Chapter 7

System Accounting

This chapter contains the following sections;

- "Process (System) Accounting" on page 157 describes how to use the accounting utilities to keep track of system usage.
- "Additional Resources" on page 174 provides information on additional accounting software.

Process (System) Accounting

IRIX provides utilities to log certain types of system activity. These utilities perform *process accounting*.

The IRIX process accounting system can provide the following information:

- the number of programs a user runs
- the size and duration of user programs
- data throughput (I/O)

Using this information, you can:

- Determine how system resources are used and if a particular user is using more than a reasonable share.
- Trace significant system events, such as security breaches, by examining the list of all processes invoked by a particular user at a particular time.
- Set up billing systems to charge login accounts for using system resources.

The next sections describe the parts of process accounting, how to turn on and off process accounting, and how to look at the various log files.

Parts of the Process Accounting System

The IRIX process accounting system has several parts:

- The IRIX kernel writes a record of each process on the system that terminates into the file */var/adm/pacct*. The file contains one record per terminated process, organized according to the format defined in */usr/include/sys/acct.h.*

  You must specifically turn on this function. See "Turning on Process Accounting" on page 159.
- Once process accounting is turned on, the *cron* program executes several accounting commands, as specified in */var/spool/cron/crontabs/adm* and */var/spool/crontabs/root.* The commands in *adm* perform monthly accounting (*monacct*), check the size of the *pacct* file (*ckpacct*), and provide a daily accounting of processes and connect time (*runacct*). The *root* crontab file runs the *dodisk* program, which provides a report on current disk usage. These commands run automatically when process accounting is turned on.
- The *login* and *init* programs record connect sessions by writing records into */etc/wtmp*. This happens by default, as long as the *wtmp* file exists.
- Records of date changes, reboots, and shutdowns are copied from */etc/utmp* to */etc/wtmp* by the *acctwtmp* command.
- The *acctwtmp* utility is automatically called by *runacct*, */usr/lib/acct/startacct*, and */usr/lib/shutacct*, once process accounting is turned on.
- The disk utilization programs *acctdusg* and *diskusg* break down disk usage by login and prepare reports. For more information on disk usage quotas, see "IRIX Admin: Disks and Filesystems." These programs are run by the *dodisk* script.

Turning on Process Accounting

To turn on process accounting:

1. Log in to the system as *root*.
2. Enter this command:

   `chkconfig acct on`

3. Enter this command:

   `/usr/lib/acct/startup`

   This starts the kernel writing information into the file */var/adm/pacct*.

Process accounting is started every time you boot the system, and every time the system boots, you should see a message similar to this:

```
System accounting started
```

Note that process accounting files, especially */var/adm/pacct*, can grow very large. If you turn on process accounting, especially on a server, you should watch the amount of free disk space carefully. See "Controlling Accounting File Size" on page 160.

Turning Off Process Accounting

To turn off process accounting, follow these steps:

1. Log in as *root*.
2. Enter this command:

   `chkconfig off`

3. Enter this command:

   `/usr/lib/acct/shutacct`

   This stops the kernel from writing accounting information into the file */var/adm/pacct.*

Process accounting is now turned off.

Controlling Accounting File Size

Process and disk accounting files can grow very large. On a busy system, they can grow quite rapidly.

To help keep the size of the file */var/adm/pacct* under control, the *cron* command runs */usr/lib/acct/ckpacct* to check the size of the file and the available disk space on the file system.

If the size of the *pacct* file exceeds 1000 blocks (by default), it runs the *turnacct* command with argument "switch." The "switch" argument causes *turnacct* to back up the *pacct* file (removing any existing backup copy) and start a new, empty *pacct* file. This means that at any time, no more than 2000 blocks of disk space are taken by *pacct* file information.

If the amount of free space in the file system falls below 500 blocks, *ckpacct* automatically turns off process accounting by running the *turnacct* command with the "off" argument. When at least 500 blocks of disk space are free, accounting is activated again the next time *cron* runs *ckpacct*.

Accounting Files and Directories

The directory */usr/lib/acct* contains the programs and shell scripts necessary to run the accounting system. Process accounting uses a login (*/var/adm*) to perform certain tasks. */var/adm* contains active data collection files used by the process accounting. Here is a description of the primary subdirectories in */var/adm*:

/var/adm/acct/nite contains files that are reused daily by *runacct*.

/var/adm/acct/sum contains the cumulative summary files updated by *runacct*.

