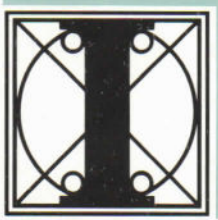
LSX Computer Line

Operating Systems

X/OS UNIX® System V-based Operating System
SAF System Administration Facilities

User Guide

X/OS



olivetti

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis
10015 Ivrea (Italy)

Information from
Olivetti Documentation

LSX Computer Line

Operating Systems

X/OS UNIX® System V-based Operating System
SAF System Administration Facilities

User Guide

olivetti

PREFACE

This manual explains the use of the System Administration Facilities (SAF) application and describes the operations which may be performed with it.

The main part of the manual is aimed at the administrator of the LSX computer system, who may or may not have previous experience of a UNIX environment.

The appendix entitled *"Customisation and Configuration of SAF"* is aimed at the administrator who has a basic knowledge of UNIX and the C programming language, and who wishes to extend the SAF application to suit his/her exact requirements.

SUMMARY

The manual consists of an introductory chapter followed by the Getting Started chapter. Next a chapter describing the SAF user interface, and a chapter describing the use of SAF. Then there are seven chapters detailing the available commands. Finally, there are three appendices: two detailing expected error messages and one explaining the customisation and configuration of SAF.

REFERENCES

Read first . . .

X/OS User Guide - Code 4043610 C

For further information, read . . .

X/OS System Administration Utilities User Manual - Code 4041480 F

DISTRIBUTION: As part of software kit (W)

FIRST EDITION: December 1987 - SAF Release 1.0

CONTENTS

9. SYSTEM MAINTENANCE COMMANDS

10. L P SPOOLER MAINTENANCE COMMANDS

1. INTRODUCTION

INTRODUCTION

The administration of the LSX operating system involves performing both frequent and infrequent tasks. Frequent commands can be laborious to make, and infrequent commands difficult to remember. The System Administration Facilities (SAF) remove these problems for some tasks, and thus reduce the scope for error.

There are some procedures which are not covered by SAF. Those requiring interactive operation cannot be done automatically, and those requiring in-depth knowledge of the task (such as patching a filesystem with *fsdb*) should be performed by an experienced Unix system administrator. However, there are a great many tasks which are made simpler by SAF.

These tasks are divided into **Maintenance Areas**, each of which can lead to a number of sub-areas and operations relating to the maintenance area. Each operation may have a number of options and/or parameters which will affect the execution of the operation or suboperations. If SAF needs a piece of information it will ask for it, clearly, using prompts.

The next chapter explains how to install the SAF package on to your system. The subsequent chapter deals with the manipulation of output and the input methods used by SAF. Then there is a chapter devoted to each Maintenance Area defining in detail all the operations available.

2. GETTING STARTED

This chapter explains the procedures necessary for the installation and configuration of the System Administration Facilities Package. These procedures would normally be carried out by the system administrator, but if a previous version of the System Administration Facilities package is on the system then this can be used to install SAF.

PRODUCT OVERVIEW

The System Administration Facilities Package is an easy to use, friendly and simple user interface to the various system administration activities and operations available on the LSX range of computers. The object of SAF is to allow a fairly inexperienced administrator to do some normal and not very specialised tasks with little danger of causing drastic upheavals in the day to day operations of the system.

DESCRIPTION OF THE KIT

The SAF kit, Issue 1 Version 0, consists of either one floppy disc, one streaming tape cartridge, or a 9-track magnetic tape.

SOFTWARE CONTENTS OF THE KIT

The files supplied are listed below.

- *SAF.o*
  This is the object file to which all extensions to SAF have to be linked.

- *Subop1.1..Subop5.5*
  These files are the C-shell scripts that perform the various operations of the sub menus.

- *english.**
  The nine files beginning with the word english are the nationalisation files for the SAF messages in the English language.

- *mengen*
  This is the menu generator procedure.

- *paint.h*, etc.
  There are five include source files with the extension *.h*

- *saf*
  This is the main executable file for the saf package

- *termcodes*
  This file contains the character terminal codes for the terminal attached to your system.

- *usr/options/saf.name*
  This file contains the saf package issue and version number.

INTERFACES

The installed system provides two programmes. The *System Administration Facilities* and an application generation package, called *mengen* described in appendix C.

INSTALLATION

KIT CHARACTERISTICS

There are no hardware or software prerequisites for running SAF. The package can be installed on to any LSX machine.

Layout of Kit After Installation

The installed package will require not more than 220Kb of hard disc space. After SAF has been installed on the hard disc the following files should have been copied from the installation media.

```
/lib/saf/SAF.o
/lib/saf/Subop1.4
/lib/saf/Subop1.5
/lib/saf/Subop1.7
/lib/saf/Subop2.2
/lib/saf/Subop2.3
/lib/saf/Subop2.4
/lib/saf/Subop2.5
/lib/saf/Subop2.6
/lib/saf/Subop3.10
/lib/saf/Subop3.11
/lib/saf/Subop3.12
/lib/saf/Subop4.3
/lib/saf/Subop5.5
/lib/saf/english.edit
/lib/saf/english.input
/lib/saf/english.md
/lib/saf/english.mengen
/lib/saf/english.more
/lib/saf/english.msg
/lib/saf/english.pmd
/lib/saf/english.saf
/lib/saf/english.vi
/lib/saf/ex_oper.c
```

```
/lib/saf/mengen
/lib/saf/paint.h
/lib/saf/pkg.h
/lib/saf/pkgnoop.h
/lib/saf/saf
/lib/saf/termcodes
/lib/saf/termcodes.h
/usr/options/saf.name
```

INSTALLATION PROCEDURE

If The *saf* package is not installed using SAF itself then it must be installed using the *admpkg* command after becoming **root** user. Refer to the *LSX Interfaces and Libraries Reference Manual.* The *admpkg* command should be entered in one of the following ways depending on the media on which your package is released.

admpkg	to install from discettes.
admpkg -dSTC	to install from streaming tape.
admpkg -dMTU	to install from magnetic tape.

The installation procedure will do the following:

1. The message **Installing the System Administration Facilities: Issue 1 Version 0** will be displayed.

2. A check is made to see if a previous version of the package exists. If it is already installed then a check will be made to see that the new package has more up to date version and issue numbers. If the installed package has higher issue or version numbers then the user will be warned and prompted as whether to continue with the installation, with the following messages:

```
**WARNING** This will install an older
Issue/Version (<fi>/<fv>) over the current
Issue/Version (<i>/<v>)
Type 'y' to install over current Issue or
'n' to stop installation:
```

The values *i* and *v* represent the Issue and Version numbers of the currently installed version. The values *fi* and *fv* represent the issue and version numbers of the replacing package. If this message should appear then reply **y** to continue the installation of the older version, or **n** to exit the procedure.

3. The space available on the hard disk is calculated. (Both the **root** and **/usr** file systems are checked.) This is compared with the space required by the corresponding directories for this installation. If there is not enough space the installation procedure exits with an error message and an error status of 1.

4. If the checks and conditions of the previous sections are successful then the procedure proceeds to install the files with the message:

   ```
   The following files are being installed:
   ```

   followed by the names of the files as they are copied.

5. After the files have been copied to the hard disc, a new login is created by inserting the following line in to the **/etc/passwd** file.

   ```
   saf::0:1:System Administration Facilities
                                   :/lib/saf:/lib/saf/saf
   ```

   Note: This login has no password.

6. The last operation is to print the message:

```
Installation of the System Administration
        Facilities:Issue 1 Version 0 is complete.
```

7. The command terminates with error 0.

Error Messages During Installation

During installation the following error situations can occur:

WARNING This will install an older Issue/Version *(<fi>/<fv>)* over the current Issue/Version *(<i>/<v>)*
Type 'y' to install over current Issue or 'n' to stop installation:

This means that SAF is currently installed and the issue or version of the currently installed one (fI<i>/<v>fR, found in the file **/usr/options/saf.name**), is more current than the one being installed.

Actions:
Respond **y**, the installation will continue anyway.
Respond **n**, the installation will stop and no changes will be made

---

ERROR System Administration Facilities cannot be installed -- Not enough space on the hard disc. There are *<USRspace>* blocks available on the /usr file system -- *<USRneeds>* blocks needed.

This means that there is not enough space on the file system which contains **/usr**. *<USRspace>* indicates how much space is available and *<USRneeds>* indicates how much space is needed.

Actions:
Either free space on the file system which contains **/usr** or rebuild the file system so that it is big enough to contain SAF.

ERROR System Administration Facilities cannot be installed --
Not enough space on the hard disc.
There are *<ROOTspace>* blocks available
on the / (root) file system --
<ROOTneeds> blocks needed.

This means that there is not enough space on the root file system. *<ROOTspace>* indicates how much space is available and *<ROOTneeds>* indicates how much space is needed.

Actions:
Either free space on the root file system or rebuild the root file system so that it is big enough to contain SAF.

ACTIVATION PROCEDURE

During the installation process a user called saf is created with super user privileges. This will automatically run SAF when you login with the command:

*login:***saf**

However to use the previous version of SAF, you must login as normal and then use the *su* command to gain superuser privileges. You will be able to run the previous version by typing:

saf -r

With the command **saf -v** just the current version number of saf will be displayed.

DEINSTALLATION PROCEDURE

The **saf** kit can only be removed using the *-wU* option of the *admpkg* command. **SAF** cannot be used to remove itself. To remove the **saf** kit from the system, first become **root** and then type in one of the following:

admpkg -wU	if installation was from discettes.
admpkg -dSTC -wU	was from streaming tape.
admpkg -dMTU -wU	was from magnetic tape.

This command will do the following:

1. print out the messages:

   Removing the System Administration Facilities:
   Issue 1 Version 0.

   The following files are being removed:

2. Next, each file named in the packages **Rlist** file will be removed.

3. Prints out the message:

   The System Administration Facilities:
   Issue 1 Version 0 has been removed.

3. THE SAF INTERFACE

SAF communicates with the system administrator via the terminal screen. The system administrator communicates with SAF via the terminal keyboard. The input is always visible on the screen, and may be reviewed or modified at any time before execution, according to the rules in the **System Administration Facilities Input** section below.

SYSTEM ADMINISTRATION FACILITIES OUTPUT

When an operation produces output on the screen, you may control the way in which the output appears and you may scroll forward and backward through the output as you desire. If the output produced is wider than 80 characters per line, it will be displayed using 132 characters per line. The output is divided into pages. Each page is displayed in a window on the screen, usually in the centre.

A page may be perused by using the arrow keys. These are the commands which may be used to control the display of the pages of the output.

right-arrow displays the next page

left-arrow displays the previous page

down-arrow scrolls one line down

up-arrow scrolls one line up

b displays the page at the beginning of the output

e displays the page at the end of the output

f enables the fast scroll mode (when a page is requested, it appears quickly on the screen, and remains there for perusal)

p	leaves the display and returns to the operation menu
s	enables the slow scroll mode (when a page is requested, it appears slowly on the screen, and remains there for perusal)

SYSTEM ADMINISTRATION FACILITIES INPUT

Input to SAF is of two different kinds; the choosing of a value for an option, or providing a specific piece of input data. There is a special type of option, called a **menu**. A menu provides a choice of areas to work in, or a choice of operations to execute. Choosing a value from a menu is done in exactly the same way as choosing a value for any other option.

Choosing an Option Value

SAF uses screen display attributes to help in the selection of option values. The underline and highlight attributes are used to identify the current option for which values are being chosen, and the reverse video attribute identifies which value is currently chosen for each option. In the example below, the option **Select maintenance area required** is currently active, and the value **User Maintenance** is currently chosen.

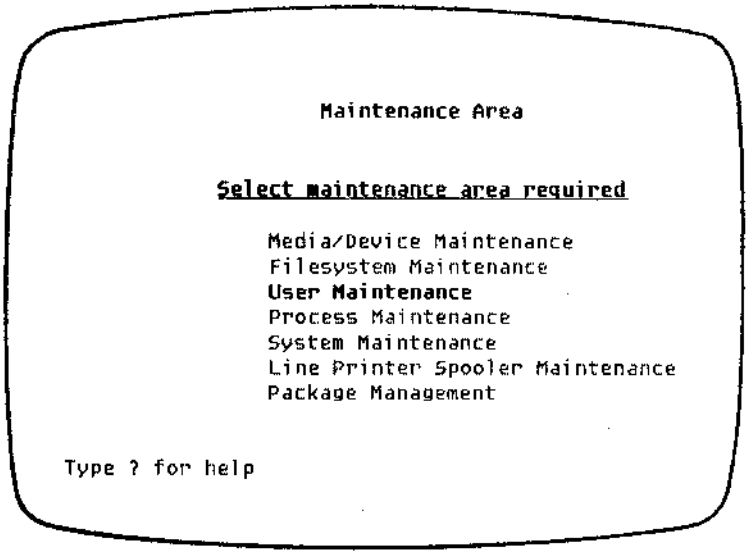


Fig. 3-1 How screen attributes help in the selection of option values

When choosing between the various values for an option, these are the commands available:

right-arrow	moves to the next option
left-arrow	moves to the previous option
down-arrow	gives the next option value
up-arrow	gives the previous option value
ESC	returns to the maintenance area menu
DEL	aborts the operation and returns to the operation menu

CR	goes to the next menu (or executes the operation)
n	goes to the next menu (or executes the operation)
p	returns to the previous menu
q	terminates the program
?	displays the help menu

By using the **up-arrow** and **down-arrow** keys, the reverse video attribute may be moved from one option value to another, thus changing the currently selected value. The arrows will both wrap around. If an option is left by moving on to another option or another menu, the chosen value is remembered until the next time that option appears. Thus, an operation may be defined, but need not be executed until another operation has been executed (perhaps to check the value of something). Another way to move around in the value selection is to type in the first few characters of the value required. As you type the the next matching value will be selected. If at any point a character does not match, then the terminal will beep and the non-matching character will be ignored. All the SAF menu options start with upper-case letters this avoids confusion with the commands **n** and **p**.

