UniFLEX™ User/System Accounting Package

technical systems consultants, inc.

UniFLEX™ User/System Accounting Package

MANUAL REVISION HISTORY

Revision	Date	Change
A	3/84	Original Release

COPYRIGHT INFORMATION

DISCLAIMER

The supplied software is intended for use only as described in this manual. Use of undocumented features or parameters may cause unpredictable results for which Technical Systems Consultants, Inc. cannot assume responsibility. Although every effort has been made to make the supplied software and its documentation as accurate and functional as possible, Technical Systems Consultants, Inc. will not assume responsibility for any damages incurred or generated by such material. Technical Systems Consultants, Inc. reserves the right to make changes in such material at any time without notice.

Syntax Conventions

The following conventions are used in syntax statements throughout this manual.

1. Items that are not enclosed in angle brackets, ´<´ and ´>´, or square brackets, ´[´ and ´]´, are "keywords" and should be typed as shown.

2. Angle brackets, ´<´ and ´>´, enclose descriptions which the user must replace with a specific argument. Expressions enclosed only in angle brackets are essential parts of the command line. For example, in the command

   ```
   /etc/sysact [<file_name>]
   ```

   the name of a file must be specified in the place indicated by <file_name> if an argument is specified.

3. Square brackets, ´[´ and ´]´, indicate optional items. These items may be omitted if their effect is not desired.

4. The underscore character, ´_´, is used to link separate words that describe one term, such as "file" and "name".

5. Characters other than spaces that are not enclosed in angle brackets or square brackets must appear on the command line as they appear in the syntax statement.

6. If the word "list" appears as part of a term in a syntax statement, that term consists of one or more of the elements described by the rest of the term, separated by spaces. For example, the term

   ```
   <command_name_list>
   ```

   represents a list of command names.

7. Some utilities support optional features, known as options, which alter the effect of the command. Options consist of either a single character or a single character, followed by an equals sign, ´=´, followed by an argument. An "option string" is a plus sign followed by one or more options. An option string may contain any number of single-character options but only one option which takes an argument. An option requiring an argument must be the last option in an option string. Thus, the command line must contain a separate option string for each option requiring an argument. It may or may not contain a separate option string for each single-character option.

(continued)

The following are valid option strings:

```
+abcdefg
+abc=<arg>
```

The following are not valid option strings:

```
+abc=<arg>de
+a=<arg>b=<arg>
```

Unless specifically stated in the documentation about a particular command, option strings may appear anywhere on the command line after the command name.

Many common terms appear (often as abbreviations) in more than one syntax statement. The manual does not explain these terms each time they appear. However, the following table describes each one.

Table 1. Common Terms Used in Syntax Statements.

Term	Meaning
char	Character
command_name	A UniFLEX command
dev	Device
dir	Directory
splr	Spooler
str	String
file_name	A valid file name

Introduction

The UniFLEX User/System Accounting Package is a useful tool for keeping track of system resources. The package contains several utilities that produce reports that show the system manager how the system is being used. This manual contains a detailed description of each utility in the package. The introduction should give the system manager a general overview of the package and the knowledge required to use it effectively.

The utilities in this package can be divided into two groups: those which merely provide useful information to the system manager regarding the use of the system, and those which are vital to the user/system accounting feature. The first group consists of "cmd_sum", "list_acct", "sys_stats", "commanduse", "maxjobs", "usertime", and "history" (not in this package, but a standard UniFLEX utility); the second, of "setupact", "sysact", "splr_sum", "usr_sum", and "user_billing".

The commands in the first group perform some very useful functions and produce a number of interesting reports. Details of these commands are documented in this manual.

The commands in the second group summarize information about usage of the system, produce reports on this information, and produce bills based on each user's consumption of various system resources. The commands are generally used as a group.

User/system accounting is initiated by the execution of two commands: the "setupact" command, followed by the "sysact" command. The "setupact" command creates three files which are necessary for the collection of the accounting data.

The first file created is "/act/history". It collects records of login and logout times for each user. Collection of the data automatically begins as soon as the file is created.

The second file created is "/act/splrlog". It collects records of the activity of the UniFLEX Enhanced Printer Spooler. Collection of information about the activity of each spooler does not automatically begin when the file "/act/splrlog" is created. Rather, collection of data for each individual spooler begins when that spooler has been invoked after the file "/act/splrlog" has been created. Therefore, during the initial setup of the user/system accounting feature, the system manager must execute the command

```
/etc/insp <splr_name>
```

with the appropriate options for each spooler after executing the "setupact" command. Of course, an active spooler must be stopped by a "pstop" command before execution of the "insp" command.

If the system uses the standard UniFLEX spooler instead of the optional Enhanced Printer Spooler, the file "/act/splrlog" should not be created. An option can be specified to the "setupact" command to instruct it not to create "/act/splrlog".

The third file created by "setupact" is "/act/acct". This file is central to user/system accounting. It collects data about each task which is executed on the system. Collection of the data does not automatically begin when the file "/act/acct" is created. Rather, it begins when the "sysact" command is executed. Because the user/system accounting information is usually of interest only while UniFLEX is in multi-user mode, and because UniFLEX automatically stops collecting information in the file "/act/acct" when it leaves multi-user mode, the "sysact" command should be executed every time UniFLEX goes from single- to multi-user mode. The system manager can automate this process by placing the following command in the file "/etc/startup":

```
/etc/sysact /act/acct
```

Having this command in the start-up file ensures that the user/system accounting feature is enabled every time UniFLEX enters multi-user mode.

The billing program uses two additional files: a file of charges, "/act/charges", and a file containing information about each user, "/act/userinfo". The file of charges must exist for the proper operation of the billing program. The user-information file need not exist. A sample file of charges is provided on the master disk. The details of its contents are described in the documentation for "user_billing". To use this file of charges, the system manager must edit it to reflect the charges at the site in question. (All charges in the sample file are 0.) If bills are to contain only information on the use of system resources, not charges, the sample file of charges can be left as is, and the billing program will function correctly.

If the user-information file does exist, the billing program writes the information it contains about each user on the user's bill. If the file does not exist or if it exists but does not contain any information about a particular user, the billing program writes just the user name on the bill. A sample user-information file is provided on the master disk. This file can be edited to create a user-information file for any system. Details of and restrictions on this file are explained in the documentation for "user_billing".

Three commands--"cmd_sum", "splr_sum", and "usr_sum"--can be used to summarize accounting information. They may be used for the following reasons:

1. To summarize the data in a file containing current accounting information and thereby reduce the amount of storage space required for the file.

2. To provide the summarized accounting information required by the "user_billing" command.

3. To produce a printed report on a specific aspect of system usage.

In its most complex form, each of these three commands reads at least two files and produces two files. One of the input files contains current accounting information; the other contains a summary of previous accounting information. The "usr_sum" command reads a third input file, which contains the history of login and logout times for the current accounting period. One of the output files is a textual report of the information read from the input files; the other is a binary file of summary information which combines the information read from the input files. This new summary file can be used as an input file in future invocations of the command which produced it.

