Run-time Configuration Scripts and Files [6]

This chapter describes scripts and files that are important in the configuration of various aspects of a UNICOS system. These scripts and files are executed or read during system initialization or when changing run levels. For a discussion of system initialization and run-level configuration, see *General UNICOS System Administration*, Cray Research publication SG–2301.

Each of the following sections discusses one script or file. The scripts and files appear alphabetically by file name (not path name).

The following scripts and files are discussed:

- `/usr/lib/cron/at.deny` and `/usr/lib/cron/at.allow`
- `/etc/bcheckrc`
- `/etc/brc` and `/etc/coredd`
- `/usr/lib/cron/cron.deny` and `/usr/lib/cron/cron.allow`
- `/etc/cshrc`
- `/etc/config/daemons`
- `/etc/fstab`
- `/etc/gettydefs`
- `/etc/ghippi#.arp` (GigaRing based systems only)
- `/etc/group`
- `/etc/hycf.`*local_network* (IOS-E based systems only)
- `/etc/inittab`
- `/etc/config/interfaces`
- `/etc/issue`
- `/etc/config/ldchlist`
- `/etc/motd`
- `/etc/netstart`
- `/etc/config/netvar.conf`

- /etc/passwd
- /etc/profile
- /etc/rc
- /etc/config/rcoptions
- /etc/shutdown
- /etc/umountem

6.1 at.deny and at.allow files

You can alter the /usr/lib/cron/at.deny and /usr/lib/cron/at.allow files. These files define users who are permitted or excluded from using the at(1) command.

Users are permitted to use at if their login appears in the file /usr/lib/cron/at.allow. If that file does not exist, the file /usr/lib/cron/at.deny is checked to determine if the user should be denied access to at. If neither file exists, only root is allowed to submit a job. An empty at.allow file means no user is allowed to use at; an empty at.deny file means no user is denied the use of at.

UNICOS is released with an empty at.deny file. The allow/deny files consist of one user name per line. Change one file or the other to allow or deny user access to at.

6.2 bcheckrc script

The /etc/bcheckrc(8) script performs the commands necessary (such as setting the system time or checking file system consistency) before the rc script (see brc(8)) is to be executed and the file systems are to be mounted. See the bcheckrc(8) man page for more information.

Note: This script is not intended to be modified directly. To modify the execution of this script, change the options in /etc/config/rcoptions.

6.3 brc and coredd scripts

The brc(8) script is used for processing system dumps. It copies system dumps to a separate file system by executing the coredd(8) shell script.

Note: These scripts are not intended to be modified directly. To modify the execution of this script, change the options in /etc/config/rcoptions.

See the brc(8) and coredd(8) man pages for more information.

6.4 cron.deny and cron.allow files

You can alter the /usr/lib/cron/cron.deny and /usr/lib/cron/cron.allow files during configuration. These files define users who are permitted or excluded from using the crontab(1) command.

Users are permitted to use cron if their login appears in the file /usr/lib/cron/cron.allow. If that file does not exist, the file /usr/lib/cron/cron.deny is checked to determine if the user should be denied access to cron. If neither file exists, only root is allowed to submit a job. An empty cron.allow file means no user is allowed to use cron; an empty cron.deny file means no user is denied the use of cron.

UNICOS is released with an empty cron.deny file. The files consist of one user name per line. Change one file or the other to allow or deny user access to cron.

6.5 cshrc file

The /etc/cshrc file is the equivalent of /etc/profile (see Section 6.20, page 102) for csh (the C shell). The release version of /etc/cshrc sets the user's file creation mode, prints the /etc/motd file on the user's screen, and tells the user if mail and news exists.

If the /etc/cshrc file exists and your login shell is /bin/csh, the file is executed by the shell upon login before your session begins. Then, if your login directory contains a file named .cshrc, it is executed by the shell before the session begins. See the cshrc(5) and csh(1) man pages for more information.