/var/adm/acct/fiscal contains periodic summary files created by *monacct*.

Daily Operation

When IRIX enters multiuser mode, */usr/lib/acct/startup* is executed as follows:

- The *acctwtmp* program adds a "boot" record to */etc/wtmp*. This record is signified by using the system name as the login name in the *wtmp* record.

- Process accounting is started by *turnacct*, which, in turn, executes *acct* on */var/adm/pacct*.
- *remove* is executed to clean up the saved *pacct* and *wtmp* files left in the *sum* directory by *runacct*.

The *ckpacct* procedure is run through *cron* every hour of the day to check the size of */var/adm/pacct*. If the file grows past 1000 blocks (default), the *turnacct* switch is executed. The advantage of having several smaller *pacct* files becomes apparent when you try to restart *runacct* after a failure processing these records.

The *chargefee* program can be used to bill users for file restores, and so on. It adds records to */var/adm/fee* that are picked up and processed by the next execution of *runacct* and merged into the total accounting records. *runacct* is executed through *cron* each night. It processes the active accounting files, */var/adm/pacct*, */etc/wtmp*, */var/adm/acct/nite/disktacct*, and */var/adm/fee.* It produces command summaries and usage summaries by login name.

When the system is shut down using *shutdown*, the *shutacct* shell procedure is executed. It writes a shutdown reason record into */etc/wtmp* and turns process accounting off.

After the first reboot each morning, the administrator should execute */usr/lib/acct/prdaily* to print the previous day's accounting report.

Setting Up the Accounting System

If you have installed the system accounting option, all the files and command lines for implementation have been set up properly. You may wish to verify that the entries in the system configuration files are correct. In order to automate the operation of the accounting system, you should check that the following have been done:

1. The file */etc/init.d/acct* should contain the following lines (among others):

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

   The first line starts process accounting during the system startup process; the second stops it before the system is brought down.

2. For most installations, the following entries should be in */var/spool/cron/crontabs/adm* so that *cron* automatically runs the daily accounting. These lines should already exist:

   ```
   0 4 * * 1-6 if /etc/chkconfig acct; then /usr/lib/acct/runacct 2>
   /var/adm/acct/nite/fd2log; fi
   ```

   ```
   5 * * * 1-6 if /etc/chkconfig acct; then /usr/lib/acct/ckpacct; fi
   ```

   Note that the above *cron* commands appear on one line in the source file. The following command, which is also all on one line in the source file, should be in */var/spool/cron/crontabs/root*:

   ```
   0 2 * * 4 if /etc/chkconfig acct; then /usr/lib/acct/dodisk >
   /var/adm/acct/nite/disklog; fi
   ```

3. To facilitate monthly merging of accounting data, the following entry in */var/spool/cron/crontabs/adm* allows *monacct* to clean up all daily reports and daily total accounting files, and deposit one monthly total report and one monthly total accounting file in the fiscal directory:

   ```
   0 5 1 * * if /etc/chkconfig acct; then /usr/lib/acct/monacct; fi
   ```

   The above command is all on one line in the source file, and takes advantage of the default action of *monacct* that uses the current month's date as the suffix for the file names. Notice that the entry is executed when *runacct* has sufficient time to complete. This will, on the first day of each month, create monthly accounting files with the entire month's data.

4. You may wish to verify that an account exists for *adm*. Also, verify that the PATH shell variable is set in */var/adm/.profile* to:

```
PATH=/usr/lib/acct:/bin:/usr/bin
```

5. To start up system accounting, simply type the commands

   `chkconfig acct on`

   and

   `/usr/lib/acct/startup`

The next time the system is booted, accounting starts.

runacct

runacct is the main daily accounting shell procedure. It is normally initiated by *cron* during nonpeak hours. *runacct* processes connect, fee, disk, and process accounting files. It also prepares daily and cumulative summary files for use by *prdaily* or for billing purposes. The following files produced by *runacct* are of particular interest:

nite/lineuse Produced by *acctcon*, reads the *wtmp* file and produces usage statistics for each terminal line on the system. This report is especially useful for detecting bad lines. If the ratio between the number of logoffs to logins exceeds about 3:1, it is quite possible that the line is failing.

nite/daytacct The total accounting file for the previous day in *tacct.h* format.

sum/tacct The accumulation of each day's *nite/daytacct* can be used for billing purposes. It is restarted each month or fiscal period by the *monacct* procedure.