Numerous options modify the operation of the three summary commands. It is possible to use only one of the input files, to produce only one of the output files, to specify the names of input and output files instead of using the default names, and so forth. The descriptions of the individual commands, which follow this introduction, provide detailed information about the use of each command.

Any of the utilities which simply read the accounting files can be executed in multi-user mode. These utilities include "list_acct", "sys_stats", "usertime", "commanduse", "maxjobs", "user_billing", and "sysact" ("sysact" used without an argument disables the accounting feature). The "setupact" command cannot be executed in multi-user mode. The remaining utilities can be executed in multi-user mode only when they neither delete a current accounting file nor update a summary file.

It is recommended that the system manager set up some kind of schedule for user billing and system accounting in order to prevent the accounting files from becoming too large. Such a schedule is especially necessary on extremely busy systems and on systems with a limited amount of disk space. Once the needs of a particular site have been determined, the system manager can write a shell script to perform all the necessary accounting functions.

The following shell script can be used to update year-to-date summary files, to produce bills for and a summary of the current accounting period, and to start a new accounting period. The user will better understand this example after reading the entire manual. It is presented here to give an indication of the power of the accounting package.

```
echo "BEGIN USER/SYSTEM ACCOUNTING PROCEDURE."
echo "BACKUP UP CURRENT ACCOUNTING FILES."
/etc/mount /dev/fd0 /usr0
chd /act
copy acct history splrlog /usr0
echo "UPDATING YEAR-TO-DATE SUMMARY FILES."
chd /etc
cmd_sum +s=/act/cmdsum.ytd
usr_sum +s=/act/usrsum.ytd
splr_sum +SUI=/act/splrsum.ytd +O=/act/splrsum.ytd
echo "SUMMARIZING CURRENT ACCOUNTING FILES"
echo "AND PRODUCING REPORT FILES."
sys_stats +c >/act/stats_rpt
cmd_sum +l >/act/cmd_rpt
usr_sum +l >/act/usr_rpt
splr_sum +O >/act/splr_rpt
echo "PRODUCING USER_BILLING REPORT FILE."
user_billing
echo "BACKING UP CURRENT AND YEAR-TO-DATE SUMMARIES."
chd /act
copy splrsum splrsum.ytd usrsum usrsum.ytd cmdsum cmdsum.ytd /usr0
date >/usr0/backup.date
/etc/unmount /dev/fd0
echo "DELETING CURRENT SUMMARIES AND"
echo "STARTING A NEW ACCOUNTING PERIOD."
kill splrsum usrsum cmdsum
/etc/setupact
echo "FINISHED.  REMOVE FLOPPY FROM DRIVE."
```

Before executing this shell script the user should place a formatted floppy disk in drive 0. This disk is used to back up all the accounting files for the current accounting period. After copying the current accounting, history, and spooler-log files onto the floppy disk, the shell script uses the commands "cmd_sum", "usr_sum", and "splr_sum" to update the year-to-date summary files (those files whose names end in ".ytd"). The year-to-date summary files must exist prior to the first execution of this shell script.

The next step is to create reports from the information in the current accounting files. This step must be separate from updating the year-to-date files because the summaries and bills for the current accounting period should not include the information in the year-to-date summary files. By default the reports from all the programs except "user_billing" are sent to standard output. Therefore, the shell script redirects the output from these commands to the appropriate report files in the directory "/act". The "user_billing" command sends its output to the file "/act/billing_rpt".

After producing these reports the shell script copies the new summary files and the year-to-date summary files to the floppy disk for backup. The "date" command produces a file containing a timestamp on the floppy. The summary files just produced are then deleted from the system disk. Finally, the "/act/setupact" command, which clears all the accounting files (prompting the user before clearing each one since they probably have data in them), starts a new accounting period.

The backups created by this shell script can be useful in creating different types of reports or more detailed reports from past accounting periods. The documentation for "usr_sum", "cmd_sum", and "splr_sum" explains the possibilities for different kinds of reports. The system manager should thoroughly read the documentation for each utility before setting up a schedule for user billing and accounting. Because "splr_sum", "cmd_sum", and "usr_sum" can clear accounting files, care should be taken to avoid clearing with one command a file that another command needs. Good backups of these files are essential to avoiding problems. Because the "/act/setupact" command clears all the accounting files, it should not be necessary to use any other command to clear them.

It is inadvisable to delete users in the middle of an accounting period because the accounting records identify each user by user number, whereas the history file identifies each user by name. The "usr_sum" command uses the password file to combine the information contained in these files. If a user is not in the password file, "usr_sum" cannot function properly. Therefore, the report produced by "user_billing", which reads the summary file produced by "usr_sum", will be incorrect for users not in the password file. The system manager can effectively delete a user by changing that user´s password. Then the user can actually be deleted at the end of an accounting period.

Instructions follow for the step-by-step installation and use of the User/System Accounting Package on a typical UniFLEX system. While many variations on the use of this package are possible, this procedure should provide a good starting point for most UniFLEX installations.

User/system accounting requires a certain amount of initial setup, followed by the periodic generation of reports and billings. The steps required for the initial setup are as follows:

1. Boot UniFLEX, remaining in single-user mode.

2. Edit the files "/act/charges" and "/act/userinfo" to reflect the appropriate charges and user information.

3. Use the editor to create a shell script whose contents are identical to the sample accounting script provided earlier in this introduction.

4. Create the three year-to-date summary files needed by the accounting script with the following commands:

```
create /act/cmdsum.ytd
create /act/usrsum.ytd
create /act/splrsum.ytd
```

   These files are all empty files.

5. Execute the command "/etc/setupact" in order to create the files that collect the accounting information. If the system does not have the UniFLEX Enhanced Printer Spooler, use the command "/etc/setupact +s" so that the file "/act/splrlog" is not created.

6. Stop each active spooler with a "pstop" command and reinvoke each spooler with an "insp" command. These commands allow the collection of information about the activity of the enhanced printer spooler to begin. This step should be skipped if the system does not have the UniFLEX Enhanced Printer Spooler.

7. Add the command "/etc/sysact /act/acct" to the multi-user startup file, "/etc/startup". This command initiates the collection of accounting data each time the system enters multi-user mode.

8. The initial setup is now complete. User/system accounting will begin as soon as the system enters multi-user mode (a "log" command is executed from single-user mode).

The preceding steps need be taken only once. After the initial setup is complete, the necessary routine accounting procedures can be performed by running the accounting script at the end of each accounting period. The accounting script must be executed from single-user mode. The length of that period is at the discretion of the system manager. It should be influenced by how often summaries and bills are desired and by how quickly the data files grow. After the accounting script is run, the file containing the user bills ("/act/billing_rpt") and the report files generated by the script (the files in "/act" which end with "_rpt") can be routed to the appropriate printer.

