

Startup and Shutdown Procedures [3]

This chapter describes the startup and shutdown of a UNICOS system. It discusses the following administration areas:

- System initialization
- System shutdown
- Run-level configuration
- System multiuser startup



Warning: This chapter contains warnings and information critical to the proper use of a Cray ML-Safe system configuration.

3.1 System initialization

The following sections describe how to initialize your Cray Research system. The following tasks are covered:

- Deadstarting the system
- Initializing the UNICOS operating system
- Setting the system date and time
- Setting the system time zone

3.1.1 Deadstarting the system

The procedures for deadstarting a UNICOS system vary according to the type of I/O hardware and system hardware. Different procedures exist for GigaRing based systems (which use the system workstation (SWS)), for systems based on an I/O subsystem model E (IOS-E) (which use an operator workstation (OWS)), and for CRAY J90 systems based on an I/O subsystem model V (IOS-V) (which use the console).

To start a GigaRing based system, use the SWS `bootsys(8)` command. For information about this command and the SWS, see the *SWS-ION Administration*

and Operations Guide, Cray Research publication SG-2204 and the *SWS-ION Reference Manual*, Cray Research publication SR-2205.

To start an IOS-E based system, issue the `load` command at the `BOOT>` prompt on the OWS, and then issue the `/bin/boot` command to start the UNICOS operating system. For more information, see the *UNICOS Installation Guide for the CRAY EL Series*, Cray Research publication SG-5201.

To start a CRAY J90 IOS-V based mainframe, use the console `bootsys(8)` command, or use the `xbootsys(8)` function available from the console `opi` graphical interface. For more information, see the *Support System Reference Manual*, Cray Research publication SR-3077, the *Support System Operator's Guide*, Cray Research publication SG-3078, and the *Support System Administrator's Guide*, Cray Research publication SG-3079.

3.1.2 Initializing the UNICOS operating system

The final phase in starting your system is to initialize the UNICOS operating system. This procedure requires that you check and reset the UNICOS time, check the UNICOS file systems, and enable the network connections. The following procedure details a typical initialization. The UNICOS initialization procedure is customized for each site; messages and questions not explained in this chapter may appear during normal UNICOS initialization at your site.

If your machine is an IOS-E based system, use the `zip` window on the OWS. If your machine is GigaRing based, use the `mfcon(8)` window on the SWS. If your machine is a CRAY J90 IOS-V based system, use the system console.

Perform the following steps:

1. To start multiuser mode (when the system is in single-user mode), enter the following command from the console window (OWS or SWS), or from the system console (CRAY J90 IOS-V based systems):

```
/etc/init 2
```

Note: Most commands typed on the UNICOS interactive console must be entered in lowercase.

2. The date verification prompt appears as follows:

```
Is the date dow MM dd hh:mm:ss tz yyyy correct? (y or n)
```

The *dow* variable is the day of the week, *MM* is the month, *dd* is the day of the month, *hh* is the hour in 24-hour time, *mm* is the minute, *ss* is the second, *tz* is the time zone acronym, and *yyyy* is the year.

Example:

```
Is the date Thu Mar 17 15:57:13 CST 1995 correct? (y or n)
```

Respond by typing *y* for yes, or *n* for no.

If the date or time is wrong, see Section 3.1.3, page 61.

If the time zone is wrong, see Section 3.1.4, page 62.

3. On a CRAY J90 IOS-V based system, this step is performed automatically. On a Cray Research IOS-E based system or on a GigaRing based system, if a dump of the system had been done previously, the system prompts as follows:

```
Enter reason for sysdump:
```

Respond by entering the reason for the dump, followed by RETURN.

The system continues with the following question:

```
Do you want to copy the dump to another file system? (y or n)
```

If you respond with *y*, the dump is written to the file `/core/core.MMdd.hhmm`, where *MM*, *dd*, *hh*, and *mm* represent the current month, day, hour, and minute, respectively. Otherwise, the dump is written to `/core.sys`. (The location of this file is determined by the options given to `coredd` in `/etc/bcheckrc`.)

Note: If the `/core.sys` file is not copied or moved to another file, the dump is overwritten when a new system dump is written to `/core.sys`.

3.1.3 Setting the system date and time

The UNICOS system date and time can be set at boot time (or during operation with the `root` login ID). This section describes the procedure for setting the date and time at boot time.

To change the date and time at boot time, respond with *n* when asked the following question:

```
Is the date dow MM dd hh:mm:ss tz yyyy correct? (y or n)
```

Enter the correct date in the following format:

```
MMddhhmm[yy[ss]]
```

The *MM* variable is the month, *dd* is the day of the month, *hh* is the hour in 24-hour time, *mm* is the minute, *yy* is the last 2 digits of the year (optional entry), and *ss* is the second (optional entry).

Example:

```
0502161595
```

This example changes the date and time to May 2, 1995, 4:15 P.M.

Verify the date and time that the system prints and continue with the startup procedure.

3.1.4 Setting the system time zone

The UNICOS kernel clock keeps track of the date and time relative to Greenwich mean time (GMT). The software must convert between GMT and local time whenever the time is entered or displayed. Local time refers to the time zone where the machine is located. The UNICOS release software is configured for use in the Central time zone (CST) of the United States. You can change the time zone used by the conversion routines if your system is not located in the Central time zone.

Note: Binary-only sites should set their time-zone information in the `TZ` environment variable, as described in the following sections. Ignore the references to source code and simply follow the formats shown for setting the `TZ` variable (`TZ= . . .`).