sum/daycms Produced by the *acctcms* program. It contains the daily command summary. The ASCII version of this file is *nite/daycms*.

sum/cms The accumulation of each day's command summaries. It is restarted by the execution of *monacct*. The ASCII version is *nite/cms*.

sum/loginlog Produced by the last login shell procedure. It maintains a record of the last time each *login* name was used.

sum/rprtMMDD
Each execution of *runacct* saves a copy of the daily report that can be printed by *prdaily*.

runacct takes care not to damage files in the event of errors. A series of protection mechanisms are used that attempt to recognize an error, provide intelligent diagnostics, and terminate processing in such a way that *runacct* can be restarted with minimal intervention. It records its progress by writing descriptive messages into the file *active*. (Files used by *runacct* are assumed to be in the *nite* directory unless otherwise noted.) All diagnostics output during the execution of *runacct* are written into *fd2log*. *runacct* complains if the files *lock* and *lockl* exist when invoked. The *lastdate* file contains the month and day *runacct* was last invoked and is used to prevent more than one execution per day. If *runacct* detects an error, a message is written to */dev/console*, mail is sent to *root* and *adm*, locks are removed, diagnostic files are saved, and execution is terminated.

To allow *runacct* to be restartable, processing is broken down into separate reentrant states. A file is used to remember the last state completed. When each state completes, *statefile* is updated to reflect the next state. After processing for the state is complete, *statefile* is read and the next state is processed. When *runacct* reaches the CLEANUP state, it removes the locks and terminates. States are executed as follows:

SETUP — The command *turnacct* switch is executed. The process accounting files, */var/adm/pacct?*, are moved to */var/adm/Spacct?.MMDD*. The */etc/wtmp* file is moved to */var/adm/acct/nite/wtmp.MMDD* with the current time added on the end.

WTMPFIX — The *wtmpfix* program checks the *wtmp* file in the *nite* directory for correctness. Some date changes cause *acctcon1* to fail, so *wtmpfix* attempts to adjust the time stamps in the *wtmp* file if a date change record appears.

CONNECT1 — Connect session records are written to *ctmp* in the form of *ctmp.h.* The lineuse file is created, and the reboots file is created showing all of the boot records found in the *wtmp* file.

ctmp is converted to *ctacct.MMDD*, which are connect accounting records. (Accounting records are in *tacct.h* format.)

The *acctprc1* and *acctprc2* programs are used to convert the process accounting files, */var/adm/Spacct?.MMDD*, into total accounting records in *ptacct?.MMDD*. The *Spacct* and *ptacct* files are correlated by number so that if *runacct* fails, the unnecessary reprocessing of *Spacct* files will not occur. One precaution should be noted: when restarting *runacct* in this state, remove the last *ptacct* file, because it will not be complete.

MERGE — Merge the process accounting records with the connect accounting records to form *daytacct*.

FEES — Merge in any ASCII *tacct* records from the file *fee* into *daytacct*.

DISK — On the day after the *dodisk* procedure runs, merge *disktacct* with *daytacct*.

MERGETACCT
: Merge *daytacct* with *sum/tacct*, the cumulative total accounting file. Each day, *daytacct* is saved in *sum/tacctMMDD*, so that *sum/tacct* can be recreated in case it is corrupted or lost.

CMS
: Merge in today's command summary with the cumulative command summary file *sum/cms*. Produce ASCII and internal format command summary files.

USEREXIT
: Any installation-dependent (local) accounting programs can be included here.

CLEANUP
: Clean up temporary files, run *prdaily* and save its output in *sum/rprtMMDD*, remove the locks, then exit.

Recovering from a Failure

The *runacct* procedure can fail for a variety of reasons—usually due to a system crash, */usr* running out of space, or a corrupted *wtmp* file. If the *activeMMDD* file exists, check it first for error messages. If the *active* file and lock files exist, check *fd2log* for any mysterious messages. The following are error messages produced by *runacct* and the recommended recovery actions:

- `ERROR: locks found, run aborted`

  The files */var/adm/acct/nite/lock* and */var/adm/acct/nite/lock1* were found. These files must be removed before *runacct* can restart.

- `ERROR: acctg already run for date: check /var/adm/acct/nite/lastdate`

  The date in *lastdate* and today's date are the same. Remove *lastdate*.

- `ERROR: turnacct switch returned rc=?`

  Check the integrity of *turnacct* and *accton*. The *accton* program must be owned by root and have the *setuid* bit set.

