

Global Utilities Manual Version 8.1

MBSMV8.1/64

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electrical, mechanical, photocopying, recording or otherwise, without the prior permission of TIS Software Limited.

Copyright 1994 -2001 Global Software

MS-DOS is a registered trademark of Microsoft, Inc.

Windows NT is a registered trademark of Microsoft, Inc.

Unix is a registered trademark of AT & T.

C-ISAM is a registered trademark of Informix Software Inc.

D-ISAM is a registered trademark of Byte Designs Inc.

Btrieve is a registered trademark of Pervasive Technologies, Inc.

TABLE OF CONTENTS

Section Description	Page Number
1. Introduction.....	10
2. \$A - Inspecting and Changing Unit Assignments	13
2.1 The Unit Prompt.....	13
2.2 To Make an Assignment	13
2.3 Deleting and Purging Assignments.....	14
2.4 \$A Example.....	15
2.5 \$A Operating Notes	15
3. \$AUTH - Authorization Customisation	17
(see Global System Manager Manual, Chapter 6)	
4. \$BYE - Terminate Global System Manager	18
(see Global Operating Manual, Chapter 4)	
5. \$C - Clear Screen.....	19
6. \$CALC - Calculating IS File and Sort Requirements	21
6.1 The Calculation Type Prompt	21
6.2 The Calculation Screens	21
6.3 The Sort Workspace Calculation	21
6.4 The Multi-phase Sort Calculation.....	22
6.5 \$CALC Examples	22
7. \$CATAL - Catalogue Maintenance	24
7.1 The Catalogue File Prompt	24
7.2 To Quit	24
7.3 Creating a New Catalogue	24
7.4 Maintaining an Existing Catalogue.....	25
7.5 The Catalogue Entry Prompt	26
7.6 Adding a New Entry.....	26
7.7 Deleting an Existing Entry	27
7.8 Amending an Existing Entry	27
7.9 The Catalogue Listing	27
7.10 Example 1 - Catalogue Maintenance.....	28
7.11 Example 2 - A Catalogue for Flexible Installation	28
8. \$CONV - File Conversion	30
8.1 The Input File Prompt.....	30
8.2 To Quit	30
8.3 The Output File Prompt.....	30
8.4 The Key and Record Length Prompts	32
8.5 The Conversion Process.....	32
8.6 Example 1 - Creating an Empty Indexed Sequential File	32
8.7 Example 2 - Creating an IS File from an RS File	33
8.8 Example 3 - Reorganising an Indexed Sequential File.....	33
9. \$CUS - System Customisation	34
(see Global System Manager Manual, Chapter 6)	

10.	\$D - Altering the Date and Time	35
11.	\$DIR, \$DIRP, \$DIRCUS - Directory Utilities	37
11.1	The Unit Prompt.....	37
11.2	To Quit	37
11.3	The Directory Display	39
11.4	Cursor keys, Page, and <CR>	40
11.5	Selecting and Marking Files	40
11.6	Selection Commands	41
11.7	Marking Commands.....	44
11.8	Sorting Commands and Miscellaneous Commands	45
11.9	Operating on Files	46
11.10	\$DIRP Directory Utility.....	49
11.11	\$DIRCUS.....	50
12.	\$DLIB - Data File Library Maintenance	52
12.1	Introduction	52
12.2	General Dialogue Structure.....	53
12.3	Inspecting an Existing Library	55
12.4	Creating a New Library.....	55
12.5	Amending an Existing Library	56
12.6	To quit	56
12.7	Library Re-organisation	56
12.8	Some Common Errors during Data Library Processing.....	56
12.9	I/O Error Handling.....	57
12.10	Inspect and Extract Operations	58
12.11	Including and Comparing Members	61
12.12	Rename and Delete Operations	64
12.13	Space Saving Operations	66
12.14	Volume checking Operations	67
12.15	ABA - Abandoning the Library Update	67
13.	\$E, \$END - Signing Off	68
	(see Global Operating Manual, Chapter 4)	
14.	\$EXEC - Timed Execution of Programs	69
14.1	Starting \$EXEC.....	69
14.2	Daily Execution	69
14.3	Restrictions in \$EXEC	71
15.	\$F - File Maintenance.....	72
15.1	The Device Prompts	72
15.2	Listing Instruction Codes and Changing Devices	72
15.3	Listing and Printing Volume Directories.....	72
15.4	Verifying Volumes and Files.....	74
15.5	Condensing Volumes and Truncating Files.....	74
15.6	Scratch Protection and Password Protection.....	77
15.7	Renaming Volumes and Files	79
15.8	Initializing and Scratching Volumes	79
15.9	Backing Up Volumes	79
15.10	Copying, Moving and Deleting Files	80
15.11	Comparing Volumes and Files	81

15.12	Serialization Instructions	82
15.13	Job Management Instructions	83
15.14	Establishing Volumes, Allocating Files	83
15.15	Patching Monitor Files and Installing Bootstraps	84
15.16	Adding and Amending Volume Descriptions	85
15.17	Selection Codes.....	86
15.18	Options to the SIZE: prompt	88
15.19	Report files.....	89
16.	\$FIND - Find Files Utility.....	90
16.1	Using \$FIND in Normal Mode	90
16.2	Using \$FIND in Special Mode	93
17.	\$GROUP - Operator Group Maintenance.....	102
17.1	\$GROUP Dialogue	102
17.2	Amending an Existing Group	102
17.3	Creating a New Group.....	103
17.4	Deleting a Group.....	103
17.5	Printing Group File Details	103
17.6	Next Page	103
17.7	Exiting \$GROUP	104
18.	\$INSOFT - Install Software.....	105
	(see Global Operating Manual, Chapter 2)	
19.	\$INSPECT - Inspect Text and Print Files	106
19.1	The Inspect File Prompt	106
19.2	To Quit	106
19.3	Using Wide Screen Mode	106
19.4	File Inspection Instructions.....	107
19.5	F - To Search for a String within a Line.....	108
19.6	G - To Search for a String starting a Line	108
19.7	\$LS - Set Line Start	109
19.8	WS - Set Write Start Pointer.....	109
19.9	WE - Set Write End Position.....	109
19.10	E - End Inspection of Current File.....	110
19.11	W - Enter Wide Mode.....	110
19.12	N - Re-enter Narrow Mode.....	110
19.13	Example 1 - Examining a report file.....	110
19.14	Example 2 - Examining Error and Warning Messages.....	110
19.15	Example 3 - Printing Part of a Listing	111
19.16	Example 4 - Converting an RS File to a Text File	111
19.17	Example 5 - Concatenating Text	112
20.	\$L - List/Dump Files	113
20.1	The Input File Name Prompt	113
20.2	To Quit	113
20.3	The Listing Option Prompt	114
20.4	Output to the Console	116
20.5	Output to File	116
20.6	Operating Notes.....	116
20.7	Example 1 - Displaying Records from a File	117
20.8	Example 2 - Dumping an IS File	117

20.9	Example 3 - Dumping the Directory Track of a Volume	118
21.	\$LIB, \$LIBM - Library Maintenance	119
21.1	Introduction	119
21.2	General Dialogue Structure.....	120
21.3	Inspecting an Existing Library	122
21.4	Creating a New Library.....	122
21.5	Amending an Existing Library	123
21.6	To Quit	123
21.7	Library Re-organisation	124
21.8	Global Symbol Checking.....	124
21.9	Some Common Errors during Library Processing.....	124
21.10	I/O Error Handling.....	125
21.11	Inspect and Extract Operations	126
21.12	Including and Comparing Members	132
21.13	Rename and Delete Operations	135
21.14	Space Saving Operations	137
21.15	Abandoning the Library Update	138
21.16	Dispersed Program Library Support	139
21.17	Job Management Support.....	144
21.18	\$LIBM	146
22.	\$LOG - Event Log Viewing.....	147
22.1	The \$LOG Menu	147
22.2	The Purge Function	147
22.3	Producing Reports.....	148
23.	\$M - Send Message to Other Users	151
24.	\$MAIL - Mailing Utility.....	153
24.1	The \$MAIL Menu	153
24.2	Send Mail	153
24.3	Take Telephone Messages.....	157
24.4	List Mail	158
24.5	Reorganise Mail File.....	165
24.6	Operator Customisation	168
24.7	Lock with Password.....	169
25.	\$MH, MN - Menu Handler and Menu Maintenance	171
	(see Global System Manager Manual, Chapter 7)	
26.	\$OPID - Operator Full Name List.....	172
26.1	\$OPID Dialogue.....	172
26.2	Using the Operator Name Information.....	172
27.	\$P - Advance the Printer.....	174
	(see Global System Manager Manual, Chapter 5)	
28.	\$PRINT - Print Files	175
28.1	Printing to Auxiliary Terminal Ports.....	175
28.2	Problems and Restrictions with AUXC/AUXP Printing	177
29.	\$RCP - Record/Playback	178

	(see Global System Manager Manual, Chapter 5)	
30.	\$REMOTE - Copy Files Between Remote Computers	179
	(see Global Operating Manual, Chapter 4)	
31.	\$REORG - Reorganise Domain or Volume	180
	(see Global Operating Manual, Chapter 4)	
32.	\$RESTORE - Restore From Backup	181
33.	\$S - Obtain System Information	184
	(see Global System Manager Manual, Chapter 5)	
34.	\$SAVE - Create Backup	185
34.1	Introduction	185
34.2	Creating a Security Catalogue	187
34.3	Saving a Backup Cycle using \$SAVE	188
35.	\$SORT - Multi-Key Sorting	193
35.1	The Option Prompt	193
35.2	The Input File Prompt	193
35.3	To Quit	194
35.4	The Output File Prompt	194
35.5	The Work File Prompt	194
35.6	The Key Start Prompt	195
35.7	The Respecify Keys Prompt	196
35.8	The Sort	196
35.9	Example 1 - Sorting a Transaction File	196
35.10	Example 2 - Tag Sorting a Large Master File	197
36.	\$SP, \$SPOOL, \$SPS - Spooler Control and Spooler Status	199
	(see Global System Manager Manual, Chapter 8)	
37.	\$STATUS - Controlling a Multi-user System or Network	200
37.1	The Status Report	200
37.2	Task Control Instructions	203
38.	\$T - Terminal Information	212
	(see Global System Manager Manual, Chapter 5)	
39.	\$TDUMP - Tape Dump and Restore	213
	(see Global Operating Manual, Chapter 4)	
40.	\$TED - Text Editor	214
	(see Global System Manager Manual, Chapter 5)	
41.	\$U - Unit Information	215
41.1	Devices on a Network or Unix Configuration	215
41.2	Diskette Devices	216
41.3	Local Hard Disks	216
41.4	Printers	216
41.5	Hard Disks on the network Master	216
41.6	Tape Drives	216

42.	\$USAGE - File Block Dump	218
42.1	Using \$USAGE	218
42.2	The \$USAGE Diagnostic Messages.....	220
42.3	The \$USAGE Warning Messages	221
43.	\$V - Preparing Diskettes and Hard Disks	222
	(see Global Operating Manual, Chapter 4)	
44.	\$VOL - List Volume Directory	223
44.1	The \$VOL Output Unit Prompt.....	223
44.2	The \$VOL Option Prompt.....	223
44.3	The \$VOL Title Prompt	223
44.4	The \$VOL Unit Prompt.....	223
45.	\$VOLSAV - Save/Restore Domain Layout	225
45.1	The \$VOLSAV Mode Prompt.....	225
45.2	Using \$VOLSAV to Save a Domain Layout	225
45.3	Using \$VOLSAV to Restore a Domain Layout.....	225
46.	\$ZAP - Applying a Zap	227
	(see Global System Manager Manual, Chapter 6)	

APPENDICES

Appendix Description	Page Number
A How Program Errors are handled	228
B Program Checks	231
C STOP Codes	233
D EXIT Codes	260
E Special Global System Manager files	279
F Running System Commands	281

1. Introduction

The Global Utilities Manual is one of a set of inter-related manuals listed below:

- Global System Manager Manual
- Global Operating Manual
- Global Utilities Manual
- Global Configurator Manual
- Global File Converters Manual
- Global Speedbase Development User Manual
- Global Speedbase Development Language Manual
- Global Development System Subroutines Manual
- Global Development File Management Manual
- Global Development Toolkit Manual
- Global Development Job Management Manual
- Global Cobol User Manual
- Global Cobol Language Manual
- Global Cobol Screen Presentation Manual
- Global Cobol Data Management Manual
- Global Cobol Screen Support Manual

The **Global System Manager Manual** contains descriptions of the most commonly used Global System Manager and Speedbase Presentation Manager commands. It is recommended preliminary reading if you are new to Global System Manager since it provides a short, end-user oriented description of the environment established by Global System Manager and Speedbase Presentation Manager. This manual also explains how to customize Global System Manager.

The **Global Operating Manual (o/s)** explains how to initiate Global System Manager on a particular operating system. For example, the "Global Operating Manual (Unix)" explains how to install, run and configure Global System Manager under Unix. The various operating manuals also describe any additional utilities, and additional functionality in standard utilities, which are dependent on the host operating system.

The **Global Utilities Manual** extends the Global System Manager Manual with descriptions of the more sophisticated utilities and detailed information required by system supervisors and computer specialists. Programmers developing Global Cobol software will find the account of program checks and the lists of STOP and EXIT codes invaluable.

The **Global Configurator Manual** explains how you can update the configuration file distributed with Global System Manager to change parameters such as the maximum number of files that can be open at any one time, the maximum number of users, buffer sizes, hardware devices and addresses, and so on.

The **Global File Converters Manual** describes the Unix and MS-DOS file converters. This manual also explains how you can write your own file converter to transfer files between the host operating system and Global System Manager.

The **Global Speedbase Development** User Manual describes the commands used for Speedbase program preparation, debugging and maintenance. These include a text editor as well as the Speedbase compiler itself.

The **Global Speedbase Development Language Manual** describes the Global Speedbase language used to develop programs to run under Speedbase Presentation Manager.

The **Global Development System Subroutines Manual** forms a logical extension of the Global Speedbase Development Language Manual and the Global Cobol Language Manual. It contains the system subroutines used for Date and Time Conversion, Arithmetic and Data Conversion, Program Management and System Management, together with special coding techniques, storage management facilities and scientific calculation facilities.

The **Global Development File Management Manual** contains a detailed description of the file organizations and access methods available in Global System Manager, together with the associated system subroutines.

The **Global Development Toolkit Manual** describes a collection of command programs designed to aid the production of large systems written in Global Cobol or Global Speedbase. Separate sections also deal with Product Translator, Intermediate Code generated by the compilers, the Metajob Management and the Macro Preprocessor languages.

The **Global Development Job Management Manual** explains how you can create catalogued procedures, known as job files, to run Global command programs, or your own applications, in a non-interactive mode in which, after initial parameterisation, operator input is supplied from the job file rather than from the console. The system might typically be used in conjunction with the \$V command program to initialize a series of diskettes or hard disk units. You can optionally suppress console dialogue from programs run under job management.

The **Global Cobol User Manual** describes the commands used for program preparation, debugging and maintenance. These include a text editor, linkage editor and librarian, as well as the Global Cobol compiler itself. The librarian enables you to group subroutines or programs into individual library files, each of which is capable of holding up to 100 members. There is also a source program cross reference utility and a string search utility which enables you to examine any number of text files for the occurrence of particular words or phrases. A powerful symbolic debugging system aids program testing: with it you are able to set traps in programs and on variables and then examine, and optionally modify, any named field before resuming execution.

The **Global Cobol Language Manual** describes the Global Cobol language used to develop programs to run under Global System Manager.

The **Global Cobol Screen Presentation Manual** contains the Global Cobol commands used for screen displays and accepts together with the special Global Cobol verbs (MAPIN, MAPOUT and MAPCLEAR) that are used with the \$FORM screen formatting command. \$FORM allows you to create a display screen format table known as a map. Then, instead of having to DISPLAY and ACCEPT individual fields you can simply code the MAPOUT statement to output the entire map, and the MAPIN statement to obtain all the input fields. Coding effort and program size is, as a result, greatly reduced in certain applications. The map is declared in a map definition (MD) coded in the data division. This manual also contains some advanced screen handling techniques and describes the AutoGuide facility.

The **Global Cobol Data Management Manual** is a complete guide to the Global Data Management system, including the DMAM access method, various associated system subroutines and the utility programs that form the Data Management System.

The **Global Cobol Screen Support Manual** describes how to create a Terminal Attribute Program (TAP) in order to support a new type of terminal under Global System Manager. This involves preparing a special format source program containing details of the keyboard and sequences used by the terminal. The \$TAP command is used to create the TAP from the source. This manual also describes the \$T command in complete detail.

2. \$A - Inspecting and Changing Unit Assignments

The \$A command allows you to examine, create, change or delete unit assignments. Any assignments made using \$A are temporary and only remain in force until you sign off, override them by using \$A again or run a program with permanent assignments. Unit assignments may also be changed when a menu entry is selected.

2.1 The Unit Prompt

When you run \$A it lists the assignments currently in force and then displays its unit prompt, which enables you to perform all necessary assignment operations (see Figure 2.1).

To redisplay the assignments currently in force, key the single character ? in response to the unit prompt:

```
UNIT: ?
.....
..... (list of current assignments)
.....
UNIT:
```

To exit from \$A key <ESC> or <CR> in response to the unit prompt.

2.2 To Make an Assignment

To make an assignment you must key the unit-id followed by the unit address. For example:

```
UNIT: DSK ADDRESS: 210
UNIT:
```

assigns the 'logical' unit-id DSK to the 'physical' address 210, overriding any previous assignment of DSK. The logical unit-id you specify cannot be of the same form as a valid physical unit address (i.e. it cannot take the form *nnn* or *xnn*).

In most Global System Manager configurations there will be a number of free entries in the assignment tables available for your own unit-ids. If you attempt a new assignment and there is no space left for it the following error message will appear:

```
ASSIGNMENT TABLES FULL
```

No assignment is made and you must delete an existing unwanted assignment before you can make the new one. Up to 29 assignments can be held in the table.

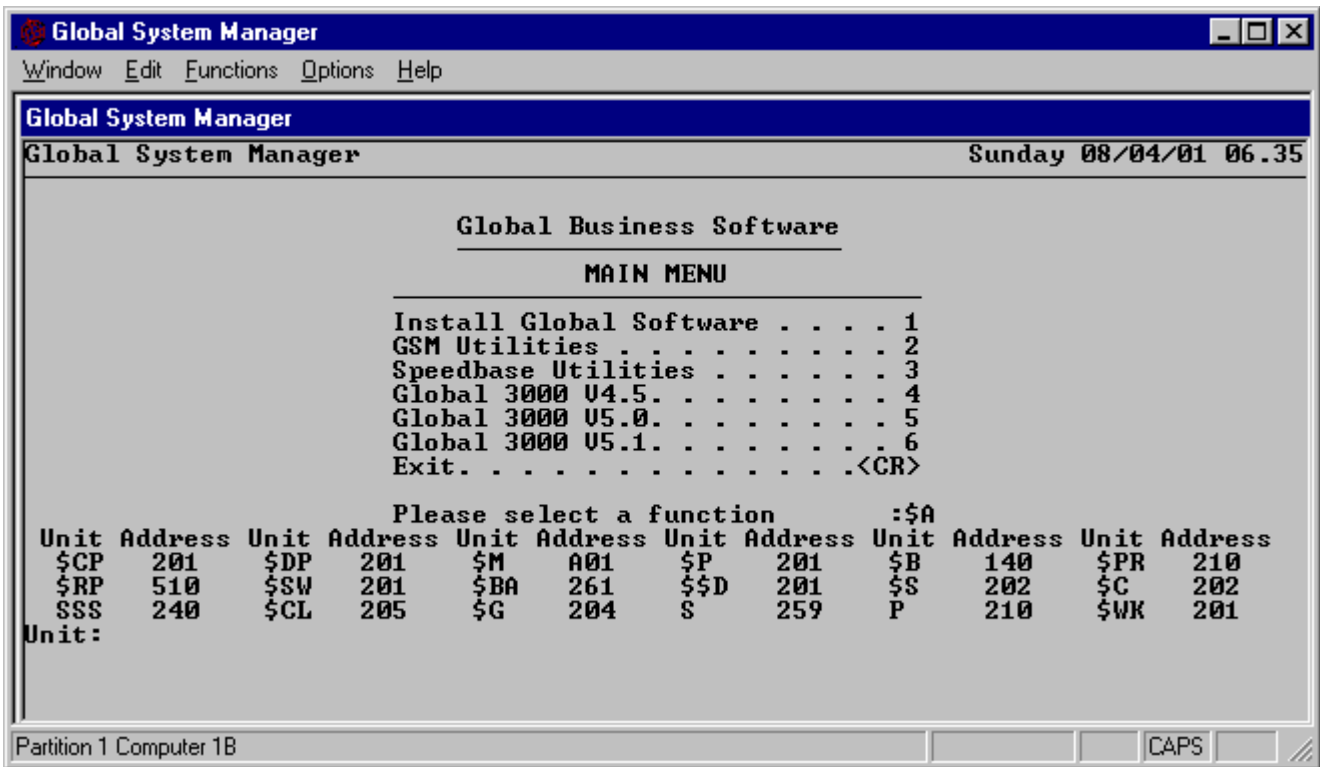


Figure 2.1 - After running \$A

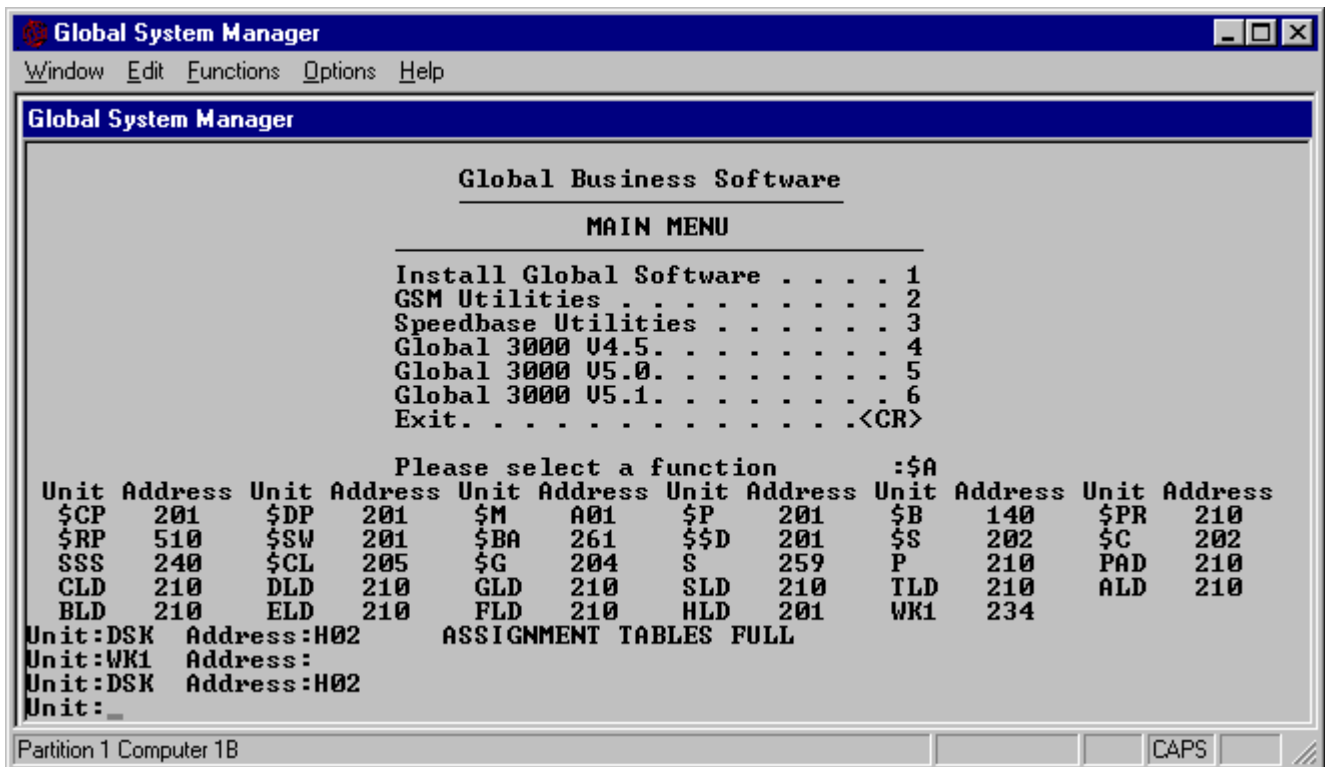


Figure 2.2 - Example dialogue

2.3 Deleting and Purging Assignments

To delete an assignment you must key the unit-id you wish to remove in response to the unit prompt and key <CR> in response to the subsequent address prompt:

```
UNIT:WK1 ADDRESS:<CR>
UNIT:
```

A deletion request is always followed by a new unit prompt. If the unit-id you specified was not actually assigned, the tables will remain unchanged by the request.

To purge the assignment table (that is to delete all new assignments made since you signed on) key <CTRL A> to the unit prompt:

```
UNIT:<CTRL A> ASSIGNMENTS PURGED
UNIT:
```

Note that this does not delete or reset permanent assignments (established by Global System Manager itself or by the use of the \$CUS Permanent Unit Customisation) which are in force when you sign on.

2.4 \$A Example

You attempt to assign unit DSK, but 29 unit assignments have already been made so the table is full. You decide the assignment for unit-id WK1 is no longer required. Therefore you delete this assignment by keying <CR> to its address prompt. You are then able to assign DSK successfully.

The resultant screen dialogue is shown in figure 2.2.

2.5 \$A Operating Notes

By altering the assignment of \$P you can change the device from which application programs (those whose names do not start with a \$) are loaded, and by modifying the \$PR assignment you can change the logical printer used by Global System Manager commands. If you run Global Cobol the assignment of \$C determines which unit the compiler and the \$FORM command use for their work files; and the \$\$S assignment establishes the volume within which the system subroutine libraries, C.\$MCOB, C.\$APF and C.\$PAGES reside.

To run a program with unit assignments other than those that have been set up in your menu, you can exit from the menu by keying READY to the menu selection prompt, set up the required assignments using \$A, and then enter the program directly (e.g. key BW to enter Global Writer) from the Global System Manager "READY" prompt (see Appendix F).

Alternatively, you might delete some or all of the assignments associated with a particular program, enter the program directly, and thus force the program to prompt for the unit assignments when they are needed.

Please note that the <SYSREQ> I system request (see section 4.13 of the Global System Manager Manual) provides some of the functionality of \$A.

Logical Unit Assignment	Description
\$B	The back-up disk drive used by \$SAVE and \$RESTORE. Also used by \$CUS to save and restore system customisation information.
\$BA	The unit address of the BACRES diskette.
\$C	The unit address of the work unit used by \$COBOL and \$FORM.

\$CL	The unit address that \$COBOL uses as the default when prompting for each Copy Library.
\$CP	The unit from which Global System Manager command programs (i.e. those whose file names start with a \$) are loaded (i.e. SYSRES).
\$DP	The unit from which diagnostic programs are loaded, and from which Global System Manager will attempt to load command programs if not found on \$CP.
\$IP	The unit address of the SYSIPL diskette if Global System Manager is not loaded from hard disk.
\$LG	The unit used to hold the Event Logging data files, \$\$LOG and \$\$MLOG.
\$M	The unit on the master computer used to hold \$\$USER which contains network-wide information on network configurations.
\$ML	The unit used to hold the Mail data file \$MAIL.
\$P	The unit from which non command programs (i.e. those whose file names do not start with a \$) are loaded.
\$OV	If this optional unit is assigned, the P.\$OV\$ system request library is automatically copied from SYSRES to the \$OV unit when Global System Manager is initiated. The \$OV unit is then used to load system requests and hold the temporary P.\$OV\$xx system request work files.
\$PR	The physical printer device to which reports will be sent if spooling is not being used or the spool unit on a disk if spooling is being used.
\$RP	The real printer on which spooled reports are printed if the spooler is in use.
\$S	The unit on which the System Subroutine libraries (C.\$MCOB, C.\$APF and C.\$PAGES) reside. This assignment is used by \$LINK when extracting System Subroutines.
\$SS	Temporary System Save unit. This unit is only assigned during Global System Manager installation.
\$SW	The unit used for the 'swap file' (\$\$SWAP) in multi-user configurations.
\$WK	Work unit used by system programs: \$LOG, \$MH etc.

Table 2.1 - Global System Manager Unit Assignments

3. \$AUTH - Authorisation Customisation

The \$AUTH utility is fully documented in Chapter 6 of the Global System Manager Manual.

4. **\$BYE - Terminate Global System Manager**

The functionality of \$BYE is dependent on the host operating system. \$BYE is documented in Chapter 4 of the Global Operating Manual.

5. \$C - Clear Screen

\$C clears the screen, restores the cursor display and re-establishes normal brightness. This is useful if a program error has led to a garbled display, or if you wish a display of screen dialogue to start at the top of the screen.

Figures 5.1 and 5.2 show the result of keying \$C.

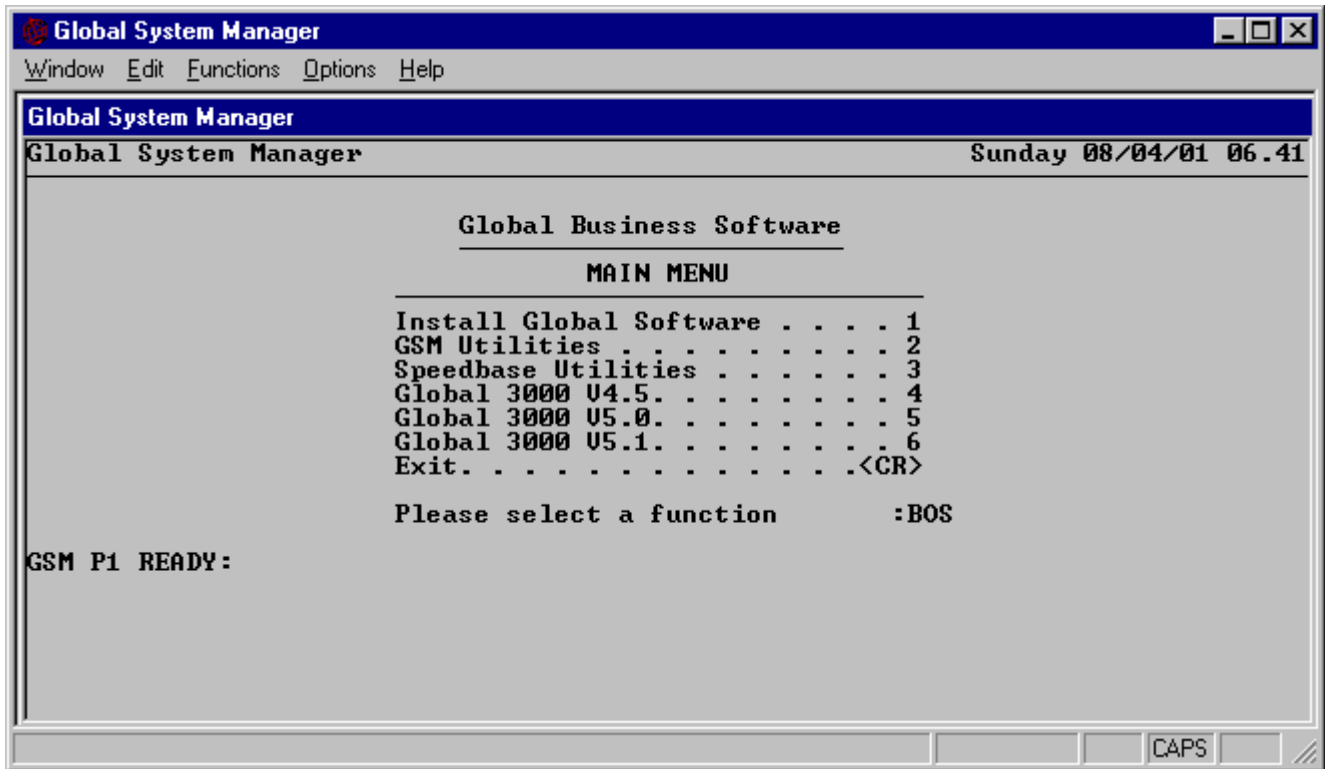


Figure 5.1 - Before keying \$C

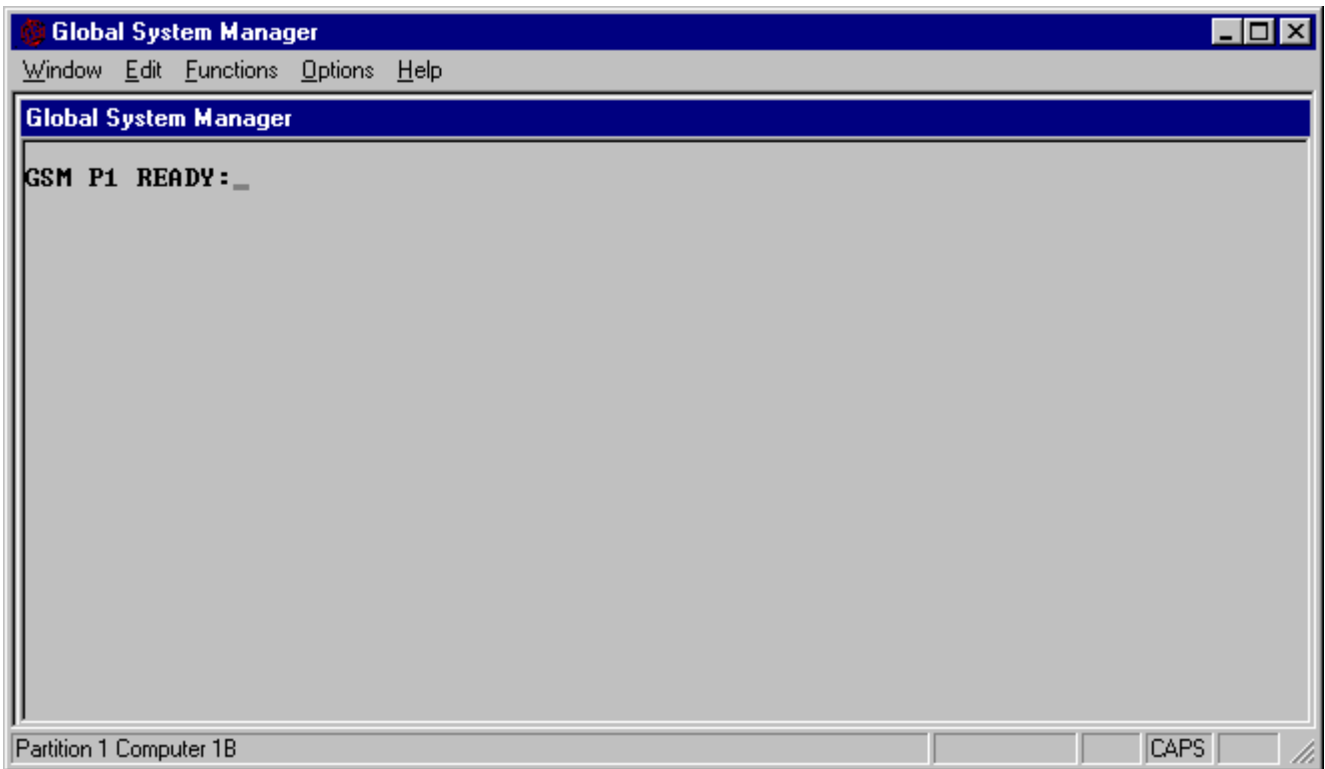


Figure 5.2 - After keying \$C

6. \$CALC - Calculating IS File and Sort Requirements

The \$CALC command allows you to calculate the space requirements of indexed sequential (IS) files and the work files used by sort programs. Two types of sort are catered for: those based around the Global Cobol SORT verb, and those incorporating MSORT\$ - the multi-phase sort system subroutine. Note that \$SORT makes use of the SORT verb, not MSORT\$.

6.1 The Calculation Type Prompt

\$CALC first displays a screenful of help text describing the 3 options and then asks for the user's selection with the following prompt:

```
Key Indexed file size, Sort workspace, Multi-phase sort workspace, <ESC>:
```

You may reply I to perform an IS file size calculation, S to perform a sort capacity calculation, or M to perform a multi-phase sort capacity calculation.

To exit from \$CALC key <CR> or <ESC> in response to any of the base-line prompts.

6.2 The Calculation Screens

Once you have selected a calculation type \$CALC displays one of three different calculation screens which prompt for information pertinent to the calculation. You may enter your own values or use the default values presented by \$CALC. Use cursor up and cursor down to move from field to field.

\$CALC performs the calculation when you respond to the last field, or key <ESC>. The results of the calculation are displayed in the lower half of the screen. The sizes are presented both in bytes and in Kbytes (or Mbytes) rounded to one decimal place (K = 1024, M = 1048576).

Some fields are dependent on the values of others, so you may not have to key responses in all the fields. For example, if you ask for a tag sort calculation the record length is always 4.

Having displayed the results of a calculation \$CALC displays the following prompt:

```
Key Amend parameters, Change calculation, <ESC> to exit:
```

A response of A allows you to modify the parameters of the calculation you have just performed and then re-calculate. If you key C then \$CALC will re-display the initial screen and allow you to choose another type of calculation. Key <ESC> or <CR> to exit from \$CALC.

6.3 The Sort Workspace Calculation

The ordinary sort calculation supplies three figures:

- the free storage required without the work file. This is the amount of memory needed to perform the sort internally without using the work file;
- the free storage required if a work file is provided (which will be much less than the previous quantity);
- the size of work file necessary.

If you are not performing a tag sort, and if you are sorting a single key, and that key is character ascending (not computational), then the SORT verb and MSORT\$ can optimize both performance and space requirements. Answer Y to the prompt below:

```
Is there one key which is character ascending:Y
```

For a multi-key sort, the memory requirement cannot be calculated exactly but is correct within 200 bytes.

You should note that the \$SORT command itself occupies 9.2K of the user area, and the sort routine included in a program when the SORT, RELEASE and RETURN statements are employed occupies 3.8K. One of these figures must always be taken into account when considering sort space requirements.

6.4 The Multi-phase Sort Calculation

The multi-phase sort calculation either tells you that the sort fits within the free user area and displays the message:

```
NO WORK FILE REQUIRED
```

or, if the free user area available is inadequate, displays the size of the work file required.

6.5 \$CALC Examples

Example 1: How much memory and file space is needed to sort 1000 100-byte records of the SADATA file using a single 8-byte character ascending key?

Example 2: How much space is needed for the SAINDEX IS file converted from SADATA if space is to be reserved for a 20% growth to 1200 records?

Figures 6.1 and 6.2 show the screens after these calculations have been performed.

The first calculation shows that the sort requires 3.3K bytes of main memory, and the use of a 108,000 byte (105.5K) work file. It is not possible to sort the records internally, since Global Cobol addressing limitations would prevent the program from accessing 108,000 bytes of memory even if it were available.

The second calculation shows that SAINDEX requires an extent size of 128.2K bytes.

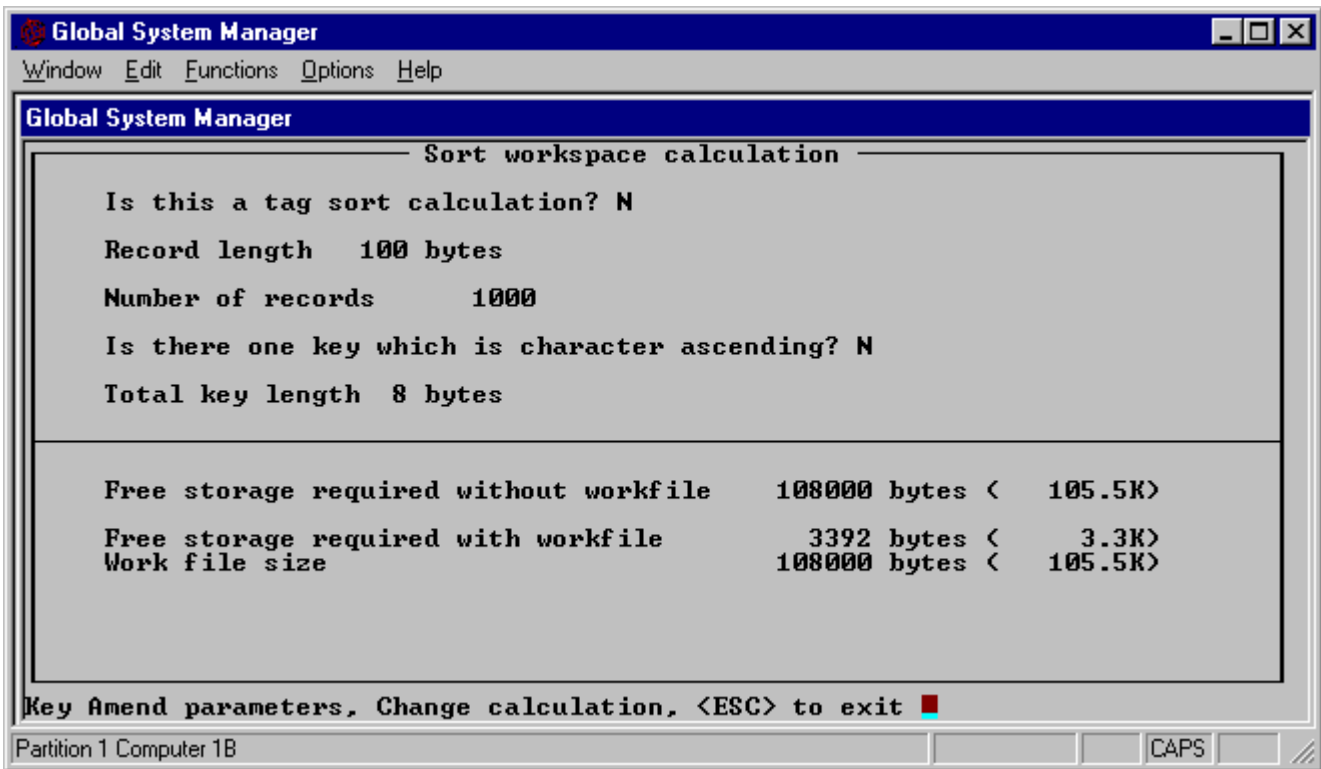


Figure 6.1 - \$CALC Example 1

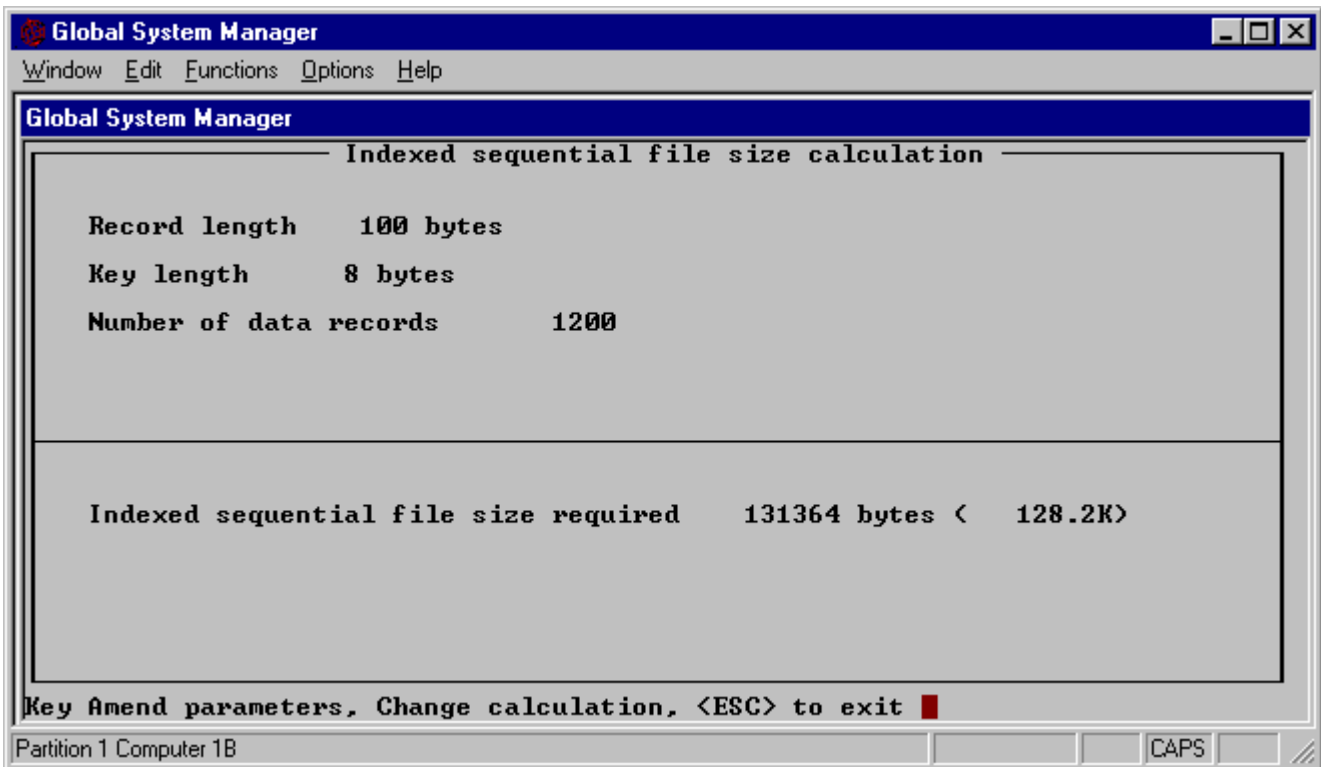


Figure 6.2 - \$CALC Example 2

7. \$CATAL - Catalogue Maintenance

The \$CATAL command allows you to create and maintain special files known as catalogues. Each record of a catalogue associates a particular file-id with the unit and volume that the named file will occupy at run-time. A program can obtain unit and volume information for a list of file definitions by using the CATA\$ system routine, described in the Global Development File Management Manual, in conjunction with a particular catalogue. This mechanism allows you to determine the placement of files when a system is installed by simply supplying the appropriate catalogue.

The Data Security System makes use of a special type of catalogue known as a **security catalogue**. \$CATAL can be used to create and update a security catalogue in the same way that a normal **data catalogue** is maintained. The difference between the two types of catalogue is brought about not by \$CATAL but by the method of operation of the routines that use the files, CATA\$ and the Data Security System. For more detail on data catalogues see under CATA\$ in the Global Development File Management Manual; data security catalogues are dealt with below.

7.1 The Catalogue File Prompt

When you run \$CATAL it displays the catalogue file prompt. You must supply the file-id and unit-id of the catalogue you wish to create or update. The following example dialogue specifies that the catalogue is to be named SACATB and resides on unit 204:

```
GSM READY: $CATAL
;
; (Screen cleared)
;
CATALOGUE MAINTENANCE
CATALOGUE: SACATB UNIT: 204
New? :
```

7.2 To Quit

To quit and return to the menu or GSM READY: prompt, key <ESC> in response to the CATALOGUE prompt.

7.3 Creating a New Catalogue

When the catalogue file is not present on the specified unit Global System Manager checks that you really do intend to create a new catalogue, (and have not simply miskeyed the file information), by displaying the following prompt:

```
New? :
```

If you reply N or <CR> the catalogue file prompt will be redisplayed so you can correct your error. If you reply Y you confirm your intention to create a new catalogue so Global System Manager replies with the catalogue entry prompt:

```
CATALOGUE: SACATB UNIT: 204
New? : Y
```

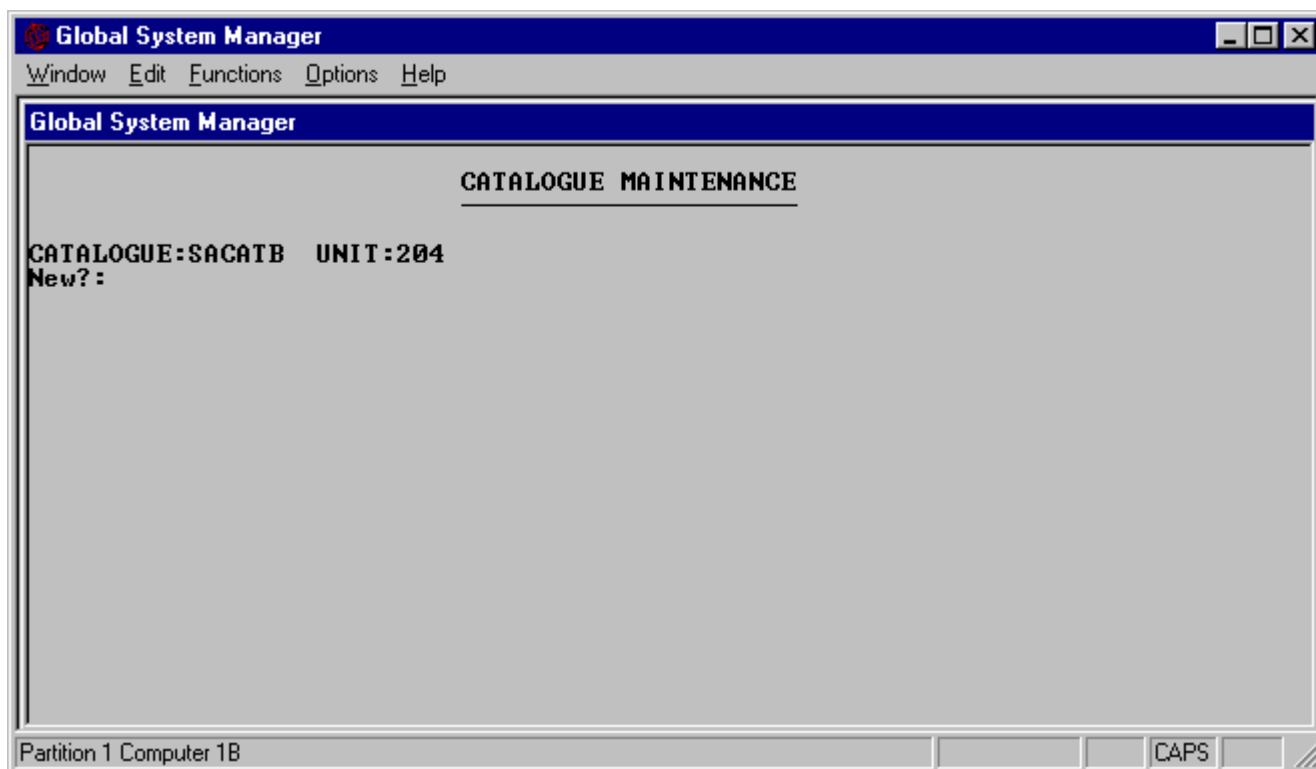



Figure 7.1 - The initial prompts

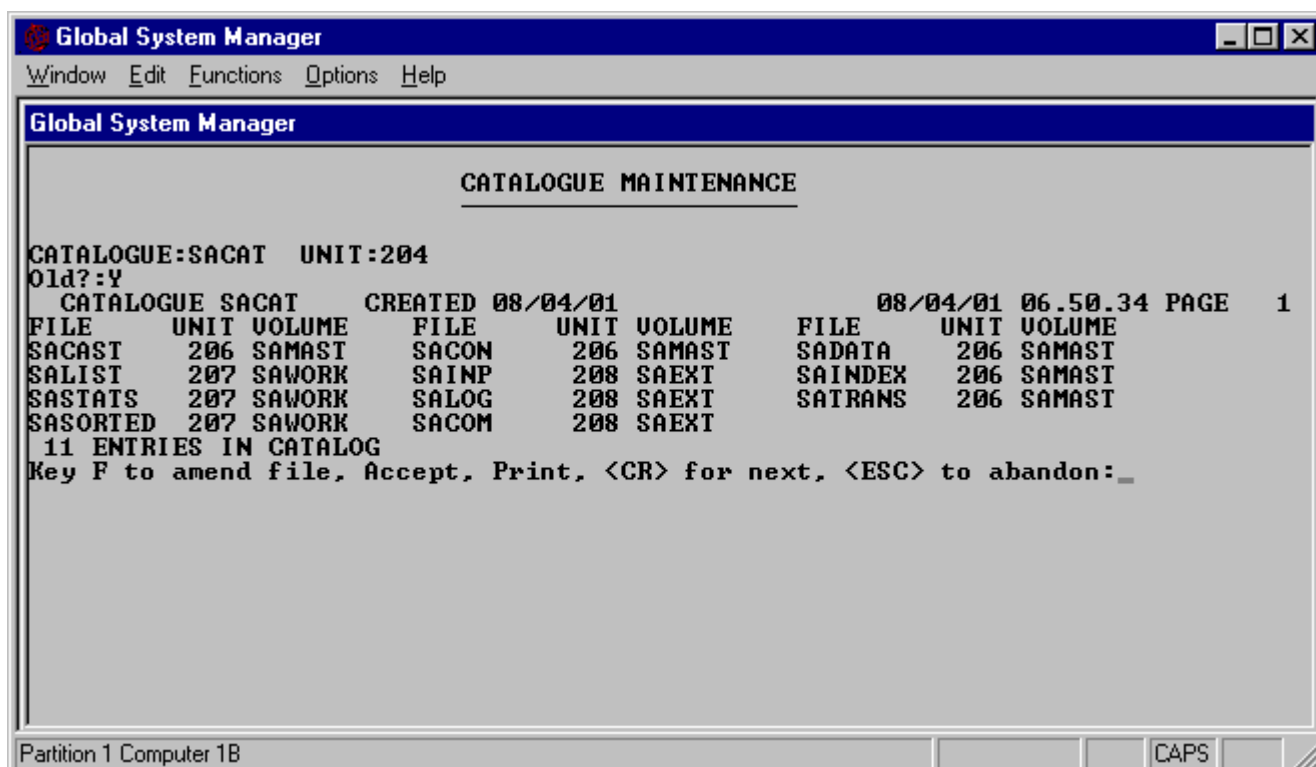


Figure 7.2 - Listing a catalogue

7.4 Maintaining an Existing Catalogue

When Global System Manager finds that the file you have specified is a valid catalogue created by some previous run of \$CATAL it checks that you really intend to maintain an existing catalogue by displaying the following prompt:

Old?:

If you reply N or <CR> the catalogue file prompt will be redisplayed. If you reply Y you confirm your intention to modify an existing catalogue so Global System Manager replies with the catalogue entry prompt:

```
CATALOGUE:SACATA UNIT:204
Old?:Y
```

7.5 The Catalogue Entry Prompt

Once you have specified the catalogue you want to maintain the catalogue list is displayed followed by the a selection prompt as follows:

```
..... (catalogue listing)
Key F to amend file, Accept, Print, <CR> for next, <ESC> to abandon:
```

Key F to this prompt to either create or amend a single entry by specifying its file-id. For example:

```
Key F to amend file, Accept, Print, <CR> for next, <ESC> to abandon:F
File-id:CUSTOMER Unit:
```

Sometimes when building the special security catalogue required by the Data Security System, you may need to specify the same file-id several times. In this case you can distinguish up to three different entries for the same file by keying the file-id by itself, as shown above; the file-id followed by the ' character; and the file-id followed by the " character. For example, CUSTOMER, CUSTOMER' and CUSTOMER" refer to three distinct catalogue entries each for the file named CUSTOMER. You should note that multiple file entries should not be used in an ordinary data catalogue, since only the first one to appear will ever be referenced.

7.6 Adding a New Entry

If there is no entry already present for the file you have specified you are prompted for the unit and volume information to be added to the catalogue:

```
Unit:unit-information Volume:volume-information
```

The *unit-information* is normally supplied as a unit-id or address (e.g. 203). However, you may optionally specify a **floating address**, an integer between -99 and 99 inclusive. This integer will be added to the unit address of the volume occupied by the catalogue to determine the actual unit on which the catalogued file resides when the entry is processed by CATA\$ or the Data Security System.

The *volume-information* is normally supplied as a volume-id, although ? may be keyed if no volume-id checking is to take place when the file is opened. In addition, you may key an asterisk in the place of any of the characters that normally appear in the volume-id. Each asterisk is treated as a **wild card** and is replaced by the equivalent character from the volume-id of the volume actually occupied by the catalogue when it is processed by CATA\$ or the Data Security System. For example, if you key **MAST and the catalogue occupies volume SAPROG, then the volume-id supplied from the entry will be SAMAST.

In this example, CUSTOMER will occupy volume SAMAST on unit 203:

```
File-id: CUSTOMER Unit: 203 Volume: SAMAST
File-id:
```

In the second example, CUSTOMER is to occupy the same volume as the catalogue itself:

```
File-id: CUSTOMER Unit: 0 Volume: *****
File-id:
```

Note that a catalogue only contains space for a maximum of 100 entries. If you attempt to add a 101st entry the following error message is displayed and immediately followed by the catalogue entry prompt:

```
TOO MANY ENTRIES
```

7.7 Deleting an Existing Entry

When an entry already exists, the following prompt appears:

```
Key D to delete, <CR> to continue:
```

Key D to delete the entry from the catalogue. A new catalogue entry prompt to be redisplayed.

7.8 Amending an Existing Entry

If you key <CR> to the delete prompt the existing unit and volume information is redisplayed and you are given the opportunity to modify it. For example:

```
File-id: CUSTOMER Key D to delete, <CR> to continue: <CR>
Unit 204: 1 Volume SAMAST: SAMAS*
File-id:
```

shows an entry being totally modified: A floating unit address is to be used in place of the previously fixed address, and a volume-id with a wild card in the sixth character position will appear instead of SAMAST.

You may reply <CR> to either of the prompts that allow you to alter entry information to indicate that the current value is to remain unchanged.

7.9 The Catalogue Listing

When the catalogue is listed a page of catalogue entry information is displayed, followed by the catalogue selection prompt. An example is shown in Figure 7.2.

The entry information is displayed from left to right, top to bottom, in the sequence in which the entries appear in the catalogue. This sequence is not important for a data catalogue, but it is critical in the case of a security catalogue, where it determines the order in which the files are written to the backup cycle.

You can arrange the catalogue in alphabetical order of file name by using the \$SORT command to sort it according to the 8-byte ascending character key beginning at byte 1 of each record.

7.9.1 The Catalogue Prompt

Each page of the catalogue listing is followed by the catalogue prompt:

```
Key F to amend file, Accept, Print, <CR> for next, <ESC> to abandon:
```

Key <CR> to display the next page of the catalogue. If there is no subsequent page the following catalogue file entry prompt is redisplayed to allow you to continue creating or modifying entries:

```
File-id:
```

Key P to the catalogue prompt to print the catalogue to unit \$PR, the logical printer. If \$PR is assigned to a disk unit the print file will be named L.\$CATAL and if a file with the same name already exists on the volume it will be automatically deleted.

Key A to the catalogue prompt to confirm that the catalogue information is now satisfactory. The existing catalogue file, if there is one, will be deleted, and the catalogue information will be written to a new file. The message CATALOGUE WRITTEN appears and the catalogue file prompt is redisplayed so that you can either process a new catalogue or quit and obtain a ready prompt.

Key <ESC> to the catalogue listing prompt if you decide not to save the catalogue information to a new file. (You might key <ESC> if you had simply used \$CATAL to inspect an existing catalogue.) Any existing file remains undisturbed. The warning message CATALOGUE NOT MODIFIED appears and the catalogue file prompt is redisplayed so that you can either process a new catalogue or quit.

7.10 Example 1 - Catalogue Maintenance

SACAT is an existing catalogue on unit 203. The following dialogue takes place to delete an entry erroneously named CUSTOMR, add the correct entry for CUSTOMER, and suppress volume-id checking on the PRODUCT file.

```
GSM READY:$CATAL
;
; (Screen cleared)
;
CATALOGUE MAINTENANCE
CATALOGUE:SACAT UNIT:203
Old?:Y
(catalogue listing)
Key F to amend file, Accept, Print, <CR> for next, <ESC> to abandon:F
File-id:CUSTOMR Key D to delete, <CR> to continue:D
File-id:CUSTOMER Unit:203 Volume:SALESV
File-id:PRODUCT Key D to delete, <CR> to continue:<CR>
Unit 204:<CR> Volume:SALESV:?
File-id:<CR>
.....
..... (listing of first page of catalogue)
.....
Key F to amend file, Accept, Print, <CR> for next, <ESC> to abandon:<CR>
.....
..... (listing of second page of catalogue)
.....
Key F to amend file, Accept, Print, <CR> for next, <ESC> to abandon:A
CATALOGUE WRITTEN
CATALOGUE:<ESC>
GSM READY:
```

7.11 Example 2 - A Catalogue for Flexible Installation

Suppose that a simple application involves only four files, PRODUCT, STATEMNT, CUSTOMER and TRANSAC and the catalogue information held for them is as shown in the following table:

File-ID	Unit Information	Volume Information
PRODUCT	0	SMAST*
STATEMNT	0	SAMST*
CUSTOMER	1	SAMEX*
TRANSAC	1	SAMEX*

A Sample Catalogue

Then, if for a large fixed disk system the catalogue itself was located on volume SAMSTA on unit 206, PRODUCT and STATEMNT would occupy the same volume. CUSTOMER and TRANSAC would reside on SAMEXA on unit 207. If a second subsystem, with its catalogue on SAMSTB on unit 208, were to be set up, then its PRODUCT and STATEMNT files would occupy the same volume, and CUSTOMER and TRANSAC would be on SAMEXB on unit 209.

8. \$CONV - File Conversion

You may use the \$CONV command to create an empty indexed sequential (IS) or relative sequential (RS) file (see example 1 below) or to perform four different types of conversion involving RS and IS files:

- RS input producing IS output. Used initially to create an IS file as in Example 2 below.
- RS input producing RS output. Used to remove logically deleted records from an RS file.
- IS input producing IS output. Used to reorganise an IS file as in Example 3 below.
- IS input producing RS output. Used to extract a sorted RS file from an IS file.

Records beginning with an asterisk are considered to be logically deleted and are physically removed by the conversion process which simply fails to transfer them from the input to the output file. An RS input file which is used to produce an IS file must contain records starting with a two-character type code in bytes 1 and 2; a two-byte work field for use by Global System Manager in bytes 3 and 4; and the ascending character key to be used for record retrieval in bytes 5 onwards.

8.1 The Input File Prompt

When you run \$CONV it displays the input file prompt. You must key <CTRL A> if you wish to create an empty file, or supply the file-id and unit-id of the RS or IS file you wish to convert. (This file will be read-only as far as \$CONV is concerned.) For example, to create an empty file:

```
GSM READY:$CONV
$65 CONVERSION INPUT:<CTRL A>
$65 CONVERSION OUTPUT:
```

or to specify that SADATA on unit 204 is to be converted:

```
GSM READY:$CONV
$65 CONVERSION INPUT:SADATA UNIT:204
$65 CONVERSION OUTPUT:
```

8.2 To Quit

To exit from \$CONV key <ESC> in response to the input file prompt.

8.3 The Output File Prompt

To describe the output file to be produced by the conversion process you must supply the output file-id, unit-id, the size of the file, and the file type (R for RS, I for IS). For example:

```
$65 CONVERSION OUTPUT:SAINDEX UNIT:205 SIZE:48000 TYPE:I
```

The **size** can be specified in bytes, Kbytes or Mbytes (e.g. 100000, 100K or 0.1M). If a size of zero is specified the maximum contiguous free space available will be allocated.

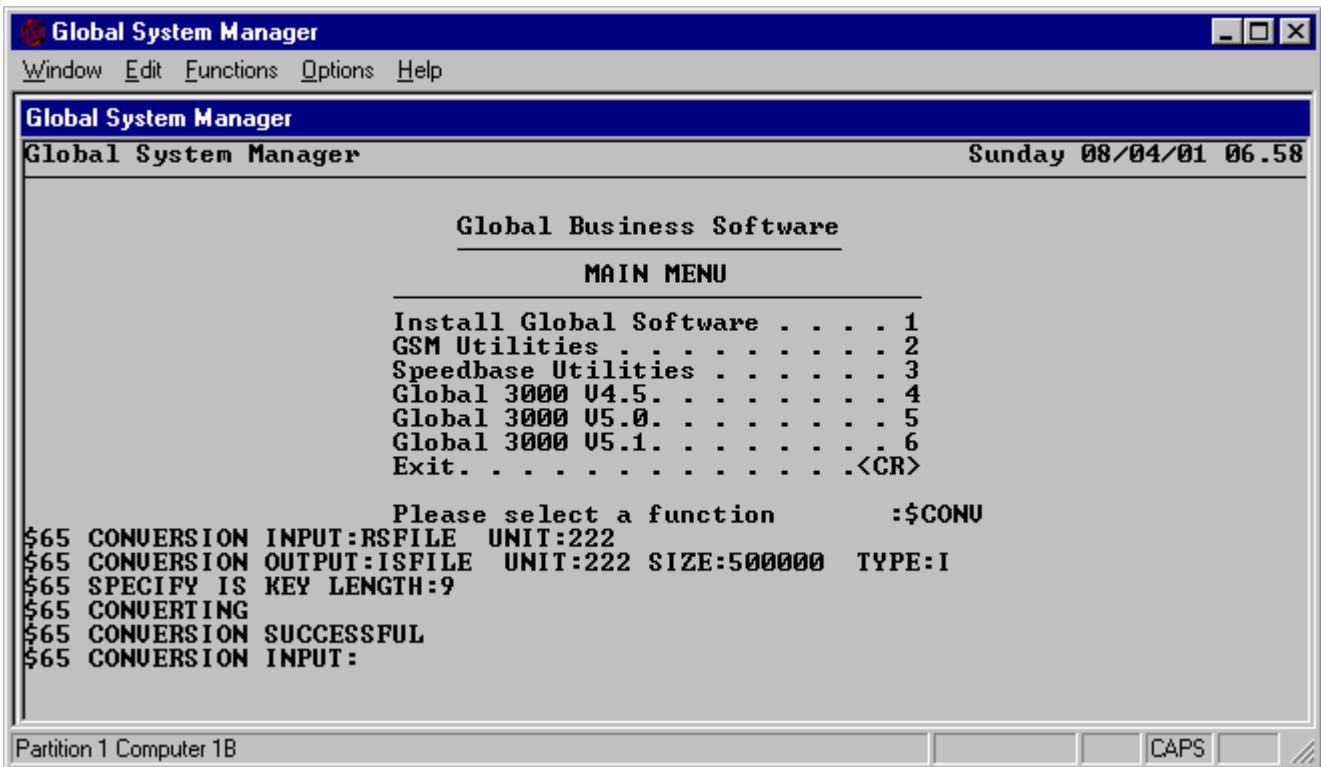


Figure 8.1 - Converting an RS file into an IS file

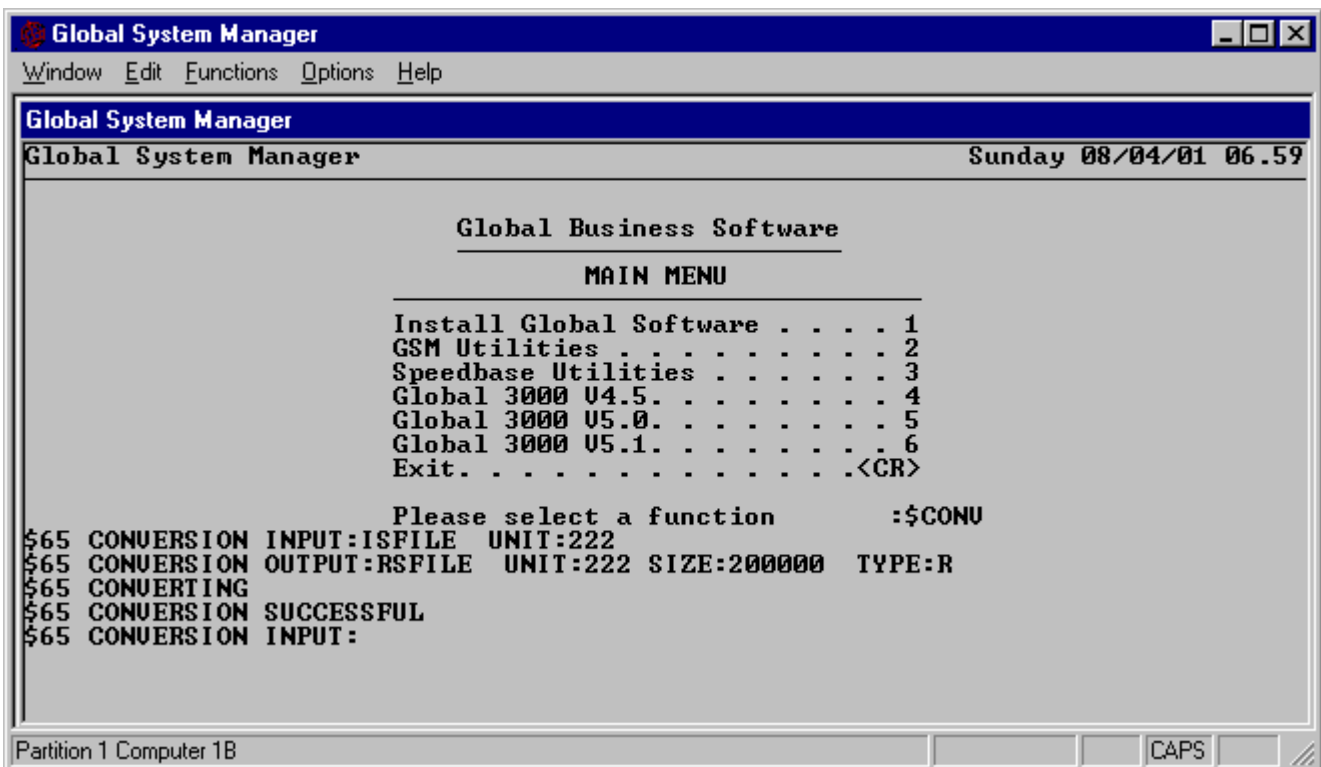


Figure 8.2 - Converting an IS file into an RS file

You may key <CR> instead of a particular value for any of the input quantities. A reply of <CR> to a file-id or unit-id prompt causes the file-id or unit-id you supplied to the input file name prompt to be used by default. A reply of <CR> to the size prompt means that the output file is created with the same size as the input file. (Note that if you are creating an IS file from an RS

file and the input file extent is full or nearly full, a reply of <CR> to the size prompt is unlikely to provide an output file of sufficient size.) A reply of <CR> to the type prompt will cause the output file to be produced with the same organization as the input file.

You should note that it is **not** permissible for the output file to overwrite the input file. The following error:

```
* IN USE ERROR ON drive description - nnn
* RETRY?:
```

will occur if you mistakenly specify both files with the same name and on the same unit.

8.4 The Key and Record Length Prompts

When you are creating a new file or an IS file from an RS file you are prompted for the key length of the new file in bytes. You must supply the correct value, between 1 and 99 inclusive. For example:

```
$65 SPECIFY IS KEY LENGTH:8
```

If you are creating a new empty file, rather than converting an existing one, you are asked for its record length in bytes. For example:

```
$65 SPECIFY RECORD LENGTH:100
```

8.5 The Conversion Process

Once you have defined the file to be created the conversion process begins. The message:

```
$65 CONVERTING
```

is output and then, if all goes well, a short time later:

```
$65 CONVERSION SUCCESSFUL
```

appears and a new input file prompt is displayed, so that you can convert another file or exit from the program.

If you are creating an IS file but the input file is not in ascending order of the key you have specified, or if the file contains records with duplicate keys, an error message of the form:

```
$65 INPUT SEQUENCE ERROR [- REC number]
```

appears. If the input file is an RS file the record number of the first out-of-sequence record detected is also displayed.

8.6 Example 1 - Creating an Empty Indexed Sequential File

For program testing purposes you wish to create an empty IS file, SAINDEX. You will add data by inserting records under program control using the WRITE statement. The records are each 100 bytes long. The first two bytes of each record are for the type code, bytes 3 and 4 are reserved for Global System Manager, and bytes 5-12 contain the key, the 8-byte customer code. The following dialogue sets up a file capable of holding exactly 100 records (there is a

512-byte overhead for the dummy high record followed by the single index block, which is 256 bytes long and starts on a 256-byte boundary):

```
GSM READY:$CONV
$65 CONVERSION INPUT:<CTRL A>
$65 CONVERSION OUTPUT:SAINDEX UNIT:205 SIZE:10512 TYPE:I
$65 SPECIFY IS KEY LENGTH:8
$65 SPECIFY RECORD LENGTH:100
$65 CONVERTING
$65 CONVERSION SUCCESSFUL
$65 CONVERSION INPUT:<ESC>
GSM READY:
```

8.7 Example 2 - Creating an IS File from an RS File

Later you wish to set up a genuine SAINDEX file from an RS file named SASORTED which has been arranged in ascending customer code order (possibly by using \$SORT). SASORTED and the new SAINDEX both occupy unit 204. You have used the \$CALC command and determined that the maximum file space required, after allowing for expansion, is 200,000 bytes:

```
GSM READY:$CONV
$65 CONVERSION INPUT:SASORTED UNIT:204
$65 CONVERSION OUTPUT:SAINDEX UNIT:<CR> SIZE:200000 TYPE:I
$65 SPECIFY IS KEY LENGTH:8
$65 CONVERTING
$65 CONVERSION SUCCESSFUL
$65 CONVERSION INPUT:<ESC>
GSM READY:
```

8.8 Example 3 - Reorganising an Indexed Sequential File

Some time later you need to reorganize the SAINDEX file. You decide to produce a new file with the same name and size, but on a different volume, as follows:

```
GSM READY:$CONV
$65 CONVERSION INPUT:SAINDEX UNIT:204
$65 CONVERSION OUTPUT:<CR> UNIT:205 SIZE:<CR> TYPE:<CR>
$65 CONVERTING
$65 CONVERSION SUCCESSFUL
$65 CONVERSION INPUT:<ESC>
GSM READY:
```

9. \$CUS - System Customisation

The functionality of the \$CUS utility is dependent on the host operating system. \$CUS is fully documented in Chapter 6 of the Global System Manager Manual. The features of \$CUS that are dependent on the host operating system are documented in Chapter 4 of the Global Operating Manual.

10. \$D - Altering the Date and Time

Sometimes you may wish to run programs for a date other than that established when you loaded Global System Manager. Alternatively, it may be necessary to change both the date and time if midnight passes. You should note that Global System Manager does **not** automatically update the date and time if processing continues after midnight, and instead extends the concept of the 24 hour clock to refer to the next day's times. So, for example, if you sign on at noon on Nov 23 1990, then 11.25 p.m. that evening will be 23.25, and if you continue working beyond midnight, 1.35 a.m. of the next day will appear as 25.35.