3.1.4.1 Time-zone information

To change the time zone in the UNICOS operating system, you must have the following information about your time zone:

1. The number of hours away from GMT
2. The direction from GMT (East or West)
3. The name of the time zone
4. Whether your area has daylight savings time
5. Whether the UNICOS algorithm for daylight savings time is correct for your site's location

The daylight savings time conversions used in the UNICOS operating system are valid for most parts of the United States, but may need to be changed for

other countries. The conversion algorithm is located in the routine `/usr/src/lib/libc/gen/ctime.c` (see `ctime(3)`).

The library routines in `/usr/src/lib/libc/gen/ctime.c` determine the daylight savings time rules by using a table containing limits for the dates on which the time change takes place. The algorithm assumes that the time change takes place on a Sunday at 0200 local time.

Table 2 lists the time zones for the United States.

Table 2. United States time zones

Zone name	Hours from GMT	Acronym
Eastern	5	EST
Central	6	CST
Mountain	7	MST
Pacific	8	PST

All the time zones in the United States are west of GMT. The release software is shipped for Central time, which is 6 hours west of GMT, with daylight savings time enabled and the time zone names of CST and CDT. If your time zone is not in Table 2, page 63, find out the correct information for your area by calling Software Product Support (SPS).

To change the time zone, edit the `/etc/inittab` file and change the `tz` line to be correct for your site. The `tz` line in the release is as follows:

```
tz::timezone:TZ=CST6CDT
```

For example, to change to Eastern time, the `tz` line would be:

```
tz::timezone:TZ=EST5EDT
```

For sites located in areas that do not have daylight savings time, specify only one time-zone name, as follows:

```
tz::timezone:TZ=EST5
```

For sites east of GMT, use a minus sign before the number of hours difference between local time and GMT, as follows:

```
tz::timezone:TZ=SOT-1
```

For the time-zone change to take effect, you must reboot the UNICOS operating system.

3.1.4.2 Time-zone example 1

The following is an example of setting the time zone for a site located in Geneva, Switzerland. There is no official time-zone acronym for this area, so the acronym MET (Middle European Time) will represent the time zone that is one hour east of GMT. Daylight savings time exists in this time zone; however, it begins and ends on days other than those in the United States.

This example uses the following conditions:

Geneva is 1 hour east of GMT. For this example, the time zone is named Middle European Time (MET). Geneva has daylight savings time, so two names must be specified: one name for standard time, and one name for daylight savings time. These two names can be the same. The name MET will be used for both standard and daylight savings time.

This example assumes the following data:

Number of hours away from GMT:	1
Direction from GMT (East or West):	East
Time zone name (standard):	MET
Time zone name (daylight savings):	MET
Beginning day of daylight savings time:	Last Sunday in March
Ending day of daylight savings time:	Last Sunday in September

You change the time-zone line in the file `/etc/inittab`. You use the data given above to modify the `tz` line released with the UNICOS operating system. You give the name of the time zone when using standard time (in the following example: MET), followed by a plus or minus sign to reflect the sites' direction east or west of GMT (in the following example: '-'), followed by the number of hours from GMT (in the following example: 1), followed by the name of the time zone when using daylight savings time (in the following example: MET).

In the following example, the `M3.5.0` is the day that daylight savings begins: March (3), the last (5) Sunday (0). The `M9.5.0` is the time daylight savings ends: September (9), the last (5) Sunday (0).

The released and modified `tz` lines follow:

```
Released:    tz::timezone:TZ=CST6CDT
```

```
Modified:    tz::timezone:TZ=MET-1MET,M3.5.0,M9.5.0
```

3.1.4.3 Time-zone example 2

The rules for the United Kingdom are different. Daylight savings time runs from the last Sunday in March to the Sunday after the fourth Saturday in October. For 1995, you would specify:

```
tz::timezone:TZ=GMT0BST,M3.5.0,M10.5.0
```

3.2 System shutdown

You must perform several steps when shutting down the UNICOS operating system. These steps must be performed in the indicated order. However, you can stop the shutdown procedure when you reach single-user mode if you are doing maintenance procedures that require the UNICOS operating system to be at the single-user run level.



Warning: You should never remove power from a peripheral device that is in service, and never power down the system until the UNICOS operating system has been halted.

The `shutdown(8)` script performs most of the shutdown functions. The `shutdown` script is described in detail in the following section. A subsequent section describes a sample shutdown procedure and gives an example of a typical shutdown session.



Warning: If your site is running a Cray ML-Safe system configuration, see Section 3.2.3, page 69, for more information.

3.2.1 The `shutdown` command

The `shutdown(8)` shell script provides an orderly method of shutting down the system. Whenever the system must be shut down, such as for a reboot, run the `shutdown` command. You may find it necessary, or convenient, to change the script to accommodate your local system configuration.

The `shutdown` script released with the UNICOS operating system provides three user exits (`shutdown.pre`, `shutdown.mid`, and `shutdown.pst`) that allow you to modify the shutdown process.

You can specify the grace period allowed between the sending of a warning message and the actual shutdown. This grace period is the number of seconds

of delay. For example, specifying a grace period of 300 results in a 5-minute delay. If no period is specified, the default period is 60 seconds.