At the end of each year of accounting, the system manager should perform step 4 to truncate the year-to-date summary files and, thus, begin a new year of data collection.

cmd_sum

Produce a summary of command usage.

SYNTAX

```
    /etc/cmd_sum [+a=<file_name_1>] [+p=<number_1>] [+s=<file_name_2>]
                 [+t=<number_2>] [+cdlnsu]
```

DESCRIPTION

By default, "cmd_sum" reads information about command usage from the current accounting file, "/act/acct", and from the file containing a summary of previous command usage, "/act/cmdsum". If a summary of previous command usage does not exist, "cmd_sum" reads only the current accounting file. This information is summarized and written to the summary file, "/act/cmdsum", replacing the existing contents of that file or, if necessary, creating the file. An option exists which sends a textual report of the summarized information to standard output. Other options permit the user to specify input and output files other than the default files. There are also options which instruct "cmd_sum" to obtain its input only from the current accounting file or only from the file containing a summary of command usage. Thus, usage statistics may be computed from the current accounting period, from previous accounting periods, or both.

The statistics gathered include the total number of times a given program was executed, the total amount of CPU time used by the program, the amount of memory used by the program, and the amount of I/O activity performed by the program.

Arguments

a=<file_name_1> The name of the file containing the accounting information. Default is "/act/acct".

p=<number_1> The number of lines per page in the report. Default is 66. Minimum is 11 because a 5-line margin appears at the top and bottom of each page.

s=<file_name_2> The name of the file from which to obtain the summary of command usage and, if the 'n' option is not specified, to which to write the updated summary information. Default is "/act/cmdsum".

t=<number_2> The number of ticks per second, which is determined by the hardware. Default is 10. At the time of this writing, all existing 6809 UniFLEX systems use the default (10).

(continued)

- This program can be used in conjunction with the programs "sys_stats", "usr_sum", and "user_billing". Since both this program and "usr_sum" may actually delete or truncate the accounting file, care must be taken in selecting the order in which these programs are executed, as well as the options that are used with them.

- A list of commands that might commonly be used to produce both summaries of the various aspects of system usage and bills for this usage appears in the introduction to this manual.

ERROR MESSAGES

Unknown option: "<char>"
The character <char> appeared within an option string but is not recognized as a valid option.

Illegal accounting file specification
The format of the specification of the accounting file is illegal. The format must be "a=<file_name>".

Illegal summary file specification
The format of the specification of the summary file is illegal. The format must be "s=<file_name>".

Unrecognizable argument, "<str>"
The program did not recognize the command-line argument.

Incompatible options: (+c and +s) or (+n and +d)
The options ´c´ (use current accounting file only) and ´s´ (use previous summary file only) may not both be specified. The options ´n´ (do not update the summary file) and ´d´ (delete the accounting file after use) may not both be specified. This incompatibility is designed to prevent the inadvertent deletion of the information in the accounting file without updating the summary file. If this effect is desired, the user must manually delete the accounting file.

Page length too short.
The number specifying the page length is less than the minimum allowed.

Must be in single-user mode
The program does not update the summary file or delete the accounting file unless the system is in single-user mode.

Unable to access accounting file: "<file_name>"
The program could not access the specified accounting file. An auxiliary message indicates the reason.

(continued)

commanduse

Report the amount of time each user has spent executing a specified command.

SYNTAX

```
/etc/commanduse <command_name> [+a=<file_name>]
```

DESCRIPTION

Given the name of a command, "commanduse" searches an accounting file and produces a report on the usage of that command. This report consists of the name of every user who has executed the specified command and the total elapsed real time each user spent running the command. If a user is no longer in the password file, "commanduse" reports the user ID number rather than the user name. If no accounting file is specified, it defaults to "/act/acct".

Arguments

<command_name>	The name of the command to report on.
<file_name>	The name of the file containing the accounting information. Default is "/act/acct".

EXAMPLES

1. /etc/commanduse basic
2. /etc/commanduse adventure +a=/act/acct.2-84

The first example reports on the usage of "basic" for each user. It bases the report on the data in the file "/act/acct".

The second example reports on the usage of "adventure". It bases the report on the data in the file "/act/acct.2-84".

ERROR MESSAGES

Cannot open accounting file ´<file_name>´.
The file specified could not be found or opened.

Unknown option: "<char>".
The option specified by <char> is not a valid option to "commanduse".

(continued)

Options Available

c Use the information from the current accounting file only. Implies the ´n´ option; incompatible with the ´s´ option.
d Delete the accounting file after use. Incompatible with the ´n´ option.
l Produce a listing of the summary data.
n Do not update the summary file.
s Use information from the summary file only. Do not use the accounting file.
u Sort commands in order of usage with the most frequently used command first.

EXAMPLES

```
1. /etc/cmd_sum
2. /etc/cmd_sum +a=/usr0/act/acct +lc
3. /etc/cmd_sum +lc
4. /etc/cmd_sum +d
```

The first example processes the accounting file "/act/acct" and the command-summary file "/act/cmdsum". It then uses the totals, broken down by command, to update the command-summary file "/act/cmdsum". This example produces no printed output.

The second example prints a summary of command usage using only the accounting file "/usr0/act/acct" (i.e., a file on a disk which has been mounted as "/usr0"). No previous summary information is read, nor is the summary file updated (the ´c´ option implies the ´n´ option).

The third example is similar to the second, but it uses the default accounting file "/act/acct". This command is useful for producing a report about the current accounting period only.

The final example processes the current accounting file "/act/acct" and the command-summary file "/act/cmdsum". It first updates the summary file with the new data, then truncates the system accounting file. Thus, a new accounting period is started.

NOTES

- The format for the arguments to this command is similar to the format for the options. However, each individual argument must be preceded by a plus sign; an argument cannot be used as part of an option string.

(continued)

list_acct

List information from an accounting file.

SYNTAX

```
/etc/list_acct [+a=<file_name>] [+n=<number_1>] [+p=<program_name>]
               [+t=<number_2>] [+u=<user_name>]
```

DESCRIPTION

The "list_acct" command sends a detailed list of the information in an accounting file to standard output. By default "list_acct" prints one line for each transaction in the file. Each transaction includes the name of the program executed, the total CPU time used by the program, the maximum amount of memory used by the program, and the time the program began execution. The user may choose to restrict "list_acct" to the data for a single user, to the data for a single program, or both. The program automatically paginates its output, but the user can alter the number of lines per page.

Arguments

a=<file_name> The name of the file containing the accounting information. Default is "/act/acct".

n=<number_1> The number of lines per page in the report. Default is 66. Minimum is 11 because a 5-line margin appears at the top and bottom of each page.

p=<program_name> Restrict output to the data for the specified program.

t=<number_2> The number of ticks per second, which is determined by the hardware. Default is 10. At the time of this writing, all existing 6809 UniFLEX systems use the default (10).

u=<user_name> Restrict output to the data for the specified user.