When there is more than one option for which values must be selected, the **right-arrow** and **left-arrow** keys will highlight and underline the next or previous option, respectively.

The **n** command will display the next menu in sequence of the options required, or, if all options have been specified, will execute the operation and display the results. Using the **p** command to display the previous menu, will eventually return to the initial maintenance area menu. This menu may also be returned to at any time during the selection process using the **ESC** key.

Non-deterministic Input Data

This is data which does not have known values and must be specified independently (e.g. a directory name). If more than one item of data is required in response to a single request (e.g. a list of group ids) the items may be separated by spaces or commas, unless a particular delimiter is specified in the input request. The input field is delimited by inverse video diamonds, as in the example below:

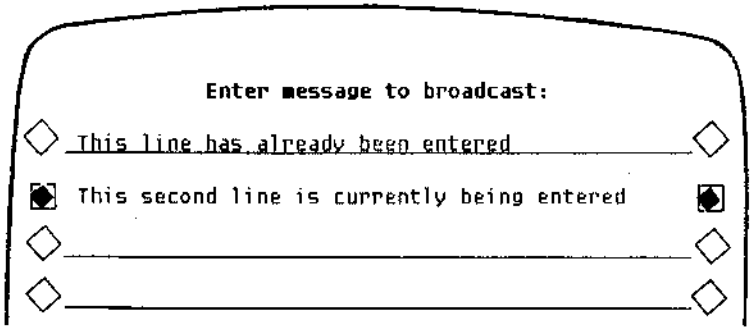


Fig. 3-2 Input fields delimited by inverse video diamonds.

When using this input mode, an editor is accepting the data. If the data has been typed correctly, a **CR** will terminate the input. However, if editing is required, the **ESC** key gives access to the edit commands. An indicator in the bottom right hand corner of the screen shows how the input character will be interpreted (as input or commands).

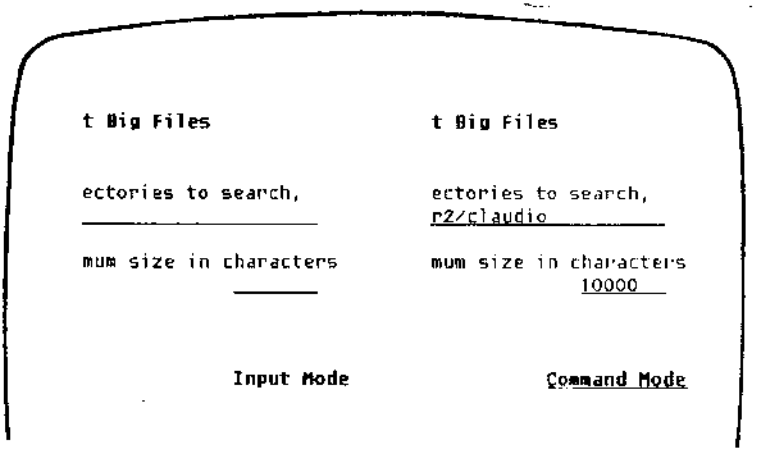


Fig. 3-3 Indicator showing interpretation of characters

Editing in SAF

The commands are most of the line editing commands available to the popular editor **vi**. A summary of the editing commands are as follows :

conventions used :

- [n] signifies an optional count - defaults to **1**
- str signifies a string of any non-control characters
- ch signifies any single non-control character
- **ESC** signifies the escape key
- word signifies a grouping of common characters; the types are **alphanumeric** or **punctuation**.

The elements of the commands are shown separated by spaces for clarity. The spaces do not form part of the command. As an example, the command to append the string *this is a string* at the current position would be typed:

athis is a stringESC

following the **a** str **ESC** format.

ESC	aborts an incomplete command
a str **ESC**	append *str* after the current position
A str **ESC**	append *str* to the end of the line
[n] **b**	position back *n* words
[n] **B**	position back *n* space delimited groups
c [n] **f** ch str **ESC**	change up to and including the *n*th *ch* with *str*
c [n] **t** ch str **ESC**	change up to but not including the *n*th *ch* with *str*
c [n] **w** str **ESC**	change the next *n* words with *str*
c [n] **W** str **ESC**	change the next *n* space delimited groups with *str*.
C str **ESC**	change from the current position to the end with *str*
d [n] **f** ch	delete up to and including the *n*th *ch*
d [n] **t** ch	delete up to but not including the *n*th *ch*
d [n] **w**	delete the next *n* words
d [n] **W**	delete the next *n* space delimited groups
D	delete from the current position to the end
[n] **e**	skip to the end of the *n*th word
[n] **E**	skip to the end of *n* space delimited groups
[n] **f** ch	skip forward to the *n*th *ch*
[n] **F** ch	skip backward to the *n*th *ch*
[n] **h**	skip backward *n* characters (a left-arrow may be substituted for the *h*)
i str **ESC**	insert *str* after the current position
I str **ESC**	insert *str* before the beginning of the line
[n] **l**	skip forward *n* characters (a space or a right-arrow may be substituted for the *l*)
p	put the characters deleted in

	the last update command, after the current position
P	put the characters deleted in the last update command before the current position
[n] **r** ch	change the next *n* characters with *ch*
R str **ESC**	overwrite from the current position with *str*
[n] **s** str **ESC**	substitute the next *n* characters with *str*
S str **ESC**	substitute the entire line with *str*
[n] **t** ch	skip forward to before the **n**th *ch*
[n] **T** ch	skip backward to after the **n**th *ch*
u	undo the last update
U	undo all commands that changed the line
[n] **w**	skip forward *n* words
[n] **W**	skip forward *n* space delimited groups
[n] **x**	delete the next *n* characters
[n] **X**	delete the previous *n* characters
y [n] **f** ch	take a copy of the string from the current position upto and including the *n*th *ch*
y [n] **t** ch	take a copy of the string from the current position upto, but not including, the *n*th *ch*
y [n] **w**	take a copy of the next *n* words
Y	take a copy of the entire line
~	switch the case of the current letter
[n] **|**	position to column *n*
;	redo the last f, F, t, or T command
,	redo the last f, F, t, or T command in the opposite direction
$	position to the end of the line
0	position to the start of the line
.	redo the last update command at the current position

The current position is marked by the character in inverse video. When you change a group of characters with the c command, the characters which are to be

replaced are highlighted until the first character is typed. Then the replacement characters are highlighted until the **ESC** key is typed. When the data is correct, a terminator will end the input and commit the characters typed to the operation. The terminators are:

CR go on to the next input field or screen (or execute if no more screens)

up-arrow go to previous input field

down-arrow go to next input field

CTRL n go to the next screen, skipping further fields on this screen (this will only be allowed if none of the fields skipped need to have input entered)

CTRL p go to the previous screen

In some cases, the input field is not blank, because a default value has been placed there, or it contains previous input to the operation. In this case, the input indicator shows **COMMAND MODE** immediately, and editing commands may be used to modify the data.

4. USING THE SYSTEM ADMIN. FACILITIES

USING THE SYSTEM ADMINISTRATION FACILITIES

This chapter will tell you how to start using the System Administration Facilities and what to expect at each stage. It is important to note that you **MUST** have **SuperUser Privileges** to use this system. These will automatically be provided if you use the login command as provided at installation:

*login:***saf**

However, if you wish to use the previous version of SAF, you must login as normal and then use the *.su* command to gain superuser privileges. You will be able to run the previous version by typing:

saf -r

You will then see a title screen appear, shortly followed by the initial menu screen.

Another option, *-v* exists. This causes the version number of SAF to be displayed. Type the command:

saf -v

the version number of SAF will be printed and then the programme will terminate.

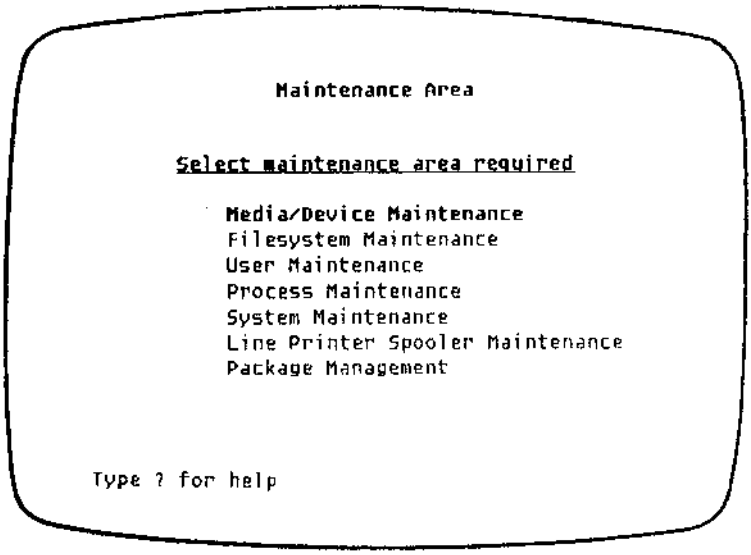


Fig. 4-1 SAF Initial Menu

The maintenance areas for which operations have been defined are those listed in the figure above.

You must identify the maintenance area in which you wish to work by using the **up-arrow** and **down-arrow** keys as explained in the **INTERFACE** Chapter. Using the **n** or **CR** key will cause one of the following menus to appear, according to the choice you made.