The shutdown script released with the UNICOS operating system performs the following functions:

- Determines the user's current directory and does not proceed if the current directory is not the root directory (`/`), `/etc`, `/ce`, or `/sysops`.
- Executes the user exit `/etc/shutdown.pre`, if it exists. If a nonzero return status is returned from the user exit, `shutdown` will prompt the user for confirmation before continuing.
- Sends a message, using `wall(8)`, warning the users who are currently logged in to the system that the system is being shut down.
- Shuts down the NQE subsystem to allow batch jobs to be checkpointed before they are terminated.
- Sends a `SIGSHUTDN` signal to all currently running processes.
- Stops the DM daemon, Tape daemon, and error logging.
- Stops `SYS1` and `SYS2` type daemons, using the `/etc/sdaemon` command.
- Sends a `SIGHUP` signal to all currently running processes.
- Sends a `SIGKILL` signal to all currently running processes.
- Shuts down system accounting, using the following command (see `acctsh(8)`):

```
/usr/lib/acct/shutacct
```

The `shutacct(8)` command records the action of shutting down system accounting in the `/etc/wtmp` file.

- Releases all logical device cache (`ldcache`).
- Runs the user exit `/etc/shutdown.mid`, if it exists. If a nonzero return status is returned from the user exit, `shutdown` will prompt the user for confirmation before continuing.
- Shuts down all configured network interfaces (defined in the `/etc/config/interfaces` file), using the `ifconfig(8)` command.
- Runs the user exit `/etc/shutdown.pst`, if it exists. If a nonzero return status is returned from the user exit, a warning message will be printed (the

shutdown cannot be stopped at this point, because all daemons and processes have been terminated).

- Unmounts all file systems. If any local file systems cannot be unmounted, the shutdown script will issue a warning message.
- Brings the system to single-user mode, using the `init(8)` command with an `s` argument, as follows:

```
/etc/init s
```

When shutdown completes, the UNICOS operating system is in single-user mode. The administrator can safely perform system-related work.

3.2.2 System shutdown configuration

To allow the shutdown process to be tailored to the needs of the site, the `shutdown(8)` script provides three user exits: `shutdown.pre`, `shutdown.mid`, and `shutdown.pst`. The following sections describe these user exits.

3.2.2.1 The `shutdown.pre` user exit

The `shutdown.pre` script is the first user exit of the shutdown script.



Warning: Modification (and/or creation) of this script must comply with the guidelines set forth in the single-user mode descriptions in Chapter 8, page 145. Additionally, the Cray ML-Safe configuration is available to normal users at the time that this script is executed. Therefore, additional care must be taken to ensure that the operations performed by this script follow the same rules and restrictions that are enforced for a security administrator, as described in Chapter 8, page 145.

If an executable named `/etc/shutdown.pre` exists, it will be executed during shutdown. At this point, nothing has been done in shutting down the system. All daemons are still running, all file systems are mounted, and all users are still active and unaware that this script is running.

A possible use of this exit would be to verify the user's permission to run the shutdown script or to run some system cleanup routines.

The shutdown script will check the return status from the `shutdown.pre` program. If the return status is nonzero, the user will be queried as to whether or not to continue the shutdown processing. At this point, the shutdown can be stopped without any effect on the system.

3.2.2.2 The `shutdown.mid` user exit

The `shutdown.mid` script is the second user exit of the `shutdown` script. If an executable named `/etc/shutdown.mid` exists, it will be executed during shutdown.

At this time, all processes (users and daemons) have been terminated, all the disk cache (`ldcache` or `pcache`) has been released, but the network interfaces are still configured, and all of the file systems are still mounted.

A possible use of this exit would be to allow NFS file systems to be unmounted before the networks are stopped.

The `shutdown` script will check the return status from the `shutdown.mid` program. If the return status is nonzero, the user will be queried whether to continue the shutdown processing or not. This exit is given to address any possible problem that may exist with the file systems still mounted and the networks that are still running.

3.2.2.3 The `shutdown.pst` user exit

The `shutdown.pst` script is the third (and last) user exit of the `shutdown` script. If an executable named `/etc/shutdown.pst` exists, it will be executed during shutdown.

At this point, all processes (users and daemons) have been terminated, but the file systems are still mounted. This is virtually single-user mode, except for the file systems.

After this point, the file systems are unmounted and `/etc/init` is invoked to go to single-user mode. The `/etc/init s` command will kill all remaining processes (including the process running the `shutdown` script), so there is no place to put a user exit beyond this point.

Because the system is virtually shut down by this point, there is no reason to halt the script if the user exit return status is not zero. The status returned from `/etc/shutdown.pst` is checked, but `shutdown` will only issue a warning message and then continue to single-user mode.

Note: Be careful in what you allow `shutdown.pst` to execute. Because the various logging daemons (such as `syslogd`) are not available to free up the space, `shutdown.pst` could potentially fill up the file system(s) containing the log files.

3.2.3 System shutdown procedures

To shut down the system, perform the following steps:

1. On a UNICOS system with `PRIV_SU` enabled, log in as `root` on the system console. On a system with `PRIV_SU` disabled (for example, a Cray ML-Safe system configuration), you must log in to the security administrator login and have an active `secadm` category to shut down the system. You will also need to activate the `secadmin` category. For more information, see Section 8.2.3.2, page 154, and Section 8.2.2, page 150.
2. Use `wall(8)` to give users a 10-minute warning. The `wall` command can read either a file or standard input for the message it sends. For more information about using the `wall` command to communicate with users, see Section 5.3.1, page 112.

In the following example, the administrator types a shutdown message (terminated by `CONTROL-d`) into standard input; `wall` broadcasts this message.

```
% /etc/wall
UNICOS is coming down in 10 minutes.
CONTROL-d
```

In addition to sending a shutdown message by using `wall`, it is recommended that you put a short message in the `/etc/issue` file (see `issue(5)`) to inform any users who log in after the warning has been sent that the system will be coming down shortly.

3. Enter the following command to begin the shutdown procedure:

```
shutdown
```

All users logged in to the system are instructed by a broadcast message to log off the system.