EXAMPLES

```
1. /etc/list_acct
2. /etc/list_acct +n=30
3. /etc/list_acct +u=system
4. /etc/list_acct +u=system +p=edit
```

The first example prints a report of each transaction in the accounting file "/act/acct".

(continued)

Unable to access summary file "<file_name>"
The program could not access the specified file. An auxiliary message indicates the reason.

SEE ALSO

setupact
sysact

maxjobs

Report the total task count for the day and compare it with the current maximum task count.

SYNTAX

```
     maxjobs
```

DESCRIPTION

The "maxjobs" command reports the total number of tasks the system has run since the last boot time. If the command is executed by the system manager, it also reports the current maximum number of tasks and the date on which it occurred. If the value returned by "maxjobs" is greater than the current maximum, the information is updated to reflect the new maximum. The system manager should run this command each day just before shutting down the system in order to maintain an accurate record of the maximum task count. No options are supported by this command.

EXAMPLES

```
    maxjobs
```

This example is the only valid form of the "maxjobs" command. It reports the total number of tasks run since the last boot time and, if the user is the system manager, the current maximum number of tasks and the date on which it occurred.

NOTES

- The information about the maximum task count is kept in the file "/etc/log/tasks.data". This file, which is automatically created by "maxjobs", is in binary form.

ERROR MESSAGES

Usage: maxjobs
The user specified an argument or option to the "maxjobs" command, which accepts neither.

Cannot create "/etc/log/tasks.data".
The operating system returned an error when "maxjobs" tried to create the file "/etc/log/tasks.data".

(continued)

SEE ALSO

setupact
sysact
usertime

setupact

Create the files necessary for user/system accounting.

SYNTAX

```
/etc/setupact [+s]
```

DESCRIPTION

The "setupact" command creates the three files required for the proper operation of the User/System Accounting Package: the file "/act/acct", which contains all the information about system accounting; the file "/act/history", which contains a record of login and logout times; and the file "/act/splrlog", which is used by the Enhanced Printer Spooler to record spooler activity. If any of these files already exist and contain data, "setupact" displays a message to that effect. The user then has the choice of deleting the contents of such a file or leaving the file untouched. Because systems without the enhanced spooler do not need the file "/act/splrlog", an option is available to specify that this file not be created.

Only the system manager may execute this command. The system must be in single-user mode.

Options Available

s   Do not create the spooler-log file, "/act/splrlog".

EXAMPLES

```
/etc/setupact
```

This example creates the three files necessary to the system accounting feature.

ERROR MESSAGES

Usage: /etc/setupact [+s]
The user specified more than one option string or argument, or an option string that did not start with ´+´. The only allowable option string is "+s".

Illegal option ´<char>´.
The user specified an argument or option other than ´s´, the only allowed option.

(continued)

The second example produces the same report as the first example, but the body of each page of the report contains thirty lines.

The third example prints a report of every transaction for the user "system". It ignores transactions for other users.

The fourth example prints each "edit" transaction executed by the user "system". This combination of options allows the extraction of data about how (and how much) a particular user uses a given program.

ERROR MESSAGES

Illegal accounting file specification
The format of the specification of the accounting file is illegal. The format must be "a=<file_name>".

Unrecognizable argument, "<str>"
The program did not recognize the command-line argument.

Unable to access accounting file: "<file_name>"
The program could not access the specified accounting file. An auxiliary message indicates the reason.

Page length too short.
The number specifying the page length is less than the minimum allowed.

SEE ALSO

setupact
sysact

splr_sum

Write a report summarizing spooler usage to standard output.

SYNTAX

```
/etc/splr_sum [+acfoprsuIORSU]
```

DESCRIPTION

By default "splr_sum" reads the spooler-log file "/act/splrlog" (a file containing information on all spooler transactions), which was produced by the Enhanced Printer Spooler. It produces a textual report of summarized spooler usage and sends it to standard output. One option instructs "splr_sum" to write a condensed form of the input information to a binary file summarizing spooler usage (a spooler-summary file). Another option instructs it to use a previously produced spooler-summary file as input in addition to the spooler-log file. The spooler-summary file, which is used by "user_billing", contains information about the files printed by each user on each kind of form on each spooler.

Numerous options allow the user to choose from a variety of reports based on only the current spooler-log file, both the current spooler-log file and the spooler-summary file, or just the spooler-summary file. The report which is produced by default contains the total number of files, pages, lines, and characters that each user printed on each form on each spooler (user totals) as well as the total number of files, pages, lines, and characters printed on each form on each spooler by all users combined (spooler totals). If the user specifies the ´r´ option, "splr_sum" produces a more detailed report, which contains the total number of pages, lines, and characters in each file printed.

Options Available

a=<file_name> The name of the spooler-log file to use. Default is "/act/splrlog".

c Clear the file "/act/splrlog" at the end of the report.

f=<form_num> Report only on the specified form number.

o Report only on the spooler-summary file. Do not use data from the current spooler-log file.

p=<page_len> The number of lines per page in the report. Default is 66. Minimum is 11 because a 5-line margin appears at the top and bottom of each page.

r Print detailed report.

s=<splr_name> Report only on the specified spooler.

u=<user_name> Report only on the specified user.

(continued)

```
Error reading "/etc/log/tasks.data".
  The "maxjobs" command could not read the file "/etc/log/tasks.data".
```

NOTES

- The standard UniFLEX printer spooler cannot maintain the spooler-log file. Thus, the "splr_sum" command only functions on systems which have the Enhanced Printer Spooler.
- The ´c´ option always clears the file "/act/splrlog", even if input comes from another file.
- Records for users not in the password file are identified by user ID in the report.

ERROR MESSAGES

Unknown option ´<char>´.
An unrecognizable option was found. This error is fatal.

Option syntax error: ´<str>´.
An option string must begin with a plus sign.

Error creating ´<file_name>´ : <reason>
The operating system returned an error when "splr_sum" tried to create the specified file.

Error opening ´<file_name>´ : <reason>
The operating system returned an error when "splr_sum" tried to open the specified file.

Error reading ´<file_name>´ : <reason>
The operating system returned an error when "splr_sum" tried to read the specified file.

Error writing ´<file_name>´ : <reason>
The operating system returned an error when "splr_sum" tried to write the specified file.

Out of space!
The "splr_sum" command uses dynamic storage for the collection of data because it cannot know in advance how many users or how many spoolers are on a system. Realistically, the space available should handle hundreds of users and numerous spoolers; no preset limits on the number of users or the number of combinations of spoolers and form numbers exist. If this message occurs, the user can execute the command with the ´R´ option, which restricts processing to the range of user IDs specified.