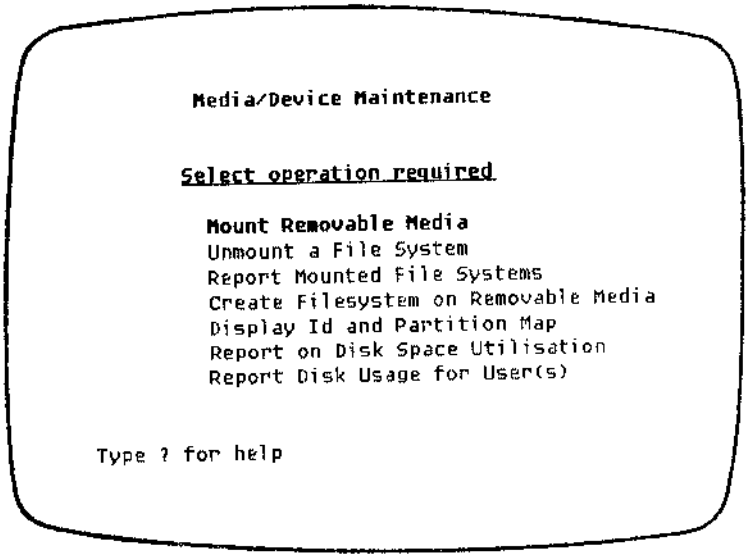


Fig. 4-2 Media/Device Maintenance Menu

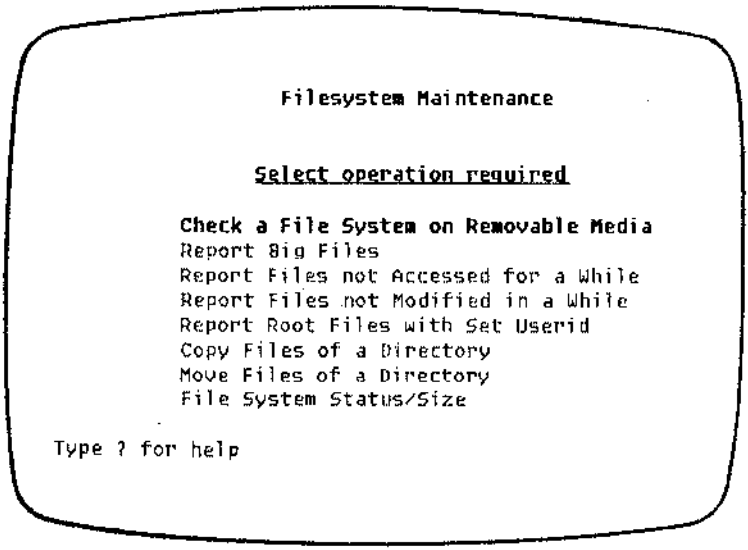


Fig. 4-3 Filesystem Maintenance Menu

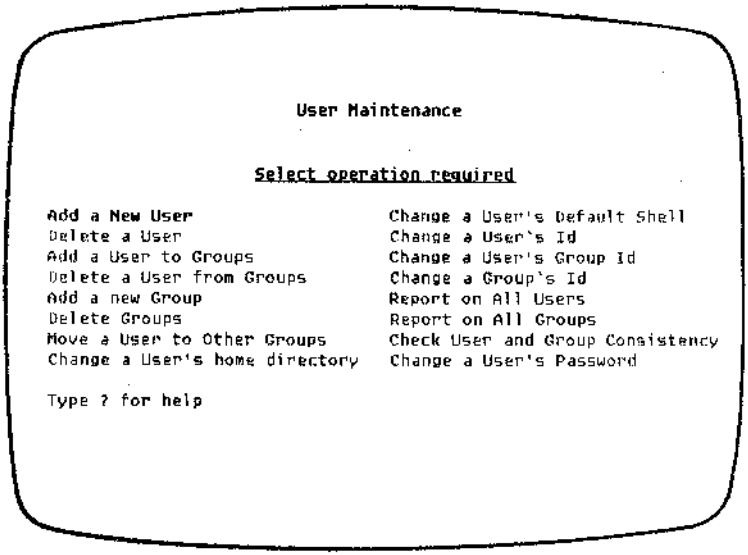


Fig. 4-4 User Maintenance Menu

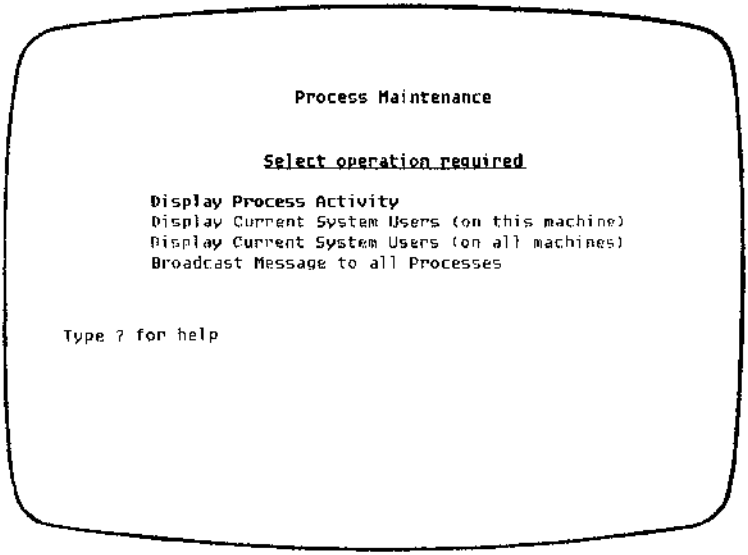


Fig. 4-5 Process Maintenance Menu

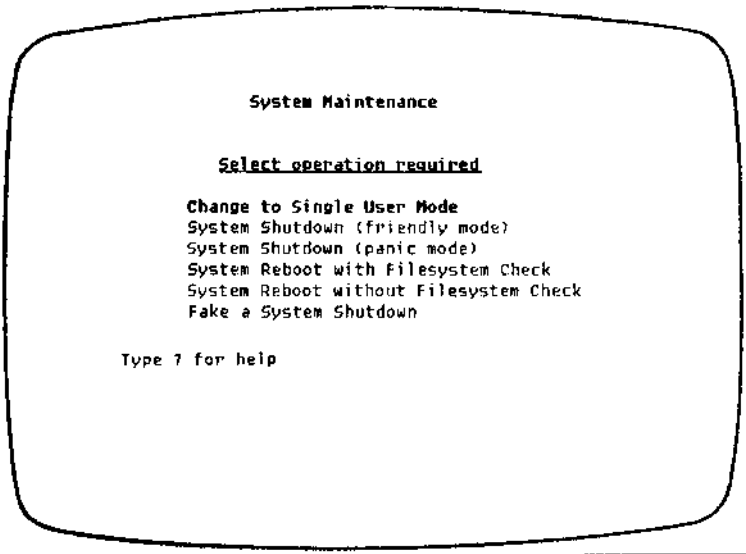


Fig. 4-6 System Maintenance Menu

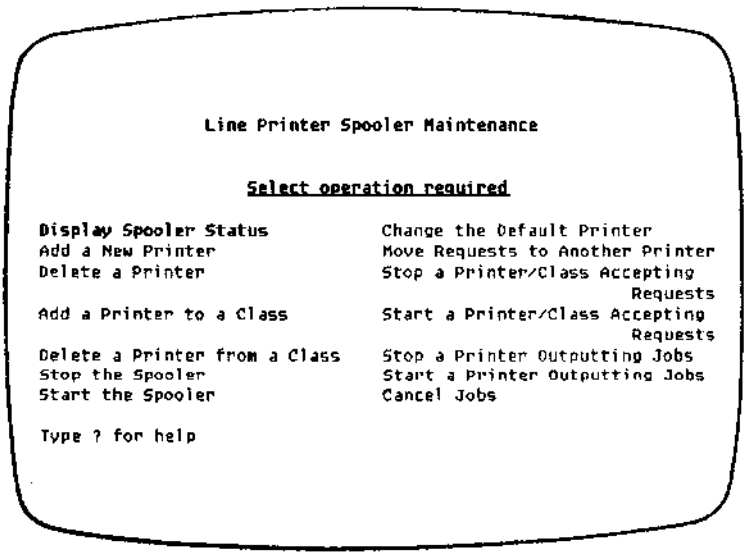


Fig. 4-7 Line Printer Spooler Maintenance Menu

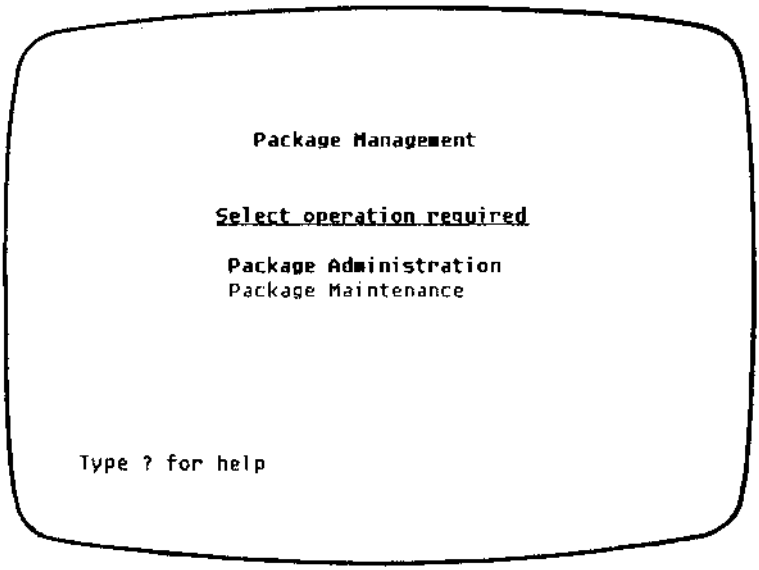


Fig. 4-8 Package Management Menu

These menus list the operations pertaining to the area, and you must select the one you require using the **up-arrow** and **down-arrow** keys as described above. For each operation, options or other specific data must be entered as required, following the methods described in the **INTERFACE** Chapter. For details of the operations defined for each area, see the relevant **COMMAND DESCRIPTIONS** Chapter.

5. MEDIA/DEVICE MAINTENANCE COMMANDS

MEDIA/DEVICE MAINTENANCE COMMANDS

This chapter describes, in detail, all the operations that have been defined for the media/device maintenance area. Operations in this maintenance area deal with removable media, which usually will mean floppy disks.

MOUNT REMOVABLE MEDIA

This operation allows a filesystem on a removable medium (floppy disk) to be mounted for use by the operating system. Once mounted, all filesystem operations may be performed upon the files.

Options

- floppy disk drive to use
  - . the only floppy drive
  - . the left hand floppy drive
  - . the right hand floppy drive
- the filesystem type
  - . 4.2 (a local filesystem)
  - . NFS (a remote filesystem)

  refer to the *LSX System Interfaces and Libraries Reference Manual - fstab*(4) for a full description of filesystem types. For information on NFS, refer to the *Network File System User Manual*.
- the mount options
  - . read-write

. read only

Input

- the name of the directory under which the filesystem is to be mounted. The directory must already exist in the operating system, and be specified starting with a /.
  Examples : **/mnt**
  /usr/flp

UNMOUNT A FILE SYSTEM

This operation will unmount a filesystem. The directory is not deleted when the operation is complete.

Options

- the name of the directory under which the filesystem is currently mounted

  . *mounted directory 1*

  . *mounted directory 2*

  . .. .. .

  . *mounted directory n*

  where *n* is the number of currently mounted filesystems.

REPORT MOUNTED FILE SYSTEMS

This operation displays the following information for each currently mounted filesystem:

- *Mount Device*; the name of the device on which the filesystem is held
- *Mounted On*; the directory in which the file system is mounted
- *Type*; the way in which the files are organised on the disk. Would usually have one of the following values:

  . 4.2

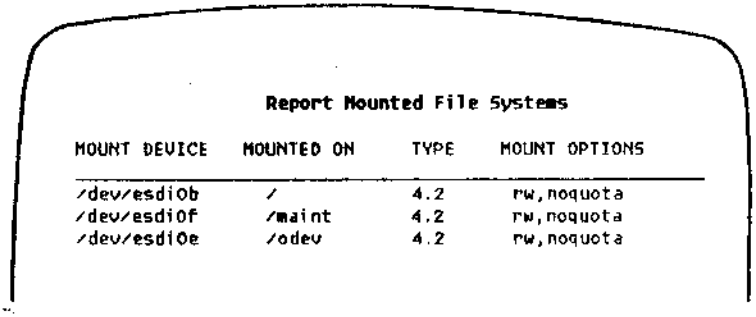
  . NFS (network file system)

  refer to the *LSX System Administration Utilities Guide* for a full description of the possible filesystem types.

- *Mount Options*; whether the filesystem is mounted as read only, or read-write.

Input

None required.



Report Mounted File Systems

MOUNT DEVICE	MOUNTED ON	TYPE	MOUNT OPTIONS
/dev/esdi0b	/	4.2	rw,noquota
/dev/esdi0f	/maint	4.2	rw,noquota
/dev/esdi0e	/odev	4.2	rw,noquota

Fig. 5-1 A typical display for 'Report Mounted File Systems'.

CREATE A FILESYSTEM ON REMOVABLE MEDIA

This operation creates an empty filesystem on a floppy disk, which can then be used as required.

Input

- a label (name) for the new filesystem. This must be an alphabetic string of a maximum of 14 characters
- an identity number for the floppy disk. This must be a numeric value of a maximum of 10 digits.

Options

- floppy disk drive to use
  - . the only floppy drive
  - . the left hand floppy drive

. the right hand floppy drive

DISPLAY ID AND PARTITION MAP

This operation produces a report showing :

- the disk label
- the disk identity number
- the number of the root partition
- the partition map (showing the sizes of different parts of the filesystem)

for a floppy disk which has been previously formatted using the **Create a Filesystem on Removable Media** command.

Options

- floppy disk drive to use
  - . the only floppy drive
  - . the left hand floppy drive
  - . the right hand floppy drive

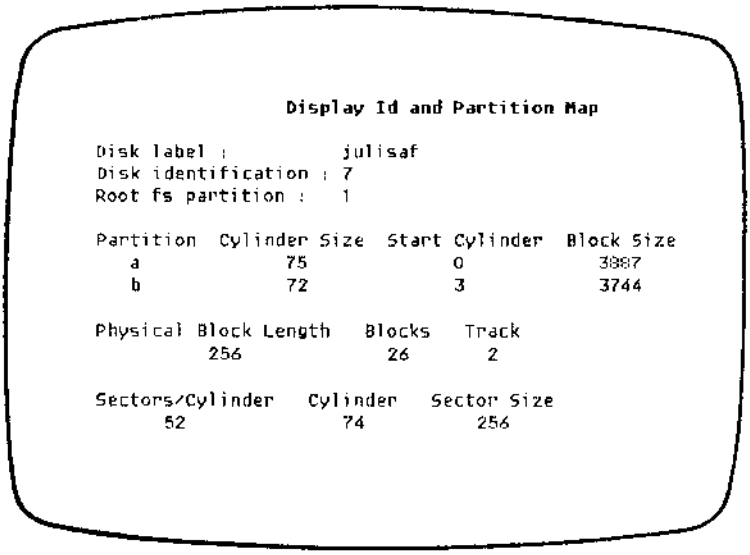


Fig. 5-2 A typical display for 'Display Id and Partition Map'.

REPORT ON DISK SPACE UTILISATION

This operation produces a report showing :

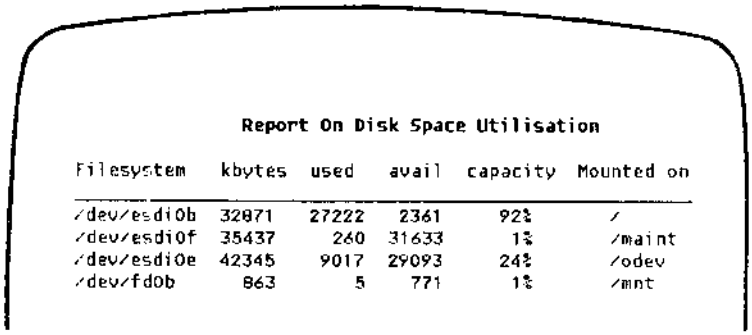
- *Filesystem*; the device on which the filesystem is held
- *Kbytes*; the total space in the filesystem (in Kbytes)
- *Avail*; the amount of disk space available (in Kbytes)
- *Used*; the amount of disk space used (in Kbytes)
- *Capacity*; the percentage usage

- *Mounted On*; the directory in which the filesystem is mounted

for all mounted filesystems (including removable filesystems).

Input

None required.



Report On Disk Space Utilisation

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/esdi0b	32871	27222	2361	92%	/
/dev/esdi0f	35437	260	31633	1%	/maint
/dev/esdi0e	42345	9017	29093	24%	/odev
/dev/fd0b	863	5	771	1%	/mnt

Fig. 5-3 A typical display for 'Report on Disk Space Utilisation'.

REPORT DISK USAGE FOR DIRECTORIES

This operation produces a report showing the amount of disk space (number of blocks) used by specified directories and their subdirectories.

Options

- type of report required

- a grand total for all directories under each of the specified directories
- individual totals for each directory under each of the specified directories

Input

- a list of directory names about which information is required.