- `ERROR: Spacct?.MMDD already exists`

  File setups probably already run. Check status of files, then run setups manually.

- `ERROR: /var/adm/acct/nite/wtmp.MMDD already exists, run setup manually`

  Self-explanatory.

- `ERROR: wtmpfix detected a corrupted wtmp file. Use fwtmp to correct the corrupted file.`

Self-explanatory.

- `ERROR: connect acctg failed: check /var/adm/acct/nite/log`

  The *acctcon1* program encountered a bad *wtmp* file. Use *fwtmp* to correct the bad file.

- `ERROR: Invalid state, check /var/adm/acct/nite/active`

  The file *statefile* is probably corrupted. Check *statefile* for irregularities and read *active* before restarting.

Restarting runacct

The *runacct* program, called without arguments, assumes that this is the first invocation of the day. The argument MMDD is necessary if *runacct* is being restarted and specifies the month and day for which *runacct* will rerun the accounting. The entry point for processing is based on the contents of *statefile*. To override *statefile*, include the desired state on the command line. For example, to start *runacct*, use the command:

```
nohup runacct 2 /var/adm/acct/nite/fd2log &
```

To restart *runacct*:

```
nohup runacct 0601 2 /var/adm/acct/nite/fd2log &
```

To restart runacct at a specific state:

```
nohup runacct 0601 WTMPFIX 2 /var/adm/acct/nite/fd2log &
```

Fixing Corrupted Files

Sometimes, errors occur in the accounting system, and a file is corrupted or lost. You can ignore some of these errors, or simply restore lost or corrupted files from a backup. However, certain files must be fixed in order to maintain the integrity of the accounting system.

Fixing wtmp Errors

The *wtmp* files are the most delicate part of the accounting system. When the date is changed and the IRIX system is in multiuser mode, a set of date change records is written into */etc/wtmp*. The *wtmpfix* program is designed to adjust the time stamps in the *wtmp* records when a date change is encountered. However, some combinations of date changes and reboots will slip through *wtmpfix* and cause *acctcon1* to fail.

The following steps show how to fix a *wtmp* file:

1. **cd /var/adm/acct/nite**
2. **fwtmp < wtmp.MMDD > xwtmp**
3. **ed xwtmp**
4. Delete any corrupted records or delete all records from beginning up to the date change.
5. **fwtmp –ic <wtmp> wtmp.MMDD**

If the *wtmp* file is beyond repair, remove the file and create an empty *wtmp* file:

6. **rm /etc/wtmp**
7. **touch /etc/wtmp**

This prevents any charging of connect time. *acctprc1* cannot determine which login owned a particular process, but it is charged to the login that is first in the password file for that user ID.

Fixing tacct Errors

If the installation is using the accounting system to charge users for system resources, the integrity of *sum/tacct* is quite important. Occasionally, mysterious *tacct* records appear with negative numbers, duplicate user IDs, or a user ID of 65,535. First check *sum/tacctprev* with *prtacct*. If it looks all right, the latest *sum/tacct.MMDD* should be patched up, then *sum/tacct* recreated. A simple patchup procedure would be:

1. Enter the command:

   cd /var/adm/acct/sum

2. Enter the command:

   acctmerg –v < tacct.MMDD > xtacct

3. Enter the command:

   ed xtacct

4. Remove the bad records.
5. Write duplicate UID records to another file.

6. Enter the command:

   acctmerg –i < xtacc t > tacct.MMDD

7. Enter the command:

   acctmerg tacctprev <tacct.MMDD> tacct

Remember that the *monacct* procedure removes all the *tacct.MMDD* files; therefore, you can recreate *sum/tacct* by merging these files.

Updating Holidays

The file */usr/lib/acct/holidays* contains the prime/nonprime table for the accounting system. The table should be edited to reflect your location's holiday schedule for the year. The format is composed of three types of entries:

- Comment Lines, which may appear anywhere in the file as long as the first character in the line is an asterisk.
- Year Designation Line, which should be the first data line (noncomment line) in the file and must appear only once. The line consists of three fields of four digits each (leading white space is ignored). For example, to specify the year as 1992, prime time at 9:00 a.m., and nonprime time at 4:30 p.m., the following entry is appropriate:

  ```
  1992 0900 1630
  ```

  A special condition allowed for in the time field is that the time 2400 is automatically converted to 0000.
- Company Holidays Lines, which follow the year designation line and have the following general format:

  ```
  day–of–year Month Day Description of Holiday
  ```

  The day-of-year field is a number in the range of 1 through 366, indicating the day for the corresponding holiday (leading white space is ignored). The other three fields are actually commentary and are not currently used by other programs.