Page length too short.
The number specifying the page length is less than the minimum allowed.

(continued)

You must be system manager to run this command.
  Only the system manager may execute the "setupact" command.

The system must be in single user mode.
  The command was executed in multi-user mode.

The file ´<file_name>´ already exists and contains data.
Do you want the file truncated?
  The file specified is not empty. If the first character of the user´s response is a ´y´, "setupact" recreates the file, thus removing all the data. Otherwise, the file remains intact.

The file ´/act/splrlog´ exists.
Do you want the file deleted?
  The ´s´ option was specified, but the file "/act/splrlog" exists. If the first character of the user´s response is a ´y´, "setupact" deletes the file. Otherwise, the file is left intact.

SEE ALSO

sysact
user_billing

sys_stats

Print statistics about system usage.

SYNTAX

```
     /etc/sys_stats [+h=<file_name>] [+c]
```

DESCRIPTION

The "sys_stats" command gathers statistics from a file containing data on the history of system use and sends them to standard output. The statistics normally gathered are total system up time, total system down time, total time spent in single-user mode, total time spent in multi-user mode, total login time per user, and number of logins per terminal. This command assumes that the system manager always sets the correct time immediately after booting UniFLEX.

The user may request "sys_stats" to include in its report the time of each system crash which has occurred since the creation of the history file. A crash is defined as the lack of a proper shut-down sequence, which should consist of a "shutup" command to take UniFLEX from multi- to single-user mode and a "stop" command to exit UniFLEX. Thus, if the user performs a hardware reset from multi-user mode, "sys_stats" records that event as a system crash.

Arguments

h=<file_name> The name of the history file to use to compute the statistics. Default is "/act/history".

Options Available

c Print a message indicating the time of each system crash which has occurred since the creation of the history file.

EXAMPLES

```
    1. /etc/sys_stats
    2. /etc/sys_stats h=/usr0/act/history +c
```

The first example prints the statistics from the system history file "/act/history".

(continued)

I[=<file_name>] The name of the spooler-summary file containing the data to use in calculations and in totals in any reports. Default is "/act/splrsum".

O[=<file_name>] Create a new spooler-summary file from the data in the input file or files. Default is "/act/splrsum".

R=<ID_1-ID_2> Report on this range of user ID numbers.

S Do not print spooler totals in the report.

U Do not print user totals in the report.

EXAMPLES

```
1.  /etc/splr_sum
2.  /etc/splr_sum +cSUO
3.  /etc/splr_sum +a=/act/splrlog.1.4 +u=joe +rs=spr
4.  /etc/splr_sum +IOc
5.  /etc/splr_sum +oUR=12-30
6.  /etc/splr_sum +I=/act/splr_sum.2.4 +O=/act/splr_sum.3.4
```

The first example produces a report of user totals and spooler totals based on the data in the file "/act/splrlog".

The second example does not produce a report. It writes a new summary file, "/act/splrsum", based on the data in the spooler-log file and clears the file "/act/splrlog" after reading it.

The third example produces both a detailed report and a report of user totals and spooler totals based on the data in the file "/act/splrlog.1.4". The reports are restricted to the activity of the user "joe" on the spooler "spr".

The fourth example creates a new spooler-summary file, "/act/splrsum", which combines the data from the old spooler-summary file and the current spooler-log file. It writes a report to standard output, then clears the spooler-log file.

The fifth example writes a report based on the data in the current spooler-summary file, "/act/splrsum". The report, which is restricted to the activity of users 12 through 30, contains only spooler totals.

The sixth example produces a report based on the data in both the current spooler-log file, "/act/splrlog", and the specified spooler-summary file, "/act/splrsum.2.4". It then combines the data from these two input files and writes it to a new summary file, "/act/splrsum.3.4".

(continued)

sysact

Enable or disable the system accounting feature.

SYNTAX

```
/etc/sysact [<file_name>]
```

DESCRIPTION

The "sysact" command controls the user/system accounting feature of the operating system. Only the system manager may execute this command. When accounting is enabled, the system maintains a file containing information about every task executed on the system. This information includes the starting and stopping date and time of the command, the name of the user who executed it, the terminal from which it was executed, and the amount of system resources consumed (CPU time and memory).

The system manager enables accounting by executing the "sysact" command with the name of the accounting file as the argument to the command. The specified file must already exist. Normally, it is the file "/act/acct". If the user does not specify a file, "sysact" disables the accounting feature. This command should be executed just before entering multi-user mode. The best way to accomplish this is to make an entry in the file "/etc/startup" which looks like this:

```
/etc/sysact /act/acct
```

The startup procedure then enables the accounting feature every time UniFLEX enters multi-user mode. UniFLEX automatically disables the accounting feature when it enters single-user mode.

Arguments

<file_name>   The name of the file to use as the accounting file.

EXAMPLES

```
1. /etc/sysact /act/acct
2. /etc/sysact
```

The first example enables the system accounting feature and stores the information in the file "/act/acct".

The second example disables the system accounting feature.

(continued)

SEE ALSO

UniFLEX Enhanced Printer Spooler
setupact
user_billing

user_billing

Produce a report containing bills for each user on the system.

SYNTAX

```
/etc/user_billing [+cdmnoprstuvCRSU]
```

DESCRIPTION

The "user_billing" command produces bills for each user on the system, based on the information contained in three files: a spooler-summary file, a user-summary file, and the file of charges, "/act/charges". The file of charges contains information on how much to charge each user per unit of each system resource. By default "user_billing" also reads information about each user from a file called "/act/userinfo".

The bills are written to a file called "/act/billing_rpt", which can later be printed. The information on each bill consists of connect time, I/O units, CPU time, paper usage, and K-Minutes (the maximum size of a task in Kbytes multiplied by the amount of CPU time in minutes the task used). Each bill also contains total charges for each category, as well as the total of all charges.

The options allow many variations of the billing report to be printed. The command can produce a report on resource usage only or a report that summarizes the total charges to all users in each category. It can also produce a verbose report, which lists connect time and charges for each terminal, as well as paper usage and charges by spooler and form number. A condensed listing, which lists only the total charges for each user, can also be produced.

Options Available

c=<cur_sym> The currency symbol to use. This symbol may contain up to twenty characters. Default is ´$´.

d Read and display the file "/act/charges". If the user specifies this option anywhere in the option string, "user_billing" displays the file, then terminates.

m Suppress printing of the user multiplier in the bill (see File of Charges).

n Do not read "/act/userinfo". Identify user only by user name on bills.

o=<rpt_file> The name of the file to contain the billing report. Default is "/act/billing_rpt".

p=<bill_len> Print bills with <bill_len> lines per bill. Default is 12. Minimum is 12.

(continued)

The second example prints the statistics from the file "/usr0/act/history" (i.e., a file on a disk which has been mounted as "/usr0"). In addition it prints a message indicating the time of each system crash recorded in the history file.