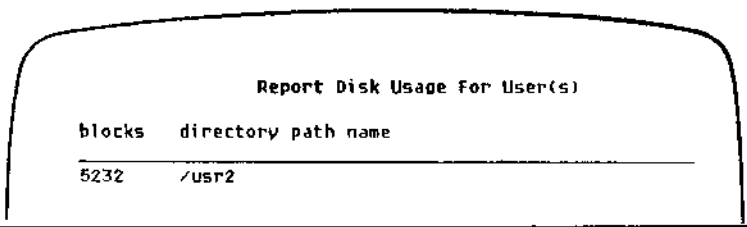


Fig. 5-4 Grand total for /usr2

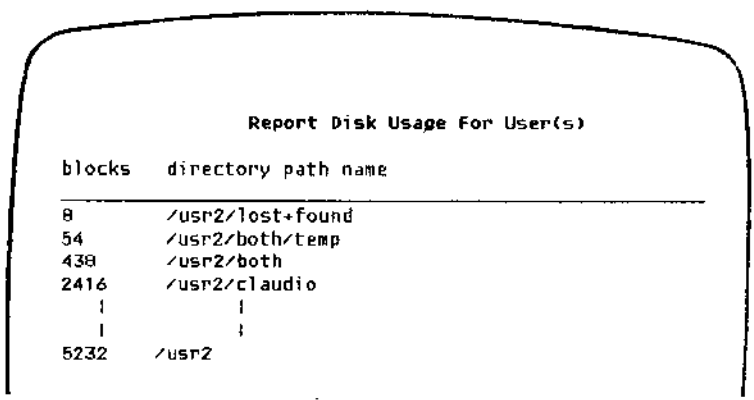


Fig. 5-5 Individual totals for /usr2

6. FILESYSTEM MAINTENANCE COMMANDS

This chapter describes, in detail, all the operations that have been defined for the filesystem maintenance area. The operations in this area deal with the performance and space utilisation of a whole filesystem.

CHECK A FILE SYSTEM ON REMOVABLE MEDIA

This operation performs a consistency and corruption check on a file system on a floppy disk. The status is reported, but problems are not repaired. If the check reports corruption, further investigation will be necessary, by consulting the instructions for *fsck*(1M). Inconsistencies checked are as follows:

- blocks claimed by more than one i-node (file information structure) or the free list
- blocks claimed by an i-node or the free list outside the range of the filesystem
- incorrect link counts
- size checks; incorrect number of blocks or directory size not 16-byte aligned. **Note** that this information is only a warning, since files containing holes will produce this message, even though the size is correct.
- bad i-node format
- blocks not accounted for anywhere
- directory checks; file pointing to unallocated i-node or i-node number out of range
- super block checks; more than 65536 i-nodes or more blocks for i-nodes than there are in the system

- bad free block list format
- total free block and/or free i-node count incorrect

Options

- floppy disk drive to use
  - . the only floppy drive
  - . the left hand floppy drive
  - . the right hand floppy drive

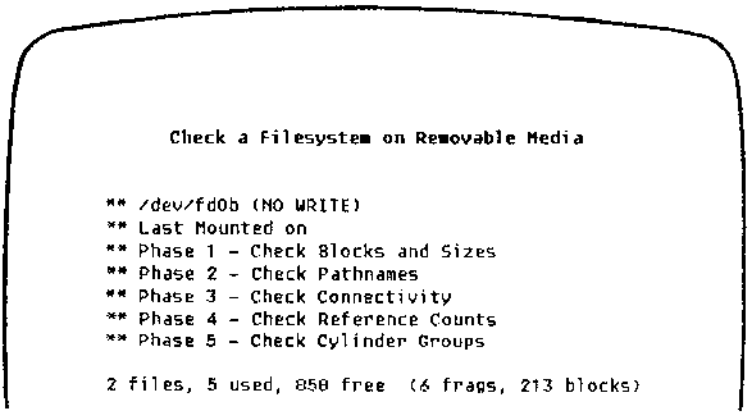


Fig. 6-1 A typical successful 'Check a Filesystem on Removable Media' report.

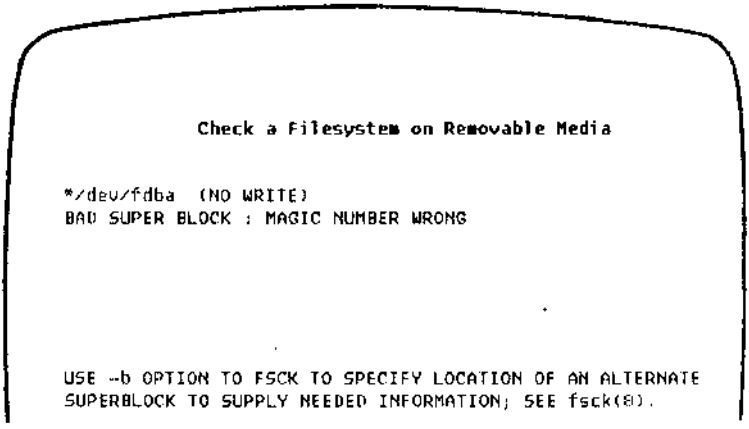


Fig. 6-2 A typical unsuccessful 'Check a Filesystem on Removable Media' report.

REPORT BIG FILES

This operation will report on any files larger than (or equal to) a specified size. General file information will also be displayed. It would typically be used to identify those files it would be most useful to remove when a filesystem is becoming full.

Input

- a list of directories to be searched
- the minimum size (in characters) of file to be reported

Options

- order of reported files
  - sorted by size (smallest first)
  - sorted by size (largest first)
  - sorted by pathname
  - unsorted

Note: An unsorted report will be produced much faster than a sorted report.

Notations Used in Output

- block size; the number of blocks occupied by the file
- permission; a string of 10 characters, where each position in the string has the following meanings:

Position	Meaning
0	File type. - = text file c = character file d = directory l = symbolic link file (for linking files between different filesystems) s = socket file (used for interprocess communication) p = pipe file (used for interprocess communication) b = block file

1	Owner read permission. r = permission - = no permission
2	Owner write permission. w = permission - = no permission
3	Owner execute permission. x = permission - = no permission
4	Group read permission. r = permission - = no permission
5	Group write permission. w = permission - = no permission
6	Group execute permission. x = permission - = no permission
7	General read permission. r = permission - = no permission
8	General write permission. w = permission - = no permission
9	General execute permission. x = permission - = no permission

- links; the number of links there are for the file. A link enables a file to be accessed by more than one name. If one user tries to delete the file, his link

will be deleted. Only if there are then no links left, will the file actually be deleted.

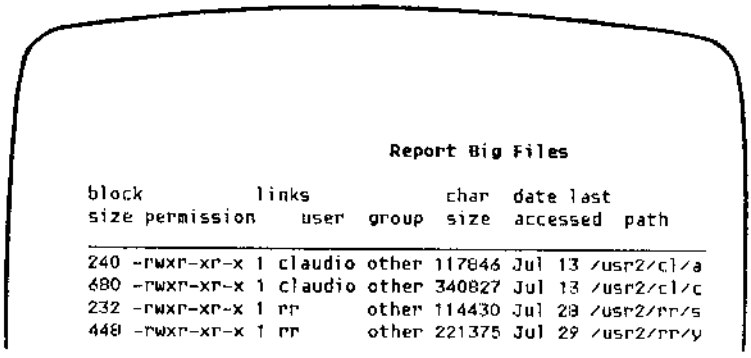
- user; the user ownership of the original file.

- group; the group ownership of the original file.

- char size; the number of characters occupied by the file.

- date last accessed; the date on which the file was last accessed.

- path; the pathname for the file.



Report Big Files

block size	permission	links	user	group	char size	date last accessed	path
240	-rwxr-xr-x	1	claudio	other	117846	Jul 13	/usr2/cl/a
680	-rwxr-xr-x	1	claudio	other	340827	Jul 13	/usr2/cl/c
232	-rwxr-xr-x	1	rr	other	114430	Jul 28	/usr2/rr/s
448	-rwxr-xr-x	1	rr	other	221375	Jul 29	/usr2/rr/y

Fig. 6-3 A typical report produced by 'Report Big Files'.

REPORT FILES NOT ACCESSED FOR A WHILE

This operation reports on files which have not been *accessed* within a specified number of days. It would typically be used to identify those files which could be removed with least inconvenience to the user when a filesystem is becoming full.

Input

- a list of directories to be searched
- number of days since last access

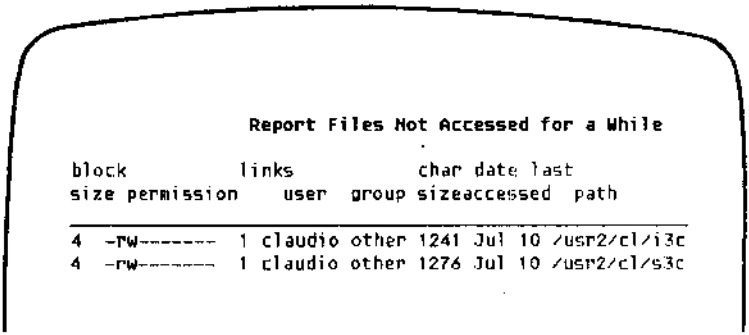
Options

- order of reported files
  - sorted by time since last access (most recent first)
  - sorted by time since last access (oldest first)
  - sorted by pathname
  - unsorted

Note: An unsorted report will be produced much faster than a sorted report.

Notations Uses in Output

Exactly as for *Report Big Files.*



Report Files Not Accessed for a While

block size	permission	links	user	group	char size	date last accessed	path
4	-rw-------	1	claudio	other	1241	Jul 10	/usr2/cl/i3c
4	-rw-------	1	claudio	other	1276	Jul 10	/usr2/cl/s3c

Fig. 6-4 A typical report produced by 'Report Files Not Accessed for a While'.

REPORT DIRECTORIES NOT MODIFIED IN A WHILE

This operation reports on directories which have not been *modified* within a specified number of days. It would typically be used to identify those directories which could be removed with least inconvenience to the user when a file system is becoming full.

Input

- a list of directories to be searched
- number of days since last modification

Options

- order of reported files

  . sorted by time since last modification (most recent first)

  . sorted by time since last modification (oldest first)

  . sorted by pathname

  . unsorted

Note: An unsorted report will be produced much faster than a sorted report.

Notations Used in Output

Exactly as for *Report Big Files*.

Report Directories Not Modified in a While

block size	permission	links	user	group	char size	date last accessed	path
2	drwxr-xr-x	4	both	other	256	Jul 7	/usr2/both/p
2	drwxr-xr-x	2	claudio	other	256	Jul 8	/usr2/cl/gen
2	drwxr-xr-x	2	claudio	other	256	Jul 8	/usr2/cl/prg

Fig. 6-5 A typical report produced by 'Report Files not Modified in a While'.

REPORT ROOT FILES WITH SET USERID

This operation reports all programs which have the userid bit set **and** have an owner of **root**. This is useful to know because such a program will allow the user who is running it to have access to the whole system. Thus, using this operation will highlight potential security problems.

Input

- a list of directories to be searched

Options

- order of reported files
    - sorted by time since last access (most recent first)
    - sorted by time since last access (oldest first)
    - sorted by pathname
    - unsorted

Note: An unsorted report will be produced much faster than a sorted report.

Notations Used in Output

Exactly as for *Report Big Files*.

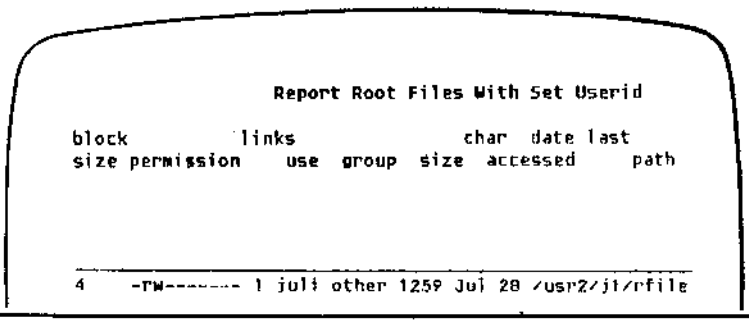


Fig. 6-6 A typical report produced by 'Report Root Files with Set Userid'.

COPY FILES OF A DIRECTORY

This operation *copies* the contents of a directory (including its subdirectories) into another directory in the filesystem. If the directory into which the files are being copied belongs to another user, the owner and group identities may be changed.

Input

- the name of the directory from which files are to be copied (must exist already)
- the name of the directory to which files are to be copied (if this does not exist, it will be created)
- new owner name for copied files (if not specified, remains unchanged)
- new group name for copied files (if not specified, remains unchanged)

MOVE FILES OF A DIRECTORY

This operation *moves* the contents of a directory (including its subdirectories) into another directory in the filesystem. If the directory into which the files are being moved belongs to another user, the owner and group identities may be changed.

Input

- the name of the directory from which files are to be moved (must exist already)
- The name of the directory to which files are to be moved (if this does not exist, it will be created)
- new owner name for moved files (if not specified, remains unchanged)
- new group name for moved files (if not specified, remains unchanged)

FILE SYSTEM STATUS/SIZE

This operation reports the current status of all mounted file systems. If the Network File System package is installed, this will operate over the whole network. The report comprises the following information for each filesystem:

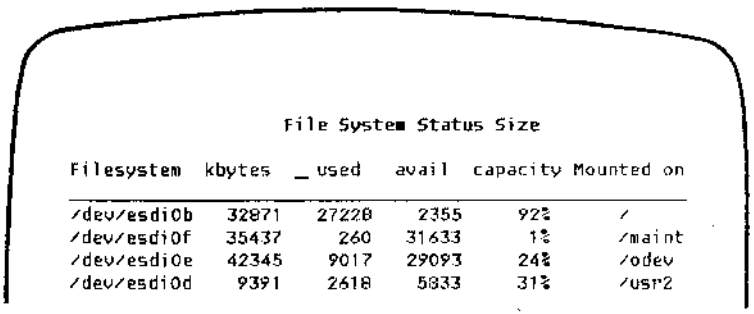
Filesystem the filesystem name

Kbytes the total number of Kbytes in the filesystem

Used	the number of Kbytes already used
Avail	the number of Kbytes available for use
Capacity	the percentage of the filesystem already used
Mounted On	the directory under which the filesystem is mounted

Input

None required.