Note that Global System Manager menus can be customised to advance the date automatically if the time is after midnight (see Chapter 7 of the Global System Manager Manual).

To change the date, time, or both, run \$D from the menu or ready prompt. \$D first asks for a new date, which you key if you want to update the date, or T to update the time. If you key T you will be prompted for a new time. Note that you can key D to the initial prompt, which will cause \$D to prompt for a new date.

Keying <CR> to either the date or the time prompt leaves the date or time unaltered. You may key a full-stop or comma rather than "/" to separate the elements of the date if this is more convenient.

If the time is past midnight when you run \$D then the following prompt is displayed:

```
IT IS AFTER MIDNIGHT - DO YOU WISH THE DATE ADVANCED?:
```

You should key "Y" to advance the date by one day and set the time back 24 hours, or "N" to leave the situation unchanged.

Important Note: Changes made using \$D only affect a single partition.

The date and time are always reset automatically by reloading Global System Manager. The date and time **may** be reset automatically if you sign off and then on again using \$E (or \$END). The automatic date/time reset depends on the partition in which \$D was executed and the Global System Manager configuration according to the following table:

Host o/s	Partition	Sign off using	Date/time reset?
BOS	1	\$E or \$END	No
BOS	2, 3, 4 etc.	\$E or \$END	Yes
MS-DOS	1	\$E or \$END	No
MS-DOS	2, 3, 4 etc.	\$E or \$END	Yes
Novell	1	\$E or \$END	No
Novell	2, 3, 4 etc.	\$E or \$END	Yes
Unix	1	\$E	Yes
Unix	1	\$END	No
Unix	2, 3, 4 etc.	\$E or \$END	Yes

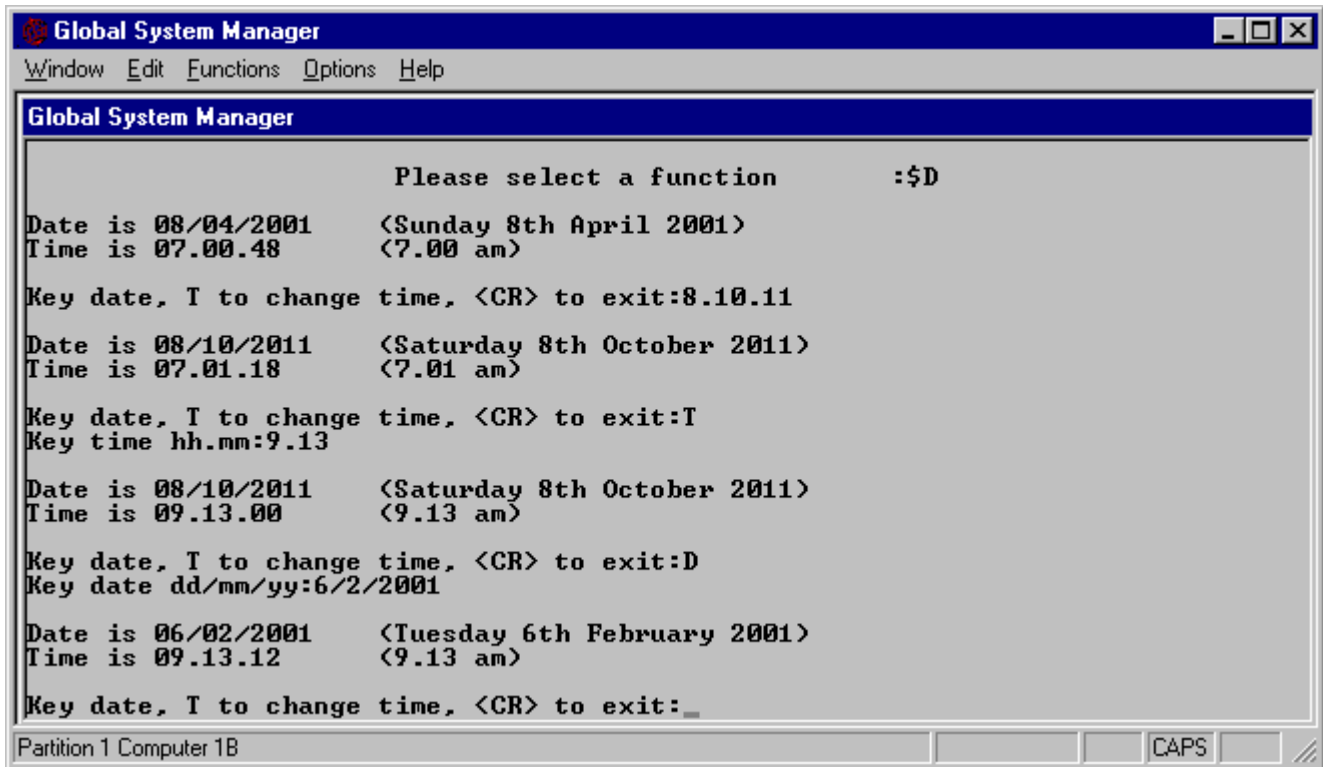


Figure 10.1 - Altering the time and date

11. \$DIR, \$DIRP, \$DIRCUS - Directory Utilities

The directory utility, \$DIR, complements \$F, the file maintenance utility (see Chapter 15), and provides facilities to display directories and domains that contain large numbers of files. In addition \$DIR can be used to copy, move, delete and rename groups of files which can be selected using a wide range of criteria. For example you might use \$DIR to do the following:

- Copy all files with creation dates that fall within a given range;
- Delete all files with names that match a particular pattern.

\$DIR CANNOT BE USED UNDER JOB MANAGEMENT. If you wish to run a file management utility under Job Management you must use \$F.

The \$DIRP utility (see section 11.10) provides a superset of the functions available with \$DIR. In addition to the display, copy, move, delete and rename operations of \$DIR, \$DIRP includes extra function pop-up menus that allow you to process the **single** file currently under the cursor. \$DIRCUS (see section 11.11) allows you to customise a user-specific menu for subsequent use by the \$DIRP utility.

11.1 The Unit Prompt

\$DIR first prompts for the address or unit-id of the device whose directory you wish to display. The device may be a diskette, hard disk domain, subvolume, or RAM disk. For example:

```
GSM READY:$DIR
.....
..... (Initial $DIR screen)
.....
Unit:201
.....
..... (Directory of 201 displayed on screen)
.....
```

When \$DIR has been newly run from the GSM READY: prompt or a menu a default value of \$DI is offered for the unit-id. If you use one particular device or subvolume a great deal you may wish to use \$CUS or menu maintenance to assign \$DI to that device so that the directory of that device can be displayed simply by keying <CR>.

11.2 To Quit

To exit \$DIR you must key <ESC> in response to the "Unit" prompt. If you key <ESC> in response to other prompts \$DIR does not exit to the GSM READY: prompt but to a higher level in its dialogue (the unit prompt is the highest level). Thus to exit from any level of \$DIR key the requisite number of <ESC>'s.

Sensitive operations are always abandoned if <ESC> is keyed. For example, if a file is about to be overwritten \$DIR will display the following prompt with a default response of N:

```
filename already exists, delete?
```

If you key Y<ESC> *filename* is NOT overwritten.

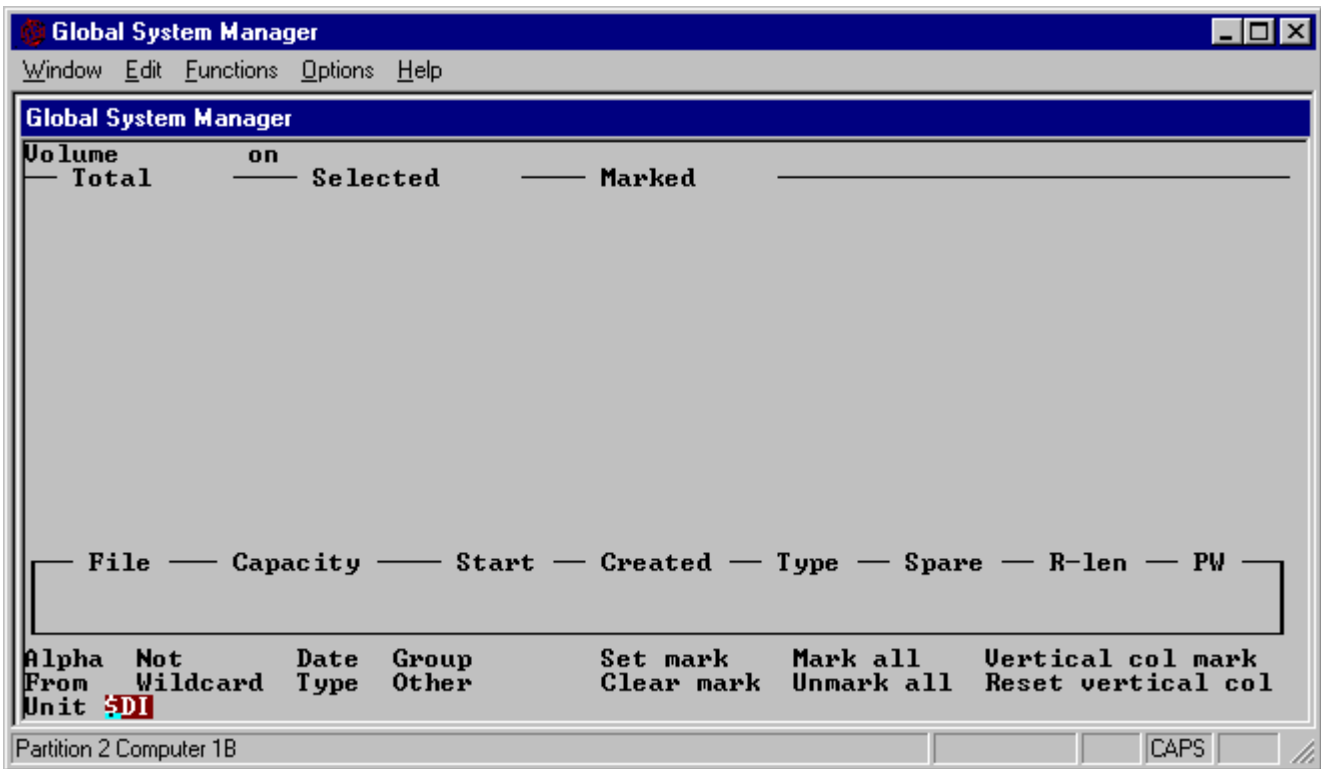


Figure 11.1 - Initial \$DIR screen

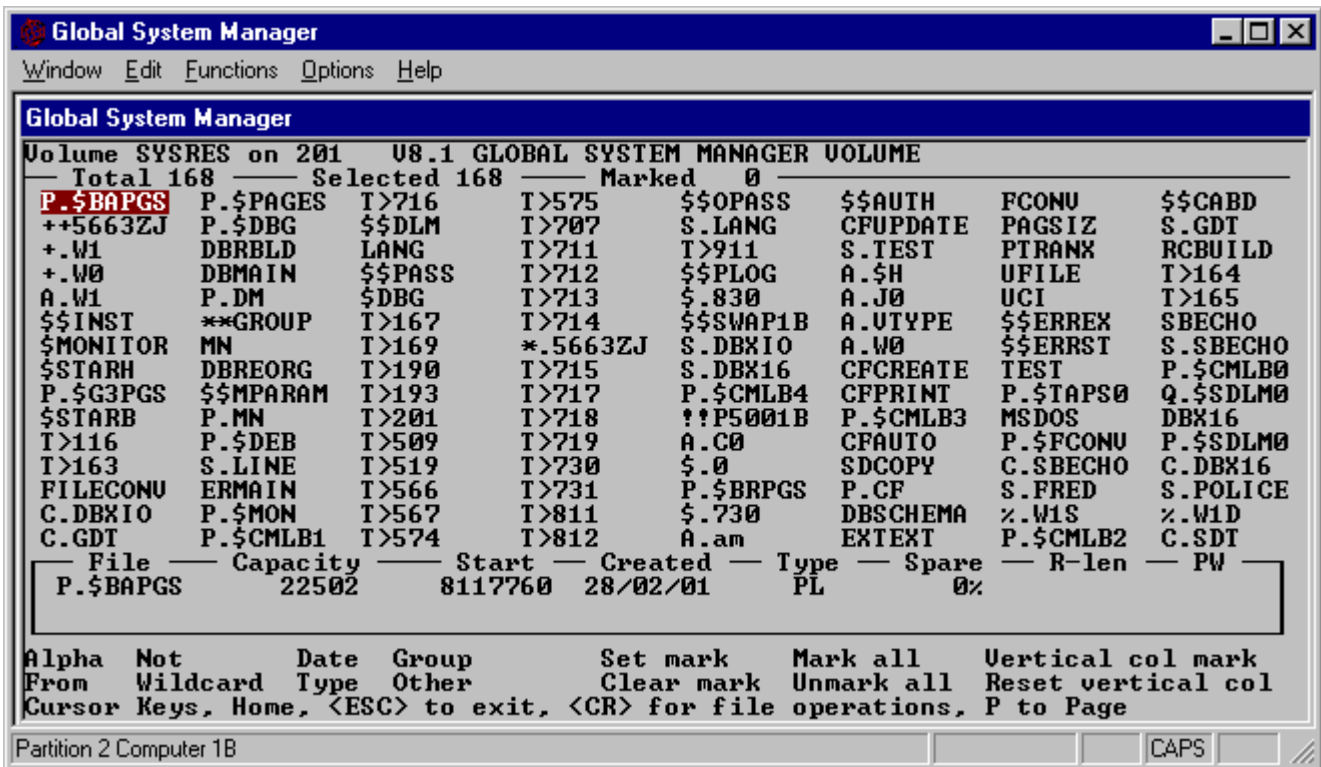


Figure 11.2 - File Selection Screen

Instruction	Group and Function	Section
Selection Instructions		
D	Select files by date	11.6.1
F	Start selection from a given file	11.6.2
N	Select files not on the current list	11.6.4
OP	Select password-protected files	11.6.5
T	Select files by file-type	11.6.3
W	Select files using wildcards	11.6.6
Z	Change wildcard meta-characters	11.6.7
File Marking Instructions		
C	Clear a single marked file	11.7.1
M	Mark all files in the selected list	11.7.2
R	Unmark all files in a vertical column	11.7.3
S	Set a mark on a single file	11.7.1
U	Unmark all files in the selected list	11.7.2
V	Mark all files in a vertical column	11.7.3
File Sorting Instructions		
A	Sort files in alphabetical order	11.8.1
G	Group all files with the same suffix	11.8.3
OC	Sort files on capacity	11.8.6
OD	Sort files by date	11.8.6
OS	Sort files by start-address	11.8.6
OT	Sort files by file-type	11.8.6
X	Sort files in alphabetical order from character	11.8.2
Miscellaneous Instructions		
I	Display subvolume and diskette information	11.8.4
K	Display file sizes and start addresses	11.8.5
ON	Display actual names of files on spool unit	11.8.7
P	Display next page of directory listing	11.4

Table 11.1 - \$DIR Instructions

11.3 The Directory Display

The directory display is divided into 4 areas:

- Volume information line;
- File name display area;
- File information box;
- Help text.

The top line of the screen is used to display the following information: The volume name of the device you have chosen; the physical address of the device (as opposed to the logical unit-id you may have used in response to the "Unit" prompt); the Long Volume Description associated with the selected volume; the number of files in the directory (or subvolumes in the domain

directory); an indication of the number of files you have selected or marked (see section 11.5 for more information about selecting and marking files). Initially all files in the directory are selected and none are marked.

The main body of the screen is taken up with a list of the files (or subvolumes) on the device. The list is arranged in columns with the first file displayed in the top left, and the second file immediately below it. If there are more than 120 files in a directory then you can key P to display the second page.

\$DIR positions a cursor over the first file in the list. On monochrome screens the cursor takes the form of an ">" character to the left of the file name. On colour screens the cursor is a coloured block which highlights the file name.

The file information box near the bottom of the screen is used to display details of the file (or sub-volume) currently under the cursor. For example, when a domain is selected, the information box includes the capacity, start-address and long volume description of the currently selected sub-volume. The cursor keys can be used to move the cursor to different files. As the cursor moves, the information within the box changes to reflect the file under the cursor.

Help text is displayed on the bottom 4 lines of the screen and changes to reflect the available commands and options. Note in particular that only a restricted set of commands is available when displaying a domain directory.

11.4 Cursor keys, Page, and <CR>

The four cursor positioning keys and the "home" key can be used to move the cursor around the screen. As the cursor moves the details in the file information box change to reflect the file under the cursor. You will also need to use the cursor positioning keys if you wish to mark files.

If the volume you are displaying contains more than 120 files then the display covers two pages. You can toggle between the two pages by keying P, by using the Global System Manager "page" command key (use \$T to find out which key this is), or by using cursor up or down when the cursor is positioned over the first or last file on the screen.

<CR> has two different uses when used at the file selection screen. If you are displaying a domain directory then <CR> causes a display of the subvolume under the cursor to replace the domain display. Keying <ESC> at the subvolume file selection screen returns you to the domain display, not the "Unit" prompt.

If you are displaying a subvolume directory then keying <CR> takes you to the "file operations" baseline prompt from where you can copy, move, delete, or rename files.

11.5 Selecting and Marking Files

\$DIR provides two methods for choosing the files that are operated on by the copy, move, delete, and rename commands. The two methods are **selecting** files and **marking** files.

The selection commands affect the list of files displayed on the screen. Once you have selected a group of files and displayed them on the screen you can immediately go on to operate on them. If, however, you wish to further refine your choice you can mark files and then operate on the marked subset only.

Marked files are indicated by an "*" to the left of the file name on a monochrome screen, and by a coloured block on a colour screen. On some monochrome screens, the name of the marked file may be flashing.

11.6 Selection Commands

11.6.1 Selection by Start Date (D)

This command selects those files from the displayed list whose creation date lies in a given range. Having keyed D at the file selection screen (see figure 11.2) the following baseline prompt is displayed:

```
From: start-date To: end-date
```

\$DIR offers a default of the current system date (today's date) for both the *start-date* and *end-date*. This allows you to easily select those files that have been created today. If you key an invalid date the "From" and "To" prompts are redisplayed. If no files fall within the range you specify, or if the *end-date* precedes the *start-date*, the message:

```
No entries selected - hit any key to continue
```

is displayed, and you will then be returned to the "Unit" prompt or the domain directory display.

11.6.2 Start Selection From Given File (F)

This command prompts for a file name and then selects and displays that file and all the ones that follow it from the original list. Having keyed F at the file selection screen (see figure 11.2) the following baseline prompt is displayed:

```
From: start-file
```

\$DIR supplies the file name under the cursor as a default. If the file name you specify is not in the initial list then \$DIR sounds the bell and aborts the command.

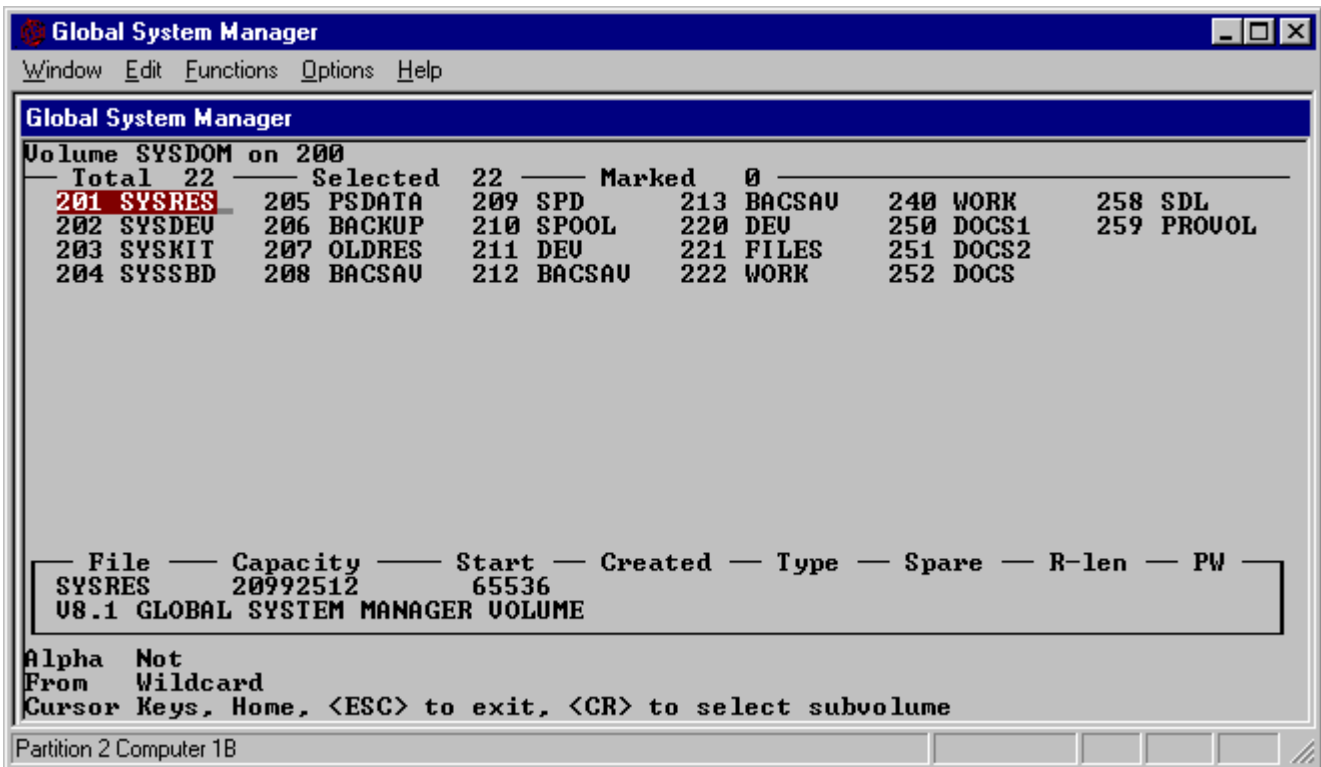


Figure 11.3 - Subvolume selection screen

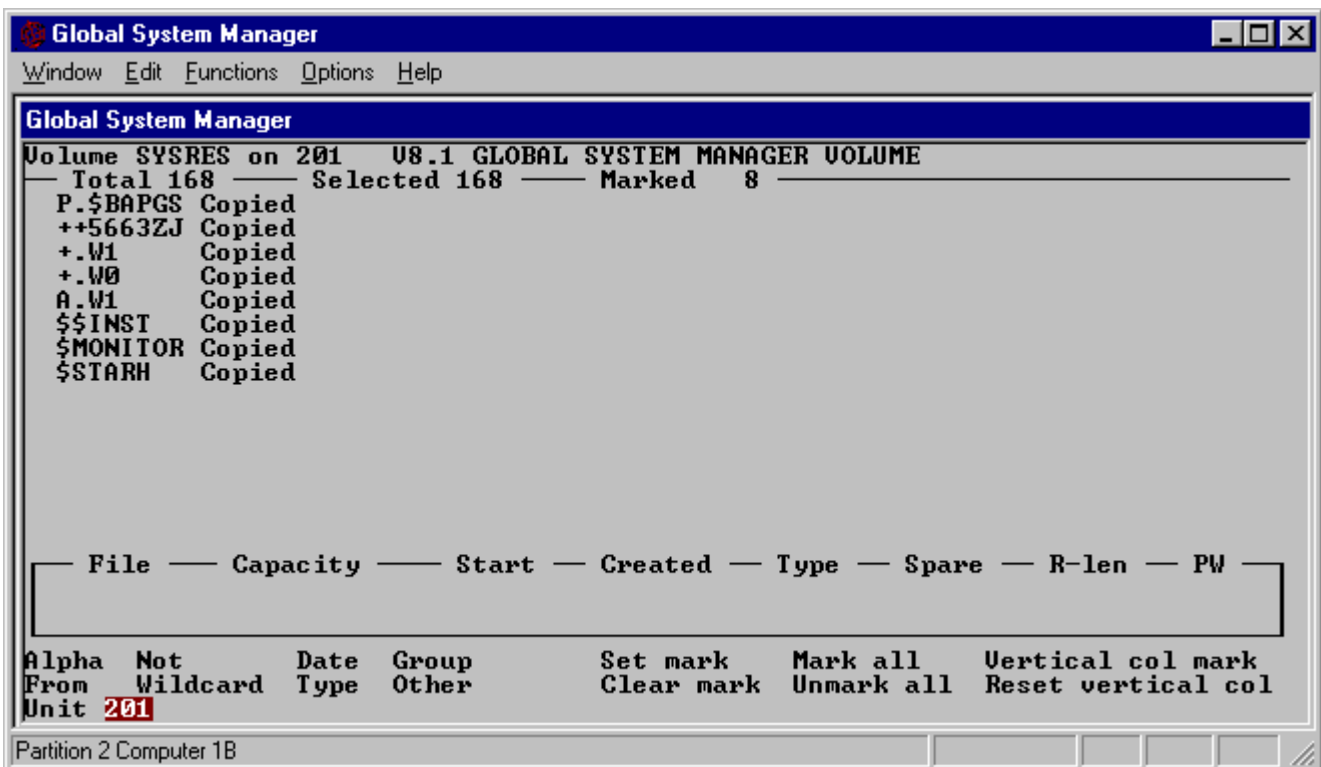


Figure 11.4 - Screen following copy operation

11.6.3 Selection by File Type (T)

This command selects those files from the displayed list with a given file type (see table 15.2). Having keyed T at the file selection screen (see figure 11.2) the following baseline prompt is displayed:

File type:type

\$DIR offers a default of TF (text file) for the file type. If you specify an invalid file type the "File type" prompt is redisplayed. If none of the files in the initial list are of the correct type the message:

No entries selected - hit any key to continue

is displayed, and you will then be returned to the "Unit" prompt or the domain directory display.

11.6.4 Complementary Selection - Not (N)

This command selects those files from the volume which are not in the currently displayed list. To use the **Not** command simply key N at the file selection screen (see figure 11.2). If the current selection consists of all the files in a volume then the complementary selection will be empty therefore keying N will result in the following message:

No entries selected - hit any key to continue

being displayed, and you will then be returned to the "Unit" prompt or the domain directory display.

For example, assume a volume contains 4 files: S.FRED, S.DAVE, FRED and DAVE - where two of the files are text files (type TF) and two are program files (type PL). You might first use the T command to select the text files - S.FRED and S.DAVE would be displayed on the screen. If you then go on to use the N command DAVE and FRED will be selected in place of the text files. Using the N command again will cause the text files to be reselected.

The N command works slightly differently if some of the displayed files are marked. The command selects and displays those files from the volume which are not marked. To elaborate on the example above: Using the N command after selecting S.FRED and S.DAVE and then marking S.DAVE, will cause S.FRED, FRED and DAVE to become selected.

11.6.5 Selecting Password Protected Files (OP)

This command allows you to select all of the files which are password protected. To use the password command first key O to reach the subsidiary "other" baseline prompt then key P to select the password command. \$DIR will immediately display all those files which are password protected. If no such files exist then an error message, explaining that no files are selected, is displayed and you will be returned to the "Unit" prompt or the domain directory display. See section 15.6.2 for full details of password protection.

11.6.6 Selecting with Wildcards (W)

This command allows you to select files by matching their names with a pattern. Two wildcard characters allow you to specify groups of characters that do not match explicitly. The default wildcard characters are "*" and "?", but these default meta-characters may be changed using the Z command (see section 11.6.7). To use the wildcard command key W at the file selection screen. \$DIR will then display the following prompt on the baseline:

File Specification:wild-card specification

with the current * wildcard character supplied as the default. The file specification can be up to 8 characters in length and can include any ASCII character in the range hex 20 - 7F. Any number of wildcard characters of either type can be used.

The "*" wildcard will match any number of characters, from 0 to 8 in number. The "?" wildcard matches exactly one character. For example:

- S.* matches all file names starting with S. and followed by any other characters. For example: S., S.A, S.XXSBPW
- * matches any file name
- *5518* matches any file name that somewhere contains the string 5518. For example: ++5518AC, #.5518
- A*C matches all names that start with A and end with C. For example: AC, ABC, ABDEFGHC
- ??? matches all 3-character file names
- ??* matches all file names that contain 2 characters or more. For example: AB, ABC but not A.

Once you have keyed the file specification followed by <CR> \$DIR displays the list of matching files. If none of the files in the initial list match the file specification the message:

```
No entries selected - hit any key to continue
```

is displayed, and you will then be returned to the "Unit" prompt or the domain directory display.

11.6.7 Changing Wildcard Selection Characters

The Z command allows you to change the characters used for multiple-character and single-character substitution (* and ? by default). It may be necessary to change the wildcard characters if the default wildcards occur in the names of files you want to select using the wildcard (W) command.

To change the wildcard characters, key Z to the file selection screen. You will then be prompted for the new characters as follows:

```
Multi-character substitution indicator: character
```

```
Single character substitution indicator: character
```

You should reply with the new substitution character in each case or with <CR> to use the old character.

11.7 Marking Commands

The commands in this section allow to mark or unmark a single file or a number of files.

11.7.1 Set and Clear Mark (S & C)

The S and C commands set and clear a single file mark. Having set or cleared the mark the cursor is advanced to the next file.

11.7.2 Mark and Unmark All (M & U)

The M and U commands mark or unmark all of the currently selected and displayed files. The cursor is positioned over the first file in the list.

11.7.3 Set and Reset Vertical Column Mark (V & R)

The V and R commands mark and unmark all of the files in a vertical column. The cursor is moved one column to the right.

11.8 Sorting Commands and Miscellaneous Commands

11.8.1 Sort into Alphabetical Order (A)

The A command redisplay the currently selected list of files in alphabetical order. Any marks in force before the sort are preserved. To use the command simply key A at the file selection screen.

11.8.2 Sort into Alphabetical Order by Character Position (X)

The X command operates in a similar way to the A command (see section 11.8.1, above) except that you can start the alphabetic ordering from any character position. For example, if you choose the fourth character as the start position, a file list of S.ABC, C.JFK, L.BBC and C.ABC will be ordered as C.ABC, L.BBC, S.ABC, C.JFK. Any marks in force before the sort are preserved. To use the command, key X at the file selection prompt. You will then be asked for the start position of the sort and must reply with a number from 1 to 8 or <ESC> to abandon the command.

11.8.3 Group Files by Suffix (G)

The G command redisplay the currently selected list of files so that files with the same suffix are displayed together. For example, if the current display contains the following list:

```
S . DAVE      B . DAVE      M . FRED
C . FRED      DAVE          S . FRED
FRED         L . FRED
```

then after the G command the screen would look like this:

```
DAVE         FRED          M . FRED
B . DAVE     C . FRED      S . FRED
S . DAVE     L . FRED
```

Any marks in force before the sort are preserved. To use the command simply key G at the file selection screen.

11.8.4 Subvolume and Diskette Information (I)

Key I at the file selection screen to obtain information about the currently displayed subvolume or diskette. No information is displayed for domains.

The **totalled file size** field is simply the sum of the file sizes of the files displayed on the screen.

The **rounded file total** is the sum of the file sizes, but with the size of each file first rounded up to an integral number of sectors. This total gives a better indication of the size of volume needed to contain the files since Global System Manager always reserves an integral number of sectors for a file.

The **largest free area** field gives the size of the largest contiguous area on the volume. The **total spare space** field gives the total free area on the volume (which may be fragmented into a number of separate areas).

The **volume size** field gives the size of the data part of the volume (i.e. the part that can be occupied by files).

11.8.5 Display Sizes and Start Addresses in Kb (K)

\$DIR displays the sizes and start addresses of files and subvolumes in the file information box near the bottom of the screen. If you key K at the file information screen the size and start address will be displayed in kilobytes (or megabytes if the file is large enough).

The K command acts as a toggle, keying K a second time causes the display to revert to the normal form of expression in bytes.

11.8.6 Other Sorting Commands (OC, OD, OS & OT)

The files in the selected and displayed list can be sorted using keys other than the file name itself. You can sort on capacity (OC), date (OD), start address (OS), and file type (OT). Any marks in force before the sort are preserved.

To use these commands first key O to reach the subsidiary "other" baseline prompt and then key C, D, S, or T to select the command you want. When the "sort by file type" option is selected, the files are sorted in order of the value of the ULORG field in the directory label (i.e. not by alphabetic order of the 2 character file type displayed by \$DIR) so that, for example, RS files (ULORG = 0) appear before IS files (ULORG = 1).

11.8.7 Names of files on a Spool unit (ON)

The ON instruction can only be used if the selected volume is a "spool unit" (see section 15.14). It displays the current spool file list with the name of the print or report file instead of the spool file sequence number and name.

To use this command you must first key O to reach the subsidiary "other" baseline prompt and then key N to select the command.

11.9 Operating on Files

Once you have selected or marked a set of files you can then go on to copy, move, delete or rename them. To reach the "file operations" prompt you must key <CR> at the file selection screen. The following prompt is then displayed on the baseline:

Key Copy, Delete, Move, Rename, <ESC> to exit:

Key C, D, M, or R to copy, delete, move, or rename your files as required.

\$DIR reports on the files being manipulated in the main portion of the screen. The list of files and the status of the operation (e.g. "Copied") are displayed in two columns, with the first

column being re-used when the second is full. Once all of the files have been operated on \$DIR returns to the "Unit" prompt.

To interrupt \$DIR while it is operating on files key <CTRL G>. \$DIR will complete the operation in progress and then display the following prompt:

```
Do you wish to abort?:
```

Key N to continue, or Y to return to the "Unit" prompt.

If an irrecoverable I/O error occurs while \$DIR is operating on a file all further operations are abandoned and \$DIR returns to the "Unit" prompt.

11.9.1 File Already Exists Prompt

When copying, moving or renaming, a file may exist on the output unit with the same name as the file being operated upon. When \$DIR attempts to copy, move or rename the identically named file it will display the following prompt:

```
file of date (size bytes) Already exists, delete?:
```

where *file* is the file name, *date* is the date of the file on the output unit and *size* is the file size in bytes of the file on the output unit. The default reply to this prompt is N. The prompt has five responses:

Y	causes the file on the input unit to replace the file on the output unit (the text " - overwritten" will appear to the right of the "Copied" or "Moved" text);
N or <CR>	cause the file on the output unit to be left in place. The input file is not copied;
<CTRL A>	abandons the operation in progress and \$DIR returns to the "Unit" prompt;
<CTRL B>	is equivalent to answering Y to this and all subsequent "File already exists, delete" prompts (if they occur);
<CTRL C>	is equivalent to answering N to this and all subsequent "File already exists, delete" prompts (if they occur).

11.9.2 Copying and Moving Files

The copy and move commands first prompt for the output unit to which the files are to be transferred. \$DO is offered as a default for the output unit. If you copy to one particular device or subvolume frequently you may wish to use \$CUS or the menu handler to assign \$DO to that unit.

You may wish to secure the original file on the input unit with a password (for example marking them as being 'old'). To achieve this, terminate the reply "C" (to copy) with <CTRL C> instead of the normal <CR>. You will then be asked for the password to be assigned to the input files as follows:

```
Key Copy, Delete, Move, Rename, <ESC> to exit:C<CTRL C>
```

Password:OLD

The files will then be copied and the input files will be secured with the password (see section 15.6.2 for a complete description on password protection).

The move command deletes the files from the input unit once they have been copied to the output unit.

11.9.3 Deleting Files

The selected or marked files are deleted. Note that unlike \$F, the \$DIR delete operation has no concept of an input or output unit. The files to be deleted are the ones displayed on the file selection screen.

11.9.4 Renaming Files

Note that unlike \$F, the \$DIR rename operation has no concept of an input or output unit. The files to be renamed are the ones displayed on the file selection screen. The rename operation allows wildcard renaming, it therefore prompts for an input and output file specification, both of which can contain wildcards.

11.9.4.1 Example 1

To Rename B. files (backup text files) as S. files use the following dialogue:

Rename:B.* as S.*

11.9.4.2 Example 2

To rename all files containing the string 4500 to equivalent names containing 9999 use the following dialogue:

Rename:*4500* as :*9999*

11.9.4.3 Example 3

To rename all files so that the first 2 characters of the file name appear at the end use the following dialogue:

Rename:??* as :*??

Note that:

- only those files whose names match the first file specification will be renamed, despite the fact that the original list (when you keyed <CR> to reach the file operations prompt) may have contained more files;
- The number of wildcard characters in the input and output file specifications must match;
- The wildcard characters in the input and output file specifications are matched left to right. Thus in example 3 the second character in a matching file name will be moved to the end of the new name, not the penultimate position;
- An internal restriction within \$DIR means that the "?" wildcard cannot immediately follow an "*" in the input file specification. For example, if you key "*?" you will be asked to re-key the specification. Note that '?'s can follow '*'s in the output specification, as shown in example 3 above;

- output file names are truncated to 8 characters.

THE RENAME FEATURE OF \$DIR SHOULD BE USED WITH CARE. It is possible with the rename operation and injudicious use of the <CTRL B> response discussed in section 11.9.1, to reduce a large directory full of files to just one file!

11.10 \$DIRP Directory Utility

The \$DIRP utility provides the same facilities as \$DIR, but extra function pop-up menus are available from the file selection screen. These menus allow you to process the file currently under the cursor in a variety of ways.

11.10.1 Standard Menus

If you key <F1> in the file selection screen you will be offered a standard menu of functions available for the file currently under the cursor. For example, if the file currently selected is the command library P.\$CMLB0, the menu displayed when <F1> is keyed will allow to dump the library file, to run library maintenance on the file, to run a selected member or to exit the menu. The <CURSOR UP> and <CURSOR DOWN> keys are used to select the menu entry. <CR> must be keyed to execute the entry.

The following menu entries are available:

11.10.1.1 Inspect File

This menu function is only available for text (type TF) and relative-sequential (type RS) files. It saves the current selection screen then chains to the \$INSPECT utility (see Chapter 19) which allows you to view the selected file. Terminating \$INSPECT will return to \$DIRP with the selection screen restored.

11.10.1.2 Dump File

This function is available for all file types. It saves the current selection screen then chains to the \$L utility (see Chapter 20) which allows you to dump the selected file to either the terminal or a printer. Terminating \$L will return to \$DIRP with the selection screen restored.

11.10.1.3 Print File

This function is only available for text (type TF) and relative-sequential (type RS) files. It saves the current selection screen then chains to the \$PRINT utility (see Chapter 28) which allows you to print the selected file to the logical printer unit \$PR. When the file has been printed \$PRINT returns to \$DIRP with the selection screen restored.

11.10.1.4 Library Maintenance

This function is only available for program libraries (type PL), compilation libraries (type CL) and data file libraries (type LB). It saves the current selection screen then chains to \$LIB (see Chapter 21) or \$DLIB (see Chapter 12) as appropriate. Terminating the relevant library maintenance utility will return to \$DIRP with the selection screen restored.

11.10.1.5 Run Program

This function is only available for program files (type PL) and will cause the program selected to be executed. Control will NOT be returned to \$DIRP when the program has completed but will be returned to either the current menu or the READY prompt.

11.10.1.6 Run Member

This function is only available for program libraries (type PL). The list of programs in the program library is displayed. You may choose the program you wish to run by moving the cursor to the correct program using the cursor keys, and then keying <CR>. Control will NOT be returned to \$DIRP when the program has completed but will be returned to either the current menu or the READY prompt.

11.10.1.7 Apply Zap

This function is only available for relative-sequential (type RS) files with a file-prefix of "Z.". It saves the current selection screen then chains to the \$ZAP utility (see Chapter 46) which allows you to apply the zap. When the zap has been applied \$ZAP returns to \$DIRP with the selection screen restored.

11.10.2 Selecting a customised menu

If you key <F2> within the \$DIRP file selection screen the customised menu as defined by \$DIRCUS (see section 11.11, below) will be displayed. You can apply any of the functions available from the menu to the file currently under the cursor by selecting the appropriate menu entry. Use the <CURSOR UP> and <CURSOR DOWN> keys to move to the required entry. Key <CR> to execute the entry selected from the pop-up menu. When the program selected by the <F2> function has completed control will return to \$DIRP with the full file details screen redisplayed.

Important notes: When the <F1> and <F2> pop-up menus are displayed, system requests and partition swapping are disabled.

When control returns to \$DIRP from an Inspect, Dump, Library, Apply Zap or Print entry from the <F1> pop-up menu, the selection screen that was displayed when the <F1> function was selected is restored. When control returns to \$DIRP from an entry on the customisable <F2> pop-up menu, all the files on the currently selected unit are displayed (i.e. the results of any previous \$DIRP selection commands are lost).

11.11 \$DIRCUS

The \$DIRCUS utility allows you define a customised menu for your operator-id for subsequent use by the \$DIRP <F2> function (see section 11.10.2, above).

On entering \$DIRCUS the details of the current menu (\$DR0000, where 0000 is your operator-id) will be displayed. If a customised menu file does not exist a new menu file with a single line (the "Exit" function), will be created. The following prompt will appear on the baseline:

Key Abandon, Delete, Insert, Update, <ESC> to exit:

11.11.1 Abandon

Key A to the baseline prompt to abandon the current menu update. If there was no existing menu file, a new one will not be created. An existing menu file will not be modified.

11.11.2 Delete

Key D to the baseline prompt to delete an existing menu entry. You will be prompted for the line number you wish to delete. You cannot delete the line for the "Exit" function. Key <ESC> to the "line number" prompt to return to the main baseline prompt.

11.11.3 Insert

Key I to the baseline prompt to insert a new menu entry. You will be prompted for the line number of the entry before which you wish to insert the new menu entry. No lines can be inserted after the "Exit" function. The maximum number of entries including the "Exit" function is 20. Key <ESC> to the "line number" prompt to return to the main baseline prompt.

When you have selected where the new line is to be inserted you will be asked to key a 20 character menu line description followed by the name of the command program you wish to execute for this menu line. You will then be prompted for the number of extra <CR>'s required. When running a menu entry, \$DIRP inserts the name and unit of the selected file into the type ahead buffer, followed by any extra <CR>'s that may be required to execute the command program. For example, to run \$INSPECT on the selected file no extra <CR>'s will be required but to run \$PRINT so that you can choose your own print unit you will require 1 extra <CR> (to run \$PRINT to print to the default listing unit (i.e. \$PR) will require 2 extra <CR>'s).

11.11.4 Update

Key U to the baseline prompt to update an existing menu entry. You will be prompted for the line number of the entry you wish to update or alternatively you can key A to update all the lines. Key <ESC> to the "line number" prompt to return to the main baseline prompt.

11.11.5 <ESC> to Exit

Key <ESC> to the baseline prompt to write the updates to your customised pop-up menu file to disk or cause a new menu for your operator-id to be created, if one did not already exist. The menu can now be used by the \$DIRP <F2> function (see section 11.10.2, above).

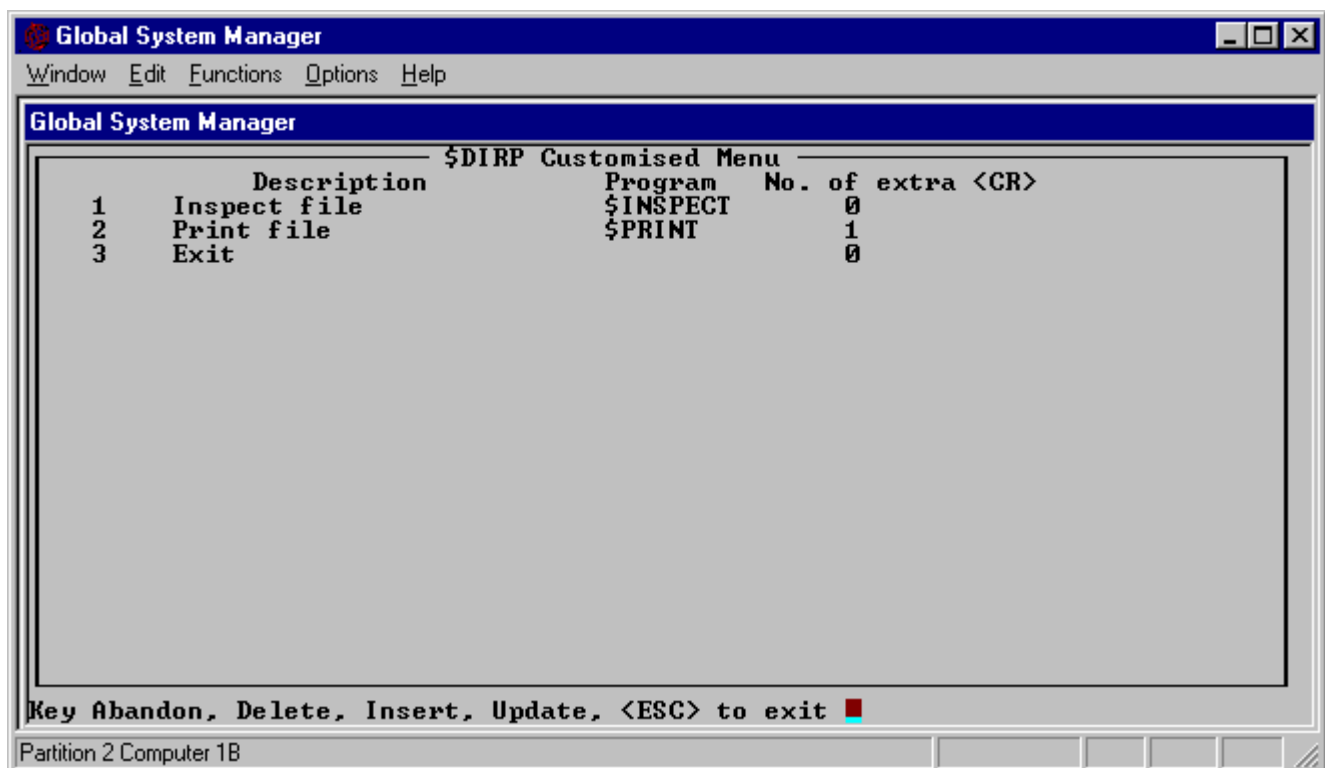


Figure 11.5 - \$DIRCUS Menu Maintenance Screen

12. \$DLIB - Data File Library Maintenance

The data file librarian, \$DLIB, is used to inspect, create or update data file libraries each of which may contain a maximum of up to 250 member files. These member files may be of any type (**except for program files**) including data libraries themselves.

Important note: *Data File Libraries* (file type LB) should not be confused with *Data Libraries* (file type DL) as documented in the Global Development File Management Manual.

\$DLIB is not used to maintain Global Cobol copy libraries. These are merely specially structured source files and are therefore updated with \$EDIT or \$ED.

Note also that \$DLIB is not used to maintain Program Libraries. These are created and amended using \$LIB, described in Chapter 21.

12.1 Introduction

A data file library is held as an individual file containing up to 250 *members*. The file-id of a data file library (sometimes referred to as the *library-id*) must begin with the prefix "E.". The members of the data file library can be of any file type excluding program files (type PL).

Each member in a data file library is uniquely identified by means of its *member-id*. The member-id is initially the file-id of the individual file which contained the member when it was first included in the library. However, the data file librarian's REN instruction can be used to rename members.

If you wish to create or amend a library, you must first of all establish it as the current *target library*. Then you can use the instructions listed in table 12.1 to include new members or update old ones. \$DLIB can only operate on one target library at a time, but the dialogue is structured so that it can process any number of targets, one after another, during a single run.

When you amend a data file library it is updated in place. Spare space must be reserved when the library is created to allow for new members or extensions to old ones. \$DLIB keeps track of any 'holes' in the library due to deletions and, when updating is complete, re-organises the library so that all the free space is collected together at the end of the file.

The remainder of this introduction explains the way in which the data file librarian dialogue is structured, and shows in outline how you inspect, create or amend a library. The introduction then describes how and when re-organisation takes place and the way in which errors are handled.

Command	Group and Function
Inspect and extract operations (section 12.10)	
LIS	List the index of a selected library *
PRI	Print the index of a selected library *
EXT	Extract a member from a selected library *
EXD	Extract a member from a selected library (followed by subsequent member deletion)
Including and comparing members operations (section 12.11)	
COP	Copy one or more selected members into target library

COM	Compare one or more selected members with originals
INC	Copy one or more selected data file libraries into the target library
Rename and delete operations (section 12.12)	
CHA	Change target library-id or title
REN	Rename or re-title target library member
DEL	Delete one or selected members from the target
Space saving operation (section 12.13)	
TRU	Truncate target library, leaving the specified amount of free space
Volume checking operation (section 12.14)	
I	Input volume check
Close target library operation (section 12.6)	
END	End target library processing

* These inspect and extract operations are the only instructions which do not require a target data file library to be established.

Table 12.1 - Data File Librarian Commands

12.2 General Dialogue Structure

When you run \$DLIB it begins by asking you to identify a target data file library. For example:

```
GSM READY:$DLIB
Data file library Maintenance
Target Library:E.SOURCE UNIT:203
```

If you only need to list or print a data file library index you need not specify a target. However, if you wish to create a new library, or amend an existing one, you must establish it as the target library by keying its file-id and unit-id (together with other information described later). In either case, eventually the data file library maintenance prompt will appear:

```
Data File Library Maintenance
:
```

You may then key any appropriate instruction from the list in table 12.1. The instruction you select will continue with its own dialogue, and eventually re-display the data file library maintenance prompt so that you can supply a further instruction, or terminate the process by keying END. When you do so the target library prompt will re-appear, so that you can operate on another library, or exit \$DLIB by keying <ESC>. Any dialogue with the librarian therefore assumes the following general form:

```
GSM READY:$DLIB
Data file library maintenance
Target Library:
(Identify first target library, if required)
Data File Library Maintenance
:first instruction
.....
..... (operate on first target library)
.....
Data File Library Maintenance
:END
Target Library:
(Identify next target library, if required)
Data File Library Maintenance
:first instruction
.....
```

```

..... (operate on next target library)
.....
Data File Library Maintenance
:END
Target Library:<ESC>
GSM READY:
    
```

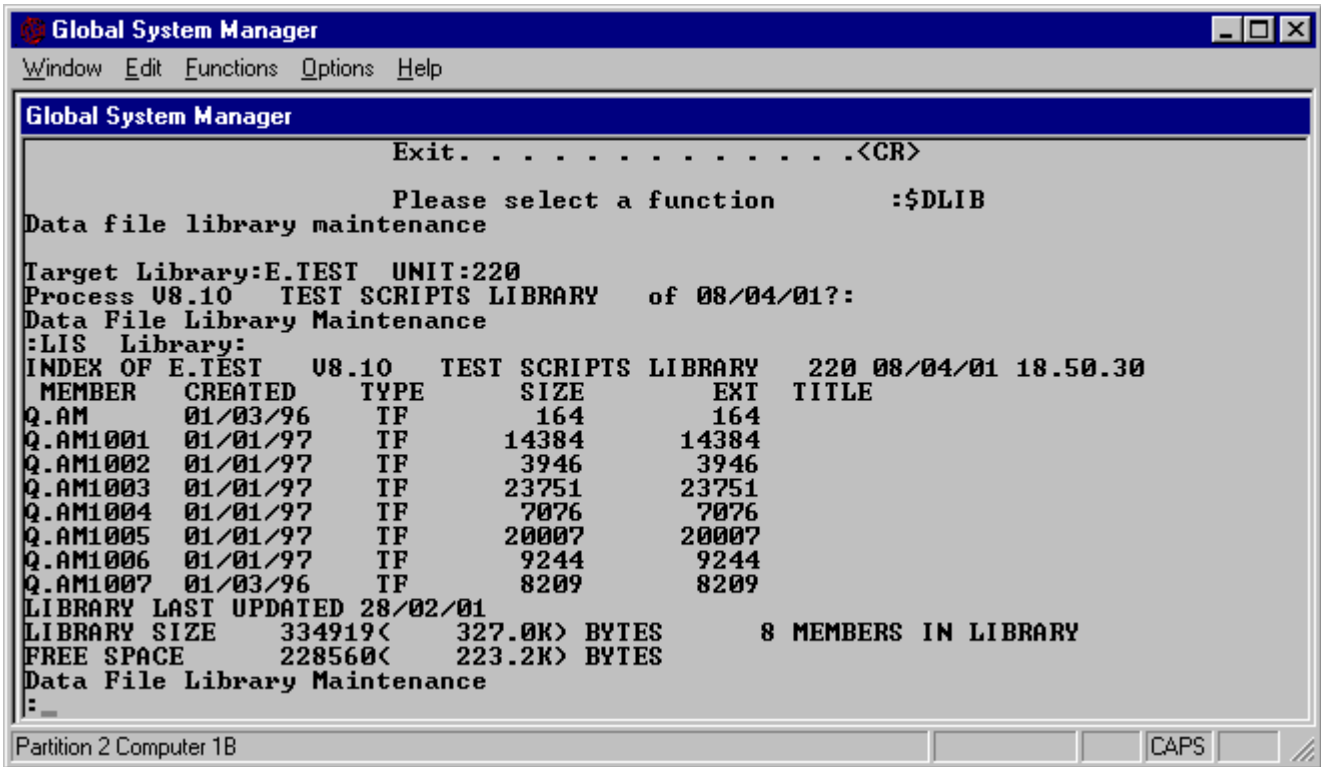


Figure 12.1 - Listing a library

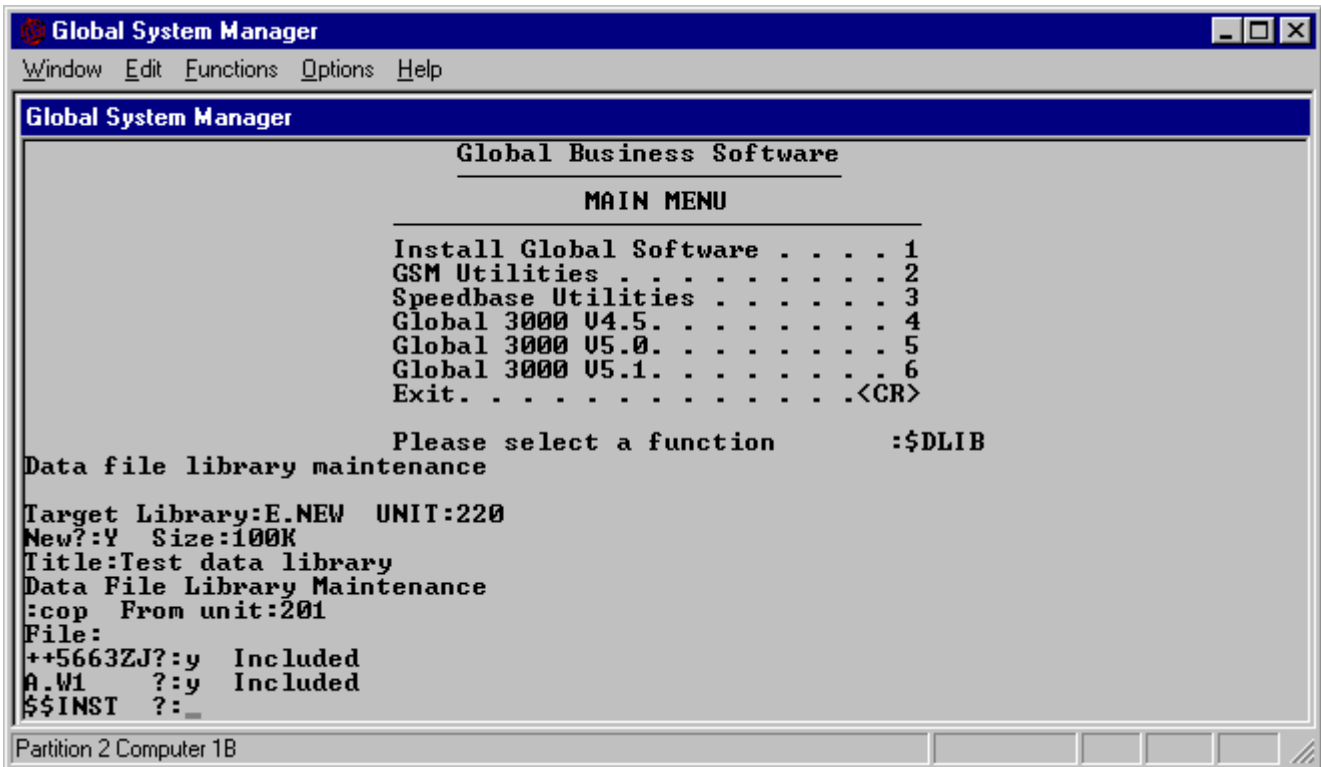


Figure 12.2 - Creating a new library and copying a file to it

Whenever you establish a target library, you must ensure it remains on-line until the next target library prompt appears following your keying of END. This is because the END processing is responsible for rewriting the updated data file library directory, and possibly compacting the members so that all the free space is collected together at the end of the file. If this is not done, the data file library will be invalid, and will be rejected when it subsequently comes to be used by the data file librarian.

12.3 Inspecting an Existing Library

If you only wish to inspect an existing data file library, or extract a member from it, there is no need to identify a target library, so you simply key <CR> to its prompt:

```
Target Library:<CR>
Data File Library Maintenance
:
(Only LIS, PRI and EXT instructions can be used)
Data File Library Maintenance
:END
Target Library:
```

If you attempt to use an instruction which requires a target library to be established your request will not be honoured. Instead, the following message will be displayed:

```
NO TARGET LIBRARY
```

followed by the data file library maintenance prompt, so that you can continue with an appropriate instruction.

12.4 Creating a New Library

To create a new data file library, you must key its library-id and unit-id to the target library prompt. Then you key Y to the subsequent confirmation prompt, and supply its size and title. For example:

```
Target Library:E.TEXT UNIT:205
New?:Y Size:100K
Title:Text Files
Data File Library Maintenance
:
(COP or INS instructions to include the initial members, and
any other instruction, such as TRU, where appropriate)
:END
Target Library:
```

If you key N, <CR> (or any single character apart from Y) to the New?: prompt, the creation operation will be abandoned, and the target library prompt will re-appear. This allows you to take corrective action if you really meant to update an existing library, but mis-keyed its name or unit.

In response to the Size: prompt you may key: A number of bytes; a number followed by the letter K, M or G, indicating that so many kilobytes, megabytes or gigabytes of storage are to be allocated; or simply 0, in which case the maximum amount of contiguous unused space will be made available. You do not have to be particularly precise in your size request since once the library is built you normally use the TRU instruction, described in section 12.13.1, to truncate the new data library and release any wanted storage for subsequent re-allocation. TRU allows you

to specify an amount of free space to remain in the library so that you can update it in place if you need to add or amend members later.

The title you specify can be up to 30 characters in length.

12.5 Amending an Existing Library

To amend an existing data file library you must key its file-id and unit-id to the target library prompt, and then key Y, <CR> (or any single character apart from N) to the subsequent process prompt to confirm that the library it identifies is indeed the one you require to modify. For example:

```
Target Library:E.TEXT Unit:205
Process TEXT FILES OF 25/11/94?:Y
Data File Library Maintenance
:
(INC or COP instructions to add or update members and any
other instruction, such as REN or DEL where appropriate)
Data File Library Maintenance
:END
Target Library:
```

If the data file library title or date of last modification, which appear as part of the process prompt, are not as expected, you should respond by keying N. A new target library prompt will then appear to allow you to re-specify the library you wish to update.

12.6 To quit

To quit and return control to the menu or GSM READY: prompt you must key END to the library maintenance prompt and <ESC> to the subsequent target library prompt:

```
Data File Library Maintenance
:END
Target Library:<ESC>
GSM READY:
```

When a target library has been created or amended you must perform the END processing to update its directory. In this case <ESC> is treated in exactly the same way as <CR>. If keyed by mistake to the library maintenance prompt itself the following warning message is displayed, followed by the library maintenance prompt itself:

```
KEY END TO REWRITE THE TARGET LIBRARY DIRECTORY
```

12.7 Library Re-organisation

If you delete one or more members from a target library, \$DLIB will re-organise it so that all the spare space is held as a contiguous area at the end of the file. For performance reasons re-organisation does not take place after every deletion, but only occurs when necessary: Before adding a member if the free storage is too fragmented to contain it; before processing instructions which cannot operate satisfactorily if the library contains "holes"; or as a part of END processing, so that the library is left in a fully compacted form. The following message is displayed:

```
Re-organising Library
```

and the normal dialogue is temporarily interrupted whilst the file is scanned and the members are moved.

12.8 Some Common Errors during Data Library Processing

You cannot process program files as part of data file libraries. If you attempt to do so the following warning message is displayed:

```
WRONG TYPE
```

The library maintenance prompt is re-displayed so that you can correct your error.

If you demount the target library by mistake, before you have keyed END, you will be requested to remount it by a prompt of the form:

```
REMOUNT volume-id ON unit-address FOR library-id:
```

where *volume-id* is the name of the volume occupied by the target library. For example:

```
REMOUNT SATEXT ON 204 FOR E.TEXT:
```

Mount the volume containing the library, then key <CR>, or any single character, to continue.

The following warning messages:

```
FILE NOT FOUND
MEMBER NOT FOUND
INSUFFICIENT SPACE ON TARGET LIBRARY
TARGET LIBRARY FULL
```

may appear at certain stages of the instruction dialogue. Processing of the current instruction is then terminated and the library maintenance prompt is re-displayed so that you can take corrective action. The first warning is displayed if you name a file which is not present on the specified unit; the second if you attempt to process a member which is not in the designated library. The third will appear if there is insufficient space available when you are updating or adding to a library. The fourth means that the target library already contains its full complement of 250 members and there is no space in its directory to hold the details of a new one.

12.9 I/O Error Handling

I/O errors are handled using the normal Global System Manager retry mechanism:

```
ERROR ON unit-address file-id error-message
RETRY?:
```

If you decide the error is irrecoverable and key N to the retry prompt, then, if only an input file is affected, \$DLIB will "clean up" to prevent a partial, inconsistent member from being added to the target library, and a new library maintenance prompt will appear so that you can continue with other work. You would normally use the LIS or PRI instruction to obtain an up-to-date index of the target library before continuing.

If an irrecoverable I/O error affects the target library itself, the data file librarian will attempt to write back a consistent index so that the damaged library can be used as input to \$DLIB itself. When \$DLIB is able to recover the library in this limited way, the following message is displayed:

```
END FORCED BY TARGET LIBRARY I/O ERROR
```

This is immediately followed by another target library prompt, so that you can set about creating the new library.

If \$DLIB is unable even to write back the index, the command is terminated with STOP 9501. In this case the error is catastrophic, and you must recover the affected target library from your own back-up copy.

12.10 Inspect and Extract Operations

Three of the four instructions described in this section are unique in as much as they are the only ones that do not require a target library to be established in order to operate successfully.

LIS and PRI allow you to display or print any data file library, and EXT enables you to create an individual file from a specified member of a selected library. EXD, which does require a target library, behaves in the same way as the EXT instruction except that the original library member is deleted from the target library.

12.10.1 LIS - List the Index of a Selected Library

You can use the LIS instruction to display the library index page by page on the screen. You must specify the library-id together with the unit it occupies:

```
Data File Library Maintenance
: LIS Library: library-id UNIT: unit-id
.....
..... (listing of specified library)
.....
Data File Library Maintenance:
```

For example, if you wish to display the index of library E.TEXT from unit 205 you must key:

```
: LIS Library: E.TEXT UNIT: 205
```

and the first page will appear.

If the LIS instruction needs to output more information than can be contained on a single screen display the prompt:

```
Next Page?:
```

is output on the base line. You must key Y, <CR> (or any single character apart from N) to obtain the next page of output. If you reply N no more of the index will be displayed, and the library maintenance prompt will re-appear.

The format of the display is that of a header, appearing on every page, an information line for each member contained in the data file library, and trailer information, only output on the very last page. The information lines are arranged in alphabetical order of member-id.

The **header** contains the library-id, the title, the unit-id and the current date and time.

Each **information line** consists of the member-id, followed by its creation date, type, size, extent and title. The size is the number of bytes of file space in the space allocated to the member that is used, the extent indicates the amount of main storage the member requires. The

type displayed is the file type of the member, and the creation date specifies the date at which the member was created or last updated.

The **trailer information** at the very end of the listing indicates the date at which the library was last updated, its size, the number of members it currently contains, and the amount of free space available for new members.

12.10.2 PRI - Print the Index of a Selected Library

You can use the PRI instruction to cause a specified data file library index to be printed on the unit assigned to \$PR:

```
Data File Library Maintenance
:PRI Library:library-id UNIT:unit-id
(The specified library index is now printed)
Data File Library Maintenance
:
```

For example, if you wish to print the index of library E.TEXT from unit 205 you must key:

```
:PRI Library:E.TEXT UNIT:205
```

The index will be printed, and another data file library maintenance prompt will appear.

When \$PR is assigned to a direct access volume, rather than a printer, the PRI instruction writes print records to the file D.fffff, where fffff is the library-id suffix.

12.10.3 EXT - Extract a Member from a Selected Library

The EXT instruction enables you to create an individual file from a specified member of a selected data file library. The library is read-only as far as EXT is concerned.

Despite the name of the command, the member is not actually physically extracted, but only a copy of it made. You must specify the library and its unit, together with the member-id and the name and unit for the new file to be created from it:

```
Data File Library Maintenance
:EXT Library:library-id UNIT:unit-id
Extract:member-id To:file-id UNIT:unit-id Extracted
Data File Library Maintenance
:
```

12.10.4 EXD - Extract and Delete a Library Member

The EXD command enables you to create an individual file from a specified member of the target data file library followed by subsequent deletion of that member. This command is a combination of the EXT command (section 12.10.3) followed by the DEL command (section 12.12.3). You must specify the target library and unit, together with the member-id and the unit for the new file to be created from it:

```
GSM READY:$DLIB
Target Library:E.TEXT UNIT:205
Process TEXTS of 22/11/90?:Y
Data File Library Maintenance
:EXD
Extract and delete:S.S1 TO:S.S1 UNIT:205
      Extracted and deleted
Data File Library Maintenance
:END
```

Target Library:

12.10.5 Operating Notes

Although the LIS, PRI and EXT instructions can function without a target library, they also operate when one has been established. In this case, to process the target library itself, you need only key <CR> to the library prompt. For example, supposing you intend to update data file library E.TEXT. You might decide to list its contents prior to including new members with the COP and INC instructions. You might start as follows:

```
GSM READY:$DLIB
Target Library:E.TEXT UNIT:205
Process TEXTS OF 22/11/90?:Y
Data File Library Maintenance
: LIS Library:<CR>
.....
..... (first page of E.TEXT's index)
.....
Next Page?:<CR>
.....
..... (second, and last page of index)
.....
Data File Library Maintenance
:
```

Prompt	Response	Normal action
FILE:	<i>file-id</i>	If the file is not another data file library, process it. If the file is another data file library, respond with the MEMBER: prompt to see which members of that library you require to process.
	<CR>	Respond with <i>file-id?</i> : selection prompts for all appropriate libraries and files on the specified unit, so you can process them if you wish.
	<CTRL A>	Terminate the instruction.
	<CTRL B>	Process every appropriate member on the unit.
<i>file-id?</i> :	Y	If the file is not another data file library, process it. If the file is another data file library, respond with the MEMBER: prompt to see which members of that library you require to process.
	N	Skip this file, and proceed to the next appropriate one on the specified unit.

	<CR>	Skip this file, and proceed to the next appropriate one on the specified unit.
	<CTRL A>	Do not process this file. Terminate the instruction.
	<CTRL B>	Process the member or members this file contains, and every member from each appropriate file remaining on the unit.
MEMBER:	<i>member-id</i>	Process the specified member.
	<CR>	Respond with <i>member-id?</i> : selection prompts for all members in this library, so that you can process them if you wish.
	<CTRL A>	Process no members from this library.
	<CTRL B>	Process all the members from this library.
<i>member-id?</i> :	Y	Process this member.
	N	Skip this member and proceed to the next.
	<CR>	Skip this member and proceed to the next.
	<CTRL A>	Do not process this member, nor any more from the current library.
	<CTRL B>	Process this member and every other one remaining in the current library.

Table 12.3 - Responding to File, Member and Selection Prompts

If you now decided to examine library E.NEW on 204 which you intend to merge with E.TEXT, you would continue:

```
Data File Library Maintenance
: LIS Library: E.NEW UNIT: 204
.....
..... (first page of E.NEW's index)
.....
```

Next Page?:
etc.

12.11 Including and Comparing Members

The instructions described in this section are employed in creating or amending a target library. You can include new members or replace existing ones using COP or INC; and check that members have been included correctly by comparing them with the originals using COM.

12.11.1 COP - Copy Member(s) into Target Library

The COP instruction allows you to copy one or more selected members into the target library. You first of all specify the input unit containing the new members, which are read-only as far as the copy operation is concerned.

In the simplest case, when you just wish to include a member held as an individual file, you key the file-id to the subsequent prompt. For example, to copy the text file S.S10 on 204 into the target library E.TEXT:

```
Data File Library Maintenance
:COP From unit:204
File:S.S10 Included
Data File Library Maintenance
:
```

If you supply a library-id in response to the File: prompt a subsequent Member: prompt asks you to specify which member or members you require. For example, to include S.SORT from library E.FREE on 209:

```
Data File Library Maintenance
:COP From unit:209
File:E.FREE Member:S.SORT Included
Data File Library Maintenance
:
```

These are the two simplest ways in which to use the File: and Member: prompts. You may also key the special responses: <CR>, <CTRL A> and <CTRL B> to serve as selection codes, with meanings similar to those used in \$F. For example, a single COP instruction might be used to copy text file S.SALES, the entire contents of E.FREE, and members S.SA200 and S.SA201 of library E.NEW, into target library E.TEXT on 205. All these are on unit 209 together with text file S.SATEST and data file library E.SAOLD which are not required:

```
GSM READY:$DLIB
Target Library:E.TEXT UNIT:205
Process TEXTS of 20/11/90?:Y
Data File Library Maintenance
:COP From unit:209
File:<CR>
S.SALES ? :Y
S.SATEST ? :N
E.FREE ? :Y Member:<CTRL B>
    S.A Included
    S.C Included
    .....
    .....
    S.SORT Included
E.NEW ? :Y Member:<CR>
    S.SA100 ? :N
    S.SA200 ? :Y
    S.SA201 ? :Y
```

```

E.SAOLD ?:<CTRL A>
Data File Library Maintenance
:END
Target Library:<ESC>
GSM READY:

```

The copy operation may be used to replace an existing member with an updated version. Both the old and the new one must have the same member-id. Then, whenever \$DLIB detects that you are attempting to include a member whose member-id matches one already present on the target library, it displays the following warning message:

```
MEMBER ALREADY EXISTS - DELETE?
```

If you reply Y the existing member will be replaced by the new version. If you key N, <CR>, (or any single character apart from Y) the target library will remain undisturbed and the new member will be ignored: if you are including a number of members the copy process will then continue with the next satisfying the selection criterion.

12.11.2 INC - Include Data File Libraries as Members

The INC operation allow other data file libraries to be copied into the target library as complete entities instead of offering to include the various members of the library. The INC command is an alternative to the COP command for use only with other data file libraries.

The dialogue for the INC command is similar to that for the COP command. For example:

```

GSM READY:$DLIB
Target Library:E.TEXT UNIT:205
Process TEXTS of 20/11/90?:Y
Data File Library Maintenance
:INC From unit:209
File:<CR>
S.SALES ? :Y
S.SATEST ? :N
E.FREE ? :Y
E.NEW ? :Y
E.SAOLD ? :<CTRL A>
Data File Library Maintenance
:END
Target Library:<ESC>
GSM READY:

```

12.11.3 COM - Compare Member(s) with Originals

The COM instruction is provided so that you can compare members previously copied or merged into the target library with their original versions. The dialogue is similar to that employed in a copy or merge operation, except that the COM instruction is used. For example, suppose you decide to update library E.SA1 with files S.SALES, S.SORT from E.FREE, and the entire contents of E.NEW, and you wish to check that the copy process has been performed accurately. This requires just one COP instruction, followed by a similar COM:

```

GSM READY:$DLIB
Data File Library Maintenance

Library:E.SA1 UNIT:208
Process V1.0 SALES LEDGER SOURCES of 22/11/90?:Y
Data File Library Maintenance
:COP From unit:209
File:<CR>
S.SALES ? :Y Included
S.SATEST ? :N
E.FREE ? :Y Member:S.SORT Included

```

```

E.NEW ?:Y Member:<CTRL B>
S.SA100 Included
.....
.....
.....
S.SA920 Included
E.SAOLD?:<CTRL A>
Data File Library Maintenance
:COM From unit:209
File:<CR>
S.SALES ?:Y Identical
S.SATEST ?:N
E.FREE ?:Y Member:S.SORT Identical
E.NEW ?:Y Member:<CTRL B>
S.SA100 Identical
.....
.....
.....
S.SA920 Identical
E.SAOLD ?:Y Member:<CTRL A>
Data File Library Maintenance
:END
Target Library:<ESC>
GSM READY:

```

Providing that every byte of the member selected from the input unit, apart from its creation date and title, matches those of the copy on the target library, the following confirmation message is displayed (as shown in the above example):

```
Identical
```

The warning messages:

```
DISCREPANCY AT nnnn
```

or:

```
NOT FOUND
```

are displayed if either the copies do not match, or if the member-id of a selected member is not present in the target library. The quantity *nnnn* is the number of the first byte within the two members not to match, counting from 0.

12.12 Rename and Delete Operations

The instructions described in this section can be used to change the file-id or the title of the target library, or rename, re-title or delete one of its members. Titles can be at most 30 characters in length.

12.12.1 CHA - Change Target Library-id or Title

The CHA instruction allows you to change the library-id or title of the target library. For example, to change E.SA, entitled SALES LEDGER SOURCES, to E.SAOLD, SALES LEDGER SOURCES (NOV 90):

```

Data File Library Maintenance
:CHA E.SA TO:E.SAOLD
Title:SALES LEDGER SOURCE (NOV 90)
Data File Library Maintenance
:

```


You may key <CR> to the Title prompt, in which case the existing title will remain unchanged. You may use field editing to edit an existing title. You may replace selected characters at the beginning of the old title by keying your amendment, terminated by <CTRL A>, as the new one.

For example, to change the title of library E.SA from V6.0 SALES LEDGER SOURCES to V6.1 SALES LEDGER SOURCES:

```
Data File Library Maintenance
:CHA E.SA TO:<CR>
Title:V6.1<CTRL A>
Data File Library Maintenance
:
```

Note that if you attempt to change a library-id and a file with the new name is already present on the same unit as the target library, the following warning message is displayed:

```
FILE ALREADY EXISTS - DELETE?:
```

If you key Y the existing file will be deleted and the change operation will continue as though it had never existed. A reply of N, <CR> (or any single character apart from Y) causes the instruction to be abandoned and the library maintenance prompt to re-appear.

