte blocks of each cache unit. This option must be used when the `-n` option is specified, and the count is greater than 0.

- `[-t type]` Type of cache (ssd, or ram) to allocate. Currently only ram cache is supported on UNICOS/mk systems. *type* is case-insensitive. This option must be used with the `-n` option.
- `-T max,[min]` Defines device cache aging (trickle-sync) parameters. The *max* value specifies the maximum age in seconds that any unit in cache may reach before automatic flushing of dirty blocks begins. All units older than *min* are flushed. If *min* is not specified, it defaults to the same value as *max*. To disable the aging parameters, set *max* to 0. The default is 0.
- `-w` Indicates that the cache allocated be marked as write-through. The `-n` option must be used with this option.
- `-z` Release cache for all devices. This option cannot be used with any other options.
- `device` Specifies the device name.

**DIAGNOSTICS**

Error messages, warnings, and debug output is issued to `stderr`. Exits with 0 if command completed successfully, non-zero otherwise.

**SEE ALSO**

`pcinit(8)`, `pcstat(8)`, `pcsync(8)`

`hdd(4)`, `mdd(4)`, `pdd(4)`, `sdd(4)`, `xdd(4)` in the *UNICOS/mk File Formats and Special Files Reference Manual*, Cray Research publication SR-2614

**NAME**

`pcinit` – Runtime initialization of pcache

**SYNOPSIS**

`/etc/pcinit [-c core] [-h hdrs] [-d maxdevs] [-s syncint] [ospe ...]`

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `pcinit` command is used to initialize pcache at runtime, rather than having it set up in the configuration file. The `pcinit` command may be run multiple times; however, all cache must be deallocated before running the `pcinit` command.

The `pcinit` command accepts the following options:

- `-c core`      Number of 4096-byte memory blocks to set aside for the cache pool. There is no default.
- `-h hdrs`      Number of cache units to allocate. Default is `core / 16`.
- `-d maxdevs`   Maximum number of devices to be cached. Default is 256.
- `-s syncint`   Interval to flush all dirty units to disk. Default is 900 seconds.
- `ospe`          The OS PE on which to initialize pcache. By default, all OS PEs having a disk actor will be initialized.

**SEE ALSO**

`pcache(8)`, `pcstat(8)`, `pcsync(8)`

**NAME**

pcstat - Displays device cache statistics

**SYNOPSIS**

/etc/pcstat [-b] [-c] [-g] [-h] [-s] [device ...]

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The pcstat command is used to examine device cache statistics. The stats may be restricted to a set of devices specified on the command line. By default, statistics for all cached devices are displayed.

The pcstat command accepts the following options:

- b Displays request size distribution statistics.
- c Displays cache configuration, including dirty unit numbers and ages.
- g Displays global cache information.
- h Displays cache hashing information.
- s Displays cache I/O summary.
- device Specifies device name.

If no options are specified, the -s option is assumed. The display options are cumulative; for example, pcstat -cs displays configuration and summary information.

**OUTPUT FORMATS**

The cache I/O summary output (-s) has the following format:

```

snow$ pcstat -s

```

Name	Read/Write stats		values in Blocks		Rate
	Reads	Writes	Hits	Misses	
/dev/xdd/root	88780	46682	122647	12815	90.54
/dev/xdd/tmp.s21	28691	91410	105617	14484	87.94

The global cache output (-g) has the following format:

```

snow$ pcstat -g

```

OS PE	RAM Cache		Cache Hdrs		Sync	Cached Devs	
	total	avail	total	avail	interval	allowed	cached
ospe_a	4096	16	512	342	300	256	2

Cache configuration (-c) has the following format:

```
snow$ pcstat -c
```

Name	Unit	Size	T	Hash	Unit Age			Dirty		#Dirty	Flags
					Max	Flush	Oldest	Max	Min		
/dev/xdd/root	100	24	M	67	0	0	73	0	0	4	---
/dev/xdd/tmp.s21	70	24	M	67	0	0	74	0	0	6	---

Cache I/O bucket output (-b) has the following format:

```
snow$ pcstat -b
```

	I/O Length Buckets (# blocks per I/O request)								
	1	2-3	4-7	8-15	16-31	32-63	64-127	127-255	255+
Read	9666	183	317	1059	6303	8178	4844	1997	0
Write	42477	197	850	187	431	8	618	0	0
Total	52143	380	1167	1246	6734	8186	5462	1997	0
Percent	67.44	0.49	1.51	1.61	8.71	10.59	7.06	2.58	0.00

Cache hash length output (-h) has the following format:

```
snow$ pcstat -h
```

Name	Hash Size	Hash Used	Shortest	Longest
			Hash Chain	Hash Chain
/dev/xdd/root	67	58	1	4
/dev/xdd/tmp.s21	67	15	1	1

**SEE ALSO**

pcache(8), pcinit(8), pcsync(8)

