NAME

runacct – run daily accounting

SYNOPSIS

/usr/lib/acct/runacct [*mmdd* [*state*]]

DESCRIPTION

runacct is the main daily accounting shell procedure. It is normally initiated using **cron**. **runacct** processes connect, fee, disk, and process accounting files. It also prepares summary files for **prdaily** or billing purposes. **runacct** is distributed only to source code licensees.

runacct takes care not to damage active accounting files or summary files in the event of errors. It records its progress by writing descriptive diagnostic messages into **active**. When an error is detected, a message is written to **/dev/console**, mail (see **mail**(1)) is sent to **root** and **adm**, and **runacct** terminates. **runacct** uses a series of lock files to protect against re-invocation. The files **lock** and **lock1** are used to prevent simultaneous invocation, and **lastdate** is used to prevent more than one invocation per day.

runacct breaks its processing into separate, restartable *states* using **statefile** to remember the last *state* completed. It accomplishes this by writing the *state* name into **statefile**. **runacct** then looks in **statefile** to see what it has done and to determine what to process next. *states* are executed in the following order:

SETUP Move active accounting files into working files.

WTMPFIX Verify integrity of **wtmp** file, correcting date changes if necessary.

CONNECT Produce connect session records in **tacct.h** format.

PROCESS Convert process accounting records into **tacct.h** format.

MERGE Merge the connect and process accounting records.

FEES Convert output of **chargefee** into **tacct.h** format, merge with connect, and process accounting records.

DISK Merge disk accounting records with connect, process, and fee accounting records.

MERGETACCT
Merge the daily total accounting records in **daytacct** with the summary total accounting records in **/var/adm/acct/sum/tacct**.

CMS Produce command summaries.

USEREXIT Any installation dependent accounting programs can be included here.

CLEANUP Clean up temporary files and exit.

To restart **runacct** after a failure, first check the **active** file for diagnostics, then fix any corrupted data files, such as **pacct** or **wtmp**. The **lock**, **lock1**, and **lastdate** files must be removed before **runacct** can be restarted. The argument *mmdd* is necessary if **runacct** is being restarted. *mmdd* 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 this, include the desired *state* on the command line to designate where processing should begin.

EXAMPLES

To start **runacct**:

example% nohup runacct 2> /var/adm/acct/nite/fd2log &

To restart **runacct**:

example% nohup runacct 0601 2>> /var/adm/acct/nite/fd2log &

To restart **runacct** at a specific *state*:

example% nohup runacct 0601 MERGE 2>> /var/adm/acct/nite/fd2log &

FILES

/var/adm/wtmp
/var/adm/pacct*incr*
/var/adm/acct/nite/active
/var/adm/acct/nite/daytacct
/var/adm/acct/nite/lock
/var/adm/acct/nite/lock1
/var/adm/acct/nite/lastdate
/var/adm/acct/nite/statefile

SEE ALSO

acctcom(1), **mail**(1), **acct**(1M), **acctcms**(1M), **acctcon**(1M), **acctmerg**(1M), **acctprc**(1M), **acctsh**(1M), **cron**(1M), **fwtmp**(1M), **acct**(2), **acct**(4), **utmp**(4)

NOTES

It is not recommended to restart **runacct** in the **SETUP** *state*. Run **SETUP** manually and restart using:

runacct *mmdd* **WTMPFIX**

If **runacct** failed in the **PROCESS** *state*, remove the last **ptacct** file because it will not be complete.

The **runacct** command can process a maximum of

6000 distinct sessions
1000 distinct terminal lines
2000 distinct login names

during a single invocation of the command. If at some point the actual number of any one of these items exceeds the maximum, the command will not succeed.