A typical session with the `shutdown` program is as follows:

```
# shutdown 300
SHUTDOWN PROGRAM
Thu Sep  1 18:51:58 EST 1992
Executing user exit: /etc/shutdown.pre
User exit /etc/shutdown.pre completed, continuing shutdown
Do you want to send your own message (y or n):y
Type your message followed by <ctrl>d...
System coming down for maintenance!
Please log off.
CONTROL-d
System coming down for maintenance!
Please log off.

(Waits for 5 minutes)

SYSTEM BEING BROUGHT DOWN NOW ! ! !
```



Warning: Cray ML-Safe processing occurs only when in multiuser mode (level 2).

When the shutdown program completes, it displays the following message:

```
INIT: SINGLE USER MODE
```

4. Issue the `sync(1)` command as follows to flush the file system cache to disk:

```
sync
```

Pause approximately 30 seconds before continuing with the following step.

5. Issue the `ldsync(8)` command as follows to flush data from all logical device caches to disk:

```
ldsync
```

At this point, you are in single-user mode and can perform any system administration work that is necessary.

6. If you want to halt the system, perform one of the following steps, depending on your hardware:

- On GigaRing based systems, use the SWS `haltsys(8)` command. For information about this command, see the *SWS-ION Administration and Operations Guide*, Cray Research publication SG-2204.
- On Cray Research IOS-E based systems, type `ehalt(8)` in a local OWS window, or use the `xhalt` function in the `opi` interface.
- On CRAY J90 IOS-V systems, toggle to the IOS console and send a Master Clear command to the mainframe by entering the `mc` command.

3.3 Run-level configuration

A *run level* is a software configuration of the system. Each run level allows only a selected group of processes to exist. Although run levels are most commonly used to configure the system in single-user or multiuser operation modes, thoughtful management of the run-level configuration on the system is a convenient method of tailoring the system's resources to accommodate users' needs.

There are two main modes of operation for the UNICOS operating system: single-user (level `s`) and multiuser (level `2`). Single-user mode is always indicated by run level `s` or `S`. Multiuser mode is typically run level `2`; however, the system can be configured to run in multiuser mode at any level from `0` to `6`.



Warning: Cray ML-Safe processing occurs only when in multiuser mode (level `2`).

The characteristics of each run level are determined by information supplied in the `/etc/inittab` file. The `/etc/inittab` file contains entries that specify actions associated with run levels. Refer to the `inittab(5)` and `init(8)` man pages for more information about the format of the `/etc/inittab` file.

One common use of the `/etc/inittab` file is to set up a run level so that certain procedures are followed automatically only the first time a run level is entered. For example, normally you are asked to verify the date and check the file systems the first time you change your system to multiuser mode. These actions are caused by an entry in the `inittab` file. Subsequent changes in run level do not result in this procedure automatically unless you specifically change the `inittab` file.

3.3.1 Changing run level

You can change your run level by issuing the following command. *level* is the run level you want to initiate:

```
/etc/init level
```

The specific actions that occur when a run level is initiated are controlled by the `/etc/inittab` file. The following sections discuss the strategies for using run levels for different purposes.

3.3.2 Strategies for using run levels

Successful use of run levels requires that you think through the requirements for the system and tailor the initializations of the various run levels to provide for convenient transitions from one run level to another.

All systems have a single-user mode (for system work that needs to be performed unencumbered by the presence of other users on the system) and at least one multiuser mode. If the system is restricted at various times to dedicated use by one or more users, there should be one or more run levels devoted to initializing the system for this dedicated use. In all cases except for single-user mode (which requires little or no initialization), initialization is carried out by the `rc` script (see `brc(8)`).

3.3.2.1 Single-user mode

Many system maintenance, modification, testing, configuration, and repair procedures are carried out while the system is in single-user mode to protect system users from potential instability and to ensure that user processes do not interfere with the system's work while it is in progress. Therefore, the purpose of performing any initialization before the system is in single-user mode is to ensure that the system is known to be in an idle state.



Warning: When you enter single-user mode, you exit the Cray ML-Safe secure processing mode. Use of the Cray ML-Safe configuration in single-user mode by an authorized administrator does not violate the requirements of the Cray ML-Safe configuration.

When the UNICOS operating system is in single-user mode, all network connections and hard-wired terminals are disabled, and only the console terminal can interact with the system. This mode of operation allows you to make necessary changes to the system without doing any other processing. The

character is the system prompt when the UNICOS operating system is in single-user mode.

In most cases, the system is brought into single-user mode either following a system boot or by `shutdown(8)`. In neither case should there be any user processes running after the system is in single-user mode (no user processes will have started following a boot, and `shutdown` kills all user processes before entering single-user mode). Thus, there should be no need for initialization related to user processes when the system enters single-user mode.

It is important that all file systems be unmounted. This provides an extra measure of protection against inadvertent damage done to a mounted file system by single-user mode development work or testing. Traditionally, the person doing the system work or testing while in single-user mode mounts only the partitions he or she requires. To help with this aspect of system work, you can provide a script in `/etc` that mounts the file systems containing system commands not usually found on the root partition (the `/usr` file system) and the home directories of the system staff.

3.3.2.2 Multiuser mode

Traditionally, run level 2 is the system's primary run level for multiuser mode.



Warning: If your site is running a Cray ML-Safe system configuration, see Chapter 8, page 145, for information on the correct configuration.