**NAME**

`pcsync` – Syncs device cache and sets cache sync interval

**SYNOPSIS**

```
/etc/pcsync [-s interval] device ...
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `pcsync` command is used to sync (flush) dirty units from cache. Cache is flushed for each named device, or all cached devices if none are specified.

The kernel flushes all cache periodically, based on an internal sync interval specified in the boot configuration file. This interval may be reset on a running system with the `-s` option. The new sync interval is set to *interval* seconds. If *interval* is 0, the automatic sync is disabled. This will not disable trickle-sync or threshold-based flushing if specified. See the `pcache(8)` man page for more information on the use of trickle-sync and threshold-based flushing.

`-s interval`      Resets internal sync interval (in seconds).

*device*            Specifies device name.

**SEE ALSO**

`pcache(8)`, `pcinit(8)`, `pcstat(8)`

**NAME**

`pddconf` – Controls the state of an IOS model E disk drive

**SYNOPSIS**

`/etc/pddconf [-d device] [function [parameter]]`

**IMPLEMENTATION**

Cray PVP systems with I/O subsystem model E

CRAY J90 series

**DESCRIPTION**

The `pddconf` command controls the state of an IOS model E (IOS-E) disk drive. If no device is specified, all open devices are displayed. `ioctl` requests are issued to both the logical and physical disk drivers to mark the file systems that are affected by the disk state change and to mark the disk driver tables that are necessary. If no options are given, a display of the current state of all drives is shown.

The `pddconf` command accepts the following options:

`-d device` Device *iopath* (for example, 0132 or 0230.3). If no *device* is specified, all open devices are shown. For disk queue sorting, specifying `all` enables or disables the global flag.

*function* The *function* argument can be one of the following:

<code>rw</code>	Sets the physical device mode to read/write.
<code>ronly</code>	Sets the physical device mode to read-only. Read requests are permitted, writes are returned with an <code>errno</code> of <code>EIO</code> . Any mounted file system slice(s) residing on the device are marked as not available for allocation.
<code>noall</code>	Sets the physical device mode to nonallocatable. Any mounted file system slice(s) residing on the device are marked as not available for allocation.
<code>up</code>	Sets the physical disk device state to up. Any previous setting of mode is still in effect.
<code>down</code>	Sets the physical device state to down and terminates all queued I/O requests with errors. Any mounted file system slice(s) residing on the device are marked as not available for allocation.
<code>spinup</code>	Spinup a disk drive. Valid only for DD-60 disk drives, spinup is also done during device open time. (Has no effect on CRAY EL series.)
<code>spindown</code>	Spindown a disk drive. Valid only for DD-60 disk drives. (Has no effect on CRAY EL series.)
<code>pripath</code>	Change disk device to primary path.

- `altpath` Change disk device to alternate path if available. (Has no effect on CRAY EL series.)
- `reset` Reset a given disk statistics.
- `disable spindle 0-4`  
Disable a given spindle of a disk array. A spindle number must be given. (Has no effect on CRAY EL series.)
- `qsort` Turn on the disk queue sorting algorithm in the disk driver for the specified device; the device flag is on by default. Specifying `all` as the device turns on a global flag that enables queue sorting for all disks in a system for which the disk queue sorting algorithm is enabled; this global flag is off by default. See the `EXAMPLES` section.
- `noqsort` Turn off the disk queue sorting algorithm for the indicated device; the device flag is on by default. Specifying `all` as the device turns off a global flag, disabling queue sorting for all disks in a system; this global flag is off by default. See the `EXAMPLES` section.
- `autoswitch on|off`  
Enable/disable autopath switching during error recovery. The system will not switch to the alternate path if the switch is disabled. The device node must be configured with an alternate path to work. (Has no effect on CRAY EL series.)
- `racerron` Routes Recovered Disk Error messages to be printed on the console. This is the default.
- `racerroff` Stops routing Recovered Disk Error messages to the console.

The display has the following format:



name	type	unit	disk devices				flg	wstrm	rstrm	qsort	ENABLED
			state	mode	altp	qon				q'ed	
0230	DD60	0	up	rw	----	0	-----	-----	yes	0	
0230	DD62	3	up	rw	----	0	-----	-----	yes	0	
0230	DD60	4	up	rw	----	1	-----	-----	yes	0	
0230	DD60	5	up	rw	----	0	-----	-----	yes	0	
0232	DD60	0	up	rw	----	1	-----	-----	yes	18	
0232	DD60	1	up	rw	----	0	-----	-----	yes	0	
0232	DD60	2	up	rw	----	0	-----	-----	yes	0	
0232	DD62	3	up	rw	----	1	-----	-----	yes	0	
0234	DD60	0	up	rw	----	1	-----	-----	yes	0	
0234	DD60	1	up	rw	----	1	-----	-----	yes	0	
0234	DD60	2	up	rw	----	1	-----	-----	yes	0	
0234	DD62	3	up	rw	----	1	-----	-----	yes	0	
0236	DD62	2	up	rw	----	0	-----	-----	yes	0	
0236	DD60	4	up	rw	----	1	-----	-----	yes	0	
0236	DD60	5	up	rw	----	1	-----	-----	yes	0	
0334	DA301	0	up	rw	----	1	0100037	0100037	yes	66	

The definitions of the headings in the preceding table are:

- name Device iopath. A lower case "a" is appended to the name when a drive is using its alternate path.
- type Device type.
- unit Device unit number.
- state Device state (init, up, or down).
- mode Device mode (read-write (rw), read-only (ro), or no allocate (na)).
- altp Device alternate path if available.
- flg Path open flags (no path open = 0, primary open = 1, alternate open = 2, both open = 3).
- wstrm Device write stream control flags (used by disk arrays).
- rstrm Device read stream control flags (used by disk arrays).
- qsort ENABLED Indicates that the global qsort flag is enabled.
- qon Indicates that sort is enabled for a particular device.
- q'ed The total number of requests that have been sorted since the system was booted.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

<b>Active Category</b>	<b>Action</b>
system, secadm, sysadm	Allowed to use this command.

If the PRIV\_SU configuration option is enabled, the super user is allowed to use this command.

**EXAMPLES**

Example 1: Use the following command to enable disk sorting for all disks with the disk queue sorting algorithm set to on:

```
# pddconf -d all qsort
pddconf: disk queue sorting ENABLED
```

Example 2: Use the following command to disable disk sorting for all disks:

```
# pddconf -d all noqsort
pddconf: disk queue sorting DISABLED
```

Example 3: Use the following command to switch the disk sorting algorithm to on for disk 1334.0:

```
# pddconf -d 1334.0 qsort
pddconf: disk 1334.0 queue sorting ON
```

Example 4: Use the following command to switch the disk sorting algorithm to off for disk 1334.0:

```
# pddconf -d 1334.0 noqsort
pddconf: disk 1334.0 queue sorting OFF
```

Example 5: Use the following command to control the routing of Recovered Disk Error messages to the console.

```
# pddconf recerron
pddconf: disk recovered error messages routed to Console

# pddconf recerroff
pddconf: disk recovered error messages not routed to Console
```

**SEE ALSO**

ioctl(2) in *UNICOS System Calls Reference Manual*, Cray Research publication SR-2012

**NAME**

`pddstat` – Displays information about the IOS model E

**SYNOPSIS**

```
/etc/pddstat [-d device.unit] [-e] [-r rate] [-l] [-q] [-s]
```

**IMPLEMENTATION**

Cray PVP systems with I/O subsystem model E

CRAY J90 series

CRAY EL series

**DESCRIPTION**

The `pddstat` command gets information from the disk table, which controls disk input/output (I/O) in an IOS model E (IOS-E). The information is attained with the `tabread` (see `tabinfo(2)`) system call and is displayed on the caller's screen.

The `pddstat` command accepts the following options:

- `-d device.unit`  
Displays specific statistics for *device.unit*. *device.unit* is the I/O path (for example, 0130.0 is the device on cluster 0, IOP 1, channel 030, unit 0).
- `-e`  
Displays error streams. This option displays error counts on a per-spindle basis for arrayed devices.
- `-r rate`  
Repeats the display each *rate* second with new information from the MIOP table. If `-r` is not specified, the display is shown only one time.
- `-l`  
Allows display of longer lines in the device (`-d`) display for screens with greater than 80 columns.
- `-q`  
Displays development information about queue lengths on each device as it is sampled at the time of each display update. This option requires that `-r` be specified.
- `-s`  
Displays scientific information about queue lengths on each device as it is sampled at the time of each display update. This option requires that `-r` be specified.

The default display has the following format:

iopath	device			state	mode	req	func	# of sectors		errors		
	type	unit						reads	writes	rd-ur	wt-ur	total
0230	DD60	0		up	rw	0	write	2656	153	0	0	0
0232	DD60	1		up	ro	1	read	5512	1189	0	0	0
0234	DD60	2		down	rw	0	wrtbhd	1267	885	0	0	0
0236	DD60	4		up	rw	1	wrtbhd	14	5	0	0	0
0236	DD60	5		up	rw	3	wrtbhd	6374	205	0	0	0

The following are definitions of the headings in the preceding table:

- iopath Device iopath (for example, 0230 means cluster 0, iop 2, channel 30)
- type Device type
- unit Device unit number
- state Device state (init, up, or down)
- mode Device mode (read-write (rw), read-only (ro), or no allocation allowed on this device (na))
- req Number of outstanding requests
- func Last function that was sent to the disk, unless closed. A full list is in `epackd.h`. The common functions are read, write, write behind, configured up or down, and closed.
- reads Number of sectors read (size of sector depends on disk type).
- writes Number of sectors written
- rd-ur Number of unrecovered read errors
- wt-ur Number of unrecovered write errors
- total Total number of read and write errors (both recovered and unrecovered)

**The -d Option**

The display for an individual device has the following format:

Command line: `pddstat -r 2 -d 1336.0`

Command: pddstat [interval: 2 sec] Page: 1 Wed Oct 9 11:42:22 1991

device							# of sectors		errors		
iopath	type	unit	state	mode	req	func	reads	writes	rd-ur	wt-ur	total
1336	DD60	0	up	rw	0	read	718	0	0	0	0
current requests					0		0 reads	0 writes	rd pct= 0		
total requests					359		359 reads	0 writes	rd pct= 100.00		
I/O errors total					0		0 reads	0 writes			
avg recovery time					0.00		0.00 reads	0.00 writes	(milliseconds)		
avg. transfer size in sectors						read:	0.000	write:	0.000		
total avg transfer sz (sectrs)						read:	2.000	write:	0.000		
avg. I/O time in milliseconds						read:	0.0000	write:	0.0000		
total avg I/O time (millisec)						read:	2.1151	write:	0.0000		
avg. I/O rate in Mbytes/sec						read:	0.0000	write:	0.0000		
total avg I/O rate (MB/sec)						read:	7.7461	write:	0.0000		
avg. wait time in milliseconds						read:	0.0000	write:	0.0000		
total avg wait time (millisecs)						read:	0.0132	write:	0.0000		
avg. number cylinders crossed						rd/wt:	0.0000	last cylinder	15		
total avg number cyls crossed						rd/wt:	0.0418				
number of waiting requests: 00						partitions crossed				1	
number of spare hits: 000000						crossings per I/O				0.0028	
I/O queue high water mark: 01											

Seek Buckets (# cylinders crossed per I/O request)										
	0	1	2-3	4-7	8-15	16-31	32-63	64-127	128-255	
Current	0	0	0	0	0	0	0	0	0	0
Percent	0	0	0	0	0	0	0	0	0	0
Total	344	15	0	0	0	0	0	0	0	0
Percent	95.82	4.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Cylinder Position Buckets (# I/O to each range )									
	0- 129	130- 260	261- 390	391- 521	522- 651	652- 782	783- 912	913-1043	
Current	0	0	0	0	0	0	0	0	0
Percent	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total	359	0	0	0	0	0	0	0	0
Percent	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	=<1173	=<1304	=<1435	=<1565	=<1696	=<1826	=<1957	=<2087	
Current	0	0	0	0	0	0	0	0	0
Percent	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total	0	0	0	0	0	0	0	0	0