12.12.2 REN - Rename or Re-title Target Library Member

You can use the REN instruction to change a particular member-id within a target library, and to alter the title of any type of member. The dialogue is of the general form:

```
Data File Library Maintenance
:REN :old-member-id as:new-member-id
Title:new-title
Data File Library Maintenance
:
```

Field editing can be used to edit an existing title for example to correct a spelling mistake.

You may key <CR> in place of a new member-id or new title, in which case the existing member-id or title will remain unchanged.

You may replace selected characters at the beginning of the old title by keying your amendment, terminated by <CTRL A>. For example, to adjust a version code by changing the title of member S.SA100 from V6.0 SALES ACCT ROOT to V6.1 SALES ACCT ROOT:

```
Data File Library Maintenance
:REN :S.SA100 AS:<CR>
Title:V6.1<CTRL A>
Data File Library Maintenance
:
```

Note that if you rename a library member and the new member-id you choose is already present on the target library, the following warning message is displayed:

```
MEMBER ALREADY EXISTS - DELETE?:
```

If you key Y the old member will be deleted and the rename operation will continue as though it had never existed. A reply of N, <CR> (or any single character apart from Y) causes the instruction to be abandoned and the library maintenance prompt to re-appear.

12.12.3 DEL - Delete Member(s) From the Target Library

The DEL instruction enables you to delete one or more selected members from the target library. To delete one particular member, you simply key its member-id in response to the Member: prompt. For example, to remove S.SA100 from the target library:

```
Data File Library Maintenance
:DEL Member:S.SA100 Deleted
Data File Library Maintenance
:
```

You may also use the special responses <CR>, <CTRL A> and <CTRL B>, to delete a selection of members, using the scheme summarised in the lower half of table 12.3. For example:

```
Data File Library Maintenance
:DEL Member:<CR>
  S.SA200 ? :N
  S.SA205 ? :N
  S.SA300 ? :Y Deleted
  S.SA310 ? :Y Deleted
  S.SA400 ? :N
  S.SA500 ? :Y Deleted
  S.SA600 ? :<CTRL A>
Data File Library Maintenance
:
```

Here S.SA300, S.SA310 and S.SA500 are deleted from the target library, which has been displayed member by member for selection, following a <CR> response to the member prompt. The keying of <CTRL A> to the select prompt terminates the process.

12.13 Space Saving Operations

The instruction described in this section can be used to limit the size of the target data library. The unwanted free space is returned to the volume for re-allocation. Note that the size of a data library cannot be increased from within \$DLIB. To increase the size of a data library, the \$F COP instruction must be used (see section 15.10).

12.13.1 TRU - Truncate Target Library

The TRU instruction returns any unwanted free space in the target library to the volume it occupies, so that the storage thus released becomes available for re-allocation. The dialogue allows you to specify how much free space you require to remain in the library, once truncation has taken place:

```
Data File Library Maintenance
:TRU New spare space:required-free-space Truncated
Data File Library Maintenance
:
```

If you never intend to update the library you will require no spare space at all, and in this case you may key <CR> or 0 in response to the 'New spare space' prompt. Normally, however, you should leave enough room to hold several extra members, to prevent re-organisation having to take place unnecessarily.

The following example shows how a data file library might be set up initially:

```
GSM READY:$DLIB
Data file library maintenance
```

```

Target Library:E.SA UNIT:205
New?:Y Size:0
Data File Library Maintenance
:
(COP or INC instructions to include the initial members).
Data File Library Maintenance
:TRU New spare space:100K
Data File Library Maintenance
:END
Target Library:<ESCAPE>
GSM READY:

```

By specifying an initial Size request of 0 in the third line of the dialogue, the maximum amount of contiguous space available is allocated to the library initially. The TRU instruction returns any unwanted space to the volume, but keeps 100K bytes spare for subsequent expansion.

Note that if you request more spare space than is actually present on the library the following warning message will be displayed and the truncation operation will be abandoned:

```
TOO LARGE
```

12.14 Volume checking Operations

The instruction described in this section can be used by job management to control the mounting of a specified input volume required for subsequent library maintenance operations.

12.14.1 I - Input Volume Checking

The I instruction prompts you for the volume-id, which it remembers so that it can open the next file specified by an instruction using volume-id checking. This will cause a mount prompt to appear if the volume is not already present on the indicated unit. For example, suppose you wish to copy the file S.OURSORT from volume OURDEV on 204 to the target library. You can use the I instruction to control the mounting of OURDEV as follows:

```

Data File Library Maintenance
:I Volume-id:OURDEV
Data File Library Maintenance
:COP Unit:204
File:OURSORT

```

If at this stage OURDEV is not present on unit 204, the following mount prompt will appear:

```
PLEASE MOUNT OURDEV ON 204 - HARD DISK AND KEY <CR>:
```

Since this is one of the special prompts that bypass job management, it will always be displayed on the console, even when the other responses to \$DLIB are supplied in the dialogue table. The operator must load the requested volume and key Y, <CR> (or any single character apart from N) to continue. A reply of N terminates job management.

12.15 ABA - Abandoning the Library Update

The ABA instruction, deliberately omitted from table 12.1, should only be used if a catastrophic error occurs making the target library completely un-usable. **The ABA command will delete the target library.**

For example, to abandon the update and delete the target library:

Chapter 12 - Data File Library Maintenance (\$DLIB)

Data File Library Maintenance

:ABA

TARGET LIBRARY WILL BE DELETED - ARE YOU SURE?:Y Abandoned

\$95 TARGET LIBRARY:

If the ABA instruction is used, the target library must be restored from back-up.

13. \$E, \$END - Signing Off

The functionality of \$E and \$END is dependent on the host operating system. \$E is documented in Chapter 4 of the Global Operating Manual. \$END is documented in Chapter 4 of the Global Operating Manual (Unix).

14. \$EXEC - Timed Execution of Programs

\$EXEC is used in conjunction with the menu system (see Chapter 7 of the Global System Manager Manual) to provide a means of executing a program, or programs, at timed intervals. For example, you might wish to backup your hard disk to tape during the night while nobody is using the computer.

\$EXEC operates by repeatedly executing a group of menu lines. To use \$EXEC you must create a sub-menu with menu entries for the program or programs you wish to be executed followed by an entry for \$EXEC. The subject menu lines must be at the start of the menu and any keyboard dialogue must be provided by Job Management or the type-ahead features of the menu system. Any menu lines following the \$EXEC menu line are not executed by \$EXEC. See figure 14.1 for an example of the arrangement of the menu lines.

14.1 Starting \$EXEC

To initiate a timed execution you must select the \$EXEC menu entry. You may do this by hand or by using the features of \$AUTH that allow the automatic execution of menu entries when Global System Manager is loaded (see section 6.3 of the Global System Manager Manual). Once you have started \$EXEC it will run continuously unless you halt the program explicitly or turn off your computer. If the computer is left on overnight \$EXEC continues to run.

\$EXEC prompts for a time range (start time and stop time) and an interval (see figure 14.2) - these values must be supplied by the type-ahead feature of the menuing system. Once \$EXEC has been started by selecting its menu line, it executes the subject menu lines at the defined intervals while the time remains within the time range. If you start \$EXEC outside of the time range the subject menu lines are first executed at the start time. If you start \$EXEC within the time range the subject menu lines are executed immediately and thereafter at the defined interval.

For example: If the type-ahead associated with the \$EXEC line in the menu specifies the following:

```
Start time (hh.mm):14.00
Stop time (hh.mm):17.00
Interval (hh.mm):1.00
```

Then if you execute the \$EXEC line in the menu and the current time is 13.30 the subject menu lines will be executed at 14.00, 15.00, 16.00 and 17.00. If the current time is 14.30 the subject menu lines will be executed at 14.30, 15.00, 16.00 and 17.00. Note that the time has to be strictly greater than the stop time before execution ceases.

14.2 Daily Execution

To set up menu lines for daily execution you must specify an interval that is longer than the time range. The first execution will occur when the start time is reached, but because the next run time falls outside the time range the next execution will be deferred to the same start time but the next day. For example:

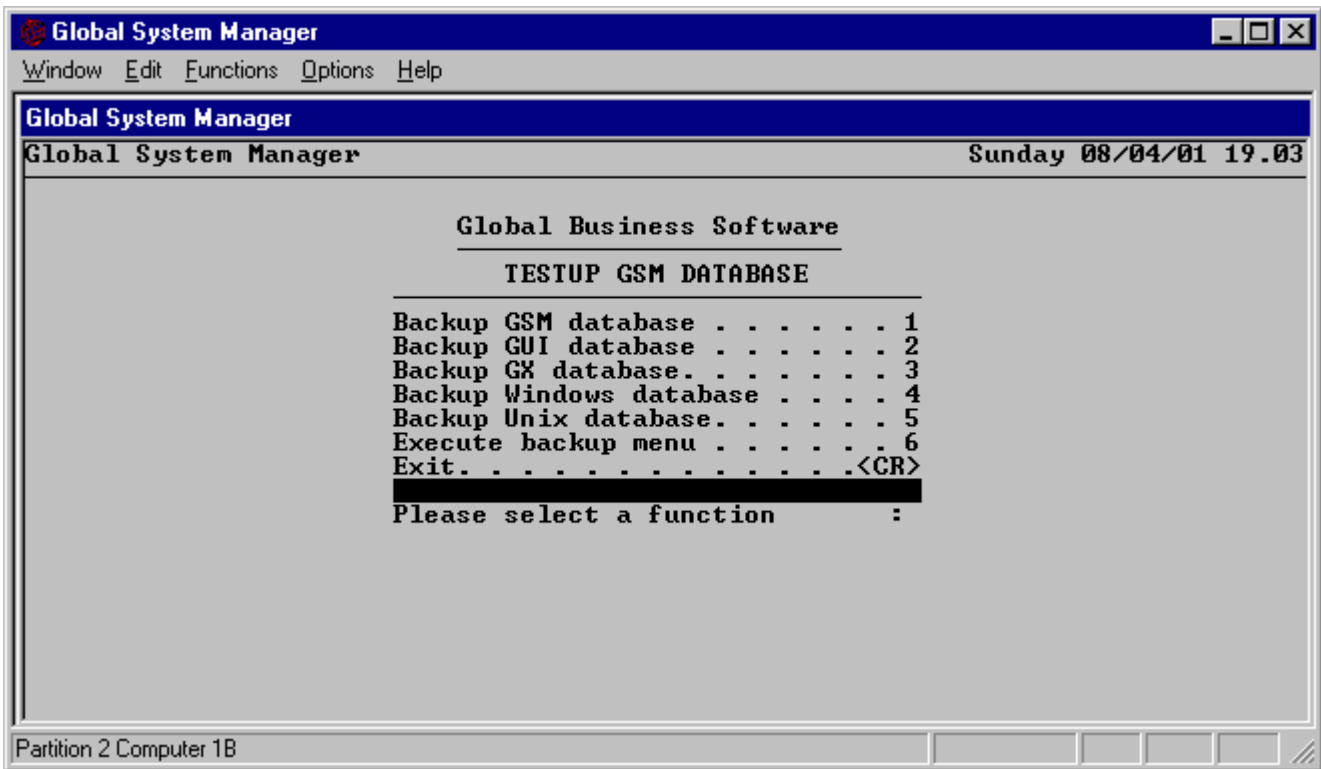


Figure 14.1 - \$EXEC Menu Lines

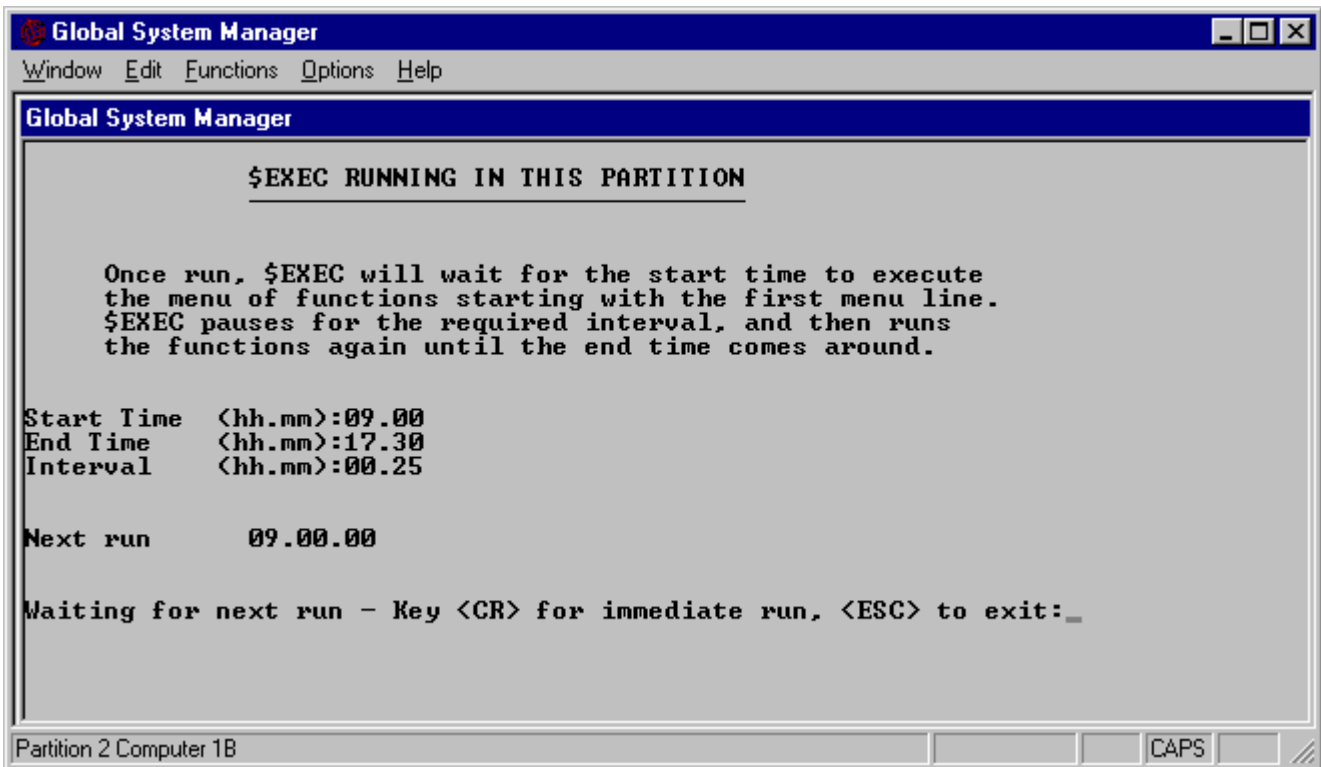


Figure 14.2 - Running \$EXEC

If the type-ahead associated with the \$EXEC line in the menu specifies the following:

```

Start time (hh.mm):19.00
Stop time (hh.mm):19.01
Interval (hh.mm):1.00
    
```

then the subject menu lines will be executed at 19.00, and since the next time of execution would normally be $19.00 + 1.00 = 20.00$ (which is outside the time range) the subject menu lines will not be executed again until 19.00 the next day.

14.3 Restrictions in \$EXEC

Although \$EXEC is potentially very powerful it does suffer from two severe restrictions:

- **None of lines to be \$EXEC'ed can chain to other menus. In practice this means that the menu lines executed by \$EXEC must not be of type M (new menu file) or type N (next menu file);**
- **Because of the interaction between \$EXEC and \$MH (the menu handler) any program to be \$EXEC'ed via a stand-alone program menu line must not use SAA-style menus.**

15. \$F - File Maintenance

The most commonly used facilities of the file utility, \$F, are described in Chapter 5 of the Global System Manager Manual, which also contains a basic introduction to this utility and should be read first if you are not familiar with the \$F command. The following account describes **all** the instruction and selection codes currently available, including those which are retained only to maintain compatibility with earlier versions of Global System Manager. The instructions are grouped thematically, so that certain instruction codes, particularly 'compound' instructions such as CCF which combine the effects of other 'simple' instructions, are covered in two or more sections. Selection codes are dealt with at the end of this chapter.

15.1 The Device Prompts

When you run \$F you are first prompted for the addresses or unit-ids of the input and output devices (i.e. the units you are copying from and to). Any valid address of a direct access device can be supplied, including addresses of hard disk drives, diskette drives and RAM disks on remote computers linked to your own in a network environment. Thus, to copy a file from a diskette on computer C to the RAM disk on computer G you might begin:

```
GSM READY:$F
$66 INPUT DEVICE:c80
$66 OUTPUT DEVICE:g09
$66 FILE MAINTENANCE
```

You may, however, wish to perform a \$F operation on one disk only; in that case supply its address or unit-id to the input device prompt and key <CR> to the output device prompt.

15.2 Listing Instruction Codes and Changing Devices

The ? instruction produces a list on the screen or printer of all the \$F instructions available - currently 58 if one excludes the 2 instructions mentioned in this section. 56 of the instructions are shown in table 15.1 with brief descriptions and references to the section in which they are described in detail. The **LOA** and **SRL** instructions are only preserved to maintain compatibility with earlier versions of Global System Manager, and should not be used.

The <CR> instruction returns you to the input device prompt, so you can specify new input and output devices.

15.3 Listing and Printing Volume Directories

The **LIS** and **PRI** instructions produce a list on the screen or printer respectively of all the files on either the input or output volumes. Table 15.2 describes the file organization codes and prefixes which may appear in these listings. The format of a domain listing produced by the LIS instruction in 7.0, and later, versions of Global System Manager differs from that produced by earlier versions. The Volume Description (see section 15.16) of each subunit is displayed instead of the "start, size and protection" information. The \$V command (see Chapter 4 of the Global Operating Manual) can be used to produce a domain listing which includes the "start, size and protection" information for each subunit.

Instruction	Function	Section
ALL	Allocate a dummy relative sequential file	15.14
BAC	Backup a Global System Manager volume	15.9.1

BAR	Backup input onto output and rename output volume	15.9.1
BCF	Backup input onto output and compare files	15.9.1
BCR	Backup, verify and rename output volume	15.9.1
CCF	Copy selected files to output and compare with input	15.10
CFI	Compare file on input and output volumes	15.11
CHA	Rename output volume	15.7
CMP	Patch monitor for memory pages	15.15.5
COM	Compare volumes duplicated by REP command	15.11
CON	Condense output volume	15.5
COP	Copy selected input files to output volume	15.10
CSF	Copy files to output and password-protect input files	15.6.2
CVS	Compare 'save' file with input or output volume	15.11
DEL	Delete selected files from output volume	15.10
DES	Amend a single volume description	15.16.1
DEU	Amend the volume descriptions for a domain	15.16.2
FIF	Serialize selected template files on output volume	15.12
FIX	Serialize all template files on output volume	15.12
I	Prompt operator for input volume when running under Job Management	15.13
INI	Initialize output volume	15.8
INS	Install and patch bootstrap on disk	15.15.4
IO	Prompt operator for both input and output volumes when running under Job Management	15.13
LIM	Limit copy buffer size	15.10
LIS	List volume directory on the screen	15.3
LI?	List directory to show product file information	15.3
LSF	List selected files on the screen	15.3
LSU	List selected subunits on the screen	15.3
LS?	List directory to show selected product file information	15.3
MCF	Move and compare selected files	15.10
MOV	Move selected files to output volume and delete from input volume	15.10
O	Prompt operator for output volume when running under Job Management	15.13
PAM	Patch monitor to change stack/bank size and number of memory banks/pages	15.15.1
PFI	Update-protect selected files on the output volume	15.6.1
PIP	Patch monitor library unit address	15.15.3
PRI	Print volume directory	15.3
PRO	Scratch-protect output volume	15.6.1
PR?	Print directory including product file information	15.3
PSF	Print selected files	15.3
PS?	Print directory showing selected product file information	15.3
PUF	Patch monitor file with address of master SYSRES volume	15.15.2
REL	Release output volume from scratch protection	15.6.1
REN	Rename selected file on output volume	15.7
REP	Replicate native input on output volume	15.9.2
RES	Restore native output volume from 'save' file	15.9.2
RFI	Release selected files from scratch protection	15.6.1
SAV	Create 'save' file from native input volume	15.9.2

SCR	Scratch output volume	15.8
SEC	Set or remove password protection on selected files	15.6.2
SER	Establish/change product serial number and expiry date	15.12
SET	Establish or alter current password	15.6.2
SIZ	Change size and organization of existing output file	15.14
SWV	Set up spool/work/normal volume and change the allocation size	15.14
TRU	Truncate selected files on output volume	15.5
VER	Verify output volume	15.4
VFI	Verify selected files on output volume	15.4

Table 15.1 - \$F Instruction Codes

The domain listing produced by the PRI instruction in 7.0, and later, versions of Global System Manager includes the volume descriptions as well as the "start, size and protection" information for each subunit.

The **LSF** and **PSF** (List/Print Selected Files) instructions work in a similar way, but are used in conjunction with a file name or selection code to list files with a common prefix or suffix.

The **LI?** and **PR?** instructions produce complete directory listings of the input or output devices, but in a special format which shows the serial number and expiry date of product files in place of the usual creation date and file start address information. See section 15.12.

The **LS?** and **PS?** instructions work in a similar way, but are used in conjunction with a selection code to list or print individual or selected files in the special product file format. See section 15.12.

The **LSU** instruction can only be used when the selected unit is a domain. A list is produced of the subunits on the domain in the same format as displayed by the **LIS** instruction. The unit number of the first sub-unit to be displayed may be specified.

15.4 Verifying Volumes and Files

The **VER** and **VFI** instructions enable you to verify the output volume or selected files on it respectively. Faulty tracks will be reported in the same way as in the comparable \$V operation.

15.5 Condensing Volumes and Truncating Files

The **CON** instruction is used to condense the output volume so that the files are held on it in the most compact form. Since the files are shuffled about in situ the instruction will have no effect if the volume is nearly full; in such a case use **BAC** (see section 15.9.1) to copy the volume to another disk and then back again to achieve the desired effect. Alternatively, use \$REORG (see Chapter 4 of the Global Operating Manual) to condense the unit in place.

You use the **TRU** instruction to truncate a single named file or a selection of files on the output volume in order to make any spare space available for reallocation. **Great care must be taken to ensure that the files you are truncating can be condensed safely without introducing file corruption.**

For example, under no circumstances should you use the \$F TRU instruction to truncate DMAM databases. If you attempt to truncate a DMAM database, the following error message will be displayed:

NOT ALLOWED ON DMAM FILES

Code	File Type	Associated File Prefixes	
BE	Bootstrap file or library	#.	Bootstrap library
BO	Physical bootstrap		
BP	Global Planner plan file	Y.	Global Planner plan file
CF	Global 2000 software integrity file		
CL/CL*	Compilation file or library (* indicates a serialized file)	C.	Compilation file or library
CW	Compiler work file		
DB	Global Finder database		
DD	Speedbase data dictionary		
DL	Data Library file		
DM	DMAM database file		
IF	Integrator file		
IN	Configuration file	++	Configuration file
IS	Indexed sequential file	A.	Configurator action file
LB	Data file library	E.	Data file library
LG	System Log file		
MN	Monitor, Nucleus file or library	+	Library of machine specific components used when Global System Manager is loaded
PF	Partially created file		
PL/PL*	Program file or library (* indicates a serialized file)	P. \$. %.	Program library from \$LIB Terminal Attribute Program Nucleus Assist File (e.g. Tape Handler)
RS	Relative sequential data file or print file	D. L. L. M. X. Z.	Print report file Global Writer or Global Graphics print file Listing file from \$COBOL \$LINK link map Listing file from \$XREF, Global Finder report file \$ZAP zap file
SA	Save file created by the \$F SAV instruction		
SB	Speedbase database		
SF	Speedbase full backup file		
SI	Speedbase full incremental backup file		
SJ	Speedbase part incremental backup file		
SK	Data skeleton		
SN	Speedbase Novell Btrieve database schema file		

SP	\$SPOOL schedule file		
ST	Speedbase TAP file	T>	Speedbase TAP file
SU	Speedbase Unix C-ISAM database schema file		
SW	Swap file		
SY	System parameterization file		
TF	Text file	B. F. Q. S. T.	Backup of \$EDIT source file \$MACRO source file Record/Playback script Source file or copy library Product translation text
US	User file		
VL	Variable length record file	K. \$\$	Global AutoClerk control file Menu file
WP	Global Writer document		
10 - 99	User-defined basic direct file		
#01 - #99	Data Security System backup file. The code gives the number of the volume backup cycle the file is named after		

Table 15.2 - Global System Manager File Types and File Prefixes

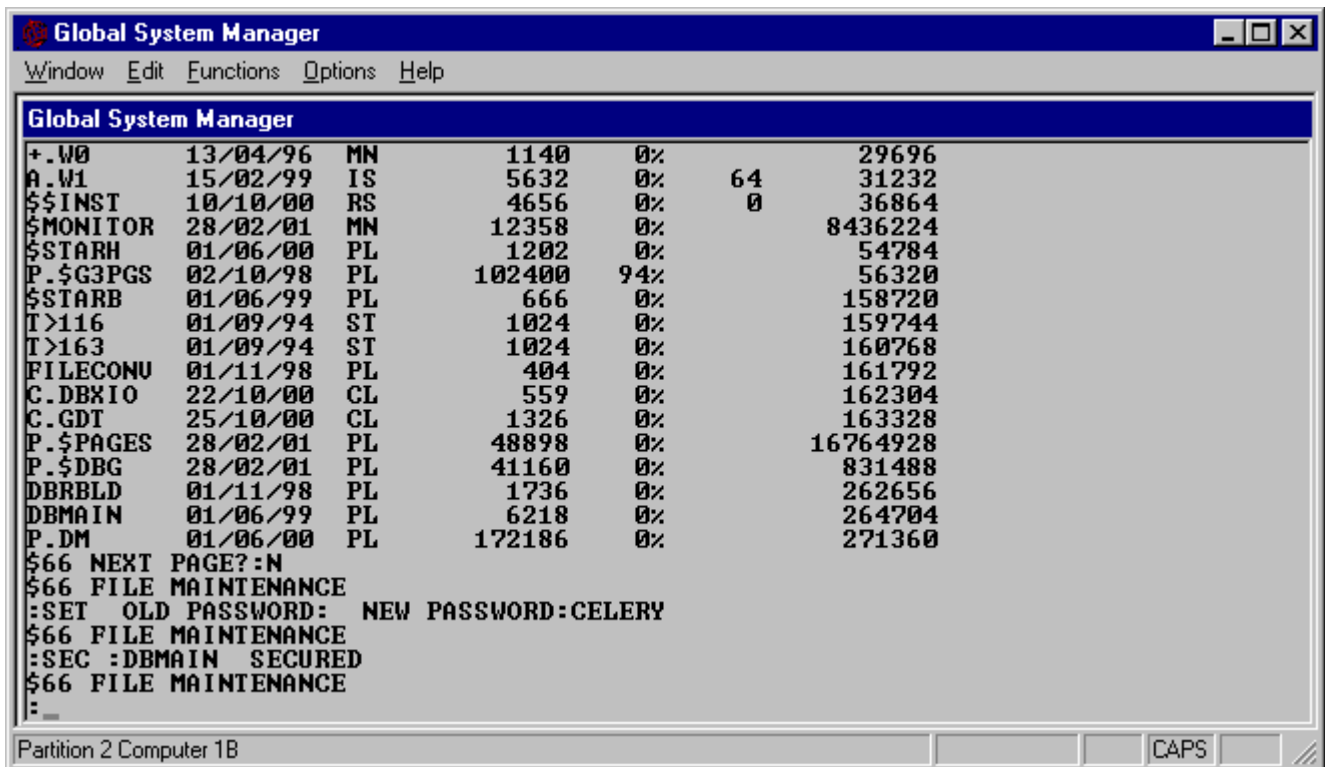
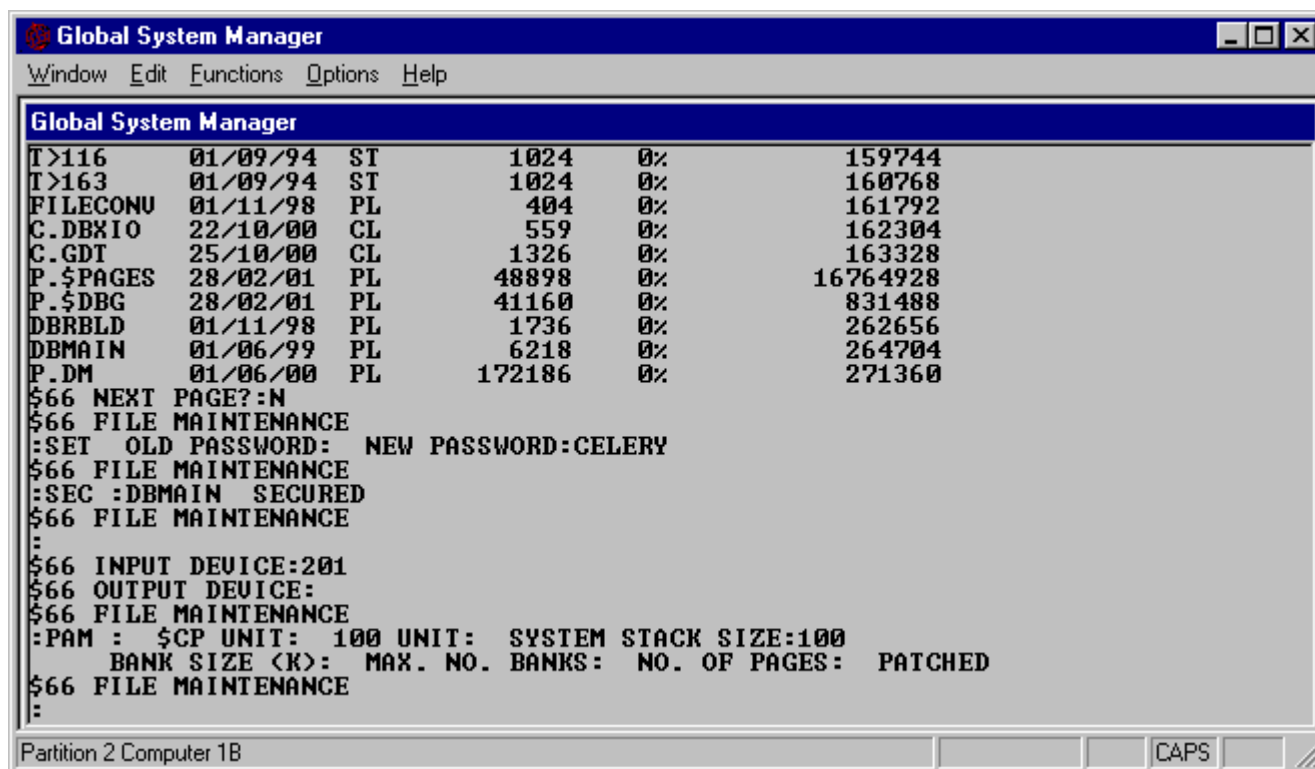


Figure 15.1 - Setting and removing a password



```

Global System Manager
Window Edit Functions Options Help

Global System Manager
T>116      01/09/94  ST      1024      0%      159744
T>163      01/09/94  ST      1024      0%      160768
FILECONU  01/11/98  PL       404      0%      161792
C.DBXIO    22/10/00  CL       559      0%      162304
C.GDT      25/10/00  CL      1326      0%      163328
P.$PAGES  28/02/01  PL     48898      0%      16764928
P.$DBG     28/02/01  PL     41160      0%      831488
DBRBLD     01/11/98  PL       1736      0%      262656
DBMAIN     01/06/99  PL       6218      0%      264704
P.DM       01/06/00  PL     172186      0%      271360
$66 NEXT PAGE?:N
$66 FILE MAINTENANCE
:SET  OLD PASSWORD:  NEW PASSWORD:CELERY
$66 FILE MAINTENANCE
:SEC :DBMAIN SECURED
$66 FILE MAINTENANCE
:
$66 INPUT DEVICE:201
$66 OUTPUT DEVICE:
$66 FILE MAINTENANCE
:PAM : $CP UNIT: 100 UNIT:  SYSTEM STACK SIZE:100
      BANK SIZE (K):  MAX. NO. BANKS:  NO. OF PAGES:  PATCHED
$66 FILE MAINTENANCE
:
Partition 2 Computer 1B
CAPS

```

Figure 15.2 - The PAM command

15.6 Scratch Protection and Password Protection

Volumes and individual or selected files can be protected from accidental scratching, deletion or unauthorized access by imposing scratch or password protection on them.

15.6.1 Scratch Protection

The **PRO** instruction protects the output volume from being scratched; however, initialization will be allowed to proceed after a warning, and individual files can still be deleted. Use the **REL** instruction to release the volume from protection.

The **PFI** instruction allows you to update-protect specified file(s) on the output volume (so you cannot delete or write to them), and can be cancelled by the **RFI** instruction. Note that if you protect a file the volume it occupies will automatically be protected as well. If you release scratch protection from the last file so protected on a volume the volume itself will still be protected; you must use **REL** to release protection at volume level. Note also that copies of a file protected by **PFI** will not themselves be protected.

15.6.2 Password Protection

Any type of file can be associated with a password so that it cannot be read unless the correct password is supplied. Note that password-protected files can still be copied or deleted without using the password, but they cannot be opened or read.

Before securing any particular file you must use the **SET** instruction to establish or alter the current password. For example, to change the current password:

```

$66 FILE MAINTENANCE
:SET OLD PASSWORD:password-1 NEW PASSWORD:password-2

```

You should key <CR> in the place of *password-1* when establishing a password for the first time in a session, and <CR> instead of *password-2* when you intend to release files, already secured by *password-1*, from protection. You only supply both inputs when you want to change passwords. For example, to change the password from CHICKEN to EGG:

```
$66 FILE MAINTENANCE
:SET OLD PASSWORD:CHICKEN NEW PASSWORD:EGG
$66 FILE MAINTENANCE
:
```

Passwords may be up to 8 characters in length, and it is recommended that they should be at least 6 characters long to maximize security.

You should note that passwords established by **SET** are available for use for as long as \$F is active, and only cleared when you quit. Files secured with such a password by use of the **SEC** instruction (see below) will, of course, remain protected until the protection is explicitly removed, again using the **SEC** instruction.

You use the **SEC** instruction to secure a file or selection of files on the output volume with a password, or to change or remove the password; the old and new passwords must have already been established using the **SET** instruction. Only files which are not already password-protected, or whose password matches the old password, can have their password changed. If the file has a different password the error message "WRONG PASSWORD" is displayed, and the file remains undisturbed. For example, suppose all the source files on an output volume are secure files, and you have used **SET**, as above, to specify that password CHICKEN is to be changed to EGG. You use the **SEC** instruction as follows:

```
$66 FILE MAINTENANCE
:SEC :S.<CTRL B>
S.SA SECURED
S.SA100 WRONG PASSWORD
S.SALES SECURED
$66 FILE MAINTENANCE
:
```

S.SA100 has not been secured with the new password, since its password is not CHICKEN. Suppose that at some later point you wish to remove password protection from S.SA:

```
$66 FILE MAINTENANCE
:SET OLD PASSWORD:EGG NEW PASSWORD:<CR>
$66 FILE MAINTENANCE
:SEC :S.SA PASSWORD REMOVED
$66 FILE MAINTENANCE
:
```

When Global System Manager comes to open the first password-protected file of a session it interrogates the operator before continuing with a prompt of the form:

```
PLEASE KEY PASSWORD FOR file-id:
```

If the operator cannot supply the password he or she must key <CR> to abandon the processing of the file. Otherwise, when the correct password is supplied Global System Manager remembers it, and continues normally. Any other protected file with the **same** password will be opened without further prompting; the prompt appears only when a file with a different password is met. This means that if all the secure files used by an application are assigned the same password, then it need only be supplied once.

The password is remembered for as long as \$F is active, and is cleared when you quit. Files secured with such a password by use of the **SEC** instruction will, of course, remain protected until the protection is explicitly removed, again using the **SEC** instruction.

The **CSF** (Copy Secure File) instruction is particularly useful for controlling the amendment of files during program development. It puts the current password on one or a selection of input files copied to the output, so that other programmers who attempt to access the source file you have copied to a work volume to amend will be prompted for the password.

15.7 Renaming Volumes and Files

The **CHA** instruction is used to change the volume-id of the output volume. The **REN** instruction renames a specified file on the output volume.

The **BAR** (Backup and Rename) instruction combines the effects of a **BAC** and a **CHA** operation: The output volume is scratched and renamed with the same volume-id as the input volume, the contents of which are copied to it a file at a time.

The **BCR** (Backup, Compare and Rename) instruction combines the effects of the **BAC**, **CHA** and **CFI** instructions: The output volume is scratched and given the same name as the input, and then the input files are copied to it a file at a time, each one in turn being compared with the original on the input volume.

15.8 Initializing and Scratching Volumes

The **INI** and **SCR** instructions described below both **destroy** any existing data on the output volume if they are allowed to proceed. If you suspect that the volume might contain valuable data, use **LIS** to display its directory to check that it may safely be initialized or scratched.

You use the **INI** instruction to initialize the output volume. It cannot be used on a domain, although it may be applied to a subvolume which has previously been initialized by the \$V command. (**INI** in fact duplicates a subset of \$V and is only included for compatibility with earlier versions of Global System Manager.) A warning message will appear if the volume has previously been protected by a **PRO** or **PFI** instruction, and may also occur spuriously for some factory-fresh volumes. Furthermore, the volume-id may appear as @@@@ if the volume has never previously been initialized.

If you do not wish to use the optimum interleaving factor for your computer, you must key the numeric access option you require in response to the OPTION: prompt. Among the access options available are: 1 = no interleaving; 52 = interleaving of alternate sectors; 53 = every third sector; 54 = every fourth sector; etc. If you attempt to initialize a subvolume which has not previously been set up by \$V, the error message "SUBVOLUME NOT ALLOCATED" will appear. The operation will be abandoned and the instruction prompt will be redisplayed.

You use the **SCR** instruction to 'scratch' (i.e. delete all the files from) the output volume, whilst leaving the volume-id intact. Note that for performance reasons you should scratch a volume rather than delete all its files using the **DEL** instruction. This is because Global System Manager remembers how much of the directory has been used to decide how many sectors need to be searched whenever a file is opened. This end of directory marker is reset by an **SCR** instruction, but is not modified by a **DEL**.

15.9 Backing Up Volumes

\$F includes commands to back-up both volumes with Global System Manager directories and volumes without Global System Manager directories.

15.9.1 Backing Up Volumes with Global System Manager Directories

The **BAC** instruction can be used to copy all the files on a Global System Manager diskette or hard disk subvolume onto another diskette or subvolume. The output volume is first scratched, and then the files on the input are copied to it one at a time. The **BAC** instruction can also be used to copy all the volumes on one hard disk domain to another. The subvolumes on the output domain must be allocated so that they correspond numerically to the subvolumes to be copied on the input domain.

The **BAR** (Backup and Rename) instruction combines the effects of the **BAC** and **CHA** instructions, the output volume is first scratched and renamed, and then the files on the input are copied to it.

The **BCF** (Backup and Compare Files) instruction combines the effects of **BAC** and **CFI** instructions. The output volume is scratched, and then the files on the input volume are copied to it one at a time, comparing each of them in turn to the original on the input.

The **BCR** (Backup, Compare and Rename) instruction combines the effects of the **BAC**, **CHA** and **CFI** instructions. The output volume is scratched and given the same name as the input, and then the files on the input are copied to it one at a time, each one in turn being compared with the original on the input.

15.9.2 Backing Up Volumes without Global System Manager Directories

The **REP** (Replicate) instruction copies the input diskette volume to the output track by track, without considering Global System Manager directories and files. It is used for copying BACNAT diskettes (those diskettes used with Global System Manager which contain host operating system files) which do not contain a Global System Manager directory. The two volumes can be compared using **COM** instruction (see section 15.11).

The **SAV** and **RES** instructions are used on single diskette computers for the same purpose. The **SAV** instruction is employed first to save the entire contents of the input volume (which **must** be a diskette) as a single 'save' file on the output volume. Note that the output volume must be larger than the input in order to accommodate a Global System Manager directory in addition to the contents of the input diskette.

Input and output devices must then be reversed, and the **RES** instruction used to transfer the contents of the 'save' file to the output volume, where they assume their original form.

The **CVS** (Compare Volume with Save File) instruction (see section 15.11) can be used after either a **SAV** or **RES** operation to compare the input volume with the 'save' file or the 'save' file with the output volume. Any discrepancies will be noted.

15.10 Copying, Moving and Deleting Files

The **COP** instruction is used to copy individual files, or a selection of files, from the input volume to the output volume. You can optionally change the name or increase the size allocation of an individual file when you copy it (see section 15.18 for a full description of the options available when specifying the size of the output file). Key <CR> to both prompts to leave the situation unchanged. Note that the **COP** instruction, unlike the **BAC** instruction (see section 15.9.1), does not first scratch the output volume, and so is useful for accumulating files from a number of different sources on a single output volume.

The **CCF** (Copy and Compare Files) instruction combines the effects of the **COP** and **CFI** instructions. It copies one or a selection of files from the input to the output volume, and compares the input and output copies to check that they are identical. The **CSF** (Copy Secure File) instruction copies one or a selection of files from the input to the output and puts password protection on the input file(s). The **SET** instruction (see section 15.6.2) must already have been used during the current \$F session to establish the current password.

The **MOV** instruction works in the same way as **COP**, but has the additional effect of deleting from the input volume any file copied to the output. The **MCF** (Move and Compare Files) instruction combines the effects of the **MOV** and **CFI** instructions, copying specified files from input to output, comparing them and, provided they are identical, deleting them from the input volume.

The **LIM** instruction can be used to limit the size of the buffer used by the **COP** instructions and its variants. This instruction is useful when copying to and from diskettes on a multi-user system where the overhead of copying large amounts of data in a single operation can affect time-slicing and perceived performance.

The **DEL** instruction deletes specified files from the output volume. Note that if you wish to delete **all** the remaining files from the output volume it's more efficient to use **SCR** rather than **DEL** <CTRL B>. This is because Global System Manager remembers how much of the directory has been used to decide how many sectors need to be searched whenever a file is opened. This end of directory marker is reset by a **SCR** instruction, but is not modified by a **DEL**. However, under some circumstances, files deleted using the \$F **DEL** instruction can be recovered using the \$RECOVER command. A directory scratched using the \$F **SCR** instruction can never be recovered.

15.11 Comparing Volumes and Files

The **COM** instruction compares the data on the input with that on the output volume, byte by byte and track by track, without considering Global System Manager directories and files. It is used to check that volumes duplicated by replication are indeed identical. Note that the volume-id prompt you might expect to see is replaced with one which displays the unit addresses involved for confirmation; volume-ids cannot be displayed since the volumes do not contain Global System Manager directories. The message "DISCREPANCY AT *byte number*" will appear if the two differ.

The **CFI** (Compare Files) instruction allows you to compare one or a selection of files on the input and output volumes. If they differ in the data they hold the message "DISCREPANCY AT *byte number*" will appear, indicating the first byte (counting from zero) which differs. If the used part of the files differs in size the message "FILE SIZE DIFFERS" will appear, and no further comparison takes place as the files are obviously different. However, since only the used part of

the files are compared you can use **CFI** to check that a file which has been copied to a larger area has been transferred correctly.

The **CVS** (Compare Volume with Save File) instruction is used to compare the 'save' file produced by a **SAV** operation with the original input volume or with the output volume created from it by a **RES** operation.

The **CCF** (Copy and Compare Files) instruction (see section 15.10) copies files from input to output and compares them with the originals on the input volume.

The **MCF** (Move and Compare Files) instruction (see section 15.10) copies files from input to output and compares them with the originals before deleting them from the input volume.

The **BCF** (Backup and Compare Files) instruction (see section 15.9.1) scratches the output volume and then copies to it all the files on the input volume, comparing each of them in turn with the original on the input.

The **BCR** (Backup, Compare and Rename) instruction (see section 15.9.1) works in the same way as **BCF**, but has the additional effect of renaming the output with the same name as the input volume.

15.12 Serialization Instructions

The instructions described in this section are used in the creation of **product files** by developers of Global Cobol software, in order to protect it from unauthorized use. The technique involves the allocation of a unique product serial number to each Global System Manager that is generated. A product serial number and optional expiry date can be affixed to any **template file**: A compilation file, compilation library, program file or program library which you yourself have created from source. Global System Manager will then ensure that the file can only be used on the system with a compatible serial number, up to the day specified by the expiry date (if any).

The **SER** instruction bears a similar relation to serialization as **SET** does to password protection: It establishes a product serial number and optional expiry date for any product files you wish to create during the current session.

After you have established the current serial number and expiry date you will normally use **BAC** or **COP** operation (or any other instruction which uses the internal copying routine) to transfer your own template files from the input to the output volume. The output files are then automatically tagged with the serial number and optional expiry date established by the **SER** operation, and will thus become product files. A directory listing of the output volume will display an asterisk next to the file organization codes of any product files present.

Note that the **LI?** and **PR?** instructions can be used to display or print a special directory listing of any volume, in which the usual creation date and file start address information is replaced by serial number and expiry date information for any product files present. The **LS?** and **PS?** instructions produce similar listings for selected files.

The currently established serial number and optional expiry date are cleared when you quit \$F. If, however, you wish to establish a new serial number or expiry date, or copy or back up template files without turning them into product files, you can re-run the **SER** instruction. You can then choose to establish a new serial number and expiry date, or key <CR> to the first

prompt to obliterate the current ones. Note that this second course of action will render ineffective any subsequent **FIX** or **FIF** operations (see below).

The **FIX** instruction is used to convert any template files on the output volume into product files without the need for a copy or backing up operation. The operation is most useful if you want to stockpile a series of template customer volumes, each of which contains the files you wish to release, copied without the **SER** instruction in force. You can then use **FIX** to serialize each of the volumes in question when the need arises.

The **FIF** instruction allows you to serialize individual, selected template files on the output volume. The message "NOT A TEMPLATE" will appear if any of the selected files is not a template file.

15.13 Job Management Instructions

The **I**, **O**, and **IO** instructions are used when \$F is run under job management. The **I** instruction is used to check that the correct input volume is mounted, the **O** instruction that the correct output volume is mounted, and the **IO** instruction combines their effects. If in any case the wrong volume is mounted a mount prompt will appear, even when the rest of the information is obtained from the job file. This is because mount prompts bypass job management.

Note that the **O** operation is more tolerant than the **I** operation in that it will accept a volume if it has the correct volume-id **or** if the volume-id is the special name "CLIENT". This second option enables you to pre-allocate a stock of volumes for distribution to customers.

Note also that the **IO** instruction should be used in preference to separate **I** and **O** instructions, since it checks that both volumes are mounted at the same time. If you use separate **I** and **O** instructions the operator may accidentally dismount the first volume when the second mount prompt appears, and this will not be detected.

15.14 Establishing Volumes, Allocating Files

The **SWV** instruction allows you to change the allocation size of a volume without deleting any of the files already present on it, or to change an ordinary volume into a spool or work volume, or vice versa.

When you have keyed the instruction and confirmed the volume-id, you are prompted for the type of volume you wish to set up. Reply **S** to make it a spool volume, **W** to make it a work volume, **N** to make it a normal volume, or **<CR>** to leave its type unchanged. If it is a work or spool volume, you will then be prompted to supply the allocation size to be used. For a work volume, this should be at least as large as the largest file you expect to be created following a request for a file of maximum size. For a spool volume, the allocation size is less critical and typically will be in the range 50 to 250 Kbytes, but should always be at least 2% of the total size (otherwise the directory will become full before all the space is used). An allocation size of 0 is not allowed.

The next two instructions are intended for use by programmers only. Their effect if used by non-programmers on Global applications software files is undefined.

The **ALL** instruction is used to allocate a dummy relative sequential file on the output volume, and to specify its name, size and record length. If a size of 0 is specified the maximum amount of contiguous space available is allocated. The relative sequential file created by **ALL** will be

empty, with the whole of the extent available for expansion. To create the file from a program it should be opened using OPEN OLD, and its records constructed by a sequence of WRITE NEXT operations.

The **SIZ** instruction can be used to change the file size (in bytes) and organization of an existing file on the output volume. The code specified in response to the organization prompt should either be an integer between 0 and 127, or one of the file organization codes defined in table 15.2. Note that files established by **SIZ** will not normally be capable of being processed by normal application programs.

15.15 Patching Monitor Files and Installing Bootstraps

15.15.1 PAM - Patch Monitor

The **PAM** instruction is run under job management during installation to set up system parameters in the monitor file. It can also be used after installation to patch the monitor in order to change the number of memory banks, their size, or the size of the system stack. Note that **PAM** is operationally faster than using \$CUS to reconfigure an installed system, but as it is not run under job management no validation takes place. This means that provided input is numeric it will be accepted, and no check is made that the values supplied are sensible. If in doubt as to what is required key <CR> to leave the situation unchanged.

In common with **PUF** and **INS**, **PAM** is held in the command library as a separate overlay (i.e. \$FX) which is loaded automatically when you select the instruction. A colon prompt appears first, so you can specify the name of the monitor file or key <CR> to accept the default and patch \$MONITOR; the message "NOT FOUND" will appear if the monitor file is not present on the specified output unit. You are then prompted for the following items:

- | | |
|--------------------|--|
| \$CP UNIT: | Key <CR> to accept the current value, or supply a new address for SYSRES, followed optionally by a letter indicating the type of system required: T for a target system, S or X for BACRES (S for BACRES without BACEXA, X for BACRES with BACEXA), or D to produce diagnostics. |
| 100 UNIT: | This is set on installation to the alias which matches the format of the distribution diskettes. Key <CR> to accept the current assignment, or supply a new address so that you can change the unit which is aliased to 100. |
| SYSTEM STACK SIZE: | Key <CR> to accept the current size as it is, or supply a new size in bytes between 0 and (at the very most) 15360. If you make the stack size too large you will be unable to reload Global System Manager. |
| BANK SIZE (K): | Key <CR> to accept the current memory bank size, or supply a new size in Kbytes, between 32 and 60. Any value outside these limits will make it impossible for you to reload Global System Manager. |
| MAX. NO. BANKS: | Key <CR> to leave the number of memory banks unchanged, or supply a new value. You should read the notes in Appendix |

A of the Global Operating Manual on Memory Allocation before doing this.

NO. OF PAGES: Key <CR> to leave the number of memory pages unchanged.

When you have replied to the last of these prompts the message "PATCHED" will appear.

Attempts to change the number of memory pages may make it impossible for you to reload Global System Manager.

15.15.2 PUF - Change Address of Master SYSRES on Network System

The **PUF** instruction is used during installation to specify the master SYSRES address on a network system. Like **PAM**, it is quicker than using \$CUS to modify an installed system.

The monitor file must be on the output unit. In response to the user file unit prompt you should key the address of SYSRES on the new master computer (e.g. Z01). If your LAN does not provide its own computer letter you must key the computer letter as the fourth character.

15.15.3 PIP - Patch Monitor Library Unit Address

The **PIP** instruction is included here for completeness only. **It must not be used.**

15.15.4 INS - Install Bootstrap File

The **INS** instruction is included here for completeness only. **It must not be used.**

15.15.5 CMP - Patch Monitor for Additional Memory Pages

The **CMP** instruction is included here for completeness only. **It must not be used.** This instruction counts the number of memory pages in the P.\$MON, P.\$PAGES, (optional) P.\$BAPGS and (optional) P.\$EXPGS libraries on SYSRES (or SYSIPL) and patches the value into \$MONITOR on SYSRES (or SYSIPL).

Any attempts to change the number of memory pages may make it impossible for you to reload Global System Manager.

15.16 Adding and Amending Volume Descriptions

Global System Manager V8.0 and later allows a 50 character *Volume Description* to be associated with each hard disk domain and subvolume. An area of the disk that was formerly unused is now utilised to hold the table of Volume Descriptions so it is not necessary to re-initialise a volume in order to use this new facility. However, on some disks the area used to hold the Volume Descriptions may not be large enough to contain descriptions for all the subvolumes. If this is the case, attempts to add Volume Descriptions for subvolumes that can't be fitted into the table will be ignored. If a domain is initialised with the V8.0 \$V, a table of sufficient size for all the subvolumes is allocated.

The second line of the report produced by the **LIS** and **PRI** instructions (see section 15.3) for a subunit contains the Volume Description for that subunit. The format of the report produced by the **LIS** and **PRI** commands for a domain contains the Volume Descriptions of each subunit on the domain.

15.16.1 DES - Amend a Volume Description

The **DES** instruction allows the 50 character long Volume Description for a domain or subunit to be amended. The current Volume Description, or spaces if a description is not present, is displayed and may be overwritten by a new 50 character text string. If there is insufficient space in the volume description table for a particular subunit, the following message will appear:

```
NO ROOM FOR DESCRIPTION - RE-INITIALISE DOMAIN
```

This message will appear for subunits with a high unit number on a domain that been initialised using a version of Global System Manager prior to V7.0. The exact number of volume descriptions that will fit on a "pre-V7.0" disk depends on the geometry of the media.

Note that if the **DES** instruction is used under Job Management and there is insufficient space in the volume description table for a particular subunit the above message is suppressed. Instead, to ensure a predictable and consistent Job Management dialogue, the Volume Description prompt is displayed although the reply is subsequently ignored.

15.16.2 DEU - Amend the Volume Descriptions for a domain

The **DEU** instruction can only be used when the selected unit is a domain. The Volume Descriptions for the domain and each subunit are displayed and may be overwritten by new 50 character descriptions.

To abort the instruction at any point key <CTRL A> to return to the instruction prompt.

15.17 Selection Codes

Certain instructions are capable of processing either a single named file, or a number of files specified by means of a selection code. Following such an instruction Global System Manager responds with a colon prompt. You must reply with either the file-id of the single file you wish to process, or the selection code:

```
$66 FILE MAINTENANCE
:xxx :file-id or selection code
```

The following selection codes are available:

- | | |
|---------|---|
| <CR> | select all files, but present the operator with a selection prompt before processing each one; |
| x. | a single character, followed by a full stop. This means select all file-ids with prefix x., but present the operator with a selection prompt before processing each one; |
| .yyyyyy | a full stop, followed by a suffix of up to six characters. This means select all file-ids with the indicated suffix (including the unprefixed file yyyyyy itself), but present the operator with a selection prompt before processing each one; |
| . | a full stop, by itself. This means: select all unprefixed file-ids (those whose second character is not a full stop), but present the operator with a selection prompt before processing each one; |

- <CTRL A> select no files. The file utility will redisplay the instruction prompt. This can be useful if you have keyed an instruction by mistake;
- <CTRL B> select all files for processing but unlike <CR> handle them automatically without presenting the operator with a selection prompt before processing each one;
- <CTRL C> the file utility will output the prompt FROM: to which you must supply a file-id. It will then process this file and all that follow it in the directory automatically, without displaying a selection prompt. If a file-id is encountered which starts with the two characters ++ the copy process is terminated without copying that file. If, however, you terminate your reply with <CTRL C> the file utility will output a TO: prompt. You should supply the name of the last file to be processed; note that in this case processing will not be halted by any intervening '+' files, and that these too will be processed.

In addition <CTRL A>, <CTRL B> and <CTRL C> can be used in place of <CR> to terminate the x., yyyyyy. and . selection strings. In this case the effect is to combine the two functions:

- string*<CTRL A> returns to the instruction prompt (useful if you have keyed the wrong string);
- string*<CTRL B> processes all the files defined by the string automatically, without a selection prompt appearing;
- string*<CTRL C> processes just those files that satisfy the string criteria, starting at the file you identify by your response to the subsequent FROM: prompt. As stated earlier, processing will halt when a '+' file is encountered, unless you terminate your reply with <CTRL C> in which case a TO: prompt will appear.

The second of these combinations is particularly useful. For example, to process all source files on a volume:

```
$66 FILE MAINTENANCE
:xxx :S.<CTRL B>
.....
(source files are processed according to the instruction keyed as xxx)
.....
$66 FILE MAINTENANCE
:
```

When you key <CR>, x., .yyyyyy or . as a selection code, the files are not simply processed automatically, but as each file-id satisfying the selection criterion is met a prompt of the following form is displayed:

```
file-id?:
```

You may reply:

- Y to cause the file to be processed;

- N to skip the processing of this file, and move on to the next one selected;
- <CR> to skip the processing of this file, and move on to the next one selected;
- <CTRL A> to cease processing files and obtain a new instruction prompt;
- <CTRL B> to process the current file, and all that remain in the selection, without further selection prompts appearing;
- <CTRL C> to return to the previous file prompt (this can be repeated).

When the "FILE ALREADY EXISTS - DELETE?" prompt is displayed you may reply with:

- Y to delete the existing file;
- N to skip the processing of this file and leave the old file unchanged;
- <CR> to skip the processing of this file and leave the old file unchanged;
- <CTRL A> to cease processing files and obtain a new instruction prompt;
- <CTRL B> to delete the current existing file, and further existing files that appear during the completion of the current instruction, without further deletion prompts appearing;
- <CTRL C> to skip the processing of the current file, leaving the old file unchanged and to skip the processing of all further files when the "FILE ALREADY EXISTS - DELETE?" prompts appears.

The \$DIR command (see Chapter 11) should be used if more sophisticated wild-card handling is required.

15.18 Options to the SIZE: prompt

The following options are available when specifying the size of the output file during a Copy or Move instruction (see section 5.10):

Reply to	Meaning
<CR>	Copy the output file with the same size as the input file
<i>nnn</i>	Specify the output file size in bytes
<i>nnnK</i>	Specify the output file size in kilobytes
<i>nnnM</i>	Specify the output file size in megabytes
<i>nnnG</i>	Specify the output file size in gigabytes
<i>+nnn</i>	The size of the output file will be <i>nnn</i> bytes larger than the size of the input file
<i>+nnnK</i>	The size of the output file will be <i>nnn</i> kilobytes larger than the size of the input file
<i>+nnnM</i>	The size of the output file will be <i>nnn</i> megabytes larger than the size of the input file

<code>+nnnG</code>	The size of the output file will be <i>nnn</i> gigabytes larger than the size of the input file
<code>-nnn</code>	The size of the output file will be <i>nnn</i> bytes smaller than the size of the input file
<code>-nnnK</code>	The size of the output file will be <i>nnn</i> kilobytes smaller than the size of the input file
<code>-nnnM</code>	The size of the output file will be <i>nnn</i> megabytes smaller than the size of the input file
<code>-nnnG</code>	The size of the output file will be <i>nnn</i> gigabytes smaller than the size of the input file
<code>%nn</code>	The size of the output file will be <i>nn</i> percent larger than the size of the input file

15.19 Report files

When the reports produced by the **PRI**, **PR?**, **PSF** and **PS?** instructions are written to a spool unit, a separate report file, D.\$F*nn* (where *nn* is the User Number) is created for each invocation of the "print" instruction. In contrast, when reports produced by the "print" instructions are written to a volume which is not a spool unit, the report file, D.\$F*nn*, is created by the first "print" instruction. Reports produced by subsequent print instructions are concatenated with the previous reports. The report file is only closed when \$F is left by keying <ESC> to the command prompt.

16. \$FIND - Find Files Utility

Your computer may contain large numbers of subvolumes and files. Under such circumstances it may be difficult to locate a particular file. The \$FIND utility scans all on-line subvolumes to locate a specified file. It can also search for a set of files using selection codes similar to those used by \$F, as well as a special set of selection codes. If you have configured a RAM disk on your computer this is also searched. \$FIND can also be used to scan the domains on a system for a particular subvolume name. \$FIND will also search for print files within spool directories.

16.1 Using \$FIND in Normal Mode

In "normal mode" \$FIND can be used to search for a particular file or subvolume.

16.1.1 Using \$FIND to Find a File

To find a given file, merely key the name of the file to the filename prompt. For example, to search for all files named P.\$CMLB0 use the following dialogue:

```
$305 FIND FILENAME:P.$CMLB0
UNIT 201  SYSRES      P.$CMLB0    01/09/94    PL    1014396
UNIT 202  SYSDEV      P.$CMLB0    10/11/94    PL    133780
```

\$FIND displays the unit number and volume name of subvolumes containing any files found, as well as the names, dates of creation, sizes and types of the files themselves.

The search can be terminated at any time by keying <CTRL G>.

Key <ESC> to the filename prompt to quit \$FIND.

\$FIND can be used to search for a series of files using \$F-style wildcards. For example, to search for all the files beginning with P. use the following dialogue:

```
$305 FIND FILENAME:P.
UNIT 201  SYSRES      P.$CMLB0    01/09/94    PL    1014396
UNIT 201  SYSRES      P.$CMLB1    01/09/94    PL    845678
UNIT 202  SYSDEV      P.$CMLB0    10/11/94    PL    133780
UNIT 204  SAPROG      P.SA        11/11/93    PL    2012346
```

In this example, \$FIND has found four program libraries with a P. prefix.

You can also search for files beginning with a given string of characters by terminating the string with <CTRL B>. For example, if you were only interested in finding command libraries you might use the following dialogue:

```
$305 FIND FILENAME:P.$<CTRL B>
UNIT 201  SYSRES      P.$CMLB0    01/09/94    PL    1014396
UNIT 201  SYSRES      P.$CMLB1    01/09/94    PL    845678
UNIT 202  SYSDEV      P.$CMLB0    10/11/94    PL    133780
```

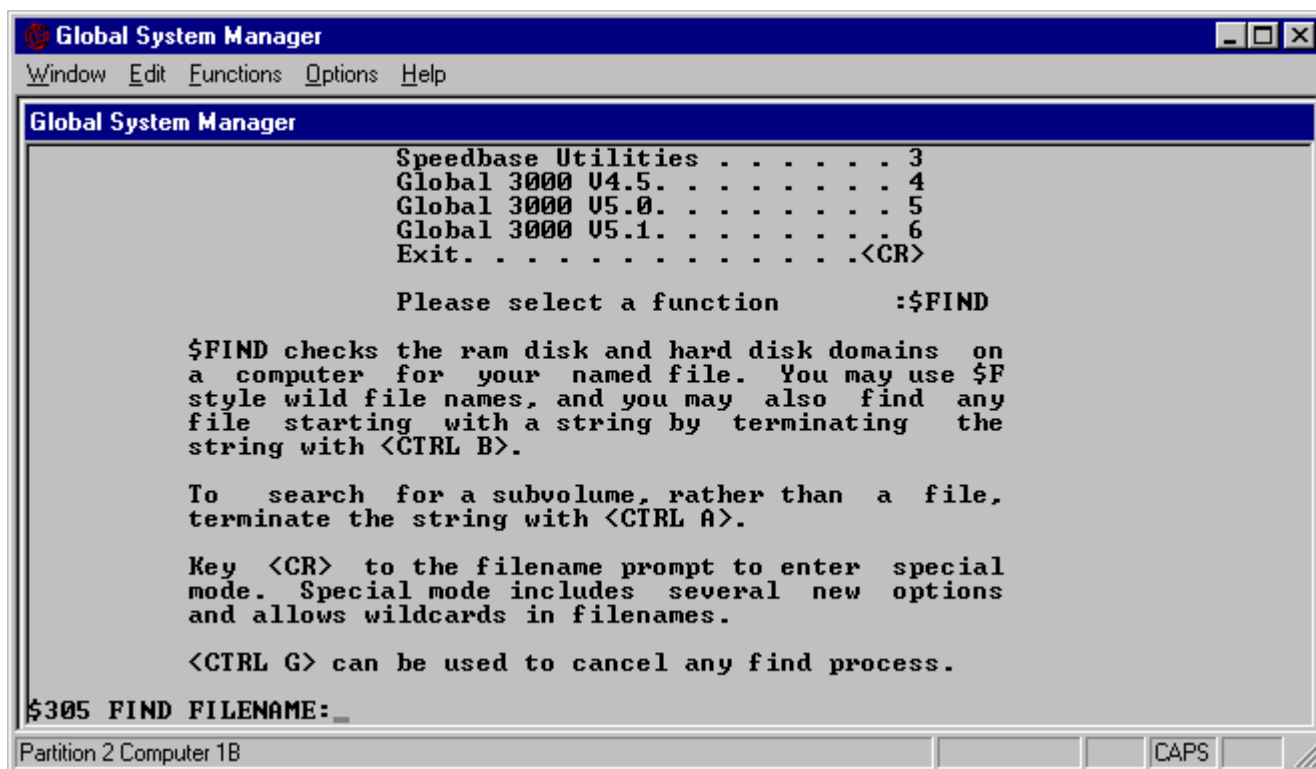


Figure 16.1 - \$FIND Initial Screen

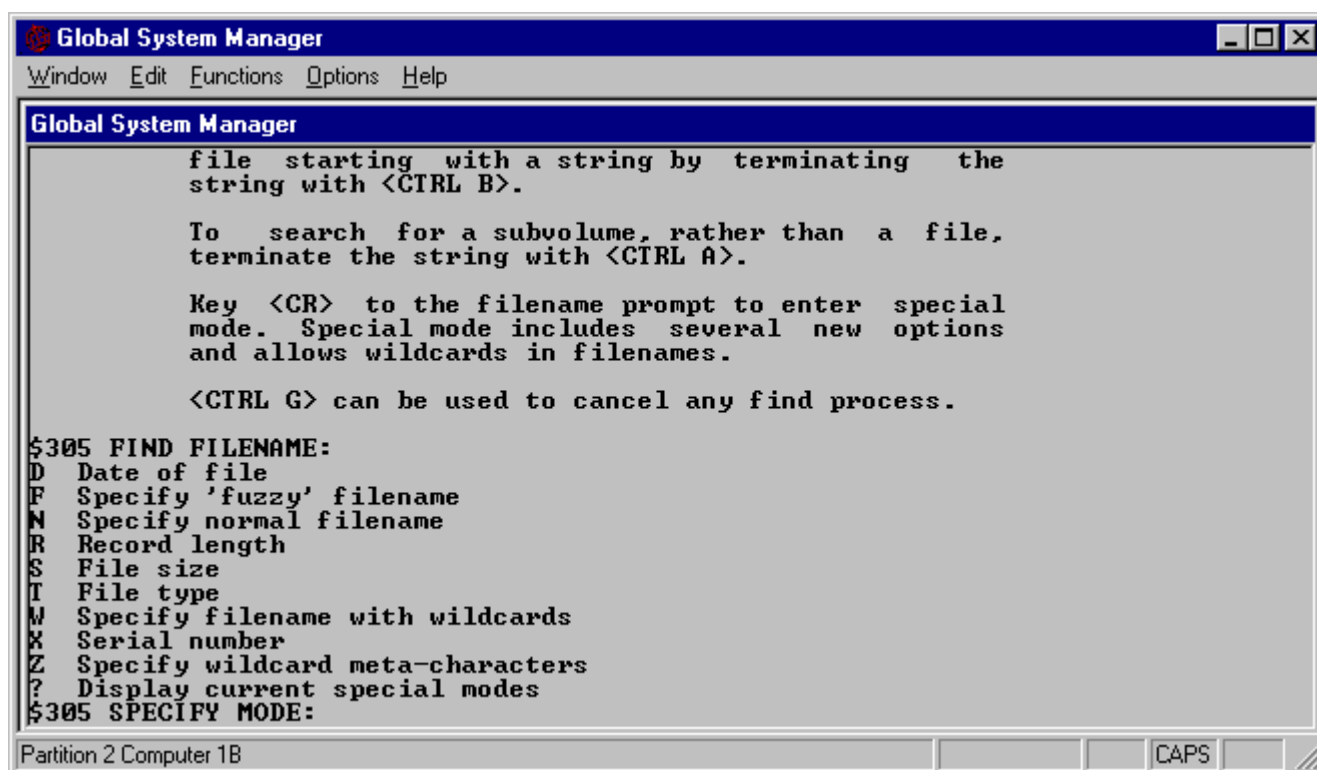


Figure 16.2 - \$FIND in Special Mode

16.1.2 The Computer/System Prompt

On a network Global System Manager configuration (or a Global System Manager (Unix) configuration), \$FIND prompts for a computer (or file-server system) letter:

ON COMPUTER (n) :

or:

ON SYSTEM (n) :

This prompt allows you to specify which computer or system you wish to scan. The default is the local computer (or system) if your computer (or system) is a file-server (i.e. the node-id is between "A" and "Z"); or the node-id of the master computer (or system) if your computer (or system) is not a file-server.

To search all computers on the network (or all Global System Manager (Unix) file-server systems), key <CTRL B> to the above prompt. To search the local computer only key <CTRL C>. The <CTRL C> option is particularly useful if the local computer is a non-file server. Furthermore, the <CTRL C> option allows you to specify the unit number of the first volume to be searched thus over-riding the default of 105.

To return to the file name prompt key <CTRL A>. To summarise:

ON COMPUTER (A) :<CR>	(search file-server A)
ON COMPUTER (A) :B	(search file-server B)
ON COMPUTER (A) :<CTRL B>	(search all file-servers)
ON COMPUTER (A) :<CTRL C>	(search local computer)
SPECIFY START UNIT:<CR>	(start search from 105)
ON COMPUTER (A) :<CTRL C>	(search local computer)
SPECIFY START UNIT:267	(start search from 267)
ON COMPUTER (A) :X<CTRL C>	(search file-server X)
SPECIFY START UNIT:250	(start search from X50)
ON COMPUTER (A) :<CTRL A>	(return to filename prompt)

16.1.3 Using \$FIND to Find a Subvolume

To find a given volume, terminate the reply to the filename prompt with a <CTRL A>. For example, to search for all volumes named SYSRES on a network use the following dialogue:

```
$305 FIND FILENAME:SYSRES<CTRL A>
ON COMPUTER (A) :<CTRL B>
UNIT A01 SYSRES 1069568
UNIT B10 SYSRES 1856488
UNIT X01 SYSRES 7654328
```

\$FIND displays the unit number, volume name and datasize of the subvolumes found.

The search can be terminated at any time by keying <CTRL G>.

You can also search for volumes beginning with a given string of characters by terminating the string with <CTRL C>. For example, if you were only interested in finding volumes that start with the string "SL" you might use the following dialogue:

```
$305 FIND FILENAME:SL<CTRL C>
ON COMPUTER (A) :<CTRL B>
```

```
UNIT A10 SLPROG 1069568
UNIT A11 SLDATA 4567868
UNIT A99 SLWORK 8069868
UNIT B10 SLDATA 3656488
UNIT X01 SLDATA 7654328
```

16.2 Using \$FIND in Special Mode

In "special mode" \$FIND can be used to search for a wide range of criteria. To enter special mode key <CR> to the filename prompt.

```
$305 FIND FILENAME:<CR>
      (help text displayed by $FIND)
$305 SPECIFY MODE:
```

The valid replies to the "mode" prompt are summarised in table 16.1.

The mode prompt is repeated until the reply is <CR> allowing several selection criteria to be combined (see section 16.2.11).

For all the replies in table 16.1 marked with a * the terminator, <CR>, <CTRL A>, <CTRL B> or <CTRL C> is significant. \$FIND maintains four separate tables of special modes - the actual table selected depends on the terminator as described in table 16.2. This feature allows very complex search conditions to be specified and makes \$FIND a very powerful tool. Complex search criteria are fully described in section 16.2.11.

16.2.1 Search for Date Range

A reply of D to the mode prompt allows you specify a date range as part of the search criteria. The following example dialogue includes a search for all files with a creation date in October 1994 in the first table of search options:

```
$305 SPECIFY MODE:D<CR>
SPECIFY START DATE (02/11/1994):1/10/94
SPECIFY END DATE (01/10/1994):31/10/94
```

The default start-date is either the current system date or the previous start-date specified for the selected table entry. A special value of "." for the start-date disables start-date checking (i.e. a reply of "." is equivalent to earliest possible Global System Manager date).

The default end-date is either the same as the start-date (if the start-date is a true date) or the current system date (if the reply to the start-date prompt was "."). A special value of "." for the end-date disables end-date checking (i.e. a reply of "." is equivalent to highest possible Global System Manager date).

Reply	Action
<CR>	If no special modes (see below) have been selected then return to normal mode; or if one, or more, special modes have been specified commence the search (single computer configuration) or prompt for the computer/system (see section 16.1.2).
<CTRL A>	Abort special mode and return to normal mode (even if one, or more, special modes have been selected)
<CTRL B>	Restore the special mode options used for the previous search. This feature can only be used if <CTRL B> is the first reply to the "mode" prompt.
<CTRL C>	Redisplay the \$FIND special mode help text.

D *	Search for files within a date range (see section 16.2.1).
F *	Search for files, specifying a "fuzzy filename" (see section 16.2.2).
N *	Search for files, specifying a "normal filename" (see section 16.2.3). This option allows a "normal mode" compatibility option within "special mode".
R *	Search for files with a specific record length (see section 16.2.4).
S *	Search for files within a specified size range (see section 16.2.5).
T *	Search for files of a particular file-type (see section 16.2.6).
W *	Search for files allowing wildcards in the filename string (see section 16.2.7).
X *	Search for files with a particular serial number (see section 16.2.8).
Z *	Change the wildcard meta-characters (see section 16.2.9).
?	Display the current special modes that have been enabled (see section 16.2.10).

Table 16.1 - Valid Replies to the \$FIND Special Mode Prompt

Terminator	Action
<CR>	Prompt for parameter(s) and save the reply in the 1st "special mode table"
<CTRL A>	Prompt for parameter(s) and save the reply in the 2nd "special mode table"
<CTRL B>	Prompt for parameter(s) and save the reply in the 3rd "special mode table"
<CTRL C>	Prompt for parameter(s) and save the reply in the 4th "special mode table"

Table 16.2 - Summary of \$FIND Special Mode Terminators

Mode	Description
A	A file is deemed to match the target file if its filename is an A nagram of the target filename.
C	A file is deemed to match the target file if the character C hecksum of its filename is the same as the character checksum of the target filename.
D	A file is deemed to match the target file if its filename D iffers from that of the target filename by a specified number of characters.
S	A file is deemed to match the target file if its filename is the S ame as that of the target filename by a specified number of characters.