Among the initializations generally carried out for multiuser mode are the following:

- Recording system startup time in `/etc/wtmp`.
- Mounting all the file systems required for normal system operation. This includes the regular system file systems (`/usr` and `/tmp`), the file system or systems containing the home directories of the system's users, and other file systems containing files to which the users must have access.
- Removing any lock files that may interfere with normal system operation (for example, a lock file for a system daemon).
- Running daemons that provide various system services. The list may include, but is not restricted to, the following:
 - `errdemon`
 - `slogdemon` (for the UNICOS multilevel security (MLS) feature)

- cron
- tapestart (for online tapes, when enabled in `/etc/receptions`)
- syslogd
- Running the `netstart(8)` script to initialize the system's TCP/IP network connections.
- Starting system accounting.
- Moving or truncating log files (for example, `/usr/lib/cron/log` or `/usr/spool/nqs/log`) to prevent them from growing without limits.
- Allowing users to log in.

You can perform some system administration tasks in multiuser mode. For example, you can perform file restore procedures (if necessary) and take periodic status checks of the system. These status checks can include the following:

- A check of free blocks (`df(1)`) remaining on all mounted file systems to ensure that a file system does not run out of space
- A check on mail (`mail(1)`) to `root` or whatever login receives requests for file restores
- A check on the number of system users (`who(1)`)
- A check of all running processes (`ps -eaf`) to determine whether some process is using an abnormally large amount of CPU time

You may want to prepare additional multiuser modes that have the following characteristics:

- They do not mount certain file systems that should be kept secure from the system's users.
- They do not provide certain services or run certain daemons.
- They do not permit users to log in to the system by using certain pieces of software (for example, TCP/IP).

3.3.2.3 Dedicated system

It is sometimes necessary to provide dedicated system time so that a particularly large or time-critical job can run unencumbered by other user processes. Additionally, there will be times at which system development work

requires that the system be brought up as though it were running in multiuser mode, when access to the machine is actually restricted to the system staff.

Access to a UNICOS system can be restricted by having the `rc` script (see `brc(8)`) alter the `ue_permbits` field in the user database (UDB) records to allow system access only to selected users.

3.3.3 Files that control run-level activity

Certain files control various aspects of a system's run levels. The following sections describe those files in detail.

3.3.3.1 The `/etc/inittab` file

The `/etc/inittab` file controls the system's available run levels. The `inittab(5)` man page explains the file's format and options in detail. This file should have the following attributes:

- The initial run level (specified by an entry with the action `initdefault`) should be single-user mode (specified by the letter `s` in the `rstate` field).
- Following the `initdefault` entry, there should be an entry with the action `timezone` to set the `TZ` environment variable to the appropriate value for the time zone in which the system is located.
- Following the `timezone` entry, there should be calls to shell scripts that actually initialize the system's state for the run level being entered.

The `bcheckrc` (see `brc(8)`) program is called by an entry with the action `bootwait` to carry out boot-time-only actions, and the `rc` (see `brc(8)`) program is called by an entry with the action `wait` to carry out actions for switching from one run level to another (including switching from the initial single-user mode to multiuser mode).

- There should be an entry with the action `wait` that links the special file `/dev/systty` to `/dev/syscon`.
- There must be an entry for all run levels, with an action of `respawn`, which executes the `getty(8)` command.
- Any run levels that accept logins from users on front-end systems need an entry with an action of `generic`. This entry instructs `init(8)` to accept login requests from daemons through the `/etc/initreq` FIFO special file (named pipe). (This is true even if the run level is intended for use by a single dedicated user; restricting access to the system is accomplished by the

`rc` script (see Section 3.3.3.3), not by limiting logins to specific devices, as is often done on traditional UNIX systems.

3.3.3.2 The `/etc/bcheckrc` script

By convention, the `bcheckrc` script (see `brc(8)`) is executed only at system boot time, before any file systems are mounted. Its main function is to set the current date and time, prompting the person booting the system for the current date, if necessary, and using the `date(1)` command to set the date.

3.3.3.3 The `/etc/rc` script

The `/etc/rc` script is traditionally executed when the `init` process goes from run level `s` or `1` to a run level greater than `1`. By default, the UNICOS operating system has a standard `/etc/inittab` file that is designed to activate the `/etc/rc` script when the following command is executed:

```
/etc/init 2
```

3.4 System multiuser startup

The `/etc/rc` script supplied by Cray Research is installed in the `/etc` directory. Its source is in file `/usr/src/skl/etc`. The file includes optional execution of local scripts at special points in the `rc` script. The `/etc/rc` script is designed so that it does not need to be modified for most sites.

Note: Previously, sites have heavily modified `/etc/rc` to suit their specific needs. Because the `/etc/rc` script is now unconditionally installed, you should not modify it. Please notify Cray Research of any modifications you found necessary so that your requirements can be considered for future releases.

When it is invoked by `/etc/init`, the `/etc/rc` script performs several operations that are required to bring the UNICOS operating system into multiuser mode. The following sections provide more detailed explanations of the events and files referenced by the `/etc/rc` startup script. Refer to the `brc(8)` man page for additional information on the `rc` script.

3.4.1 Load the `/etc/config/rcoptions` file

If there is a file named `rcoptions` in the `/etc/config` directory, the `/etc/rc` script uses the standard shell command dot (`.`) to *source* the file into its environment variable name space. Existing variables are overridden.



Warning: If your site is running a Cray ML-Safe system configuration, see Chapter 8, page 145, for information on the correct configuration.

If you are using the Installation and Configuration Menu System, the `rcoptions` (shown in Table 3) can be defined on the Configure System->Startup (`/etc/rc`) configuration menu. The parameters can have one of three values:

- YES
- NO
- ASK

If the parameter is set to `YES` or `NO`, the associated action is taken or skipped unconditionally; that is, the operator is not prompted for a response. If the parameter is set to `ASK`, the operator is interrogated at boot time; each parameter description begins with the associated operator prompt.