Percent	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	=<2218	=<2348	=<2479	=<2610					
Current	0	0	0	0					
Percent	0.00	0.00	0.00	0.00					
Total	0	0	0	0					
Percent	0.00	0.00	0.00	0.00					

Length Buckets (# sectors per I/O request)								
	1	2-3	4-7	8-15	16-31	32-63	64-127	128+
Current	0	0	0	0	0	0	0	0
Percent	0	0	0	0	0	0	0	0
Total	0	359	0	0	0	0	0	0
Percent	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00

The following is an explanation of the information in the pddstat display:

current requests                    0                    0 reads                    0 writes    rd pct=    0  
 Requests since the last refresh of the display; total, number of reads, number of writes, and the percentage of the requests that were reads.

total requests                    359                    359 reads                    0 writes    rd pct= 100.00  
 I/O requests for this unit since the last deadstart of the system; total, reads, writes, and percentage of the requests that were reads.

I/O errors total                    0                    0 reads                    0 writes  
 Count of number of I/O errors for this unit since the last deadstart of the system; total number, number on reads, and number on writes.

avg recovery time                    0.00                    0.00 reads                    0.00 writes (milliseconds)  
 Average time spent in recovery in milliseconds. Average is reported for total I/O errors, reads errors, and write errors.

avg. transfer size in sectors    read:    0.000                    write:    0.000  
 Average transfer size of I/O requests to this unit since the last refresh of the display. The size is reported in number of disk sectors for this device (for example, for DD-60 the sector size is 4 blocks or 2048 words). Transfer size is reported for reads and writes.

total avg transfer sz (sectrs) read: 2.000 write: 0.000  
 Average transfer size of I/O requests to this unit since the last deadstart of the system. The size is reported in number of disk sectors for this device (for example, for DD-60 the sector size is 4 blocks or 2048 words). Transfer size is reported for reads and writes.

avg. I/O time in milliseconds read: 0.0000 write: 0.0000  
 Average time to complete an I/O request in the last refresh span of this display. The average is calculated for read and write requests and is reported in milliseconds per request.

total avg I/O time (millisec) read: 2.1151 write: 0.0000  
 Average time to complete an I/O request since the last deadstart of the system. The average is calculated for read and write requests and is reported in milliseconds per request.

avg. I/O rate in Mbytes/sec read: 0.0000 write: 0.0000  
 The I/O rate of the requests to this disk unit in the last refresh span of the display is calculated using the amount of time that the request took to complete divided by the number of requests. It is displayed for reads and writes in megabytes per second.

total avg I/O rate (MB/sec) read: 7.7461 write: 0.0000  
 The I/O rate of the requests to this disk unit in the period since the last deadstart of the system in megabytes per second; reads and writes.

avg. wait time in milliseconds read: 0.0000 write: 0.0000  
 The average time waiting on the I/O queue before submitting the request to the IOS is calculated for the time span since the last refresh of the display. The average is per I/O request and is an indication of the busy state of the unit. It is calculated for reads and writes.

total avg wait time (millisecs) read: 0.0132 write: 0.0000  
 The average wait time on I/O queue waiting for the IOS to accept the request is calculated for the time span since the last system deadstart.

avg. number cylinders crossed rd/wt: 0.0000 last cylinder 15  
 On each I/O request the number of cylinders that must be passed to be able to perform the I/O is accumulated. This line is the average number of cylinders that were crossed per I/O request in the time span since the last display refresh.  
 The last cylinder to which I/O was done is displayed.

total avg number cyls crossed rd/wt: 0.0418  
 This indicates the average number of cylinders crossed per I/O request on this unit since the last system deadstart.

number of waiting requests: 00 partitions crossed 1  
 Number of I/O requests outstanding in the IOS for this unit.  
 The number of file system partitions crossed on I/O requests is accumulated by the disk driver. This represents crossings since the last system deadstart.

number of spare hits: 000000                      crossings per I/O        0.0028  
 The number of times I/O was done to spare cylinders for this device are accumulated by the driver. Spares are not contiguous to the location of the rest of the file.

Last deadstarted is displayed. This is an indication of the variety of file systems used on this unit.

I/O queue high water mark: 01  
 This represents the longest I/O queue on this unit since the last deadstart of the system.

Seek Buckets (# cylinders crossed per I/O request)  
 The number of seeks of different length is calculated and displayed for varying seek spans. The length of span is kept on an exponential increasing size (that is, the first bucket is 1 cylinder and each subsequent bucket is twice the length of the previous). There are 4 calculations: the number in the last refresh span, the percentage of the seeks in the last span that fall into each bucket, the total number of seeks of this length since the last system deadstart, and the percentage of the seeks since the last deadstart to be of the indicated length range.

Cylinder Position Buckets (# I/O to each range )  