6.6 daemons file

The `/etc/config/daemons` file is used by the `sdaemon`(8) command for controlling the starting and stopping of system daemons. (The `sdaemon`(8) command is used by the system startup procedures supplied with UNICOS to start daemon processes necessary for system operation.) To access this file through the UNICOS installation and configuration menu system (ICMS), use the following menu selection:

```
Configure system
    ->System daemons configuration
        ->System daemons table
```

An entry in the `/etc/config/daemons` file has the following format:

daemongroup tag groupstart kill pathname arguments

daemongroup	The group to which this daemon belongs.
tag	A convenient tag to use when referring to the daemon. You can use the name of the daemon's executable file or any other name you choose.
groupstart	Indicates whether this daemon should be started when its group is started. A value of `YES` indicates this daemon will be started as part of its group. A value of `ASK` indicates that the operator will be asked at startup time if the daemon should be started. Any other value (`NO` is suggested) indicates that the daemon will not be started as part of its group. A daemon with a *groupstart* of `NO` may be started individually.
kill	A value indicating how the daemon process may be terminated.
pathname	The path name of the daemon executable.
arguments	Command-line arguments to be used when starting the daemon.

A full explanation of the format of the `/etc/config/daemons` file and its use by the `sdaemon`(8) command may be found on the `sdaemon`(8) man page.

The following *daemongroup* values are used by the system startup procedure supplied with UNICOS to start groups of daemons in the following order:

`SYS1`	Non-networking daemon processes started before the networks are present
`TCP`	Daemon processes related to TCP/IP
`NFS`	Daemon processes related to NFS
`SYS2`	Non-networking daemon processes started after the networks are present

You may choose to define additional groups to suit your local configuration.

6.7 `fstab` file

The `/etc/fstab` file (see `fstab`(5)) contains a list of file systems that the `bcheckrc`(8) script checks by invoking the `mfsck`(8) command. UNICOS programs read `/etc/fstab`, but do not write information to it; you must create and maintain the information in this file.

Note: The `mfsck`(8) command was formerly called `gencat`(8).

Use the following menu selection to change this file:

```
Configure System
    ->File System (fstab) Configuration
```

However, if you want to change the file manually, this section contains the information you need.

The `/etc/fstab` file is an ASCII file. The fields are separated by white space (tabs or spaces); each group is separated from the next by a newline character. Each entry in `fstab` has the following format:

filesystem directory type options frequency passnumber

The first and last fields are used by `mfsck` (see the `mfsck`(8) man page for more information). See also the routines `getfsent`, `getfsspec`, `getfstype`, and `getfsfile` described on the `getfsent`(3) man page for information on reading records from `/etc/fstab`.

The /etc/fstab file contains entries to make mounting file systems easier. When the mount(8) command is invoked with only a directory argument or an incomplete argument list, it searches fstab for the missing arguments.

The *type* field must have one of the following values. Each value specifies a file system type.

Type	File system
NC1FS	UNICOS file system
NFS	Network file system
PROC	/proc file system

See the fstab(5) man page for further information on fstab. See also the mount(8) man page.

6.8 gettydefs file

The /etc/gettydefs file contains information used by the getty(8) command to set up the speed and terminal settings for a line. getty is used for the IOS-E terminals.

The information in gettydefs also specifies the appearance of the login prompt.

The /etc/gettydefs file is an ASCII file. The fields are separated by pound signs (#); each group is separated from the next by a newline character. Each entry in gettydefs has the following format:

```
label#initial_flags#final_flags#login_prompt#next_label
```

The fields contain the following information:

label	String against which getty tries to match its second argument.
initial_flags	Initial ioctl(2) settings to which the terminal is to be set if a terminal type is not specified with getty.
final_flags	Flags that are set just before getty executes login(1).

login_prompt	The login prompt. All characters and white space in the field are included in the prompt.
next_label	The desired speed, specified by a break character. If this field does not contain the speed, `getty` searches for the entry with *next_label* as its *label* field and sets up the terminal for those settings.

See the `gettydefs`(5) and `getty`(8) man pages for detailed information on the `/etc/gettydefs` file. Also see the `login`(1), `vi`(1), and `ioctl`(2) man pages for more information on `gettydefs`.

6.9 `ghippi#.arp` file

The `/etc/ghippi#.arp` file lets you configure the hardware addresses of known hosts on local networks that are directly connected to a GigaRing based system.

6.10 `group` file

The `/etc/group` file (see `group`(5)) is provided to translate group names to group IDs and group IDs to names. It is not used for user validation.