Table 3. rcoptions Decide String Parameters

Parameter	Description
RC_ACCT	(Do you want to start accounting?) If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the accounting subsystem is activated. The default value for this parameter is YES.
RC_CONTERR	(Do you want to continue with system startup?) If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then the startup will continue in the face of errors from any commands or utilities during system startup. A selection value of NO instructs /etc/rc to terminate system startup and immediately return to single-user mode upon encountering an error during system startup. A selection value of YES instructs /etc/rc to enter multiuser mode regardless of any errors encountered during startup. The default value for this parameter is ASK.
RC_DCE	(Do you want to start DCE?) If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then DCE is started. The default value for this parameter is ASK.
RC_DFS	(Do you want to start DFS?) If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then DFS is started. The default value for this parameter is ASK.
RC_FSCK	(Do you want to execute fsck?) If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then mfsck will run on each file system listed in /etc/fstab. Setting this to NO will still cause mfsck to run fsck, but it will do so in quick mode (fsck -q). The default value for this parameter is YES.

Parameter	Description
RC_FSCK_Y	<p>(Do you want to automatically respond yes to the fsck prompts for ... ?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then all the fsck invocations during the boot process will have the -y flag present. The -y flag's presence causes fsck to assume a YES response to all of its prompts.</p> <p>The default value for this parameter is NO.</p>
RC_MKTMP	<p>(Do you want to mkfs /tmp?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, mkfs(8) is used to remake the /tmp file system device. If this parameter is set to NO, fsck(8) is used to check the /tmp device.</p> <p>The default value for this parameter is ASK.</p>
RC_MKUTMP	<p>(Do you want to mkfs /usr/tmp?).</p> <p>If there is a /usr/tmp device configured, this parameter is checked; otherwise, it is ignored.</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, mkfs(8) is used to remake the /usr/tmp file system device. If this parameter is set to NO, fsck(8) is used to check the /usr/tmp device. If the /usr/tmp file system device is remade, recovery of the vi or ex editing file is impossible.</p>
RC_NET	<p>(start the network?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the /etc/netstart script is executed.</p> <p>The default value for this parameter is YES.</p>
RC_NFS	<p>(Do you want to start NFS?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then NFS is started.</p> <p>The default value for this parameter is ASK.</p>
RC_PCACHE	<p>(Do you want the partition cache activated?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then the partition cache is activated with the following command:</p> <pre>/etc/pcache -f /etc/config/pchlist</pre> <p>The default value for this parameter is YES.</p>

Parameter	Description
RC_SADC	<p>(Do you want to start the system activity daemon?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the system activity data collection process is activated.</p> <p>The default value for this parameter is YES.</p>
RC_SMT	<p>(Do you want to start the FDDI SMT daemon?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the FDDI SMT daemon is started.</p> <p>The default value for this parameter is ASK.</p>
RC_TAPE	<p>(Do you want to start the tape daemon?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the /etc/tpinint command is executed.</p> <p>The default value for this parameter is ASK.</p>
RC_TCP	<p>(Do you want to start TCP?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the networks are started.</p> <p>The default value for this parameter is ASK.</p>
RC_USRMNT	<p>(Do you want to mount the user file systems?)</p> <p>If the /etc/fstab file exists, this parameter is checked; otherwise, it is ignored.</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, the file systems in /etc/fstab are mounted. The /etc/fstab file has the following format:</p> <p><i>file_system_device_name mount_point_directory</i></p> <p>(If you are using the configuration tool, select fstab, load your /etc/fstab file, and modify the File system configuration menu.)</p> <p>The default value for this parameter is YES.</p>
RC_YP	<p>(Do you want to start NIS (formerly YP)?)</p> <p>If this parameter is set to YES, or if it is set to ASK and the operator replies YES to the question, then NIS is started.</p> <p>The default value for this parameter is ASK.</p>

If you are using the Installation and Configuration Menu System, the security `rcoptions` parameters (shown in Table 4, page 81) can be defined on the Configure System->Multilevel Security (MLS) Configuration->MLS System Options menu.

Table 4. `rcoptions` Non-decide String Parameters

Parameter	Description
<code>RC_CRONLOGDIR</code>	The path to the <code>cron</code> log directory. The default value for this parameter is <code>/usr/lib/cron</code> .
<code>RC_LOG</code>	The file name in which the startup entries and various information is sent. The default value for this parameter is <code>/etc/rc.log</code> .
<code>RC_NFSLOG</code>	The file name of the NFS log. The default value for this parameter is <code>/etc/nfs.log</code> .
<code>RC_SECLOW</code>	The lower security level (used by <code>mkfs(8)</code> to construct <code>/tmp</code> and <code>/usr/tmp</code>). The default value for this parameter is 0.
<code>RC_SECHIGH</code>	The upper security level (used by <code>mkfs(8)</code> to construct <code>/tmp</code> and <code>/usr/tmp</code>). The default value for this parameter is 0.
<code>RC_SECMASK</code>	The security compartments mask. The default value for this parameter is <code>none</code> .

If you are using the Installation and Configuration Menu System, the following variables (shown in Table 5, page 82) can be defined in the Configure System->Special system device definitions menu.