 The cylinder position of each I/O is known at the time of I/O. The driver counts the number of each that fall within equal size ranges of cylinder numbers. The display shows four lines of information: The number of requests in each range during the last refresh span; the percentage of the requests in the last refresh span to fall within each cylinder range; the number of requests in each range during the life of the system; and the percentage of requests that fall within each range since the last system deadstart.

Length Buckets (# sectors per I/O request)  
 The length of each request to the displayed unit is accumulated by the disk driver during its operation. The lengths are counted within ranges of lengths which are not of equal size, but rather vary by powers of 2 in length. The display indicates the number and percentage of requests within each range for the last refresh cycle and the number and percentage of requests within each range for the time since the last system deadstart.

### The -e Option

The -e option displays error counts for array devices only. As the following example shows, error counts are shown on a per-stream (per-spindle) basis. Columns 0 through 3 represent errors attributed to spindles 0 through 3. The P column displays errors attributed to the parity spindle. The ? column displays errors that cannot be attributed to any specific spindle, such as a high-speed channel error. A minus sign preceding the count on columns 0 through 3 or P signifies the drive has been logically removed from the array (see `pddconf(8)`).

This display identifies a spindle that may be a candidate for removal from the array because of errors (see `pddconf(8)`). It is a summary of current array errors only. More detailed information on specific errors is logged and can be extracted using the `errprt(8)` command.

The following example shows that spindle 2 of drive 0334.0 has a problem. The minus sign in front of the count indicates that spindle 2 has been logically removed from the array using the `pddconf(8)` command. It also shows a read error on drive 0334.1 that was not attributable to any single spindle.



drive stream error counts

device	P	3	2	1	0	?	request
334.0	0	0	-89459	0	0	0	reads
	0	0	-4356	0	0	0	writes
334.1	0	0	0	0	0	1	reads
	0	0	0	0	0	0	writes

**The -q Option**

The pddstat command provides the -q option to display more information about each device and its I/O queues. The following is an example of the -q option:

Command line: pddstat -r 10 -q

Command: pddstat [interval: 10 sec] Page: 1 Mon Oct 28 16:03:27 1991

device			# of sectors		more	I/O queue lengths					
iopath	type	unit	req	reads	writes	req	samp	max	1- 4	<10	>=10
0234	DD61	2	0	85417	15655	0	17	0	0	0	0
0236	DD61	3	0	10359	0	0	17	0	0	0	0
0236	DD61	6	0	267219	7578	0	17	0	0	0	0
1030	DD60	0	0	840874	95655	147	17	0	0	0	0
1032	DD60	1	0	2935	20297	0	17	0	0	0	0
1034	DD60	2	0	1176338	1293458	4	17	0	0	0	0
1036	DD60	3	0	440665	448455	0	17	0	0	0	0
1130	DD60	0	0	11581	2506	0	17	0	0	0	0
1132	DD60	0	0	95454	24387	6	17	0	0	0	0
1132	DD60	1	0	15694	8443	0	17	0	0	0	0
1132	DD60	2	0	6326	4297	0	17	0	0	0	0
1134	DD60	4	0	275460	271289	189	17	0	0	0	0
1136	DD60	6	0	479087	522378	135	17	0	0	0	0
1230	DD60	0	0	282645	158304	243	17	0	0	0	0
1232	DD60	4	0	85185	92801	5	17	0	0	0	0
1234	DD60	6	3	1213315	569692	664	17	2	2	0	0

The headings in the preceding display are as follows:

- iopath Path to the I/O device (cluster, iop, iop channel).
- type Type of device on the channel (for example, DD-60, DD-61, DD-40).
- unit The unit number of a device that may or may not be daisy chained.
- req Number of requests for this device which are currently outstanding in the IOS-E.

sectors read	Number of sectors which have been read from this device since the last system deadstart.
sectors written	Number of sectors written to this device since the last system deadstart.
more req	Number of read or write requests to this device since the last refresh cycle (last sample taken from device's statistics tables).
samp	Number of samples or refresh cycles since this display was started.
I/O queue lengths	The I/O queue lengths give an indication of how busy each device has been. The lengths are values determined at the time that samples are taken, that is, when the display is about to be updated. The I/O queue represents the requests that are waiting in the mainframe for the IOS to be ready to receive another request for the device in question.
max	The longest I/O queue that was found for this device when the sample statistics were taken. The queue may have been longer between the times that the samples were taken.
1-4	Times that the sampler found the queue of I/O requests (to go to the IOS) to be at least 1 and less than or equal to 4.
<10	Times that the sampler found the queue of I/O requests to be greater than 4 and less than 10.
>=10	Times that the sampler found the length of the I/O queue for this device to be greater than or equal to 10.

### The -r Option

The -r option takes the preceding information and runs it through a refreshing screen display. When using the -r option for a refreshing display, the following commands control the screen:

- > Increases refresh time
- < Decreases refresh time
- R Scrolls
- r Ends scroll
- + Displays next page of data
- Displays previous page of data
- n Displays next device (if the -d option is used on the command line)
- p Displays previous device (if the -d option is used on the command line)
- q Exits screen mode

**The -s Option**

The pddstat command provides the -s option to display more information about each device and its I/O queues. The following is an example of the -s option:

Command line: pddstat -r 10 -s