The / `etc/group` file is an ASCII file that contains the following information for each group:

- Group name
- Encrypted password
- Numerical group ID (GID)
- Comma-separated list of user names allowed in the group

The `password` field is not used in UNICOS and is set to an asterisk (*). See the `group`(5) man page for more information on the `/etc/group` file and `udb`(5) and *General UNICOS System Administration*, Cray Research publication SG–2301, for more information on the user database (UDB) and the `group` file.

6.11 `hycf.`*local_network* files

The `/etc/hycf.`*local_network* files let you configure the hardware addresses of known hosts on local networks that are directly connected to an IOS-E based system. There is one file for each local network connected to the mainframe.

6.12 `inittab` file

The `/etc/inittab` file controls the actions of the `init`(8) process. It contains information on processes and scripts that are to be executed at the various run levels. See *General UNICOS System Administration*, Cray Research publication SG–2301, for a description of UNICOS run levels and run-level configuration.

The `/etc/inittab` file is composed of position-dependent entries that have the following format:

```
id:rstate:action:process
```

Each entry is delimited by a newline character; however, a backslash (\) preceding a newline character indicates a continuation of the entry. Up to 512 characters for each entry are permitted. Comments may be inserted in the process field, using a # sign at the beginning of the field.

The entry fields are as follows:

Field	Description
id	1 to 4 characters that uniquely identify an entry.
rstate	Run level in which this entry is to be processed.

action	Keywords that specify how to treat the process in the *process* field. The following actions are recognized by `init`:
	`boot` `bootwait` `generic`
	`initdefault` `off` `once`
	`ondemand` `respawn` `sysinit`
	`timezone` `wait`

process	An `sh`(1) command to be executed.

Be sure to change the time zone entry in the `/etc/inittab` file to reflect your site's time zone. The ICMS will copy your time zone entry if you are performing an upgrade installation, but initial installations must make the change to

`/etc/inittab` manually. For instructions on changing the time zone, see *General UNICOS System Administration*, Cray Research publication SG–2301.

For more information about the structure of the `/etc/inittab` file, see `inittab`(5) in the *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR–2014.

6.13 `interfaces` file

The `/etc/config/interfaces` file contains the list of TCP/IP interfaces that may be configured using the `initif`(8) command. The `initif` command is used by the system startup procedures supplied with UNICOS. A full explanation of the format of the `/etc/config/interfaces` file may be found on the `initif`(8) man page and in the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304. To access this file through the ICMS, use the following menu selection:

```
Configure system
    ->Network configuration
        ->General network configuration
            ->Network interface configuration
```

6.14 `issue` file

The `/etc/issue` file (see `issue`(5)) is displayed before the login prompt. It is used to echo any important messages users might need to know prior to logging in, such as the following:

```
The system is dedicated.
Please do not log on.
```

See *General UNICOS System Administration*, Cray Research publication SG–2301, for more information on the `/etc/issue` file.

6.15 `ldchlist` file

The `/etc/config/ldchlist` file is used by the `rc` system start-up script supplied with UNICOS to initialize the logical device cache to be used during normal system operation. To access this file through the ICMS, use the following menu selection:

```
Configure system
    ->Disk configuration
        ->logical device cache
```

For more information, see the description of `rc` on the `brc`(8) man page and in *General UNICOS System Administration*, Cray Research publication SG–2301.

6.16 `motd` file

The `/etc/motd` file (see `motd`(5)) is the UNICOS message-of-the-day file. It generally contains the UNICOS release level and any important messages that must be conveyed to each user as they log in. It is released containing a warning to unauthorized users. The following is the released version of the `/etc/motd` file:

```
cat /etc/motd
This is a private computer facility.   Access for any reason must be
specifically authorized by the owner. Unless you are so authorized,
your continued  access and any other use may  expose you to criminal
and/or civil proceedings.
```

6.17 `netstart` script

The `netstart` script is executed by the `rc` script to initialize UNICOS networking software. It starts each type of UNICOS networking software (for example, TCP/IP or OSI) by calling the appropriate startup script or command for that software type.

The list of scripts executed by `/etc/netstart` is as follows:

Command	Description
`/etc/netstart.pre`	The script to be run before `netstart` to accommodate local requirements.
`/etc/nwmstart`	Command used to initialize the underlying network media
`/etc/tcpstart`	Command to initialize TCP/IP software.
`/etc/nfsstart`	Command to initialize the network file system (NFS) software.