NOTES

- The format for the argument to this command is similar to the format for the option. However, the argument must always be preceded by a plus sign; it cannot be used as part of an option string.
- UniFLEX gathers no system statistics unless the file "/act/history" exists. The system manager should create this file in order to enable the statistics-gathering feature.
- The execution of "sys_stats" does not alter the contents of the specified history file. The history file must be recreated to begin a new collection period.

ERROR MESSAGES

Illegal history file specification "<file_name>"
The format of the specification of the history file is invalid. The format must be "h=<file_name>".

Illegal option "<str_or_char>"
A command-line argument could not be parsed into one of the available options or arguments.

Can´t access history file "<file_name>"
The program could not access the specified history file. An auxiliary message indicates the reason.

Warning: Some records in the history file are invalid.
If the format of one or more records in the specified history file is not correct, "sys_stats" ignores those records during processing and prints this message prior to printing the statistics. As indicated, this message is a warning to the user that the data collected may be invalid.

SEE ALSO

history
setupact

```
[1] Connect Charges (per minute)
 0  0.05
 03 0.025
 *  0.01
[2] I/O Units Charges (per 1000 units)
    0.5
[3] CPU Time Charges (per second)
   0.0075
[4] M-Second Charges (per 1000)
   .5
[5] Paper Charges (per page)
 spr 10 0.30
 ppr 1 0.005
 spr 0 0.01
 *  0 0.02
 ppr * 0.03
 * * 0.001
[6] Storage Units Charges (per block, per month)
 0.0
[7] User Id Multipliers
 30 .5
 100 1.1
 450 1.3
```

The file of charges contains seven separate sections. With one exception, explained later, the sections may appear in any order, but the number representing each section is fixed. That is, section 1 always contains the connect charges, section 2 always contains the I/O charges, and so forth. All that is required to mark the start of a section is a left-hand square bracket, ´[´, followed immediately by the number representing that section. The "user_billing" program ignores all other characters on the line. Every section except section 7, which contains the user-ID multipliers, must contain a charge even if the amount of the charge is 0. All blank lines and leading spaces in the file of charges are ignored.

Section 1 contains the charges for connect time. Each entry in this section consists of a terminal number, followed by the charge per minute of connect time for that terminal. The terminal number can be one or two digits. An asterisk, ´*´, in the field for the terminal number refers to all terminals not already specified. The last charge in the section must be an asterisk, followed by a charge, even if the list of terminals is complete. The maximum number of terminals allowed is forty-eight.

Section 2 details the charges for I/O units. In this section the "user_billing" command expects to find exactly one entry, which is the charge per thousand I/O units. I/O units are charged to a task based on the number of read and write system calls it executes. Because I/O activity is slow compared with CPU speed, this number really is a measure of how much impact a particular user´s tasks had on overall

(continued)

ERROR MESSAGES

No permission.
The user is not the system manager. Only the system manager may execute this command.

Accounting was previously enabled.
An attempt was made to enable accounting when it was already enabled.

The specified file does not exist.
The accounting file specified could not be found. The system manager must create this file before enabling accounting.

The specified file is not a regular file.
The specified file was a directory or device.

SEE ALSO

cmd_sum
commanduse
list_acct
setupact
user_billing
usertime
usr_sum

Section 7 contains user-ID multipliers. These multipliers allow the system manager to charge individuals or groups of users different amounts for system resources. The format for the entries in this section is

```
<top_user_ID> <multiplier>
```

The entries are order-dependent and work in the following manner: all charges for users whose user ID numbers are greater than the user ID number specified in the previous entry and less than or equal to the user ID number specified in the current entry are multiplied by <multiplier>. For example, if section 7 looks like this:

```
[7] User Id Multipliers
  10 2.0
  50 1.0
  95 0.5
```

"user_billing" interprets it as

User ID		
From	To	Multiplier
0	10	2.0
11	50	1.0 (not multiplied)
51	95	0.5
> 95		1.0 (not multiplied)

Thus, "user_billing" multiplies all charges for users with IDs from 0 to 10 (inclusive) by 2.0, and all charges for users with IDs from 51 to 95 (inclusive) by 0.5. The charges for other users are not modified. Section 7 is the only section in the file of charges which may be empty. It must be the last section in the file. A maximum of ten entries is allowed.

If the user specifies the ´d´ option to "user_billing", it reads and displays the file of charges, so that the user can verify that the charges are correct (see Options Available). After displaying all the charges, the program stops.

Format of the User-information File

The "user_billing" command may also use the user-information file, "/act/userinfo". By default the billing program reads this file for specific information about individual users. If the file does not exist or if the user specifies the ´n´ option, the billing program uses only

(continued)

r=<ID_1-ID_2> Bill only users in the specified range.

s=<splr_sum> The name of the file to use as the spooler-summary file. Default is "/act/splrsum".

t=<number> Number of ticks per second, which is determined by the hardware. Default is 10. At the time of this writing, all existing 6809 UniFLEX systems use the default (10).

u=<usr_sum> The name of the user-summary file. Default is "/act/usrsum".

v Print verbose bills.

C Print condensed billing report.

R Print resource use only, not charges.

S Print only a summary of the total charges by all users in each category.

U=<user_name> Report only on the specified user.

Files Used by "User_billing"

The "user_billing" command uses many files to produce the billing report. The two main input files are a spooler-summary file and a user-summary file. These two files are produced by the "splr_sum" and "usr_sum" commands. The "user_billing" command produces a bill for each user in the user-summary file. If a spooler-summary file exists, "user_billing" reads the corresponding spooler-summary records for that user and calculates the amount of paper used and the charge for it. If a spooler-summary file does not exist (only systems with the Enhanced Printer Spooler support spooler accounting), no figures concerning paper are printed. A spooler-summary file is not essential to the proper functioning of the "user_billing" command. However, if the user-summary file does not exist or is empty, "user_billing" cannot produce meaningful results, even if a spooler-summary file exists.

The File of Charges, "/act/charges"

The third file required by "user_billing" is the file of charges, "/act/charges". The file of charges supplied on the master disk contains the bare minimum of data needed for the correct operation of the "user_billing" program. All the charges in the file are 0. The user only needs to change each 0 to the appropriate charge for the specified resource. A copy of a sample file (not the one on the disk) and an explanation of the file follow:

(continued)

By default "user_billing" produces bills that are twelve lines long. The user can, however, change the length of the bills by using the ´p´ option. If a bill is shorter than the specified length, "user_billing" pads it with blank lines. If a bill is longer than the specified length, "user_billing" prints the entire bill, then pads it with blank lines until the number of lines it contains is a multiple of the specified bill length. Thus, if the bills are printed on special forms, each bill starts at the top of a new form.

The user multiplier appears on the bill unless specifically disabled by the ´m´ option.