Table 5. rcoptions File System String Parameters

Parameter	Description
DUMPDEV	The dump device used by the <code>dumpsys(8)</code> utility. It should be assigned the value of <code>dmpdev</code> in the configuration file on the SWS. The default value for this parameter is " ".
DUMPDIR	The directory created on the file system (DUMPFS) into which the dump directory is created. The value of DUMPDIR is passed to <code>/etc/coredd</code> , which creates a unique directory for the dump (<i>mmddhhmm</i>) and copies the dump and current versions of <code>/unicos</code> and <code>/etc/crash</code> to that directory. The directory structure looks like the following: <code>/DUMPMPT/DUMPDIR/mmddhhmm</code> The default value for this parameter is " ".
DUMPMPT	The mount point onto which the file system (DUMPFS) is temporarily mounted for <code>/etc/coredd</code> to copy the dump onto when the system is brought up in multiuser mode. If this directory does not exist, it is created by <code>/etc/coredd</code> . The default value for this parameter is <code>/mnt</code> .
DUMPFS	The file system to which a system dump is copied when the system comes up in multiuser mode. The value of DUMPFS is passed to <code>/etc/brc</code> , which calls <code>/etc/coredd</code> . The file system is temporarily mounted on the selected mount point (DUMPMPT), and the dump is copied onto that file system. The file system is unmounted and then mounted in its usual place (specified by <code>fstab</code>) as the system comes up. The default value for this parameter is <code>core</code> .
MNTTMPOPTS	The temporary (<code>/tmp</code>) mount options. This specifies any miscellaneous options for the <code>mount(8)</code> command (for example, <code>quotas</code> or number of inodes) used to initialize <code>/tmp</code> . Note that any values specified for the <code>-o</code> or <code>-t</code> options will be overridden. The <code>-o</code> option is forced to <code>rw</code> in <code>/etc/rc</code> , and the <code>-t</code> option is forced to <code>NC1FS</code> . The default value for this parameter is " ".

Parameter	Description
MNTUTMPOPTS	<p>The user temporary (<code>/usr/tmp</code>) mount options. This specifies any miscellaneous options for the <code>mount(8)</code> command (for example, quotas or number of inodes) used to initialize <code>/usr/tmp</code>. Note that any values specified for the <code>-o</code> or <code>-t</code> options will be overridden. The <code>-o</code> option is forced to <code>rw</code> in <code>/etc/rc</code>, and the <code>-t</code> option is forced to <code>NC1FS</code>.</p> <p>The default value for this parameter is <code>" "</code>.</p>
PIPEDEV	<p>The device name of the pipe file system. It is typically the same as <code>ROOTDEV</code>. It should be assigned the value of <code>pipedev</code> in the configuration file on the SWS.</p> <p>The default value for this parameter is <code>root</code>.</p>
ROOTDEV	<p>The device name of the <code>/</code> file system, or the <code>root</code> file system device. It should be assigned the value of <code>rootdev</code> in the configuration file on the SWS.</p> <p>The default value for this parameter is <code>root</code>.</p>
SRCDEV	<p>The file system that contains the UNICOS source. It is used in single-user mode to mount <code>/usr/src</code> and used by secure labeling to install trusted subject programs.</p> <p>The default value for this parameter is <code>src</code>.</p>
SWAPDEV	<p>The swap logical device name. It should be assigned the value of <code>swapdev</code> in the configuration file on the SWS.</p> <p>The default value for this parameter is <code>swap</code>.</p>
TMPDEV	<p>The device name of the <code>/tmp</code> file system. The <code>/etc/rc</code> script must mount the <code>/tmp</code> file system to facilitate the system boot. Optionally, <code>/etc/rc</code> makes the <code>tmp</code> system device or uses <code>fsck(8)</code> to check it and then mounts it.</p> <p>The default value for this parameter is <code>tmp</code>.</p>
TMPOPTS	<p>The temporary (<code>/tmp</code>) <code>mkfs(8)</code> options. This specifies any miscellaneous options for the <code>mkfs(8)</code> command (for example, quotas) used to initialize <code>/tmp</code>.</p> <p>The default value for this parameter is <code>" "</code>.</p>
USRDEV	<p>The logical device of the <code>/usr</code> file system device. The <code>/etc/rc</code> script must mount the <code>/usr</code> file system to enable access to administrative directories.</p> <p>The default value for this parameter is <code>usr</code>.</p>

Parameter	Description
USRTMPDEV	The device name of the <code>/usr/tmp</code> file system device, if it is configured. This parameter can be set to blank (" ") in <code>/etc/config/rcoptions</code> . A blank value indicates that <code>/usr/tmp</code> is part of the <code>/usr</code> file system device; in this case, the parameter is ignored. If this value is not blank, <code>/etc/rc</code> can (optionally) make and mount the file system. The default value for this parameter is <code>usr_tmp</code> .
USRTMPOPTS	The user temporary (<code>/usr/tmp</code>) mount options. This specifies any miscellaneous options for the <code>mkfs(8)</code> command (for example, quotas) used to initialize <code>/usr/tmp</code> . The default value for this parameter is " ".

3.4.2 Set up the `/etc/rc` log file

The `/etc/rc` log file is defined by the `RC_LOG` parameter; output from the `/etc/rc` script is captured in this file. The log file is reused (overwritten) for each boot. If the `RC_LOG` parameter is set to a blank value, all output goes to `/dev/console`. The `RC_LOG` parameter is exported so that scripts called by `/etc/rc` can also access the log file.

3.4.3 Execute `/etc/rc.pre`

If the local executable script `/etc/rc.pre` exists, it is executed at this time. That is, the `/etc/rc.pre` script is executed after the `/etc/rc` log file is established and before any file system mounts are done.

This file is not supplied by Cray Research.



Warning: This script does not meet the requirements of a Cray ML-Safe system configuration.

3.4.4 Make and mount `/tmp`

If the `RC_MKTMP` parameter is set to `YES`, or if it is set to `ASK` and the operator replies `YES` to the question, the `/tmp` file system device is created. It is always checked (using `fsck(8)`) and then mounted at this time.