Table 16.3 - Summary of \$FIND Fuzzy Modes

For example:

```
SPECIFY START DATE (02/11/1994):. NO START DATE CHECKING
SPECIFY END DATE (02/11/1994):. NO END DATE CHECKING
```

If the reply to the "end-date" prompt is numeric then it is used as the number of days since the start-date. For example:

```
SPECIFY START DATE (02/11/1994):<CR>
SPECIFY END DATE (02/11/1994):1 DATE RANGE FROM 02/11/1994 TO 03/11/1994
```

If the reply to the "start-date" prompt is <CTRL A> or <CTRL C>, the "D option" is aborted and the table entry is not updated. If the reply is <CTRL B> the table entry is marked as "not in use". If the reply is <CR> the default entry is used. If a new start-date is specified it is not added to the table until a valid end-date has been specified.

If the reply to the "end-date" prompt is <CTRL C> the "D option" is aborted and the table entry is not updated. If the reply is <CTRL A> the "start-date" prompt is redisplayed. If the reply is <CR> the default value is used and the table entry is marked as "in use". If a new end-date is specified the table entry is updated and marked as "in use".

16.2.2 Search for Fuzzy Files

A reply of F to the mode prompt allows you specify a "fuzzy filename" as part of the search criteria. This option allows you to select any one of the following "fuzzy modes" as described in table 16.3.

The following example dialogue includes a search for all filenames that are anagrammatic with P.\$CMLB0:

```
$305 SPECIFY MODE:F
SPECIFY FUZZY FILENAME ( ):P.$CMLB0
SPECIFY FUZZY MODE [Anag, Chksum, Diffs, Same] (A):A
```

The default filename is either <NULL> or the previous filename specified for the selected table entry. The length of the fuzzy filename is always 8 characters (i.e. trailing spaces are significant).

If the reply to the "filename" prompt is <CTRL A> or <CTRL C>, the "F option" is aborted and the table entry is not updated. If the reply to the "filename prompt" is <CTRL B> the table entry is marked as "not in use". If the reply to the "filename" prompt is <CR> the default value is used and the table entry is marked as "in use". If a new filename is specified the table entry is updated and marked as "in use". If the filename consists of 8 <SPACE> characters the "N option" is aborted, and the table entry is not updated (i.e. the initial default is not meaningful).

The following example dialogue includes a search for all filenames that have the same character checksum as P.\$CMLB0:

```
$305 SPECIFY MODE:F
SPECIFY FUZZY FILENAME ( ):P.$CMLB0
SPECIFY FUZZY MODE [Anag, Chksum, Diffs, Same] (A):C
```

If the reply to the "mode" prompt is <CTRL C>, the "F option" is aborted and the table entry is not updated. If the reply is <CTRL A> the "fuzzy filename" prompt is redisplayed.

If the fuzzy mode is "D" or "S" you are prompted both the number of significant characters and the condition to be used when testing the number of significant characters as shown in table 16.4.

The following example dialogue includes a search for all filenames that differ from P.\$CMLB0 by 1 character exactly:

```
$305 SPECIFY MODE:F
SPECIFY FUZZY FILENAME ( ):P.$CMLB0
SPECIFY FUZZY MODE [Anag, Chksum, Diffs, Same] (A):D
SPECIFY CHARACTER COUNT (2):1
SPECIFY COMPARE MODE [<, {, =, }, >] (=):=
```

If the reply to the "character count" prompt is <CTRL C>, the "F option" is aborted and the table entry is not updated. If the reply is <CTRL A> the "fuzzy mode" prompt is redisplayed.

If the reply to the "compare mode" prompt is <CTRL C>, the "F option" is aborted and the table entry is not updated. If the reply is <CTRL A> the "character count" prompt is redisplayed.

The following example includes a search for all filenames that differ from P.\$CMLB0 by 2 characters, or less:

```
$305 SPECIFY MODE:F
SPECIFY FUZZY FILENAME ( ):P.$CMLB0
SPECIFY FUZZY MODE [Anag, Chksum, Diffs, Same] (A):D
SPECIFY CHARACTER COUNT (2):2
SPECIFY COMPARE MODE [<,{,=,},>] (=):{
```

The following example includes a search for all filenames that include 4, or more, characters that match P.\$CMLB0:

```
$305 SPECIFY MODE:F
SPECIFY FUZZY FILENAME ( ):P.$CMLB0
SPECIFY FUZZY MODE [Anag, Chksum, Diffs, Same] (A):S
SPECIFY CHARACTER COUNT (2):4
SPECIFY COMPARE MODE [<,{,=,},>] (=):}
```

Mode	Description
D =	A file is deemed to match the target file if its filename differs from that of the target file by exactly the specified number of characters.
D >	File is deemed to match the target file if its filename differs from that of the target file by more than the specified number of characters.
D }	A file is deemed to match the target file if its filename differs from that of the target file by no less than the specified number of characters.
D <	A file is deemed to match the target file if its filename differs from that of the target file by less than the specified number of characters.
D {	A file is deemed to match the target file if its filename differs from that of the target file by no more than the specified number of characters.
S =	A file is deemed to match the target file if its filename matches that of the target file by exactly the specified number of characters.
S >	A file is deemed to match the target file if its filename matches that of the target file by more than the specified number of characters.
S }	A file is deemed to match the target file if its filename matches that of the target file by no less than the specified number of characters.
S <	A file is deemed to match the target file if its filename matches that of the target file by less than the specified number of characters.
S {	A file is deemed to match the target file if its filename matches that of the target file by no more than the specified number of characters.

Table 16.4 - Summary of \$FIND Fuzzy Modes "D" and "S"

16.2.3 Search for Normal Files

A reply of N to the mode prompt allows you specify a normal filename as part of the search criteria. The following example dialogue includes a search for the filename P.\$CMLB0 in the second table of search options:

```
$305 SPECIFY MODE:N<CTRL A>
SPECIFY RECORD LENGTH ( ):P.$CMLB0
```

The default filename is either <NULL> or the previous filename specified for the selected table entry. If the reply to the "filename" prompt is <CTRL A> or <CTRL C>, the "N option" is aborted and the table entry is not updated. If the reply is <CTRL B> the table entry is marked as "not in use". If the reply is <CR> the default value is used and the table entry is marked as "in use". If a new filename is specified the table entry is updated and marked as "in use". If the filename consists of 8 <SPACE> characters, the "N option" is aborted, and the table entry is not updated.

As for "standard mode" filenames (see section 16.1.1), \$F-style wildcards can be used (e.g. a reply of "P." indicates all files that start with the P. prefix). Similarly, you can also specify those files that begin with a given string of characters by terminating the string with <CTRL B>.

16.2.4 Search for Record Length

A reply of R to the mode prompt allows you specify a record length as part of the search criteria. The following example dialogue includes a search for record length 100 in the fourth table of search options:

```
$305 SPECIFY MODE:R<CTRL C>
SPECIFY RECORD LENGTH (64):100
```

The default record length is either 64 or the previous record length specified for the selected table entry.

If the reply to the "record length" prompt is <CTRL A> or <CTRL C>, the "R option" is aborted and the table entry is not updated. If the reply is <CTRL B> the table entry is marked as "not in use". If the reply is <CR> the default value is used and the table entry is marked as "in use". If a new record length is specified the table entry is updated and marked as "in use".

The record length is only meaningful, and thus is only tested, for RS and IS files.

16.2.5 Search for Size Range

A reply of S to the mode prompt allows you specify a size range as part of the search criteria. The following example dialogue includes a search for all files with a size between 16 and 64 bytes, inclusive:

```
$305 SPECIFY MODE:S<CR>
SPECIFY MINIMUM FILESIZE (1 ):16
SPECIFY MAXIMUM FILESIZE (16 ):64
```

The file size can be specified in kilobytes (by suffixing the numeric reply with a "K"), in megabytes (by suffixing the numeric reply with a "M") or gigabytes (by suffixing the numeric reply with a "G"). For example, to search for all files between 1 and 2 megabytes:

```
$305 SPECIFY MODE:S<CR>
SPECIFY MINIMUM FILESIZE (16 ):1M
SPECIFY MAXIMUM FILESIZE (1M ):2M
```

If the reply to the "minimum filesize" prompt is <CTRL A> or <CTRL C>, the "S option" is aborted and the table entry is not updated. If the reply is <CTRL B> the table entry is marked as "not in use". If the reply is <CR> the default entry is used. If a new minimum filesize is specified the table entry is updated.

If the reply to the "maximum filesize" prompt is <CTRL C> the "S option" is aborted and the table entry is not updated. If the reply is <CTRL A> the "minimum filesize" prompt is

redisplayed. If the reply is <CR> the default value is used and the table entry is marked as "in use". If a new maximum filesize is specified the table entry is updated and marked as "in use".

16.2.6 Search for File Type

A reply of T to the mode prompt allows you specify a file-type code (see table 15.2) as part of the search criteria. The following example dialogue includes a search for file-type "TF" (i.e. text files) in the first table of search options:

```
$305 SPECIFY MODE:T<CR>
SPECIFY FILETYPE (PL):TF
```

The default file-type code is either PL or the previous file-type code specified for the selected table entry.

If the reply to the "file-type" prompt is <CTRL A> or <CTRL C>, the "T option" is aborted and the table entry is not updated. If the reply is <CTRL B> the table entry is marked as "not in use". If the reply is <CR> the default value is used and the table entry is marked as "in use". If a new file-type code is specified the table entry is updated and marked as "in use".

16.2.7 Search for Wildcard Files

A reply of W to the mode prompt allows you specify a filename that includes wildcard characters as part of the search criteria. Both single-character wildcards and multiple-character wildcards may be used. There are no restrictions on the use of the single character wildcard (which is, by default, a "?" character but which may be changed by the "Z" command - see section 16.2.9): Any number of single-character wildcards (up to 8) are allowed in the template filename. Some restrictions apply to the use of the multiple character wildcard (which is, by default, a "*" character but which may be changed by the "Z" command - see section 16.2.9): Only one single-character wildcard in the template filename. Any combinations of single-character and multiple-character wildcards are allowed.

The following example dialogue includes a search for all filenames that end with the string "ABC":

```
$305 SPECIFY MODE:W
SPECIFY FUZZY FILENAME ( ):ABC
```

The default filename is either <NULL> or the previous filename specified for the selected table entry. The number of characters actually keyed to the filename prompt is used as the length of the filename (i.e. trailing spaces are not significant).

If the reply to the "filename" prompt is <CTRL A> or <CTRL C>, the "W option" is aborted and the table entry is not updated. If the reply to the "filename prompt" is <CTRL B> the table entry is marked as "not in use". If the reply to the "filename" prompt is <CR> the default value is used and the table entry is marked as "in use". If a new filename is specified the table entry is updated and marked as "in use". If the filename consists of 8 <SPACE> characters the "W option" is aborted, and the table entry is not updated (i.e. the initial default is not meaningful).

16.2.8 Search for Serial Number

A reply of X to the mode prompt allows you specify a serial number as part of the search criteria. The following example dialogue includes a search for serial number 1234567 in the third table of search options:

```
$305 SPECIFY MODE:X<CTRL B>
SPECIFY SERIAL NUMBER (1):1234567
```

The default serial number is either the serial number for your system (as displayed by \$S) or the previous serial number specified for the selected table entry.

If the reply to the "serial number" prompt is <CTRL A> or <CTRL C>, the "X option" is aborted and the table entry is not updated. If the reply is <CTRL B> the table entry is marked as "not in use". If the reply is <CR> the default value is used and the table entry is marked as "in use". If a new serial number is specified the table entry is updated and marked as "in use".

The "X option" is unique because it allows a logical-NOT option in the search criteria. To indicate that the serial criteria is to be "all serialised files WITHOUT a particular serial number", instead of "all serialised files WITH a particular serial number", terminate the reply to the "serial number" prompt with <CTRL D>. If the terminator is not <CTRL D>, "normal logic" is assumed.

16.2.9 Change Wildcard Meta Characters

A reply of Z to the mode prompt allows you to change the characters used for multiple-character and single-character substitution ("*" and "?" by default). It may be necessary to change the wildcard characters if the default wildcards occur in the names of the files you want to select using the W option. The following example dialogue changes the single-character wildcard character to "!" and the multiple-character wildcard character to "+" for the 2nd table of special mode parameters:

```
$305 SPECIFY MODE:Z<CTRL A>
SPECIFY SINGLE CHAR WILDCARD (?):!
SPECIFY MULTIPLE CHAR WILDCARD (*):+
```

If the reply to the "single character wildcard" prompt is <CTRL A> or <CTRL C>, the "Z option" is aborted. If the terminator to the "multiple character wildcard" prompt is <CTRL C> the "Z option" is aborted. If the reply to the "multiple character wildcard" prompt is <CTRL A> the "single character wildcard" prompt is redisplayed. See section 16.2.7 for a description of how the wildcard characters are used.

The Z option is unique in that it includes an option that allows the entries in all 4 parameter tables to be updated. To enable this option the terminator to the reply of Z to the "mode prompt" must be <CTRL D>. For example, to change the single-character wildcard character to "?" and the multiple-character wildcard character to "*" for all 4 tables of special mode parameters:

```
$305 SPECIFY MODE:Z<CTRL D>
SPECIFY SINGLE CHAR WILDCARD (?):?
SPECIFY MULTIPLE CHAR WILDCARD (*):*
```

16.2.10 Display the Current Special Modes

A reply of ? to the "specify mode" command displays the currently enabled modes (in addition to the help text normally displayed when <CTRL C> is keyed to the prompt). For each special mode (i.e. D, F, N, R, S, T, W, X, Z) the "in-use" entries in each of the 4 tables are displayed. For example, after specifying the following complex set of search conditions:

```
$305 SPECIFY MODE:D
SPECIFY START DATE (12/11/1994):11.11.94
SPECIFY END DATE (12/11/1994):13.11.94
$305 SPECIFY MODE:F
```

Chapter 16 - Find Files Utility (\$FIND)

```
SPECIFY FUZZY FILENAME ( ):TEST1
SPECIFY FUZZY MODE [Anag, Chksum, Diffs, Same] (A):A
$305 SPECIFY MODE:N
SPECIFY FILENAME ( ):TEST2
$305 SPECIFY MODE:N<CTRL C>
SPECIFY FILENAME ( ):TEST3<CTRL B>
$305 SPECIFY MODE:R
SPECIFY RECORD LENGTH ( 64 ):77
$305 SPECIFY MODE:S
SPECIFY MINIMUM FILESIZE ( 1 ):1K
SPECIFY MAXIMUM FILESIZE (1K ):2K
$305 SPECIFY MODE:T
SPECIFY FILETYPE (PL):SY
$305 SPECIFY MODE:W
SPECIFY FILENAME ( ):*XYZ?A
$305 SPECIFY MODE:X
SPECIFY SERIAL NUMBER ( 1 ):1234567
$305 SPECIFY MODE:X<CTRL A>
SPECIFY SERIAL NUMBER ( 1 ):9990001
$305 SPECIFY MODE:X<CTRL B>
SPECIFY SERIAL NUMBER ( 1 ):9010001<CTRL D>
$305 SPECIFY MODE:X<CTRL C>
SPECIFY SERIAL NUMBER ( 1 ):9659999
```

the table displayed by the ? command will look like:

	<CR>	<CTRL A>	<CTRL B>	<CTRL C>
D	12/11/94-13/11/94			
F	TEST1	8 A		
N	TEST2	8		TEST3
5				
R	0077			
S	00001024-00002048			
T	SY			
W	*XYZ?A	6		
X	1234567	9990001	9010001 !	9659999
Z	? *	? *	? *	? *

Note that the number of significant characters in each of the "normal", "fuzzy" (always 8) and "wildcard" filenames is displayed immediately after the filename.

Note also that the "!" displayed after a serial number indicates the "logical NOT" option (see section 16.2.8).

16.2.11 Complex Search Conditions

If more than 1 special mode has been selected, \$FIND issues the following prompt before the search commences:

```
Combine modes by ANDing or ORing (O):
```

You must reply either A to indicate that all the search criteria must be met for a file to match (i.e. a logical AND of all the search criteria) or O to indicate that any one of the search criteria must be met for a file to match (i.e. a logical OR of all the search criteria). A reply of <CTRL A> will return to the "specify mode" prompt.

The following dialogue will initiate a search for all text files (i.e. type TF) with a size less than 1000 bytes:

Chapter 16 - Find Files Utility (\$FIND)

```
$305 SPECIFY MODE:T
SPECIFY FILETYPE (SY):TF
$305 SPECIFY MODE:S
SPECIFY MINIMUM FILESIZE (1K ):1
SPECIFY MAXIMUM FILESIZE (2K ):999
$305 SPECIFY MODE:<CR>
Combine modes by ANDing or ORing (O):A
```

The following dialogue will initiate a search for all text files (i.e. type TF) or any file with a size less than 1000 bytes:

```
$305 SPECIFY MODE:T
SPECIFY FILETYPE (SY):TF
$305 SPECIFY MODE:S
SPECIFY MINIMUM FILESIZE (1K ):1
SPECIFY MAXIMUM FILESIZE (2K ):999
$305 SPECIFY MODE:<CR>
Combine modes by ANDing or ORing (O):O
```

The multiple tables within \$FIND make very complex search criteria possible. The following dialogue will initiate a search for all files with a size of 100, 200, 300 or 400 bytes:

```
$305 SPECIFY MODE:S
SPECIFY MINIMUM FILESIZE (1K ):100
SPECIFY MAXIMUM FILESIZE (2K ):100
$305 SPECIFY MODE:S<CTRL A>
SPECIFY MINIMUM FILESIZE (1K ):200
SPECIFY MAXIMUM FILESIZE (2K ):200
$305 SPECIFY MODE:S<CTRL B>
SPECIFY MINIMUM FILESIZE (1K ):300
SPECIFY MAXIMUM FILESIZE (2K ):300
$305 SPECIFY MODE:S<CTRL C>
SPECIFY MINIMUM FILESIZE (1K ):400
SPECIFY MAXIMUM FILESIZE (2K ):400
$305 SPECIFY MODE:<CR>
Combine modes by ANDing or ORing (O):O
```

Note that \$FIND does not check complex search criteria for reasonableness so that it is possible to specify nonsensical search criteria that are impossible to fulfill (for example it is possible to search for a file of size 100 and 200 and 300 and 400 bytes!).

Note also that \$FIND does not allow a combination of AND and OR relations in complex search criteria. For example, it is NOT possible to search for all text files with sizes of either 100 or 200 bytes as this would require 3 conditions (i.e. type TF; size between 100 and 100 bytes; size between 200 and 200 bytes) combined with and AND and an OR relationship.

17. \$GROUP - Group File Maintenance

The \$GROUP group file maintenance utility allows you to group together sets of operator-id's. For example, you may wish to identify a group of operators belonging to a certain work team (e.g. Sales, Development) or a group of operators in a specific geographical location (e.g. 1st floor, 2nd floor etc.). Other Global System Manager utilities (e.g. \$MAIL, \$M, \$STATUS) include options to operate on all the operators defined in a particular group (e.g. \$MAIL can be used to send a mail message to all the operators in a specified group).

17.1 \$GROUP Dialogue

On entering \$GROUP the initial screen (see figure 17.1) contains the current group list with the following main baseline prompt:

Key Amend, Create, Delete, Print, <CR> for next, <ESC> to exit:

17.2 Amending an Existing Group

Key A to amend an existing group. You are prompted for the name of the group to amend. Once an existing group has been specified you are given the chance to amend the group description. For example, you might wish to amend a group of sales staff as follows:

```
Key Amend, Create, Delete, Print, <CR> for next, <ESC> to exit:A
Key group-id to amend, <CR> to exit:SALES
Description:All the Sales Staff
```

The group member screen is displayed (see figure 17.2). This screen displays the group name and description together with a list of all the operators who are members of the group. Both the 4-character operator-id and the 35-character Long Operator Name are displayed. The following baseline prompt is issued:

Key Add, Delete, Print, <CR> for next, <ESC> to exit:

17.2.1 Add Operator to Group

Key A to add a new operator to the group. You are only prompted for the operator-id. The Long Operator Name is obtained automatically from the \$\$OPID file (see Chapter 26). You can continue adding operators to the group until you key <CR> to the "operator-id" prompt. A group can contain a maximum of 99 members.

17.2.2 Delete Operator from Group

Key D to delete a member from a group. You will be prompted for the operator-id of the member you wish to delete. You can continue deleting operators from the group until you key <CR> to the "operator-id" prompt.

17.2.3 Print Details of Group

Key P to print the details of the currently selected group to logical unit \$PR (creating a report file D.\$GROUP).

17.2.4 Next and Back

If there is insufficient space to display all the groups on a single screen key <CR> to the baseline prompt to display the next page of members.

If any page other than the first in the operator list is displayed the baseline prompt described above is replaced by:

Key Add, Back, Delete, Print, <CR> for next, <ESC> to exit:

Key B to redisplay the first page in the operator list.

17.2.5 Exit

Key <ESC> to the baseline prompt to return to the initial group list screen.

17.3 Creating a New Group

Key C to the main baseline prompt to create a new group. You are then prompted for the name and description of the group to create. For example, to create a group of sales staff:

```
Key Amend, Create, Delete, Print, <CR> for next, <ESC> to exit:C
Key new group-id, <CR> to exit:SALES
Description:All the Sales Staff
```

The group member screen will then be displayed. As with the amend function (see section 17.2) you can add or delete operators from the group or print the group details.

17.4 Deleting a Group

Key D to the main baseline prompt to delete an existing group. For example, to delete the SALES group:

```
Key Amend, Create, Delete, Print, <CR> for next, <ESC> to exit:D
Key group-id to delete, <CR> to exit:SALES
Delete All the Sales Staff ?:Y
```

If you key N to the delete confirmation prompt the group will not be deleted and you will be returned to the original baseline prompt.

17.5 Printing Group File Details

Key P to the main baseline prompt to print the group file details. For example:

```
Key Amend, Create, Delete, Print, <CR> for next, <ESC> to exit:P
```

The group file details will be printed to unit \$PR (creating a report file D.\$GROUP).

17.6 Next Page

If there are more groups defined than can be displayed on a single screen of the group list you can key <CR> to the baseline prompt to view the next page of the group list.

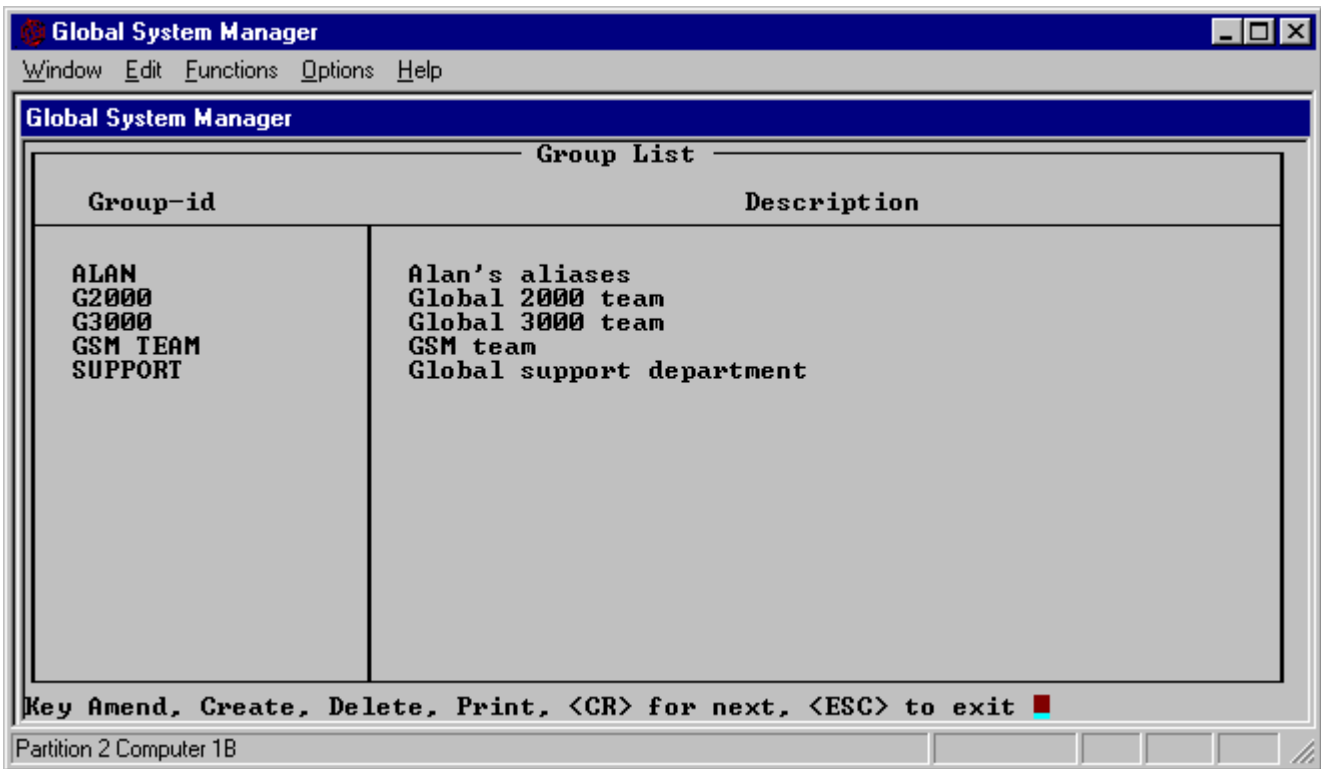


Figure 17.1 - \$GROUP Initial Screen

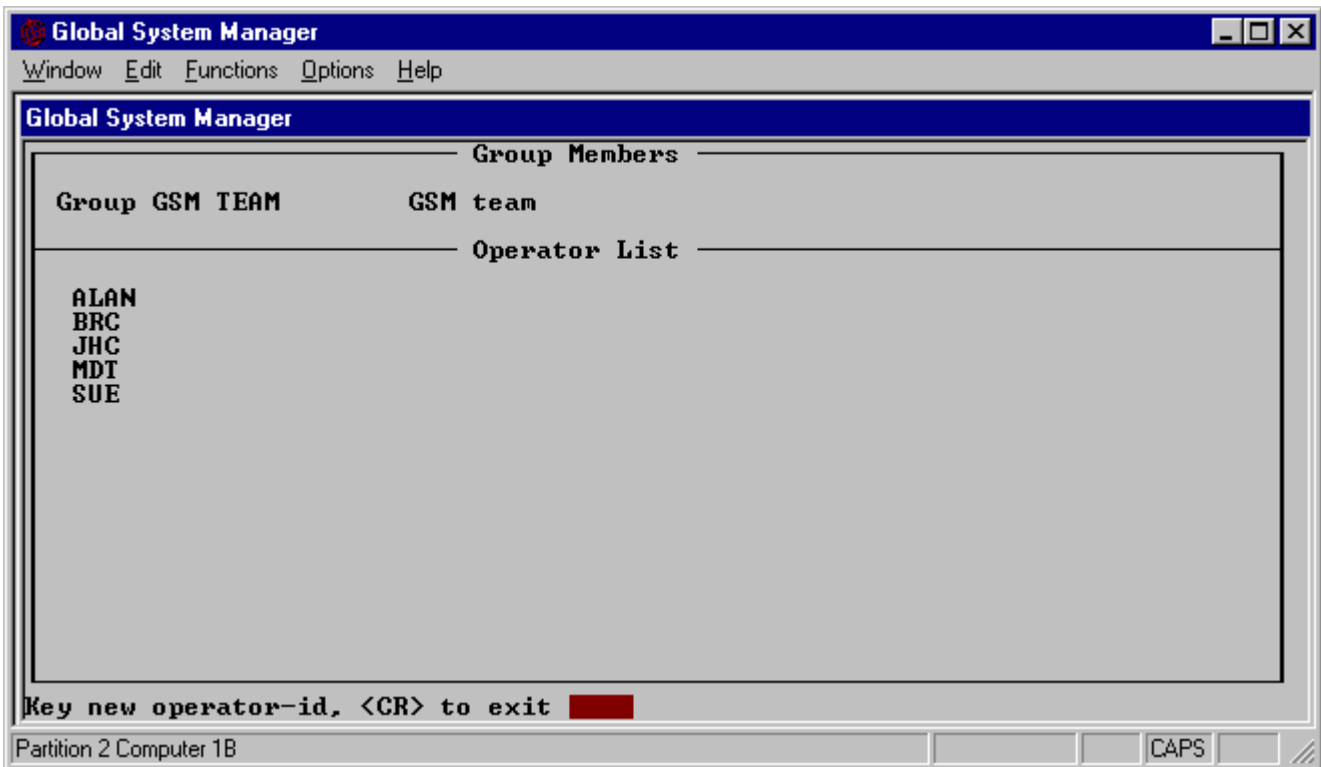


Figure 17.2 - \$GROUP Group Member Screen

17.7 Exiting \$GROUP

When you have completed the changes to the group file, key <ESC> to the main baseline prompt to exit \$GROUP.

18. \$INSOFT - Install Software

\$INSOFT is documented in Chapter 2 of the Global Operating Manual.

19. \$INSPECT - Inspect Text and Print Files

The \$INSPECT command allows you to examine any text or print file. It can also be used to extract parts of such files, concatenate them, produce a re-arrangement of the text within a file and convert text format to relative sequential print format, or vice versa. \$INSPECT can also be employed to print selected parts of any text or print file.

19.1 The Inspect File Prompt

When you run \$INSPECT it first displays the inspect file prompt, so that you can specify the file-id and unit-id of the file you wish to process. For example:

```
$48 INSPECT:D.SALES UNIT:210
```

indicates that the operator wishes to inspect the report file D.SALES on unit 210.

19.1.1 Inspecting Files on a Spool Unit

If the unit you specify is a spool unit, \$INSPECT will scan the volume to determine if more than one copy of the report file is present on the spool unit. If more than one copy is present on the spool unit, \$INSPECT will prompt you for version of the file. For example, assuming 3 copies of report L.TEST exist on spool unit A08 (as spool files 001ALAN3, 002ALAN3, 003ALAN3), an attempt to inspect L.TEST on A08 will result in the following dialogue:

```
$48 INSPECT:L.TEST UNIT:A08
003 ALAN/3 OF 23/08/95?:<CR>
002 ALAN/3 OF 23/08/95?:<CR>
001 ALAN/3 OF 23/08/95?:<CR>
```

Note the alternative versions are displayed in reverse order of sequence number so that, in general, the most recent version of the print report will be displayed first.

If \$INSPECT is running under job management, the latest version of a spooled file is always used, without the additional prompt, to ensure that \$INSPECT maintains a consistent dialogue regardless of the type of unit you specify.

19.2 To Quit

You may quit and return to the menu or GSM READY: prompt by simply keying <ESC> in response to any prompt.

19.3 Using Wide Screen Mode

\$INSPECT will always make use of the maximum logical width of screen possible, so that you may, where appropriate, use <SYSREQ> L and R to inspect the listing, rather than needing to use the \$LS instruction (see below).

The "W" command (see section 19.11, below) may be used to switch the screen into compressed, wide mode provided the screen supports this mode. The 8.1 version of \$INSPECT does not present you with the option to enter wide mode before displaying the first page of the file (earlier versions of \$INSPECT allowed wide mode to be selected before displaying the first page of the report).

Instruction	Description
E	Exit to process next file (see 19.10)
F	Find a line containing a specified string and display a page with the line at the top of the page (see 19.5)
G	Get a line starting with a specified string and display a page with the line in the middle of the page (see 19.6)
P	Page (i.e. move forward so that the bottom line of the current page becomes the top line of the new page)
-P	Page backwards (i.e. move backwards so that the top line of the current page becomes the bottom line of the new page)
WS & WE	Used to print or write out selected parts of the file under inspection (see 19.8 & 19.9)
\$LS	Prompts you for the starting column of the part of the line to be displayed (see 19.7)
<i>n</i>	(a number between 0 and 9999). Moves forward <i>n</i> lines and displays the page thus selected
<i>-n</i>	(a number between -9999 and -1). Moves backward <i>n</i> lines and displays the page thus selected
<CR>	Following an F, G, P, -P, <i>n</i> or <i>-n</i> instruction it causes the instruction to be repeated, otherwise it is ignored
W	Enter wide mode (if available)
N	Re-enter narrow mode

Table 19.1 - Summary of Inspect Instructions

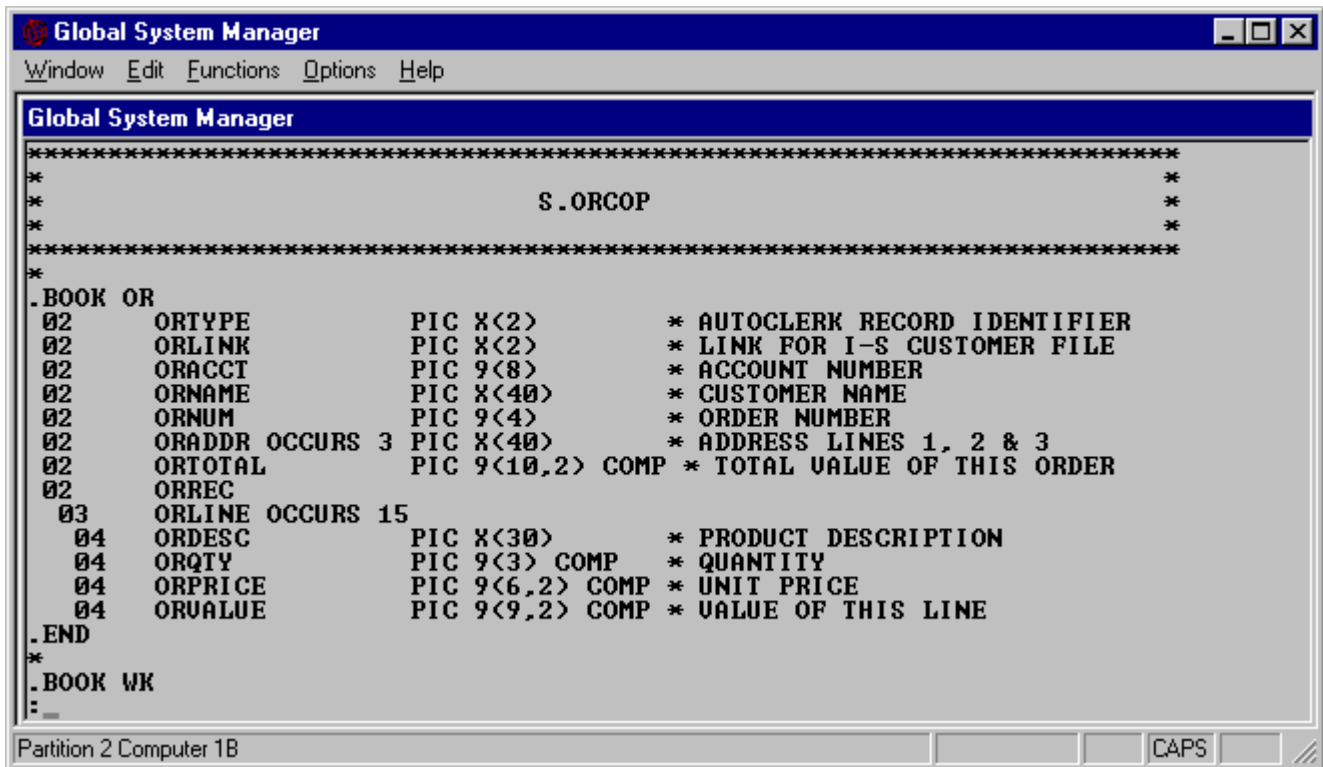


Figure 19.1 - Inspecting a file

19.4 File Inspection Instructions

Once you have entered the details of the file to be inspected, the first page of the file is displayed, followed by the \$INSPECT instruction prompt. This prompt is a single colon displayed on the base line, in character position 1.

The group of lines displayed by \$INSPECT at any time is termed a page. The number of lines in a page is one less than the number available on the screen as the base line is always reserved for the instruction prompt, as shown in figure 19.1.

When lines are displayed print control characters such as form feeds or multiple line feeds are ignored. Each line appears on the screen directly below its predecessor. Normally, if the line length is greater than the screen width, only the start of each line is displayed. However, other parts of longer lines may be examined by using the \$LS instruction.

The instructions that you may key in response to the prompt are summarised in Table 19.1 and explained in more detail, when necessary, in the remainder of this section. Note that if you key an instruction such as -9999 which would take you past the beginning of the file then the very first page of the file will be displayed again. If you select a line past the end of the file by, say 9999, or if an F or G instruction fails to find a line satisfying the specified search criteria then the following special message will appear as the top line of an otherwise blank page:

```
*** END OF FILE ***
```

19.5 F - To Search for a String within a Line

Once an F instruction has been keyed, an extra colon prompt is displayed on the base line. You must reply with the characters of the string that you are searching for. Spaces in the string are significant. Sufficient characters should be supplied to uniquely identify the required line, but if the search finds an unwanted line containing the search string the search can be resumed again by keying <CR>.

The search always starts at the second line of the screen and whenever a match is found a page is displayed with the line containing the specified string at the top of the page.

19.6 G - To Search for a String starting a Line

If a G instruction is keyed, an extra colon prompt is displayed on the base line. You must reply with the starting characters of the line to be searched for. Spaces or tabs at the beginning of the line must not be keyed. Multiple spaces or tabs should not be keyed within the search string. Sufficient characters should be supplied to uniquely identify the line, but if the search finds an unwanted line starting with the characters keyed the search can be resumed again by keying <CR>.

The first line examined is normally the second line of the screen except when one search (G or <CR> resumption) immediately follows another. In this case the first line examined is always the one immediately following the last line found. Whenever a line is found a page is displayed with that line in the middle of the screen.

The G instruction can be used to search for errors in a compilation listing, as all error lines start with an asterisk (see example 2 in section 19.14).

Note that the G instruction differs from the F instruction in two respects. Firstly, it will only find strings located at the start of a line. Secondly, it displays the line containing the string in the middle of the screen.

19.7 \$LS - Set Line Start

\$INSPECT normally displays each line with its first character in the first column of the screen. Up to 132 characters may be displayed on a line, depending on the width of the screen. If the screen is narrower than the lines being inspected, you can use the \$LS instruction to inspect other parts of the line. If you key the \$LS instruction you will be prompted, by a single colon on the same line, for the starting column (1 to 150) of the part of the line to be displayed. For example:

```
:$LS :41
```

This example causes \$INSPECT to display columns 41 onwards of the files being inspected, rather than columns 1 onwards. This option remains in force until overridden by a subsequent \$LS instruction.

Note that the \$LS instruction does not affect the search instructions F and G, which always examine lines starting at column 1. Nor does it affect the operation of the WS and WE instructions.

19.8 WS - Set Write Start Pointer

The WS and WE instructions are used to print or write to an output file selected parts of the files under inspection. Each time that a WS instruction is keyed you are prompted for the **file-id** and **unit-id** of the output file, unless an output file is already open. The new file will be produced as a text file if the file-id you supply begins with an S. prefix and is on a direct access unit which is not a spool unit. In all other cases a relative sequential print file will be produced. You can use this convention to convert from text to print file format, or vice versa. For example, assuming direct access unit 210 is not a spool unit, the following dialogue will result in a text file named S.SALES1:

```
OUTPUT FILE:S.SALES1 UNIT:210
```

whereas this dialogue will result in a relative sequential print file named SALES2:

```
OUTPUT FILE:SALES2 UNIT:210
```

You may reply <CR> to both the OUTPUT FILE: and the UNIT: prompts. In this case a relative sequential print file will be produced on the unit assigned to \$PR, the logical printer. If \$PR is assigned to a direct access device the output file will be named D.\$INSPE.

The WS instruction sets the **write start pointer** to the line which is at the top of the current page displayed on the screen. The write start pointer is initialised to address the first line in the file being inspected, and is reset whenever an E instruction is keyed.

19.9 WE - Set Write End Position

The WE instruction causes lines to be written to the output file from the line addressed by the write start pointer up to and including the line at the top of the page currently displayed on the

screen. If the line pointed to by the write start pointer is later in the file than the top line of the screen the following message is displayed on the base line and the instruction is ignored:

```
START LINE PAST TOP OF SCREEN - IGNORED
```

Once the lines have been written to the output file, the following prompt is displayed:

```
DO YOU WISH TO CLOSE THE OUTPUT FILE?:
```

You should key Y if you intend to end extraction and correctly close the output file. If, however, you reply <CR> (or any single character apart from Y) the file remains open so you can continue extending it.

19.10 E - End Inspection of Current File

To examine a second (or subsequent) file, or to end inspection, key the E instruction. The inspect file prompt will then appear to allow you to either continue processing files or to return to the menu or GSM READY: prompt by keying <ESCAPE>.

If you attempt to end inspection, while an output file is open (i.e. if the WS command has been used without a corresponding WE command) \$INSPECT will display the following warning message:

```
WARNING OUTPUT FILE OPEN - ARE YOU SURE?:
```

Key Y to confirm that inspection of the current file is to end (see example 5 in section 19.17). Key N if you have made a mistake and wish to close the output file, using the WE command, before ending the inspection.

19.11 W - Enter Wide Mode

This causes the display to go into the compressed, wide mode, if the terminal you are using supports this feature.

19.12 N - Re-enter Narrow Mode

This causes the display to revert to non-compressed, narrow mode.

19.13 Example 1 - Examining a report file

The following dialogue takes place on a 24-line screen if you examine the first, second and last page of the report file D.SALES on unit 210:

```
GSM READY:$INSPECT
$48 INSPECT:D.SALES UNIT:210
.....
..... (first page of 23 lines)
.....
: <CR>
.....
..... (second page of 23 lines)
.....
: 9999 (force to end of file)
*** END OF FILE ***
: -23 (display last page)
.....
.....
.....
```



```
:E
$48 INSPECT:<ESCAPE>
GSM READY:
```

19.14 Example 2 - Examining Error and Warning Messages

You wish to examine all the errors and warnings in a compilation listing file L.MB210 which has been produced on unit B09. The following dialogue will take place:

```
GSM READY:$INSPECT
$48 INSPECT:L.MB210 UNIT:B09
.....
..... (first page of 23 lines)
.....
:G :* (search for lines starting with '*')
.....
*** ERROR..... (page containing 1st error)
.....
: <CR>
.....
*** ERROR.... (page containing 2nd error)
.....
: <CR>
*** END OF FILE ***
..... (end of file - all errors inspected)
:E
$48 INSPECT:<ESCAPE>
GSM READY:
```

19.15 Example 3 - Printing Part of a Listing

You wish to obtain a print-out of source listing S.SALES on unit 221 for the program section MAIN only. You know that the only EXIT statement within section MAIN is the final statement of the section. The following dialogue will take place:

```
GSM READY:$INSPECT
$48 INSPECT:S.SALES UNIT:221
.....
..... (first page)
.....
:F :SECTION MAIN
SECTION MAIN
.....
..... (page with SECTION MAIN as top line)
.....
:WS OUTPUT FILE:<CR> UNIT:<CR>
..... (printer output)
:F :EXIT
EXIT
.....
..... (page with EXIT as top line)
:WE DO YOU WISH TO CLOSE THE OUTPUT FILE?:Y
..... (section is printed)
$48 INSPECT:<ESCAPE>
GSM READY:
```

19.16 Example 4 - Converting an RS File to a Text File

You require to convert a relative sequential file, COMMMFILE, to text file format as S.COMMS, so that it can be modified using \$TED or \$EDIT. COMMMFILE is online on 204 (not a spool unit) and S.COMMS will occupy the same volume. You use WS to convert the whole file:

```
GSM READY:$INSPECT
$48 INSPECT:COMMFIL UNIT:204
.....
.....          (first page)
.....
:WS OUTPUT FILE:S.COMMS UNIT:204
:9999
*** END OF FILE ***
:WE DO YOU WISH TO CLOSE THE OUTPUT FILE:Y
:E
$48 INSPECT:<ESCAPE>
GSM READY:
```

19.17 Example 5 - Concatenating Text

While editing the file S.PROG on subunit 210, \$EDIT has exhausted its output file. The editor work file, which is named S.\$EDI01 in a single-user system, will contain the first part of the source of S.PROG, containing most, if not all, of the updates applied in this editing session, and S.PROG itself will still contain the original source. This example shows how \$INSPECT can be used to concatenate the editor work file with the latter part of S.PROG, producing the source file S.RECOV, which, after careful checking, is renamed as S.PROG, completing the recovery process. The following dialogue will take place:

```
GSM READY:$INSPECT
$48 INSPECT:S.$EDI01 UNIT:210
:WS OUTPUT FILE:S.RECOV UNIT:210
.....
.....          (first page of editor work file)
.....
:9999          (to end of editor work file file)
*** END OF FILE ***
:-P
.....
.....          (last page of edit work file - make a note of the last line)
.....
:P          (return to end of editor work file)
:WE DO YOU WISH TO CLOSE THE OUTPUT FILE?:N
:EWARNING OUTPUT FILE OPEN - ARE YOU SURE?Y
          (end inspect of S.$EDI01)
$48 INSPECT:S.PROG UNIT:210
.....
.....          (first screen of original source file)
.....
:          (supply inspect instructions to cause the first line
          that is to be concatenated from the original source
          file to be positioned at the top of the page)
.....
.....          (first page that is to be concatenated)
.....
:WS
:9999          (advance to end of original source file)
*** END OF FILE ***
:WE DO YOU WISH TO CLOSE THE OUTPUT FILE?:Y
:E
$48 INSPECT:S.RECOV UNIT:210
.....
.....          (first page of recovered source file)
.....
:          (supply inspect instructions to check carefully that
          the file has been successfully recovered)
:E
$48 INSPECT:<ESCAPE>
GSM READY:
```

Now that S.RECOV has been checked to be satisfactory, \$F can be used to rename it as S.PROG, replacing the original source. Note that the editor work file will be re-used the very next time that \$EDIT is run, so it is essential that this recovery procedure is carried out (or the editor work file is renamed) before \$EDIT is run again.

20. \$L - List/Dump Files

The \$L command allows you to produce a hexadecimal dump of all or part of a file or volume. Files of any organisation and format can be processed. The dump appears as a sequence of fixed-length records, and each line contains hexadecimal and ASCII translations side by side.

20.1 The Input File Name Prompt

When you run \$L it first displays an identifying message followed by the input file name prompt:

```
GSM READY:$L
$63 SELECTIVE FILE LIST/DUMP PROGRAM
$63 INPUT FILE NAME:
```

You must specify the file-id and unit-id of the file you wish to list. For example:

```
$63 INPUT FILE NAME:SADATA UNIT:209
```

The special reply <CTRL B> to the input file name prompt informs \$L that you wish to dump the sectors of a volume, and you are then prompted for the unit containing that volume. The unit supplied must address a discrete direct access volume or domain, not a subvolume of a domain. Each sector of the volume is treated as though it were a record of an RS file, where the physical sector number corresponds to the record number and the record size is the sector size. For example:

```
$63 INPUT FILE NAME:<CTRL B> UNIT:150
```

If you have replied previously to the input file name and unit prompts during this execution of \$L, a reply of <CR> to either prompt is acceptable and causes input to default to the previously supplied value.

20.2 To Quit

You may quit and return to the menu or ready prompt by simply keying <ESC> to any prompt.

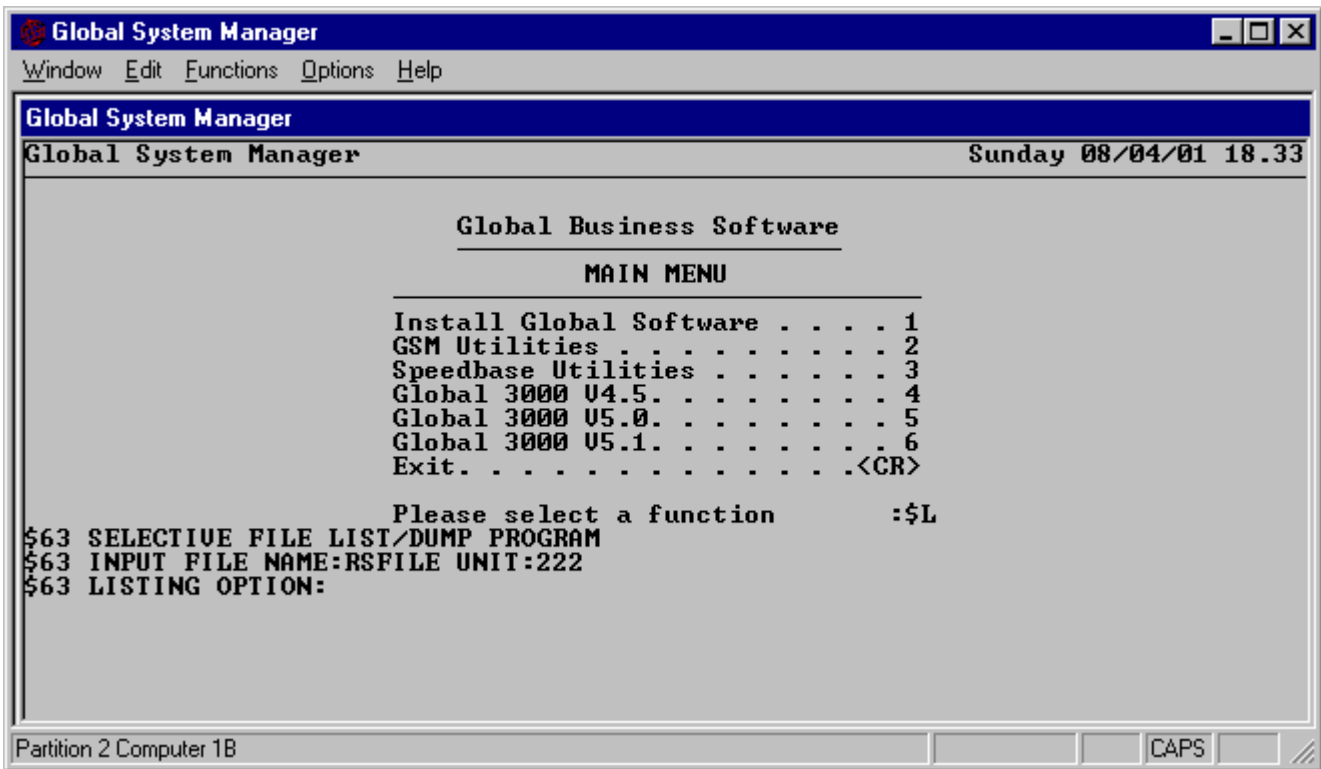


Figure 20.1 - Specifying the file

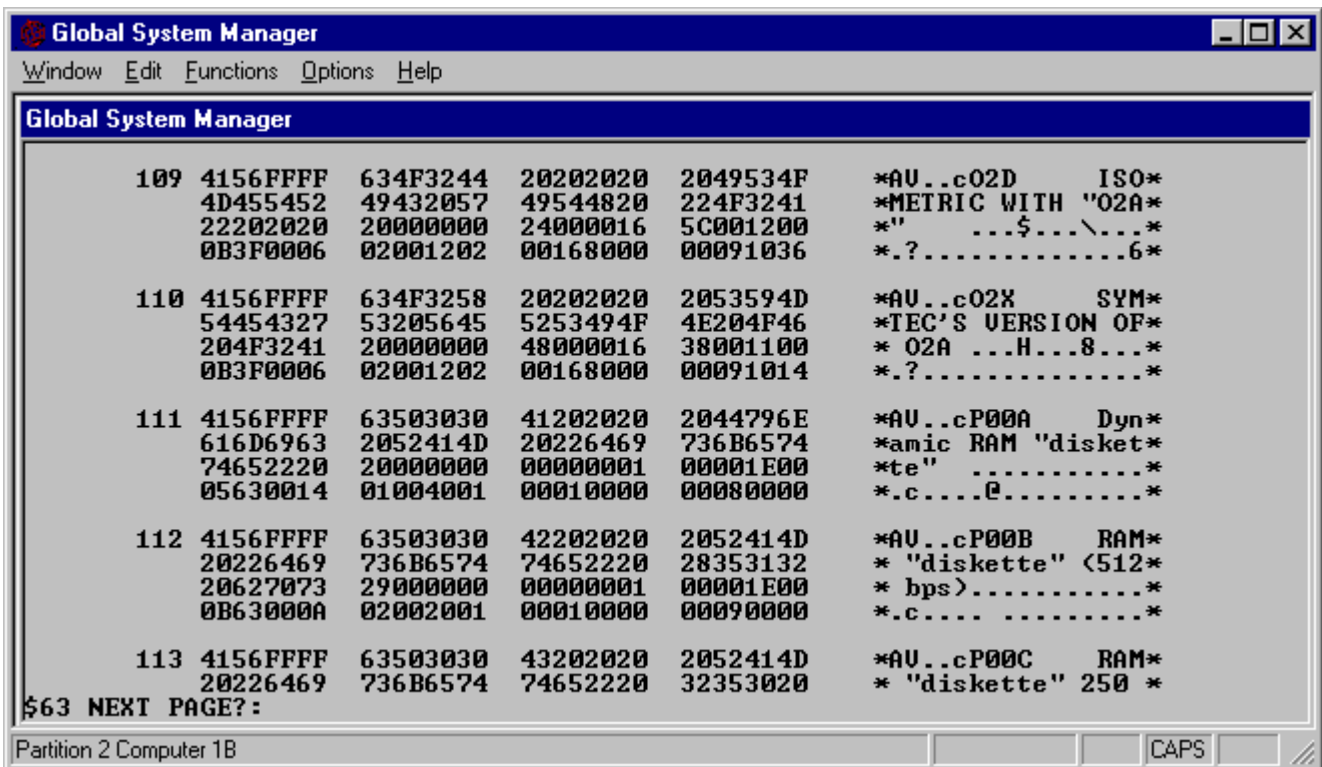


Figure 20.2 - Example page of dump

20.3 The Listing Option Prompt

Once the input file has been defined the listing option prompt:

```
$63 LISTING OPTION:
```

is output repeatedly to enable you to specify zero or more options to control the listing. The valid options and their meanings are as follows:

Option	Meaning
<i>An</i>	This option is only valid when dumping the contents of a volume. The value <i>n</i> is a one or two digit access option specifying the interleaving of sectors within a track.
ASC	Display hexadecimal bytes with ASCII translation (default).
D	Direct all subsequent dumps to the display. This option remains in force for the current file until cancelled by a P option.
<i>En</i>	Set the ending record number to <i>n</i> , a decimal integer. The first record number is 1.
EBC	Display hexadecimal bytes with EBCDIC translation.
<i>On</i>	Set a byte offset <i>n</i> which is to be added to all file or volume addresses. This permits a correctly-aligned dump in cases where fixed-length records do not start at the first byte of the file. For example, the overflow records of an IS file.
OVERFLOW	Used with indexed sequential files (ISAM) to list the overflow records. The first overflow record becomes record 1, and earlier parts of the file are inaccessible.
P	Redirect all subsequent dumps to unit \$PR, the logical printer. A subsequent prompt of WIDE FORMAT is displayed; key Y for wide format or anything else for normal format.
<i>P=uuu</i>	Redirect all subsequent dumps to unit <i>uuu</i> . As above you are asked if you want to print in wide format.
<i>Rn</i>	Set the record length to <i>n</i> , a decimal integer giving the size of the record in bytes.
<i>Sn</i>	Set the starting record number to <i>n</i> , a decimal integer.
<i>n</i>	Select single-record <i>n</i> mode, with <i>n</i> , a decimal integer, being the number of the first record to be dumped.

The special replies of <CTRL C> and ? will list on the screen the current options in force, including details of the input file type, record length and size, data size and total file size. You may return to the input file prompt by keying <CTRL A> to the listing option prompt.

Each time a new input file or volume is selected, the listing options are restored to default values. The dump will be directed to the console, the starting record number and ending record number will be set to select every record/sector of the file/volume, and the byte offset will be set to 0. For a file, the record length will be set to that established when an RS or IS file was created, or to 256 if the file is of another organisation, without a fixed record length. For a volume, the record length will be set to the sector size and the access option will be set to 1 (no interleaving).

To proceed once all the required options have been specified, key <CR> in response to the final listing option prompt. If you are dumping a range of records, the starting record number will be incremented as each complete record is listed. When the listing is complete, the listing option prompt is repeated. You may now change the settings of one or more options to obtain another listing of the same file.

If you have selected single-record mode, by supplying a record number to the listing option prompt, then after each record has been dumped the following prompt will appear to allow you to select the next record to be dumped:

```
$63 NEXT RECORD?:
```

The following replies are allowed:

<i>Reply</i>	<i>Action</i>
Y or <CR>	display the next record;
N	terminate the current display and redisplay the listing option prompt;
B	display the previous record;
<i>nnn</i>	select a specific record (where <i>nnn</i> is a decimal integer)

The following error message:

```
RECORD SIZE TOO LONG
```

indicates that the dump cannot be produced because \$L has insufficient work space to hold a single record of the file. In this case use the listing option prompt to specify a smaller record size (ideally a divisor of the true record size) and try again.

20.4 Output to the Console

When the dump is output to the console, each page of output is followed by the prompt:

```
$63 NEXT PAGE?:
```

which appears on the base line immediately below the dumped data of the file. You key Y or <CR> to proceed to obtain the display of the next page. If you reply with a decimal integer, the starting record number will be set to that value and the next page displayed will start with the dump of that record.

If you reply N the output process will be terminated and the listing option prompt will be redisplayed. If you now key <CR> to this prompt, the listing will resume from the beginning of the last record to be output, if this were only partially displayed, or from the next record otherwise. Thus <CR> has the effect of aligning the dump so that the start of the current record coincides with the start of the display.

20.5 Output to File

If your computer does not possess a printer, you may decide to output the dump to a direct access volume in order to have it printed on another machine later. In this case the dump will occupy the file D.\$L. If this file already exists on the output volume it will be deleted and the new dump file will replace it.

20.6 Operating Notes

You can use the E listing option to access records beyond the end of the file (though not beyond the end of the extent allocated to the file). This allows you to examine the records written to a new file by a program which has failed before closing that file.

If you use \$L on an indexed sequential file and allow the start, end and record length to be defaulted, the records of the prime data area will be output satisfactorily, then will come the index, which may not be meaningful to you, and this will be followed in turn by fragmented records of the overflow area. This is because \$L is not sensitive to file organisation and can be used to dump any type of file. It simply treats any file as a string of contiguous fixed length records.

It is possible use the O option, or the OVERFLOW option, to offset all file addresses by the byte address of the first overflow record, and thus obtain a correctly aligned dump of the overflow area, but a more effective approach is to run the conversion program, \$CONV, to extract a relative sequential file from the indexed sequential file and then use \$L on the new file. Example 2 shows \$CONV being used in conjunction with \$L to dump an indexed sequential file.

20.7 Example 1 - Displaying Records from a File

This example shows how to use \$L to display selected records of an RS file on the console, in hexadecimal format. Firstly, to display the first few records:

```
GSM READY:$L
$63 SELECTIVE FILE LIST/DUMP PROGRAM
$63 INPUT FILE:SADATA UNIT:204
$63 LISTING OPTION:<CR>
.....
..... (first page of information)
.....
$63 NEXT PAGE?:<CR>
.....
..... (second page of information)
.....
$63 NEXT PAGE?:
```

If a page finishes with a partial record, the next page can be easily displayed starting with a complete dump of this record as follows:

```
$63 NEXT PAGE?:N
$63 LISTING OPTION:<CR>
.....
..... (next page aligned with start of record)
.....
$63 NEXT PAGE?:
```

To continue displaying at a specific record, without returning to the listing option prompt, proceed as follows:

```
$63 NEXT PAGE?:1234
.....
..... (page starting with record 1234)
.....
$63 NEXT PAGE?:
```

Figure 20.2 shows a typical page of information. Note how the display is subdivided into numbered records. Each line holds 16 bytes of information which appears both in hexadecimal form and in an ASCII translation.

20.8 Example 2 - Dumping an IS File

Following a program test you wish to dump the indexed sequential file SAINDEX. You first of all use \$CONV to convert the file to relative sequential form as SAWORK and then you print all the records of this file:

```
GSM READY:$CONV
$65 CONVERSION INPUT:SAINDEX UNIT:204
$65 CONVERSION OUTPUT:SAWORK UNIT:<CR> SIZE:<CR> TYPE:R
$65 CONVERTING
(the conversion process, which may take some time, depending
on the size of SAINDEX, takes place)
$65 CONVERSION SUCCESSFUL
$65 CONVERSION INPUT:<ESC>

GSM READY:$L
$63 SELECTIVE FILE LIST/DUMP PROGRAM
$63 INPUT FILE NAME:SAWORK UNIT:204
$63 LISTING OPTION:P WIDE FORMAT? N
$63 LISTING OPTION:<CR>
(the whole work file is now printed to the $PR unit)
$63 LISTING OPTION:<ESC>
GSM READY:
```

20.9 Example 3 - Dumping the Directory Track of a Volume

You wish to obtain a printed dump of the directory track of a diskette on unit 150, (which need not necessarily contain a Global System Manager directory). You already know that the directory track is the first track on the volume and contains 26 sectors (i.e. sectors 1 to 26 are to be dumped). The sectors are interleaved so every other sector is skipped (access option 52). Proceed as follows:

```
GSM READY:$L
$63 SELECTIVE FILE LIST/DUMP PROGRAM
$63 INPUT FILE NAME:<CTRL B> UNIT:150
$63 LISTING OPTION:P WIDE FORMAT Y
$63 LISTING OPTION:A52
$63 LISTING OPTION:E26
$63 LISTING OPTION:<CR>
(the dump is now produced in wide format on the printer)
$63 LISTING OPTION:<ESC>
GSM READY:
```

21. \$LIB & \$LIBM - Library Maintenance

You can use the librarian, \$LIB, to inspect, create or update program or compilation library files, each of which may contain up to 100 executable programs, subroutines or screen format maps. There are special features to allow you to set up dispersed libraries with files on a number of exchangeable volumes, and to run the command under job management.

Note that \$LIB is not used to maintain Global Cobol copy libraries. These are merely specially structured source files and are therefore updated with \$EDIT or \$ED.

Note also that \$LIB is not used to maintain Data File libraries. These are created and amended using the Data File Librarian, \$DLIB, described in Chapter 12.

The \$LIBM utility, described in section 21.18, allows you to create a new library by merging members from one, or more, existing libraries.

21.1 Introduction

\$LIB can process two types of libraries: compilation libraries and program libraries. Each such library is held as an individual file, containing up to 100 **members**. The file-id of a library (sometimes referred to as the **library-id**) must begin with the prefix P. to identify a program library, or C. in the case of a compilation library. The members of a compilation library are either subroutines created by the Global Cobol compiler, \$COBOL, or maps produced by the Screen Formatter, \$FORM. A program library contains main programs or overlays created by \$LINK, the linkage editor, or relocatable programs produced by \$RELOC. Members from the two types of library cannot be mixed.

Each member contained in a library is uniquely identified by means of its **member-id**. The member-id of a subroutine is the program name defined in the PROGRAM statement that begins its source listing. The member-id of a map is the map-id specified when \$FORM was used to create it. The member-id of a main program, overlay or executable program is initially the file-id of the individual program file which contained the member when it was first included in the library. However, the librarian's REN instruction can be used to rename program library members, although the member-ids of compilation library members cannot be changed.

If you wish to create or amend a library, you must first of all establish it as the current **target library**. Then you can use the instructions listed in table 21.1 to include new members or update old ones. \$LIB can only operate on one target library at a time, but the dialogue is structured so that you can process any number of targets, one after another, during a single run.

When you amend a library file it is updated in place. Spare space must be reserved when the library is created to allow for new members or extensions to old ones. \$LIB keeps track of any 'holes' in the library due to deletions and, when updating is complete, re-organises the library so that all the free space is collected together at the end of the file.

Command	Group and Function
Inspect and extract operations (section 21.11)	
LIS	List the index of a selected library *
PRI	Print the index of a selected library *

EXT	Extract a member from a selected library *
EXD	Extract a member from a selected library (followed by subsequent member deletion)
Including and comparing members operations (21.12)	
MER	Merge one or more selected members into target library
COP	Copy one or more selected members into target library
COM	Compare one or more selected members with originals
Rename and delete operations (21.13)	
CHA	Change target library-id or title
REN	Rename or re-title target library member
DEL	Delete one or selected members from the target
Space saving operations (21.14)	
TRU	Truncate target library, leaving the specified amount of free space
NSD	Remove symbolic debug records from target library
Dispersed program library support operations (21.16)	
OFF	Introduce stubs for selected offline members
LNK	Introduce stubs for selected online members
DLN	Delete stubs for selected offline members
Job management support operations (21.17)	
I	Mount named input volume
Close target library operations (21.6)	
END	End target library processing

* These inspect and extract operations are the only instructions which do not require a target library to be established.

Table 21.1 - Librarian Commands

The remainder of this introduction explains the way in which the librarian dialogue is structured, and shows in outline how you inspect, create or amend a library. The introduction then describes how and when re-organisation takes place, how duplicate globals are reported, and the way in which errors are handled.

The remaining sections of this chapter define the groups of related librarian instructions listed in table 21.1

21.2 General Dialogue Structure

When you run \$LIB it begins by asking you to identify a target library. For example:

```
GSM READY:$LIB
$95 TARGET LIBRARY:P.BW UNIT:203
```

If you only need to list or print a library index you need not specify a target. However, if you wish to create a new library, or amend an existing one, you must establish it as the target library by keying its file-id and unit-id (together with other information described later). In either case, eventually the library maintenance prompt will appear:

```
$95 LIBRARY MAINTENANCE
:
```

You may then key any appropriate instruction from the list in table 21.1. The instruction you select will continue with its own dialogue, and eventually redisplay the library maintenance prompt so that you can supply a further instruction, or terminate the process by keying END. When you do so the target library prompt will re-appear, so that you can operate on another library, or exit \$LIB by keying <ESC>. Any dialogue with the librarian therefore assumes the following general form:

```
GSM READY:$LIB
$95 TARGET LIBRARY:
  (Identify first target library, if required)
$95 LIBRARY MAINTENANCE
  :first instruction
  .....
  ..... (operate on first target library)
  .....
$95 LIBRARY MAINTENANCE
  :END
$95 TARGET LIBRARY:
  (Identify next target library, if required)
$95 LIBRARY MAINTENANCE
  :first instruction
  .....
  ..... (operate on next target library)
  .....
$95 LIBRARY MAINTENANCE
  :END
$95 TARGET LIBRARY:<ESC>
GSM READY:
```

MEMBER	CREATED	TYPE	SIZE	TITLE
\$A	01/03/96	MC	2998	UNIT ASSIGNMENT
\$AUTH	08/02/01	MC	24160	AUTH. VALIDATING PROG.
\$BYE	28/02/01	MC	126	TERMINATE PROCESSOR
\$CUEMJ	08/02/01	MC	3434	INSTALL EXTRA CUSTOMISA
\$CUS	08/02/01	MC	29530	GSM CUSTOMISATION
\$CUSA	08/02/01	MC	25866	CUSTOMISATION EXTENSION
\$CUSB	08/02/01	MC	9298	APPLY PRICE UPGRADE
\$CUSEXT	08/02/01	MC	15808	CUSTOMISATION EXTENSION
\$CUSMJ	01/09/94	MC	794	SAVE SYSRES METAJOB
\$CUSP	04/01/00	MC	3898	MAIN PROGRAM
\$CUSP00	04/01/00	MC	7656	MAIN PROGRAM
\$CUSP01	04/01/00	MC	3950	HEADER DETAILS
\$CUSP02	04/01/00	MC	2360	***** FORMAT LETTER SCREEN
\$CUSP03	04/01/00	MC	5774	TESTING SEQUENCES
\$CUSP04	04/01/00	MC	3506	UTILITY SEQUENCE TWO
\$CUSP05	04/01/00	MC	4570	LINE, HOPPER, EMPHASIS
\$CUSP06	04/01/00	MC	2406	GRAPHICAL & CHARACTER T
\$CUSP07	04/01/00	MC	4286	SEQUENCE USAGE ONE
\$CUSP08	04/01/00	MC	3246	UTILITY SEQUENCE ONE
\$CUSP09	04/01/00	MC	3234	UTILITY SEQUENCE THREE
\$CUSP10	04/01/00	MC	3296	UTILITY SEQUENCE FOUR

Figure 21.1 - Listing a library

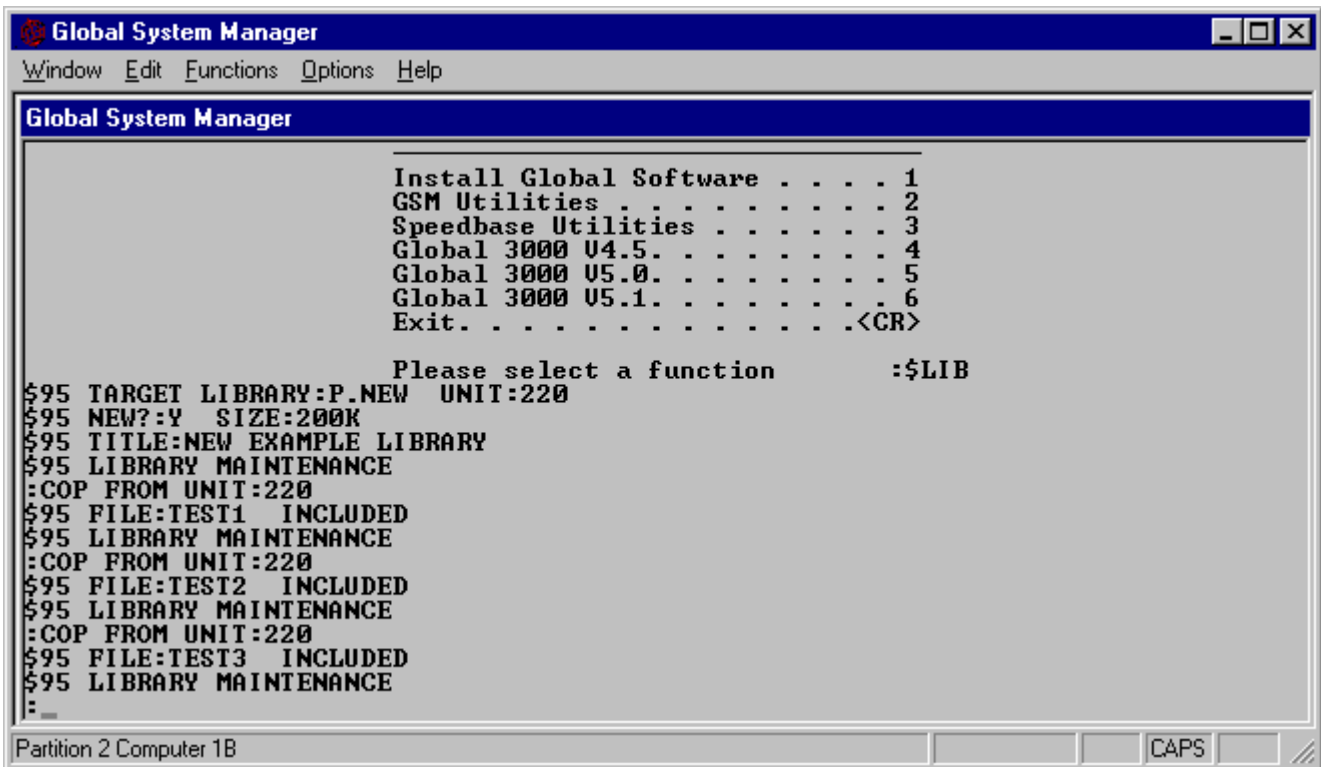


Figure 21.2 - Creating a new library and copying a file to it

Whenever you establish a target library, you **must** ensure it remains online until the next target library prompt appears following your keying of END. This is because the END processing is responsible for rewriting the updated library directory, and possibly compacting the members so that all the free space is collected together at the end of the file. If this is not done, the library will be invalid, and will be rejected when it subsequently comes to be used by the linker or loader, or indeed the librarian itself.

21.3 Inspecting an Existing Library

If you only wish to inspect an existing library, or extract a member from it, there is no need to identify a target library, so you simply key <CR> to its prompt:

```

$95 TARGET LIBRARY:<CR>
$95 LIBRARY MAINTENANCE
:
(Only LIS, PRI and EXT instructions can be used)
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:

```

If you attempt to use an instruction which requires a target library to be established your request will not be honoured. Instead, the following message will be displayed:

```
NO TARGET LIBRARY
```

followed by the library maintenance prompt, so that you can continue with an appropriate instruction.

21.4 Creating a New Library

To create a new library, you must key its library-id and unit-id to the target library prompt. Then you key Y to the subsequent confirmation prompt, and supply its size and title. For example:

```
$95 TARGET LIBRARY:P.SA UNIT:205
$95 NEW?:Y SIZE:100K
$95 TITLE:SALES LEDGER SYSTEM
$95 LIBRARY MAINTENANCE
:
(MER or COP instructions to include the initial members, and
any other instruction, such as TRU, where appropriate)
:END
$95 TARGET LIBRARY:
```

If you key N, <CR> (or any single character apart from Y) to the NEW?: prompt, the creation operation will be abandoned, and the target library prompt will re-appear. This allows you to take corrective action if you really meant to update an existing library, but miskeyed its name or unit.

In response to the SIZE: prompt you may key: A number of bytes; a number followed by the letter K or M, indicating that so many kilobytes or megabytes of storage are to be allocated; or simply 0, in which case the maximum amount of contiguous unused space will be made available. You do not have to be particularly precise in your size request since once the library is built you normally use the TRU instruction, described in 21.14.1, to truncate the new library and release any unwanted storage for subsequent re-allocation. TRU allows you to specify an amount of free space to remain in the library so that you can update it in place if you need to add or amend members later.

The title you specify can be up to 30 characters in length. It is displayed on the console when a program library is attached, and, in the case of a compilation library, appears on the map listing when it is used in a linkage edit.

21.5 Amending an Existing Library

To amend an existing library you must key its file-id and unit-id to the target library prompt, and then key Y, <CR> (or any single character apart from N) to the subsequent process prompt to confirm that the library it identifies is indeed the one you require to modify. For example:

```
$95 TARGET LIBRARY:P.SA UNIT:205
$95 PROCESS SALES LEDGER SYSTEM OF 25/11/90?:Y
$95 LIBRARY MAINTENANCE
:
(MER or COP instructions to add or update members and any
other instruction, such as REN or DEL where appropriate)
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:
```

If the library title or date of last modification, which appear as part of the process prompt, are not as expected, you should respond by keying N. A new target library prompt will then appear to allow you to respecify the library you wish to update.