File System Status Size

Filesystem	kbytes	_ used	avail	capacity	Mounted on
/dev/esdi0b	32871	27220	2355	92%	/
/dev/esdi0f	35437	260	31633	1%	/maint
/dev/esdi0e	42345	9017	29093	24%	/odev
/dev/esdi0d	9391	2618	5833	31%	/usr2

Fig. 6-7 A typical report produced by 'File System Status Size'.

7. USER MAINTENANCE COMMANDS

This chapter describes, in detail, all the operations that have been defined for the user maintenance area. This is one of the system administrator's busiest areas of activity. It deals with adding, removing, and changing details of users on the system.

ADD A NEW USER

This operation will create a new login id for a user.

Input

- a user name for use in response to the *login:* prompt
  (must be alphanumeric and unique)
- a user id number by which the system knows the user
  (must be numeric and unique)
- the name of the group to which the user will belong
  (must already exist)
- a home directory; the directory into which the user will be placed upon login
  (will be created if it does not exist)
- a shell (environment) to be used upon login
  (defaults to the C shell)
- a comment
  (typically the user's name and phone number)

DELETE A USER

This operation will remove a user id number and name from the system.

Options

- leave the user's files as they are
- delete the user's files
- change the ownership of the user's files

Input

- The name of the user to be deleted.

If ownership of the user's files is to be changed, then the following must be supplied :

- new user name (must already exist)
- new group name (optional - if specified, must already exist)

Note: the files will remain in their current position. The *move directory* operation should be used if required.

ADD A USER TO GROUPS

This operation adds a user to a group (or groups).

Input

- user name (must exist already)
- a list of groups to which the user is to be added, separated by spaces (must all exist already)

Note: if the user already belongs to one or more of the specified groups, no update will be made at all until the list has been edited.

DELETE A USER FROM GROUPS

This operation removes a user from a group (or groups).

Input

- user name
- a list of groups from which the user is to be removed, separated by spaces. A * in this field signifies 'all groups to which the user belongs'.

Note: if the user does not belong to one or more of the specified groups, no update will be made at all until the list has been edited.

ADD A NEW GROUP

This operation creates a new group and populates it with users.

Input

- new group name (must be alphanumeric and unique)
- new group id number (must be numeric and unique)
- a list of group members (it is permissible to create an empty group)

DELETE GROUPS

This operation removes a group (or groups) from the system.

Input

- a list of group names which are to be removed, separated by spaces.

Note: Any files which are owned by the deleted group(s) will now have no group access, and should therefore have their group identities changed using the *find*(1) command.

MOVE A USER TO OTHER GROUPS

This operation removes the user from all groups which he is currently a member of, and adds him to the new groups which are specified. Therefore, if a user wants to remain in a group of which he is already a member, the **Add a User to Groups** operation should be used in preference to this one.

Input

- user name (must exist already)
- a list of group names to which the user must be added (must all exist already)

CHANGE A USER'S HOME DIRECTORY

This operation will change the user's home directory to a different one. No files are moved to the new home directory (not even **.login**).

Input

- user name
- name of new home directory (must exist already)

CHANGE A USER'S DEFAULT SHELL

This operation will change the shell program which is run upon login.

Input

- user name
- the name of the new shell program (must have general execute permission)

CHANGE A USER'S ID

This operation will change a user's id, and the ownership of all files in the home directory and subdirectories, which may take a few minutes.

Note 1: files owned by the user in other directories, must have their ownership identity changed by *find*(1).

Note 2: if this operation is aborted, the resulting state of the file ownership will be undefined.

Input

- name of user whose id number is to be changed
- new user id number (must be numeric and unique)

CHANGE A USER'S GROUP ID

This operation will change a user's group id, and the ownership of all files in the home directory and subdirectories, which may take a few minutes.

Note 1: files owned by the user in other directories, must have their group ownership identity changed by *find*(1).

Note 2: if this operation is aborted, the resulting state of the file ownership will be undefined.

Input

- name of the user whose group id is to be changed
- new group name for the user (must exist already)

CHANGE A GROUP'S ID

This operation will change the id of a group, and the group ownership of all files belonging to all users in that group. This will take many minutes.

Note: if this operation is aborted, the resulting state of the file ownership for the group will be undefined.

Input

- name of the group whose id number is to be changed
- new group id number (must be numeric and unique)

REPORT ON ALL USERS

This operation will generate a report of all users known to the system, which gives general user information.

Options

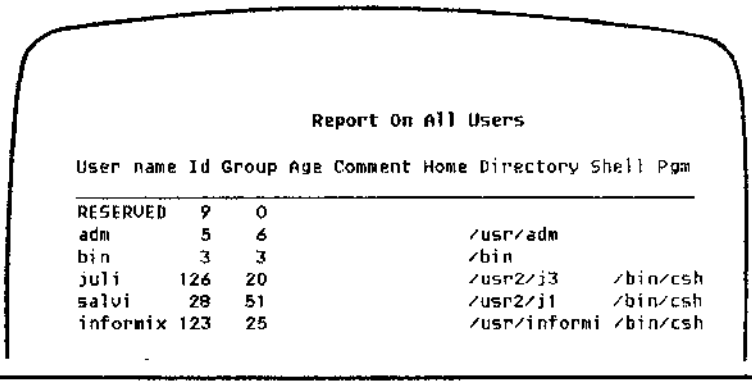
- order of sorting for report
  - . sorted by user name
  - . sorted by user id
  - . sorted by group id

Notations Used in Output

- User name; the name by which the user is known
- Id; the id number by which the system knows the user
- Group; the id number of the main group to which the user belongs
- Age; the number of days before the user's password expires
- Comment; the comment associated with the user at create time

- Home Directory; the directory into which the user is placed at login time

- Shell Pgm; the shell which is used by the user at login time



Report On All Users

User name	Id	Group	Age	Comment	Home Directory	Shell Pgm
RESERVED	9	0				
adm	5	6			/usr/adm	
bin	3	3			/bin	
juli	126	20			/usr2/j3	/bin/csh
salvi	28	51			/usr2/j1	/bin/csh
informix	123	25			/usr/informi	/bin/csh

Fig. 7-1 A typical 'Report on All Users'.

REPORT ON ALL GROUPS

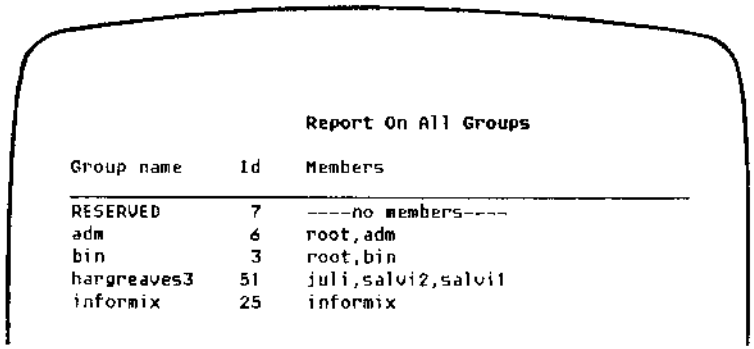
This operation will generate a report of all groups known to the system, which gives general group information.

Options

- order of sorting for report
  - . sorted by group name
  - . sorted by group id

Notations Used in Output

- Group name; the name by which the group is known
- Id; the id number by which the system knows the group
- Members; a list of the users who are members of the group



Report On All Groups

Group name	Id	Members
RESERVED	7	----no members----
adm	6	root,adm
bin	3	root,bin
hargreaves3	51	juli,salvi2,salvi1
informix	25	informix

Fig. 7-2 A typical 'Report on All Groups'.

CHECK USER AND GROUP CONSISTENCY

This operation will check the consistency between user definitions and group memberships. The checks include:

- all group members should be valid user names
- all user names should belong to one or more groups
- all fields should be present and valid

Input

None required.

REMOVE A USER'S PASSWORD

This operation will remove a user's password. If a user has forgotten his password, this operation must be used, as there is no way of finding the value of a password.

Input

- user name

8. PROCESS MAINTENANCE COMMANDS

PROCESS MAINTENANCE COMMANDS

This chapter describes, in detail, all the operations that have been defined for the process maintenance area. This maintenance area deals with the effect of processes on the system. The operations whose descriptions follow are for reporting and investigating. Remedial action is at the discretion of the system administrator, depending on the nature of the problem.

DISPLAY PROCESS ACTIVITY

This operation displays system information about selected processes. The information displayed will be headed using the following notation:

PID the process id; the number by which the process is known

TTY control terminal of the process

TIME cpu time used by the process (user + system time)

S the state of the process, described as follows:

- **O** non-existent processes
- **R** runnable processes
- **T** stopped processes
- **S** sleeping processes
- **W** waiting processes
- **Z** terminated processes
- **I** intermediate processes

X growing processes

COMD the command which is running in the process

NI the process scheduling increment; this value is used in computation of priorities by the scheduler - see *nice*(2)

SZ virtual size of the process in blocks

UID name of the process owner

PPID numerical id of the process' parent process

PRI process priority (this is a non-positive value when the process is in a non-interruptible wait state). Higher numbers mean lower priority.

ADDR the memory or disk address of the process for use by the scheduler when swapping the process in and out of core memory

WCHAN the event on which the process is waiting (an address in the system)

F flags associated with the process, including:

01 process is swapped in (in core)

02 system process

04 process locked in core

10 process being swapped

20 process being traced by another process

40 another tracing flag

These flags are octal and additive - a value of **3** means that both **01** and **02** are true.

C processor utilisation for scheduling

STIME starting time of the process

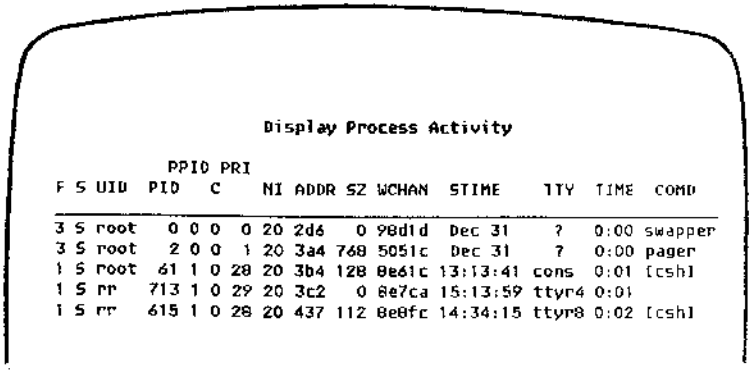
In addition to this information, a process may be indicated as:

<defunct> if it has exited and has a parent, but has not yet been waited upon. (See *wait*(2)).

Options

The options for selecting the process reporting are as follows:

- All Data Available for All Processes
- Data Only for Processes with Terminals



Display Process Activity

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	COMD
3	S	root	0	0	0	0	20	2d6	0	98d1d	Dec 31	?	0:00	swapper
3	S	root	2	0	0	1	20	3a4	768	5051c	Dec 31	?	0:00	pager
1	S	root	61	1	0	28	20	3b4	128	8e61c	13:13:41	cons	0:01	[csh]
1	S	rr	713	1	0	29	20	3c2	0	8e7ca	15:13:59	ttyr4	0:01	
1	S	rr	615	1	0	28	20	437	112	8e8fc	14:34:15	ttyr8	0:02	[csh]

Fig. 8-1 A typical process activity display.

See Also

ps(1).

DISPLAY CURRENT SYSTEM USERS (On This Machine)

This operation displays the status of logged-in users and other processes.

Options

The data display options are:

- All Data Available
- Time of last boot
- Dead and not respawned processes
- Lines waiting for login
- Init processes other than getty or users
- System run level; priority of the process
- Status of TTY (+ writable, - not writable, ? hung); the state of a terminal
- Time of last change of system date
- Useful process information

Notation Used in the Display

NAME the name of the user or process whose information is being displayed

LINE the terminal on which the process is running

TIME The time at which the process started

IDLE The time during which the process is completely inactive

PID The id number by which the process is known to the system

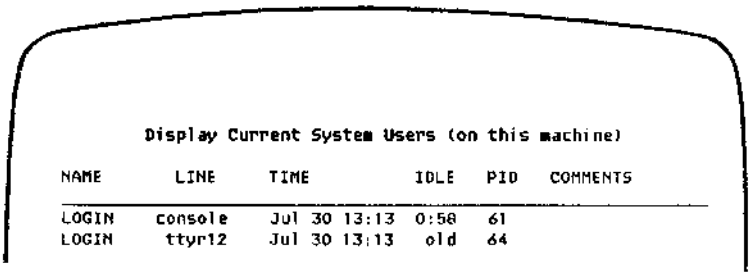


Fig. 8-2 A 'Lines Waiting for Login' display.

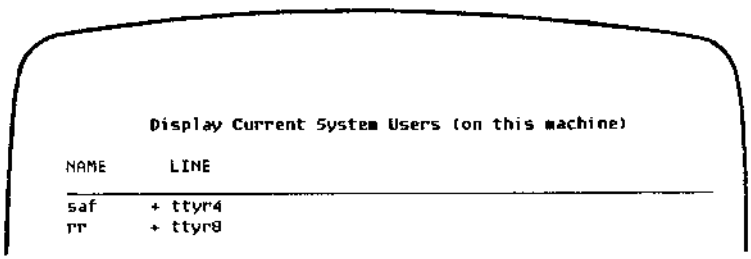


Fig. 8-3 A 'Status of TTY' display.