```

Command: pddstat [interval: 10 sec] Page: 1 Wed Apr 1 16:41:13 1992
      device
      I/O queue lengths
      average max
      iopath type unit samp 1 2-3 4-7 8-15 16-31 >=32 queue queue
      -----
0230 DD60 0 802 0 0 0 0 0 0 0 0.000 0
0232 DD60 0 802 0 0 0 0 0 0 0 0.000 0
0234 DD60 0 802 0 0 0 0 0 0 0 0.000 0
0236 DD60 0 802 2 0 1 2 0 0 0 0.034 9
1030 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1032 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1034 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1036 DD60 0 802 0 0 1 0 0 0 0 0.005 4
1130 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1132 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1134 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1136 DD60 0 802 0 0 0 1 1 1 1 0.112 54
1230 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1232 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1234 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1236 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1330 DD60 0 802 0 0 0 0 0 0 0 0.000 0
1332 DD60 0 802 0 0 0 1 0 0 0 0.010 8
1334 DD60 0 802 0 0 0 0 0 1 0 0.022 18
1336 DD60 0 802 0 0 0 0 0 0 0 0.000 0
2030 DD50 0 802 0 0 0 0 0 0 0 0.000 0
2032 DD50 0 802 0 0 0 0 0 0 0 0.000 0
2034 DD50 0 802 0 1 0 0 0 0 6 2.968 665
2036 DD50 0 802 0 0 0 0 0 0 1 0.148 119
    