3.4.5 Mount the `/usr` file system

The `/etc/rc` script mounts the `/usr` file system. The file system is not checked with `fsck(8)`, because it is usually in the `/etc/fstab` table, and, therefore, is checked before `/etc/rc` is executed (see `bcheckrc` on the `brc(8)` man page and `fstab(5)`).

3.4.6 Make and mount `/usr/tmp`

If you have a `/usr/tmp` file system device, this action is taken. If the `RC_MKUTMP` parameter is set to `YES`, or if it is set to `ASK` and the operator replies `YES` to the question, the file system is remade. It is always checked (using `fsck(8)`) and then mounted.

3.4.7 Preserve interrupted `vi/ex` sessions

If the `/usr/tmp` file system was not remade, `expreserve` (see `vi(1)`) is executed to preserve any interrupted `vi` or `ex(1)` editing sessions.

3.4.8 Mount user file systems

If the `/etc/fstab` file exists, `/etc/rc` uses the list it contains to mount the user file systems. Depending on the number of user file systems configured, it may take a significant amount of time to mount them; therefore, a speedometer is displayed at this point in the `/etc/rc` script.

3.4.9 Mount `/proc`

The `/proc` file system mount point is set, and the `/proc` (special) file system is mounted. Because various UNICOS subsystems require the presence of the `/proc` file system, this step is mandatory.

3.4.10 Activate logical device cache

If the `/etc/config/ldchlist` file exists, the `/etc/rc` script uses the list it contains to assign cache to the designated file systems. See `ldcache(8)` for more information.

3.4.11 Execute `/etc/rc.mid`

If the local executable script called `/etc/rc.mid` exists, it is executed at this time. That is, the `/etc/rc.mid` script is executed after the user file systems are mounted, but before any administrative cleanup is done or any daemons are active.

This file is not supplied by Cray Research.



Warning: This script does not meet the requirements of a Cray ML-Safe system configuration.

3.4.12 Perform administrative cleanup

The `/etc/rc` script performs the following cleanup operations:

1. If `coredd` left a mail file, the file is mailed to root (see `coredd(8)`).
2. The `/usr/adm/acct/nite/lock*` files are removed.
3. Maintenance is performed on the super-user log (`su` log).
4. Maintenance is performed on the cron log (see `cron(8)`).
5. Maintenance is performed on the file `/etc/wtmp`.

3.4.13 Start the security log daemon

The UNICOS MLS feature `/etc/rc` script starts the security log daemon.

3.4.14 Start accounting

If the `RC_ACCT` parameter is set to `YES`, or if it is set to `ASK` and the operator replies `YES` to the question, the accounting startup script is activated.

3.4.15 Start system activity data collection

If the `RC_SADC` parameter is set to `YES`, or if it is set to `ASK` and the operator replies `YES` to the question, the activity data collection startup daemon is activated.

3.4.16 Activate category `SYS1` system daemons

The `sdaemon(8)` command is used to start all category `SYS1` daemons at this time. `SYS1` daemons are those that must be started before network startup.

For more information, see the description of the `netstart` script in the *UNICOS Configuration Administrator's Guide*, Cray Research publication SG-2303.

3.4.17 Activate `netstart`

If the `RC_NET` parameter is set to `YES`, or if it is set to `ASK` and the operator replies `YES` to the question, the network startup script is activated. The `netstart(8)` script initializes all networking activities.

3.4.18 Activate category `SYS2` system daemons

The `sdaemon(8)` command is used to start all category `SYS2` daemons at this time. `SYS2` daemons are those that must be started after network startup.

For more information, see the description of the `netstart` script in the *UNICOS Configuration Administrator's Guide*, Cray Research publication SG-2303.

3.4.19 Create network access list

With the UNICOS MLS feature, the secure network access list (NAL) is created in the Installation and Configuration Menu System in the Configure System->Multilevel Security (MLS)->Configuration->MLS Network Security Options->Network Protocol Security Configuration->MLS Network Access List (NAL) Sets menu.

The NAL is defined in the `/etc/config/spnet.conf` file, along with the workstation access list (WAL) and CIPSO map-domain sets. When the UNICOS operating system is put into multiuser state, the `spnet.conf` file is read by the `/etc/spnet` command to load the NAL, WAL and CIPSO map tables into the kernel.

3.4.20 Set MLS wildcard files and directories

With the UNICOS MLS feature, the `/etc/spwcard` program (see `spwcard(8)`) is executed. This program assigns certain directories and files, such as `/dev/null`, a wildcard label, which allows them to contain files at varying security levels.

Note: If your site has already configured the system to support the Cray ML-Safe configuration, using the `spwcard` command does not relabel any file or directory (for example, relabel a multilevel directory (MLD) with a wildcard label).

3.4.21 Execute `/etc/rc.pst`

If the local executable script called `/etc/rc.pst` exists, it is executed at this time. That is, the `/etc/rc.pst` script is executed after all daemons are started, and just before the multiuser startup process is finished.

This file is not supplied by Cray Research.



Warning: Modification (and/or creation) of this script must comply with the guidelines set forth in the single-user mode descriptions in Chapter 8, page 145.

Additionally, the Cray ML-Safe system configuration is available to normal users at the time that this script is executed. Therefore, additional care must be taken to ensure that the operations performed by this script follow the same rules and restrictions that are enforced for a security administrator, as described in Chapter 8, page 145.

3.4.22 Complete the multiuser startup

As its final task, `/etc/rc` writes an entry into `/etc/boot.log`, if that file exists.