

Communicating with Users [8]

As a system administrator, you must communicate with your users frequently. Several methods of communication are available for you to use. The method to use in any specific instance generally is determined by the urgency of your message.

The following list describes the types of communication you will maintain with users, as well as the commands associated with that kind of communication:

<u>Type of communication</u>	<u>Command or file</u>
Issuing emergency messages only	<code>/etc/wall</code>
Issuing critical messages	<code>/etc/issue</code>
Issuing special messages (message of the day)	<code>/etc/motd</code>
Issuing normal (noncritical) communication to all users	<code>/usr/news</code>
Communicating with specific users	<code>write</code> and <code>mail</code>

This chapter describes when you should use each type of communication and gives examples of each.

8.1 Related user communication documentation

The following documentation contains detailed information covered in this section:

- *UNICOS User Commands Reference Manual*, Cray Research publication SR-2011: `mail(1)`, `news(1)`, `su(1)`, `wall(1)`, and `write(1)` man pages
- *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR-2014: `issue(5)` and `motd(5)` man pages

8.2 Issuing emergency messages only

To write (broadcast) emergency messages to all users currently logged in, use the `/etc/wall` command. When a super user executes this command, it overrides message suppression; therefore, use it with discretion.

To run the command, type the `/etc/wall` command. The `wall` command responds by telling you to type your message and to press `CONTROL-d` when you are finished.

To ensure that all users who are currently logged in see a message sent by `wall`, run the command while you have `root` privileges; otherwise, the message goes only to users who allow messages to be written to their terminals (see `mesg(1)`).

Users who are not currently logged in will never see the message; thus, `wall` is not a suitable method for communicating a message to all users who have accounts on the system.

Typically, the `wall` command is used to send the following messages:

- Warnings that the system will soon be brought down for scheduled downtime. Users who log in after the message is sent, however, miss the message and should be notified by the `/etc/issue` file (see the `login(1)` man page).
- Warnings that the system must be brought down immediately to address a system emergency.
- Warnings that a particular file system has run out of disk space and that users should make an immediate effort to delete any unneeded files (see the description of the `-g` option on the `wall(8)` man page).

The following is an example of creating a `wall` message:

```
# /etc/wall
Please log off.  The system will be coming down in 5 minutes for file
system adjustment.  It will be brought back up within the hour.
CONTROL-d.
```

8.3 Issuing critical messages

The `/etc/issue` file is displayed while a user is logging in before he or she has successfully logged in to the system. It is an ordinary text file, and you may place messages in it by using any UNICOS text editor. All interactive users can determine from the message whether they want to log in to the system.

Messages placed in `/etc/issue` should be brief and so important that users may need the information to decide whether to log in to the system. Possible messages include the following:

- Warnings that the system will be brought down soon (so that users who do not see a wall message are not surprised when the system is brought down shortly after they log in)
- Warnings that the system is being used for dedicated time and that not all users can log in

For example, if the message states that the system is going down for maintenance in 5 minutes, a user may choose not to log in to the system at this time. When messages are no longer applicable, be sure to remove them.

The following is an example of creating an issue message:

```
# vi /etc/issue
This machine is being brought down in 5 minutes, 5:30 PM
```

To delete an issue message, enter the following command:

```
# rm /etc/issue
```

8.4 Issuing special messages (message of the day)

The `/etc/motd` (message-of-the-day) file is displayed to users after they are logged in to the system. The `/etc/motd` file is an ordinary text file, and you may place messages in it by using any UNICOS text editor.

You should place messages in `/etc/motd` that are less immediate than those requiring the use of the `wall` command, but that are important enough that users should be forced to see them. You should remove messages from `/etc/motd` as soon as they are no longer needed. Suitable topics for using the `/etc/motd` file include the following:

- Messages to users of scheduled down time
- Warnings to users to clean up unnecessary files on a particular file system or systems
- Brief explanations of recent problems that may have affected users, often with a pointer to a news item that contains a more detailed explanation

Note: Be sure to remove messages when they are no longer applicable; otherwise, your users may start to ignore them. To remove the message of the day, use the `cp /dev/null /etc/motd` command (rather than `rm /etc/motd`); otherwise, `/etc/profile` tries to `cat` a file that does not exist.