```

The headings in the preceding display are as follows:

- iopath Path to the I/O device (cluster, iop, iop channel).
- type Type of device on the channel (for example, DD-60, DD-61, DD-40).
- unit The unit number of a device that may or may not be daisy chained.
- samp Number of samples or refresh cycles since this display was started.

**I/O queue lengths**

The I/O queue lengths give an indication of how busy each device has been. The lengths are values determined at the time that samples are taken, that is, when the display is about to be updated. The I/O queue represents the requests that are waiting in the mainframe for the IOS to be ready to receive another request for the device in question.

- 1 Times that the sampler found the queue of I/O requests (to go to the IOS) to be 1.
- 2-3 Times that the sampler found the queue of I/O requests (to go to the IOS) to be greater than 1 and less than 4.
- 4-7 Times that the sampler found the queue of I/O requests (to go to the IOS) to be greater than 3 and less than 8.
- 8-15 Times that the sampler found the queue of I/O requests (to go to the IOS) to be greater than 7 and less than 16.
- 16-31 Times that the sampler found the queue of I/O requests (to go to the IOS) to be greater than 15 and less than 32.
- >=32 Times that the sampler found the length of the I/O queue for this device to be greater than or equal to 32.

**average queue**

Total number of I/O requests found by the sampler divided by the number of samples.

**max queue**

The longest I/O queue that was found for this device when the sample statistics were taken. The queue may have been longer between the times that the samples were taken.

**NOTES**

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

**Active Category****Action**

system, secadm

Shell redirected I/O is not subject to security label restrictions.

If the `PRIV_SU` configuration option is enabled, for the super user, shell redirected I/O is not subject to security label restrictions.

**SEE ALSO**

`iocstat(8)`

**NAME**

`pddtest` – Physical disk device test

**SYNOPSIS**

`pddtest [-cvwr] [-u iou] [-n pas] [-s sec] [-p pat] [-P data] special`

**IMPLEMENTATION**

Cray PVP systems with I/O subsystem model E

CRAY J90 series

CRAY EL series

**DESCRIPTION**

The `pddtest` command provides a basic data integrity test of any logical or physical disk device referenced by the character or block special file *special*. By default, only reads are done, but writes can be done optionally as described below. Eight data patterns are tested, one per pass. The low-order 32 bits of each sector contains the block number for that sector.

- c Compares data written to data read. Data must have been sequentially written for compare to succeed.
- v Writes verbose description of what the test is doing to `stdout`.
- w Does writes as well as reads. This will destroy any previous data.
- r Runs on error. The default is to stop on error.
- u *iou* Does reads and writes in *iou* number of sectors. The default is usually 1 cylinder.
- n *pas* Runs *pas* number of passes through the test and stop.
- s *sec* Runs the sections corresponding to the bit mask *sec*. The default is to run all sections, although writes are done only if `-w` is specified.

Section	Bit	Mask	Action
0	2 <sup>0</sup>	01	Sequential writes
1	2 <sup>1</sup>	02	Random writes
2	2 <sup>2</sup>	04	Sequential reads
3	2 <sup>3</sup>	10	Random reads

- p *pat* Runs pattern number *pat*. By default all patterns are run, one per pass. Patterns are as follows:

<i>pat</i>	<b>data pattern</b>
0	000000000000000000000000
1	017777777777777777777777

```

2 01777770000037777600000
3 00000007777740000177777
4 01777777777740000000000
5 00000000000037777777777
6 01252525252525252525252
7 00525252525252525252525

```

`-P data` Runs the user specified data pattern *data*.

`special` Character or block special file.

## NOTES

If this command is installed with a privilege assignment list (PAL), a user with one of the following active categories is allowed to perform the actions shown:

Active Category	Action
system, secadm	Allowed to specify any file.
sysadm	Allowed to specify any file, subject to security label restrictions on the file's path. Shell redirected I/O is subject to security label restrictions.

If the `PRIV_SU` configuration option is enabled, the super user is allowed to specify any file.

## WARNINGS

The `pddtest` command should be run exclusively in read-only mode on a mounted file system. Do not run `pddtest` on a slice being used by a mounted file system with options selected that do writes and/or data compares. The `-w` option will cause filesystem data corruption and the `-cw` option may cause data miscompares, even if you use the `pddconf(8)` command to set the `NOALLOC` option.

## EXAMPLES

Example 1: Runs all sections and all patterns doing writes and data compares:

```
pddtest -cvw /dev/pdd/scr0134.0
```

Example 2: Runs only sequential sections and all patterns doing writes and data compares:

```
pddtest -cvws 5 /dev/pdd/scr0134.0
```

Example 3: Runs all sections and pattern number 2 only doing writes and data compares:

```
pddtest -cvwp 1 /dev/pdd/scr0134.0
```

Example 4: Runs all sections and data patterns and does not stop on error:

```
pddtest -cvwr /dev/pdd/scr0134.0
```

**SEE ALSO**

pddconf(8)

pdd(4) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014

**NAME**

`ping` – Sends ICMP ECHO\_REQUEST packets to network hosts

**SYNOPSIS**