If the user specifies the ´v´ option, the report produced contains a list of the connect time and charges for each terminal used. It also contains information on paper usage, listed by spooler and form number.

The user can request a condensed billing report by using the ´C´ option. The condensed report consists of the name of each user (or the user information) and the total charges to each user. A summary containing the total charges to all users is printed at the end of the report.

The ´S´ option tells "user_billing" to print only a summary of the billing report. No bills are produced. The contents of this summary depend on whether the user selects a condensed, verbose, or default report.

EXAMPLES

```
1. /etc/user_billing
2. /etc/user_billing +d
3. /etc/user_billing +nvr=10-20 +u=usumfile.3.14 +s=ssumfile.3.14
4. /etc/user_billing +c=Yen +mvp=30
5. /etc/user_billing +RS
6. /etc/user_billing +U=joe +u=usum.1.4 +s=ssum.1.4 +o=u_joe.1.4
```

The first example uses the files "/act/splrsum" and "/act/usrsum" to produce a billing report in the file "/act/billing_rpt". The user-information file, "/act/userinfo", is read and used in the bills. Charges and resource usage are reported in the default billing format. The dollar sign, ´$´, is used as the currency symbol (by default).

The second example simply reads the file "/act/charges" and writes it to standard output. This procedure allows the user to verify that the file of charges is correct.

The third example produces a verbose billing report on users 10 through 20 inclusive. It uses the data in the summary files specified by the ´u´ and ´s´ options. The user information file is not used.

(continued)

system performance. I/O activity is costly to the system because it takes CPU time away from other users in order to perform the specified I/O operations.

Section 3 contains the charges for CPU time. In this section the "user_billing" command expects to find exactly one charge, which is the charge per second of CPU time. The CPU time used for billing is the sum of system CPU time and user CPU time.

Section 4 specifies the charge for K-Minutes, another measure of the impact of a task on overall system performance (maximum size of the task in Kbyes multiplied by CPU time). Tasks which are both small and use little CPU time are charged the least because they have much less impact on overall system performance than larger, more time-consuming tasks. Large tasks can burden the system and degrade performance by forcing swapping to take place. Users whose charges for K-Minutes are high are probably executing tasks that are large, time-consuming, or both.

Section 5 contains charges for the use of paper. Each charge has the following form:

```
<spooler_name> <form_number> <charge>
```

where <form_number> is the form number associated with the specified spooler and <charge> is the charge per page of that form.

An asterisk in the field for the spooler name refers to all spoolers not explicitly mentioned. An asterisk in the field for the form number refers to all forms not explicitly mentioned. The final entry in the spooler charges section must be

```
* * <charge>
```

Charges of the form

```
* <form_number> <charge>
```

and

```
<spooler_name> * <charge>
```

can appear anywhere in this section, not only after all occurrences of a particular form number or spooler. No limit is imposed on the number of spoolers or form numbers allowed in the file.

Section 6 is for charges for storage space (per block, per month). An accounting system for storage space has not yet been implemented on UniFLEX, so this section can be omitted. However, if the section header is in the file of charges, a charge must be present, even though it has no effect on the billing process.

(continued)

Bill length too short.
The number specifying the bill length is less than the minimum allowed.

Error creating ´<file_name>´ : <reason>.
The operating system returned an error when "user_billing" tried to create the specified file.

Error reading ´<file_name>´ : <reason>.
The operating system returned an error when "user_billing" tried to read the specified file.

Error opening ´<file_name>´ : <reason>.
The operating system returned an error when "user_billing" tried to open the specified file.

Error writing ´<file_name>´ : <reason>.
The operating system returned an error when "user_billing" tried to write to the specified file.

Error seeking to <number> in ´<file_name>´ : <reason>.
The operating system returned an error when "user_billing" tried to seek to the specified location (in hexadecimal).

Spooler summary file ´<file_name>´ not found : Assuming no paper charges.
This is merely an informative message. Systems without the Enhanced Printer Spooler cannot produce the spooler-summary file.

´/act/userinfo´ not found - will use user names.
The ´n´ option was not specified, but the file "/act/userinfo" could not be found.

User name ´<user_name>´ too long.
A user name in the user-information file was more than eight characters long. The name is truncated and processing continues.

Out of space!
The "user_billing" command uses some dynamic storage when producing bills, mostly to keep track of the information from the user-information file. Realistically, the program should be able to process hundreds of users. If space limitation is a problem, the user should execute "user_billing" with the ´n´ option, which disables the processing of the user information file.

SEE ALSO

setupact
splr_sum
sysact
usr_sum

user names in the bills. The format of the file "/act/userinfo" is as follows:

```
<user_name>
<up_to_512_characters_of_information_about_this_user>
@
```

The user name must appear on a line by itself. Following the user name are up to 512 characters which, by default, appear on the bill before the charges. The information typed here is echoed on the bill exactly (including carriage returns). To terminate a field of user-information, an "at" sign, ´@´, must appear in column 1 on a line by itself. If "user_billing" does not find a user in the user-information file, it prints only the user name on that user´s bill. The user names need not appear in any order. An example of an information file follows:

```
system
System Manager
Technical Systems Consultants
111 Providence Road
Chapel Hill, NC 27514
@
bin
Keeper of Binaries
Office 3a
Building 5
@
```

Format of Bills

The "user_billing" command produces one bill per user. Each bill lists resources used and charges for these resources. The bills are all sent to one file. By default that file is "/act/billing_rpt", but the user may specify any file by using the ´o´ option.

Bills can take on a number of different forms depending on the options selected. The basic form of a bill is

```
<user_name_or_user_information>

[(User multiplier = <user_multiplier>)]
  Connect Time      <hours>:<min>:<sec>     <cur_sym> <charge>
  K-I/O Units            <number>                    <charge>
  CPU Time          <hours>:<min>:<sec>              <charge>
  K-Minutes              <number>                    <charge>
  Paper Usage            <number>   Pages            <charge>
                                                  -----------
  User Total                                <cur_sym> <charge>
```

(continued)

usertime

Report the time spent by a selected user executing a command or list of commands.

SYNTAX

```
/etc/usertime <user_name> <command_name_list> [+a=<file_name>]
```

DESCRIPTION

The "usertime" command searches an accounting file and produces a report about the specified user. The report contains the amount of elapsed real time the user spent executing each of the specified commands. This command is useful for determining how much a particular user is using a selected set of programs.

Arguments

<user_name>	The name of the user to report on.
<command_name_list>	The list of commands to report on. The maximum number of commands allowed is thirty-two.
a=<file_name>	The name of the file to use as the accounting file. Default is "/act/acct".

EXAMPLES

```
1. /etc/usertime joe edit
2. /etc/usertime mary basic edit chess +a=/act/acct.2-84
```

The first example reports the amount of time user "joe" spent executing the "edit" program. The report is based on data in the default accounting file, "/act/acct".