21.6 To Quit

To quit and return control to the menu or GSM READY: prompt you must key END to the library maintenance prompt and <ESC> to the subsequent target library prompt:

```
$95 LIBRARY MAINTENANCE
:END
```

```
$95 TARGET LIBRARY:<ESC>
GSM READY:
```

When a target library has been created or amended you must perform the END processing to update its directory. In this case <ESC> is treated in exactly the same way as <CR>. If keyed by mistake to the library maintenance prompt itself the following warning message is displayed, followed by the library maintenance prompt itself:

```
$95 KEY END TO REWRITE THE TARGET LIBRARY DIRECTORY
```

21.7 Library Re-organisation

If you delete one or more members from a target library, \$LIB will re-organise it so that all the spare space is held as a contiguous area at the end of the file. For performance reasons re-organisation does not take place after every deletion, but only occurs when necessary: Before adding a member if the free storage is too fragmented to contain it; before processing instructions, such as TRU and NSD, which cannot operate satisfactorily if the library contains "holes"; or as a part of END processing, so that the library is left in a fully compacted form. The following message is displayed:

```
$95 RE-ORGANISING LIBRARY
```

and the normal dialogue is temporarily interrupted whilst the file is scanned and the members are moved.

21.8 Global Symbol Checking

When you have finished creating or amending a compilation library, the END processing checks whether the library contains multiple definitions of the same global symbol, since unplanned duplicates can cause problems at linkage edit time. Whenever a duplicate definition is found \$LIB displays a warning message of the form:

```
$95 WARNING - GLOBAL symbol DEFINED IN BOTH member1 AND member2
```

For example:

```
$95 WARNING - GLOBAL MLR DEFINED IN BOTH SA100 AND SA204
```

Such messages immediately precede the new target library prompt. Each identifies the symbol involved, the first routine (in alphabetical order) in which it is defined, and the second or subsequent routine in which a duplicate definition appears.

If more than 250 global symbols are defined by the members of a library the following warning message is displayed:

```
$95 WARNING - MORE THAN 250 GLOBALS IN LIBRARY
```

This indicates that a complete index record cannot be built, and, in consequence, subsequent attempts to linkage edit programs using the library may fail. Therefore, in the unlikely event of this limit being reached, you should immediately create two or more smaller library files from the affected one.

21.9 Some Common Errors during Library Processing

If you have established a program library as a target, you cannot process compilation files or libraries, and similarly, when the target is a compilation library, you cannot operate on program files or libraries. If you attempt to do so the following warning message is displayed:

```
WRONG TYPE
```

The library maintenance prompt is redisplayed so that you can correct your error.

If you demount the target library by mistake, before you have keyed END, you will be requested to remount it by a prompt of the form:

```
$95 REMOUNT volume-id ON unit-address FOR library-id:
```

where *volume-id* is the name of the volume occupied by the target library. For example:

```
$95 REMOUNT SAPROG ON 205 FOR P.SA:
```

Mount the volume containing the library, then key <CR>, or any single character, to continue.

The following warning messages:

```
FILE NOT FOUND
MEMBER NOT FOUND
$95 INSUFFICIENT SPACE ON TARGET LIBRARY
$95 TARGET LIBRARY FULL
```

may appear at certain stages of the instruction dialogue. Processing of the current instruction is then terminated and the library maintenance prompt is redisplayed so that you can take corrective action. The first warning is displayed if you name a file which is not present on the specified unit; the second if you attempt to process a member which is not in the designated library. The third will appear if there is insufficient space available when you are updating or adding to a library. The fourth means that the target library already contains its full complement of 100 members and there is no space in its directory to hold the details of a new one.

21.10 I/O Error Handling

I/O errors are handled using the normal Global System Manager retry mechanism:

```
$04 ERROR ON unit-address file-id error-message
$04 RETRY?:
```

If you decide the error is irrecoverable and key N to the retry prompt, then, if only an input file is affected, \$LIB will "clean up" to prevent a partial, inconsistent member from being added to the target library, and a new library maintenance prompt will appear so that you can continue with other work. You would normally use the LIS or PRI instruction to obtain an up-to-date index of the target library before continuing.

If an irrecoverable I/O error affects the target library itself, the librarian will attempt to write back a consistent index so that the damaged library can be used as input to \$LIB itself, even though it cannot be submitted to the linker or librarian. Possibly you will be able to copy all but one affected member into a new library. When \$LIB is able to recover the library in this limited way, the following message is displayed:

```
$95 END FORCED BY TARGET LIBRARY I/O ERROR
```


This message is immediately followed by another target library prompt, so that you can set about creating the new library.

If \$LIB is unable even to write back the index, the command is terminated with STOP 9501. In this case the error is catastrophic, and you must recover the affected target library from your own backup copy.

21.11 Inspect and Extract Operations

Three of the four instructions described in this chapter are unique in as much as they are the only ones that do not require a target library to be established in order to operate successfully.

LIS and PRI allow you to display or print any library index, and EXT enables you to create an individual program file from a specified member of a selected library. EXD, which does require a target library, behaves in the same way as the EXT instruction except that the original library member is deleted from the target library.

21.11.1 LIS - List the Index of a Selected Library

You can use the LIS instruction to display the library index page by page on the screen. You must specify the library-id together with the unit it occupies:

```
$95 LIBRARY MAINTENANCE
: LIS LIBRARY: library-id UNIT: unit-id
.....
..... (listing of specified library)
.....
$95 LIBRARY MAINTENANCE
:
```

For example, if you wish to display the index of library P.SA from unit 205 you must key:

```
: LIS LIBRARY: P.SA UNIT: 205
```

and the first page will appear.

If the LIS instruction needs to output more information than can be contained on a single screen display the prompt:

```
$95 NEXT PAGE?:
```

is output on the base line. You must key Y, <CR> (or any single character apart from N) to obtain the next page of output. If you reply N no more of the index will be displayed, and the library maintenance prompt will re-appear.

MEMBER	CREATED	TYPE	SIZE	TITLE
LIBRARY P.\$CMLB1 U8.10 FULL SYSTEM LIBRARY 201 08/04/01 18.45.39				
\$BBS	22/05/00	MC	19274	U09.58 FILE CONVERT BBS FILES
\$C	01/09/94	MC	1112	U8.1 CLEAR SCREEN
\$CALC	01/03/96	MC	17878	U8.1 CALCULATION COMMAND
\$CATAL	01/03/96	MC	8012	U8.1 CATALOGUE COMMAND
\$CDUMJ	04/01/00	MC	1866	***** \$CDUPD METAJOB
\$CDUPD	04/01/00	MC	8418	***** PREPARE SYSRES FOR CD U
\$COMPRES	08/07/99	MC	10590	***** FILE COMPRESSION
\$CONU	01/03/96	MC	8350	U8.1 RS/IS CONVERSION
\$CUSNET	08/02/01	MC	6202	***** NETWORK WIDE CUSTOMISAT
\$CUSP18	04/01/00	MC	4130	***** CHARACTER TRANS
\$CUSP20	04/01/00	MC	832	U8.1 CHARACTER TRANS
\$CUSP21	04/01/00	MC	1094	U8.1 CHARACTER TRANS
\$CUSP22	04/01/00	MC	402	U8.1 CHARACTER TRANS
\$CUSP23	04/01/00	MC	1202	U8.1 CHARACTER TRANS
\$CUSP24	04/01/00	MC	844	***** CHARACTER TRANS
\$CUSP25	04/01/00	MC	418	***** CHARACTER TRANS
\$CUSP26	04/01/00	MC	1254	***** CHARACTER TRANS
\$CUSP27	04/01/00	MC	1234	***** CHARACTER TRANS
\$CUSPUP	04/01/00	MC	2534	U8.1 MAIN PROGRAM
\$CUSUPD	08/02/01	MC	17744	U8.1 UPDATE REVISION
\$DLMEEXEC	02/06/99	32	5482	U8.1 CALL 32-BIT DLMS
\$95 NEXT PAGE?:				

Figure 21.3 - A Program Library

MEMBER	CREATED	TYPE	SIZE	TITLE
LIBRARY C.\$MCOB U8.10 SYSTEM SUBROUTINES 1 204 08/04/01 18.46.16				
AB\$A	12/11/97	CO	809	***** BDAM BASIC DIRECT A. M.
AB\$H	12/11/97	CO	1242	***** BDAM FOR DLIB
AI\$A	12/11/97	CO	2465	***** INDEXED SEQ. A. M.
AL\$A	31/03/98	CO	5062	U8.1T DATA LIBRARY ACCESS MET
AM\$A	09/02/99	CO	12313	***** DATA MANAGEMENT A.M.
AN\$A	01/01/97	CO	4387	U8.1 NATIVE ISAM (NIAM)
AP\$A	31/12/97	CO	747	***** PHYSICAL SECTOR A. M.
AR\$A	12/11/97	CO	1591	***** RSAM WITHOUT BLOCKING
AR\$B	12/11/97	CO	2260	***** RELATIVE SEQ A. M.
AR\$H	12/11/97	CO	1200	***** RSAM FOR DATA LIBRARIES
AS\$C	28/02/01	CO	7499	U8.10 READ ONLY SPAM FOR CISA
AS\$N	28/02/01	CO	3909	U8.10 READ ONLY SPAM FOR BTRI
AS\$R	28/02/01	CO	4705	U8.10 READ ONLY SPAM
AT\$A	22/05/00	CO	1504	***** TEXT FILE ACCESS METHOD
AT\$H	22/05/00	CO	1743	***** TFAM FOR DATA LIBRARIES
AU\$A	07/07/98	CO	1113	U8.1U VARIABLE LENGTH ACCESS
AZ\$A	01/09/94	CO	338	U8.1 ISAM INDEX READ AREA
BA\$A	01/09/94	CO	418	U8.1 CUST\$
BC\$A	01/09/94	CO	161	U8.1 CMND\$ - INVOKE SYSTEM REQ
BE\$A	12/11/97	CO	263	***** DELE\$
BG\$A	31/03/98	CO	1279	U8.1T CATA\$
\$95 NEXT PAGE?:				

Figure 21.4 - A Compilation Library

Figures 21.3 and 21.4 show the contents of P.\$CMLB0, the Global System Manager command library (a program library), and C.\$MCOB, a Cobol system library (a compilation library). The format of the display is that of a header, appearing on every page, an information line for each

member contained in the library, and trailer information, only output on the very last page. The information lines are arranged in alphabetical order of member-id.

The **header** contains the library-id, the title, the unit-id and the current date and time.

Each **information line** consists of the member-id, followed by its creation date, type, size and title. The date indicates when a subroutine was last compiled, a map created or updated, or an executable program last linked or relocated. The size is the number of bytes of file space occupied within the library, not the amount of main storage the member requires. Table 21.2 defines the codes that may appear in the TYPE column.

The **trailer information** at the very end of the listing indicates the date at which the library was last updated, its size, the number of members it currently contains, and the amount of free space available for new members.

Type	Meaning
AS	Absolute assembler program created by FCONV.
CO	Compilation file (i.e. subroutine) created by the Global Cobol compiler.
MA	Map created by the Screen Formatter, \$FORM.
MC	Global Cobol program or command, created by \$LINK.
PI	Position independent program created by FCONV.
Rn	Relocatable program, address form n, created by \$RELOC.
ST	Stub locating an offline member, created by the librarian's OFF instruction.

Table 21.2 - Types of Library Member

21.11.2 PRI - Print the Index of a Selected Library

You can use the PRI instruction to cause a specified library index to be printed on the unit assigned to \$PR:

```
$95 LIBRARY MAINTENANCE
:PRI LIBRARY:library-id UNIT:unit-id
(The specified library index is now printed)
$95 LIBRARY MAINTENANCE
:
```

For example, if you wish to print the index of library P.SA from unit 205 you must key:

```
:PRI LIBRARY:P.SA UNIT:205
```

The index will be printed, and another library maintenance prompt will appear.

When \$PR is assigned to a direct access volume, rather than a printer, the PRI instruction writes print records to the file D.\$LIB, extending the file if it already exists. This allows you to accumulate information concerning a number of volume directories on a single direct access file when no printer is immediately available.

21.11.3 EXT - Extract a Member from a Selected Library

The EXT instruction enables you to create an individual program file from a specified member of a selected library. The library is read-only as far as EXT is concerned.

Despite the name of the command, the member is not actually physically extracted, but only a copy of it made. You must specify the library and its unit, together with the member-id and the name and unit for the new file to be created from it:

```
$95 LIBRARY MAINTENANCE
:EXT LIBRARY:library-id UNIT:unit-id
$95 EXTRACT:member-id TO:file-id UNIT:unit-id EXTRACTED
$95 LIBRARY MAINTENANCE
:
```

21.11.4 EXD - Extract and Delete a Library Member

The EXD command enables you to create an individual program file from a specified member of the target library followed by subsequent deletion of that member. This command is a combination of the EXT command (section 21.11.3) followed by the DEL command (section 21.13.3). You must specify the target library and unit, together with the member-id and the unit for the new file to be created from it:

```
GSM READY:$LIB
$95 TARGET LIBRARY:P.SA UNIT:205
$95 PROCESS SALES LEDGER SUITE OF 22/11/90?:Y
$95 LIBRARY MAINTENANCE
:EXD
$95 EXTRACT AND DELETE:SA100 TO:SA100 UNIT:205
      EXTRACTED AND DELETED
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY
```

Prompt	Response	Normal action
FILE:	<i>file-id</i>	If the file consists of an individual map, subroutine or program, process it. If the file is a library itself, respond with the MEMBER: prompt to see which members of that library you require to process.
	<CR>	Respond with <i>file-id?</i> : selection prompts for all appropriate libraries and files on the specified unit, so you can process them if you wish.
	<CTRL A>	Terminate the instruction.
	<CTRL B>	Process every appropriate member on the unit.
<i>file-id?</i> :	Y	If the file consists of an individual map, subroutine or program, process it. If the file is a library itself, respond with the MEMBER: prompt to see which members of that library you require to process.
	N	Skip this file, and proceed to the next appropriate one on the specified unit.
	<CR>	Skip this file, and proceed to the next appropriate one on the specified unit.
	<CTRL A>	Do not process this file. Terminate the instruction.
	<CTRL B>	Process the member or members this file contains, and every member from each appropriate file remaining on the unit.
MEMBER:	<i>member-id</i>	Process the specified member.
	<CR>	Respond with <i>member-id?</i> : selection prompts for all members in this library, so that you can process them if you wish.

	<CTRL A>	Process no members from this library.
	<CTRL B>	Process all the members from this library.
<i>member-id?:</i>	Y	Process this member.
	N	Skip this member and proceed to the next.
	<CR>	Skip this member and proceed to the next.
	<CTRL A>	Do not process this member, nor any more from the current library.
	<CTRL B>	Process this member and every other one remaining in the current library.

Table 21.3 - Responding to File, Member and Selection Prompts

21.11.5 Operating Notes

Although the LIS, PRI and EXT instructions can function without a target library, they also operate when one has been established. In this case, to process the target library itself, you need only key <CR> to the LIBRARY: prompt. For example, supposing you intend to update program library P.SA. You might decide to list its contents prior to including new members with the COP and MER instructions.

You might start as follows:

```
GSM READY:$LIB
$95 TARGET LIBRARY:P.SA UNIT:205
$95 PROCESS SALES LEDGER SUITE OF 22/11/90?:Y
$95 LIBRARY MAINTENANCE
: LIS LIBRARY:<CR>
.....
..... (first page of P.SA's index)
.....
$95 NEXT PAGE?:<CR>
.....
..... (second, and last page of index)
.....
$95 LIBRARY MAINTENANCE
:
```

If you now decided to examine library P.NEW on 204 which you intend to merge with P.SA, you would continue:

```
$95 LIBRARY MAINTENANCE
: LIS LIBRARY:P.NEW UNIT:204
```

```

.....
..... (first page of P.NEW's index)
.....
$95 NEXT PAGE?:
etc.

```

21.12 Including and Comparing Members

The instructions described in this section are employed in creating or amending a target library. You can include new members using COP or MER; replace existing ones with COP; and check that members have been included correctly by comparing them with the originals using COM.

21.12.1 COP - Copy Member(s) into Target Library

The COP instruction allows you to copy one or more selected members into the target library. You first of all specify the input unit containing the new members, which are read-only as far as the copy operation is concerned.

In the simplest case, when you just wish to include a member held as an individual file, you key the file-id to the subsequent prompt. For example, to copy the program OURSORT on 204 into the target library P.SA:

```

$95 LIBRARY MAINTENANCE
:COP FROM UNIT:204
$95 FILE:OURSORT INCLUDED
$95 LIBRARY MAINTENANCE
:

```

If you supply a library-id in response to the FILE: prompt a subsequent MEMBER: prompt asks you to specify which member or members you require. For example, to include \$SORT from library P.FREE on 209:

```

$95 LIBRARY MAINTENANCE
:COP FROM UNIT:209
$95 FILE:P.FREE MEMBER:$SORT INCLUDED
$95 LIBRARY MAINTENANCE

```

These are the two simplest ways in which to use the FILE: and MEMBER: prompts. You may also key the special responses: <CR>, <CTRL A> and <CTRL B> to serve as selection codes, with meanings similar to those used in \$F. For example, a single COP instruction might be used to copy program SALES, the entire contents of P.FREE, and members SA200 and SA201 of library P.NEW, into target library P.SA on 205. All these are on unit 209 together with program file SATEST and program library P.SAOLD which are not required:

```

GSM READY:$LIB
$95 TARGET LIBRARY:P.SA UNIT:205
$95 PROCESS SALES LEDGER SYSTEM OF 20/11/90?:Y
$95 LIBRARY MAINTENANCE
:COP FROM UNIT:209
$95 FILE:<CR>
SALES ? :Y
SATEST ? :N
P.FREE ? :Y MEMBER:<CTRL B>
    $A INCLUDED
    $C INCLUDED
    .....
    .....
    .....
    $SORT INCLUDED
P.NEW ? :Y MEMBER:<CR>
SA100 ? :N

```

```

SA200 ? : Y
SA201 ? : Y
P.SAOLD ? : <CTRL A>
$95 LIBRARY MAINTENANCE
: END
$95 TARGET LIBRARY : <ESC>
GSM READY :

```

The copy operation may be used to replace an existing member with an updated version. Both the old and the new one must have the same member-id. Then, whenever \$LIB detects that you are attempting to include a member whose member-id matches one already present on the target library, it displays the following warning message:

```
MEMBER ALREADY EXISTS - DELETE?
```

If you reply Y the existing member will be replaced by the new version. If you key N, <CR>, (or any single character apart from Y) the target library will remain undisturbed and the new member will be ignored: If you are including a number of members the copy process will then continue with the next satisfying the selection criterion.

21.12.2 MER - Merge Member(s) into Target Library

The MER instruction enables you to include one or more selected members into the target library, rather like a copy operation. The difference is that a merge **cannot** be used to update an existing member, since members already present on the target library take precedence over new ones with the same member-id. When a match is detected \$LIB simply displays the following warning message and ignores the later member:

```
MEMBER ALREADY EXISTS
```

If you are including a number of members the merge process will then continue with the next to satisfy the selection processing.

Merge dialogue is exactly the same as that used for a copy operation, except that the MER instruction is used in place of COP. Simple examples such as:

```

$95 LIBRARY MAINTENANCE
:MER FROM UNIT:209
$95 FILE:OURSORT INCLUDED
$95 LIBRARY MAINTENANCE
:

```

or:

```

$95 LIBRARY MAINTENANCE
:MER FROM UNIT:209
$95 FILE:P.FREE MEMBER:$SORT INCLUDED
$95 LIBRARY MAINTENANCE
:

```

are not particularly realistic, since in these cases it would be more flexible to use the COP instruction, in case you needed to update an old version of OURSORT or \$SORT. A more relevant use of the merge operation is provided by the following example.

Suppose unit 212 contains subroutines required by a development project, and represents the status of the routines when they were thoroughly unit tested, say, a month previously. In the interim period, corrections have been made to some of these routines, and new ones have been added. The additional members involved are all on unit 211. Then the following procedure could be used to create an up-to-date subroutine library, C.SR, on unit 210:


```

GSM READY:$LIB
$95 TARGET LIBRARY:
$95 TARGET LIBRARY:C.SR UNIT:210
$95 NEW:Y SIZE:0
$95 TITLE:COMBINED SUBROUTINE LIBRARY
$95 LIBRARY MAINTENANCE
:
:MER FROM UNIT:211
$95 FILE:<CTRL B>
.....
..... (all members from 211 are included, from both
..... individual files and compilation libraries)
.....
$95 LIBRARY MAINTENANCE
:
:MER FROM UNIT:212
$95 FILE:<CTRL B>
.....
..... (members from 212 are included, except when
..... they have already been superseded members
..... previously merged from 211)
.....
$95 LIBRARY MAINTENANCE
:TRU NEW SPARE SPACE:<CR> (explained in 21.14.1)
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:<ESC>
GSM READY:

```

Of course, a similar effect could have been achieved using a COP instruction, providing 212 had been processed before 211. However, whenever an update occurred you would need to reply Y to the member already exists prompt, which would be operationally less convenient. There would be more data transferred, since old members would have been copied prior to deletion. Furthermore because of the deletions a library re-organisation would be necessary which is not needed using the merge technique.

21.12.3 COM - Compare Member(s) with Originals

The COM instruction is provided so that you can compare members previously copied or merged into the target library with their original versions. The dialogue is similar to that employed in a copy or merge operation, except that the COM instruction is used. For example, suppose you decide to update library P.SA1 with program SALES, \$SORT from P.FREE, and the entire contents of P.NEW, and you wish to check that the copy process has been performed accurately. This requires just one COP instruction, followed by a similar COM:

```

GSM READY:$LIB
$95 TARGET LIBRARY:P.SA1 UNIT:208
$95 PROCESS V1.0 SALES LEDGER SYSTEM OF 22/11/90?:Y
$95 LIBRARY MAINTENANCE
:COP FROM UNIT:209
$95 FILE:<CR>
SALES ? :Y INCLUDED
SATEST ? :N
P.FREE ? :Y MEMBER:$SORT INCLUDED
P.NEW ? :Y MEMBER:<CTRL B>
SA100 INCLUDED
.....
.....
.....
SA920 INCLUDED
P.SAOLD? :<CTRL A>
$95 LIBRARY MAINTENANCE
:COM FROM UNIT:209
$95 FILE:<CR>
SALES ? :Y IDENTICAL

```

```

SATEST ? :N
P.FREE ? :Y MEMBER:$SORT IDENTICAL
P.NEW ? :Y MEMBER:<CTRL B>
SA100 IDENTICAL
.....
.....
.....
SA920 IDENTICAL
P.SAOLD ? :<CTRL A>
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:<ESC>
GSM READY:

```

Providing that every byte of the member selected from the input unit, apart from its creation date and title, matches those of the copy on the target library, the following confirmation message is displayed (as shown in the above example):

```
IDENTICAL
```

The warning messages:

```
DISCREPANCY AT nnnn
```

or:

```
NOT FOUND
```

are displayed if either the copies do match, or if the member-id of a selected member is not present in the target library. The quantity *nnnn* is the number of the first byte within the two members not to match, counting from 0.

21.13 Rename and Delete Operations

The instructions described in this section can be used to change the file-id or the title of the target library, or rename, re-title or delete one of its members. Titles can be at most 30 characters in length.

21.13.1 CHA - Change Target Library-id or Title

The CHA instruction allows you to change the library-id or title of the target library. For example, to change P.SA, entitled SALES LEDGER SYSTEM, to P.SAOLD, SALES LEDGER SYSTEM (NOV 90):

```

$95 LIBRARY MAINTENANCE
:CHA P.SA TO:P.SAOLD
OLD TITLE SALES LEDGER SYSTEM
NEW TITLE:SALES LEDGER SYSTEM (NOV 90)
$95 LIBRARY MAINTENANCE
:

```

You may key <CR> in place of the new library-id or new title, in which case the existing library-id or title will remain unchanged. You may replace selected characters at the beginning of the old title by keying your amendment, terminated by <CTRL A>, as the new one. For example, to change the title of library C.SA from V6.0 SALES LEDGER ROUTINES to V6.1 SALES LEDGER ROUTINES:

```

$95 LIBRARY MAINTENANCE
:CHA C.SA TO:<CR>
OLD TITLE:V6.0 SALES LEDGER ROUTINES
NEW TITLE:V6.1<CTRL A>
$95 LIBRARY MAINTENANCE

```

:

Note that if you attempt to change a library-id and a file with the new name is already present on the same unit as the target library, the following warning message is displayed:

```
FILE ALREADY EXISTS - DELETE?:
```

If you key Y the existing file will be deleted and the change operation will continue as though it had never existed. A reply of N, <CR> (or any single character apart from Y) causes the instruction to be abandoned and the library maintenance prompt to re-appear.

21.13.2 REN - Rename or Re-title Target Library Member

You can use the REN instruction to change a particular member-id within a target program library, and to alter the title of any type of member. The dialogue is of the general form:

```
$95 LIBRARY MAINTENANCE
:REN :old-member-id [AS:new-member-id]
OLD TITLE old-title
NEW TITLE:new-title
$95 LIBRARY MAINTENANCE
:
```

The dialogue shown in square brackets is omitted for a compilation library, because it is only possible to rename subroutines or maps by recompiling using \$COBOL, or by creating an updated map using \$FORM. For example, to correct a spelling mistake in the title of subroutine SAMLRL:

```
$95 LIBRARY MAINTENANCE
:REN :SAMLRL
OLD TITLE BANK RATS APPLICATION
NEW TITLE:BANK RATE APPLICATION
$95 LIBRARY MAINTENANCE
:
```

You may key <CR> in place of a new member-id or new title, in which case the existing member-id or title will remain unchanged.

You may replace selected characters at the beginning of the old title by keying your amendment, terminated by <CTRL A>. For example, to adjust a version code by changing the title of program SA100 from V6.0 SALES ACCT ROOT to V6.1 SALES ACCT ROOT:

```
$95 LIBRARY MAINTENANCE
:REN :SA100 AS:<CR>
OLD TITLE V6.0 SALES ACCT ROOT
NEW TITLE:V6.1<CTRL A>
$95 LIBRARY MAINTENANCE
:
```

Note that if you rename a program library member and the new member-id you choose is already present on the target library, the following warning message is displayed:

```
MEMBER ALREADY EXISTS - DELETE?:
```

If you key Y the old member will be deleted and the rename operation will continue as though it had never existed. A reply of N, <CR> (or any single character apart from Y) causes the instruction to be abandoned and the library maintenance prompt to re-appear.

21.13.3 DEL - Delete Member(s) From the Target Library

The DEL instruction enables you to delete one or more selected members from the target library. To delete one particular member, you simply key its member-id in response to the MEMBER: prompt. For example, to remove SA100 from the target library:

```
$95 LIBRARY MAINTENANCE
:DEL MEMBER:SA100 DELETED
$95 LIBRARY MAINTENANCE
:
```

You may also use the special responses <CR>, <CTRL A> and <CTRL B>, to delete a selection of members, using the scheme summarised in the lower half of table 21.3. For example:

```
$95 LIBRARY MAINTENANCE
:DEL MEMBER:<CR>
  SA200 ? :N
  SA205 ? :N
  SA300 ? :Y DELETED
  SA310 ? :Y DELETED
  SA400 ? :N
  SA500 ? :Y DELETED
  SA600 ? :<CTRL A>
$95 LIBRARY MAINTENANCE
:
```

In the above example, SA300, SA310 and SA500 are deleted from the target library, which has been displayed member by member for selection, following a <CR> response to the member prompt. The special reply of <CTRL A> to the select prompt terminates the process.

21.14 Space Saving Operations

The two instructions described in this section can be used to limit the size of the target library either by reducing the size of the members themselves (NSD) or by returning unwanted free space to the volume for re-allocation (TRU). **Note that the size of a library cannot be increased from within \$LIB.** To increase the size of a library, the \$F COP instruction must be used (see section 15.10).

21.14.1 TRU - Truncate Target Library

The TRU instruction returns any unwanted free space in the target library to the volume it occupies, so that the storage thus released becomes available for re-allocation. The dialogue allows you to specify how much free space you require to remain in the library, once truncation has taken place:

```
$95 LIBRARY MAINTENANCE
:TRU NEW SPARE SPACE:required-free-space TRUNCATED
$95 LIBRARY MAINTENANCE
:
```

If you never intend to update the library you will require no spare space at all, and in this case you may key <CR> or 0 in response to the NEW SPARE SPACE: prompt. Normally, however, you should leave enough room to hold several extra members, to prevent re-organisation having to take place unnecessarily.

The special response -A will leave sufficient spare space to create stubs in the library for up to 100 members in total. It is intended for use under Job Management by installation software.

The following example shows how a program library might be set up initially:

```
GSM READY:$LIB
$95 TARGET LIBRARY:P.SA UNIT:205
$95 NEW:Y SIZE:0
$95 LIBRARY MAINTENANCE
:
(COP or MER instructions to include the initial members).
$95 LIBRARY MAINTENANCE
:TRU NEW SPARE SPACE:40K
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:<ESCAPE>
GSM READY:
```

By specifying an initial SIZE request of 0 in the third line of the dialogue, the maximum amount of contiguous space available is allocated to the library initially. The TRU instruction returns any unwanted space to the volume, but keeps 40K bytes spare for subsequent expansion.

Note that if you request more spare space than is actually present on the library the following warning message will be displayed and the truncation operation will be abandoned:

```
TOO LARGE
```

21.14.2 NSD - Remove Symbolic Debug Records

Normally Global Cobol compilation files created by \$COBOL contain symbolic debug records which, in turn, become part of any program files produced by linking the compilations. The records do not affect normal program execution in any way, since they are only interrogated by \$DEBUG, the symbolic debugging system. However, they may occupy a sizeable proportion of a library, since there will be an 11-byte record for each symbol employed in a subroutine, or for each symbol of each subroutine used in constructing a program. The symbolic debug records can be removed at compile or link time by specifying the NSD option when \$COBOL or \$LINK is executed. However, it may prove simpler to eliminate them from the target library itself when they are no longer needed by using the NSD instruction:

```
$95 LIBRARY MAINTENANCE
:NSD target-library-id?:Y SYMBOLIC DEBUG RECORDS REMOVED
$95 LIBRARY MAINTENANCE
:
```

Following the reply of NSD to the \$LIB command prompt, the library-id of the target is displayed in a confirmation prompt, to check that you really do wish to remove symbolic debug records from the library in question. The operation will only proceed if you key Y. For example:

```
$95 LIBRARY MAINTENANCE
:NSD P.SA?:Y SYMBOLIC DEBUG RECORDS REMOVED
$95 LIBRARY MAINTENANCE
:
```

21.15 Abandoning the Library Update

The ABA instruction, deliberately omitted from table 21.1, should only be used if a catastrophic error occurs making the target library completely unusable. **The ABA command will delete the target library.**

For example, to abandon the update and delete the target library:

```

$95 LIBRARY MAINTENANCE
:ABA
$95 TARGET LIBRARY WILL BE DELETED - ARE YOU SURE?:Y ABANDONED
$95 TARGET LIBRARY:

```

If the ABA instruction is used, the target library must be restored from backup.

21.16 Dispersed Program Library Support

The two instructions described in this section are mainly for use in software installation jobs. They allow large program libraries to be split up into smaller ones and dispersed onto separate disks, but interlinked so that Global System Manager knows where to find all the programs required.

A dispersed program library consists of a number of files, each of which has the same library-id but occupies a different volume. For example, when Global Cobol is installed onto hard disk, a volume is allocated on the domain called SYSDEV containing P.\$CMLB0, which is linked to the P.\$CMLB0 on SYSRES.

A dispersed program library therefore consists of two or more P. files with the same name occupying different volumes. The volumes may be exchangeable (e.g. diskettes all occupying the same unit address) or they may occupy different addresses (e.g. different volumes on a domain or a library split between diskette and RAM disk).

By using the OFF or LNK instructions you can add small dummy members, known as **stubs**, to each part of a dispersed library. Each stub contains the volume name of the disk containing the program with that name and also the unit address if it is on a different unit. If you list the library (as shown in figure 21.3) the stub will appear as:

```

OFFLINE - LOOK ON SYSRES
or:
OFFLINE - LOOK ON SYSRES - 201

```

If the operator attempts to run a program represented by a stub, Global System Manager reassigns \$P or \$CP as necessary and makes sure the correct volume is mounted, giving a message such as if it is not:

```
PLEASE MOUNT SYSRES ON 1ST HARD DISK - 201 AND KEY <CR>:
```

Note that if a stub had **not** been present for the volume, Global System Manager would have simply output the program prompt which it uses whenever a required program or command is missing. For example:

```
$01 PROGRAM SALES IN LIBRARY P.SL REQUIRED ON 204:
```

The operator would now have to guess which of the other volumes actually contained the program. If the wrong one was mounted, the program prompt would simply re-appear. There might be a frustrating search until the right volume was obtained.

You can use the OFF instruction to interconnect your own dispersed libraries if you are developing systems which exceed the capacity of the available direct access volumes. Each separate file of such a library must be created or amended as an individual target library. The members whose code the file actually contains are included using copy or merge operations in

the normal way. Then stubs for **every one** of the offline members, residing on the other volumes occupied by the dispersed program library, must be introduced using the OFF instruction.

21.16.1 OFF - Introduce Stubs for Offline Members

The OFF instruction allows you to introduce one or more selected stubs into the target library. Operationally the dialogue is very similar to that of the MER (merge) instruction (see section 21.12.2), except that OFF is keyed in place of MER, and there is an additional prompt to determine the volume to be occupied by the offline members. For example, to introduce a stub into the target library to inform Global System Manager that program SALES will be offline on SAPRG1:

```
$95 LIBRARY MAINTENANCE
:OFF FROM UNIT:204 VOLUME:SAPRG1
$95 FILE:SALES OFFLINED
$95 LIBRARY MAINTENANCE
```

Note that SALES may not actually be on SAPRG1 at the time of using the OFF instruction: You may move it to the correct volume at some later stage before you come to use the target library. Similarly, in the following example, stubs are introduced into the target library, P.SA, for all the members currently occupying P.NEW. Eventually, if the dispersed library is correctly constructed, all those members will either have to be part of another P.SA library on SAPRG1, or will have to appear as individual program files on that volume:

```
$95 TARGET LIBRARY:P.SA
$95 PROCESS SALES LEDGER SYSTEM OF 14/02/88?:Y
$95 LIBRARY MAINTENANCE
:OFF FROM UNIT:204 VOLUME:SAPRG1
$95 FILE:P.NEW MEMBER:<CTRL B>
      SA100 OFFLINED
      SA200 OFFLINED
      .....
      .....
      SA950 OFFLINED
$95 LIBRARY MAINTENANCE
:
```

Note that if the members whose stubs you are including are already occupying the destination volume, you can simply key <CR> to the VOLUME: prompt. The librarian will then obtain the volume-id to use in the stubs from the label of the input volume itself.

A stub introduced by the OFF instruction will replace one with the same member-id which is already present on the target library. However, a stub will never replace the code of an existing member. If this is attempted the following warning message is displayed in place of the confirmation OFFLINED which is normally displayed to show that the proper interconnection has been made:

```
MEMBER ALREADY EXISTS
```

Sometimes the members selected for processing by the OFF instruction may themselves be represented by stubs introduced by a previous use of OFF. In this case the volume information in these secondary stubs is copied unchanged to the target library, and displayed as the volume-id in the confirmation message, providing of course, that the corresponding real member is not already present on the target library. Example 2 (see section 21.16.6) shows how secondary stubs may be used.

21.16.2 LNK - Link Together On-line Libraries

The LNK instruction is similar to OFF, but is intended for use when the two libraries are usually on-line, but occupy different unit addresses. You might use LNK to connect together libraries for two separate but related products installed on two different volumes of a domain (or indeed, on two volumes on separate domains). You can also use LNK if one library is resident on RAM disk and the other is on diskette or hard disk. This is usually necessary because most RAM disks are too small to hold the complete library, and so you can only put the frequently used overlays in the RAM disk.

The dialogue is the same as for the OFF instruction, except that there is an additional prompt to determine the unit to be occupied by the programs not in the library. For example, to introduce a stub to indicate that file SALES, currently on unit 204, is to be off-line on SAPRG1 on unit 109:

```
$95 LIBRARY MAINTENANCE
:OFF FROM UNIT:204 VOLUME:SAPRG1 ON UNIT:109
$95 FILE:SALES OFFLINED
$95 LIBRARY MAINTENANCE
```

If the program is already on the correct volume and unit you can simply key <CR> to the "volume" and "on unit " prompts. If the file name you supply is a library, you will be prompted for each member as for the OFF instruction.

A stub introduced by LNK will replace one with the same member name which is already present in the library. However, a stub never replaces a program that is already in the library. If you link to a library that already contains stubs, the information in these stubs will be copied to stubs in the target library.

21.16.3 DLN - Delete Stub Links

The DLN instruction allows you delete stubs for a single linked member or a selected group of members according to the volume-id and unit criteria. The dialogue is as follows:

```
$95 LIBRARY MAINTENANCE
:DLN
$95 UNLINK STUBS VOLUME:volume-id UNIT:unit-id
$95 MEMBER:member-id
```

If you want to delete a single line you must key <CR> to the volume and unit prompts and key the member name to the member prompt. For example, to delete the link member SA100 you must key:

```
$95 LIBRARY MAINTENANCE
:DLN
$95 UNLINK STUBS VOLUME:<CR> UNIT:<CR>
$95 MEMBER:SA100 DELETED
```

To delete all linked members found on a specific volume-id you must key the volume-id to the volume-id prompt, <CR> to the unit-id prompt and <CTRL B> to the member prompt. The dialogue to delete all links to volume SAPRG1 would be as follows:

```
$95 LIBRARY MAINTENANCE
:DLN
$95 UNLINK STUBS VOLUME:SAPRG1 UNIT:<CR>
$95 MEMBER:<CTRL B>
      SA100 DELETED
      SA200 DELETED
```



```

.....
.....
SA950 DELETED

```

To delete members with links to a specified unit-id you must key <CR> to the volume prompt, the unit to the unit-id prompt and <CTRL B> to the member prompt. Should you only want to delete members with both a specific unit and volume you must reply to both volume and unit prompts and key <CTRL B> to the member prompt.

21.16.4 Operating Notes

Remember that as each file of a dispersed library is built, **every member** belonging to the total library must be held as an individual file on the same volume, or have a stub created for it by the OFF instruction. Failure to obey this rule will mean that either some members, or some stubs, will be missing - a situation which will normally lead to an unexpected program prompt appearing.

If a stub contains wrong information Global System Manager may issue a mount prompt that the operator is unable to honour, or a program prompt if the volume exists but does not contain the member in question. If Global System Manager detects a stub for a member, mounts the indicated volume, and then finds only another stub, it terminates the load operation with a stop code: A stub must always identify the volume upon which the code of its member resides.

Although you can use DEL to delete a stub, the other instructions that can affect individual members (e.g. COP, MER, COM and REN) do not operate on stubs. In consequence members represented only by stubs are omitted from selections obtained by keying <CR> or <CTRL B> to the MEMBER: prompt. If you request to copy or merge a specific member, but there is only a stub for it, the instruction will be terminated with the NOT FOUND warning message. The same will occur if you attempt to compare a stub with a real member, or rename a member which is only represented by a stub within the target library.

21.16.5 Example 1 - Interconnecting Two Library Files

Suppose \$LIB has already been used to create two library files named P.SA, which occupy diskettes named SAPRG1 and SAPRG2, because there is not enough room to hold the entire library on a single volume. The two files have been created by copying or merging members in the normal way. They are quite suitable for development working where program prompts are acceptable. Now, however, it is decided to interconnect them using the OFF instruction so that mount prompts appear whenever a volume change is needed. Both library files have sufficient spare space (46 bytes per stub) to allow the update to take place. SAPRG1 is on 204, and SAPRG2 on 205:

```

GSM READY:$LIB
$95 TARGET LIBRARY:P.SA UNIT:204
$95 PROCESS SALES LEDGER PART ONE OF 14/02/88?:Y
$95 LIBRARY MAINTENANCE
:OFF FROM UNIT:205 ON VOLUME:<CR> (SAPRG2 will be used)
$95 FILE:P.SA MEMBER:<CTRL B>
.....
..... (Stubs added to P.SA on SAPRG1 for offline members on SAPRG2)
.....
$90 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:P.SA UNIT:205
$95 PROCESS SALES LEDGER PART TWO OF 14/02/88?:Y
$95 LIBRARY MAINTENANCE
:OFF FROM UNIT:204 ON VOLUME:<CR> (SAPRG1 will be used)
$95 FILE:P.SA MEMBER:<CTRL B>

```

Chapter 21 - Library Maintenance (\$LIB & \$LIBM)

```
.....  
..... (Stubs added to P.SA on SAPRG2 for offline members on SAPRG1)  
.....  
$95 LIBRARY MAINTENANCE  
:END  
$95 TARGET LIBRARY:<ESCAPE>  
GSM READY:
```

21.16.6 Example 2 - Interconnecting a Third File

Some little time after the previous example, more sales ledger programs have been developed, and the new components occupy library P.SA on SAPRG3. It is decided to interconnect the new library file with the existing two. Initially SAPRG1 is on 204, and SAPRG3 on 205:

```
GSM READY:$LIB  
$95 TARGET LIBRARY:P.SA UNIT:204  
$95 PROCESS SALES LEDGER PART ONE OF 14/02/88?:Y  
$95 LIBRARY MAINTENANCE  
:OFF FROM UNIT:205 ON VOLUME:<CR> (SAPRG3 will be used)  
$95 FILE:P.SA MEMBER:<CTRL B>  
.....  
..... (Stubs added to P.SA on SAPRG1 for offline members on SAPRG3)  
.....  
$95 LIBRARY MAINTENANCE  
:END  
(Replace SAPRG1 by SAPRG2 before replying)  
$95 TARGET LIBRARY:P.SA UNIT:204  
$95 PROCESS SALES LEDGER PART TWO OF 14/02/88?:Y  
$95 LIBRARY MAINTENANCE  
:OFF FROM UNIT:205 ON VOLUME:<CR> (SAPRG3 will be used)  
$95 FILE:P.SA MEMBER:<CTRL B>  
.....  
..... (Stubs added to P.SA on SAPRG2 for offline members on SAPRG3)  
.....  
$95 LIBRARY MAINTENANCE  
:END  
$95 TARGET LIBRARY:P.SA UNIT:205  
$95 PROCESS SALES LEDGER PART THREE OF 25/03/88?:Y  
$95 LIBRARY MAINTENANCE  
:OFF FROM UNIT:204 ON VOLUME:<CR> (SAPRG2 will be used)  
$95 FILE:P.SA MEMBER:<CTRL B>  
.....  
..... (Stubs added to P.SA on SAPRG3 for offline members on SAPRG2 and,  
..... because of secondary stubs, for members on SAPRG1)  
.....  
$95 LIBRARY MAINTENANCE  
:END  
$95 TARGET LIBRARY:<ESCAPE>  
GSM READY:
```

This dialogue therefore performs three OFF operations:

- OFF P.SA from SAPRG3, to include the required new stubs in P.SA on SAPRG1;
- OFF P.SA from SAPRG3, to include the required new stubs in P.SA on SAPRG2;
- OFF P.SA from SAPRG2 to include the stubs needed by the new P.SA library on SAPRG3.

Note that the fourth operation that you might have expected:

- OFF P.SA from SAPRG1 to include the stubs needed by the new P.SA library on SAPRG3.

is in fact unnecessary, because secondary stubs already present in P.SA on SAPRG2 establish the necessary interconnection to SAPRG1 during the third OFF operation.

21.16.7 Example 3 - Installing Part of a Library on RAM Disk

Suppose that a suite of programs has been split by \$LIB into a small library of frequently used overlays, P.SA1, and a second library of the remaining components, P.SA. This has been done so that the programs in P.SA1 can be held in RAM disk when installed onto a suitable computer.

The libraries are to be installed on volume SAPROG on unit 204. However, before the programs are run, P.SA2 will be copied to the RAM disk (\$\$WORK on 109) and renamed as P.SA. In order that the programs will be automatically loaded from the correct unit, the libraries must be linked as follows:

```
GSM READY:$LIB
$95 TARGET LIBRARY:P.SA UNIT:204
$95 PROCESS SALES LEDGER MAIN PART OF 14/02/88?:Y
$95 LIBRARY MAINTENANCE
:LNK FROM UNIT:<CR> ON VOLUME:$$WORK ON UNIT:109
$95 FILE:P.SA1 MEMBER:<CTRL B>
.....
..... (stubs added to P.SA)
.....
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:P.SA1 UNIT:204
$95 PROCESS SALES LEDGER COMMON OVERLAYS OF 14/02/88?:Y
$95 LIBRARY MAINTENANCE
:LNK FROM UNIT:<CR> ON VOLUME:<CR> ON UNIT:<CR>
$95 FILE:P.SA MEMBER:<CTRL B>
.....
..... (stubs added)
.....
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:
```

21.17 Job Management Support

The features described in this section are intended for use when the librarian runs under Job Management. They allow you to check that the correct output volume is loaded before processing a target library, and that specific input volumes are mounted when required. There is special support for the COP instruction, so that identical dialogue can be used both to add new members or replace existing ones.

21.17.1 Mounting the Target Library Volume

If you key the special response <CTRL C> to the target library prompt \$LIB will ask you for a volume-id and then repeat the prompt. For example:

```
GSM READY:$LIB
$95 TARGET LIBRARY:<CTRL C> VOLUME-ID:SAPRGA
$95 TARGET LIBRARY:
```

Then, when you have identified the target library, \$LIB will open it using volume-id checking, and if the volume you have specified is not on the unit you have indicated, it will be requested by a normal mount prompt. For instance, the dialogue introduced above might continue:

```
$95 TARGET LIBRARY:P.SA UNIT:204
PLEASE MOUNT SAPRGA ON 204 - HARD DISK AND KEY <CR>:
```

A mount prompt always appears, even when the rest of the responses supplied to \$LIB are obtained from the dialogue table, since it is one of the special ones which bypass job management. The operator must key Y, <CR> (or any single character apart from N) to cause the job to continue.

The mount prompt will be redisplayed, over and over again if necessary, until the correct volume is loaded. However, a reply of N causes the mount operation to be abandoned and, if job management was in control, it will be terminated.

21.17.2 I - Mount Input Volume

The I instruction enables you to control the mounting of a specified input volume required for subsequent library maintenance operations. It prompts you for the volume-id, which it remembers so that it can open the next file specified by an instruction using volume-id checking. This will cause a mount prompt to appear if the volume is not already present on the indicated unit. For example, suppose you wish to copy the file named A1 from volume OURDEV on 204 to be the target library on 205. You can use the I instruction to control the mounting of OURDEV as follows:

```
$95 LIBRARY MAINTENANCE
:I VOLUME-ID:OURDEV
$95 LIBRARY MAINTENANCE
:COP UNIT:204
$95 FILE:A1
```

If at this stage OURDEV is not present on unit 204, the following mount prompt will appear:

```
PLEASE MOUNT OURDEV ON 204 - HARD DISK AND KEY <CR>:
```

As explained above, since this is one of the special prompts that bypass job management it will always be displayed on the console, even when the other responses to \$LIB are supplied in the dialogue table. The operator must load the requested volume and key Y, <CR> (or any single character apart from N) to continue. A reply of N terminates job management.

21.17.3 The COP Instruction under Job Management

You will recall from section 21.12.1 that when the COP instruction is used to replace an existing member on the target library with an update the following prompt is displayed:

```
MEMBER ALREADY EXISTS - DELETE?:
```

allowing you to choose whether or not to replace the member in question. When \$LIB runs under job management a response of Y is automatically supplied to this prompt. This allows you to code exactly the same dialogue, whether you intend to add a new member, or replace an existing one. For example:

```
$95 LIBRARY MAINTENANCE
:COP FROM UNIT:204
$95 FILE:A1
```

will cause A1 to either be added to the target library, if it is not already present, or replaced with the new version, if it is.

21.17.4 Example Job Management Dialogue

In the previous section we described how a sequence of COP and OFF instructions could be used to install a dispersed library named P.SA on newly initialised volumes named SAPRG1 and SAPRG2. P.SA is created from libraries P.SAA and P.SAB residing on distribution volumes named SAA and SAB respectively. The following dialogue could be encoded in a job description so that the installation process could be performed under job management. The new volumes are to be created on unit 204, the existing ones read from 205:

```
GSM READY:$LIB
$95 TARGET LIBRARY:<CTRL C> VOLUME-ID:SAPRG1
$95 TARGET LIBRARY:P.SA UNIT:204
$95 NEW?:Y SIZE:0
$95 TITLE:DISPERSED LIBRARY PART 1
$95 LIBRARY MAINTENANCE
:I VOLUME-ID:SAA
$95 LIBRARY MAINTENANCE
:COP FROM UNIT:205
$95 FILE:P.SAA MEMBER:<CTRL B>
$95 LIBRARY MAINTENANCE
:I VOLUME-ID:SAB
$95 LIBRARY MAINTENANCE
:OFF FROM UNIT:205 ON VOLUME:SAPRG2
$95 FILE:P.SAB MEMBER:<CTRL B>
$95 LIBRARY MAINTENANCE
:TRU NEW SPARE SPACE:<CR>
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:<CTRL C> VOLUME-ID:SAPRG2
$95 TARGET LIBRARY:P.SA UNIT:204
$95 NEW?:Y SIZE:0
$95 TITLE:DISPERSED LIBRARY PART 2
$95 LIBRARY MAINTENANCE
:COP FROM UNIT:205
$95 FILE:P.SAB MEMBER:<CTRL B>
$95 LIBRARY MAINTENANCE
:I VOLUME-ID:SAA
:OFF FROM UNIT:205 ON VOLUME:SAPRG1
$95 FILE:P.SAA MEMBER:<CTRL B>
$95 LIBRARY MAINTENANCE
:TRU NEW SPARE SPACE:<CR>
$95 LIBRARY MAINTENANCE
:END
$95 TARGET LIBRARY:<ESCAPE>
GSM READY:
```

You may find it instructive to compare this with the original example in 21.16.6 where comments were used to describe the volume mounting requirements and the effect of the various COP and OFF instructions.

21.18 \$LIBM

The \$LIBM utility may be used to create a new library by merging the members from one, or more, existing libraries.

You are first prompted for the new library name, unit and title. If the library already exists, you will be asked if you want to delete it. Note that \$LIBM behaves differently from \$LIB when it encounters an existing library: \$LIBM prompts you to delete the file; \$LIB attempts to open the existing library.

Once the library details have been supplied you will be prompted for files and libraries to be merged into the new library. For example:

```
$40 NEW LIBRARY :P.SA UNIT:204
```

Chapter 21 - Library Maintenance (\$LIB & \$LIBM)

```
$40 TITLE:SALES LEDGER
$40 MERGE:P.SA UNIT:205
$40 MERGE:P.SA1 UNIT:205
$40 MERGE:<CR>
$40 LIBRARY MERGE COMPLETE
```

If you supply the name of a library ALL the members from that library will be merged into the new library without prompting for individual members. Note that the dialogue for \$LIBM is significantly different from the dialogue required to perform a \$LIB merge (see section 21.12.2).

22. \$LOG - Event Log Viewing

The event logging system allows events within Global System Manager to be recorded and reported. Events are initially recorded in \$\$LOG, a relative-sequential file. System programs, applications and other components of Global System Manager write records to \$\$LOG. Because relative-sequential files are inflexible as a basis for producing reports, \$\$LOG is merged into a DMAM database, \$\$MLOG, before reports are produced. \$LOG maintains \$\$LOG and \$\$MLOG and allows you to produce a range of reports.

If the Event Logging option was not selected when Global System Manager was installed you can use the "Install extra facilities" option from the \$CUS System Maintenance menu to install Event Logging without the need to re-install Global System Manager (see section 6.1.4.13.1 in the Global System Manager Manual).

To de-activate Event Logging at any time, simply remove the \$LG logical unit assignment using the \$CUS "Permanent Unit Assignment" customisation.

22.1 The \$LOG Menu

Before displaying its main menu, \$LOG automatically merges the contents of \$\$LOG into the master log database file, \$\$MLOG, and then empties \$\$LOG. \$LOG then displays a menu which allows you print a wide range of reports (see figure 22.1). In addition, \$LOG provides facilities to selectively discard records from the \$\$MLOG database, or to purge the database completely.

22.2 The Purge Function

The purge function allows you to clear records from the \$\$MLOG database which are no longer of interest. You can purge records using a wide range of criteria. For example, you can purge all records before a certain date thus preserving only the most recent system history. To display the purge selection screen key 1 at the main menu, the screen in Figure 22.2 is then displayed.

The upper part of the purge screen gives details about the current usage of \$\$MLOG. The middle section of the screen gives details of a series switches which allow you to control the types of log records that are purged. Lastly, by modifying the "purge prior to" date, you can purge all out of date records.

To modify the switches that control the types of record to be purged key A to the following prompt:

```
Key Amend, Date, Purge, <ESC> to abandon:
```

Only the log records concerning backup history are normally preserved, but you can preserve other types of record by changing the appropriate switch from Y to N.

To purge log records created before a given date, key D to the purge screen baseline prompt and specify the date.

To start the purge operation key P in response to the baseline prompt.

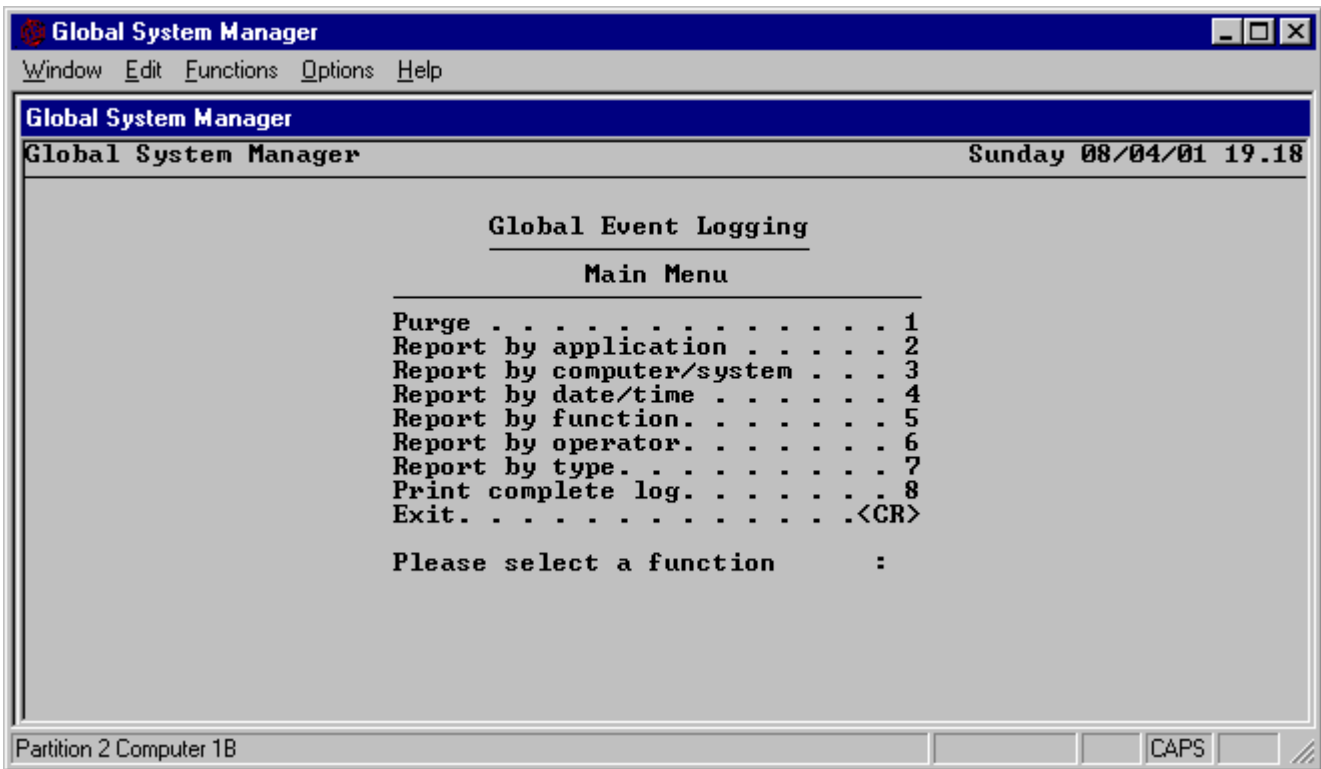


Figure 22.1 - Initial \$LOG menu

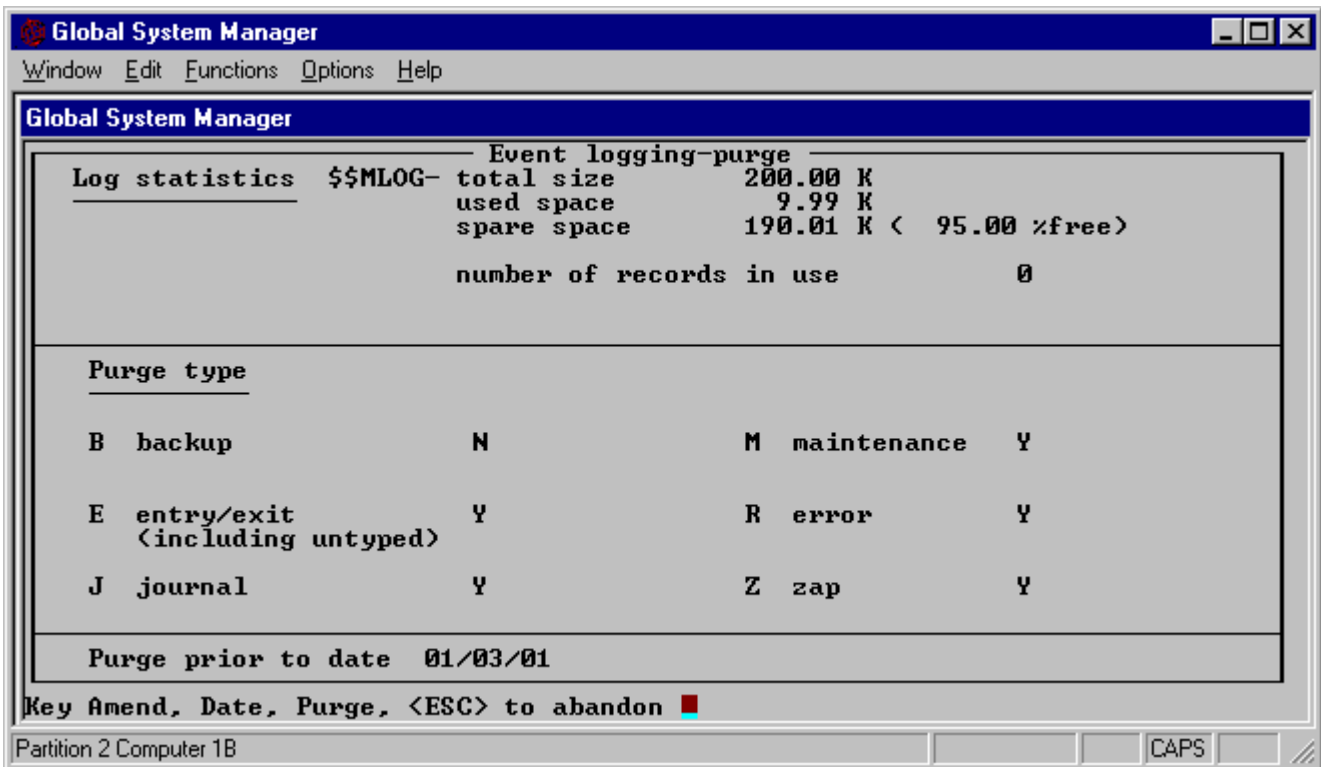


Figure 22.2 - Purge selection screen

22.3 Producing Reports

Menu entries 2 to 8 allow you to produce reports using a wide range of criteria, or to print the whole of the event log. You can produce selective reports by:

- Application;
- Computer;
- Date/time;
- Function;
- Operator;
- Record type.

The selective reports can be printed to the standard printer unit \$PR, displayed on the screen, or both displayed and printed. The "print complete log" option (menu entry 8) always sends its output to \$PR.

The selective reports are printed on the basis of keys associated with each of the log records. The keys are of varying types, but for each report you are prompted for a start and an end value for the key. The reports will only contain details of the records whose keys fall within the range you specify.

22.3.1 Report by Application

A 2-character application code is defined for each of the Global 2000 products (e.g. OZ for Global Organizer). At present only a few Global 2000 applications write event log records to \$\$LOG, but the Global System Manager menu system can be configured to write to the event log when you select a menu entry (see Chapter 7 of the Global System Manager Manual).

A large number of Global System Manager utilities write records to the event log. The application code for these programs is a \$ followed by an upper or lower-case letter. Table 22.1 lists the application codes used by Global System Manager.

22.3.2 Report by Computer

The report is produced on the basis of a single-character computer-id.

22.3.3 Report by Date/time

Records are selected on the basis of their date and time of creation.

22.3.4 Report by Function

The function code serves to identify an application function. These codes are documented in the application manuals.

22.3.5 Report by Operator

The report is produced on the basis of the 4-character operator-id used when you sign on (either explicitly or when an automatic sign-on takes place).

Application code	Description
\$\$	Global System Manager load or operator sign-on

\$A	\$AUTH
\$C	\$CUS
\$D	\$DEBUG and program check handling
\$E	Operator sign-off
\$M	\$MH (the menu handler)
\$S	\$STATUS
\$T	\$TAPE
\$Z	\$ZAP
\$d	DMAM Utility
\$e	\$MONITOR - I/O errors
\$r	\$RESTORE
\$s	\$SAVE
\$t	\$TDUMP
BE	Extension Kit installation
BT	Common Toolkit installation
CF	Configurator installation or configuration file update
ED	Speedbase Development installation
FC	File Converters installation
IE	Assembler Interface Kit installation
MK	Cobol Programming Kit installation

Table 22.1 - System Application Codes**22.3.6 Report by Type**

The log record type serves to identify the type of record being created, and the nature of the event it records. Records can be selectively purged from \$\$MLOG on the basis of the log record type. Table 22.2 lists the log type records used by Global System Manager.

Code	Type	Description
B	Backup	Used by backup programs and procedures (to log both backups and restores).
E	Entry/exit	Used by Global System Manager sign-on and sign-off, and when programs are invoked by the menu system (\$MH).
J	Journal	Used by applications to form part of a transaction audit trail.
M	Maintenance	Used by system maintenance functions and by a number of Global System Manager utilities.
R	Error	Used by the Global System Manager I/O error handling mechanisms, on entry to the debugging system, or for application detected errors. If an event log record has been created as a result of a STOP CODE, EXIT CODE or PROGRAM CHECK the code number (e.g. EXIT CODE 1001 AT ED72) and Last Program Loaded will be included in the message.
Z	Zap/Update	Used by any process which applies an update to the system. Specifically, \$ZAP, the Autozap mechanism, and product installation.

Table 22.2 - Log Type Records

23. \$M - Send Message to Other Users

The \$M command allows you to send a message of up to 30 characters to a specific user, users, or group of users. After clearing the screen, \$M displays the following prompt:

```
Send message to other operators
Key operator-id, ? to list, <CTRL A> for group, <CTRL B> to broadcast, <ESC>:
```

The list option displays the operator-id's of all the users currently logged onto the computer (or network). If a valid operator-id is given the operator-id prompt is repeated until the reply is <CR>. The following example dialogue sends two messages to operators AJU1 and AJU2:

```
Send message to other operators
Key operator-id, ? to list, <CTRL A> for group, <CTRL B> to broadcast,
<ESC>:AJU1

Key operator-id:AJU2
Key operator-id:<CR>
Key message, <ESC>:THIS IS AN EXAMPLE MESSAGE
Key message, <ESC>:THIS IS ANOTHER MESSAGE
Key message, <ESC>:<ESC>
```

In a network environment, the broadcast option allows the same message to be transmitted to all the users on a particular computer, or to all users on the network. The following example illustrates how \$M may be used to send a message to all the operators connected to computer-X:

```
Send message to other operators
Key operator-id, ? to list, <CTRL A> for group, <CTRL B> to broadcast,
<ESC>:<CTRL B>

Computer-id, <CTRL B> for all (A ):X
Key message, <ESC>:THIS IS A TEST MESSAGE
```

The following example illustrates how \$M may be used to send a message to all the users on a network:

```
Send message to other operators
Key operator-id, ? to list, <CTRL A> for group, <CTRL B> to broadcast,
<ESC>:<CTRL B>

Computer-id, <CTRL B> for all (A ):<CTRL B>
Key message, <ESC>:THIS IS ANOTHER MESSAGE
```

If the Operator Group option is enabled then the <CTRL A> option allows you to send a message to all the members of a particular group. For example:

```
Send message to other operators
Key operator-id, ? to list, <CTRL A> for group, <CTRL B> to broadcast,
<ESC>:<CTRL A>

Key Group-id, ? to list:SALES
Key message, <ESC>:THIS IS A SALES MESSAGE
```

The list option displays all the currently available groups as defined by the \$GROUP utility (see Chapter 17).

The message prompt is repeated, to allow you to send multiple messages to a single operator, a number of operators, a group of operators, all the operators on a computer or all the operators on a network, until the terminator to the message is either <ESC>, to exit \$M, or <CTRL A>, to return to the initial "Key operator-id" prompt. For Global System Manager (Unix) configurations the dialogue is slightly different to allow you to send multiple messages to a single operator, a

number of operators, a group of operators, all the operators on a System or all the operators on a Unix computer.

Depending on the length of the message that you supply, useful system information may be appended to the text of the message. If the length of the message is 19 characters, or less, the message text is prefixed by the current time (i.e. *hh:mm*) and the operator-id of the sender (i.e. *oooo*). If the length of the message is between 20 and 25 characters, the message text is prefixed by the operator-id of the sender (i.e. *oooo*). If the length of the message is more than 25 characters no extra information is appended to the message. For example:

<i>Message keyed</i>	<i>Message sent as</i>
ABCDEFHIJKLMNOPQRS	<i>hh:mm</i> <i>oooo</i>
ABCDEFHIJKLMNOPQRS	<i>oooo</i> ABCDEFHIJKLMNOPQRST
ABCDEFHIJKLMNOPQRST	<i>oooo</i> ABCDEFHIJKLMNOPQRST
ABCDEFHIJKLMNOPQRSTUVWXYZ	ABCDEFHIJKLMNOPQRSTUVWXYZ

\$M is equivalent to the \$STATUS message (MES) command (see Chapter 37).

24. \$MAIL - Mail Utility

The \$MAIL utility can be used to send mail, take telephone messages, view existing mail and read telephone messages. Mail and telephone messages can also be read by invoking <SYSREQ> N. Telephone messages can be written (i.e. jotted) by invoking <SYSREQ> J. These two system requests are fully described in Chapter 4 of the Global System Manager Manual. In addition, whenever you sign-on, Global System Manager will inform you if there is any mail for you. This feature is optional and can be turned off within the Operator Customisation option of \$MAIL (see section 24.6.1.1).

Mail can be sent to all users, a specific operator, a group of operators or all the users of a particular computer or system. The mail details are held in the \$\$MAIL database on unit \$ML.

The \$MAIL reorganise mail facility may be used by supervisors to selectively discard records from the mail file and to reorganise the \$\$MAIL database.

24.1 The \$MAIL Menu

On entering \$MAIL the main menu shown in figure 24.1 will be displayed. The various options are described in the following sections.

24.2 Send Mail

The send mail function allows you to send four types of mail:

- operator mail;
- group mail;
- computer or system mail;
- broadcast messages.

When this option is selected the standard mail message screen as shown in 24.2 is displayed.

24.2.1 Entering the Message Details

The standard mail message details screen always includes the operator-id of the sender (i.e. your operator-id), and the current date and time. The following message details can now be entered.

24.2.1.1 Subject

A 30 character short description of the mail message.

24.2.1.2 Activation date

This is the first day on which the mail will be activated (i.e. the first day that the message will be listed when the recipient signs on to Global System Manager). This is usually the current date, but if, for example, you wish to leave a reminder for a future date (e.g. to leave a reminder for yourself tomorrow) then the activation date should be that future date. Mail items which have not been activated are not displayed at sign-on or when listing mail using <SYSREQ> N. However, all mail, including mail that has not been activated, is displayed when using the \$MAIL list function (see section 24.4.1).

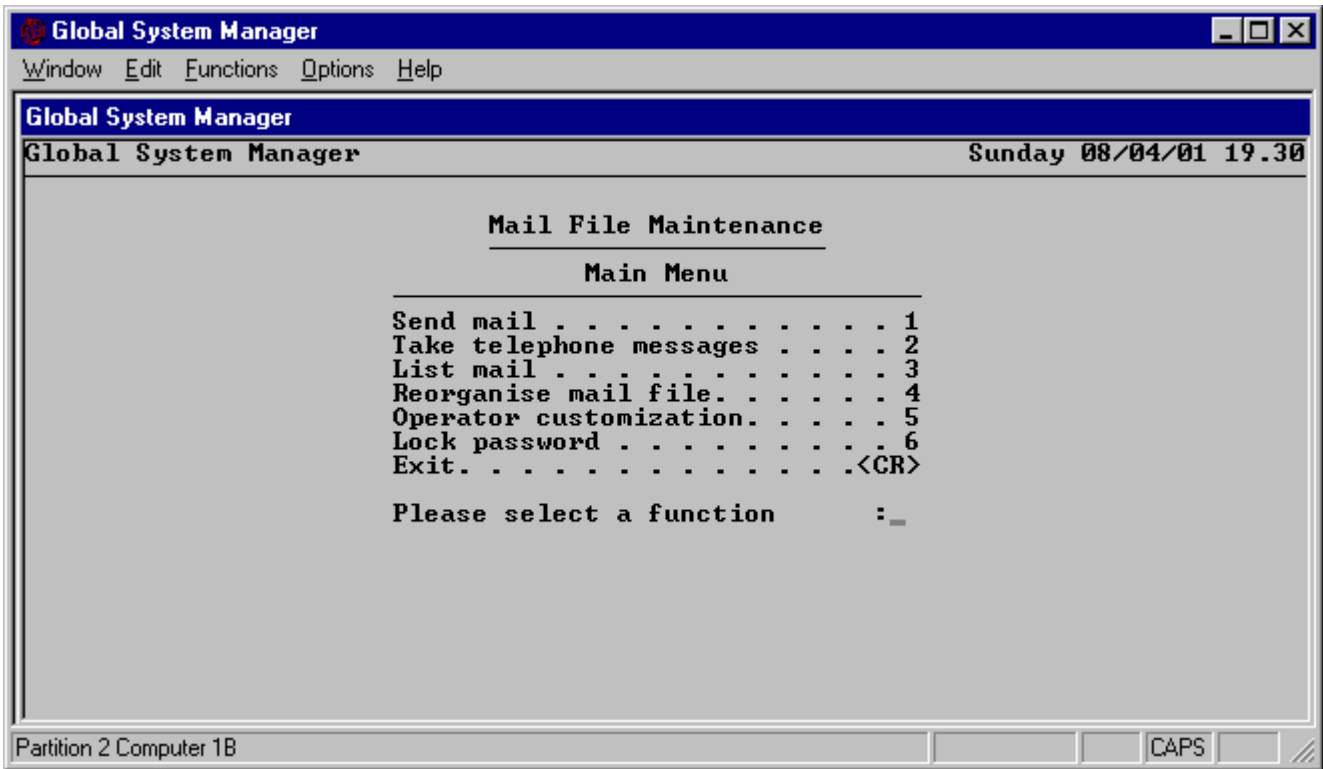


Figure 24.1 - Initial \$MAIL menu

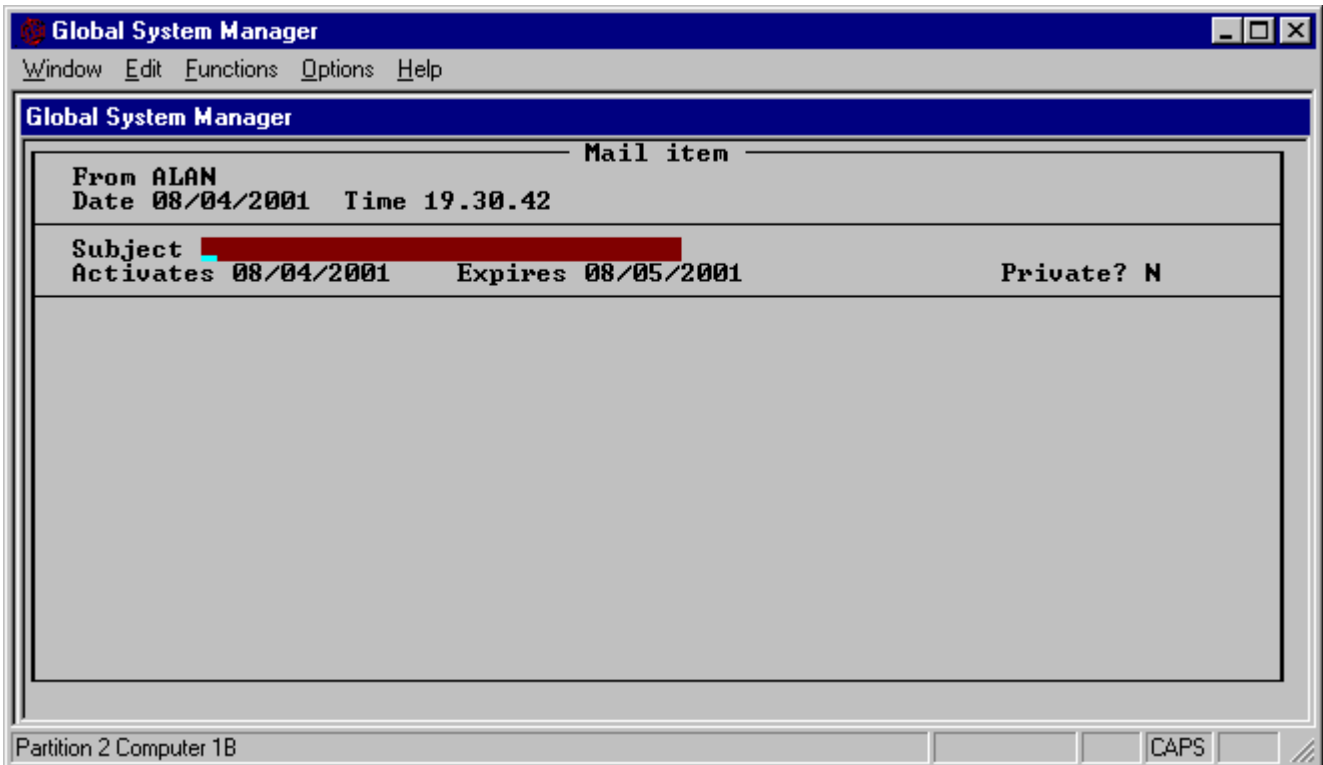


Figure 24.2 - Standard Mail Form

24.2.1.3 Expiry date

This is the date on which the mail item will expire. Expired mail items are not displayed at sign-on or when listing mail using <SYSREQ> N. However, all mail including expired mail, is displayed when using the \$MAIL list function (see section 24.4.1).