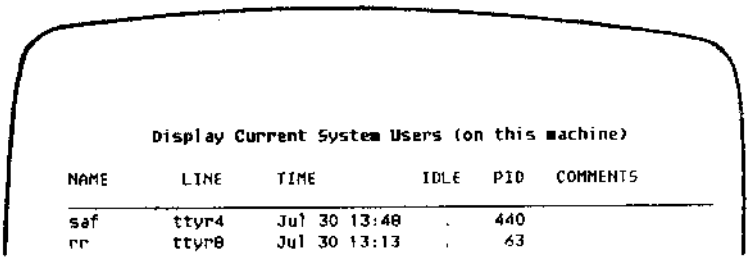


Fig. 8-4 A 'Useful Process Information' display.

BROADCAST MESSAGE TO ALL PROCESSES

This operation allows a message to be formatted, and broadcasted to every terminal on the system. It can be used for any messages that the System Administrator wishes to broadcast, such as informing users of new utility packages available on the system, etc..

Input

- the message to be broadcast. Four input fields are available, which may be filled with text as normal for non-deterministic input data (see the **INTERFACE** chapter).

9. SYSTEM MAINTENANCE COMMANDS

This chapter describes, in detail, all the operations that have been defined for the system maintenance area. These operations deal with the system as a whole, and thus can have far-reaching consequences for all users.

CHANGE THE SYSTEM DATE AND TIME

This operation allows the system date and time to be changed, if required.

Input

The input fields are all presented containing the current values, which may be edited as required:

the year	(entered as a 2-digit number)
the month	(entered as an alphabetic name of up to 14 characters **or** a 2-digit number)
the day	(entered as a 2-digit number)
the hour	(entered as a 2-digit number)
the minute	(entered as a 2-digit number)

CHANGE TO SINGLE USER MODE

This operation will issue a number of warnings over a number of minutes, inhibit further logins, and will then:

- log off all users

- shut down all system processes
- change the run state to *single user*.

Options

- certainty of wanting to log off all users
  - . do not want to log off all users
  - . do want to log off all users
- number of minutes over which warnings are to be given
  - . none
  - . 1 minute
  - . 3 minutes
  - . 5 minutes
  - . 10 minutes
  - . 30 minutes

Input

- A message to be displayed to users trying to login. Several input fields are provided for its definition.

SYSTEM SHUTDOWN (Friendly Mode)

This operation will issue a number of warning over a number of minutes, inhibit further logins, and will then:

- log off all users
- shut down all system processes
- shut down the machine by stopping the operating system.

Options

- certainty of wanting to log off all users
  - . do not want to log off all users
  - . do want to log off all users
- number of minutes over which warnings are to be given
  - . none
  - . 1 minute
  - . 3 minutes
  - . 5 minutes
  - . 10 minutes
  - . 30 minutes

Input

- A message to be displayed to users trying to login. Several input fields are provided for its definition.

SYSTEM SHUTDOWN (Panic Mode)

This operation will stop the machine without any warning or attempt to tidy up. It should be used only in extreme circumstances, such as a fire on a disk drive.

Options

- is sync required before shutdown
  - . no sync required
  - . sync required

SYSTEM REBOOT WITH FILESYSTEM CHECK

This operation will issue a number of warnings over a number of minutes, inhibit further logins, and will then:

- log off all users
- shut down all system processes
- reboot the system
- check all file systems.

Only after this is complete will anyone be able to login.

Options

- certainty of wanting to log off all users
  - . do not want to log off all users
  - . do want to log off all users
- number of minutes over which warnings are to be given.
  - . none
  - . 1 minute
  - . 3 minutes
  - . 5 minutes
  - . 10 minutes
  - . 30 minutes

Input

- A message to be displayed to users trying to login. Several input fields are provided for its definition.

SYSTEM REBOOT WITHOUT FILESYSTEM CHECK

This operation will perform a reboot without any warning to users, and should only be used in extreme circumstances. It will:

- log off all users

- shut down all system processes

- reboot the system

Users will be able to login as soon as the system comes up.

Options

- is sync required before stopping

  . sync not required

  . sync required

- mode required for restart

  . single user mode

  . multi user mode

FAKE A SYSTEM SHUTDOWN

This operation goes through the motions of shutting down the system without actually doing it. This would be used as a method of clearing the system if it is required for other purposes. It will be left running normally, and it will be possible for users to log in if they try.

Options

- number of minutes before users are made aware of the fact that the system has not really shut down

  . none

- 1 minute
- 3 minutes
- 5 minutes
- 10 minutes
- 30 minutes

Input

- A message to be displayed to users trying to login. Several input fields are provided for its definition.

10. L P SPOOLER MAINTENANCE COMMANDS

L P SPOOLER MAINTENANCE COMMANDS

This chapter describes, in detail, all the operations that have been defined for the line printer spooler maintenance area. This maintenance area deals with the line printer spooling system, which is described in the *LSX System Administration Utilities User Guide*. Each command is cross-referenced to its equivalent in the above mentioned guide, from where further information may be obtained.

DISPLAY SPOOLER STATUS

This operation displays the status of various components of the line printer spooling system.

Options

- Destinations Accepting Requests
- Class Names and Member Printers
- The Default Destination
- Outstanding Requests by Destination or Request
- Outstanding Requests by User
- Status of Printers
- Status of Scheduler
- Status Summary
- Full Status
- Devices Associated with Printers

Input

- if displaying status of Outstanding Requests by Destination or Request

  . a list of request id numbers **OR** a list of printer names **OR** a list of printer class names, which must all be separated by commas. If no list is entered, all outstanding requests will be displayed.

- if displaying status of Outstanding Requests by User

  . a list of user names, separated by commas. If no list is entered, all outstanding requests will be displayed.

See Also

lpstat(1).

ADD A NEW PRINTER

This operation will:

- create a new printer
- add it to a class (optional). If a class is specified which does not exist, it will be created.
- associate the printer with a device

Input

- the new printer name (must be alphanumeric and unique)

- the class name (if specified, must be alphanumeric)
- the device name (must exist in the system already)

See Also

lpadmin(1M).

DELETE A PRINTER

This operation removes a printer. If the printer is the only member of a class, the class is also removed.

Input

- the printer name

See Also

lpadmin(1M).

ADD A PRINTER TO A CLASS

This operation adds an existing printer to a class.

Input

- printer name
- class name. If a class name is specified which does not exist, it will be created (must be alphanumeric).

See Also

lpadmin(1M).

DELETE A PRINTER FROM A CLASS

This operation removes a printer from a class. If the printer is the only member of the class, the class is also removed.

Input

- printer name
- class name

See Also

lpadmin(1M).

STOP THE SPOOLER

This operation halts the spooler. It does not prevent users from queuing jobs to be printed. The spooler must be stopped before most of the other spooler operations may be performed.

Input

None required.

See Also

lpshut(1M).

START THE SPOOLER

This operation restarts the spooler.

Input

None required.

See Also

lpsched(1M).

CHANGE THE DEFAULT PRINTER

This operation (re)defines the default printer. This is the printer to which jobs will be directed if none is specified on the *lp*(1) command.

Input

- the name of the (new) default printer (must already exist)

See Also

lpadmin(1M).

MOVE REQUESTS TO ANOTHER PRINTER

This operation will move a job (or jobs) from one printer or class to another. If every job on a printer is moved, then that printer will be disabled.

Options

- which jobs to move
  - all jobs on a printer
  - selected jobs

Input

- if moving all jobs on a printer:
  - printer name from which to move jobs
  - target printer name (must exist already)
- if moving selected jobs
  - job request ids to be moved, separated by spaces
  - target printer name (must exist already)

See Also

lpmove(1M).

STOP A PRINTER/CLASS ACCEPTING REQUESTS

This operation prevents users from queuing jobs to a specific printer, or class of printers. If desired, a message may be specified, which will be displayed to the user when the *lp*(1) command is attempted. This is useful in events such as a printer breaking down.

Input

- name of printer or class which must not accept requests
- one-line reason which will be displayed to user (optional)

See Also

reject(1M).

START A PRINTER/CLASS ACCEPTING REQUESTS

This operation allows users to submit jobs to a printer, or class of printers, which has previously been rejecting requests.

Input

- name of printer which may now accept requests

See Also

accept(1M).

STOP A PRINTER OUTPUTTING JOBS

This operation stops the spooler from sending files to the printer. It does not stop jobs from being queued to the printer. This would typically be used when the paper needs changing, or for any temporary situation which stops the spooler from printing. If desired, a message may be specified, which will be displayed to the user when the *lp*(1) command is attempted.

Input

- name of printer to be temporarily halted
- one-line reason which will be displayed to user (optional)

Options

- whether or not to cancel any jobs currently printing on the printer

See Also

disable(1).

START A PRINTER OUTPUTTING JOBS

This operation allows jobs to be sent to a printer, after they have previously been temporarily stopped.

Input

- name of printer which may now print jobs

See Also

enable(1).

CANCEL JOBS

This operation removes a job (or jobs) from the printing queues.

Options

- which jobs to cancel
  - all jobs on a specified printer
  - specified jobs only

Input

- if cancelling all jobs on a printer
  - . name of printer from which to cancel jobs
- if cancelling specified jobs
  - . list of request ids, separated by spaces

See Also

cancel(1).

11. PACKAGE MANAGEMENT COMMANDS

PACKAGE MANAGEMENT COMMANDS

This chapter describes, in detail, all the operations that have been defined for the package management area. This area deals with the management of user installable packages, and divides into two sub areas. The two areas are PACKAGE ADMINISTRATION and PACKAGE MAINTENANCE. The specific details for the package maintenance of individual packages are described in the relevant package documentation. The operations involved under package administration are described below.

PACKAGE INSTALLATION

This operation installs an X/OS software kit from a specified device. Reference should be made to the Getting Started section of the relevant manual for the kit being installed.

Options

- The device from which to install.
    - The Right Floppy
    - The Left Floppy
    - The Only Floppy
    - 9-Track Tape
    - The Streaming Tape Cartridge

PACKAGE REMOVAL

This operation does the converse of **PACKAGE INSTALLATION.** The options available are identical to those available for that operation. The media with which the package was installed should be loaded into the relevant device, so that the package de-installation procedures and associated files may be accessed.

Options

- The device from which to install.
  - The Right Floppy
  - The Left Floppy
  - The Only Floppy
  - 9-Track Tape
  - The Streaming Tape Cartridge

LIST ALL INSTALLED PACKAGES

This operation prints out the details from all the **/usr/options/*.name** files. This has the effect of printing the names and Issue/Version numbers of all the X/OS packages installed on the system.

Options

None available.

INTEGRATE PACKAGE INTO SAF

This operation integrates the package maintenance procedures into SAF for use via the PACKAGE MAINTENANCE menu option. This is done by creating and executing a make file with entries from the directory **/lib/saf/pex**.

Options

None available.

A. SAF ERROR MESSAGES

This appendix contains lists of the expected error messages from SAF, and where necessary, an explanation of their meanings.

STANDARD ERROR MESSAGES

Messages arising from user errors, in alphabetical order.

A component of the path does not exist -
One of the subdirectories specified in naming a directory does not exist. It is probably spelt wrongly.

A component of the path is not a directory -
One of the subdirectories specified in naming a directory is not a directory. A filename has probably been specified.

'All Groups' not allowed for this command -
Commands which allow a list of group names to be replaced by a '*' indicate this in their descriptions. The command currently being used is not one of these.

Already member -
You cannot add a user to a group of which he is already a member.

Character not found -
Input editing error.

Count illegal -
Input editing error.

Directory must begin with a '/'

Illegal position -
Input editing error.

Must be a non white-space character -
Input editing error.

No characters in record -
Input editing error.

No more room to insert -
Input editing error.

No room for string -
Input editing error.

Non extant members -
One or more group members do not exist. Something is probably spelt wrongly.

Not that many characters before position -
Input editing error.

Not that many characters in record -
Input editing error.

Not member -
You cannot remove a user from a group of which he is not a member.

Only alphabetic input accepted

Only alphanumeric input accepted

Only numeric input accepted

Past end of line -
Input editing error.

Past end of record -
Input editing error.

The floppy is not accessible -
The floppy disk specified cannot be read. Possibly it is not properly inserted in the drive.

The floppy is write protected

The group name does not exist

The path resides on a read-only filesystem -
An operation has been requested which requires write access to a read-only filesystem. For example, copying files.

The pathname exists and is not a directory

The user name does not exist

These do not exist *name1 name2 .... name n* -
The specified list of groups, directories etc. contains at least one entity which does not exist.

This class does not exist

This device does not exist

This directory does not exist

This field must have data in it

This group already exists

This group does not exist

This group id already exists

This printer already exists

This printer does not exist

This printer/class does not exist

This program suffers from *error no* **- see** *intro*(2) -
The shell program specified cannot be used. The *error no* will be explained in the *LSX Programmers Reference Manual*.

This shell program does not exist

This shell program is not executable

This user already exists

This user does not exist

This user id already exists

Unrecognised command

INTERNAL ERROR MESSAGES

Messages arising from operating system faults. The user has not made an error. In all cases, abort the operation using the DEL key, and retry the operation. If the error still occurs, exit from SAF, and take appropriate remedial action.

An I/O error occurred while creating the directory

Can't create a temporary file

Can't open work file

Can not open /etc/mtab

Error *error no* occurred while creating the directory

Name exists but I can't stat it

B. MENGEN ERROR MESSAGES

MENGEN ERROR MESSAGES

This appendix contains a list of the error messages which may be produced during the use of *mengen* (see Appendix C). They appear in alphabetical order. In most cases the messages are self-explanatory, but any that are not are accompanied by a brief explanation.