The second example reports the amount of time user "mary" spent in the programs "basic", "edit", and "chess". The report is based on the data in the file "/act/acct.2-84".

ERROR MESSAGES

Too many command names listed (limit is 32).
The user specified more than thirty-two command names.

(continued)

The fourth example produces a verbose billing report and sets the currency symbol to "Yen". The bills produced in the report are each thirty (or a multiple of thirty) lines long. The user multipliers are not shown on the bills.

The fifth example uses the default input files to create a report of resource use only. No bills are produced, and no charges are shown.

The final example prints a billing report which shows information only for user "joe". It uses the summary files specified by the ´u´ and ´s´ options, and produces an output file called "u_joe.1.4".

ERROR MESSAGES

"/act/charges" not found!
A file of charges must exist for the proper operation of the "user_billing".

Error interpreting charges file : <reason>
An error was detected in the format of the file of charges. The offending line is printed along with an explanation of why the file is in error.

Option syntax error: ´<str>´
An option string must begin with a plus sign.

Too few lines per bill!
The argument to the ´p´ option is either less than 2 or could not be interpreted by "user_billing".

Illegal number of ´ticks´ specified!
The number of ticks specified was less than 1.

User ´<user_name>´ not found!
The user specified with the ´U´ option could not be found in the password file.

Unknown option ´<char>´.
An unknown option was detected on the command line.

´C´ and ´R´ cannot both be specified!
A condensed listing (´C´ option) shows charges only, so the ´R´ option (resource only) cannot be used with it.

Bad user range specified!
The user range specified with the ´r´ option was illegal. The range must be of the form +r=<lower>-<upper>.

(continued)

usr_sum

Produce a summary of each user´s usage of system resources.

SYNTAX

```
/etc/usr_sum [a+=<file_name_1>] [+h=<file_name_2>] [+p=<number_1>]
             [+s=<file_name_3>] [+t=<number_2>] [+cdlnsu]
```

DESCRIPTION

By default "usr_sum" reads information about each user´s usage of system resources from the current accounting file, "/act/acct", from the history file, "/act/history", and from the file containing a summary of previous user usage, "/act/usrsum". If a summary of the previous use of resources does not exist, "usr_sum" reads only the current accounting file and the history file. This information is summarized and written to the summary file, "/act/usrsum", replacing the existing contents of that file or, if necessary, creating the file. One option instructs "usr_sum" to send a textual report of the summarized information to standard output. Other options permit the user to specify input and output files other than the default files. There are also options which instruct "usr_sum" to obtain its input only from the current accounting file or only from the file containing a summary of user usage. Thus, usage statistics may be computed from the current accounting period, from previous accounting periods, or both.

The statistics gathered for each user include the total login time, the number of login sessions, the total CPU time, and the memory usage of all programs executed by the user.

Arguments

a=<file_name_1>	The name of the file containing the accounting information. Default is "/act/acct".
h=<file_name_2>	The name of the file containing the history of system usage. Default is "/act/history".
p=<number_1>	The number of lines per page in the report. Default is 66. Minimum is 11 because a 5-line margin appears at the top and bottom of each page.
s=<file_name_3>	The name of the file to contain the summary of user usage. Default is "/act/usrsum".

(continued)

NOTES

- The format for the arguments to this command is similar to the format for the options. However, each individual argument must be preceded by a plus sign; an argument cannot be used as part of an option string.
- This program can be used in conjunction with the programs "sys_stats", "cmd_sum", and "user_billing". Since both this program and "usr_sum" may actually delete or truncate the accounting file, care must be taken in selecting the order in which these programs are executed, as well as the options that are used with them.
- A list of commands that might commonly be used to produce both summaries of the various aspects of system usage and bills for this usage appears in the introduction to this manual.

ERROR MESSAGES

Unknown option: "<char>"
The character <char> appeared within an option string but is not recognized as a valid option.

Illegal accounting file specification
The format of the specification of the accounting file is illegal. The format must be "a=<file_name>".

Illegal history file specification
The format of the specification of the history file is illegal. The format must be "h=<file_name>".

Illegal summary file specification
The format of the specification of the summary file is illegal. The format must be "s=<file_name>".

Unrecognizable argument: "<str>"
The program did not recognize the command-line argument.

Incompatible options: (+c and +s) or (+n and +d)
The options ´c´ (use current accounting file only) and ´s´ (use previous summary file only) may not both be specified. The options ´n´ (do not update the summary file) and ´d´ (delete the accounting file after use) may not both be specified. This incompatibility is designed to prevent the inadvertent deletion of the information in the accounting file without updating the summary file. If this effect is desired, the user must manually delete the accounting file.

(continued)

User ´<user_name>´ does not exist.
The "usertime" command could not find the specified user in the password file.

Cannot open accounting file ´<file_name>´.
The specified accounting file could not be found or opened.

Unknown option letter: "<char>".
The option specified by <char> is not a valid option to "usertime".

SEE ALSO

commanduse
setupact
sysact

t=<number_2> Number of ticks per second, which is determined by the hardware. Default is 10. At the time of this writing, all existing 6809 UniFLEX systems use the default (10).

Options Available

c Use information from the current accounting file only. Implies the ´n´ option; incompatible with the ´s´ option.

d Delete the accounting and history files after use. Incompatible with the ´n´ option.

l Produce a listing of the summary data.

n Do not update the summary file.

s Use information from the summary file only. Do not use the accounting file.

u Sort commands in order of usage with the most frequently used command first.

EXAMPLES

```
1. /etc/usr_sum
2. /etc/usr_sum +a=/usr0/act/acct +h=/usr0/act/history +lc
3. /etc/usr_sum +lc
4. /etc/usr_sum +d
```

The first example processes the system accounting file "/act/acct" and the command-summary file "/act/cmdsum". It then uses the totals, broken down by command, to update the summary file "/act/cmdsum". This example produces no printed output.

The second example prints a summary of command usage using the accounting file "/usr0/act/acct" and the history file "/usr0/act/history" (i.e., files on a disk which has been mounted as "/usr0"). No previous summary information is read, nor is the summary file updated (the ´c´ option implies the ´n´ option).

The third example is much the same as the second, but it uses the default accounting file "/act/acct". This command is useful for producing a report about the current accounting period only.

The final example processes the current accounting file "/act/acct" and the command-summary file "/act/usrsum". It first updates the summary file with the new data, then truncates the system accounting file. Thus, a new accounting period is started.

(continued)

Page length too short.
The number specifying the page length is less than the minimum allowed.

Must be in single-user mode
The program does not update the summary file or delete the accounting file unless the system is in single-user mode.

Unable to access accounting file: "<file_name>"
The program could not access the specified accounting file. An auxiliary message indicates the reason.

Unable to access summary file "<file_name>"
The program could not access the specified file. An auxiliary message indicates the reason.

SEE ALSO

history
setupact
sysact
user_billing