24.2.1.4 Private Flag

This flag is used to indicate whether the mail is private to the recipient of the message. For operator mail, the "private flag" prevents other operators who normally have access to the recipients mailbox (see section 24.4.1) from viewing the mail item. For group mail, the "private flag" prevents supervisors who normally have access to all group mail from reading the mail item. For computer, system or broadcast mail, the "private flag" is ignored.

24.2.1.5 Text

Up to thirty lines of text may be entered using the standard Global System Manager text editing facility (see the description of \$TED in Chapter 5 of the Global System Manager Manual).

24.2.1.6 Completing the Mail Message

After entering the text, key <ESC> to indicate that the mail item is ready to send. The following baseline prompt will be displayed:

```
Key A to amend, P to print, <CR> to send, <ESC> abandon:
```

Key A if you want to amend any details of the mail message.

Key P to obtain a print out of the mail message. The report is sent to the \$PR unit.

Key <ESC> to abandon if you decide that you do not want to send the mail item.

24.2.2 Sending the Mail Message

Key <CR> to the above baseline prompt to send the mail message. You are then prompted for the type of mail message:

```
Key Computer, Group, Mailbox, Broadcast, <ESC> to exit:
```

If you are running on a single computer configuration the 'Computer' option will not appear. If you are using Global System Manager (Unix) the 'Computer' option will be replaced by the 'System' option. Key the appropriate letter for the type of mail you want to send.

24.2.2.1 Sending Mail to a Mailbox

Key M to the above baseline prompt to send mail to one or more individual operators. You are prompted for the mailbox-id's (i.e. operator-id's) of the recipients. For example to send a mail message to both JACK and JILL use the following dialogue:

```
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:M
Key mailbox-id, ? to list, <ESC> to exit:JACK
Key mailbox-id, ? to list, <ESC> to exit:JILL
Key mailbox-id, ? to list, <ESC> to exit:<ESC>
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:<ESC>
```

Key ? to the mailbox-id prompt to list all the operators **currently** using Global System Manager. If you wish to display the entire list of operators then, provided the operator list has been set up using \$OPID (see Chapter 26), <SYSREQ> Y can be used to display and return operator-id details. <SYSREQ> Y is fully described in Chapter 4 of the Global System Manager Manual. If

the recipient is not in the operator-id file, the following warning message will appear on your status line:

```
NOT FOUND IN OPID FILE
```

All recipients of the mail message (in this example, both JACK and JILL) will be sent a status line message indicating that there is outstanding mail (see Appendix B of the Global System Manager Manual).

24.2.2.2 Sending Group Mail

Key G to the above baseline prompt to send mail to one or more groups of operators (see Chapter 17 for a description of the \$GROUP utility). You are prompted for the destination group, or groups. For example, to send a mail message to both the 1STFLOOR and 2NDFLOOR groups, use the following dialogue:

```
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:G
Key group-id, ? to list, <ESC> to exit:1STFLOOR
Key group-id, ? to list, <ESC> to exit:2NDFLOOR
Key group-id, ? to list, <ESC> to exit:<ESC>
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:<ESC>
```

You may key ? the group-id prompt to list the all the available groups.

All operators in the selected groups (in this example, all users in the 1STFLOOR and 2NDFLOOR groups) will be sent a status line message indicating that there is outstanding mail (see Appendix B of the Global System Manager Manual).

24.2.2.3 Sending Computer/System Mail

Key C to the above baseline prompt for computer mail (for all network configurations other than Global System Manager (Unix) configurations) or S for system mail (for Global System Manager (Unix) configurations). You will be prompted for the destination id of the computer or system. The computer/system prompt is repeated to allow you to send mail to more than one computer or system. For example to send a mail message to all users on computers X and Z you will need to enter the following dialogue:

```
Key Computer, Group, Mailbox, Broadcast, <ESC> to exit:C
Key computer-id, <ESC> to exit:X
Key computer-id, <ESC> to exit:Z
Key computer-id, <ESC> to exit:<ESC>
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:<ESC>
```

All users of the destination computer(s) or system(s) (in this example, computers X and Z) will receive a status line message indicating that there is outstanding computer or system mail (see Appendix B of the Global System Manager Manual).

24.2.2.4 Sending Broadcast Mail

Key B to the above baseline prompt to place a mail item on the broadcast notice board. All users currently using Global System Manager will receive a status line message indicating that there is outstanding broadcast mail (see Appendix B of the Global System Manager Manual).

24.2.2.5 Sending Mail to Different Types of Recipients

It is possible to send the mail message to various types of recipients. For example, to send a mail item to the SALES group and to users of computer X and also to the user with mailbox-id (operator-id) of AJU use the following dialogue:


```

Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:G
Key group-id, ? to list, <ESC> to exit:SALES
Key group-id, ? to list, <ESC> to exit:<ESC>
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:C
Key computer-id, <ESC> to exit:X
Key computer-id, <ESC> to exit:<ESC>
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:M
Key mailbox-id, ? to list, <ESC> to exit:AJU
Key mailbox-id, ? to list, <ESC> to exit:<ESC>
Key Mailbox, Group, Computer, Broadcast, <ESC> to exit:<ESC>
    
```

24.3 Take Telephone Messages

The take telephone messages option allows you to transcribe a telephone message for a particular operator.

24.3.1 Entering the Telephone Message

On selecting this option the standard telephone form (see figure 24.3) is displayed. This form allows you to specify the details of the telephone message.

24.3.1.1 Message For

You MUST specify the operator-id of the recipient at the 'Message for' prompt, although all other fields, described below, are optional. If the operator has not been added to the operator-id file (using \$\$OPID, see Chapter 26) the following warning message will appear on your status-line:

```
NOT FOUND IN OPID FILE
```

24.3.1.2 Private Flag

This flag is used to indicate whether the telephone message is private to the designated recipient. The "private flag" prevents other operators who normally have access to the recipients mailbox (see section 24.4.2) from viewing the telephone message.

24.3.1.3 Message Text

Up to 5 lines, each of 70 characters, can be used to take the message.

Important note: Each line is accepted individually so that the standard Global System Manager text editing facility cannot be used (contrast with taking mail messages, described in section 24.2.1.5).

If the message text is shorter than 5 lines of information, key <ESC> to complete the "Message Text" update and proceed directly to the "Message From Information" (see next section). Key <ESC> to any of the prompts described in sections 24.3.1.4 and 24.3.1.5 to proceed directly to the telephone message baseline prompt, described in section 24.3.1.6.

24.3.1.4 Message From Information

The standard form allows to specify the following information regarding the person giving you the message:

<i>Prompt</i>	<i>Description</i>
Message from	30 characters of free-format text
of	40 characters of free-format text

Phone number 20 characters of free-format text

Fax number 20 characters of free-format text

24.3.1.5 Telephone Message Options

The standard telephone message form includes the following 6 options that you can mark with a "Y" or "N" flag as appropriate:

<i>Option</i>	<i>Default flag value</i>
Telephoned?	Y
Please ring back?	N
Returned your call?	N
Will ring back?	N
Wants to see you?	N
Urgent?	N

24.3.1.6 Completing the Telephone Message

When the telephone message details have been completed the following baseline message is displayed:

Key A to amend, P to print, <CR> to send, <ESC> to abandon:

Key A to amend the message detail if, for example, you have made an error while taking the telephone message and you need to make corrections.

Key P to print a copy of the telephone message to the \$PR unit.

Key <ESC> to abandon and exit if you decide that you do not want to send the telephone message. A confirmation prompt will be displayed to which you must key Y to abandon.

24.3.2 Sending the Telephone Message

Key <CR> the baseline prompt to send the telephone message to the specified operator. A status line message, indicating that there is an outstanding telephone message, will be displayed on the recipient's screen.

You are then prompted to send the telephone message to additional recipients. For example, if you consider the message that you have just taken to be appropriate for another operator, RICK, use the following dialogue:

Key A to amend, P to print, <CR> to send, <ESC> to abandon:<CR>
Additional recipient, ? to list, <ESC> to exit:RICK

You may specify several additional recipients for the telephone message. Key ? to obtain a list of all the operators who are currently using Global System Manager. If a full operator list has been established using \$OPID (see Chapter 26) you can view this list using <SYSREQ> Y (see Chapter 4 of the Global System Manager Manual).

24.4 List Mail

The list mail function allows you to view mail items or telephone messages. When you select the list mail function, the list mail menu is initially displayed (see figure 24.4). The various options are described below.

24.4.1 List Mailbox

This option allows you to list your personal mail. The initial screen shows the list of your outstanding personal mail with the most recently sent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column.

The following baseline prompt is displayed:

Key Change, Omit, Print, up/down to select, <CR> to examine, ESC> to exit:

Key <UPARROW> or <DOWNARROW> to select a different current mail item. Key P to print the list of mail items to the \$PR unit. Key O to omit those mail items that have been marked as "read" from the list.

24.4.1.1 Examining a Mail Item

When you have used the <UPARROW> and <DOWNARROW> keys to select the mail item to view, key <CR> to examine the chosen item. The mail item will be displayed with the following baseline prompt:

Key Amend, Delete, Next, Print, Redirect, Send Reply, <CR> for next, <ESC>:

Key A to amend the "read flag". This flag indicates whether or not you have read the mail item. If you set this flag to Y the mail item is marked as read and will not be displayed at sign-on, or by <SYSREQ> N.

Key N (for Next Page) to toggle between the first half and the second half of the 30 lines of mail text. Key D to delete the mail item from your list. Key P to print the current mail item to the \$PR unit.

Key <CR> to leave the current mail item and retrieve the next mail item (or return to the list screen if there are no further mail items in the list).

Key R to redirect the current mail item to another operator for review. You must specify the name of the operator to whom you want to redirect the mail item. The mail item will be deleted from your list and will be sent to the mailbox of the operator to whom it was redirected. Key ? to the "redirect mail" prompt to obtain a list of all the operators who are currently using Global System Manager. If a full operator list has been established using \$OPID (see Chapter 26) you can view this list using <SYSREQ> Y (see Chapter 4 of the Global System Manager Manual).

Key S to send a reply to the author of the current mail message. You are given the opportunity to key up to 30 lines of text. These lines can be edited using the standard text editing facilities (see the description of \$TED in Chapter 5 of the Global System Manager Manual). The reply will be sent to mailbox of the sender of the current mail item and a status line message will be displayed on her, or his, screen to indicate there is outstanding mail.

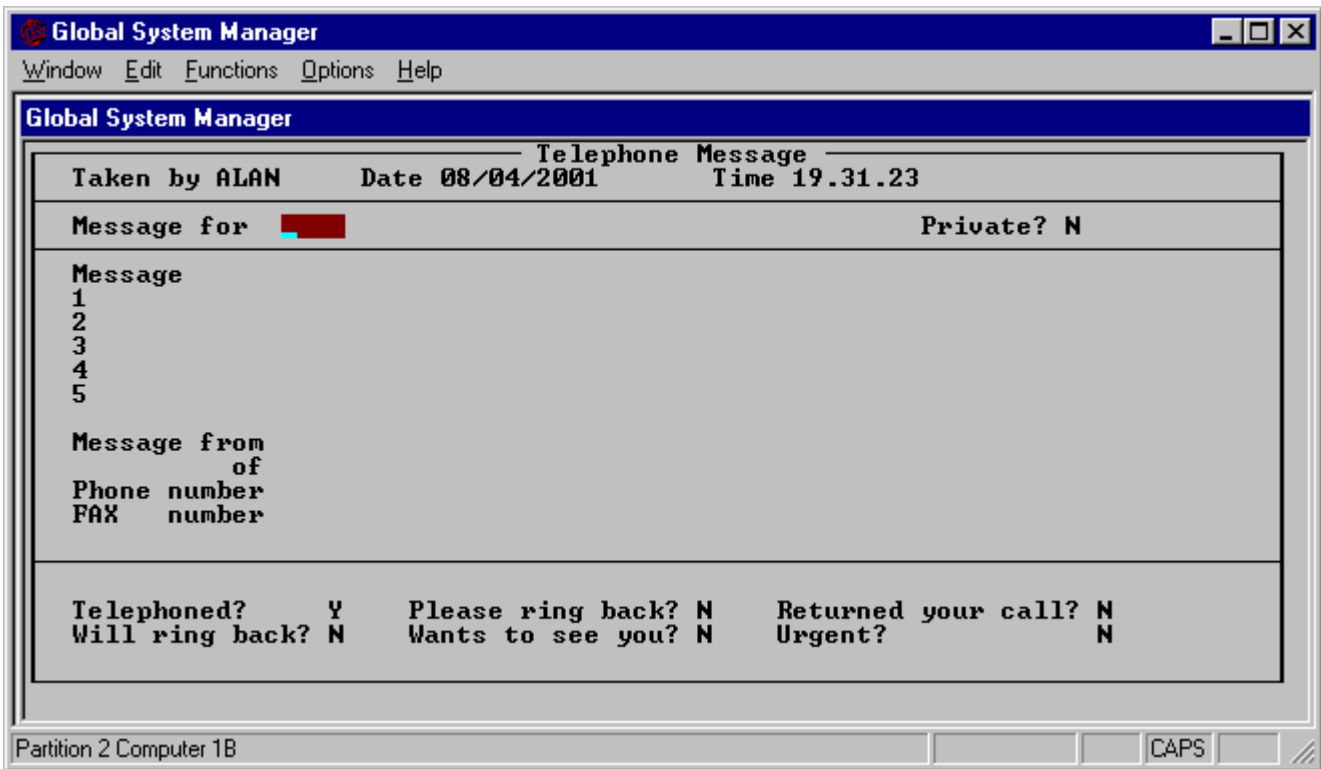


Figure 24.3 - Telephone Message Form

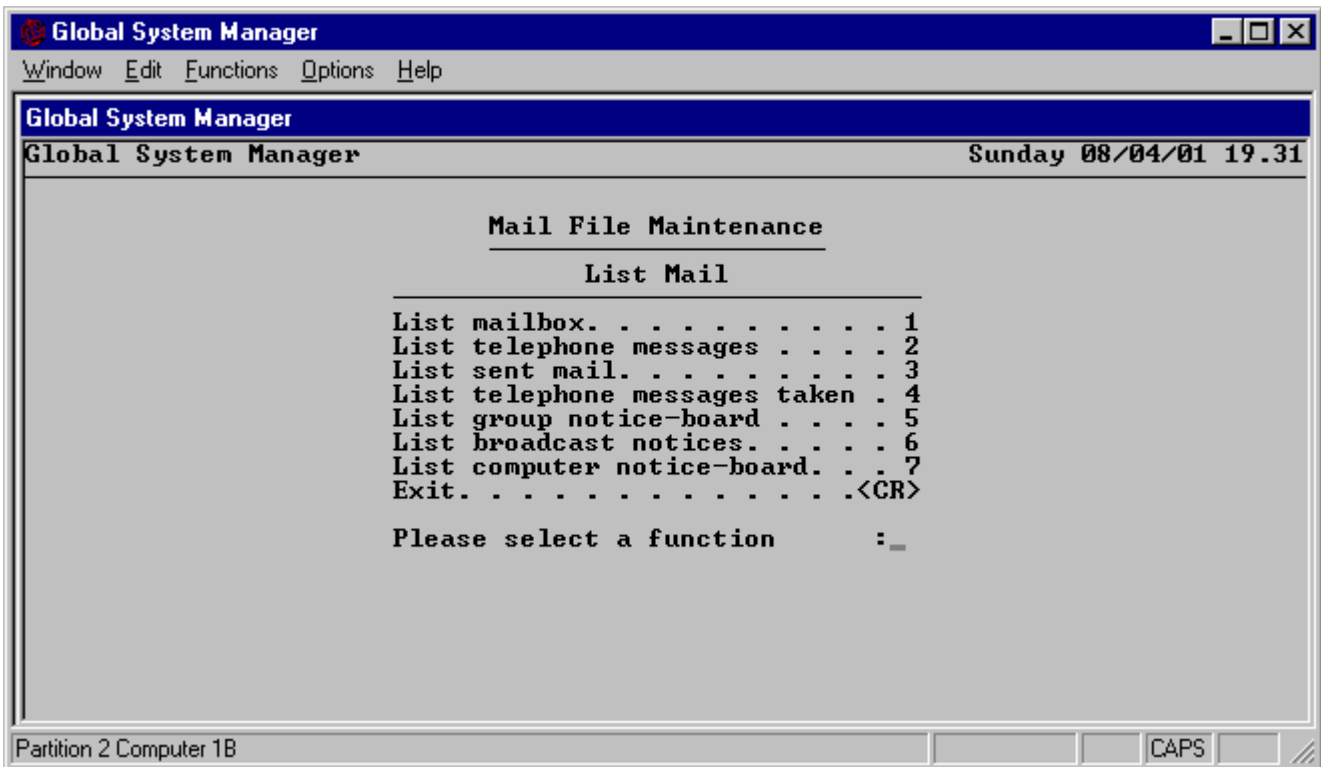


Figure 24.4 - The List Mail Menu

24.4.1.2 Changing the Mailbox

Key C to change the current mailbox (on entry to the List Mailbox option of \$MAIL the mailbox selected is your own). You are only allowed to list the mail items for another operator provided you have access to the operators mailbox (see section 24.6.1.4). You are prompted for the

mailbox (operator-id) of the user whose personal mail you wish to view. You may inspect all the mail items sent to this operator except for those marked as "private".

If you do not have access to a mailbox, the following warning message is displayed:

```
Key Change, Omit, Print, up/down to select, <CR> to examine, <ESC> to exit:C
Mailbox, <ESC> to exit:PRIV
ACCESS DENIED Key <CR>
```

24.4.2 List Telephone Messages

This option allows you to inspect any telephone messages that have been taken for you by other operators. The initial screen shows the list of outstanding telephone messages, the most recent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column.

The following baseline prompt is displayed:

```
Key Change, Omit, Print, up/down to select, <CR> to examine, <ESC> to exit:
```

Key <UPARROW> or <DOWNARROW> to select a different current telephone message. Key P to print the telephone message list to the \$PR unit. Key O to omit those telephone messages that have been marked as "read" from the list.

24.4.2.1 Examining a Telephone Message

When you have used the <UPARROW> and <DOWNARROW> keys to select the telephone message to view, key <CR> to examine the chosen item. The telephone message will be displayed with the following baseline prompt:

```
Key Amend, Delete, Print, Redirect, <ESC>:
```

Key A to amend the "read flag". This flag indicates whether or not you have read the telephone message. If you set this flag to Y the telephone message is marked as read and will not be displayed at sign-on, or by <SYSREQ> N.

Key D to delete the telephone message from your list. Key P to print the current telephone message to the \$PR unit.

Key R to redirect the current telephone message to another operator. You must specify the name of the operator to whom you want to redirect the message. The telephone message will be deleted from your list and will be sent to the operator to whom it was redirected. Key ? to the "redirect mail" prompt to obtain a list of all the operators who are currently using Global System Manager. If a full operator list has been established using \$OPID (see Chapter 26) you can view this list using <SYSREQ> Y (see Chapter 4 of the Global System Manager Manual).

24.4.2.2 Changing the Mailbox

Key C to change the current mailbox (on entry to the List Telephone Message option of \$MAIL the mailbox selected is your own). You are only allowed to list the telephone messages for another operator provided you have access to the operators mailbox (see section 24.6.1.4). You are prompted for the mailbox (operator-id) of the user whose personal mail you wish to view. You may inspect all the telephone messages sent to this operator except for those marked as "private". If you do not have access to a mailbox, the following warning message is displayed:

```
Key Change, Omit, Print, up/down to select, <CR> to examine, <ESC> to exit:C  
Mailbox, <ESC> to exit:PRIV  
ACCESS DENIED Key <CR>
```

24.4.3 List Sent Mail

This option allows you to list the mail messages sent by you. The initial screen shows the list of mail sent, with the most recent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column.

The following baseline prompt is displayed:

```
Key Print, up/down to select, <CR> to examine, <ESC> to exit:
```

Key <UPARROW> or <DOWNARROW> to select a different current mail item. Key P to print the list of main items to the \$PR unit.

24.4.3.1 Examining a Sent Mail Item

When you have used the <UPARROW> and <DOWNARROW> keys to select the mail item to view, key <CR> to examine the chosen item. The mail item will be displayed with the following baseline prompt:

```
Key Amend, Delete, Next, Print, Recipients, <CR> for next, <ESC>:
```

Key A to amend the details of the current mail item (see section 24.2.1). Key N (for Next Page) to toggle between the first half and the second half of the 30 lines of mail text. Key D to delete the mail item from your "sent mail list" and from the lists of all the recipients of the mail item. Key P to print the current mail item to the \$PR unit.

Key <CR> to leave the current mail item and retrieve the next mail item (or return to the list screen if there are no further mail items in the list).

Key R to list the list of current recipients for the mail item. The recipient option is NOT available for broadcast notices.

If the mail item is a mailbox item (i.e. not a Group, Computer/System or Broadcast message), then if a recipient has read the mail item, a "*" character will be displayed to the left of the operator-id. The recipient list (i.e. operator-id's, groups or computer/system node-id's) is displayed with the following baseline prompt:

```
Key Add, Delete, <CR> for next, <ESC> to exit:
```

Key <CR> to page through the list of current recipients.

Key A to add another recipient (i.e. operator-id, group or computer/system node-id) to the current list. You are prompted for an operator-id, group or computer/system id, as appropriate for the type of mail item.

Key D to delete a recipient (i.e. operator-id, group or computer/system node-id) from the current list. You are prompted for an existing operator-id, group or computer/system id, as appropriate for the type of mail item.

24.4.4 List Telephone Messages Taken

This option allows you to list the telephone messages that have been taken by you (using either \$MAIL or <SYSREQ> J). The initial screen shows the list of outstanding telephone messages, the most recent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column.

The following baseline prompt is displayed:

```
Key Print, up/down to select, <CR> to examine, <ESC> to exit:
```

Key <UPARROW> or <DOWNARROW> to select a different current telephone message. Key P to print the telephone message list to the \$PR unit.

24.4.4.1 Examining a Telephone Message

When you have used the <UPARROW> and <DOWNARROW> keys to select the telephone message to view, key <CR> to examine the chosen item. The telephone message will be displayed with the following baseline prompt:

```
Key Amend, Delete, Print, Recipients, <ESC>:
```

Key A to amend the telephone message. You must not attempt to amend the "Message for" field.

Key D to delete the telephone message from your telephone messages taken list, and from the telephone message lists of all the recipients. Key P to print the current telephone message to the \$PR unit.

Key R to send the telephone message to additional recipients. You are prompted for a list of operators who will be sent the current telephone message.

24.4.5 List Group Notice Board

This option allows you to display the mail sent to a particular group for which you are a member. If you are a supervisor then you may view non-private mail for those groups that you are not a member of. The initial screen shows the list of groups of which you are a member.

The following baseline prompt is displayed:

```
Key group-id, Back, Full list, <CR> for next, <ESC> to exit:
```

Key B to page backwards through the list of groups. Key <CR> to page forwards through the list of groups.

The F option, which is only available if you are a supervisor, status, will display a full list of all the groups defined in the \$\$GROUP file by the \$GROUP utility (see Chapter 17 for a description of the \$GROUP utility).

Key the name of a valid group to obtain a list of all the group messages that have been sent to that group, with the most recent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column. The following baseline prompt is displayed:

```
Key Print, up/down to select, <CR> to examine, <ESC> to exit:
```

Key <UPARROW> or <DOWNARROW> to select a different group mail message. Key P to print the group mail message list to the \$PR unit.

24.4.5.1 Examining a Group Message

When you have used the <UPARROW> and <DOWNARROW> keys to select the group mail item to view, key <CR> to examine the chosen item. The mail item will be displayed with the following baseline prompt:

Key Amend, Delete, Next, Print, <CR> to page, <ESC>:

Key A to amend the "read flag".

Key N (for Next Page) to toggle between the first half and the second half of the 30 lines of mail text. Key D to delete the mail item from the "group mail list". The delete option is only available to supervisors. Key P to print the current mail item to the \$PR unit.

Key <CR> to leave the current mail item and retrieve the next group mail item (or return to the list screen if there are no further mail items in the list).

24.4.6 List Broadcast Notice Board

This option allows you to display all the current broadcast messages. When this option is selected a list of all the broadcast messages is displayed, with the most recent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column. The following baseline prompt is also displayed:

Key Print, up/down to select, <CR> to examine, <ESC> to exit:

Key <UPARROW> or <DOWNARROW> to select a different group mail message. Key P to print the broadcast mail message list to the \$PR unit.

24.4.6.1 Examining a Broadcast Message

When you have used the <UPARROW> and <DOWNARROW> keys to select the broadcast mail item to view, key <CR> to examine the chosen item. The mail item will be displayed with the following baseline prompt:

Key Delete, Next, Print, <CR> for next, <ESC>:

Key N (for Next Page) to toggle between the first half and the second half of the 30 lines of mail text. Key D to delete the mail item from the "broadcast mail list". The delete option is only available to supervisors. Key P to print the current mail item to the \$PR unit.

Key <CR> to leave the current mail item and retrieve the next broadcast mail item (or return to the list screen if there are no further mail items in the list).

24.4.7 List Computer/System Notice Board

This option, which is only available on network or Global System Manager (Unix) configurations, allows you to display all the messages on the Computer/System notice board. When this option is selected a list of all the computer/system messages is displayed, with the most recent item at the top of the list. The current item is indicated either by a highlighted line or by a ">" character in the left hand column. The following baseline prompt is also displayed:

Key Change, Print, up/down to select, <CR> to examine, <ESC> to exit:

Key <UPARROW> or <DOWNARROW> to select a different computer/system mail message. Key P to print the computer/mail message list to the \$PR unit. The C to change option allows you to list the mail items for a different computer (or system). Key C followed by the id of the target computer (or system).

24.4.7.1 Examining a Computer/System Message

When you have used the <UPARROW> and <DOWNARROW> keys to select the computer/system mail item to view, key <CR> to examine the chosen item. The mail item will be displayed with the following baseline prompt:

```
Key Delete, Next, Print, <CR> for next, <ESC>:
```

Key N (for Next Page) to toggle between the first half and the second half of the 30 lines of mail text. Key D to delete the mail item from the "computer (or system) mail list". The delete option is only available to supervisors. Key P to print the current mail item to the \$PR unit.

Key <CR> to leave the current mail item and retrieve the next computer (or system) mail item (or return to the list screen if there are no further mail items in the list).

24.5 Reorganise Mail File

The reorganise function may only be used by supervisors. Furthermore, if there is a password (see section 24.7) it must be supplied. **The supervisor must ensure that no other users are accessing the mail file when a reorganisation is about to take place.**

The Reorganise Mail File option displays the database statistics as well as information about the number of records used in each record set (see figure 24.5). The following baseline prompt is displayed:

```
Key R to reorganise, <ESC> to exit:
```

Key R to obtain the Database Reorganise sub-menu (see figure 24.6). This menu allows you to selectively discard mail and telephone message records. Note that the term 'mail' for these entries refers to both types of mail that are targeted to a specific operator. It refers to both "mailbox mail" (see section 24.2.2.1) and "telephone messages" (see section 24.3).

24.5.1 Purge Mail Sent to Operator

This option allows you to purge all the items of mail sent to a particular operator. If the reply to the operator-id prompt is <CTRL B> you can delete all contents of all the mailboxes. **Use this option with great care!**

You are given the option to purge all operator mail items or only those items of mail that have been read by the operator. In order to proceed you must key Y to the following confirmatory message:

```
Delete mail or telephone messages sent to operator oooo?:
```

24.5.2 Purge Mail Sent by Operator

This option allows you to purge all the items of mail sent by a particular operator. If the reply to the operator-id prompt is <CTRL B> you can delete all the mail and telephone messages. **Use this option with great care!**

In order to proceed you must key Y to the following confirmatory message:

Delete mail sent by 0000?:

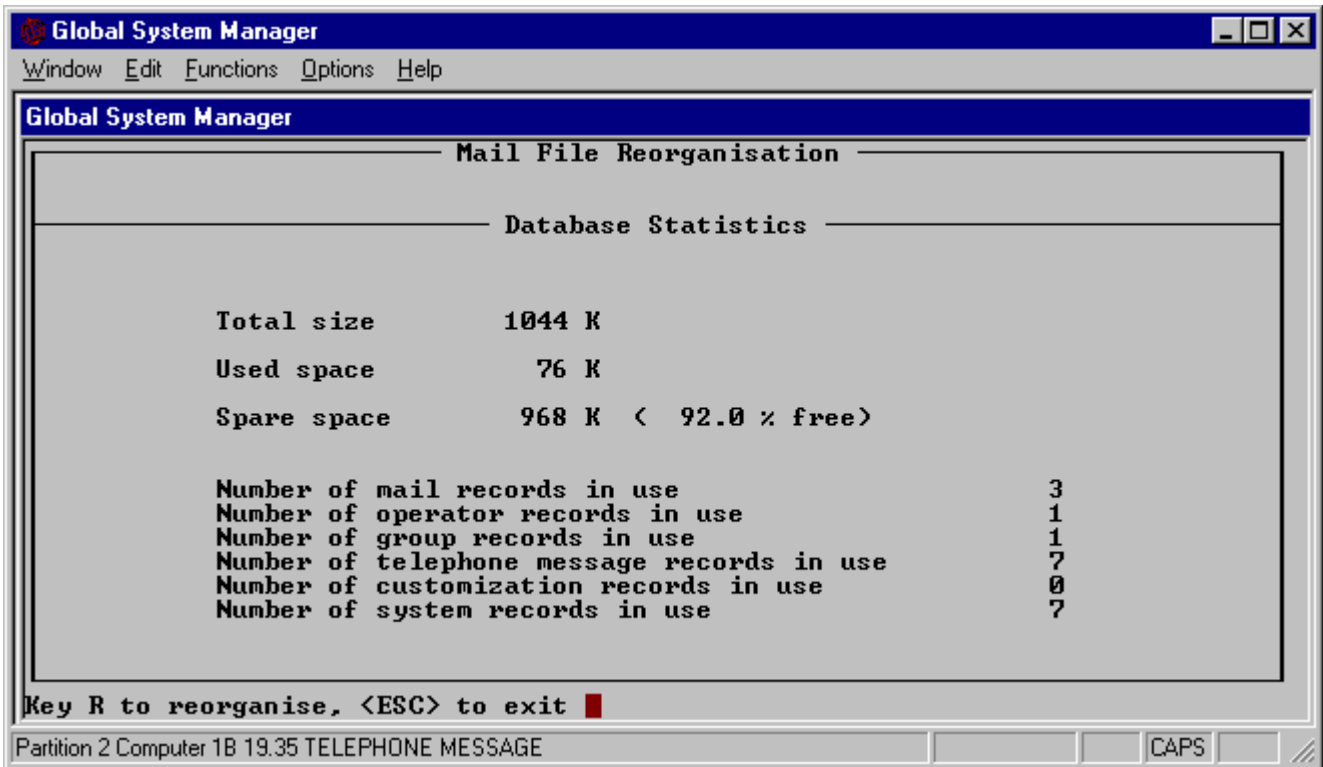


Figure 24.5 - The Statistics Screen

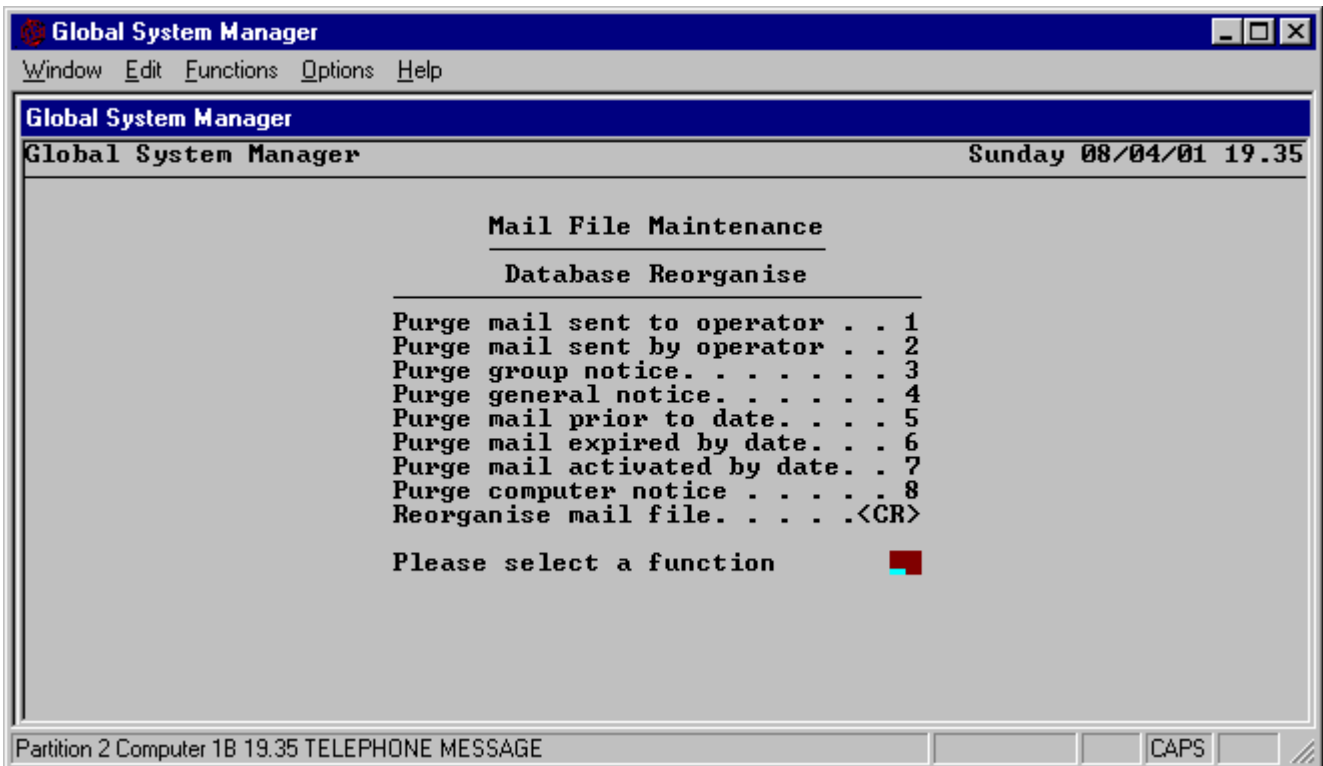


Figure 24.6 - The Reorganise Submenu

24.5.3 Purge Group Notice

This option allows you to purge all the items of mail sent to a particular group. If the reply to the group prompt is <CTRL B> you can delete all group mail. **Use this option with great care!**

In order to proceed you must key Y to the following confirmatory message:

```
Delete mail sent to gggggggggg?:
```

24.5.4 Purge General Notice

This option allows you to purge all the items of mail sent to a general notice board (i.e. all broadcast messages). In order to proceed you must key Y to the following confirmatory message:

```
Delete all general notices?:
```

24.5.5 Purge Mail Prior to Date

This option allows you to purge all the items of mail sent before a particular date. In order to proceed you must key Y to the following confirmatory message:

```
Delete all mail and notices sent before dd.mm.yy?:
```

24.5.6 Purge Mail Expired by Date

This option allows you to purge all the items of mail that expired before a particular date. In order to proceed you must key Y to the following confirmatory message:

```
Delete all mail and notices expired before dd.mm.yy?:
```

24.5.7 Purge Mail Activated by Date

This option allows you to purge all the items of mail activated before a particular date. In order to proceed you must key Y to the following confirmatory message:

```
Delete all mail and notices activated by dd.mm.yy?:
```

24.5.8 Purge Computer/System Notice

This option, which is only available on network or Global System Manager (Unix) configurations, allows you to purge all the items of mail sent to a particular computer (or system). If the reply to the computer-id (system-id) prompt is <CTRL B> you can delete all computer (or system) mail. **Use this option with great care!**

In order to proceed you must key Y to the following confirmatory message:

```
Delete mail sent to id?:
```

24.5.9 Reorganise Mail File

Key <CR> to the Database Reorganise sub-menu to reorganise the mail file. As the reorganisation proceeds, messages will indicate the progress of the reorganisation. For example:

```
Reorganising mail records . . .
Reorganising mailbox records . . .
Reorganising system notice records . . .
```

```
Reorganising group records . . .  
Reorganising mailbox records . . .  
Reorganisation complete Key <CR>:
```

Key <CR> to return to the main \$MAIL menu.

24.6 Operator Customisation

The operator customisation option allows operator specific mailbox customisation for \$MAIL. If you are a supervisor then you are allowed to amend the mailbox customisation for any operator provided that you can supply the lock password. If you are not a supervisor then you can only amend your own mailbox customisation.

If you are a supervisor the following prompt will appear:

```
Key L to list all operators, <CR> for own:
```

Key L to list all the operators who are currently customised. A list of customised operators will appear. The following baseline prompt is displayed:

```
Key operator name, Next, Back, Print, <ESC> to exit:
```

Key N or B to display the next or previous screens respectively. Key P to print the list of customised operators to the \$PR unit.

Key an operator-id, of at least 2 characters, to create a new mailbox customisation record or to amend an existing record. Key <CR> to create or amend your own mailbox customisation record.

24.6.1 Customisation Screen

When an operator-id has been specified, the customisation screen shown in figure 24.7 appears. The following baseline prompt the following baseline prompt is displayed:

```
Key Abandon, Delete, Print, Update, <ESC> to exit:
```

Key A to abandon the mailbox customisation for the selected operator. Key D to delete the customisation record for the selected operator. Key P to print the customisation record for the selected operator to the \$PR unit.

Key U to update the fields on the customisation screen.

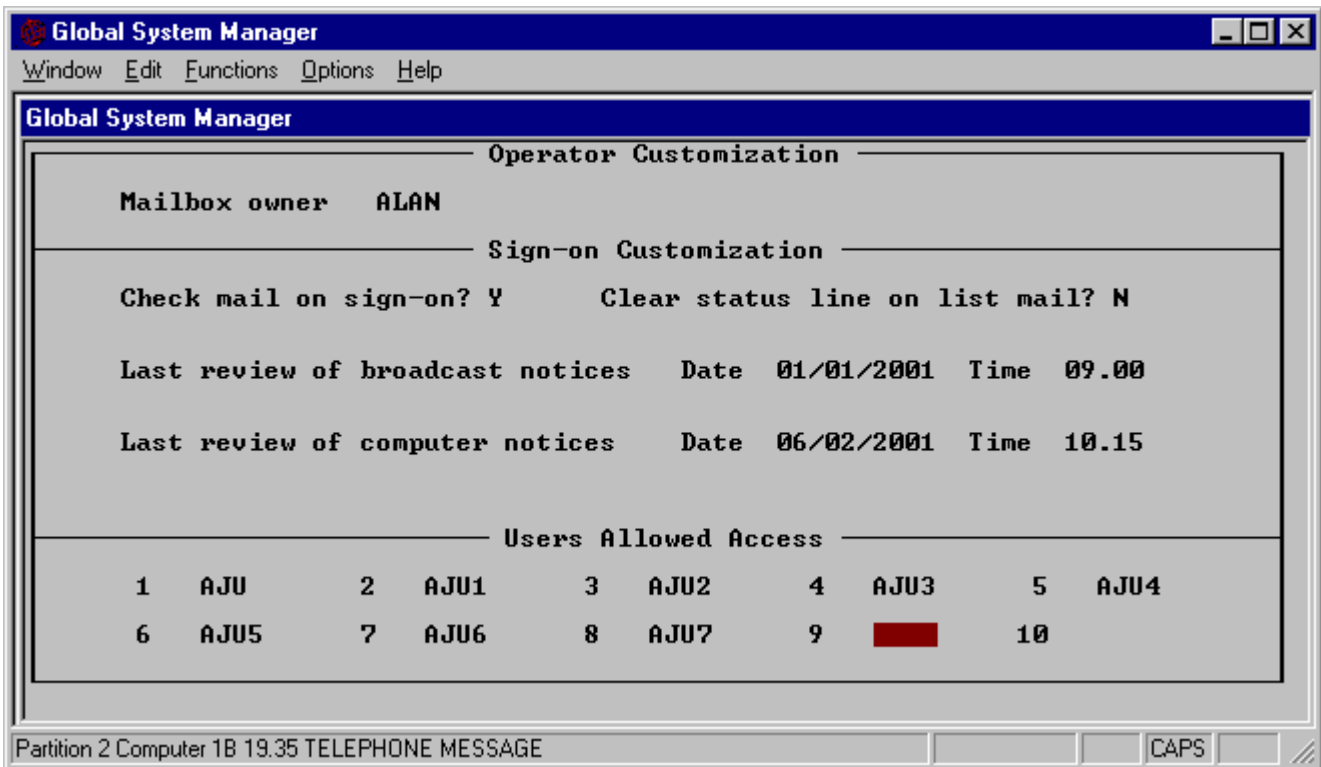


Figure 24.7 - Customisation Screen

24.6.1.1 Check mail on sign-on

This flag is used to indicate whether the \$\$MAIL file is checked by Global System Manager at sign-on. If this option is enabled and mail is available the operator will be informed when he, or she, signs on to Global System Manager. Set this flag to 'N' to suppress the warning messages that may appear at sign-on. This \$MAIL customisation must be applied if, for example, the operator has been customised (using \$AUTH) to sign-on and run a menu entry automatically.

24.6.1.2 Last review of broadcast notices

The last review date and time are available for broadcast notices and are an indication of the last time this notice-board was accessed by the operator. Broadcast notices which have been sent or activated prior to the last review time, will not be displayed at sign-on.

24.6.1.3 Last review of computer (system) notices

The last review date and time are available for computer (or system) notices are an indication of the last time this notice-board was accessed by the operator. Computer (or system) notices which have been sent or activated prior to the last review time, will not be displayed at sign-on.

24.6.1.4 Users Allowed Access

You are allowed to specify up to 10 operators who are allowed access to your non-private mail or telephone messages.

24.7 Lock with Password

The lock password function allows you to specify the password which prevents unauthorised entry into the Reorganise Mail File function (see section 24.5) and also prevents unauthorised users from updating the customisation records of other operators (see section 24.6).

If there is an existing lock password you are asked to supply this password first. You are prompted for the new password. Key <ESC> to leave the password unchanged. Key <CTRL A> to remove the password.

25. \$MH & MN - Menus and Menu Maintenance

The Menu Handler (\$MH) and Menu Maintenance utility (MN) are fully documented in Chapter 7 of the Global System Manager Manual.

26. \$OPID - Operator List Maintenance

The \$OPID command allows you to associate a 35-character Long Operator Name and a 15-character Operator Detail text string with a standard 4 character operator-id.

26.1 \$OPID Dialogue

On entry to \$\$OPID, the current list of operators is displayed in alphabetical order together with the associated Long Operator Names and Operator Detail information (see figure 26.1). The following baseline appears:

Key Amend, Back, Delete, Insert, Start, <CR> for next, <ESC> to exit:

26.1.1 Amend Existing Entry

Key A to amend an existing entry. You will be prompted for operator-id you wish to amend. You are allowed to change both the associated Long Operator Name and the Operator Details for the selected operator-id. You can continue amending entries until you key <ESC> to the operator-id prompt.

26.1.2 Delete Existing Entry

Key D to delete an operator-id from the current list. You are prompted for the name of the operator that you wish to delete. You can continue deleting entries until you key <ESC> to the operator-id prompt.

26.1.3 Insert New Entry

Key I to insert a new operator into the current list. You are prompted for the operator-id, the Long Operator Name and the Operator Details text. You can continue inserting new entries until you key <ESC> to the operator-id prompt.

26.1.4 Start Display from a Given Entry

Key S to start displaying the operator list from a given entry. You are prompted for the operator-id from where you wish to start the next display.

26.1.5 Paging Commands

The B and <CR> options allow you to page backwards and forwards through the list.

26.2 Using the Operator Name Information

The operator name information supplied to \$OPID is held in the Operator Name file \$\$OPID. <SYSREQ> Y can be used to display the contents of the Operator Name file (see section 4.29 of the Global System Manager Manual).

It is anticipated that the Operator Details text will be used to hold useful operator specific information such as a phone extension of that operator.

Note that, at present, there is no interface between the Operator Name information, held in the \$\$OPID file, and the Operator Authorization information, held in the \$\$AUTH file.

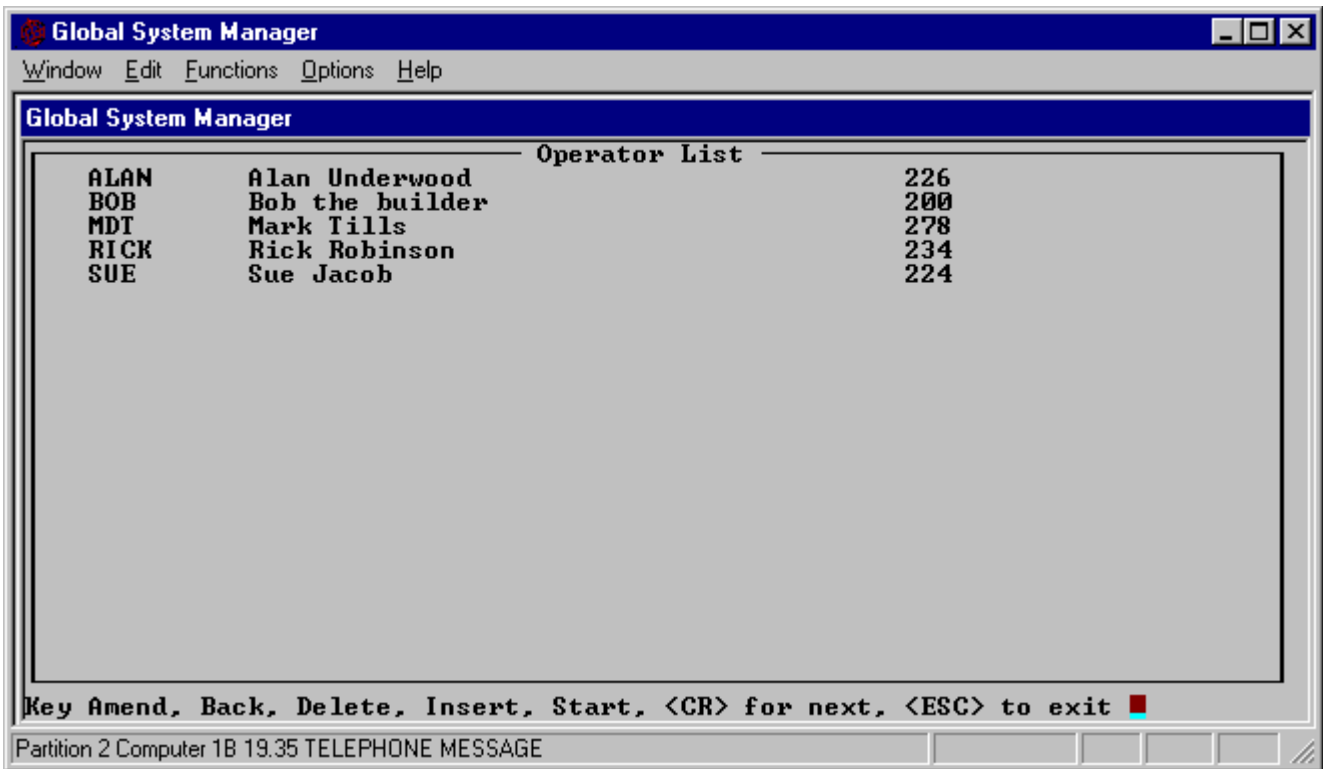


Figure 26.1 - Operator List

27. \$P - Advance the Printer

\$P, the Print Page utility, is fully documented Chapter 5 of the Global System Manager Manual.

28. \$PRINT - Print Files

The \$PRINT command enables you to print out text files (type TF) or report files (type RS) that have been written to a disk unit. You can also use it to display such files on the screen. You are first prompted for the names and units of up to 20 files, and then for the listing unit on which they are to be printed.

\$PRINT first prompts for the file-ids and unit-ids of the files you wish to print and then prompts for the unit-id of the printer. For example, to print the text file S.PROG and the RS file D.\$F04 to the standard printer unit \$PR use the following dialogue:

```
$46 PRINT FILE:S.PROG UNIT:209
$46 PRINT FILE:D.$F04 UNIT:206
$46 PRINT FILE:<CR>
$46 LISTING UNIT:<CR>
```

A reply of <CTRL B> to the \$46 PRINT FILE prompt allows you to print all of the TF and RS files on a unit. For example:

```
$46 PRINT FILE:<CTRL B> UNIT:209
$46 PRINT FILE:<CR>
$46 LISTING UNIT:501
```

A reply of <CR> to the listing unit prompt will cause the file(s) to be sent to the standard printer, \$PR.

A reply of <CTRL A> will display the selected file(s) on your screen.

If the file being printed is a text file (i.e. filetype TF) with a file name prefixed by "B.", "Q.", "F.", "S." or "T." \$PRINT will append line numbers to each line printed.

28.1 Printing to Auxiliary Terminal Ports

Most characters sent to a serially connected terminal are displayed immediately on the screen or control the display in some way. Some terminals, however, allow printers to be connected to an auxiliary serial port on the terminal. Characters can be redirected to the auxiliary port by first sending a special escape sequence. To continue displaying on the screen another sequence must be sent.

\$PRINT allows you define 3 escape sequences (each of up to 20 characters in length) needed to control such local printing. The PRINTER SETUP SEQUENCE is sent to the terminal (not the printer) to perform any initialisation the terminal may require before the auxiliary port is used. For example, it might be used to set up the baud rate of the auxiliary port. Each print line is then sent to the terminal prefixed by the PRINTER ON SEQUENCE and suffixed by the PRINTER OFF SEQUENCE.

```

Global System Manager
Window Edit Functions Options Help
Global System Manager
Global System Manager Sunday 08/04/01 19.40

Global Business Software
MAIN MENU
Backup GSM database . . . . . 1
Install Global Software . . . . . 2
GSM Utilities . . . . . 3
Speedbase Utilities . . . . . 4
Global 3000 V4.5. . . . . 5
Global 3000 V5.0. . . . . 6
testup GSM database . . . . . 7
Global 3000 V5.1. . . . . 8
Exit. . . . . <CR>

Please select a function      :$PRINT
UNIT:207
$46 PRINT FILE:D.$CUS
$46 PRINT FILE:
$46 LISTING UNIT:500
$46 PRINTING D.$CUS FROM UNIT 207 COMPLETED
$46 PRINTING COMPLETED, _

Partition 2 Computer 1B
CAPS

```

Figure 28.1 - Running \$PRINT

```

Global System Manager
Window Edit Functions Options Help
Global System Manager
Global Business Software
MAIN MENU
Backup GSM database . . . . . 1
Install Global Software . . . . . 2
GSM Utilities . . . . . 3
Speedbase Utilities . . . . . 4
Global 3000 V4.5. . . . . 5
Global 3000 V5.0. . . . . 6
testup GSM database . . . . . 7
Global 3000 V5.1. . . . . 8
Exit. . . . . <CR>

Please select a function      :$PRINT
UNIT:205
$46 PRINT FILE:S.XX
$46 PRINT FILE:
$46 LISTING UNIT:AUXC
$46 PRINTER SETUP SEQUENCE IN HEX:1B5467
$46 PRINTER ON SEQUENCE IN HEX:1B5634
$46 PRINTER OFF SEQUENCE IN HEX:1B5634 CUSTOMIZED
$46 LISTING UNIT:AUXP

Partition 2 Computer 1B
CAPS

```

Figure 28.2 - Printing to an Auxiliary Printer Port

To define the escape sequences your terminal uses key AUXC to the listing unit prompt. You will then be prompted for the three sequences in turn. These sequences are saved on disk within a customized version of \$PRINT.

To print to the auxiliary port printer key AUXP in response to the listing unit prompt. The sequences you defined by using the AUXC command will be used when transmitting to the port and will remain in force until you use the AUXC option again.

28.2 Problems and Restrictions with AUXC/AUXP Printing

Note that because only a single set of escape sequences are saved within \$PRINT the AUXC/AUXP facility can only sensibly be used with a single terminal type.

Some screens do not implement the auxiliary port feature in a sensible way. The following problems have been experienced:

1. Some screens continue to interpret escape sequences meant only for the printer and therefore corrupt the screen contents;
2. The PRINTER OFF sequence used to switch the terminal back to ordinary mode is sent in error to the printer as well;
3. When the printer signals that it is busy by sending an X-OFF character to the terminal this is not reflected back to the computer, so Global System Manager continues to send characters to the terminal/printer which are consequently lost.

29. \$RCP - Record/Playback Utility

\$RCP, the Record/Playback utility, is fully documented Chapter 5 of the Global System Manager Manual.

30. \$REMOTE - Copy Files Between Remote Computers

The functionality of the \$REMOTE utility dependent on the host operating system. \$REMOTE is fully documented in Chapter 4 of the Global Operating Manual.

31. \$REORG - Reorganise Domain or Volume

The functionality of the \$REORG utility dependent on the host operating system. \$REORG is fully documented in Chapter 4 of the Global Operating Manual.

32. \$RESTORE - Restore From Backup

(Note that restore operations using dialogue similar to that of the \$RESTORE command program can be performed under application program control by using the REST\$ routine described in the Global Development File Management Manual.)

To restore a volume from a backup cycle previously created by \$SAVE, you must firstly prepare a new copy of the volume, if necessary, using \$V. If you are going to re-use the existing volume, you should remember that the restore process will destroy any files that are currently on it. The dialogue which follows shows how you would use this to recover SAMPLE data from the backup cycle created in the example described in section 34.3.1.

```
GSM READY:$RESTORE
.....
..... (explanatory text)
.....
Key name of first backup volume:SAMPLE-A1
Key F for full restore, S for selective restore, L to list backup (F):<CR>
```

Note that since the files used by complex applications are often closely linked you must be extremely careful when restoring individual files. Such selective restoration of files should normally only be carried out by Service Centres and application programmers. You may find the \$F list function useful to check you have the right diskettes (see Chapter 15).

```
Key first/only unit to restore to:210
MOUNT BACKUP SAMPLE-A1 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
Restore SAMPLE DATA COPY of 22/11/90 09.45.07 (N)?:Y
```

(This prompt allows you to cancel the restore operation, by keying N, if the title or date on which the backup copy was taken are not as expected. Following your reply of Y data is transferred from the first backup volume.)

```
... restoring file1 to SAMPLE on 210
... restoring file2 to SAMPLE on 210
... etc.
MOUNT BACKUP SAMPLE-A2 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
... restoring file3 to SAMPLE on 210
... restoring file4 to SAMPLE on 210
... etc.
MOUNT BACKUP SAMPLE-A3 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
... restoring file5 to SAMPLE on 210
... restoring file6 to SAMPLE on 210
... etc.
RESTORE COMPLETE - REMOVE BACKUP DISKETTE FROM 1ST DISKETTE DRIVE - 140
GSM READY:
```

The hard disk volume SAMPLE is now restored to its original state on unit 210. You should of course preserve the backup diskettes in case you need to repeat the operation.

Should the restore operation fail to complete successfully, one of the following error messages will be displayed:

```
$30 RESTORE CANNOT PROCEED - BACKUP CONTAINS WRONG DATA
$30 RESTORE ABORTED - file-id MASTER REQUIRED ON BACKUP UNIT
$30 RESTORE ABORTED - file-id MASTER CORRUPT OR MISSING
$30 RESTORE ABORTED - BACKUP VOLUME nn CORRUPT OR MISSING
```

The first message is displayed when the operator fails to reply Y to the restore confirmation prompt because the title or creation date of the backup cycle were not as expected.

The second message is displayed if \$RESTORE detects that it is about to request that a master volume be mounted on the backup unit itself. This probably indicates that the unit-id for the principal master volume, supplied in response to the restore prompt, was specified incorrectly.

The third message is displayed if the operator replies N to any mount prompt, or if there is an irrecoverable I/O error when accessing a master file.

The fourth message is displayed if the operator replies N to the backup prompt, or if internal checks reveal that the backup cycle is inconsistent. For example, it might be only partially complete, having been produced by a previous save operation which failed.

Do not use \$RESTORE with Global 2000 Business software or Global Finder - it could cause serious problems with the identification of different backup cycles.

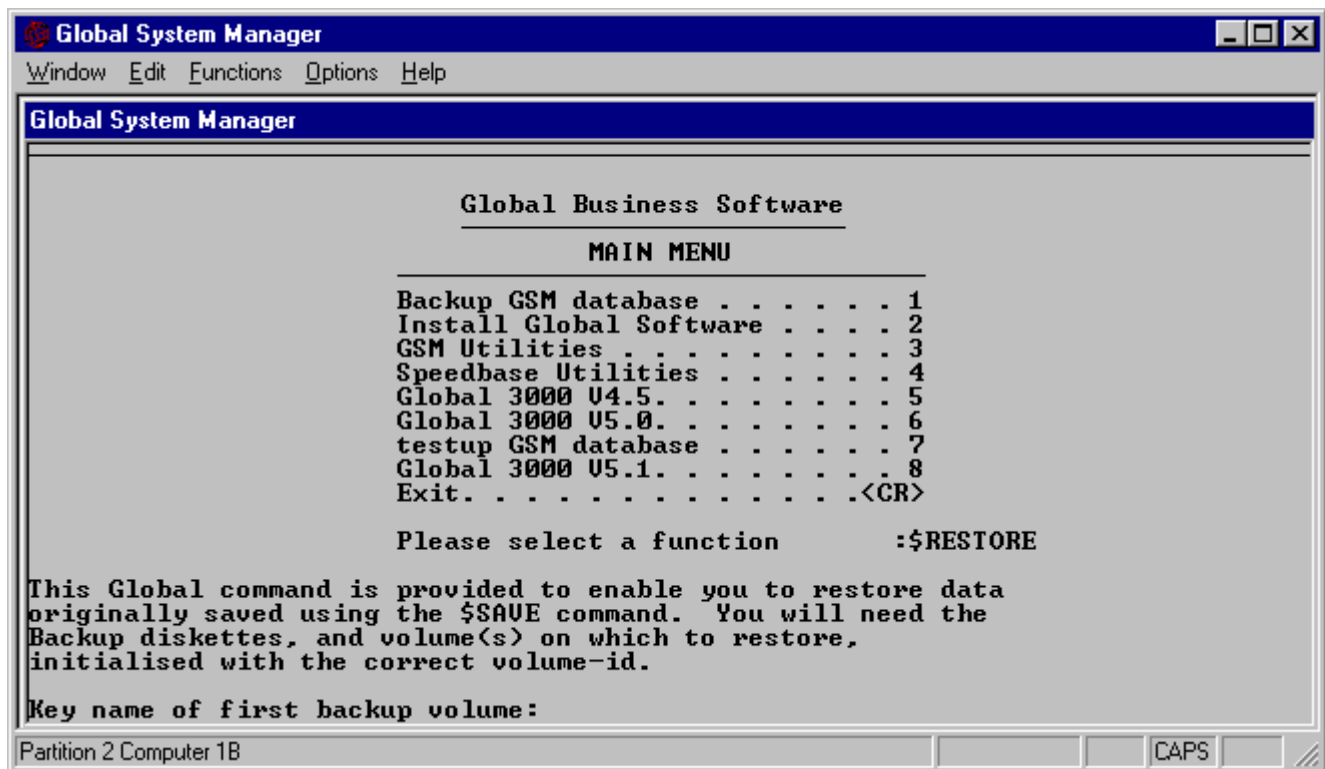


Figure 32.1 - After keying \$RESTORE



Figure 32.2 - Sample dialogue

33. \$S - Obtain System Information

\$S, the Obtain System Information utility, is fully documented Chapter 5 of the Global System Manager Manual.

34. \$SAVE - Create Backup

(Note that save operations using dialogue similar to that of the \$SAVE command program can be performed under application program control by using the SAVE\$ routine described in the Global Development File Management Manual.)

34.1 Introduction

The Data Security System enables you to save, restore or move up to 100 different master files or complete master volumes at a time, providing that no more than 127 files are involved in total. Data from a single master file can occupy up to 99 backup volumes, so it is quite possible to use the system to secure large fixed-disk files on a number of much smaller capacity diskettes. Alternatively, master files from a number of exchangeable volumes may be collected together and preserved on a single backup cycle.

Before you use the Data Security System you will normally require to create a security catalogue defining the files and volumes involved, identify the backup device by assigning \$B to its unit address, and initialize a pool of backup volumes.

34.1.1 The Security Catalogue

The master volumes and files to participate in save, restore or move operations are normally defined in a special catalogue file, known as a security catalogue, which you normally establish on the first master volume to be saved, the **principal master volume**. (The diskettes of the backup cycle will take their names from the name of the volume on which the security catalogue resides.) Each entry of the catalogue specifies either a particular file or an entire volume to be preserved, and indicates the unit which the file or volume will occupy when \$SAVE is run. By re-assigning the unit-ids employed in the catalogue, or by using floating unit addresses, you can restore the files or volumes to different units, and thus use a save followed by a restore to move bulk data via the medium of the backup cycle. The special conventions you must follow when building a security catalogue are described in section 34.2.

The \$SAVE command allows you to backup a single named file, or all the files of a particular volume, without the need to construct a security catalogue, since the full generality of the catalogue is not required when the scope of the save operation can be specified very simply. In this case the **principal master volume** is considered to be the single master volume involved.

34.1.2 The Backup Unit-id, \$B

During a save or restore operation the volumes of a backup cycle are mounted one by one on the unit assigned to \$B, so it is important that you make the correct assignment before attempting to use the system. \$B is established during the installation of Global System Manager, but can be changed by using \$A, <SYSREQ> I, or the "Permanent unit assignments" option of \$CUS.

34.1.3 Backup Cycle Names

A backup cycle is always named after the principal master volume, but the name is further qualified by means of an additional character known as the **cycle-id**. Thus, for example, backup cycle C of the principal master volume SAMAST is named as SAMAST-C.

The number of different cycles to be used is determined by application and operational requirements. For example, it is fairly typical, though by no means universal, to save the

contents of master files at the end of each working day. In this case it might prove operationally convenient to maintain five different backup cycles, one for each weekday. They might, for example, have cycle-ids A, B, C, D and E and be known as SAMAST-A, SAMAST-B, SAMAST-C, SAMAST-D and SAMAST-E.

34.1.4 Backup Volumes

Each backup volume within a cycle is identified by means of a **volume number** between 01 and 99. During save and restore operations the volumes are accessed one by one in ascending volume number order.

You should use the \$V command (see Chapter 4 of the Global Operating Manual) to initialise or format a number of diskettes on unit \$B, each of which must be given the name BACKUP. It is better to be generous, and create more backup volumes than you need, than to run out of them during a save operation.

You will initially write only the name BACKUP on the adhesive label, but eventually, once it is used in a save operation, you will complete the label so it is of the form:

BACKUP/vvvvvv-cnn

where vvvvvv is the volume-id of the principal master volume; c is the cycle-id and nn is the volume number. For example, supposing two backup volumes are required to contain the data from cycle A of principal master volume SAMAST. Then they would be labelled:

BACKUP/SAMAST-A

and:

BACKUP/SAMAST-A2

The identifier which appears under the word BACKUP in these labels is known as the backup-id. Global System Manager displays it in the prompts used to request backup volumes during save and restore operations.

34.1.5 Backup Files

Each backup volume produced during a save operation contains a single backup file which can itself hold the data of several master files or, alternatively, contain just a fragment of a single large master.

The information from which the backup-id is composed is part of the directory information kept for the backup file. Thus by reading the file label, Global System Manager is able to differentiate between backup volumes, even though they all possess the same volume-id. The file-id of the backup file consists of the first part of the backup-id, excluding the volume number. The volume number itself is encoded in the field used to indicate the file type. Thus, if you use the \$F's LIS instruction to examine BACKUP SAMAST-A2 you will find:

- The volume-id is BACKUP;
- One file is present, named SAMAST-A;
- The type field appears as #02, indicating that this is volume 2 of the cycle.

34.1.6 Master Volumes

When using \$SAVE or \$RESTORE you should always ensure that no other users are working with files on the master volumes involved. If they are, the save or restore operation will be halted and the following message will be displayed:

```
* IN USE ERROR ON unit file - RETRY?:
```

You will only be able to continue when no other user is accessing a master volume.

In most Global System Manager configurations the swap file (\$\$SWAP*nn*) and the user file (\$\$USER), which are normally resident on SYSRES, are normally in use by Global System Manager itself (see Appendix E). You can use the \$STATUS GET command to close these files if you are using single or multi-user systems, but \$\$USER is not affected by the GET command when used on network systems. If you wish to save SYSRES on single and multi-user systems you can run \$STATUS to acquire exclusive control before employing \$SAVE. To save SYSRES on a network system you will have to load Global System Manager from the BACRES volume (see Chapter 2 of the Global Operating Manual).

34.2 Creating a Security Catalogue

Except when you intend to use \$SAVE to simply backup a single volume or file, you must create a security catalogue defining the scope of the save operation, and establish it on the principal master volume, whose volume-id is to be employed in naming the backup cycle. A security catalogue is built in the normal way, using the \$CATAL command. It must contain an entry for every individual master file or complete master volume that you wish to save. It is important that every file whose name appears in the catalogue is in existence when the save operation takes place, since the operation will not complete satisfactorily unless each such master is present. Normally the catalogue contains an entry for itself, so that it is preserved and restored along with the other master files that it defines.

Although a security catalogue has the same format as an ordinary file catalogue, there are special additional considerations which apply when setting it up, which are summarized below.

34.2.1 Entry Sequence

In an ordinary file catalogue the sequence in which the entries appear is unimportant. However, this is far from the case with a security catalogue, since master files are saved, one by one, in the order that their entries appear within the catalogue. Thus to avoid unnecessary volume changes you should group all files belonging to the same master volume together, with those files that reside on the principal master volume itself at the very front of the catalogue.

34.2.2 Unit-id and Volume-id Information

The unit-id you specify in a security catalogue entry indicates the unit that the corresponding master file will occupy during save and restore operations. This is not necessarily the same as the unit used for the file during normal working. You must be careful that you do not inadvertently assign a master file to the backup unit itself, since this obvious error will cause the save process to fail.

If you wish you may omit the volume-id information altogether. If no volume-id is specified then save and restore operations will assume that the file resides, along with the catalogue, on the principal master volume itself.

34.2.3 Saving a Complete Master Volume

You may use a special file name beginning with ** in a security catalogue entry to cause every file on its volume to be saved, without having to name the individual file explicitly.

34.2.4 Example 1

The table below shows the contents of a very simple security catalogue which requests that the files SASECURE, PRODUCT and STATEMENT of volume SAMAST, together with the entire contents of SAMEXT be saved. The master volumes are clearly exchangeable, since during the save operation they both occupy unit 100. (The unit used for the backup volumes is not in the catalogue: it is, of course, determined by the assignment of \$B.)

File-ID	Unit-ID	Volume-ID
SASECURE	100	SAMAST
PRODUCT	100	SAMAST
STATEMEN T	100	SAMAST
**SAMEXT	100	SAMEXT

Example Catalogue for Diskette-based Masters

Considering the catalogue entries one by one, SASECURE is the name of the catalogue itself, PRODUCT and STATEMNT are two application master files, and **SAMEXT is a special file name beginning with two asterisks, used to request the Data Security System that all files of the volume are to be saved. It is a useful convention to name the special file-id itself after the volume it is responsible for, as in this example.

In practice this catalogue is rather inflexible, since absolute unit-ids have been specified, forcing the restore operation to always return the data to unit 100. If floating unit addresses had been used (e.g. 0 instead of 100) it would be possible to restore to any unit, since floating addresses are relative to the unit address of the principal master volume, which can be respecified when \$RESTORE is run. An alternative to floating addresses, which also preserves flexibility, is to use logical unit-ids instead. For example if DSK had been used in place of 100, and it was necessary to transfer the data to volumes on 140, then it would only be necessary to reassign DSK to 140 before running \$RESTORE, for the move to take place. Of course, the catalogue is quite adequate if you only ever intend to use it to secure and recover data on unit 100.

34.2.5 Example 2

In a fixed disk system SAMAST and SAMEXT might be adjacent subvolumes of a domain (i.e. subvolumes occupying consecutive subunit addresses). The following security catalogue, using floating unit addresses, could be used to save the same data as in the previous example, and allow the flexibility of moving it to another pair of consecutive unit addresses if necessary.

File-ID	Unit-ID	Volume-ID
SASECURE	0	SAMAST
PRODUCT	0	SAMAST
STATEMEN T	0	SAMAST
**SAMEXT	1	SAMEXT

Example Catalogue for Fixed Masters

34.3 Saving a Backup Cycle using \$SAVE

You use the \$SAVE command to save a specified backup cycle. When you run the command, and after some explanatory text has been displayed, you are asked whether you want to use a catalogue or save a single file or unit. If you choose the latter you are asked to specify the name of the file and/or the address of the unit. If a catalogue is to be used instead you are prompted for its name and address. Mount prompts are used to request master files which are not found to be online as the save operation progresses.

Do not use \$SAVE with Global 2000 Business software or Global Finder - it could cause serious problems with the identification of different backup cycles.

Before you run \$SAVE you should prepare a sufficient number of backup diskettes, each initialized (or formatted) with the volume-id BACKUP; it is not enough to simply rename an old diskette as BACKUP using the \$F CHA instruction. You should also choose a backup cycle-id - a letter, A to Z - and a title, up to 30 characters in length, by which the backup is to be known.

The backup volumes are called for, one by one, as they are needed. You may either supply a new volume, or reuse an existing one. In the latter case Global System Manager will check that the current backup file, which will be overwritten by the save operation, corresponds to the expected backup-id.

You may specify that each backup file be immediately re-read once it has been produced, to verify that the information has been saved satisfactorily. If an I/O error occurs on a backup volume, Global System Manager allows you to supply a new volume as a replacement and continue.

Each backup file is flagged as protected during the save operation to prevent it being inadvertently destroyed through operational error.

34.3.1 Example 3

The following dialogue shows how you run \$SAVE to back up a volume called SAMPLE occupying unit 210 onto three diskettes of cycle A, entitled SAMPLE DATA COPY:

```
GSM READY:$SAVE
.....
..... (explanatory text)
.....
Key C to use a Catalogue, or S to save single file or unit (S):S
Key name of file to save, or <CR> for whole unit:<CR> unit:210
Backup cycle code (A):A (or <CR>)
Backup volumes should be labelled SAMPLE-A1, SAMPLE-A2, etc.
Verify diskettes after backup (Y)?:Y (or <CR>)
MOUNT BACKUP SAMPLE-A1 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
Backup title:SAMPLE DATA COPY
... copying file1 from SAMPLE on 210 .....
... copying file2 from SAMPLE on 210 .....
... etc.
... verifying Backup disk 1
MOUNT BACKUP SAMPLE-A2 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
... copying file3 from SAMPLE on 210 .....
... copying file4 from SAMPLE on 210 .....
... etc.
... verifying Backup disk 2
MOUNT BACKUP SAMPLE-A3 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
... copying file5 from SAMPLE on 210 .....
... copying file6 from SAMPLE on 210 .....
```

```

... etc.
... verifying Backup disk 3
SAVE COMPLETE - REMOVE BACKUP VOLUME FROM 1ST DISKETTE DRIVE - 140
Print report (Y)?:Y (or <CR>)
(Report detailing save operation is printed - keep it for reference)

```

This example dialogue results in three BACKUP diskette volumes called BACKUP SAMPLE-A1, BACKUP SAMPLE-A2 and BACKUP SAMPLE-A3 being produced. \$SAVE calls for BACKUP volumes until all the data involved has been transferred, so you should make sure you prepare enough of them to have some spare in case you miscalculated the number needed.

Should the save operation fail to complete successfully, one of the following error messages will be displayed:

```

$29 SAVE CANNOT PROCEED - BACKUP NOT TO BE OVERWRITTEN

$29 SAVE ABORTED - file-id MASTER REQUIRED ON BACKUP UNIT

$29 SAVE ABORTED - file-id MASTER CORRUPT OR MISSING

$29 SAVE ABORTED - BACKUP VOLUME number CORRUPT OR MISSING

```

The first message is displayed when the operator fails to reply Y to the overwrite confirmation prompt because the title and creation date of the backup cycle were not as expected.

The second message is displayed if \$SAVE detects that it is about to request that a master volume be mounted on the backup unit itself. This indicates either an error in the security catalogue, or that the unit-id for the principal master volume, supplied in response to the save prompt, was specified incorrectly.

The third message is displayed if the operator replies N to any mount prompt, or there is an irrecoverable I/O error on the master file, or if a master file defined in the security catalogue is not present on its master volume. The file-id that appears as part of the message indicates the file involved. If the file-id identifies the security catalogue file itself the message means that either there was a permanent I/O error accessing the security file, or the security file was not present on the principal master volume or, if a security file was present it was not a valid catalogue.

The fourth message is displayed if the operator replies N to the backup prompt or to the continuation prompt which follows a permanent I/O error on a backup volume.

In the example above the diskette drive at unit 140 was selected to be the backup device. The actual unit address employed depends on the assignment of the \$B unit-id. If this assignment was not made when Global System Manager was installed the following prompt will appear when you run \$SAVE:

```
PLEASE ASSIGN $B:
```

You must reply with the address of the diskette unit you require to use.