Already folded -
A box may only be folded once.

Already unfolded

Another box or margin is in the way

Can't delete heading or only line

Can't delete only line

Can't fold an input screen -
Only option boxes may be folded, since input boxes must have only one input per line, and there may be only one box per input screen.

Current position is in a box -
You may not create a new box within an existing box.

Current position is not in a box -
Box operations may only be done from within a box.

Legal only for empty screens -
Only empty screens may be removed.

Legal only for unprotected option screens -
Boxes may only be created or pasted in unprotected option screens, unless the screen is empty, because input screens may only contain one box.

Legal only for unprotected option and input screens -
You may only cut or remove boxes from unprotected option or input screens. If trying to remove an operation screen, the entry for that screen in the menu on the screen above should be deleted using the e commands.

Maximum boxes already defined -
The maximum number of boxes allowed on a screen is 10.

Name already exists

Name already used

No more room -
Using the **reduce** command in a folded box, the box cannot be reduced any further.

No undefined items

Not enough room below box -
The box must be moved higher on the screen before another line may be added, or before it may be unfolded.

Not enough room to the right of box -
The box must be moved to the left on the screen before it may be folded, or before a line may be added into a folded box which would cause the current contents to be shifted in such a way that the box needs to become bigger.

Nothing was saved for that identifier

Only legal for an operation screen -
You cannot use the **x** command on any other screen type.

Only works on a folded box -
The **reduce** command will only operate on a folded box. It is unnecessary on any others.

Screen still has boxes -
Only screens with no boxes may be removed.

The CUT item has a different type -
Only option boxes may be pasted into option screens, and only input boxes may be pasted into input screens.

There are still undefined items -
This is a warning that some screens have not been filled with boxes at the time of exiting from *mengen*. It will be necessary to locate them using the **u** command, and define them.

This screen is delete protected

This screen is insert protected

Too close to another box or margin

Unrecognised command

You must insert some lines after the heading

C. CUSTOMISATION AND CONFIGURATION OF SAF

CUSTOMISATION AND CONFIGURATION OF SAF

The customisation and configuration of SAF are done using an application generation package, called *mengen*. This system produces menu-driven applications with interfaces to user-written code, and modifies existing applications, such as SAF.

In the terms of SAF, **customisation** and **configuration** are two distinct phases.

CUSTOMISATION allows all menus and prompts to appear in the language of the country in which the system is being used. It is also called 'nationalisation'. This is done by editing the screen data, which consists of boxes of text. Using the screen formatter **e** command (edit), every English phrase may be replaced by its foreign equivalent. Full details concerning the Screen Formatter appear later in this appendix.

CONFIGURATION allows for the addition and/or subtraction of operations to/from the system. This involves:

- altering existing menu screens
- adding new option and input screens / deleting existing option and input screens
- adding new code

Editing of the screens may be done very easily, using the screen formatter commands detailed below.

THE SCREEN FORMATTER

Both of the above described functions are carried out via a screen formatter, with an integral editor, which allows you to:

- create/delete/alter screens
- create/delete/alter/move boxes within the screens
- alter the contents of boxes within the screens

SCREEN TYPES

There are four types of screen with which the screen formatter may deal:

Area An area is a way of grouping together a number of related operations. An area screen contains one menu of all the specified areas. The areas may be subdivided further into sub-areas. Their menus would also appear on area screens. In SAF there is one area screen - the initial menu.

Operation There is an operation screen for each area (or, if subdivided, for each sub-area). An operation screen contains one menu of all the specified operations relating to that area.

Option An option screen contains one or more option boxes. These contain the values for the options, which are used as input (see the **INTERFACE** chapter).

Input An input screen contains **one** input box. It contains the prompts and input fields for non-deterministic data input (see the **INTERFACE** chapter).

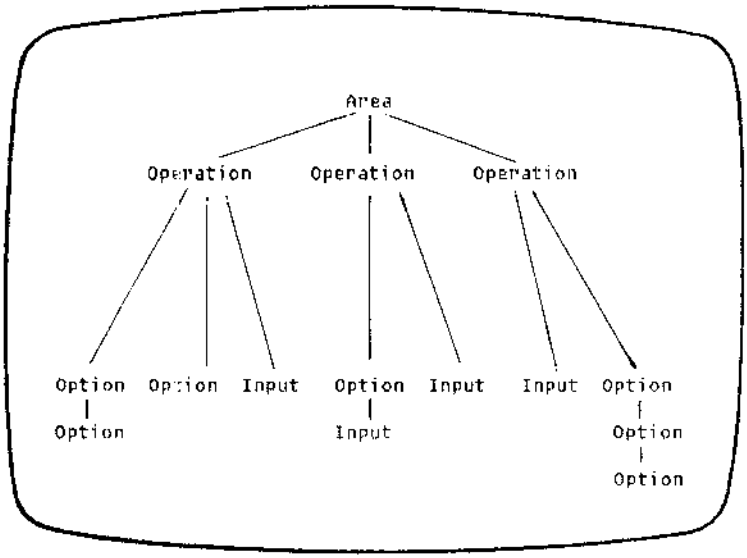


Fig. C-1 A typical screen hierarchy.

Note: if an operation does not require any input, it need not be followed by option or input screens.

Protected Screens

The Option and Input screens which are part of the initially supplied application, (SAF), are protected. This means you cannot insert or delete lines or boxes into them. Area and Operation screens, however, are not protected, so that additional areas and operations may be added into them, and unwanted ones may be removed. Great care must be taken when removing things from unprotected screens, since once they are removed, they cannot be recovered.

BOXES

The text on the various screens described above is represented in *mengen* by boxes. Each block of text is bounded by a box, which is treated as a complete entity which can be moved around the screen at will. For example, an operation screen contains one box, which in turn contains the menu of operations. The whole menu may be moved to another part of the screen with just one command, rather than moving each line at a time. A box may contain:

- a menu (menu and operation screens)
- an option and its values (option screens)
- a group of input prompts and fields (input screens)

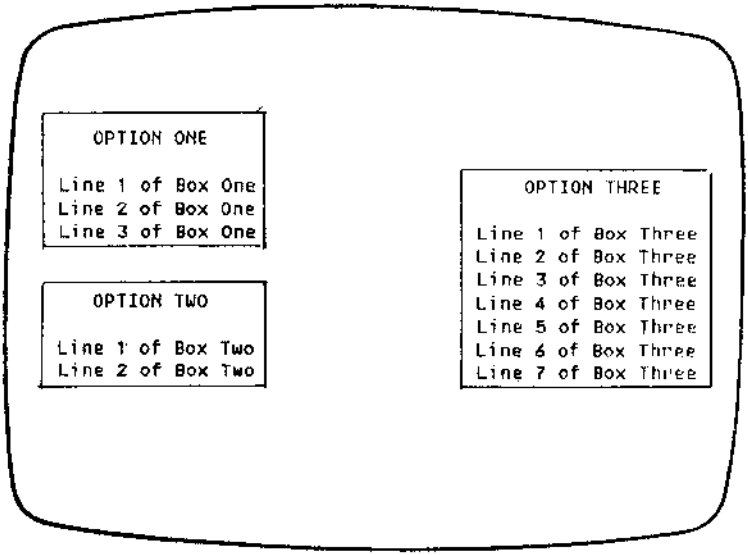


Fig. C-2 An Option screen of boxes.

HOW TO USE THE SCREEN FORMATTER

To begin using the screen formatter, you must type:

mengen

if you want to use the latest version of the menu data files, or:

mengen -r

if you want to use the previous version of the menu data files, stored as backup. You will be placed immediately into screen formatter mode.

When you have completed the required customisation and/or configuration, the formatter is exited using the **q** command. You will be offered the following choices:

Backed-Up Original — the new data will be saved in the same file as the old data: *language*.md , but keeping the old data in a backup file: *language*.md.old .

Original — the new data will be saved in the same file as the old data: *language*.md , overwriting it.

New File — the new data will be saved in a new file, which you will be asked to specify. This would typically be used if you were customising the screen data for another language and wanted to save it under a new language filename (see **CONFIGURING SAF FOR DIFFERENT LANGUAGES**).

Abort the Quit — return to screen formatter mode, perhaps to define undefined items.

The available commands are explained below.

THE HELP COMMAND

? details of available commands for the current mode will be displayed

FORMATTING COMMANDS

Upon entry to the screen formatter, you will be in **Screen Formatter** mode. In this mode you can move the cursor around the screen using the **arrow** keys, and you can gain access to the various types of commands detailed in following sections. In this mode you can also manipulate existing boxes or create new one, using the following commands:

n creates a minimum sized box and puts you directly into the editing mode, so that you may immediately fill in the contents of the box. There are four types of boxes:

area box requires a heading and at least one entry (a blank line will automatically be placed after the heading).

operation box as for area box

option box as for area box

input box requires at least one entry. Only **one** input field may be specified on each line, the input field itself being denoted by a number of **#** characters, equal to the number of characters in the input field. Lines may appear without any input fields.

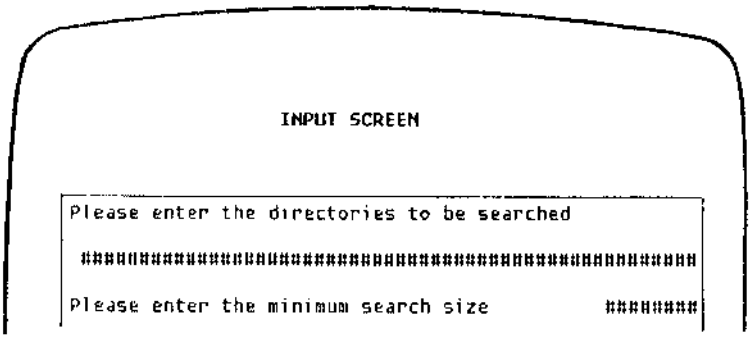


Fig. C-3 The definition of an input field

Upon quitting from the editor, the name of a 'screen pre-accept routine' will be requested. This is optional, and would be needed if the screen was only to be displayed in certain circumstances. For example, different inputs will be requested in some operations, according to the value selected in a previous option. This arises because all screens belonging to an operation are linked together serially, rather than in a tree structure, and to implement a tree structure, a mechanism exists for skipping screens. In addition, for each input field specified, the following information will be requested:

- the allowable input types. Use the **right-arrow** and **left-arrow** keys to choose between: alphabetic, alphanumeric, numeric, all types.

- whether or not the field is allowed to be blank. Use the **right-arrow** and **left-arrow** keys to choose between: yes, no.

- the name of a field pre-accept routine. This would typically be used to provide a default value for an input field

- the name of a field post-accept routine. This would typically be used to validate the input which the user enters. For example, checking that a specified printer exists, and supplying the necessary accept or reject message to the screen.

r removes a box from an option screen or an input screen (you cannot remove a box from an operation screen or an area screen). Once the box is deleted, any pointers from it to associated routines, are lost. (Any routines which are now never called will still be linked into the system). If you inadvertently remove a box, it is possible to abandon the whole session using the **DEL** command. This will, however, lose any other changes you have made. You are given the choice to **QUIT WITHOUT SAVE** or **CONTINUE**.

c 'cuts' a box from an option screen or an input screen (you cannot cut a box from an operation screen or an area screen). You will be asked to save the box as an identifier of a single upper or lower case letter.

p 'pastes' a previously cut option box into an unprotected option screen, or input box into an empty input screen. You will be asked to give the one letter identifier of the box which you wish to paste. **Note** that a cut box may only be pasted once. The box

identifier will then be deleted.

R removes an option screen or an input screen (you cannot remove an operation screen or an area screen). The screen to be removed must be empty. If you inadvertently remove a screen, it is possible to abandon the whole session using the **DEL** command. This will, however, lose any other changes you have made. You are given the choice to **QUIT WITHOUT SAVE** or **CONTINUE**.

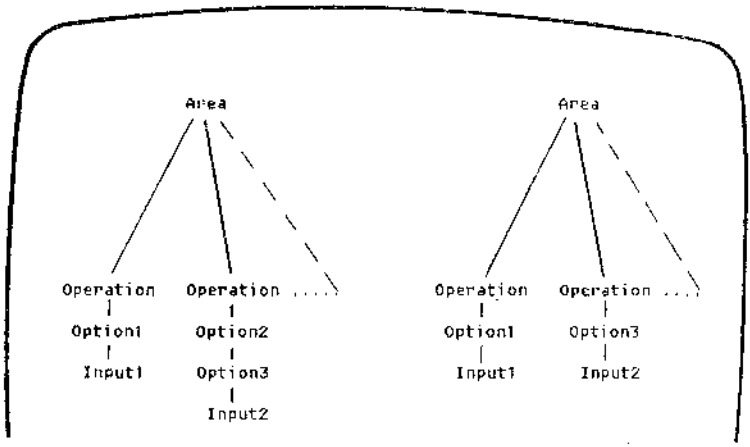


Fig. C-4 The effects of removing a screen.

x gives you the name of the execute routine and the number of parameters for each operation on an **operation screen.** If the box is folded, the first half will be given first, and using the **CR** key the second half may be displayed.

i displays the 'id' of the box in which the cursor is currently positioned. This is of use when writing screen pre-accept routines.

SCREEN POSITIONING COMMANDS

These commands allow you to move around the screen hierarchy in order to work on the screens which already exist, or to create new ones.

f will take you forward to the next screen. There are two types of action which may then follow:

choice	an area screen or an operation screen will give a choice of screens to go to next. Use the **up-arrow** and **down-arrow** keys to choose which one to go to, and then use the **CR** key to take you there.
no choice	within an operation, in an option screen or an input screen, there is only one

In either of these two cases, if no screen exists where you have chosen to go, the following options will be offered:

create *type* screen	you will be placed in a blank screen of the requested type.
do nothing	no screen will be created, and you will be returned to the screen formatter command mode.