```
ping [-c count] [-d] [-f] [-i wait] [-l preload] [-n] [-p pattern] [-q] [-r] [-R]
[-s packetsize] [-v] host
```

**IMPLEMENTATION**

All Cray Research systems

**DESCRIPTION**

The `ping` command uses the Internet control message protocol's (ICMP) mandatory ECHO\_REQUEST datagram to elicit an ICMP ECHO\_RESPONSE from a host or gateway. ECHO\_REQUEST datagrams (also called pings) have an Internet Protocol (IP) and ICMP header, followed by a `struct timeval`; an arbitrary number of bytes are used to fill out the packet.

The `ping` command accepts the following options and operands:

- `-c count` Stops after sending (and receiving) *count* ECHO\_RESPONSE packets.
- `-d` Sets socket-level debugging (SO\_DEBUG) on the socket being used.
- `-f` Floods `ping`. Outputs packets as fast as they come back, or 100 times per second, whichever is greater. For every ECHO\_REQUEST sent, a period is printed; for every ECHO\_REPLY received, a backspace is printed. This provides a rapid display of the number of packets being dropped. Only the super user can use this option. The `-f` option can overload a network, and it should be used with caution.
- `-i wait` Waits *wait* seconds between sending each packet. The default is 1 second. This option is incompatible with the `-f` option.
- `-l preload` Sends the number of packets specified by *preload* as fast as possible before falling into its normal mode of behavior.
- `-n` Specifies numeric output only. No attempt will be made to look up symbolic names for host addresses.
- `-p pattern` Pads the packet with bytes of a specified pattern. You may specify up to 16 bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network. For example, `-p ff` will cause the sent packet to be filled with all 1's.
- `-q` Specifies quiet output. Nothing is displayed except the summary lines at startup and completion.



- r Bypasses the normal routing tables (by setting flag `SO_DONTROUTE`) and sends requests directly to a host on an attached network. If the host is not on a directly attached network, an error is returned. You can use this option to reach a local host through an interface that has no established route.
- R Records the route. Includes the `RECORD_ROUTE` option in the `ECHO_REQUEST` packet and displays the route buffer on returned packets. The IP header is only large enough to display nine routes. Many hosts ignore or discard this option.
- s *packetsize* Specifies the number of data bytes to be sent. The default is 56, which translates to 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.
- v Enables verbose output. Lists ICMP packets, other than `ECHO_RESPONSE`, that are received.
- host* Specifies the network host. This operand is required.

When using `ping` for fault isolation, run `ping` to the local host to verify that the local network interface is running. Then, run `ping` to hosts and gateways increasingly farther away. `ping` sends one datagram per second, and it prints one line of output for each `ECHO_RESPONSE` returned. If there is no response, no output is produced. Round-trip times and packet loss statistics are computed. When all responses are received or the program times out (with a *count* specified), or if the program is terminated with a `SIGINT`, a brief summary is displayed.

## NOTES

This program is intended for network testing, measurement, and management. It must be used primarily for manual fault isolation. Because of the load `ping` can impose on the network, you should not use `ping` during normal operations or from automated scripts.

### ICMP Packet Details

An IP header without options is 20 bytes. An ICMP `ECHO_REQUEST` packet contains an additional 8 bytes worth of ICMP header, followed by an arbitrary amount of data. When a packet size is given, this indicates the size of this extra piece of data; the default is 56. Thus, the amount of data received inside of an IP packet of type `ICMP ECHO_REPLY` always will be 8 bytes more than the requested data space (the ICMP header). If the data space is at least 8 bytes in length, `ping` uses the first 8 bytes of this space to include a time stamp, which it uses in the computation of round trip times. If less than 8 bytes of padding are specified, no round trip times are given.

### Duplicate and Damaged Packets

The `ping` command will report duplicate and damaged packets. Duplicate packets should never occur, and seem to be caused by inappropriate link-level retransmissions. Duplicates may occur in many situations and are rarely (if ever) a good sign, although the presence of low levels of duplicates may not always be cause for alarm. Damaged packets are obviously a cause for alarm and often indicate broken hardware somewhere in the `ping` packet's path (in the network or in the hosts).

### Trying Different Data Patterns

The internetwork layer should never treat packets differently depending on the data contained in the data portion. Unfortunately, data-dependent problems do occur in networks and remain undetected for long periods of time. In many cases, the particular pattern that will have problems is something that does not have sufficient transitions (for example, all 1's or all 0's, or a pattern right at the edge, such as almost all 0's.) It is not necessarily enough to specify a data pattern of all 0's (for example) on the command line, because the pattern that is of interest is at the data link level, and the relationship between what you type and what the controllers transmit can be complicated. This means that if you have a data-dependent problem, you will probably have to do a lot of testing to find it. You may find a file that either cannot be sent across your network or that takes much longer to transfer than other similar length files. You can then examine this file for repeated patterns that you can test by using the `-p` option of `ping`.

### TTL Details

The transistor-transistor logic (TTL) value of an IP packet represents the maximum number of IP routers that the packet can go through before being thrown away. Typically, each router in the Internet decrements the TTL field by exactly one. The TCP/IP specification states that the TTL field for TCP packets should be set to 60, but many systems use smaller values (4.3BSD uses 30, 4.2BSD used 15). The maximum possible value of this field is 255, and most UNIX systems set the TTL field of ICMP `ECHO_REQUEST` packets to 255. This is why you will find you can `ping` some hosts, but not reach them with `telnet(1B)` or `ftp(1B)`.

In normal operation, `ping` prints the TTL value from the packet it receives. When a remote system receives a `ping` packet, it can do one of the following three things with the TTL field in its response:

- Not change it; this is what Berkeley UNIX systems did before the 4.3BSD-Tahoe release. In this case, the TTL value in the received packet will be 255 minus the number of routers in the round-trip path.
- Set it to 255; this is what current Berkeley UNIX systems do. In this case, the TTL value in the received packet will be 255 minus the number of routers in the path from the remote system to the host using `ping`.
- Set it to some other value. Some machines use the same value for ICMP packets that they use for TCP/IP packets (for example, either 30 or 60). Others may use completely random values.

### BUGS

Many hosts and gateways ignore the `RECORD_ROUTE` option.

The maximum IP header length is too small for options like `RECORD_ROUTE` to be completely useful.

Using the `-f` option is not recommended in general, and using it on the broadcast address should be done only under very controlled conditions.

**SEE ALSO**

ifconfig(8)

ftp(1B), netstat(1B), telnet(1B) in the *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011