34.3.2 Example 4

In the following example a security catalogue has been set up in order to specify various files and volumes on various units which the operator requires to be saved. Note that in this example the computer used is on a network, and so files and volumes on remote devices (e.g. g01, H04)

can be saved. Note also that the backup volumes take the name of the volume on which the catalogue resides, rather than the name of the volume which is to be saved:

```
GSM READY:$SAVE
.....
..... (explanatory text)
.....
Key C to use a Catalogue, or S to save a single file or unit (S):C
Key Security Catalogue name (SECURITY):SAVCATAL unit:210
Backup cycle code (A):<CR>
Backup volumes should be labelled SAMPLE-A1, SAMPLE-A2, etc.
Verify diskettes after backup (Y)?:Y
MOUNT BACKUP SAMPLE-A1 ON 1ST DISKETTE DRIVE - 140 - AND KEY<CR>:<CR>
Backup title:SAMPLE BACKUP USING A CATALOGUE
... copying file1 from SAMPLE on 202 .....
... copying file2 from SADATA on 203 .....
... etc.
... verifying Backup disk 1
MOUNT BACKUP SAMPLE-A2 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
... copying file3 from BWSYS on g01 .....
... copying file4 from BWPROG on H04 .....
... etc.
... verifying Backup disk 2
MOUNT BACKUP SAMPLE-A3 ON 1ST DISKETTE DRIVE - 140 AND KEY <CR>:<CR>
... copying file5 from SYSRES on A00 .....
... copying file6 from BWAARC on a94 .....
... etc.
... verifying Backup disk 3
SAVE COMPLETE - REMOVE BACKUP VOLUME FROM 1ST DISKETTE DRIVE - 140
Print report (Y)?:Y (or <CR>)
(Report detailing save operation is printed for future reference)
```

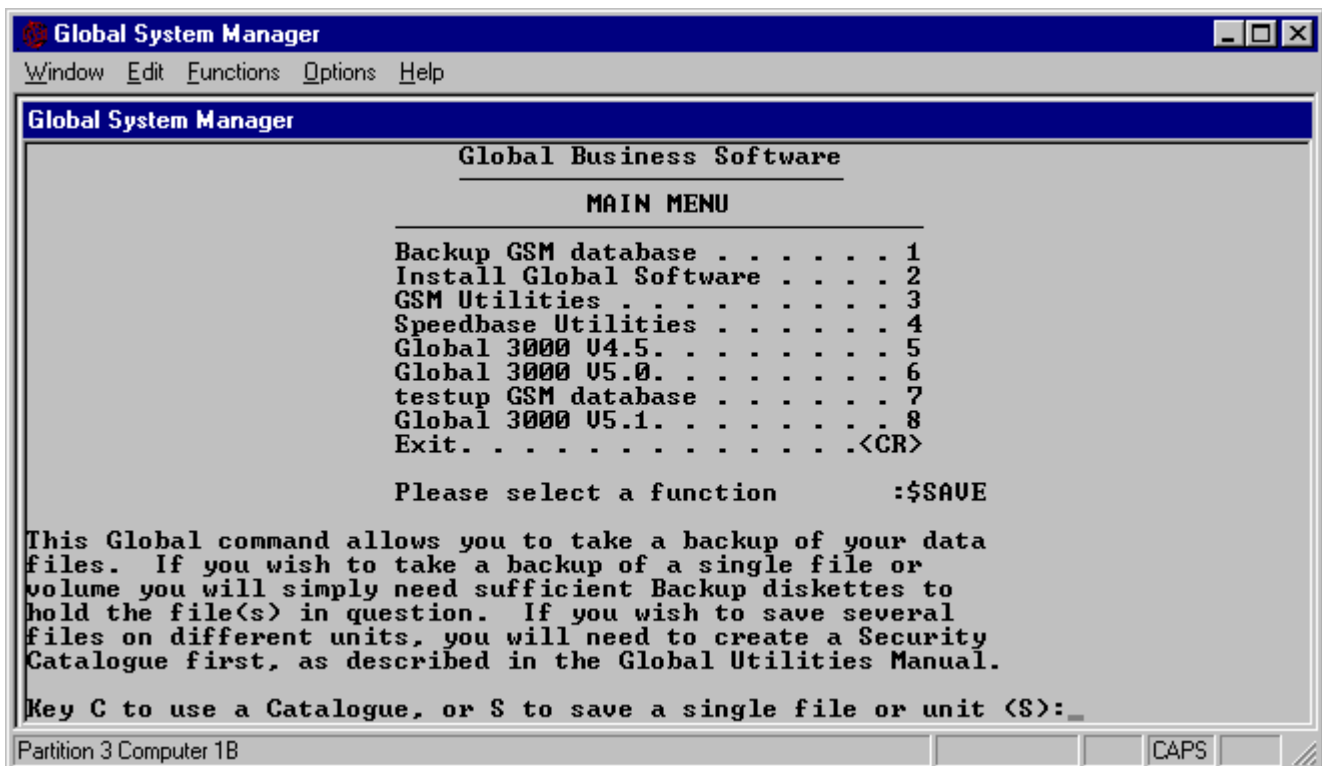


Figure 34.1 - After running \$SAVE

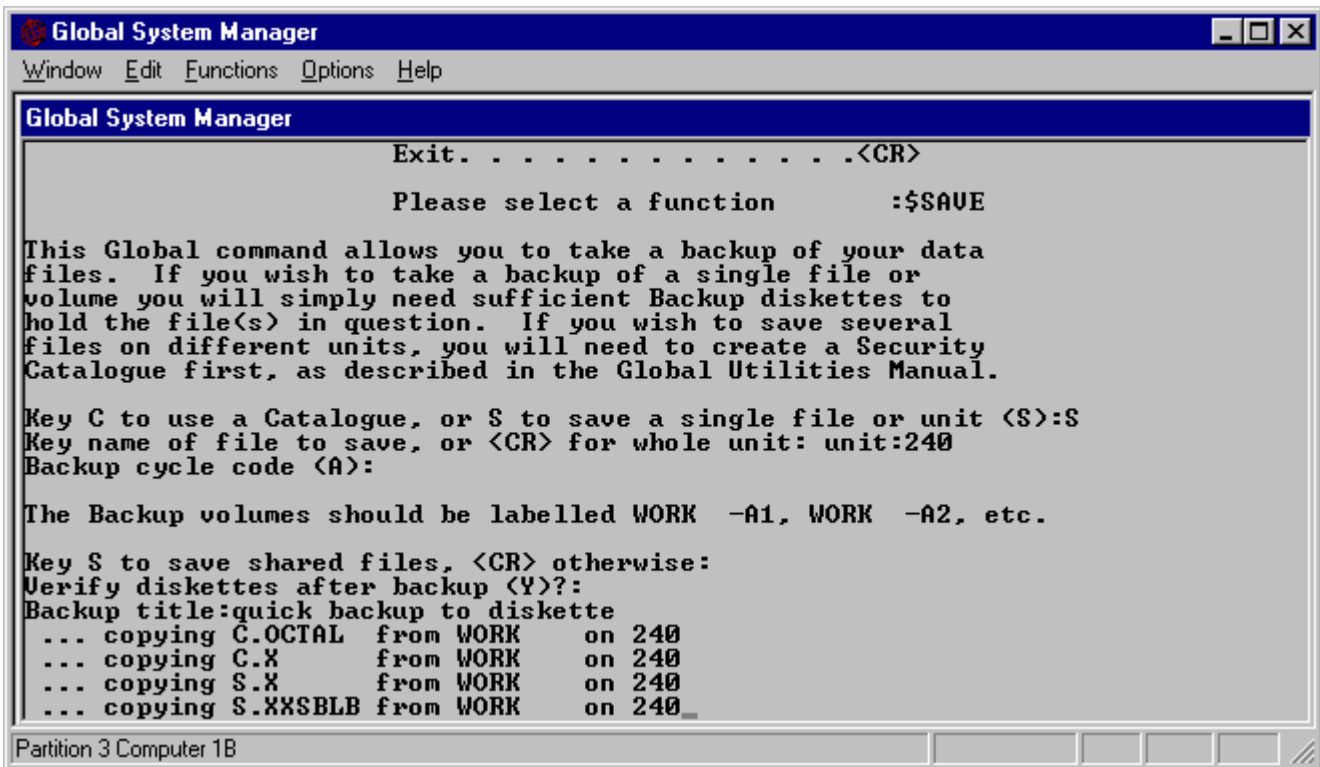


Figure 34.2 - Typical Dialogue

35. \$SORT - Multi-Key Sorting

The \$SORT command allows you to sort one or more relative sequential files using up to 9 keys. The keys may be character or computational fields to be arranged in either ascending or descending sequence. You can choose to perform either a **full sort**, in which case all the data from the input file(s) is processed by the sort phase, or a **tag sort**, where only the key and a pointer to the original record are actually sorted. A work file must always be specified, although it will not be created if there is sufficient memory available. In the case of a full sort the input file, which is normally read-only as far as \$SORT is concerned, may optionally be overwritten.

Because the file pointer used by the tag sort is made the least significant part of the internal sort key, it, unlike the full sort, preserves the original order of records which have duplicate key values. However, the main advantage of the tag sort is that its capacity is normally much greater than that of the full sort, since only the key together with the 4-byte file pointer are actually sorted. You may thus be able to tag sort files which exceed the capacity of the full sort. Where capacity is not a problem the tag sort may be rather slower than the full sort, because the records of the input file are each randomly accessed via the sorted file pointer at the time the output file is created.

It is difficult to give firm guidelines as to the relative performance of the two methods because although there is extra random access needed to create the tag sort's output file, this may be compensated for by less file activity during the sorting phase. If you frequently perform a long full sort then you might try replacing it by a tag sort to see if it is faster.

35.1 The Option Prompt

When you run \$SORT it first displays its option prompt, so that you can specify whether a full sort or tag sort is required. For the full sort, simply reply <CR>:

```
GSM READY:$SORT
$61 OPTION:<CR>
$61 SORT INPUT:
```

For the tag sort, key the word 'TAG', and then <CR> to the subsequent option prompt:

```
GSM READY:$SORT
$61 OPTION:TAG
$61 OPTION:<CR>
$61 SORT INPUT:
```

35.2 The Input File Prompt

Once you have satisfied the option prompt \$SORT asks you to supply the file-id and the unit-id of the relative sequential file (or files) you wish to sort. For example:

```
$61 SORT INPUT:SADATA UNIT:204
$61 SORT INPUT:<CR>
```

indicates that you wish to process file SADATA on unit 204. Note that the sort input prompt is repeated until you key <CR>, so that two or more input files can be sorted and merged into a single output file.

The following error message is displayed if the record length of the input file exceeds 2000 bytes:

```
$61 RECORD SIZE TOO LARGE
```

Internal space constraints prevent \$SORT from processing such a file. You will have to write your own Global Cobol program using SORT, RELEASE and RETURN, if it is necessary to handle very large records. You may need to sort the key together with a pointer to the data, rather than the data itself.

35.3 To Quit

To quit and return control to the menu or GSM READY: prompt, simply key <ESC> in response to the input file prompt.

35.4 The Output File Prompt

The output file prompt:

```
$61 SORT OUTPUT:
```

requests that you specify the file-id and unit-id of the file which is to contain the sorted information. This file will be allocated exactly the same size extent as the input file(s). For example:

```
$61 SORT OUTPUT: SASORTED UNIT: 205
$61 SORT WORK FILE:
```

assigns the file named SASORTED to unit 205.

You may key <CR> in place of either the file-id or unit-id in which case the file-id or unit-id supplied to the input file prompt will be used.

For the full sort, you may choose to make the input and output file the same. In this case the following warning message is displayed:

```
$61 INPUT FILE WILL BE OVERWRITTEN - CONTINUE?:
```

Any reply other than Y will cancel the sort. If you reply Y, the sort will continue but if it does not complete successfully, your input file will be lost.

In the case of the tag sort, the input file is still required at the time the output file is created, so that the full input record can be retrieved via the sorted file pointer. If, by mistake, you specify the two files to be the same, the sort is terminated with the following error message:

```
$61 TAG SORT CANNOT OVERWRITE THE INPUT FILE
```

You should rerun \$SORT and either select the full sort, or use different input and output files.

35.5 The Work File Prompt

Once valid output file information has been obtained the work file prompt is displayed. You must key the file-id and unit-id of the work file. This file will be allocated the maximum amount of contiguous free space available, but it will be deleted once the output file has been created. For example:

```
$61 SORT WORK FILE:WORKFILE UNIT:203
$61 KEY START:
```

assigns a sort work file `WORKFILE` on unit 203.

You may key <CR> instead of the file-id in which case Global System Manager will use the name:

`$SRTWKnn`

where `$SRTWK` are those characters, and `nn` is a unique number. This default is recommended since it is always guaranteed to generate a unique work file name.

You may key <CR> in place of the unit-id, and in this case the work file will be allocated to the same unit as the input file.

If the work file and unit are the same as either the input or output file and unit, then the following message is displayed immediately followed by the work file prompt:

```
$61 SORT WORK FILE CANNOT BE YOUR INPUT OR OUTPUT FILE
```

To continue you must specify a different file as the work file.

35.6 The Key Start Prompt

Once the sort work file has been identified the key start prompt is output repeatedly to allow you to specify the origin, length and type of the keys:

```
$61 KEY START:origin LENGTH:length TYPE:type
$61 KEY START:
```

The first set of responses defines the primary key, the next set (if any) the secondary key, and so on. For a full sort a maximum of 9 keys may be specified and the total key length must not exceed 99 bytes. For the tag sort these limits are 8 and 95 respectively, since the method requires the extra 4-byte file pointer to be treated as the least significant part of the key. Overlapping keys are processed normally, but all the keys you specify must lie wholly within the input file record. The responses must be input as shown in table 35.1.

Response	To be input as ...
<i>origin</i>	an integer between 1 and 2000 inclusive, specifying the byte number of the start of the key. (The first byte of the record counts as byte number 1.)
<i>length</i>	the length of the key in bytes: 1-99 inclusive for ascending character keys; 1-8 inclusive, for other types of key.
<i>type</i>	one of the following codes: XA, X, <CR> for an ascending character key XD for a descending character key CA, C for an ascending computational key CD for a descending computational key

Table 35.1 - Specifying Key Information

Once all the keys have been defined, reply <CR> in response to the key start prompt to cause the option prompt to appear. For example:

```
$61 KEY START:5 LENGTH:12 TYPE:XA
$61 KEY START:22 LENGTH:4 TYPE:CD
$61 KEY START:18 LENGTH:2 TYPE:X
$61 KEY START:<CR>
$61 SORTING
```

The sorting message will appear automatically, without the need to reply <CR>, when you specify the 8th key for a tag sort, or the 9th key for a full sort.

The following error messages may appear when you are specifying key information. The key start prompt will be immediately redisplayed allowing you to correct the error:

INVALID	The <i>length</i> you keyed was not between 1 and 99 inclusive, or the <i>type</i> was not one of the valid codes listed in the table;
NOT WITHIN RECORD	All or part of the key lies outside the record;
KEY TOO LONG	The <i>key length</i> you supplied was greater than 8 bytes, but the <i>key type</i> was other than ascending character.

35.7 The Respecify Keys Prompt

If you mistakenly define keys whose total length, including the tag sort's internal key, sums to more than 99 bytes the following prompt is displayed:

```
$61 TOTAL KEY LENGTH > 99 - RESPECIFY KEYS?:
```

You may key Y or <CR> to cause all previous key definitions to be 'forgotten' and a new key start prompt to be output so that you can begin the process of key specification again. If you reply N control returns to the beginning of the sort, which displays its input file name prompt.

35.8 The Sort

Once you have specified the keys sorting takes place. The following message is displayed:

```
$61 SORTING
```

If all goes well, some time later, the following message is displayed:

```
$61 SORT SUCCESSFUL
```

If there is insufficient internal memory to perform the sort, or if the work file is not large enough, \$SORT will be terminated with a stop code, the meaning of which is defined in Appendix C. For example, the following message indicates that the work file is not large enough:

```
$91 TERMINATED - STOP 11503
```

If the sort is terminated in error a partial output file and work file may exist. They may be deleted either by rerunning \$SORT or by using the \$F DEL instruction.

35.9 Example 1 - Sorting a Transaction File

A file named SATRANS on unit 210 is to be sorted in place into ascending sequence of customer code and transaction amount. Because the output from the sort is to overwrite the input file, a full sort rather than a tag sort must be used. The sort work file, \$SRTWKnn, is to occupy unit 211. The customer code is eight bytes of character information at the start of each record. The transaction amount consists of four computational bytes, immediately following the customer code. The following dialogue takes place:

```
GSM READY:$SORT
$61 OPTION:<CR>
$61 SORT INPUT:SATRANS UNIT:210
$61 SORT INPUT:<CR>
$61 SORT OUTPUT:<CR> UNIT:<CR>
$61 INPUT FILE WILL BE OVERWRITTEN - CONTINUE?:Y
$61 SORT WORK FILE:<CR> UNIT:211
$61 KEY START:1 LENGTH:8 TYPE:<CR>
$61 KEY START:9 LENGTH:4 TYPE:C
$61 KEY START:<CR>
$61 SORTING
$61 SORT SUCCESSFUL
GSM READY:
```

Note that the keys used in this sort are not necessarily unique since the same customer could possibly have initiated two or more transactions for the same amount. Records with the same keys will, of course, be grouped together on the output file, but, because a full sort rather than a tag sort has been used, their order is unpredictable: It is not necessarily the same as the order in which the records appeared on the input file.

35.10 Example 2 - Tag Sorting a Large Master File

The relative sequential file SADATA is to be sorted into ascending order of the unique customer code, so that it can be converted to indexed sequential format using the \$CONV command (see Chapter 8). The customer code is 8 bytes in length, and, since it is to become the key of an indexed sequential file, is located at byte 5 onwards of each record. A tag sort is to be used to create a new file named SASORTED. All the files involved, including the work file, \$SRTWKnn, are on unit 204:

```
GSM READY:$SORT
$61 OPTION:TAG
$61 OPTION:<CR>
$61 SORT INPUT:SADATA UNIT:204
$61 SORT INPUT:<CR>
$61 SORT OUTPUT:SASORTED UNIT:<CR>
$61 SORT WORK FILE:<CR> UNIT:<CR>
$61 KEY START:5 LENGTH:8 TYPE:<CR>
$61 KEY START:<CR>
$61 SORTING
$61 SORT SUCCESSFUL
GSM READY:
```

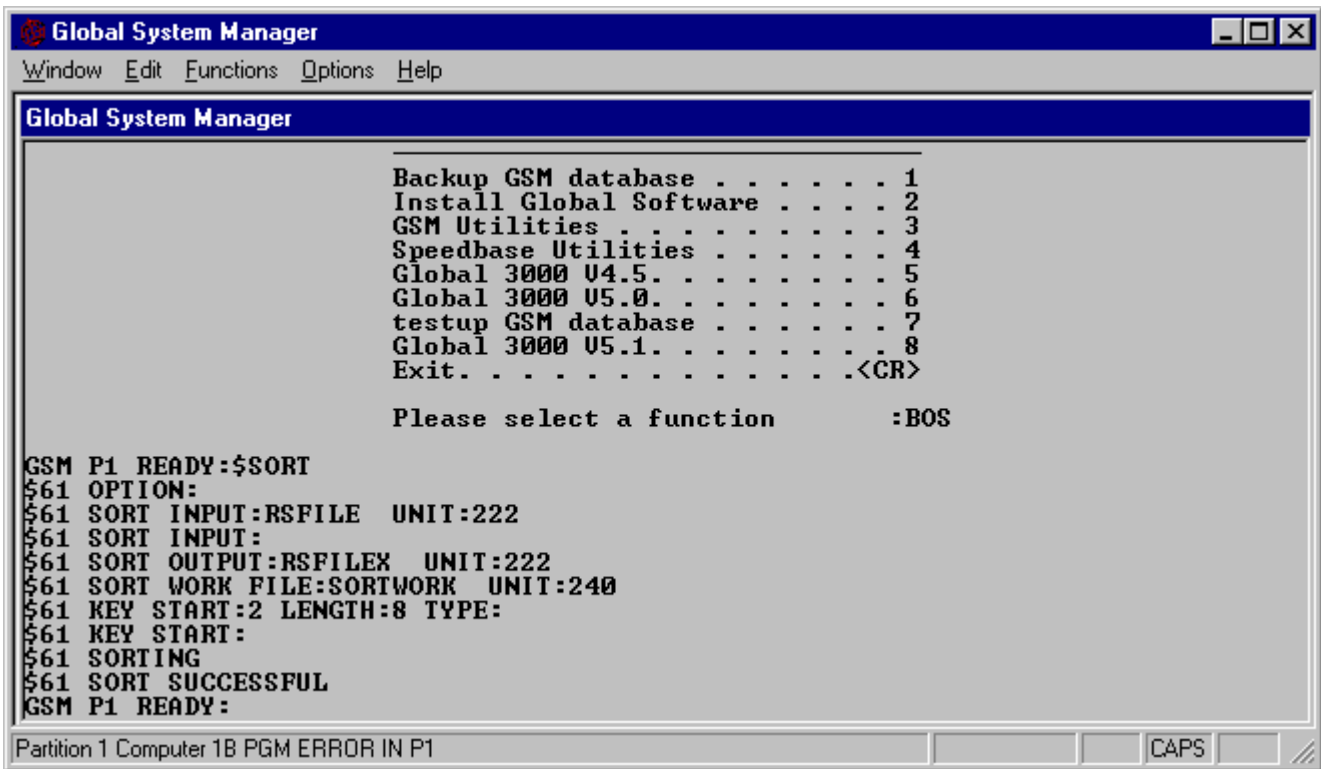


Figure 35.1 - Single key sorting

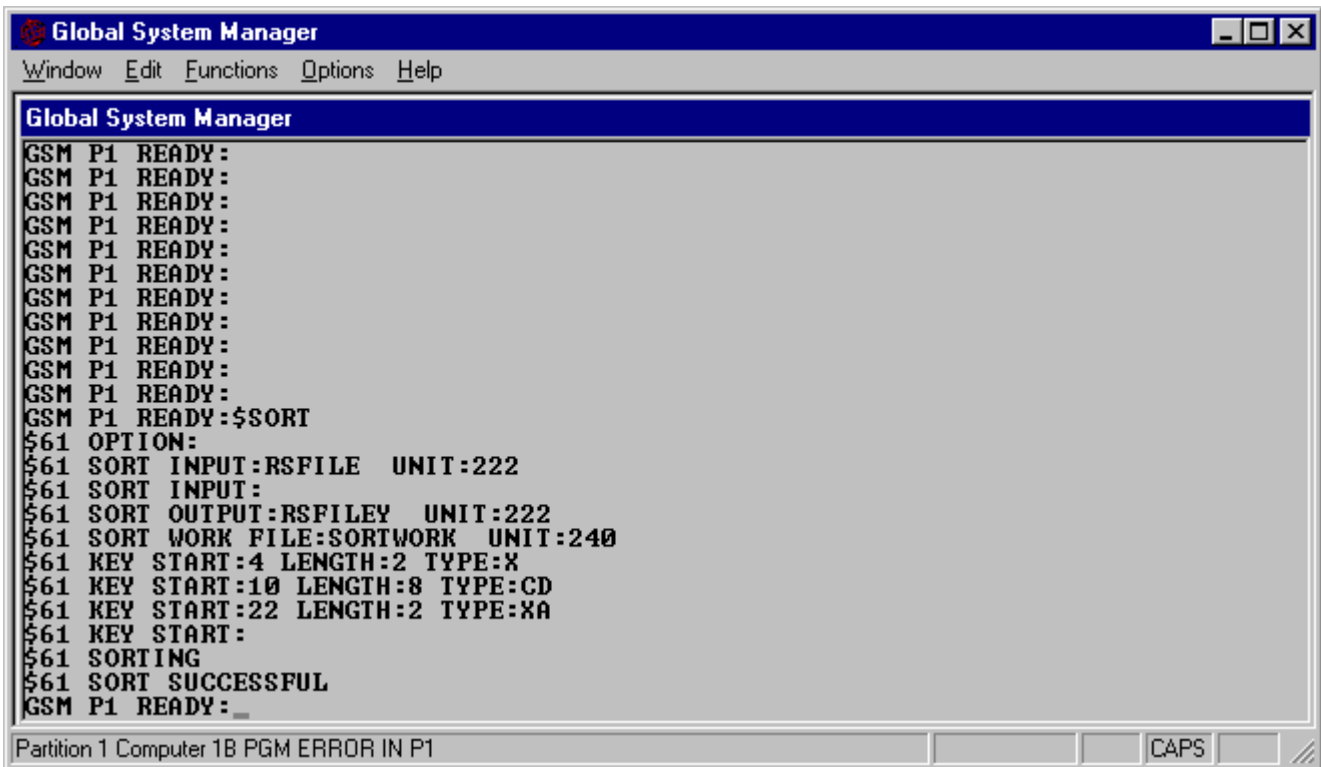


Figure 35.2 - Multi-key sorting

36. \$SP, \$SPOOL & \$SPS - Spooler and Spooler Status

The \$SP, \$SPOOL and \$SPS utilities are fully documented in Chapter 8 of the Global System Manager Manual.

37. \$STATUS - User Control

The \$STATUS command is provided under all multi-user and network versions of Global System Manager to enable you to check which operators are working with the system and which partitions are in use. The status report produced by the command also highlights serious error conditions, such as those affecting the screen or swap file, which cause the affected programs or programs to be cancelled.

\$BYE is an overlay which runs \$STATUS and executes the TER instruction (see section 37.2.19) to terminate Global System Manager on the computer from which it is run. \$BYE or the TER instruction should be used before powering down a computer in a network environment, as explained in section 37.2.19.

\$M is another overlay which runs \$STATUS and executes the MES instruction (see section 37.2.12) to send a message to a specific user or users.

Operators who are supervisors are able to use special task control instructions to cancel or restart the jobs of other users of their computer, obtain or release exclusive control of Global System Manager on their computer, quiesce the system to prevent new operators from signing on towards the end of a working session, or to terminate Global System Manager on their computer. The special task control instructions are only available to supervisors: If 'ordinary' operators (i.e. non supervisors) use \$STATUS, the command simply displays its status report and then returns to the menu or GSM READY: prompt.

37.1 The Status Report

When you run \$STATUS the status report is displayed automatically. If you are a supervisor, this will be followed by a task control prompt as shown in figure 37.1. If the status report contains more than one screen of information, \$STATUS will display the following message:

```
$47 NEXT PAGE:
```

Key N to prevent any further pages of the status report from being displayed. Key Y (or any character except N) to display the next page of the status report.

If \$STATUS is running under Job Management, any next page prompts are suppressed so that the first message is always the task control prompt:

```
$47 TASK CONTROL
```

The USER column of the report lists the user numbers associated with the screens your configuration supports. An asterisk appears beside your own user number.

The OPERATOR column begins with the operator-id of the person currently signed on to the screen. If no-one is signed on the column will remain blank. The letter in brackets is the *authority code* of the operator. Codes S - Z denote *supervisors* able to use task control instructions.

```

Global System Manager
Window Edit Functions Options Help

Global System Manager
USER OPERATOR                STATUS                20.08.38

COMPUTER 1B ** OURS ** -
1* ALAN (S) 712              P1 EXECUTING          P2 SUSPENDE
P3 SUSPENDE                P4 SUSPENDE
P5 SUSPENDE                P6 SUSPENDE
P7 SUSPENDE                P8 SUSPENDE
P9 SUSPENDE

10  AJU  (S) 712             P1 ACTIVE             P2 SUSPENDE
P3 SUSPENDE                P4 SUSPENDE
P5 SUSPENDE                P6 SUSPENDE
P7 SUSPENDE                P8 SUSPENDE
P9 SUSPENDE

19  AJU2 (S) 712            P1 ACTIVE             P2 SUSPENDE
P3 SUSPENDE                P4 SUSPENDE
P5 SUSPENDE                P6 SUSPENDE
P7 SUSPENDE                P8 SUSPENDE
P9 SUSPENDE

28                               P1 NOT SIGNED ON     P2 NOT SIGNED ON
                               P3 NOT SIGNED ON     P4 NOT SIGNED ON

COMPUTER A FILE SERVER
COMPUTER B FILE SERVER
COMPUTER Q FILE SERVER
$47 NEXT PAGE?:_

Partition 1 Computer 1B

```

Figure 37.1 - A typical \$STATUS report

```

Global System Manager
Window Edit Functions Options Help

Global System Manager
COMPUTER 1B ** OURS ** -
1* ALAN (S) 712              P1 EXECUTING          P2 SUSPENDE
P3 SUSPENDE                P4 SUSPENDE
P5 SUSPENDE                P6 SUSPENDE
P7 SUSPENDE                P8 SUSPENDE
P9 SUSPENDE

10  AJU  (S) 712             P1 ACTIVE             P2 SUSPENDE
P3 SUSPENDE                P4 SUSPENDE
P5 SUSPENDE                P6 SUSPENDE
P7 SUSPENDE                P8 SUSPENDE
P9 SUSPENDE

19  AJU2 (S) 712            P1 ACTIVE             P2 SUSPENDE
P3 SUSPENDE                P4 SUSPENDE
P5 SUSPENDE                P6 SUSPENDE
P7 SUSPENDE                P8 SUSPENDE
P9 SUSPENDE

28                               P1 NOT SIGNED ON     P2 NOT SIGNED ON
                               P3 NOT SIGNED ON     P4 NOT SIGNED ON

COMPUTER A FILE SERVER
COMPUTER B FILE SERVER
COMPUTER Q FILE SERVER
$47 NEXT PAGE?:N
$47 TASK CONTROL:QUI ARE YOU SURE? <N>:Y QUIESCE IN PROGRESS
$47 TASK CONTROL:TER_

Partition 1 Computer 1B

```

Figure 37.2 - Quiescing and terminating the system

On network configurations the list of users is divided into sections with each section starting with a heading line showing the computer identification letter (or hexadecimal code if the computer's LAN-id does not lie in the range A-Z) along with a description of the computer

specified using the DES command, and an indication of the status of the computer (e.g. NOT IN USE).

The third and fourth columns indicate the status of the operator's concurrent partitions. The various messages which may appear are summarized in table 37.2. If the operator is using a different computer from yours in a network configuration then no information is available about the status of any programs being run by that operator, and the fourth column will contain the message 'SEPARATE COMPUTER'.

The status report may optionally be followed by one or more of the following warning messages:

```
*** YOU HAVE EXCLUSIVE USE OF PROCESSOR
*** QUIESCE IS IN PROGRESS - NEW OPERATORS CANNOT SIGN ON
*** WARNING - USER FILE HAS BEEN PURGED - REBOOT
*** MASTER COMPUTER HAS BEEN REBOOTED
```

The first message is displayed if you have acquired exclusive control of your computer and have failed to relinquish it. Other users of the same computer are locked out until you do so. This can arise either because you used the \$STATUS GET instruction and have not yet cancelled its effects using REL (see section 37.2.17), or because an application program you ran previously obtained exclusive control using the GETX\$ system routine, but failed to call RELX\$ subsequently to relinquish it. Both GETX\$ and RELX\$ are described in the Global Development System Subroutines Manual.

The second message indicates that a QUI (Quiesce) instruction has been issued (see section 37.2.16). In this case no new operators can log onto the computer until the quiesce command has been cancelled.

The third message indicates the \$\$USER user file on the master computer has been purged using the PUR instruction (see section 37.2.15). You should reload Global System Manager on your computer otherwise the \$STATUS status report may not be accurate.

The fourth message indicates that Global System Manager has been reloaded on the master computer. You should reload Global System Manager on your computer, and all other computers on a network, otherwise the \$STATUS status report may not be accurate.

Instruction	Description
<CR>	Obtain an updated status report
<CTRL C>	List help information ("?" also displays help information)
<ESC>	Quit
CAN	Cancel the jobs of a particular user
COM	List those computers that have loaded Global System Manager since the user file was last purged
DES	Amend the table of computer descriptions
DIS	Disconnect a user and their screen
FIL	List all currently open files
GET	Gain exclusive control of the computer
LIL	List only those systems that are currently in use
LIS	List details of a selected system only

LOC	List all currently active locks
LOG	Write a message to the Event Logging file describing some system event
MES	Send a message to a specific user or users or a group of users
MON	Monitor terminals using modem control lines
PRI	Print the details displayed by the <CR>, FIL and LOC instructions
PUR	Cause the user file to be reallocated when Global System Manager is next reloaded on the master computer, thereby purging it of irrelevant or incorrect information
QUI	Quiesce Global System Manager
REL	Release exclusive control of the computer
RES	Restart a user following an error
TER	Terminate execution of Global System Manager

Table 37.1 - \$STATUS Instructions

\$STATUS Message	Meaning
ACTIVE	A partition is operating normally.
NOT SIGNED ON	This indicates that no operator is signed on at the screen.
NOT IN USE	This indicates that the computer is not currently in use.
INACTIVE FOR <i>nnn</i> MINS	The program has been waiting for input for the length of time shown. It is likely that the screen is no longer being used. *
SUSPENDED	The program has suspended itself for a period of time. *
AWAITING SCREEN	The indicated partition requires the screen but cannot access it because another partition is using it.
CANCELLED	The program has been cancelled using the CAN instruction. It may be restarted using RES.
NOT CONNECTED	The operator's screen or computer is not connected. *
LINE DOWN	Output is halted because the line connecting the screen to the computer is out of service. When dial-up lines are used this indicates that the screen in question has rung off. *
WAITING FOR <CTRL Q>	The operator has accidentally keyed <CTRL S> on the screen, causing output to be halted until <CTRL Q> is keyed. This condition normally only arises on screens which use X-ON/X-OFF protocol. *
SCREEN I/O ERROR	An irrecoverable I/O error has occurred when accessing the operator's screen. *
SWAP I/O ERROR	An irrecoverable I/O error has occurred on this program's segment of the swap file. More details appear in a message displayed on the affected operator's screen. *
SEPARATE COMPUTER	The screen is attached to a separate computer.
SEPARATE SYSTEM	The screen is attached to a separate system.
FILE SERVER	This system is a file server on a Global System Manager (Unix) or Global System Manager (Novell) configuration.

* These conditions cannot be detected for users of a different computer in a networking environment.

Table 37.2 - Status Messages and their Meanings

37.2 Task Control Instructions

When you have a supervisor authority code and run the \$STATUS command the following prompt appears immediately following the status report:

```
$47 TASK CONTROL:
```

You may key any of the instructions listed in table 37.1.

In a network configuration you can only use the RES instruction for users sharing your computer, and not for users on different computers. Similarly, the TER instruction will only terminate execution of Global System Manager on your own computer. However, both the LOC and FIL instructions can be used to report on files on other file servers on the network.

Note that not all of these instructions are available under some host operating systems. Please consult the Global Operating Manual for further details.

Unlike most of the other Global System Manager commands, \$STATUS calls the LOGOF\$ subroutine, so that it is possible to terminate a Global System Manager session while \$STATUS is running in another partition on your screen or on another screen on your computer.

37.2.1 To Quit

To quit and return to the menu or GSM READY: prompt simply key <ESC> in response to the task control prompt.

37.2.2 CAN - Cancel a User's Jobs

You use the cancel instruction to cancel all partitions of a particular user. CAN frees any resources such as files and printers that the jobs were using, so it is normally employed as a matter of course if the status report shows a job to have failed. You key the instruction followed by the user number or operator-id. For example:

```
$47 TASK CONTROL:CAN USER:3 CANCELLED
$47 TASK CONTROL:
```

If you use the cancel instruction wrongly, one of the following error messages appears and the instruction has no effect:

```
INVALID
```

or:

```
CANNOT CANCEL SELF
```

or:

```
CANNOT CANCEL USER ON ANOTHER PROCESSOR
```

The first is displayed if you key a user number or operator-id not corresponding to one displayed on the status report. The remaining two mean that you specified your own user number or computer-id, or one of a user of another computer in a network configuration.

When the cancel operation is successful Global System Manager attempts to display the following message on the affected operator's screen:

```
$94 SESSION CANCELLED BY SUPERVISOR
```


Even if this immediately causes another screen I/O error, any resources that the user program possessed will have been released.

The CAN command will also have the effect of returning the user to the host operating system. Attempts to log back to Global System Manager will fail because Global System Manager is no longer active on that screen.

The CAN instruction only affects screens on the same computer (or system). It cannot be used to cancel users on another computer (or system).

37.2.3 COM - Report on Computer Arrangement

The COM instruction displays a table showing the computers on the network on which Global System Manager has been used since the user file was last purged together with their current status (IN USE, NOT IN USE or ** OURS **) and the descriptions set up for them using DES.

37.2.4 DES - Amend Table of Computer Users

This instruction allows you to update the table of computer descriptions. You are prompted for the computer-id you wish to update and then asked whether you want to delete or amend the present description.

If you key <CR> to amend, the present description is displayed so that you can type a replacement description of up to 50 characters in length.

If you key D to delete, the description will be removed from the table, and subsequent \$STATUS displays will show the computer with a blank description field.

To add a description to the table, key the computer-id followed by <CR> to amend. A new description may now be entered.

37.2.5 DIS - Disconnect Screen and User

The DIS command is equivalent to the RES command, but in addition, the user is returned to the host operating system.

Nevertheless Global System Manager will have attempted to redisplay the sign-on text, the display will have been stalled because the screen has been logically disconnected from Global System Manager. If, however, you log back on to Global System Manager, the stalled display will be freed and the operator-id prompt displayed.

37.2.6 FIL - List Open Files

You use the FIL instruction to list all the files that are currently open on a computer. This is particularly useful if you want to check whether any other users are in the process of updating files before, for example, switching the computer off.

On a network configuration, \$STATUS prompts for the LAN-id of the computer you want to enquire about. you can either a specify a LAN-id, <CR> for your own computer, or <CTRL B> for all computers on the network.

The FIL instruction displays a list of open files each followed by a list of users with the particular file open, detailing their user numbers, computer-ids, operator-ids and partition numbers.

The maximum open count that is displayed for shared files is 126. If a particular file is opened more than 126 times the open count will be displayed as 126.

37.2.7 GET - Get Exclusive Control

You use the GET instruction to acquire exclusive control of your computer. The instruction should be employed with great care since until exclusive control is released your screen is the only one able to operate and all the other users of the computer are locked out (including any other partitions on your screen). For example:

```
$47 TASK CONTROL:GET EXCLUSIVE CONTROL GRANTED
$47 TASK CONTROL:
```

Note that on a network configuration, GET does not prevent operators using other computers from working, nor does it stop them from accessing files that may be resident on your computer from across the network: It merely gives you exclusive control of your own computer.

37.2.8 LIL - List in use computers

The LIL instruction lists the status report but does not include those computers that are NOT IN USE. The LIL instruction is ignored on Global System Manager (Unix) configurations, where the default status report, produced by the <CR> instruction, excludes any SYSTEM's that are NOT IN USE.

37.2.9 LIS - List details of a selected computer only

This command prompts for a computer-id or system-id and displays the status report for that computer or system only.

37.2.10 LOC - List Active Locks

The LOC instruction produces a list on the screen of all the currently active file locks on a computer. The name of the file, its unit address, the user number of the user currently locking it, the computer LAN-id (if there is one), the operator-id and partition number of the user currently locking the file, and the 4 byte region code and the number of times it has been locked at that region are shown. If there are no active locks the message 'NO LOCKS OUTSTANDING' is displayed.

If you are working on a network configuration then \$STATUS prompts for the LAN-id of the computer you want to enquire about. You can either specify a LAN-id, <CR> for your own computer, or <CTRL B> for all computers on the network. The entries are displayed with one lock per line in the following form:

```
ffffff UNIT uuu USER n COMPUTER l OPERATOR oooo/p LOCKED #hhhhhhh m TIMES
```

For example:

```
$$SP UNIT X18 USER 8 COMPUTER X OPERATOR ALAN/8 LOCKED #46494C45 1 TIME
```

For shared locks, the index in the Lock Table is displayed. For example:

```
LKWORK UNIT X01 SHARED LOCK, LK 27 LOCKED #00000000 2 TIMES
```

If the lock applies to a Speedbase record, then if possible, the record type, record length and relative record number of the record locked are displayed.

If the LWORK\$ subroutine has been used to lock a work unit then some of the lock information displayed by \$STATUS is replaced by asterisks, and a hexadecimal dump of part of the Lock Table entry is displayed. For example:

```
***** UNIT *** USER 4 COMPUTER X OPERATOR ALAN/4 LOCKED #58303120 1 TIME
#0400FF5830312001
```

37.2.11 LOG - Write Record to Event Log File

The LOG command allows you to write a record to the Event Logging file \$\$LOG, recording some special system event. Consequently it can be used to comment log records that have been produced by the system automatically. For example, following a re-bootstrap of a the computer during the middle of the day you might choose to record the reason for the re-start.

The LOG command prompts for a message which may be up to 50 characters in length:

```
Key message to log:
```

37.2.12 MES - Send a Message to Another User

The MES instruction allows you to send a message of up to 30 characters to a specific user, users or group of users. After keying MES the following prompt is displayed:

```
Send message:
```

When a message has been supplied, you are prompted for a target user (or users):

```
To user:
```

You should key either the user number or operator-id if you wish to send the message to a single specific user, <CTRL B> to send the message to all users on a computer or all users on a network.

To send a message to a group of operators, groups having been defined using \$GROUP, key <CTRL A> to the user prompt. The following message will then be displayed on the screen:

```
Key group-id, ? to list:
```

Key either the group-id or ? to list the current groups.

After the message has been sent the user prompt will be repeated, thus allowing you to send the same message to another user or group of users.

Messages are displayed on the status line of the screen and can be cleared by keying <SYSREQ> Z (see Chapter 4 of the Global System Manager Manual).

Depending on the length of the message that you supply, useful system information may be appended to the text of the message. If the length of the message is 19 characters, or less, the message text is prefixed by the current time (i.e. *hh:mm*) and the operator-id of the sender (i.e. *oooo*). If the length of the message is between 20 and 25 characters, the message text is prefixed by the operator-id of the sender (i.e. *oooo*). If the length of the message is more than 25 characters no extra information is appended to the message.

For example:

*Message keyed**Message sent as*

```

ABCDEF GHIJK LMNOPQRS          hh:mm          0000
ABCDEF GHIJK LMNOPQRS
ABCDEF GHIJK LMNOPQRST        0000 ABCDEF GHIJK LMNOPQRST
ABCDEF GHIJK LMNOPQRSTU VWXY  0000 ABCDEF GHIJK LMNOPQRSTU VWXY
ABCDEF GHIJK LMNOPQRSTU VWXYZ ABCDEF GHIJK LMNOPQRSTU VWXYZ

```

37.2.13 MON - Monitor Serial Lines

The MON command can be used to monitor serial terminals that are connected to the computer via modems, and makes sure that if the communications line goes down the Global System Manager session on that terminal is returned to a known state. This facility provides protection against unauthorised use of a user session that might be left otherwise be left open. Only a limited number of computers have the appropriate software and hardware to support this command. Consult the Global Operating Manual for further details.

Once in monitor mode \$STATUS watches all serial lines and when it detects that the modem control lines have "dropped" (i.e. that the telephone connection has been broken) it automatically re-starts that screen (as if a RES instruction had been executed). Global System Manager then attempts to transmit the sign-on dialogue, but this will be stalled until a connection is re-established. When the connection is re-established (the remote user dials in again) the sign-on dialogue will be transmitted and the operator can sign-on.

When you first start monitor mode by keying MON, \$STATUS prompts for the time in seconds between scans of the serial lines:

```
Delay between each check of the serial lines in seconds (60):
```

You can specify a different frequency for the checking, but too small a value may affect the performance of the computer. \$STATUS then displays the following message:

```
Line monitor running, key <CTRL G> to interrupt
```

Keying <CTRL G> terminates the line monitoring and returns you to the \$47 TASK CONTROL prompt.

37.2.14 PRI - Print current details

The PRI instruction prompts for a listing unit and then produces a report on the **current** status. The information normally displayed by the <CR> (status list), FIL (open files) and LOC (outstanding locks) instructions is included in the report. Details of all computers on a network system are printed.

The PRI instruction only prints the current 'snapshot' of the system status at the time the report is produced.

37.2.15 PUR - Purge User File

The PUR instruction causes the user file to be deleted and reallocated when the system is next bootstrapped. This has the effect of purging it of irrelevant, incorrect or out of date information. For example, if a particular computer has been signed on to the network at different times with different configuration files or network addresses it may appear more than once in the status

report; this can lead to confusion since the messages ACTIVE, INACTIVE and COMPUTER NOT IN USE can appear next to the same computer identification letter. Using the PUR instruction means that when the system is next bootstrapped only the 'genuine', current users are listed.

Once a PUR command has been issued you must re-load Global System Manager on all of the computers on the network. If other computers are allowed to sign on before Global System Manager has been reloaded on the master computer some of the methods that Global System Manager uses to ensure system integrity are negated, and file corruption can result.

Any users who are specified as automatically signing on in the \$AUTH authorization file (see Chapter 6 of the Global System Manager Manual) will have to be signed on manually the first time the system is bootstrapped after purging the user file, as this information is lost when the file is reallocated.

37.2.16 QUI - Quiesce the System

You use the QUI instruction to quiesce the system towards the end of a working session so that new operators cannot sign on, or to prevent users signing onto the computer while you are performing sensitive operations (e.g. while installing new software). It affects all users, whether or not they are sharing your computer. As each existing operator signs off his or her screen is locked out. For example:

```
$47 TASK CONTROL:QUI ARE YOU SURE?:Y QUIESCE IN PROGRESS
$47 TASK CONTROL:
```

A quiesce which is in progress can be cancelled by rekeying the QUI instruction and replying N to the ARE YOU SURE? prompt. If you do this new operators will be able to sign on to the system. For example:

```
$47 TASK CONTROL:QUI ARE YOU SURE?:N
$47 TASK CONTROL:
```

Normally the supervisor responsible for system shutdown will contact the other operators (e.g. using the \$STATUS MES instruction or \$M or \$MAIL) to tell them that the work session is nearing its end before keying the quiesce instruction. They are expected to sign off as soon as it is convenient. The supervisor will then examine the status report periodically until he or she determines that no-one else is on the system. Finally the supervisor will terminate Global System manager using the TER instruction, described below, and then power down the system. Figure 37.2 shows a typical status report on the supervisor's screen at the end of the session.

37.2.17 REL - Release Exclusive Control

The REL instruction releases exclusive control if you had previously acquired it using GET, or if an application program had obtained it by calling the GETX\$ system routine:

```
$47 TASK CONTROL:REL EXCLUSIVE CONTROL RELEASED
$47 TASK CONTROL:
```

The instruction always displays the message 'EXCLUSIVE CONTROL RELEASED' even if you did not have exclusive control in the first place.

37.2.18 RES - Restart a User

You use the RES instruction to restart a user whose screen has been locked out. A lockout condition will be indicated by one of the following error messages appearing on the status report for the user:

CANCELLED

or:

SWAP I/O ERROR

or:

SCREEN I/O ERROR

Except in the case of a screen error, a further \$94 message appears on the affected screen itself. To restart, key the instruction followed by the user number or operator-id:

```
$47 TASK CONTROL:RES USER:AJU RESTARTED
$47 TASK CONTROL:
```

The restart will free any resources such as files and printers that the programs were using.

If you use the restart instruction wrongly, one of the following error messages appears and the instruction has no effect:

INVALID

or:

CANNOT RESTART SELF

or:

CANNOT RESTART USER ON ANOTHER PROCESSOR

The first is displayed if you key a user number or operator-id not corresponding to one displayed on the status report. The remaining two mean that you specified your own user number or operator-id, or one of a user of another computer in a networking system. When the restart operation is successful the sign-on message appears on the affected screen, followed by an operator prompt. The user must sign on again.

The RES instruction only affects partitions on the same computer (or system). It cannot be used to restart users on another computer (or system).

37.2.19 TER - Terminate Global System Manager

You use the terminate instruction under when you wish to close down Global System Manager on your computer, while Global System Manager continues to run on other computers. It can also be used to close down a single computer system. TER checks that there are no other users signed on to your computer nor any other users on the network accessing files on your computer. If these conditions are satisfied it then frees any central resources used for networking, and terminates execution.

The terminate instruction causes you to return to the host operating system. However, if the host operating system is BOS you will have to bootstrap the computer again if you wish to reload Global System Manager.

If other operators are still active \$STATUS displays the following warning message and will not terminate Global System Manager:

OTHER USERS STILL SIGNED ON THIS PROCESSOR

If other operators are accessing files on your computer across a network \$STATUS displays the following warning message and will not terminate Global System Manager:

```
FILE filename TYPE type ON UNIT uuu SHARED n USERS
      User n Computer a Operator oooo
OTHER USERS HAVE THESE FILES OPEN ON YOUR COMPUTER
YOU MAY NOT TERMINATE UNTIL THEY HAVE FINISHED
Key <CR> to retry, <ESC> to abandon
```

Key <CR> to retry the TER instruction again once these users have signed off (by using \$E - see Chapter 4 of the Global Operating Manual), or closed the necessary files (usually by exiting from an application).

If \$STATUS prevents the termination of the Global System Manager session because other operators are accessing files on your computer across a network, you can key <CTRL B> to the retry prompt to terminate Global System Manager **WITHOUT** performing the open file checking. This option may be useful if a workstation on a network has been rebooted while running Global System Manager resulting in spurious open files on a file-server. Without this option it is not possible to use the \$STATUS TER instruction to terminate Global System Manager on the file-server. **THIS OPTION MUST BE USED WITH GREAT CARE BECAUSE TERMINATING GLOBAL SYSTEM MANAGER WHILE FILES ARE GENUINELY IN-USE MAY CAUSE DATA CORRUPTION.**

38. \$T - Obtain Terminal Information

\$T, the Obtain Terminal Information utility, is fully documented Chapter 5 of the Global System Manager Manual.

39. \$TDUMP - Cartridge Tape Dump and Restore

The functionality of the \$TDUMP utility is dependent on the host operating system. \$TDUMP is documented in Chapter 4 of the Global Operating Manual.

40. \$TED - Text Edit Utility

\$TED, the Text Edit utility, is fully documented in Chapter 5 of the Global System Manager Manual.

41. \$U - Obtain Unit Information

The \$U command displays a series of screens describing the units supported on your system. You can print this information if you wish:

```
GSM READY:$U
.....
.....      (first screen of unit information)
.....
Key P to print, <CR> to page, <ESC> to exit:<CR>
.....
.....      (second screen of unit information)
.....
Key P to print, <CR> to page, <ESC> to exit:<CR>
.....
.....      (eventually the display will wrap round, and the
.....      first 'page' will reappear)
.....
Key P to print, <CR> to page, <ESC> to exit:P
.....      (all the information is printed)
GSM READY:
```

You should keep this and the other printouts of system information for reference purposes.

41.1 Devices on a Network or Unix Configuration

On network or Global System Manager (Unix) configurations additional dialogue appears.

41.1.1 Devices on a Network Configuration

On network configurations there is additional dialogue which enables you to find out the units that are supported by other computers on the network. In this case the final prompt is of the form:

```
Key P to print, C for different computer, <CR> to page, etc.
```

If you key C a further prompt will appear:

```
Specify computer-id (A):
```

The default given in brackets is the computer identification letter of the master computer on the network. You should key <CR> to obtain details of the units it supports, or key the computer identification letter of another computer.

Important note: Information about printer units or tape drives is not given for other computers on the network.

41.1.2 Devices on a Global System Manager (Unix) Configuration

On Global System Manager (Unix) configurations there is additional dialogue which enables you to find out the units that are supported by other systems in the configuration. In this case the final prompt is of the form:

```
Key P to print, S for different system, <CR> to page, etc.
```

If you key S a further prompt will appear:

```
Specify System-id (A):
```

The default given in brackets is the identification letter of the master system process in the configuration. You should key <CR> to obtain details of the units it supports, or key the identification letter of another system.

Important note: Information about printer units or tape drives is not given for other systems.

41.2 Diskette Devices

You may find that sometimes more than one unit address is used for the same diskette drive. This enables different diskette formats (**not** diskettes of different physical sizes) to be used on the same drive, providing the appropriate address is specified. For example, an IBM PC compatible 3½" diskette drive can normally support the following types of 3½" diskettes amongst others:

<i>Unit</i>	<i>Format</i>	<i>Capacity</i>	<i>Tracks</i>		<i>Sectors/track</i>	<i>Sides</i>
140	O2A	1.44 Mb	80	18	2	
170	B3B	720 Kb	80	9	2	

Normally you will work with the addresses associated with the highest capacity format available. Other addresses, if present, may be employed for accessing diskettes formatted on other machines or the special format SYSIPL volume, if one is used by the bootstrap procedure.

41.3 Local Hard Disks

\$U displays the following details of each hard disk attached to your computer:

- Domain volume description and formatted capacity;
- Disk geometry (e.g. number of cylinders and sector size);
- Unit numbers of the subunits on the domain;
- Maximum number of files per sub-volume.

Note that on some host operating systems the disk geometry displayed by \$U, and used by Global System Manager, may describe a "virtual disk" rather than a physical device. Consult your Global Operating Manual for further details.

41.4 Printers

\$U also describes the unit numbers and descriptions of each printer attached to your computer. The printer unit descriptions shown by \$U are displayed in error messages of the form:

```
* NOT READY ERROR on printer description - uuu
```

41.5 Hard Disks on the network Master

As explained in Chapter 2 of the Global System Manager Manual, the hard disk units (i.e. units in the range 200 to 299) on the master computer of a network are aliased to units 6xx on **all** the computers on the network. The information displayed for the hard disks on the master (units 600 - 699) is in the same format as displayed for local hard disks (see section 41.3).

41.6 Tape Drives

If your computer contains a tape drive which can be used by Global System Manager (i.e. by either \$TDUMP or \$TAPE), the following line will appear at the end of the \$U report:

```
TAPE DRIVE
```

On some host operating systems, tape backups and restores can only be performed by utilities supplied with the host system. In this case \$TDUMP and \$TAPE will not be available. Consult your Global Operating Manual for further details.

```

Global System Manager
Window Edit Functions Options Help

Global System Manager
UNIT FORMAT  DESCRIPTION
190  T259Z  3rd Discrete Data File                < 99 FILES >
      4 SURFACE(S) OF 16383 TRACKS WITH 64 512-BYTE SECTORS
191-199      SUBUNITS OF DOMAIN 190
200  T259Z  1st Discrete Data File                <250 FILES >
      4 SURFACE(S) OF 16383 TRACKS WITH 64 512-BYTE SECTORS
201-259      SUBUNITS OF DOMAIN 200
260  T259Z  2nd Discrete Data File                <250 FILES >
      4 SURFACE(S) OF 16383 TRACKS WITH 64 512-BYTE SECTORS
261-299      SUBUNITS OF DOMAIN 260
500          DOS Printer
501          DOS Printer
510          Windows Printer
520          DOS Printer
Key P to print, C for different computer, B for back, <CR> to page, <ESC>:_

Partition 1 Computer 1B
CAPS

```

Figure 41.1 - A typical \$U report

42. \$USAGE - File Block and Channel Diagnostics

The \$USAGE utility displays details of three of the important control blocks used by the File Executive. In particular, \$USAGE displays the usage of the following control blocks:

- File channels (DC-blocks). A file channel is required each time an application opens a file (regardless of whether the file is already open by another user);
- File blocks (DF-blocks). A file block is required for every distinct open file;
- Lock Table Entries (LK-blocks). An entry in the Lock Table is required for every file that is locked.

If an insufficient number of any of these of control blocks is configured on a system, the following error message will occur when running Global 2000 and Global 3000 applications:

```
* TOO MANY OPEN FILES ON description - uuu - RETRY?:
```

The number of File Channels, File Blocks and Lock Table Entries can be increased using either Global Configurator (see the Global Configurator Manual) or \$CUS (see Chapter 6 of the Global System Manager Manual).

42.1 Using \$USAGE

\$USAGE has been designed to run in a background partition, to constantly monitor Global System Manager resources. \$USAGE suspends processing for a variable time-period (see section 42.1.2, below) in order to conserve computer resources.

42.1.1 Specifying the target computer (or system)

When \$USAGE is run from either a menu or the GSM Pn READY: prompt it displays the following prompt to allow you to specify the computer (or system) to be diagnosed:

```
Specify node-id of target computer (A):
```

The reply must be an upper-case letter. The default is the computer node-id of your computer, or the node-id of the master computer if your computer is not a file-server. The special reply of <CTRL C> allows you to specify your own computer if it is not configured as a file-server.

Note that this prompt will not appear on single computer Global System Manager configurations.

On a multiple computer configuration (e.g. a Global System Manager (Novell NetWare) configuration) or a Global System Manager (Unix) configuration, \$USAGE will display the following message:

```
Attached to LAN Executive
```

On a single computer configuration, or if you reply <CTRL C> to the above prompt, \$USAGE will display the following message:

```
Attached to File Executive
```

42.1.2 The Suspend Period

In order to conserve computer resources, \$USAGE does not constantly monitor the control block usage but only examines the control blocks every few seconds, or so. The "sampling rate" may be specified in terms of seconds. The following sample dialogue will cause \$USAGE to only examine the control blocks every 15 seconds:

```
Enter suspend period in seconds (10):15
```

42.1.3 \$USAGE modes

\$USAGE can operate in one of three modes:

- Display-only mode: Examine and display the control block usage at the specified sampling rate;
- Monitor-only mode: Examine control block usage at the specified sampling rate but only display warning messages when the control block usage crosses a pre-defined threshold;
- Monitor-and-Display mode: Examine and display the control block usage at the specified sampling rate AND display warning messages when the control block usage crosses a pre-defined threshold.

42.1.3.1 \$USAGE in display-only mode

To run \$USAGE in display-only mode, reply N to the following prompt:

```
Display warning if usage too high ? (Y):N
```

The \$USAGE diagnostic messages (see section 42.2) will be displayed at the specified sampling rate (e.g. every 10 seconds).

42.1.3.2 \$USAGE in monitor-only mode

To run \$USAGE in monitor-only mode, terminate the reply to the "suspend period in seconds" prompt (see section 41.1.2, above) with <CTRL A>. For example:

```
Specify suspend period in seconds (10):10<CTRL A>
```

\$USAGE will respond by displaying the following message:

```
Warning display enabled
```

and prompt for the control block usage thresholds which must be expressed as a percentage usage. A different percentage can be specified for each of the three control blocks monitored by \$USAGE. For example:

```
Specify percentage of DC-blocks for warning (75):80
Specify percentage of DF-blocks for warning (80):85
Specify percentage of LK-blocks for warning (85):90
```

When any of the control block usage thresholds are exceeded, \$USAGE will display a warning message as described in section 42.3.

42.1.3.3 \$USAGE in monitor-and-display mode

To run \$USAGE in monitor-and-display mode, reply Y, or <CR>, to the following prompt:

```
Display warning if usage too high ? (Y):Y
```

\$USAGE will prompt for the control block usage thresholds as described in section 42.1.3.2, above.

When any of the control block usage thresholds are exceeded, \$USAGE will display a warning message as described in section 42.3, below.

42.2 The \$USAGE Diagnostic Messages

In display-mode or monitor-and-display mode, \$USAGE displays the following diagnostic information at the specified sampling rate:

Time	The current time-of-day. Note that because of rounding in the Global System Manager time-slicing, the time will not always be incremented by the sampling rate;
DCtot	The total number of File Channels (DC-blocks) configured. This value will remain constant;
DCuse	The number of File Channels (DC-blocks) that were in use during the sampling. This "snapshot" value will vary according to application usage. If this value ever reaches the DCtot value the following message will be displayed on a screen somewhere on the system:
	* TOO MANY OPEN FILES ON ...
DCmax	The maximum number of File Channels (DC-blocks) that have been in use since Global System Manager was loaded on the computer that is being diagnosed. This "high watermark" value will increase monotonically;
DFtot	The total number of File Blocks (DF-blocks) configured. This value will remain constant;
DFuse	The number of File Blocks (DF-blocks) that were in use during the sampling. This "snapshot" value will vary according to application usage. If this value ever reaches the DFtot value the following message will be displayed on a screen somewhere on the system:
	* TOO MANY OPEN FILES ON ...
DFmax	The maximum number of File Blocks (DF-blocks) that have been in use since Global System Manager was loaded on the computer that is being diagnosed. This "high watermark" value will increase monotonically;
LKtot	The total number of Lock Table Entries (LK-blocks) configured. This value will remain constant;
LKuse	The number of Lock Table Entries (LK-blocks) that were in use during the sampling. This "snapshot" value will vary according to application usage. If this value ever reaches the LKtot value the following message will be displayed on a screen somewhere on the system:

* TOO MANY OPEN FILES ON ...

LKmax The maximum number of Lock Tables (LK-blocks) that have been in use since Global System Manager was loaded on the computer that is being diagnosed. This "high watermark" value will increase monotonically.

Any time \$USAGE is running in display-mode or monitor-and-display mode you can abort the command and return to the "suspend period" prompt (see section 42.1.2, above) by keying <CTRL G>.

42.3 The \$USAGE Warning Messages

In monitor-mode or monitor-and-display mode, \$USAGE displays the following diagnostic information whenever the threshold usage for a control-block is being exceeded:

```
xx-block limit exceeded ( nn%)
```

where *xx* is the type of control block (e.g. DC, DF or LK) and *nn* is the actual percentage usage. For example:

```
DC-block limit exceeded ( 90%)
```

Any time \$USAGE is running in monitor-mode or monitor-and-display mode you can abort the command and return to the "suspend period" prompt (see section 42.1.2, above) by keying <CTRL G>.

As a rule of thumb, you are recommended to use Global Configurator or \$CUS to increase the number of a particular control block if the usage of that control block ever exceeds 75%.

43. \$V - Preparing Diskettes and Hard Disks

The functionality of the \$V utility is dependent on the host operating system. \$V is documented in Chapter 4 of the Global Operating Manual.

44. \$VOL - List Volume Directory

The \$VOL utility produces an expanded directory listing of a volume, combining a listing of the files, as given by \$F, with details of the contents of libraries. Thus the listing produced by \$VOL gives a full listing of all the programs and compilations a volume contains, whether they appear as separate files or within a library. All types of libraries are recognised by \$VOL: Program libraries (created and maintained using \$LIB, see Chapter 21), compilation libraries (created and maintained using \$LIB, see Chapter 21) and data file libraries (created and maintained using \$DLIB, see Chapter 12).

If a member of a program library contains symbolic debug records, a '*' is added to the end of the member name. All files within data file libraries also appear and are listed as members of their library.

Copy libraries are also analysed to produce a listing of the contents. Certain assumptions are made about the format of the copy library: there are assumed to be no lines other than comments and PAGE statements preceding the first book; if there are lines preceding the first book the first such line is assumed to contain the library title; any text following the book name is taken as the title of the book.

44.1 The \$VOL Output Unit Prompt

When you run \$VOL, it first prompts for an output unit:

```
$505 LISTING UNIT:
```

Reply <CR> if you want the listing printed on \$PR, <CTRL A> if you want it displayed on the screen, or enter a unit-id to cause the listing to be printed on that unit.

44.2 The \$VOL Option Prompt

After you have supplied the output unit, an option prompt will appear:

```
$505 DIRECTORY OPTION:
```

Reply <CTRL C> to list the available options. Normally, the default options are sufficient, and you should key <CR> to use these.

44.3 The \$VOL Title Prompt

When you reply <CR> to the option prompt, a title prompt appears:

```
$505 TITLE (UP TO 30 CHARS):
```

If you supply a title, of up to 30 characters, this will be printed on each page of the listing. If you reply <CR>, the title will be left blank.

44.4 The \$VOL Unit Prompt

When you have supplied a title, a unit prompt appears. You should mount the volume to be listed, and supply its unit address. The unit may be a volume or a domain.

If a domain is specified you will be prompted for each subvolume in turn. Key <CTRL B> to list all the subvolumes on the domain.

When the report is complete, \$VOL displays an explanatory message and redisplay the unit prompt so that further volumes can be listed.

If you reply <CTRL A> to the unit prompt, the listing unit prompt will be redisplayed.

Reply <CR> or <ESCAPE> to the unit prompt to quit.

45. \$VOLSAV - Restore/Save Domain Layout

The \$VOLSAV utility saves the layout of a domain (i.e. sub unit allocation and Volume Descriptions) in a template file. The same utility may also be used to recreate a domain layout from a template file. It is expected that \$VOLSAV will be particularly useful when used in conjunction with \$TAPE, the Streamer Tape Utility.

\$VOLSAV only saves and restores the **layout** of a domain. It does not save the **contents** of a domain.

45.1 The \$VOLSAV Mode Prompt

When you run \$VOLSAV, it first prompts for an operating mode:

```
GSM READY:$VOLSAV
Key Save, Restore domain, <CR> to exit:
```

Key S to save a domain layout, R to restore a domain layout or <CR> to quit.

45.2 Using \$VOLSAV to Save a Domain Layout

In "save" mode, \$VOLSAV first prompts for the address of the unit to be saved. This unit must be a domain.

\$VOLSAV then prompts for the name and unit of the domain layout file. The default filename is \$\$DOMxxx, where xxx is the address of the domain.

The following example dialogue saves the layout of domain A00 in a domain layout file \$\$DOMA00 on unit \$DP:

```
GSM READY:$VOLSAV
Key Save, Restore domain, <CR> to exit:S

Domain unit:A00
Save under filename:$$DOMA00 Unit:$DP
Reading domain information...
DOMAIN SAVED
GSM READY:
```

45.3 Using \$VOLSAV to Restore a Domain Layout

In "restore" mode, \$VOLSAV first prompts for the name and unit of the domain layout file. This must be a template file created using the "save" mode. \$VOLSAV then prompts for the address of the unit to be recreated. This unit must be a domain.

If the domain is not currently empty, it may be scratched before the layout is recreated. If this option is selected all data on the volume will be destroyed but the directory will be rebuilt to exactly reproduce the layout saved in the template file. If the domain is not scratched before the layout is recreated, only those subunits listed in the template file that are not already allocated on the domain will be created.

The following example dialogue completely restores the layout of domain A00 from a domain layout file \$\$DOMB00 on unit \$DP. All the data on unit A00 will be destroyed:

```
GSM READY:$VOLSAV
Key Save, Restore domain, <CR> to exit:R
```

Chapter 45 - Restore/Save Domain Layout (\$VOLSAV)

```
Key save file name:$$DOMB00 Unit:$DP
Key domain unit id:A00
Scratch domain (Y/N)?:Y
Reallocating domain layout...
DOMAIN REALLOCATED
```

During a restore, the following error messages may appear:

```
VOLUME xxxxxx UNIT nn IS OUTSIDE DOMAIN UNIT RANGE
```

```
VOLUME xxxxxx IS TOO LARGE TO RESTORE
```

\$VOLSAV uses \$V (see Chapter 4 of the Global Operating Manual) to reallocate each sub-unit in turn. Consequently, any errors reported by the \$V "Initialise sub unit" option may appear during a \$VOLSAV "Restore".

The version of \$VOLSAV distributed as part of Global System Manager V8.1 and V8.0 saves and restores spooler and work file allocation sizes. The format of the \$\$DOMxxx file created by \$VOLSAV V8.1 is the same as that created by \$VOLSAV V8.0. However, the format of the \$\$DOMxxx file created by \$VOLSAV V8.1 (and \$VOLSAV V8.0) is different from that created by \$VOLSAV V7.0.

The corollary is that \$VOLSAV V8.1 and V8.0 will restore domain layouts saved using \$VOLSAV V7.0. However, \$VOLSAV V7.0 will NOT restore domain layouts saved using \$VOLSAV V8.1 and V8.0.

46. \$ZAP - Applying a Zap

The \$ZAP utility is fully documented in Chapter 6 of the Global System Manager Manual.

Appendix A - How Program Errors are Handled

When a program error is detected, Global System Manager invokes the error reporting component of the debugging system by loading a special system request, \$OV\$?. This system request is present in the P.\$OV\$ library on SYSRES. Unless you are using the symbolic debugging system described in the Global Cobol User Manual you will not be able to resume the failing program, but you will be able to obtain the necessary diagnostic information required to investigate the program error.

A.1 Program Checks - Errors Detected by the Interpreter

The Global Cobol interpreter translates compiled Global Cobol programs held on disk, or in memory, into machine code instructions which a particular computer can execute. The interpreter will signal a program check if a program attempts an illegal instruction, such as a division by zero, or if the program itself has been corrupted. In either case Global System Manager displays an error message of the form:

```
$91 PGM CHECK number AT location
```

For example:

```
$91 PGM CHECK 11 AT 1EF2
```

The *number* is an integer between 2 and 16 inclusive, indicating the type of program check which has occurred (6, 7 and 8 in particular indicate program corruption), and the *location* identifies the address of the failing instruction. This need not necessarily correspond to the starting location of a Global Cobol statement, because a single statement sometimes compiles as several instructions.

The meanings of the different types of program check which may occur are described in detail in Appendix B.

A.2 Stop Codes - Errors Detected by Global Software

If Global System Manager, or a software module running under it, detects a condition that prevents the current program from continuing Global System Manager displays an error message of the form:

```
$91 TERMINATED - STOP code
```

For example:

```
$91 TERMINATED - STOP 8001
```

The *code* defines the condition that led to the program being terminated. Stop codes greater than 99 are generated by Global System Manager and are defined in Appendix C. Codes between 1 and 99 are available for general use and should be described in the documentation of the failing program itself.

A.3 Exit Codes - Failure to Respond to an Exception

Many Global Cobol statements signal a condition known as an **exception**, to indicate that an event has occurred which requires special handling by the controlling program. For example, an

exception is signalled when the end of an input file is reached, or on failure to find the specified key in a table which is being searched.

A statement which can generate an exception should normally be followed by an ON EXCEPTION statement to introduce the code to handle the special condition. If this is not the case and an exception occurs, the interpreter detects that the program in control does not contain the necessary logic to cope with the event, and Global System Manager displays an error message of the form:

```
$91 TERMINATED - EXIT code
```

For example:

```
$91 TERMINATED - EXIT 1201
```

The *code* defines the exception condition which the faulty program failed to trap. Exit codes greater than 99 are generated by Global System Manager and are defined in Appendix D. Codes between 1 and 99 are available for general use, and will be described in the documentation of the failing program itself.

A.4 Program Traps

The following message:

```
$91 TRAP AT location
```

usually appears as the result of a programmer using the debugging system to set the trap flag in a particular Global Cobol statement as explained in the Global Cobol User Manual. When it occurs during operational running it normally indicates that the program has erroneously branched to an area which does not contain valid Global Cobol instructions (e.g. part of the data division). This message can also indicate that the copy of the program held on external storage has become corrupt. The *location* is the address of the faulty instruction that the program was attempting to execute.

A.5 Programs Cancelled by the Operator

If you suspect that a program is looping you can interrupt it by keying <CTRL W>, or <SYSREQ> W if the keyboard you are using has a prefix key instead of CTRL. Global System Manager then displays the break prompt:

```
$91 BREAK?:
```

You can now key Y to enter the appropriate debugging system, as explained in section A.6 below. A response of N or <CR> causes the interrupted program to be resumed as though the break had never taken place. This response is useful if you hit <CTRL W> by mistake or simply used the break facility to temporarily halt a program which was monopolizing a multi user system.

A.6 Diagnostic and Debugging Facilities

Following any of the error messages:

```
$91 PGM CHECK number AT location
```

Appendix A - How Program Errors are Handled

\$91 TERMINATED - STOP *code*

\$91 TERMINATED - EXIT *code*

\$91 TRAP AT *location*

or a reply of Y to the break prompt, Global System Manager displays the diagnostics prompt:

Activate debugger, key D for Diagnostics report, H for Help, <ESC> to exit:

Note that if the screen is being used in formatted mode the initial error prompt (e.g. \$91 PGM CHK) is displayed on the baseline followed by a "," prompt. If the "comma prompt" appears, you must key <CR> to display the actual diagnostics prompt.

Program errors are dealt with differently if you are using \$DEBUG to debug a program. Consult the Global Cobol User Manual for further information.

When the diagnostic prompt appears, you may key one of the following replies:

- H to display a screenful of help text;
- <CTRL A> to display diagnostic information on the screen;
- D to send the diagnostic information to the standard printer device \$PR. You can associate up to 10 lines of text with the diagnostic report. The cursor keys and other editing keys can be used to enter the text. Note that until you have left the text edit mode by keying <ESC> you cannot switch partitions;
- <ESC> to abandon your program and return to the GSM Pn READY: prompt or the menu without producing a diagnostic report.

Important note: Although earlier versions of Global System Manager allowed R (to resume), this facility has been removed. Do not attempt to use this technique as the results will be unpredictable.

If the diagnostics prompt appears unexpectedly you should reply D or <CR> to cause a diagnostic report to be written to the standard printer assigned to \$PR. If you know the cause of the problem and do not need to produce a report, simply key <ESC> to return to the GSM Pn READY: prompt or menu in the normal way. After the diagnostic report has been created, the following prompt will be displayed:

Full memory dump?:

This allows you to produce, in addition to the diagnostic report, a dump of the whole of memory (typically 20-50 pages). You should reply N or <CR> if you do not want the full dump, or Y to produce the dump. Normally just the diagnostic report and a description of the immediately preceding dialogue will provide sufficient information for your software supplier to fix the problem.

Appendix B - Program Checks

This appendix describes the different types of program checks that can result in a program being terminated with an explanatory message of the form:

```
$91 PGM CHECK number AT location
```

For example:

```
$91 PGM CHECK 3 AT 08E2
```

where the *location* is the Cobol address of the failing instruction. In the fuller diagnostic report which can be obtained using D instruction of \$DEBUG, the *number* is translated into a **mnemonic** which appears at the beginning of the first line. The headings of the descriptions below contain the program check numbers together with their corresponding mnemonics.

3 ILL LOAD

The failing instruction is part of a Cobol statement using an indexed variable where the index is not in the valid range 1 to 32767.

Alternatively, the failing instruction is an Intermediate Code instruction which specified a qualifier whose value is not in the valid range 0 to 32767. (The Intermediate Code is described in the Global Cobol Language Manual.)

- 6 ILL OP
- 7 ILL QUAL
- 8 ILL JUMP

These program checks should not normally occur. If one does, suspect either program corruption or an erroneous transfer of control to an area which does not contain procedure division statements. The failing 'instruction' contained an illegal operation code (ILL OP) or an illegal qualifier value (ILL QUAL) or attempted to branch to a location which was not on an even byte boundary (ILL JUMP).

If the program check address is in a data area, or is outside your program, then the last transfer of control address and the link stack in the diagnostic report should indicate the point within your program at which the erroneous transfer of control took place.

10 NUM CONV

The numeric conversion program check takes place if a MOVE instruction attempts to move a display numeric field to a computational or display numeric field and the first field does **not** contain a valid numeric string. Note that all spaces is not a valid numeric string.

The numeric conversion program check can be avoided by coding an ON OVERFLOW statement after any MOVE statement likely to suffer an overflow condition. The program check location is that of the failing instruction within the MOVE statement.