The following is an example of creating a message of the day:

```
# vi /etc/motd
=====
This machine will be brought down at 5:30 PM today for
maintenance work. It should be down for only 1 hour.
=====
```

To delete the message of the day, enter the following command:

```
# cp /dev/null /etc/motd
```

8.5 Issuing noncritical communication to all users

The `news` command is the preferred method for delivering noncritical messages to your users. Files you place in the `/usr/news` directory should be ordinary text files you create with any UNICOS text editor. You should create news files that have meaningful names in the `/usr/news` directory.

Note: Be sure to clean out this directory on a regular basis by removing items older than some arbitrary age, such as, 2 months. You can do this by using the `cron` command (for information about using the `cron` command, see the `cron(8)` man page).

Because users are not notified of the existence of a new news file until the next time they log in, and because there is no guarantee that any given user will see the file (a user may choose to ignore the item by not running the `news` command), `/usr/news` is appropriate for items that are not time-sensitive or items that are of interest to only some of the system's users. These categories include the following:

- Notices about recent system changes, such as a newly installed version of a command or library
- Explanations of imminent system reconfigurations or changes

- Explanations of recent system problems and their possible effects on users

Each time a user executes `news`, the `$HOME/.news_time` file is updated.

To display the contents of all current news items, invoke the `news` command without any options.

The following options are commonly used with the `news` command:

<u>Option</u>	<u>Description</u>
<code>-s</code>	Displays a count of current news items (that is, those created since the last modification of <code>\$HOME/.news_time</code>)
<code>-n</code>	Displays the names of current news items

The `/etc/profile` file usually contains a `news -n` command so that each user receives a list of current news items at login time.

The following is an example of creating a news item:

```
# vi /usr/news/chem_access
We have now added a new group called usrchem to control access to the
chemistry applications. If you need to belong to this group, call the
help desk for validation.

                John Doe (jd@cray)
```

To execute a news item, enter the following command:

```
# /usr/news/filename
```

To delete a news item, enter the following command:

```
# rm /usr/news/chem_access
```

8.6 Using the `write` command

The `write` command initiates immediate person-to-person communication with a user who is logged in by opening that user's `tty` or `pty` for writing and copying

each line of text you type to his or her screen. To write to a user who has a login name of `dolores`, for example, you would issue the following command:

```
# write dolores
```

If user `dolores` happened to be logged in on more than one `tty` or `pty`, you could specify the connection:

```
# write dolores tty001
```

If, in this example, user `dolores` is currently logged in, a message appears on her screen indicating that you are writing to her. Typically, user `dolores` replies by writing back to your account; each line of text she types appears on your screen.

Given the immediate nature of its communication, `write` allows you to perform the following functions:

- Converse with a user
- Obtain information about what a user is doing
- Warn a specific user to stop what he or she is doing
- Instruct a specific user to clean up his or her directories

Because each typed line appears on the other user's terminal without regard for what that person may be typing at the moment, it is easy for the other user's messages to your terminal to appear to interfere with your typing. This problem is customarily solved by having the two users take turns typing, ending a message with an `o` on a line by itself (standing for "over," much as in a two-way radio conversation). To end such a session, either user then ends a message with an `oo` on a line by itself (for "over and out"). Thus, a typical "conversation" carried out by using the `write` command might look like this (your input appears in **bold**):

```
# write dolores
Message from dolores (tty001) - Mon May 11 08:20:15 - ...
Yes
o
Please clean up your account, we are out of space.
o
All right, I will.
o
Thank you.
oo
<EOT>
```

Because many users either do not know of this etiquette when using `write`, or do not follow it, they think that `write` is difficult to use. In practice, it is used rather sparingly, mainly when more convenient forms of communication (such as simply calling the user on the telephone) are impossible. Taking steps to educate your user community in the proper use of the `write` command will prove valuable when `write` is the appropriate communication method.

8.7 Using the `mail` command

The `mail` command provides a way to leave messages for specific users, whether or not they are currently logged into the system.

The `mail` command is used as shown in the following example:

```
mail ralph
Type in message
CONTROL-d
```

You may specify more than one account name; in which case, copies of the message go to each user specified. The next time users to whom you (or anyone else) have sent mail messages log in to the system, the system alerts them to the fact that they have mail messages waiting. The `mail` command is thus particularly well suited for messages such as the following:

- Instructions to clean up directories
- Asking or responding to questions
- General communication

In theory, there is no guarantee that the recipient of a mail message will actually see the message, because the recipient may choose not to run the `mail`

command to read the message; however, in practice, most users read their mail when they log in.

For more information, see the `mail(1)` and `mailx(1)` man pages.