`/etc/ypstart`	Command to initialize network information service (NIS) software. (NIS was formerly known as yellow pages.)
`/etc/netstart.pst`	The script to be run after `netstart` to accommodate local requirements.

See the `netstart`(8) man page for additional information about the `netstart` script.

6.18 `netvar.conf` file

The `/etc/config/netvar.conf` file contains a list of command-line arguments for the `netvar`(8) command. These arguments are used by the `/etc/tcpstart` script supplied with UNICOS to initialize TCP/IP kernel variables at system startup. For a complete list of arguments to the `netvar`(8) command and further information about its operation, see the `netvar` man page and the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304.

To access this file through the ICMS, use the following menu selection:

```
Configure system
    ->Network configuration
        ->TCP/IP network configuration
            ->kernel parameters
```

6.19 passwd file

The `/etc/passwd` file contains one entry per user. Each entry consists of the login name, encrypted password (this field is set to an asterisk (`*`) or to `*,pw->pw_age` if password aging is active), user ID (UID), group ID (GID), a comment field, initial working directory, and the program the user uses as a shell.

The `/etc/passwd` file exists in UNICOS for compatibility with predecessor systems and plays no role in the validation and management of system users. That function is performed by the user database (UDB). For information on the relationship of `/etc/passwd` to the UDB, see *General UNICOS System Administration*, Cray Research publication SG–2301, and the `udb`(5), `udbgen`(8), and `udbsee`(1) man pages. For the format of `/etc/passwd` and more information on its content, see the `passwd`(5) man page.

6.20 `profile` file

If the `/etc/profile` file exists, it is executed by the standard shell upon login before your session begins. Then, if your login directory contains a file named `.profile`, it is executed by the shell before the session begins. The file `.profile` is useful for setting exported environment variables.

6.21 `rc` script and `rcoptions` file

The `rc` (see `brc`(8)) script is executed by the `init`(8) command using an entry in the `/etc/inittab` file (see `inittab`(5)) when changing the run level from single-user to multiuser mode.

To access this file through the ICMS, use the following menu selection:

```
Configure system
    ->Startup (/etc/rc) configuration
```

Note: The `rc` script is not intended to be edited directly. To modify the execution of this script, change the options in `/etc/config/rcoptions`. To perform tasks not controlled by `rcoptions`, create the following files:

- `/etc/rc.pre`
- `/etc/rc.mid`
- `/etc/rc.post`

These files are executed before, during, and after the `rc` script.

The UNICOS `rc` script is intended to provide a high degree of system startup configuration without modification of the script. This includes mounting file systems; activating accounting logging, error logging, and system activity logging; and starting system daemons. To do this, `rc` references a number of subsidiary configuration files and calls various configurable system utilities.

For a complete discussion of the capabilities and configuration of the `rc` script supplied with UNICOS, see the `brc`(8) man page and *General UNICOS System Administration*, Cray Research publication SG–2301.

The `/etc/config/rcoptions` file contains a list of environment variable definitions that control the startup configuration of the system. These definitions are read in by the `rc` (see `brc`(8)) script supplied with UNICOS. For a list of environment variables and explanation of their effect on system startup,

see the description of the `/etc/rc` script in *General UNICOS System Administration*, Cray Research publication SG–2301.

6.22 `shutdown` script

The `shutdown`(8) script is a part of the UNICOS operation procedure. It primarily terminates all currently running processes in an orderly and cautious manner. The procedure is as follows:

1. All users logged on the system are notified by a broadcast message to log off the system. You may display your own message at this time. Otherwise, the standard file save message is displayed.
2. All user processes are killed and daemons are shut down.
3. `shutdown` unmounts all file systems.

After the script completes, the dynamic blocks of all file systems are updated before the system is to be stopped (see `ldsync`(8) and `sync`(1)). This must be done before rebooting the system to ensure file system integrity.

After `shutdown` completes, the system is in single-user mode and is essentially equivalent to the state of the system after a reboot procedure. The release version of `shutdown` also shuts down the NQS daemon so that jobs can be recovered when the system reboots.

See *General UNICOS System Administration*, Cray Research publication SG–2301, for more information on the `shutdown` script.

6.23 `umountem` script

The `umountem`(8) script unmounts all mounted file systems except the current root file system. It is called by the `shutdown`(8) shell procedure to ensure that all file systems are truly unmounted before going to single-user mode. This script may also be called manually.