Use the **right-arrow** and **left-arrow** keys to choose the required action, followed by **CR** .

b will take you back to the previous screen in the hierarchy. Note that there is never a choice, because any given screen may have only one parent.

u will position the cursor on an element which has an undefined 'next' screen. This is a screen which exists, but which contains no boxes. The **CR** key will take you to the screen, into which you may enter boxes, or which you may delete. Using this command

will help you to eliminate incomplete menu generation.

ESC will abort an **f** or a **u** command, and return you to the screen formatter command mode.

SCANNING COMMANDS

The scanning commands allow the cursor to be moved around the screen with more ease than using only the **arrow** keys, which are normally available under the screen formatter.

s this command gives access to the scanning commands listed below.

right-arrow moves the cursor into the next box to the right of the cursor's current position, if there is one. Otherwise it moves to the right hand edge of the screen.

left-arrow moves the cursor into the next box to the left of the cursor's current position, if there is one. Otherwise it moves to the left hand edge of the screen.

up-arrow moves the cursor into the next box above the cursor's current position, if there is one. Otherwise it moves to the top of the screen.

down-arrow moves the cursor into the next box below the cursor's current position, if there is one. Otherwise it moves to the bottom of the screen.

h centres the cursor horizontally on the screen.

v centres the cursor vertically on the screen.

c centres the cursor both horizontally and vertically on the screen.

ESC terminates the scanning commands and returns to the screen formatter command mode.

MOTION COMMANDS

These commands allow for the easy repositioning of menu boxes on the screen. The command invoked will only be successfully executed if there is no other box which would be partially or totally obscured by the movement.

- **m** this command gives access to the motion commands listed below. Note that the cursor must be positioned in the box which you wish to reposition, before using this command, otherwise the error message "CURRENT POSITION NOT IN A BOX" will appear.

  up-arrow move the box upwards one line at a time

  down-arrow move the box downwards one line at a time

  right-arrow move the box to the right one column at a time

  left-arrow move the box to the left one column at a time

  l positions the box at the left hand side of the screen

  r positions the box at the right hand side of the screen

u positions the box at the top of the screen

d positions the box at the bottom of the screen

h centres the box horizontally on the screen

v centres the box vertically on the screen

ESC terminates the motion commands and returns to the screen formatter command mode.

EDITING COMMANDS

To edit the contents of a box, there is a special subset of edit commands as follows:

e this command gives access to the editing commands. The commands which are available, are the line-editing commands listed under **Non-deterministic Input Data** in the **INTERFACE** chapter. Note that as characters are added, the box will expand to accommodate them. If characters have been deleted, leaving the box too wide, it will be truncated on the right upon exit from the editor. The line commands, which are also terminators for the Input Mode, are:

down-arrow moves the cursor down onto the line below its current position

up-arrow moves the cursor up onto the line above its current position

CTRL d deletes the current line. All structures in the hierarchy below will also be removed. The box will

automatically be redimensioned as required. If you inadvertently remove a hierarchy of structures, such as an area, you may abandon the whole session using the **DEL** command. This will, however lose any other changes you have made. You are given the choice to **QUIT WITHOUT SAVE** or **CONTINUE**.

CTRL i inserts a new line below the current position of the cursor. As with the above command, any necessary redimensioning of the box is automatically done. If a new entry is created in an operation screen, the name of the execution routine for the new operation will be requested. You will also be given the choice of creating an option screen, an input screen or no screen, for the operation.

CTRL f 'folds' a box. The single column within a box is split into two adjacent columns, for esthetic reasons (if a list of options is becoming too long). The box is redimensioned accordingly.

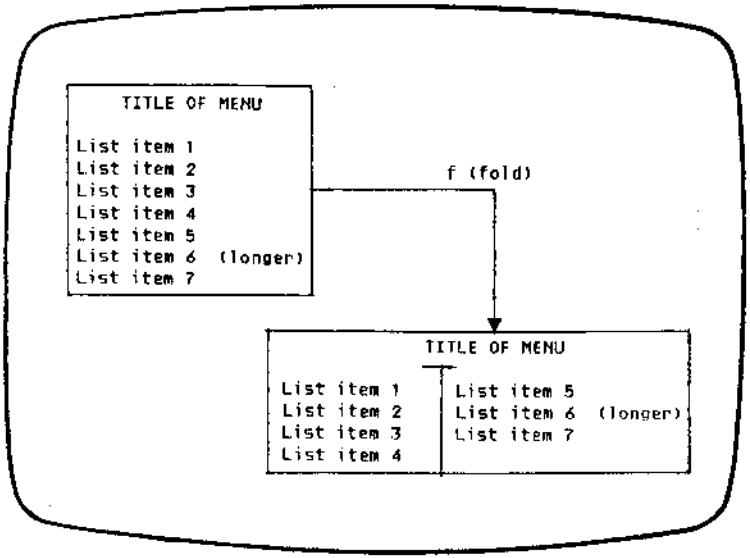


Fig. C-5 The effects of folding a box.

CTRL u 'unfolds' a box. Two adjacent columns within a box are joined to form a single column. The box is redimensioned accordingly.

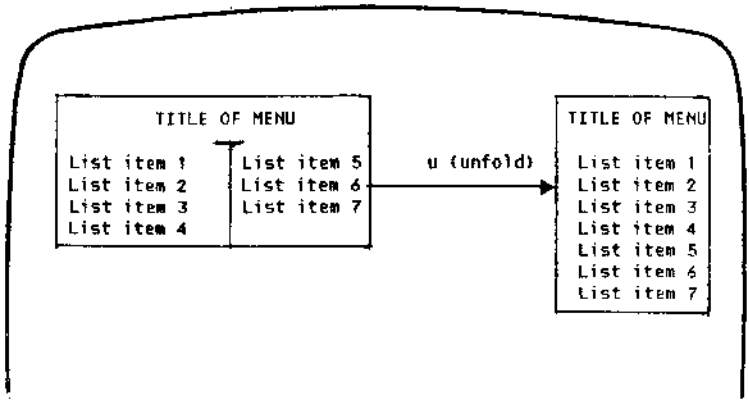


Fig. C-6 The effects of unfolding a box.

CTRL r truncates a column from the left hand side of a folded box, so that the right hand column moves one space closer to the left hand column.

q terminates the editing commands and returns to the screen formatter command mode, truncating the edited box from the right as necessary. If it is an input screen which has been edited, various information is requested, as described under command **n** of the FORMATTING COMMANDS.

ADDING NEW OPERATIONS TO SAF

THE PROGRAMMING INTERFACE

Execution Routines

An execution routine is a "C" function which creates and executes the command to perform the operation selected by the SAF user. One of these must exist for every operation which is added to the system. It must build the command in the external character buffer **command**, including any necessary parameters, and must then execute the command using the routine:

```
exec_command(size,hlen);
```

where **size** may take the values:

- 0, for 1 screen of output or less with no headings
- 80, for one screen of output, or more, at a width of 80 characters, with optional heading lines
- 132, for one screen of output, or more, at a width of 132 characters, with optional heading lines

and **hlen** is the number of lines for the heading to the output, below which the output will be displayed. **Note:** the more heading lines are specified, the smaller will be the window for displaying the output pages.

The following is the simplest form of execution routine which may exist:

```
#include <sysadm.h>

void
routine name(x,y,z)
int x,y,z;
```

```
{
sprintf(command,"command name %d,%d,%d", x,y,z);
exec_command(0,0);
}
```

All the option values selected from the option screens are presented as INTEGER parameters (starting from 1) in the order of presentation on the screen. (In this example, there are 3 option values).

Input will be found in the array *tc_acc[n]*, where n is the number of the input as requested of the user by SAF, starting from 0. **Note** that input for a screen will overwrite input from the previous screen. Therefore, if there is more than one input screen for an operation, input from the previous screen must be saved.

Nothing must be displayed directly to the screen. Error messages may be displayed by calling the *error* routine:

```
error(string);
```

Screen Pre-accept Routines

These routines, of type INT, are called before an input screen is displayed, and would typically be used to initialise all input fields, or to decide whether or not input from the screen is needed (according to values of previous option boxes). If input is not needed, the screen should not be displayed.

The value of an option box is in the range 1-n (n being the number of values for that option). The current value of any option box will be returned as an *int* from the routine:

```
BOXval("id");
```

where **id** is the id of the box, as given by the **i** command

in the Screen Formatter. The following values must be 'returned' by the screen pre-accept routine:

AV_OK to accept the screen

AV_NEXT to skip the screen

Any other 'return' will cause a program abort for debugging purposes.

Field Pre-accept Routines

These routines of type void are called before the user enters data into an input field. They may be used to provide a default value for an input field. The default value should be moved, as a string, into the buffer *tc_acc[tc_accidx]* with *strcpy(3)*.

Field Post-accept Routines

Theses routines, of type INT, are used to validate input from Input screens. They are called without parameters, but using external input.

The input for the routine will be found in the buffer *tc_acc[tc_accidx]*.

One of the following values must be 'returned':

AC_OK the input is valid and accepted, move on to next field

AC_NOTOK the input is not valid and needs re-entering

AC_NEXT skip the rest of the fields

AC_PREV go back to the previous screen

RECREATING SAF

Upon exiting *mengen*, you will be informed if it is necessary to recreate SAF. It will be necessary if operations have been added or removed, or if screen or field accept routines have been added or removed. It will not be necessary if box positions have been changed, or lines within boxes have been changed.

Makefile in /lib/saf should be edited to include any new source modules. Then, typing:

make

will recreate the saf program. All the necessary object files will be created by the makefile.

CONFIGURING SAF FOR DIFFERENT LANGUAGES

This facility has not yet been implemented.

TERMINAL CONFIGURATION

There is a database which must hold a definition of codes for each terminal type on which SAF and MENGEN are to be run. These codes define how screen attributes, commands, etc. will be produced on each given terminal type. The database is held in /lib/saf/termcodes, and must be edited as necessary. The database as supplied for SAF, contains codes for an Olivetti WS584 in VT100 emulation mode. Entries must be added for any other terminal types used. The codes may be obtained from the terminal specification manual. Below is a list of the codes as they are defined for an Olivetti WS584 in VT100 emulation mode. **Note:** the codes must be defined in upper case, and **each code** below should be defined, if supported by the hardware. Entries in lower case will be taken as comments. The functionality of undefined codes cannot be used, thus degrading the visual aspect of SAF, and hence

its ease of use. The format is:

- the upper case code terminated by at least one white space character

- the definition of the code terminated by at least one white space character

- an optional comment.

```
vt100
        movement commands (relative)
        HO \033[0;0H        home
        CU \033[1A          cursor up (no scroll)
        SU \033M            scroll up
        CD \033[1B          cursor down (no scroll)
        SD \033E            scroll down
        CR \033[1C          cursor right non destructive
        CL \033[1D          cursor left non destructive

        movement commands (absolute)
        CM \033[%R;%CH

        clear commands
        LC \033[2K          whole line clear
        RC \033[0K          rest of line clear
        SC \033[2J          whole screen clear
        PC \033[0J          rest of page clear

        attributes
        RS \033[0m          reset

        HI \033[1m          highlight
        RE \033[7m          reverse
        UL \033[4m          underline
        BK \033[5m          blink

        RH \033[7;1m        reverse + highlight
        UH \033[4;1m        underline + highlight
        BH \033[5;1m        blink + highlight
```

```
RB \033[7;5m      reverse + blink
RU \033[7;4m      reverse + underline
BU \033[4;5m      blink + underline

RBH \033[7;5;1m   reverse + blink + highlight
RUH \033[7;4;1m   reverse + underline + highlight
BUH \033[5;4;1m   blink + underline + highlight
RBU \033[7;5;4m   reverse + blink + underline

RBUH \033[7;5;4;1m  rev + blink + uline + hlight

graphics commands
EG \033(0         enter graphics
XG \033(B         exit graphics

graphics characters
TR k              box - top right
TL l              box - top left
BR j              box - bottom right
BL m              box - bottom left
HL q              horizontal line
VL x              vertical line
TT w              top T
BT v              bottom T
RT u              right T
LT t              left T
CX n              cross
BC a              block
DI`               diamond shape (or equivalent)
CA                cursor appearance
                  (not available for VT100)

the `mark` character
MK \033[7m\033(0`\033(B\033[0m

screen formatting
SS \033[?4h       slow scroll
FS \033[?4l       fast scroll
BS \033[?3h       big screen - 132 chars
LS \033[?3l       little screen - 80 chars
SR \033[%R;%Cr    create scroll region
```

```
RR \033[r          reset scroll region
DT \033#3          double height letters (top line)
DB \033#4          double height letters (bottom)
DW \033#6          double width
SA \033[?7h        set auto wrap around
DA \033[?7l        disable auto wrap around
ST \033H           set tab
CT \033[3g         clear tabs

special input keys
AU  \033[A         arrow up
AD  \033[B         arrow down
AR  \033[C         arrow right
AL  \033[D         arrow left
SP  \040           space char in graphics
```

Notes

\nnn is interpreted as an octal code
^l is interpreted as the control character

When using the codes *CM* and *SR* defined above:
%R is replaced with the row number (1-n)
%C is replaced with the column number (1-n)

If *R* and/or *C* are upper case in the definition, replacement is with ASCII value
If *R* and/or *C* are lower case in the definition, replacement is with binary value.
The optional constant expressions:

+number to be added to value

or:

-number to be subtracted from value

may follow *R* and/or *C*.

Code 4051340 J (0)
Printe in Italy

olivetti