Daily Reports

runacct generates five basic reports upon each invocation. They cover the areas of connect accounting, usage by person on a daily basis, command usage reported by daily and monthly totals, and a report of the last time users were logged in. The following paragraphs describe the reports and the meanings of their tabulated data.

In the first part of the report, the from/to banner should alert the administrator to the period reported on. This period runs from the time the last accounting report was generated until the time the current accounting report was generated. It is followed by a log of system reboots, shutdowns, power fail recoveries, and any other record dumped into */etc/wtmp* by the *acctwtmp* program. See the acct(1M) reference page for more information.

The second part of the report is a breakdown of line utilization. The TOTAL DURATION field tells how long the system was in multiuser state (able to be accessed through the terminal lines). The columns are:

LINE — The terminal line or access port.

MINUTES — The total number of minutes the line was in use during the accounting period.

PERCENT — The total number of minutes the line was in use divided into the total duration of the accounting period.

SESS — The number of times this port was accessed for a *login* session.

ON — This column has little significance. It previously gave the number of times that the port was used to log a user on; but since *login* can no longer be executed explicitly to log in a new user, this column should be identical with SESS.

OFF — The number of times a user logged off and also any interrupts that occur on that line. Generally, interrupts occur on a port when the *getty* is first invoked after the system is brought to multiuser state. This column comes into play when the # OFF exceeds the # ON by a large factor. This usually indicates that the multiplexer, modem, or cable is going bad, or that there is a bad connection somewhere. The most common cause of this is an unconnected cable dangling from the multiplexer.

During real time, */etc/wtmp* should be monitored, since this is the file from which connect accounting is geared. If it grows rapidly, execute *acctcon1* to see which line is the noisiest. If the interrupting is occurring at a furious rate, general system performance will be affected.

Daily Usage Report

The daily usage report gives a by-user breakdown of system resource utilization. Its data consists of:

UID
: The user ID.

LOGIN NAME
: The login name of the user; more than one login name can exist for a single user ID, and this entry identifies which login name used the resource.

CPU (MINS)
: The amount of time the user's process used the central processing unit. This category is broken down into PRIME and NPRIME (nonprime) utilization. The accounting system's idea of this breakdown is located in the */usr/lib/acct/holidays* file. As delivered, prime time is defined to be 0900 through 1700 hours.

KCORE-MINS
: A cumulative measure of the amount of memory a process uses while running. The amount shown reflects kilobyte segments of memory used per minute. This measurement is also broken down into PRIME and NPRIME amounts.

CONNECT (MINS)
: The amount of time that a user was logged into the system. If this time is high and # OF PROCS is low, this indicates that the user was logged in for a long period of time without actually using the system. This column is also subdivided into PRIME and NPRIME utilization.

DISK BLOCKS
: When the disk accounting programs have been run, the output is merged into the total accounting record (*tacct.h*) and shows up in this column. This disk accounting is accomplished by the program *acctdusg*.

OF PROCS
: The number of processes invoked by the user. Large numbers in this column indicate that a user may have had a shell running out of control.

O SESS
: Number of times the user logged onto the system.

DISK SAMPLES
: Number of times disk accounting was run to obtain the average number of DISK BLOCKS listed earlier.

FEE
: An often unused field in the total accounting record, the FEE field represents the total accumulation of widgets charged against the user by the *chargefee* shell procedure. See acctsh(1M). The *chargefee* procedure is used to levy charges against a user for special services performed such as file restores, and so on.

Daily Command and Monthly Total Command Summaries

These two reports are virtually the same except that the Daily Command Summary reports only on the current accounting period, while the Monthly Total Command Summary tells the story for the start of the fiscal period to the current date. In other words, the monthly report reflects the data accumulated since the last invocation of *monacct*.

The data included in these reports tells an administrator which commands are used most heavily. Based on those commands' characteristics of system resource utilization, the administrator can decide what to weigh more heavily when system tuning.

These reports are sorted by TOTAL KCOREMIN, which is an arbitrary yardstick but often a good one for calculating "drain" on a system.

COMMAND NAME
: The name of the command. Unfortunately, all shell procedures are lumped together under the name *sh* since only object modules are reported by the process accounting system. The administrator should monitor the frequency of programs called *a.out* or *core* or any other name that does not seem quite right. Often people like to work on their favorite version of a personal program, but they do not want everyone to know about it. *acctcom* is also a good tool for determining who executed a suspiciously named command and also to see if superuser privileges were abused.

NUMBER CMDS
: The total number of invocations of this particular command.

TOTAL KCOREMIN
: The total cumulative measurement of the amount of kilobyte segments of memory used by a process per minute of run time.

TOTAL CPU-MIN
: The total processing time this program has accumulated.

TOTAL REAL-MIN
: The total real-time (wall-clock) minutes this program has accumulated. This total is the actual "waited for" time as opposed to kicking off a process in the background.

MEAN SIZE-K
: The mean of the TOTAL KCOREMIN over the number of invocations reflected by NUMBER CMDS.

MEAN CPU-MIN
: The mean derived between the NUMBER CMDS and TOTAL CPU-MIN.

HOG FACTOR
: This gives a relative measure of the total available CPU time consumed by the process during its execution. It is a measurement of the ratio of system availability to system utilization. It is computed by the formula:

total CPU time / elapsed time

CHARS TRNSFD
: This column, which may contain a negative value, is a total count of the number of characters pushed around by the **read** and **write** system calls.

BLOCKS READ
: A total count of the physical block reads and writes that a process performed.

Files in the /var/adm Directory

The files listed here are located in the */var/adm* directory:

diskdiag diagnostic output during the execution of disk accounting programs

dtmp output from the *acctdusg* program

fee output from the *chargefee* program, ASCII *tacct* records

pacct active process accounting file

pacct? process accounting files switched by *turnacct*

Spact?.MMDD
: process accounting files for MMDD during execution of *runacct*

Files in the /var/adm/acct/nite Directory

The following files are located in the */var/adm/acct/nite* directory:

active	used by *runacct* to record progress and print warning and error messages. *activeMMDD* is the same as *active* after *runacct* detects an error
cms	ASCII total command summary used by *prdaily*
ctacct.MMDD	connect accounting records in *tawcct.h* format
ctmp	output of *acctcon1* program, connect session records in *ctmp.h* format
daycms	ASCII daily command summary used by *prdaily*
daytacct	total accounting records for one day in *tacct.h* format
disktacct	disk accounting records in *tacct.h* format, created by *dodisk* procedure
fd2log	diagnostic output during execution of *runacct* (see *cron* entry)
lastdate	last day *runacct* executed in date *+%m%d* format
lock lock1	used to control serial use of *runacct*
lineuse	tty line usage report used by *prdaily*
log	diagnostic output from *acctcon1*
logMMDD	same as log after *runacct* detects an error
reboots	contains beginning and ending dates from *wtmp* and contains a listing of reboots
statefile	used to record current state during execution of *runacct*
tmpwtmp	*wtmp* file corrected by *wtmpfix*
wtmperror	place for *wtmpfix* error messages
wtmperrorMMDD	same as *wtmperror* after *runacct* detects an error
wtmp.MMDD	previous day's *wtmp* file

Files in the /var/adm/acct/sum Directory

The following files are located in the */var/adm/acct/sum* directory:

cms	total command summary file for current fiscal period in internal summary format

cmsprev	command summary file without latest update
daycms	command summary file for yesterday in internal summary format
loginlog	created by *lastlogin*
pact.MMDD	concatenated version of all *pacct* files for *MMDD*, removed by remove procedure after reboot
rprtMMDD	saved output of *prdaily* programs
tacct	cumulative total accounting file for current fiscal period
tacctprev	same as *tacct* without latest update
tacctMMDD	total accounting file for *MMDD*
wtmp.MMDD	saved copy of *wtmp* file for *MMDD*, removed by remove procedure after reboot

Files in the /var/adm/acct/fiscal Directory

The following files are located in the */var/adm/acct/fiscal* directory:

cms?	total command summary file for *fiscal?* in internal summary format
fiscrpt?	report similar to *prdaily* for *fiscal?*
tacct?	total accounting file for *fiscal?*

Additional Resources

Ask your Silicon Graphics sales representative for information on additional tools available. For example, SHARE II™ for IRIX is an optional product allowing additional administrative control of system resources including disk space, CPU entitlement, memory (real or virtual), number of processes, printer pages, terminal and modem connect-time, network packets, and more.