11 O/FLOW

The overflow program check takes place if any of the overflow conditions (with the exception of DISPLAY overflow) defined in the Global Cobol Language Manual (and not covered under numeric conversion, above) takes place, and an ON OVERFLOW statement has not been coded to process the condition. The program check location identifies the failing instruction. Note that a field set to all ASCII zeros is not a valid display numeric if the field is signed or contains decimal places.

12 MEM VIOL

The memory violation program check occurs when a user program attempts to write to an area containing protected code or data private to Global System Manager. The program check location identifies the statement which tried to write to the protected area. Protected areas include certain read-only system variables and data areas within modules on the system stack.

This program check usually indicates erroneous use of an index or base. It can also arise within the \$MONITOR if an I/O intercept routine, introduced using the ON ERROR statement, erroneously contains either an ENTRY statement or a CALL statement.

13 PARM STK

The parameter stack program check occurs when a routine has been called with the incorrect number of parameters. This is normally due to an incorrectly coded CALL statement, but may also occur if a routine is entered other than by a CALL.

The program check location is within the ENTRY statement of the routine which has been passed the incorrect number of parameters. If the program check location is not within the user program, or is an access method entry point, or if the CALL statement is coded correctly then the check is due to the previous CALL invoking a section name or paragraph name rather than an entry name.

14 LINK STK

The link stack program exception occurs when all available levels of subroutine are in use and a further PERFORM or CALL statement is attempted. The program check location is that of an instruction within the failing PERFORM or CALL statement. Global System Manager V8.1 includes 10 more link stack entries than previous versions.

There are 16 levels of subroutine reserved for user programs, as explained in Appendix C of the Global Cobol Language Manual.

16 NO BASE

The no base program check occurs if you reference a linkage section item which is based on a pointer whose value has not been established. It normally indicates that a BASE statement has been omitted, or that an item in a linkage section group in a subroutine has been referenced before the corresponding ENTRY statement has been executed.

Appendix C - STOP Codes

This appendix describes the different types of stop code that can result in a program being terminated with an explanatory message of the form:

§91 TERMINATED - STOP *code*

The number code in the message is the **stop code**. The meanings of the different codes are described in the rest of this appendix. Stop codes are displayed by Global System Manager and define fatal error conditions which prevent the program continuing.

The following notes are referred to, as appropriate, in the stop code descriptions in the following pages:

Note 1. When a stop code is the result of an error in file handling, the failing file operation and affected file-id can be found from the diagnostic report.

Note 2. When a stop code is the result of an I/O error, if it is due to the operator deciding to abandon an attempted retry, or being unable to make an assignment, or being unable to load a required volume, the dialogue preceding the TERMINATED - STOP message will fully explain the events leading up to termination.

Note 3. Some of the stop codes described here are included for completeness only. They should never occur. If they do, suspect program corruption or try reloading Global System Manager.

Note 4. This stop code will cause a failure routine, set up using the ON FAILURE clause of a DMAM file definition, to be entered if one has been defined. The value of the indicated stop code will be in the accumulator when the failure routine is entered, so that it may perform discriminatory processing on the stop code. If no failure routine has been specified the stop code will appear on the screen as part of a TERMINATED - STOP message.

Note 5. This stop code has occurred during error checking performed by the Speedbase Presentation Manager, and indicates that the database is corrupt. In this event the database should be restored from backup or rebuilt before any further processing is performed.

STOP 101	An attempt has been made to load a program from an invalidly formatted or corrupt program file.
STOP 102	An attempt has been made to load a program from an invalidly formatted or corrupt program library file.
STOP 103	An attempt has been made to load a program into memory above the base of the work space area managed by the FREE\$ system routine. This stop can occur during debugging if a program is loaded while symbols introduced by an N instruction are still active.
STOP 104	A system integrity error has occurred when accessing a program file or program library.

Appendix C - STOP Codes

STOP 105	The 1134-byte index area addressed by system variable \$\$INDE was partially or entirely outside available memory.
STOP 106	A library stub was found, after 3 previous stub references. Only 3 stub chains are allowed in a dispersed library system.
STOP 107	The program was compiled or linked under a later and incompatible version of Global System Manager.
STOP 108	A system integrity error has occurred when accessing a program library.
STOP 201	The program attempted to open a file which was already open. This stop code may also result if a program overwrites an FD. See note 1.
STOP 202	The program attempted to close a file which was not open. This stop code may also result if a program overwrites an FD. See note 1.
STOP 203	You have attempted a CLOSE TRUNCATE or CLOSE DELETE operation on a shared file.
STOP 204	A system integrity error has occurred during a system variable validation operation. See note 3.
STOP 205	A system integrity error has occurred during a system variable validation operation. See note 3.
STOP 206	A program attempted to close a file that had been set up with a file size smaller than that actually used.
STOP 302	The program attempted to open an FD containing a unit-id of "? " (i.e. a question mark followed by two blanks). Such a unit-id is invalid: this message usually means that the program has omitted to obtain file information for the FD from the catalogue or the file information routine, FILE\$, before attempting the open. See note 1.
STOP 402	A severe error has occurred in the printer handling, probably because of a corrupt printer control file (i.e. \$\$Pnnnxx).
STOP 403	A severe error has occurred in the printer handling, probably because of a corrupt printer control file (i.e. \$\$Pnnnxx).
STOP 404	The mount message print line supplied to the printer when printing a file is empty or contains invalid characters.
STOP 405	The version of the Printer Control File is incorrect for this version of Global System Manager. Update the Printer Control File using the current version of \$CUS.
STOP 603	A program using DISPLAY LINE... COL... has been incorrectly linked (a V5.2 or later subroutine library has been used on a V5.1 or earlier system). See note 3.

STOP 605	A program has set the variable \$\$FESC to a value greater than the length of the input field. This stop code may also appear if an ACCEPT is attempted when Global System Manager has suffered internal corruption.
STOP 801	A DISPLAY...LINE or ACCEPT...LINE statement has been attempted but the program has not previously established formatted mode by using the CLEAR statement.
STOP 803	The POSI\$ system routine has been called to position the cursor, but the screen does not support standard cursor control. Note that POSI\$ can only be used on screens for which \$\$TYPE > 0.
STOP 804	A DISPLAY...LINE or ACCEPT...LINE statement has been attempted, but either the line number or column number was negative or zero.
STOP 805	The POSI\$ system routine has been called to position the cursor, but it specified x,y co-ordinates which were not within the screen area, including the base line. (The co-ordinates are counted from the top left hand corner, which is 1,1.)
STOP 1201	A SEARCH or SCAN statement has been executed, but the length of the table entry involved in the operation was wrongly specified as zero.
STOP 1701	Job Management has been terminated. See note 3.
STOP 1720	There is insufficient memory available to run a job.
STOP 173x	An attempt has been made to execute a job which has become corrupt, or which was produced by a corrupt version of \$JOB.
STOP 2101	An incorrect number of parameters has been passed to SVC 21 (case conversion).
STOP 2901	The catalogue file definition passed to the SAVE\$ system routine is invalid. Either the FD is not closed, or its organization is not relative sequential, or no volume-id has been specified.
STOP 2902	The SAVE\$ system routine is unable to continue because there is no free memory available for it to use.
STOP 2904	You have attempted to save files requiring more than 99 backup volumes with a single call of the SAVE\$ system routine.
STOP 2905	When a third parameter is passed to the SAVE\$ routine it must be the six characters "SHARED". This error occurs if any other value is supplied.
STOP 3001	The catalogue file definition passed to the REST\$ system routine is invalid. Either the FD is not closed, or its organization is not relative sequential, or no volume-id has been specified.

Appendix C - STOP Codes

STOP 3002	The REST\$ system routine is unable to continue because there is no free memory available for it to use.
STOP 3401	The length parameter passed to the variable length accept or display routine (ACCE\$ or DISP\$) was not an integer in the valid range 1 to 80 inclusive.
STOP 3402	An internal error has been detected in \$TDUMP. See note 3.
STOP 3403	The \$TDUMPWK work file has become corrupt during a \$TDUMP report printing operation. See note 3.
STOP 3404	An internal data mismatch has been detected during a \$TDUMP tape write operation. See note 3.
STOP 3407	Corrupt data has been found on a tape during a \$TDUMP operation.
STOP 3410	\$TDUMP has detected data which is out of sequence on its tape. See note 3.
STOP 3411	An internal data mismatch has been detected during verification by \$TDUMP. See note 3.
STOP 3412	\$TDUMP initialization logic has found the default tape length to be greater than the maximum tape size specified in the configuration file. See note 3.
STOP 3420	An internal error has arisen in \$TDUMP while writing a tape using the RAM disk interface. See note 3.
STOP 3502	The CUST\$ system routine has encountered an exception condition (e.g. I/O error, not found, or too large) when reading the program file. See note 3.
STOP 3503	Attempt to use CUST\$ on a program which is relocatable or in assembler code.
STOP 3601	The FREE\$ system routine has attempted to release more work space than had previously been allocated.
STOP 3602	The FREE\$ system routine has attempted to set the size of the program area while the work space was still in use. All work space must have been released by FREE\$ before the size of the program area can be set.
STOP 3603	An invalid function code has been passed to the FREE\$ system routine.
STOP 3604	The work space size passed to FREE\$ was negative.
STOP 3710	An internal error has occurred during system initiation. See note 3.
STOP 3801	An internal error has occurred in \$SPOOL. See note 3.

Appendix C - STOP Codes

STOP 4001	An attempt has been made to run a program which uses the V4.2 LOCK statement on a network system. The program must be changed to use the V5.0 (or later) LOCK...FD statement.
STOP 4002	An internal error has been detected during the execution of a LOCK statement. See note 3.
STOP 4301	The compiler has detected end of file in a copy book. See note 3.
STOP 4303	The message file, \$COMWORK, used by the compiler is corrupt. You may need to reinstall Global System Manager.
STOP 4304	The message file, \$COMWORK, used by the compiler is not the correct version. See note 3.
STOP 4305	The program being compiled contains too many symbols. The error should only occur if the program contains more than about 1300 symbols. Consider dividing your program into subroutines.
STOP 4306	The number of symbols defined in the program being compiled has exceeded the space available on the compiler work file. This error should only occur when in excess of 1200 symbols are generated. Consider dividing your program into separately compiled sub-routines, and combining them using the linkage-editor.
STOP 4307	The compiler work file space has been exhausted due to your compiling a particularly large program. Increase the size of the user area if possible. Try to remove any unnecessary symbols from your program, and consider dividing it into subroutines.
STOP 4308	The compilation file generated by the compiler is greater than 100K bytes in length. This should not occur.
STOP 4309	There is insufficient memory available to run the compiler. Approximately 26K bytes are required.
STOP 4320	\$COBOL has detected an internal error. See note 3.
STOP 4398	The Global Cobol compiler component for Global System Manager (Unix) is too small to hold the data area defined for the C compiler on this computer.
STOP 4399	The Global Cobol compiler component for Global System Manager (Unix) is not available on this computer.
STOP 4401	The linker has found an invalid format compilation module. See note 3.
STOP 5201	There is insufficient memory available to allow \$JOB to be run.
STOP 5202	The overlay \$JOBMAN used by \$JOB is corrupt.
STOP 5401	The cross-reference has detected end of file in a copy book. See note 3.

STOP 5403	The message file \$COMWORK, used by the cross-reference is corrupt. You may need to reinstall Global System Manager.
STOP 5404	The message file, \$COMWORK, used by the cross-reference is not the correct version. See note 3.
STOP 5405	The program being cross-referenced contains too many symbols. The error should only occur if the program contains more than about 1300 symbols. Consider dividing your program into subroutines.
STOP 5406	\$XREF has detected an internal error. See note 3.
STOP 5407	The cross-reference work file space has been exhausted due to your compiling a particularly large program. Try to remove any unnecessary symbols from your program, and consider dividing it into subroutines.
STOP 5409	There is insufficient memory available to run the cross-reference. Approximately 20K bytes are required.
STOP 5601	This error indicates that your file converter has not been linked at location #2000, or that the first item in its data division is not the native file interface block. (Having loaded the handler, FCONV checks to see whether the identifier #4E464841, which begins the interface block, is present at location #2000.)
STOP 5602	This error indicates that your file converter signalled an exception condition other than one of those defined in section 4.2.2 of the Global File Converters Manual. A PIC 9(4) COMP field situated at location #500 contains the actual value supplied as the condition in the EXIT WITH condition statement that led to the error.
STOP 5603	An attempt has been made to execute a file converter from a menu or the GSM READY: prompt. File converters can only be executed from the native file utility, FCONV.
STOP 5604	One of the parameters passed to FCONV from your file converter is incorrect (N-BLOCK, N-SIZE, N-LENGTH are validated).
STOP 57xx	A system integrity error has occurred during initiation or product generation.
STOP 572x	The generated product software on the diskette is corrupt, or has been incorrectly serialized by your reseller.
STOP 6003	A program file being relocated using the \$RELOC command is not in the correct program file format.
STOP 6004	A program file being relocated using the \$RELOC command is not in the correct program file format.

STOP 6006	The base and displacement programs supplied to the \$RELOC command do not match.																		
STOP 6007	A system integrity error has occurred in \$RELOC. See note 3.																		
STOP 6101	The internal overlay area used by \$SORT is too small. See note 3.																		
STOP 61xx	An internal error has occurred in the File Converter SVC-61. Refer to the Global File Converters Manual for full details.																		
STOP 6301	An internal error has been detected in \$L. See note 3.																		
STOP 6501	An internal error has occurred in \$CONV. See note 3.																		
STOP 6603	A software error (memory violation) has arisen in the file utility command. See note 3.																		
STOP 6604	The file utility, which processes files without using an access method, has obtained an executive exception which it is unable to process. See note 3.																		
STOP 6606	An internal error has occurred in the file utility, \$F. See note 3.																		
STOP 6610	Attempt to generate from an invalidly formatted master diskette.																		
STOP 6611	Attempt to generate from an invalidly formatted master diskette.																		
STOP 69xy	An invalid pairing of conversion types passed to the Unix C-ISAM field conversion routine been attempted. Where x is the input conversion type (0 if > 100), and y is the output conversion type (0 if > 100). The values of x and y may be: <table data-bbox="502 1317 837 1646"> <tr><td>1</td><td>Computational</td></tr> <tr><td>2</td><td>Display Numeric</td></tr> <tr><td>3</td><td>Character</td></tr> <tr><td>4</td><td>Date</td></tr> <tr><td>5</td><td>Pointer</td></tr> <tr><td>6</td><td>Floating point</td></tr> <tr><td>7</td><td>Decimal</td></tr> <tr><td>8</td><td>Integer</td></tr> <tr><td>9</td><td>Time</td></tr> </table> <p>Refer to the Global File Converters Manual for full details.</p>	1	Computational	2	Display Numeric	3	Character	4	Date	5	Pointer	6	Floating point	7	Decimal	8	Integer	9	Time
1	Computational																		
2	Display Numeric																		
3	Character																		
4	Date																		
5	Pointer																		
6	Floating point																		
7	Decimal																		
8	Integer																		
9	Time																		
STOP 7201	The CONV\$ system routine has been passed an FD that was not of RS or IS organization.																		
STOP 7202	The IS output file FD passed to the CONV\$ system routine did not have a valid key length, and the input file was not IS.																		
STOP 7203	The FD passed to the CONV\$ system routine creating an empty IS file did not contain a valid record length.																		

- STOP 7502 The file list passed to the catalogue routine, CATA\$, as its second parameter is invalid. The list must consist of between 1 and 16 **closed** file definitions defined in contiguous working storage. This error will occur if any additional data declarations are interspersed amongst the file definitions, or if the parameter supplied did not identify a file definition, or if any FD within the list was open, or if there were less FDs present than specified by the third parameter.
- STOP 7503 The file list count passed to the catalogue routine, CATA\$, as its third parameter is invalid. The count must be a PIC 9(4) COMP variable or computational literal between 1 and 16 inclusive. It should indicate the number of FDs in the file list supplied as the second parameter.
- STOP 7602 The file definitions passed to the copy routine, COPY\$, had different organizations. For COPY\$ to succeed the FDs for the input and the output files must have identical ORGANISATION clauses.
- STOP 7701 The FD passed to the file information routine, FILE\$, was not closed. You must only pass to this routine FDs which are closed.
- STOP 7801 The area length passed to the PRIN\$ system routine was not in the range 2 to 255 inclusive.
- STOP 7810 An attempt has been made to preformat '+ format' diskettes on a domain.
- STOP 7811 An attempt has been made to preformat '+ format' diskettes when no track image is provided by Global System Manager for the format.
- STOP 8002 An end of file exception or an invalid key exception has occurred in the relative sequential access method during an operation on a C-ISAM file.
- STOP 8003 The program attempted to read or write a record of a relative sequential file to or from an area which was partially or totally outside the user area. See note 1.
- STOP 8004 An attempt has been made to write a forms control message to a printer using a WRITE statement instead of a WRITE NEXT statement. See note 1.
- STOP 8005 If this error occurs as a result of an OPEN NEW operation it means that the FD passed to the relative sequential access method specified an invalid record length, not in the range 1 to 32767. See note 1.
- If the error occurs when opening an existing file using OPEN OLD or OPEN SHARED it means that the record length saved in the file's label is invalid. An invalid, zero, record length will appear in the label if you fail to close a new file during the run which creates it, or if you use the file utility's ALL instruction to allocate a file with zero-length records. See note 1.

STOP 8006	This error occurs in a multi user system if a program attempts to issue an OPEN SHARED operation on a relative sequential file for which OPTION RESET has been specified, and the file is already opened for another user. See note 1.
STOP 8007	An illegal operation has been attempted on a relative sequential file (e.g. a REWRITE). See note 1.
STOP 8009	The relative sequential access method with C-ISAM files is not supported under this version of Global System Manager.
STOP 8103	The program attempted to read or write a record of an indexed sequential file to or from an area which was partially or totally outside the user area, the storage available to a Global Cobol applications program. See note 1.
STOP 8104	A READ, WRITE or REWRITE operation specifying an invalid key starting with a byte of high values was attempted on an indexed sequential file. See note 1.
STOP 8105	A REWRITE operation attempted to change the key of a record on an indexed sequential file. See note 1.
STOP 8106	An indexed sequential file definition (FD), or the file itself, is corrupt. See note 1. (The ISAM link is zero.)
STOP 8107	An illegal operation has been attempted on an indexed sequential file (e.g. WRITE NEXT or OPEN NEW). See note 1.
STOP 8109	The version of the indexed sequential access method which uses C-ISAM files is not available under this version of Global System Manager.
STOP 8110	An internal error has occurred in the indexed sequential access method whilst opening a file which was not properly closed when it was last updated. See note 3. (Invalid overflow area.)
STOP 8111	The ISAM file is corrupt. A key read from a record in an overflow chain is less than the prime data key. See note 1.
STOP 8135	An attempt has been made to REWRITE a record using the indexed sequential access method on a C-ISAM file and no previous record has been read.
STOP 8150	An error has occurred using the indexed sequential access method on a C-ISAM file when reading a header or trailer record.
STOP 8151	An end of file error or an invalid key error has occurred during an operation using the indexed sequential access method on C-ISAM files.
STOP 8202	A READ, WRITE or CLOSE operation has been attempted on a file which was not open, or an OPEN operation has been attempted on an already open file, using the PSAM access method.

STOP 8203	An attempt has been made to use a physical sector access method READ or WRITE operation to transfer a sector to or from an area which is partially or totally outside the user area, the storage available to a Global Cobol applications program.
STOP 8207	An illegal operation (i.e. one other than OPEN OLD, READ, WRITE or CLOSE) was attempted using the physical sector access method.
STOP 8303	The program attempted to read or write a record of a text file to or from an area which was partially or totally outside the user area, the storage available to a Global Cobol applications program. See note 1.
STOP 8304	An internal error has occurred in the text file access method. See note 3.
STOP 8305	A READ or READ NEXT operation was attempted on a text file, but no terminator was found within the first 256 bytes of the record being accessed. See note 1.
STOP 8306	A READ or READ NEXT was attempted on a text file, but the record length was zero or negative.
STOP 8307	An illegal operation has been attempted on a text file (e.g. a WRITE or REWRITE). See note 1.
STOP 8403	The program attempted to read a record of a variable length record file to or from an area which was partially or totally outside the user area, the storage available to a Global Cobol applications program. See note 1.
STOP 8404	The record length specified in bytes 3 and 4 of the record supplied for a variable length record file READ, READ NEXT or WRITE NEXT operation is not in the valid range, 4 to 32767 inclusive. See note 1.
STOP 8405	The record length specified in bytes 3 and 4 of the record supplied for a variable length record file WRITE operation is not the same as the record length of the record that the WRITE is attempting to replace. See note 1.
STOP 8406	A READ or READ NEXT was attempted on a variable length record file, but the record you tried to retrieve is greater in length than the maximum specified by your FD's RECORD LENGTH statement. See note 1.
STOP 8407	An illegal operation was attempted on a variable length record file (e.g. a REWRITE). See note 1.
STOP 8408	An internal error has occurred in \$LINK while reading a compilation file. See note 3.
STOP 8503	The program attempted to read or write a record of a basic direct file to or from an area which was partially or totally outside the user area, the storage available to a Global Cobol applications program. See note 1.

Appendix C - STOP Codes

STOP 8507	An illegal operation was attempted on a basic direct file (e.g. a REWRITE). See note 1.
STOP 8603	An attempt has been made to read or write a record wholly or partly outside the user area to or from a data library. See note 1.
STOP 8604	An attempt has been made to write an illegal record to a data library. See note 1.
STOP 8605	Invalid record length for data library. See note 1.
STOP 8606	An attempt has been made to open a new data library but the size specified in the SIZE statement in the FD was less than 200 bytes + the record length. See note 1.
STOP 8607	An illegal operation has been attempted on a data library.
STOP 8801	An irrecoverable I/O error has arisen while reading or writing data within a DMAM database file during file processing. The database will need to be restored from backup. See notes 1, 2 and 4.
STOP 8802	An unexpected error condition has arisen whilst processing a DMAM file. See notes 1, 3 and 4.
STOP 8804	An attempt to delete a key from an index has failed because the old key value cannot be found. The index is corrupt. Either rebuild the index or restore the database from backup. See notes 1 and 4.
STOP 8805	A READ PHYSICAL has been attempted on an illegal record address (most likely the address of 0 or -1 left by a start of file or end of file condition).
STOP 8806	An attempt to recover the last operation on a DMAM file has failed because of a key clash with a non-unique key. See notes 1, 3 and 4.
STOP 8807	An illegal operation (e.g. OPEN NEW) was attempted using the Data Management access method. See note 1.
STOP 8808	An attempt has been made to open a C-ISAM format DMAM database or a Btrieve format DMAM database on a version of Global System Manager that does not include the Universal Channel Interface is not available.
STOP 8809	An attempt has been made to open a DMAM database using a read-only version of DMAM, but there was an outstanding write operation noted in the header (due to some earlier process failing). The database must be opened with a full version of DMAM so that the pending operation may be completed, before it may be processed with a read-only version. See note 4.
STOP 8810	An attempt to recover the last operation on a DMAM database has failed because of an illegal key alteration. See notes 1, 3 and 4.

- STOP 8811 An attempt has been made to open an index on a DMAM file, but the space available in the FD to hold the key was too small. See notes 1, 3 and 4.
- STOP 8812 The key of the last record read from a DMAM database does not match the key contained in the record on the file when a subsequent REWRITE takes place. This is most likely due to inadequate security surrounding the updating of records on the database. See note 1.
- STOP 8813 An attempt has been made to REWRITE or DELETE a record on a DMAM database without a successful preceding READ operation. See note 1.
- STOP 8814 An attempt has been made to DELETE a record from a DMAM database without first obtaining a DELETE-LOCK on the record. See note 1.
- STOP 8820 An attempt has been made to process a DMAM database using a conventional file processing statement (READ, WRITE etc.) after it was opened by a call of DTOPN\$ for data take-on. This is not allowed.
- STOP 8821 An index block in a DMAM database has become corrupt. The database should be restored from backup. See notes 1 and 4.
- STOP 8822 An attempt to add a key to an index has resulted in an index split which would cause the index to contain more than 20 levels. Rebuilding the index might cure this problem. See notes 1 and 4.
- STOP 8823 A DMAM database has become full when attempting to allocate a new index block. The database may be in an inconsistent state and must be increased in size and rebuilt before any further processing takes place. The last record written should then be recovered. See notes 1 and 4.
- STOP 8824 A DMAM database has become full when attempting to allocate a new record. The database must be increased in size before any further processing takes place. The last record written will not be recovered. See notes 1 and 4.
- STOP 8825 An invalid deleted chain pointer has been detected during a record or IDB addition on a DMAM database file. The database is corrupt and must either be rebuilt or restored from backup. See note 4.
- STOP 8826 An index split during index rebuilding using IRBLD\$ has cascaded past the root IDB. See note 3.
- STOP 8827 The deleted chain index in the DMAM database has become corrupt. This may mean further corruption of the database. See note 3.
- STOP 8830 A duplicate key has been detected when performing an index rebuild on a unique index. Either duplicate records have been added using data take-on procedures, or the database has become corrupt. Before any further processing may be performed on the database it will be necessary to redefine the offending index as non-unique and to repeat the index rebuild process, or to restore from backup. See note 4.

STOP 8831	The record information was not available in the DMAM file when recovering the last DMAM operation. The database should be restored from backup. See note 4.
STOP 8833	The C-ISAM format DMAM database or Btrieve format DMAM database Access Method is unable to produce a unique lock region for this record. Your C-ISAM or Btrieve file for this record set has become too large. Try deleting unwanted records and reorganising the record set.
STOP 8863	The IRBLD\$ routine has been called on a C-ISAM or Btrieve schema file but the C-ISAM or Btrieve version of DMAM has not been linked into the program.
STOP 8902	A READ, WRITE or CLOSE operation has been attempted on a file which was not open, or an OPEN operation has been attempted on an open file using the Direct Volume Access Method.
STOP 8903	An attempt has been made to use the Direct Volume Access Method READ or WRITE operation to transfer data to or from an area which is partially or totally outside the user area, the storage available to a Global Cobol application program.
STOP 8907	An illegal operation has been attempted on a Direct Volume file.
STOP 9101	A disk error has arisen when attempting to exit from a system request and reload the original program data.
STOP 9501	A catastrophic I/O error has occurred on a library being updated using \$LIB or \$DLIB. The library is unusable, and must be completely recreated.
STOP 9502	A system integrity error has been detected by \$LIB. See note 3.
STOP 9504	A compilation library accessed by \$LIB is corrupt, and cannot be processed.
STOP 9506	An internal error has been detected in \$LIB or \$DLIB. See note 3.
STOP 9510	A system integrity error has occurred while accessing a library.
STOP 9511	A system integrity error has occurred while accessing a library.
STOP 9602	An unexpected error condition has arisen when processing a C-ISAM Indexed-Sequential Access Method schema file or the C-ISAM file. See notes 1, 3 and 4.
STOP 9603	The program attempted to read or write a record using C-ISAM Indexed-Sequential Access Method to or from an area which was partially or totally outside the user area. See note 1.

Appendix C - STOP Codes

STOP 9607	An illegal operation (e.g. OPEN NEW) was attempted using C-ISAM Indexed-Sequential Access Method. See note 1.
STOP 9610	An attempt was made to run the C-ISAM Indexed-Sequential Access Method on pre-V8.1 Global System Manager.
STOP 9613	An attempt has been made to REWRITE or DELETE a record using C-ISAM Indexed-Sequential Access Method without a successful preceding READ operation. See note 1.
STOP 9614	The schema file supplied to the C-ISAM Indexed-Sequential Access Method is corrupt. You should restore from back-up.
STOP 9703	The program attempted to read or write a record using the Direct Unix Access Method to or from an area which was partially or totally outside the user area. See note 1.
STOP 9707	An illegal operation (e.g. REWRITE) was attempted using the Direct Unix Access Method.
STOP 9798	The Unix path name passed to the Direct Unix Access Method is not terminated with a byte containing LOW-VALUES.
STOP 9799	An attempt was made to use the Direct Unix Access Method on a non-Unix Global System Manager configuration.
STOP 9803	The program attempted to read or write a record using the Direct DOS Access Method to or from an area which was partially or totally outside the user area, the storage available to a Global Cobol applications program. See note 1.
STOP 9807	An illegal operation (e.g. REWRITE) was attempted using the Direct DOS Access Method.
STOP 9898	The DOS path name supplied to the Direct DOS Access Method is not terminated with a byte of LOW-VALUES.
STOP 9899	An attempt was made to use the Direct DOS Access Method on a non-DOS Global System Manager configuration.
STOP 9903	The program attempted to read a record using the Speedbase access method from an area which was partially or totally outside the user area, the storage available to a Global Cobol applications program. See note 1.
STOP 9907	An illegal operation has been attempted on a Speedbase file using the Speedbase access method (e.g. OPEN NEW). See note 1.
STOP 9910	A fatal error condition has arisen when processing a Speedbase dictionary. The dictionary file is corrupt and the database will need to be restored from backup. See notes 1, 2 and 4.

STOP 9911	An attempt has been made to open an index on a Speedbase file using the Speedbase access method but the space available in the FD to hold the key was too small. See notes 1, 3 and 4.
STOP 9912	The key of the record read using the Speedbase access method does not match the key in the index pointer. The record or index data has become corrupt. The database should be restored from backup. See notes 1 and 4.
STOP 9913	The record read during a Speedbase access method operation was marked as deleted. The index or record has become corrupt and the database should be restored from backup. See notes 1 and 4.
STOP 9921	A Speedbase access method operation has failed because an index block in the Speedbase file has become corrupt. See notes 1 and 4.
STOP 10101	An internal error has been detected by the \$CONV command. See note 3.
STOP 10201	Should not occur. \$DEBUG checks to see if the partition it is debugging is a valid. If certain system area fields are not as expected then this stop code results.
STOP 10801	The ASSIG\$ routine has been called to assign a unit-id which is an invalid unit address. This is not permitted.
STOP 11001	The file definition supplied as the parameter to the VOLID\$ system routine was not closed.
STOP 11302	A MAPOUT, MAPCLEAR or MAPIN statement has been executed whilst Global System Manager is in teletype mode. It is essential that formatted mode be established using the CLEAR statement before any map processing statement receives control.
STOP 11303	The map passed by the MD has not been linked into the program.
STOP 11307	This error means that you have attempted to access a map produced under a later release, using the mapping routine belonging to an earlier one. In general, the error can only occur when maps are separately loaded, since if they are linked with the programs that use them the process of linking will itself include the appropriate version of the mapping routine.
STOP 11309	A MAPOUT, MAPCLEAR or MAPIN statement has attempted to access a field from an area not defined by an entry in the area names list.
STOP 11310	This error occurs if the recno field in the map definition is not in the range 0 to <i>n</i> inclusive, where <i>n</i> is the number of records defined when the map was created. For maps without records the recno field must always be zero.
STOP 11311	The map associated with the current MAPIN operation employs a validation routine, and this has established a special default value for a field, but this new value does not conform to the field's picture clause. (Only display numeric and computational fields can be affected.) Examine the FV

(described in section 6.1 of the Global Cobol Screen Presentation Manual) to determine the cause of the error.

- STOP 11312 The formatted area of the current screen (\$\$WIDE characters by \$\$LINE-1 lines) is smaller than the screen area defined when the map used by the current MAPOUT or MAPIN statement was created. It is therefore impossible to display the map on this screen and in consequence this error is signalled.
- STOP 11313 A MAPOUT...PRINT operation was attempted, but the third operand of the statement was not the filename of an open relative-sequential FD with a record size between the screen width and 133 inclusive.
- STOP 11315 A MAPOUT...RECORD, MAPCLEAR...RECORD or MAPIN...RECORD statement was attempted, but the recno field in the associated map definition erroneously contained the value 0. It should contain a positive record number in the range 1 to n , where n is the number of records defined when the map was created.
- STOP 11316 A MAPOUT...FIELD, MAPCLEAR...FIELD or MAPIN...FIELD statement involving a field which is part of a record was attempted, but the recno field in the associated map definition erroneously contained the value 0. It should contain a positive record number in the range 1 to n , where n is the number of records defined when the map was created.
- STOP 11317 The mapping routine has been called recursively and this is not allowed. This occurs if a MAPOUT ... PRINT or MAPIN statement is executed from a validation routine, or from an end of job routine if the program was interrupted in the mapping routine itself. The error will also take place if a validation routine fails to return control properly using the EXIT or EXIT WITH condition statement, as explained in section 6.1.3 of the Global Cobol Screen Presentation Manual. The error may also be signalled during debugging if you attempt to resume the program under test other than at the failing statement.
- STOP 11318 The internal save area within the mapping routine is too small. See note 3.
- STOP 11320 A line sequence error has occurred during a MAPOUT...PRINT operation. This will occur if you use this operation on a map that uses the field sequence option.
- STOP 11321 An irrecoverable I/O error has occurred when \$FORM attempted to open its work file.
- STOP 11322 The \$FORM command has erroneously attempted to access data outside the 30,000 byte extent allocated to its work file. See note 3.
- STOP 11323 An irrecoverable I/O error has occurred when \$FORM attempted to read or write its work file.

Appendix C - STOP Codes

STOP 11326	The current map contains too many items for \$FORM to process. Each separate field text string is an item in this sense, but fields and text within records are counted only once. In the current implementation \$FORM is capable of processing a maximum of 200 items.
STOP 11401	The map passed by the MD in the WINDO\$ or WRES\$ subroutines has not been linked into the program.
STOP 11402	The formatted area of the current screen (\$\$WIDE characters by \$\$LINE - 1 lines) is smaller than the screen area defined when the map used by the WIDO\$ or WRES\$ subroutines was created. It is therefore impossible to display the map on the screen.
STOP 11501 values.	The sort control area specified in a SORT statement contained invalid values.
STOP 11502	A sort has failed because there is insufficient free memory available. The calculation command, \$CALC, can be used to calculate the memory required.
STOP 11503	A sort has failed because the work file was too small. The size of work file required can be calculated using the calculation command, \$CALC.
STOP 11504	A RELEASE statement was executed either during the output phase of a sort, or while no sort was in progress.
STOP 11505	A RETURN statement was executed either during the input phase of a sort, or while no sort was in progress.
STOP 11506	A SORT statement was executed while a sort was already in progress.
STOP 11508	A sort, using a work file, has failed because there was insufficient free memory available, or because it required more than 32766 bytes of free memory.
STOP 11701	The RELX\$ or GETX\$ system routine has encountered an irrecoverable I/O error when opening the swap file.
STOP 11901	The CHAR\$ or ECHO\$ system routine has been used on a screen which is half-duplex, and hence always echoes characters keyed.
STOP 12001	The length parameter supplied to the HX-BI\$ or BI-HX\$ system routine was not a positive integer.
STOP 12201	The OPEN\$ routine has been passed an FD which has already been opened either by OPEN\$ or by an OPEN statement.
STOP 12202	The LIST\$ or CLOSE\$ system routine has been passed an FD which has not previously been opened using OPEN\$.

Appendix C - STOP Codes

STOP 12601	The operand passed to the VIDEO\$ system routine was not in the valid range.
STOP 12701	The length supplied to the PASS\$ system routine was not in the range 1 to 80 inclusive.
STOP 129xx	An internal error has occurred in the RT-11 file converter. See note 3.
STOP 13001 integer.	The range supplied to the RAND\$ system routine was not a positive integer.
STOP 14501	The FD passed to the SECUR\$ system routine was not open.
STOP 14502	The FD passed to the SECUR\$ system routine had been declared as "ORGANISATION UNDEFINED". An organization must be specified.
STOP 14702	A message of more than 80 characters has been passed to the JOB\$ system routine. This error can occur if the terminating plus sign of the message has been omitted.
STOP 14703	A pause of more than 80 characters has been passed to the JOB\$ system routine. This error can occur if the terminating minus sign of the pause has been omitted.
STOP 14704	The characters <NULL> or <CR> passed to the JOB\$ system routine within a response must be immediately preceded by a single colon.
STOP 14705	A response of more than 80 characters has been passed to the JOB\$ system routine. This error can occur if the terminating space character of the response has been omitted.
STOP 14706	The word DIALOG(UE) passed to the JOB\$ system routine must be followed by a space and then the word DIVISION.
STOP 14707	Dialogue packets passed to the JOB\$ system routine have not been preceded by a DIALOGUE DIVISION packet.
STOP 14708	A second DIALOGUE DIVISION packet has been passed to the JOB\$ system routine. Only one is allowed.
STOP 14709	The word ENDJOB passed to the JOB\$ system routine has been followed by further characters. The last dialogue packet passed to JOB\$ must end with the word ENDJOB, which causes the job to be initiated. This error can occur if the terminating space of the packet has been omitted.
STOP 14710	Dialogue passed to the JOB\$ system routine contains a character whose value lies outside the permitted range #20 to #7E inclusive.
STOP 14711	A colon in dialogue passed to the JOB\$ system routine is immediately followed by a space character. The colon introduces a response: if this

response is null or <CR>, then the characters <NULL> or <CR> must follow the colon.

- STOP 14712 The word ENDJOB has been passed to the JOB\$ system routine to initiate a job, but no dialogue has been previously received by JOB\$. At least one response, pause or message must be supplied.
- STOP 14713 Dialogue passed to the JOB\$ system routine contains a zero-length message (i.e. two consecutive plus signs).
- STOP 14714 Dialogue passed to the JOB\$ system routine contains a zero-length pause (i.e. two consecutive minus signs).
- STOP 1473x An internal inconsistency has been detected by the JOB\$ system routine. See note 3.
- STOP 1474x An internal inconsistency has been detected by the JOB\$ system routine. See note 3.
- STOP 14801 The CALC\$ system routine has been passed an FD in which the key length or record length has not been established correctly.
- STOP 15101 An attempt has been made to use the RENA\$ system routine on a file which is assigned to a printer.
- STOP 16201 You have attempted to use a command program which was distributed with a different release of Global System Manager. Command programs can only be used with the release with which they are distributed.
- STOP 16301 The serial number supplied to the FIX\$ routine was negative. Negative serial numbers are not allowed.
- STOP 16701 The file/region specified in the UNLOCK statement was not currently locked.
- STOP 16702 The FD specified in a LOCK statement had not previously been OPEN SHARED.
- STOP 16703 An attempt has been made to lock a region more than 127 times.
- STOP 16901 The QLOAD\$ system routine has found that the currently attached program library is not the one from which the quick index was built.
- STOP 16902 QLOAD\$ cannot load the requested program because to do so would overwrite workspace previously acquired using the FREE\$ routine.
- STOP 16903 The program that QLOAD\$ has been requested to load is corrupt. Alternatively, and more likely, the wrong disk or diskette has been mounted. To optimize performance QLOAD\$ does not check that the correct volume is online. This must be done by the controlling program.

Appendix C - STOP Codes

STOP 19001	A zero string length has been supplied to the AS-RL\$ or RL-AS\$ routine.
STOP 19101	A zero string length has been supplied to the OC-BI\$ or BI-OC\$ routine.
STOP 19201	A non-positive string length has been supplied to the AS-EB\$ or EB-AS\$ routine.
STOP 19301	A non-positive string length has been supplied to the BS-BI\$ or BI-BS\$ routine.
STOP 19401	The stack parameter passed to the LOAD\$ or SDATA\$ routine was not U, S or T.
STOP 19402	The relocatable program file to be loaded by the LOAD\$ routine is corrupt.
STOP 19403	The relocatable program file to be loaded by the LOAD\$ routine is corrupt.
STOP 19404	The relocatable program file to be loaded by the LOAD\$ routine is corrupt.
STOP 19405	The address form of a relocatable assembler program requested using LOAD\$ is not the same as the address form of Global System Manager, implying that the program would be incapable of executing properly.
STOP 19407	The relocatable program to be loaded by the LOAD\$ routine has been produced using a later, incompatible, version of Global System Manager.
STOP 19410	A non-positive allocation size has been passed to the SDATA\$ routine.
STOP 19411	An attempt has been made to load a program using LOAD\$ with the same name as a data area already allocated by SDATA\$, or vice versa.
STOP 19412	Routine SDATA\$ has been requested to allocate a data area, but an area with the same name but different size has already been allocated.
STOP 19501	The stack parameter passed to the ENTRY\$ or UNLO\$ routine was not U, S or T.
STOP 19701	The line length passed to the TEXT\$ system routine was not in the valid range 2 to 255 inclusive.
STOP 19801	The FD passed to the ISUSE\$ routine was not open.
STOP 19802	The FD passed to the ISUSE\$ routine was not ISAM.
STOP 19901	The menu control area passed to MENU\$ defined less than 4 or more than 16 lines.
STOP 19902	The menu control area passed to MENU\$ is incorrectly terminated (it should end with two full stops or periods i.e. "..").
STOP 20001	An invalid data format area has been passed to the FDAT\$ routine.

Appendix C - STOP Codes

STOP 20101	Invalid parameters have been passed to the BOX\$ routine.
STOP 20102	The BOX\$ routine has been asked to draw a box deeper than 41 lines.
STOP 20201	The WIDE\$ routine has been called to set a screen width wider than 192 characters.
STOP 20301	An invalid partition number has been passed to the RDSCR\$ system routine.
STOP 20401	The SCRN\$ routine has suffered an internal error. See note 3.
STOP 20601	The TYP\$ routine has been called with an invalid partition number (i.e. less than 1 or greater than the number of partitions on the screen).
STOP 21101	The DBSTA\$ routine has been called but the DMAM FD passed to it was not open.
STOP 21102	The DBSTA\$ routine or the DBCLC\$ routine has been called to access a C-ISAM or Btrieve schema file but the C-ISAM or Btrieve version of DMAM has not been linked into the program.
STOP 21201	The DTWRT\$ or DTCLS\$ routine has been called, but the FD passed was previously opened using a conventional OPEN statement, not a call on DTOPN\$. This is not allowed.
STOP 21202	The DTOPN\$, DTWRT\$ or DTCLS\$ routine has been called to access a C-ISAM or Btrieve schema file but the C-ISAM or Btrieve version of DMAM has not been linked into the program.
STOP 21301	An invalid key length (negative or greater than 63) has been specified to the DBRFK\$, DBRLK\$, DBRNK\$, DBREK\$ or DBRPK\$ routine. This is not allowed.
STOP 21302	The DBRFK\$, DBRLK\$, DBRNK\$, DBREK\$ or DBRPK\$ routine has been called on a schema file but the version of DMAM for C-ISAM or Btrieve has not been linked into the application program.
STOP 21401	The VIEW\$ subroutine (which goes on to invoke the \$VIEW system request) has not been passed a valid RSAM FD.
STOP 21601	Attempt to write past the end of a line (i.e. column 133).
STOP 21602	An irrecoverable program error has occurred.
STOP 21603	\$MH cannot be run on V5.0 systems or earlier.
STOP 21609	An empty child menu has been created within the menu file with a link pointing to it from a parent menu. You should use MN to create at least one entry in the child menu.

Appendix C - STOP Codes

STOP 21610	The menu file is corrupt, in particular, the menu number that associates parent and child menus has been corrupted.
STOP 21612	An irrecoverable I/O error has occurred when reading a menu file.
STOP 21613	An attempt has been made to run a change menu file line (type "M") from an application type menu.
STOP 21620	The print file is not closed.
STOP 21622	The print file record length is 0 which is invalid.
STOP 21624	The Common Section MN\$PR was not found at the correct address. It must be at #1000.
STOP 21626	An overflow occurred when calculating the number of lines since a page throw.
STOP 21627	The file length exceeds the size of the menu file.
STOP 21628	Invalid file operation attempted (i.e. not a read or write).
STOP 21701	The DMAM FD passed to the DBRTT\$ routine was not open.
STOP 21702	The FD passed to the DBRTT\$ routine was not a DMAM FD.
STOP 21801	An irrecoverable I/O error has occurred when reading a menu file.
STOP 21902	An irrecoverable I/O error has occurred when reading a menu file.
STOP 22401	A DMAM file was not open has been passed to the DBKES\$ subroutine.
STOP 22601	An invalid partition number has been passed to the DBUF\$ routine as its only parameter.
STOP 22701	An invalid function number has been passed to the GROUP\$ subroutine.
STOP 25301	A Speedbase I/O operation has been performed on the I/O channel of a window's target record type during execution of the window.
STOP 25302	An unsupported Speedbase I/O operation has been attempted by the window processor. This error indicates a system software malfunction.
STOP 25303	A Speedbase clear operation has been attempted on a window that has not yet been displayed.
STOP 25304	A Speedbase DISPLAY WINDOW . . TEXT statement has been executed when the text of the window was already displayed.

- STOP 25305 An attempt has been made to display a Speedbase window using the DISPLAY WINDOW statement while the I/O channel of the window's target record type did not contain a record.
- STOP 25306 A recursive call has been made on a Speedbase window that is executing. For example, this error will occur if a window is cleared from the routines section while still executing as a result of a prior entry to the window.
- STOP 25307 The Speedbase index key length coded within a window does not match the key-length of the target record type. This error should not occur when processing records stored on the database.
- STOP 25308 A Speedbase CLEAR statement has been performed while windows are executing. This error can also occur if a dependent frame without the NOCLEAR option has been EXECed while windows are still active in the controlling frame.
- STOP 25501 The vector name supplied as the first parameter of the VEC\$ routine was not an upper or lower case letter.
- Alternatively, a Speedbase Database not open or there was a Speedbase generation number mismatch. This condition arises when a database required by the loaded program is not open, or the open database has a generation number different from that expected by the program. This error occurs if old, superseded versions of programs are run.
- STOP 25502 The dimension supplied as the third parameter of the VEC\$ routine was not a positive integer.
- Alternatively, a Speedbase Rewrite or Delete was attempted without a current record. An application program has attempted to REWRITE or DELETE a record which was not locked at the time the instruction was executed.
- STOP 25503 Part or all of the vector defined by the second, third and optional fourth parameters supplied to the VEC\$ routine was outside the calling program's user area. Assuming the start address defined by the second parameter is within the program then the dimension or displacement must be too large.
- Alternatively, there was a Speedbase I/O area outside current partition. A database access verb has specified a data record area which is not within the current program partition. Suspect program file corruption or a compiler malfunction.
- STOP 25504 The list of variables supplied as the only parameter of the RESET\$ routine is invalid. It should consist of upper or lower case letters terminated by a period (full stop).
- Alternatively, there is a Speedbase Primary index key modification error. A REWRITE instruction has attempted to modify the target record's primary index key. This is illegal.

- STOP 25505 Data record not locked. This stop code indicates an internal error within the Speedbase DBMS. The M\$DW I/O routine has detected a data re-write without the presence of a record level lock.
- STOP 25506 The Speedbase Master record is not locked. A WRITE or REWRITE instruction has taken place in which master records to be linked to the target record were not locked. The DBMS has recovered from this error and completed the I/O request correctly before terminating the program with this code.
- STOP 25507 A Speedbase GET key is negative. The RRN specified in the GET verb's KEY clause was negative. This is invalid. This error can also arise if a GET statement is coded without KEY clause. This occurs when no I/O has so far taken place, or the last retrieval returned an end-of-file or start-of-file condition.
- STOP 25513 Illegal line or col. The line or column number coded as a variable in a Speedbase DISPLAY or ACCEPT... LINE statement is invalid. The number was either not positive or else exceeded the boundaries of the screen.
- STOP 25514 The Speedbase PF structure exceeds 16 levels. An attempt has been made to execute a PF construct with more than 16 levels of headers or trailers using the PRINT statement. The PF statement may specify a PF as a header or trailer, and this PF may itself specify a further PF as a header or trailer. However, this is limited to a maximum of 16 levels of PFs.
- STOP 25515 An internal error has been detected within the Speedbase database rebuild utility, and it is likely that this is caused by an programming problem within this program. This problem could alternatively be caused by extremely serious corruption within the database or database dictionary files.
- STOP 25530 An attempt has been made to run a Speedbase frame directly, rather than using the \$BA command.
- STOP 25539 An attempt has been made to run a Speedbase frame without the Speedbase system module being loaded.
- STOP 25540 An unsupported ACCEPT or DISPLAY statement call has been made by a Global Cobol module linked into a Speedbase frame.
- STOP 25541 An indexing error has occurred within a copy-library specified during a compilation using \$SDL. The copy-library is probably corrupt.
- STOP 25543 A frame address mismatch has occurred within Speedbase. An internal Speedbase system error has occurred. You should write down the details from the screen and note in detail what you were doing just before the stop code occurred. This information should be forwarded to your software supplier for analysis.

- STOP 25550 An internal Speedbase system error has occurred. You should write down the details from the screen and note in detail what you were doing just before the stop code occurred. This information should be forwarded to your software supplier for analysis.
- STOP 25551 Speedquery was unable to find the database dictionary (or it was in use) on the same volume as the database itself. Make sure that the dictionary is available to Speedquery.
- STOP 25552 An irrecoverable I/O error occurred while reading the Speedbase database dictionary. This probably means that the dictionary has become corrupted and should be restored from backup. Care should be taken to ensure that the correct version is recovered.
- STOP 25553 Speedquery has exhausted the available memory during the execution of a critical part of the query and was unable to recover. You should simplify the query or increase the user partition size to obtain more memory.
- STOP 25554 An irrecoverable I/O error has occurred while trying to read or write the Speedquery work file. This could be caused by a bad track on the \$P subvolume. You should verify the volume before continuing.
- STOP 25555 Speedquery was unable to create a work file large enough for the current query. You should allocate more file space on the current \$P volume before restarting the query.
- STOP 25556 Speedquery has detected a corrupt database index. You should perform a rebuild or a restore to recover the index file.
- STOP 25557 Speedquery has detected that the database data files are corrupt. You should perform a rebuild or a restore to recover the database.
- STOP 25558 Speedquery has detected that the saved query file requested is corrupt. You should perform a restore to recover the query from backup.
- STOP 25559 Speedquery was unable to find the saved query requested.
- STOP 25560 Incompatible saved query. Speedquery has detected that the database dictionary has been modified since the query was saved. You must re-develop the query.
- STOP 25561 Speedquery was unable to open the print file. This could be caused by a system error, no room on the spool unit, or no room available on the \$PR volume. Please check before continuing.
- STOP 25562 An irrecoverable I/O error in Speedbase occurred while trying to write to the print file. This could be caused by a system corruption, lack of space on the spool unit, lack of space on the allocated \$PR subvolume, or a hardware error.

- STOP 25563 The DX file created to hold the Speedbase database structure is too small. This is caused by an excessively complex database. Contact your software supplier with the details.
- STOP 25564 A bad header was found in the Speedbase dictionary. This may mean that the dictionary has somehow become corrupted and will have to be restored from backup. Care should be taken to ensure that the correct version is recovered.
- STOP 25565 Speedquery was unable to read the dictionary. This may mean that the dictionary has somehow become corrupted and will have to be restored from backup. Care should be taken to ensure that the correct version is recovered. This error may also be caused by a bad disk track, in which case you should verify the dictionary.
- STOP 25566 An irrecoverable I/O error occurred while opening or closing the DX file. This could be caused by a system corruption, a corrupt DX file, or a bad disk track. Verify the subvolume, then delete the existing DX file. When restarted, Speedquery automatically recreates the DX file.
- STOP 25567 An irrecoverable I/O error occurred while reading or writing the DX file. This could be caused by a system corruption, a corrupt structure file, or a bad disk track. You should verify the subvolume then delete the existing DX file. When restarted Speedquery will recreate the DX file.
- STOP 25568 Insufficient room for the DX file on the database subvolume. Speedquery needs 4.5 Kbytes of spare space to create the DX file. You should allocate the space then restart Speedquery.
- STOP 25569 An I/O error occurred while trying to read the list of available files on the \$P subvolume using Speedbase. This is usually caused by a corrupt directory. Please check and rectify before continuing.
- STOP 25570 An I/O error occurred while reading or writing the sort work file on the SQW (Speedquery) subvolume. Verify the subvolume before continuing.
- STOP 25571 An internal system error within Speedquery has occurred while trying to perform a sort. You should write down the details from the screen and note in detail what you were doing just before the STOP code occurred. This information should be forwarded to your software supplier for analysis.
- STOP 25580 Speedbase was unable to open the pop-up image swap file required by the last loaded frame. This will occur if there is insufficient free space on logical unit "BAW", or if an I/O error occurs during opening of the swap-file.
- STOP 25581 An I/O error has occurred during swap-file processing. This may be due to a disk error.
- STOP 25582 Using the B\$CDB, B\$XCL or B\$XSH routines, an attempt was made to
access a database which was not open at the time of the call.

- STOP 25583 I/O error in B\$XFAM. The Unix error result code will have been displayed immediately before this stop code. This error should not occur in application programs.
- STOP 25584 Unsupported I/O Request in XA\$FAM. Internal Systems Software error.
- STOP 25586 The next free Speedbase data-record slot was permanently locked during a WRITE operation. See note 5.
- STOP 25587 Find exceeds 16 IDX Levels in the Speedbase database. A random index search has exceeded 16 levels of index. See note 5.
- STOP 25588 Dummy high not found in Speedbase IDB scan. The HIGH-VALUES index terminator was not found during an index scan. See note 5.
- STOP 25589 GVA over/underflow. A Speedbase WRITE, REWRITE or DELETE statement has caused a GVA field to overflow on one of the associated Master records. GVA over-flow will occur if the computational capacity of the field is exceeded. See note 5.
- STOP 25590 Link fails by permanent lock. A Speedbase WRITE or REWRITE instruction failed because one of the master records to be linked was not locked. The DBMS recovery procedure was then unable to recover from this error because the master record to be linked was permanently locked by another partition. See note 5.
- STOP 25591 Link fails by non-existent master. A Speedbase WRITE or REWRITE instruction failed because one of the master records to be linked was not locked. The DBMS recovery procedure was then unable to recover from this error because the master record specified by the master access key did not exist. See note 5.
- STOP 25592 Speedbase index key not found in delete. During the DELETE verb processing, all index key entries referencing the record to be deleted are normally removed. The DELETE verb was unable to find one of these index keys during processing. See note 5.
- STOP 25593 Illegal index key for Speedbase access. A WRITE or REWRITE verb has attempted to create an index entry starting with a high-values byte (#FF). This is illegal. See note 5.
- STOP 25594 Free IDB pool in a Speedbase database is exhausted. A WRITE or REWRITE instruction required a free index block to complete an index key addition. The free index pool was however empty. See note 5.
- STOP 25595 Link to unused or deleted IDB. An error has been detected where an index block is incorrectly linked to a deleted or unused IDB. Your Speedbase database is corrupt and should be restored or rebuilt.
- STOP 25596 Irrecoverable I/O error has taken place during an IDB or data transfer. Your Speedbase database is corrupt and should be restored or rebuilt.

- STOP 25597 Negative Speedbase IDB or data record. The IDB/data transfer routine within the DBMS has detected a request for a negative record number. Internal error. See note 5.
- STOP 25598 Speedbase file condition. A file condition has arisen during a data transfer to or from disk, such as an attempt to transfer data to or from an area outside the database file extents. See note 5.
- STOP 25599 Speedbase IDB and record key mismatch. An index entry has been read which does not match the index key held on the record as stored on the database. See note 5.

Appendix D - EXIT Codes

This appendix describes the different types of exception conditions which can result in a program being terminated with the message:

`§91 TERMINATED - EXIT code`

The number *code* in the message is the **exit code**. The meanings of the different codes are described in the rest of this appendix.

The program has been terminated because it failed to trap the exception with an ON EXCEPTION or ON OVERFLOW **statement immediately** following the statement responsible for generating the condition. The program check location in the diagnostic report, obtained by keying D to the debug prompt, identifies where the ON statement should have been coded to handle the situation. The last two digits of the exit code are the same as the exception condition number that the component signalling the exception returned in system variable \$\$COND.

The following notes are referred to, as appropriate, in the subsequent exit code descriptions:

Note 1. When a file processing statement signals an exception which is not trapped, the type of file processing statement responsible, together with the affected file-id, can be found from the diagnostic report. For a description of the exceptions that file processing statements can suffer see the Global Development File Management Manual.

Note 2. When an exception has resulted because of an I/O error, either OPTION ERROR must have been specified in the affected FD, or the error must have been detected by a system routine, otherwise the user program would have been terminated. The dialogue preceding the program check message will fully explain the events leading up to termination.

Note 3. Some of the exit codes described here are included for completeness only. They should never occur. If they do, suspect program corruption.

Note 4. When an exception has resulted because of a problem in loading a program, the program-id in the second line of the diagnostic report identifies the program in question. It is impossible to trap exceptions signalled by the EXEC statement but those detected by LOAD can be handled in the normal way as explained in the Global Cobol Language Manual.

EXIT 101	An I/O error occurred when attempting to LOAD or EXEC a program. See notes 2 and 4.
EXIT 102	A LOAD or EXEC statement required a program or library which could not be found. (You replied <CR> to the program required prompt.) See note 4.
EXIT 103	A LOAD or EXEC statement required a program whose program file, although present, was too large for the available memory. See note 4.
EXIT 201	The operator has replied N to the mount prompt to signify that the requested volume is not available. The user program has failed to trap the I/O error exception signalled by the affected OPEN statement. See notes 1 and 2.

Appendix D - EXIT Codes

EXIT 202	Either an OPEN NEW statement was attempted and the file already exists, or an OPEN OLD or OPEN SHARED statement failed to find a file with the file-id and organisation specified in its FD. In any case the user program failed to trap the resulting exception. See notes 1 and 2.
EXIT 401	An irrecoverable I/O error has occurred which the program has failed to trap. See note 1.
EXIT 402	An executive exception of 2 has occurred from a basic I/O interface. See note 1.
EXIT 601	A program using the ACCE\$ system routine has failed to trap the exception which indicates that the operator has keyed the null string (i.e. <CR> by itself). This exception may also occur in an ACCEPT...NULL statement due to program corruption.
EXIT 602	An ACCEPT, ACCEPT LINE... COL... or CALL ACCE\$ operation has timed out (due to use of \$\$ATIM) and the program has failed to trap the resulting exception.
EXIT 801	A CLEAR statement or a call on the CLEAR\$ system routine has suffered an inappropriate terminal type exception which the user program has failed to trap.
EXIT 803	A program using the DISPLAY...LINE statement to output a computational variable has suffered an overflow condition, but failed to trap it with an ON OVERFLOW statement coded immediately following.
EXIT 1001	An irrecoverable I/O error has occurred which the program has failed to trap. See note 2.
EXIT 1002	A program has failed to trap a file condition. See note 1.
EXIT 1003	An I/O operation has attempted to access memory outside the user area. See note 3.
EXIT 1004	An I/O operation has failed because the relevant executive is busy. The operation should be retried.
EXIT 1201	A SEARCH or SCAN statement has suffered a table operation exception which the user program has failed to trap.
EXIT 1501	A Global System Manager routine converting a hexadecimal string to a pointer value has failed to trap the exception which indicates that the format of the string is invalid. See note 3.
EXIT 1601	A DISPLAY statement attempting to output a computational variable suffered an overflow condition which the user program has failed to trap.

EXIT 2401	An attempt has been made to input a character without echoing it on a half-duplex screen, and Global System Manager has failed to trap the resulting exception. See note 3.
EXIT 29xx	A program using SAVE\$ has failed to trap an exception signalled by the routine.
EXIT 30xx	A program using REST\$ has failed to trap an exception signalled by the routine.
EXIT 3301	A program using the LIBR\$ system routine has failed to trap the exception which results when an irrecoverable I/O error occurs accessing the text library.
EXIT 3302	A program using the LIBR\$ system routine has failed to trap the exception which results when the file specified in the FD supplied to LIBR\$ either cannot be found, or is not a text file.
EXIT 3303	A program using the LIBR\$ system routine has failed to trap the exception which results when a library format error is detected. A preceding screen message will give further information.
EXIT 3601	A program using the FREE\$ system routine to acquire dynamic storage has failed to trap the exception which signals that there is insufficient memory available to honour its request.
EXIT 4001	A LOCK file statement had detected that the file is already locked, but the program has failed to trap the resulting exception.
EXIT 6801	DEVIN\$ has signalled an exception for a unit or format not supported on this computer and the exception has not been trapped by the program.
EXIT 7102	A program using the DELE\$ routine has failed to trap the exception signalled when the file to be deleted was not found.
EXIT 7201	The file conversion routine, CONV\$, has signalled an irrecoverable I/O error on its input or output file but the user program has failed to trap the resulting exception. See note 2.
EXIT 7202	The file conversion routine, CONV\$, has signalled that the input file cannot be found; that the keys of a relative sequential file used to create an indexed sequential file were out of sequence; or that output file capacity was exceeded. The user program, however, has failed to trap the resulting exception.
EXIT 7501	The catalogue routine, CATA\$, has signalled that an irrecoverable I/O error has occurred when reading the catalogue but the user program has failed to trap the resulting exception. See note 2.

Appendix D - EXIT Codes

- EXIT 7502 The catalogue routine, CATA\$, has signalled that the catalogue file is not present or the file supplied is not a valid catalogue. However, the user program has failed to trap the resulting exception.
- EXIT 7601 The file copy routine, COPY\$, has signalled that an irrecoverable I/O error has occurred on its input or output file but the user program has failed to trap the resulting exception. See note 2.
- EXIT 7602 The file copy routine, COPY\$, has signalled that the input file cannot be found or that the output file capacity is less than that of the input file. However, the user program has failed to trap the resulting exception.
- EXIT 7702 A program using the FILE\$ system routine has failed to trap the exception resulting when the operator replies <CR> to the file-id prompt and there is no default.
- EXIT 7902 The DELE\$ system routine has been unable to find the file it is to delete and the user program has not trapped the resulting exception.
- EXIT 8001 An irrecoverable I/O error has arisen while processing a relative sequential file with OPTION ERROR specified in its FD. The user program has failed to trap the resulting exception. See note 1 and note 2.
- EXIT 8002 A file condition has arisen while processing a relative sequential file but the user program has failed to trap the resulting exception. See note 1.
- EXIT 8101 An irrecoverable I/O error has arisen while processing an indexed sequential file with OPTION ERROR specified in its FD. The user program, however, has failed to trap the resulting exception. See note 1 and note 2.
- EXIT 8102 A file condition has arisen while processing an indexed sequential file but the user program has failed to trap the resulting exception. See note 1.
- EXIT 8103 The link count in the overflow chain has exceeded the limit while processing an indexed sequential file but the user program has failed to trap the resulting exception. See note 1.
- EXIT 8201 An irrecoverable I/O error has arisen while accessing the native volume using the physical sector access method. The FD specified OPTION ERROR but your program (the file converter) failed to trap the resulting exception.
- EXIT 8202 You have attempted to use the physical sector access method on a subvolume of a domain, and failed to trap the resulting exception. The access method can only be used on domains and discrete volumes.
- EXIT 8301 An irrecoverable I/O error has arisen when processing a text file with OPTION ERROR specified in its FD, but the user program has failed to trap the resulting exception. See notes 1 and 2.

Appendix D - EXIT Codes

EXIT 8302	A file condition has arisen when processing a text file but the user program has failed to trap the resulting exception. See note 1.
EXIT 8303	A READ or READ NEXT was attempted on a text file, but the record you tried to retrieve was greater in length than the maximum specified in the FD's RECORD LENGTH statement. This has resulted in an exception being signalled which the user program has failed to trap. See note 1.
EXIT 8401	An irrecoverable I/O error has arisen while processing a basic direct file with OPTION ERROR specified in its FD. The user has failed to trap the resulting exception. See notes 1 and 2.
EXIT 8402	A file condition has arisen when processing a variable length record file but the user program has failed to trap the resulting exception. See note 1.
EXIT 8501	An irrecoverable I/O error has arisen while processing a basic direct file with OPTION ERROR specified in its FD. The user has failed to trap the resulting exception. See notes 1 and 2.
EXIT 8502	A file condition has arisen while processing a basic direct file but the user program has failed to trap the resulting exception. See note 1.
EXIT 8601	An I/O error has occurred on accessing a data library but the user program has failed to trap the resulting exception.
EXIT 8602	A file condition has occurred on accessing a data library but the user program has failed to trap the resulting exception.
EXIT 8801	An internal error has arisen in the Data Management access method during file processing. See notes 1 and 3.
EXIT 8802	A file condition has arisen while processing a DMAM database file but the user program has failed to trap the resulting exception. See note 1.
EXIT 8803	A file condition has arisen while processing a DMAM database file but the user program has failed to trap the resulting exception. This will be a key requires rebuild exception from an OPEN operation, or a lock unavailable exception from a READ. See note 1.
EXIT 8804	An attempt has been made to REWRITE a record on a DMAM database having altered a unique key. This is not permitted, and the user program has failed to trap the resulting exception.
EXIT 8899	An internal error has arisen in the DMAM Access Method during file processing. See notes 1 and 3.
EXIT 8901	An irrecoverable I/O error has arisen while processing using the Direct Volume Access Method with OPTION ERROR specified in the FD. The user has failed to trap the resulting exception. See notes 1 and 2.

Appendix D - EXIT Codes

EXIT 8902	A file condition has arisen while processing using the Direct Volume Access Method, but the user program has failed to trap the resulting exception. See note 1.
EXIT 8903	The block supplied to the Direct Volume Access Method cannot be read.
EXIT 9601	An internal error has arisen in the C-ISAM Indexed-Sequential Access Method during file processing. See notes 1 and 3.
EXIT 9602	A file condition has arisen while using the C-ISAM Indexed-Sequential Access Method but the user program has failed to trap the exception. See note 1.
EXIT 9603	A file condition has arisen while using the C-ISAM Indexed-Sequential Access Method but the user program has failed to trap the resulting exception. This will be a lock unavailable exception from a READ operation. See note 1.
EXIT 9604	An attempt has been made to REWRITE a record using the C-ISAM Indexed-Sequential Access Method having altered a unique key. This is not permitted, and the user program has failed to trap the resulting exception.
EXIT 9701	An irrecoverable I/O error has occurred while processing using the Direct Unix Access Method with OPTION ERROR specified in the FD. The user has failed to trap the resulting exception. See notes 1 and 2.
EXIT 9702	A file condition has arisen while processing using the Direct Unix Access Method, but the user program has failed to trap the resulting exception. See note 1.
EXIT 9801	An irrecoverable I/O error has arisen while processing using the Direct DOS Access Method with OPTION ERROR specified in the FD. The user has failed to trap the resulting exception. See notes 1 and 2.
EXIT 9802	A file condition has arisen while processing using the Direct DOS Access Method, but the user program has failed to trap the resulting exception. See note 1.
EXIT 9901	An irrecoverable I/O error has arisen while processing a Speedbase file using the Speedbase access method with OPTION ERROR specified in its FD but the user has failed to trap the resulting exception. See note 1.
EXIT 9902	A file condition has arisen while processing a Speedbase file using the Speedbase access method but the user program has failed to trap the resulting exception. See note 1.
EXIT 9903	A lock unavailable exception has arisen when reading from a Speedbase file using the Speedbase access method.
EXIT 10301	Either the volume description required from a GTDES\$ call is not available or the volume description to be amended using a PTDES\$ call does not

exist. This may be because there is insufficient space in the volume description table (in which case to achieve sufficient space the domain will need to be initialised) or because an I/O error has occurred when trying to access the volume description table.

- EXIT 10601 The DS-DY\$ or DL-DY\$ routine has been passed a string which does not contain a valid format date, and the program has failed to trap the resulting exception.
- EXIT 10801 A program using the ASSIG\$ system routine has failed to trap the exception which signals that the unit address supplied to the routine was invalid (i.e. not an integer between 100 and 999 inclusive, or a LAN address consisting of a computer identification letter followed by two integers (only on network systems) or a question-mark followed by two blanks).
- EXIT 10802 A program using the ASSIG\$ system routine has failed to trap the exception which signals that the assignment tables are full.
- EXIT 10803 A program using the ASSIG\$ system routine has failed to trap the exception which indicates that an attempt has been made to determine or delete the assignment of a unit-id which is not assigned.
- EXIT 10901 A Global System Manager command program using the internal file deletion routine has failed to trap the exception signalling an irrecoverable I/O error. See note 3.
- EXIT 10902 A Global System Manager command program using the internal file deletion routine has failed to trap the exception signalling that the operator has replied N to the delete prompt. See note 3.
- EXIT 11001 A program using the VOLID\$ system routine has failed to trap the exception signalled when an irrecoverable I/O error occurs on trying to read the volume label.
- EXIT 11002 A program using the VOLID\$ system routine has failed to trap the exception signalled when the volume does not support a standard Global System Manager directory.
- EXIT 11101 A program using the DT-DY\$ system routine has failed to trap the exception signalled when the first parameter is not a valid internal format date.
- EXIT 11201 The TIME\$ routine has been called, but the system does not support a timer and the program has failed to trap the resulting exception.
- EXIT 11301 A program using a MAPIN statement has failed to trap the exception which is signalled when no fields are selected for input. In the case of a MAPIN...FIELD statement, this indicates that the field number is invalid. An invalid field number is either one which was not defined when the map was

created, or is one which identifies an output field, rather than an input or update field.

- EXIT 11302 A program using a MAPIN statement when \$\$ESC is set to 2 has failed to trap the exception signalled when the operator keys <ESC>.
- EXIT 11303 A program using a MAPIN statement involving a map employing a validation routine has failed to trap the exception signalled when the routine generates the validation exception (3) to indicate that the MAPIN operation is to be abandoned.
- EXIT 11401 The version of Global System Manager on which the WINDO\$ or WRES\$ subroutine has been run does not support GTSCR\$ or PTSCR\$. This is either because it is too early a version or because there is no screen image kept.
- EXIT 11402 The save area passed as a parameter to the WINDO\$ or WRES\$ subroutine is not large enough to hold the information from the screen. It needs to be at least 3 times the size of the area covered by the MD.
- EXIT 11501 An irrecoverable I/O error has arisen on the sort work file whilst trying to check its capacity in a sort invoked by the SORT statement. See note 2.
- EXIT 11502 A sort invoked by a SORT statement has insufficient capacity to sort the number of records specified. If a work file was supplied, then this is too small; otherwise if there is no work file the memory available to the sort is insufficient.
- EXIT 11503 A sort, invoked by a SORT statement and with a work file supplied, has insufficient memory available to be able to sort the number of records specified.
- EXIT 11511 An I/O error has arisen on the sort work file during a SORT operation.
- EXIT 11512 A sort invoked by a SORT statement has run out of space in the work file during the RELEASE phase.
- EXIT 11513 A sort has failed due to there being insufficient memory.
- EXIT 119xx A non-graphic character has been input by the CHAR\$ system routine, but the program has failed to trap the resulting exception. The last 2 digits of the exception code indicate the type of control character input.
- EXIT 11919 The CHAR\$ system routine has timed out (due to use of \$\$ATIM) and the program has failed to trap the resulting exception.
- EXIT 12001 The string to be converted by the HX-BI\$ system routine contained a non-hexadecimal digit, and the program has failed to trap the resulting exception.

Appendix D - EXIT Codes

EXIT 12202	The LIST\$ routine has signalled end of directory, but the program has failed to trap the exception.
EXIT 12203	The LIST\$ routine has signalled that a file is already open shared, but the program has failed to trap the exception.
EXIT 12204	The LIST\$ routine has signalled that a file is already open exclusively, but the program has failed to trap the exception.
EXIT 12401	A program using a MAPOUT...FIELD or MAPCLEAR...FIELD statement has failed to trap the exception which indicates that the field number is invalid (i.e. was not defined when the associated map was built).
EXIT 12402	A program using MAPOUT...ALL, MAPOUT...OUTPUT, MAPOUT...RECORD or MAPOUT...FIELD has failed to trap the display overflow exception which is signalled when a numeric output or update field contains a value which does not conform to its picture clause, and so can only be displayed as asterisks.
EXIT 12601	The VIDEO\$ routine has been called, but the operation requested is not supported by your screen and the program has failed to trap the resulting exception.
EXIT 12701	The PASS\$ system routine has timed out (due to the use of \$\$ATIM) and the program has failed to trap the resulting exception.
EXIT 12801	The number supplied to the square root routine, SQRT\$, was negative or greater than or equal to 10^{12} , and the program has failed to trap the resulting exception.
EXIT 13001	The field passed to the HMS-T\$ routine did not contain a valid format time, and your program has failed to trap the resulting exception.
EXIT 13101	The CHECK\$ system routine has been invoked on a terminal that does not support type-ahead, and the program has failed to trap the resulting exception.
EXIT 13102	The CHECK\$ system routine has been invoked when the type-ahead buffer was empty, and the program has failed to trap the resulting exception.
EXIT 13103	The CHECK\$ system routine has been invoked with a COUNT parameter under a pre-V6.0 system, and the program has failed to trap the resulting exception.
EXIT 13301	A program using the KEYS\$ system routine has failed to trap the exception signalled when the terminal attribute program could not be accessed either because of an irrecoverable I/O error, or because it was not available. See note 2.
EXIT 13302	A program using the KEYS\$ system routine has failed to trap the exception resulting when the terminal attribute program is not on line.

Appendix D - EXIT Codes

EXIT 13401	A program using the HMS-T\$ system routine has failed to trap the exception resulting when the supplied time is not in the valid format.
EXIT 13501	Overflow has occurred in an EDIT statement but your program has failed to trap the resulting exception.
EXIT 13801	A program using the DS-DT\$ or DL-DT\$ system routines has failed to trap the exception resulting when the supplied date is invalid.
EXIT 14301	A program using the MIDCH\$ routine has failed to trap the exception which indicates that the time is later than 24.00.
EXIT 14601	A program using the NKM-C\$ routine has failed to trap the execution resulting when the supplied size was not in the valid format.
EXIT 14701	A program using the JOB\$ routine has failed to trap the exception indicating that there is insufficient free memory available to generate the job.
EXIT 14801	The file size computed by the CALC\$ system routine is greater than the maximum possible file size (about 2000Mb) and the program has failed to trap the resulting exception.
EXIT 14901	A program using the PROG\$ system routine has failed to trap the exception indicating that an irrecoverable I/O error has occurred.
EXIT 14902	A program using the PROG\$ system routine has failed to trap the exception indicating that the program required was not on-line.
EXIT 15001	A program using the SECS\$, TIME\$ or MIDN\$ system routines has failed to trap the exception resulting when the machine does not support time of day processing.
EXIT 15102	Either the file to be renamed by the RENA\$ system routine does not exist, or there is already a file present with the new name, and the program has failed to trap the resulting exception.
EXIT 15501	The START\$ routine has been called, but there is no foreground available, and the program has failed to trap the resulting exception.
EXIT 15502	The START\$ routine has been called to swap into a non-existent partition and the program has failed to trap the resulting exception.
EXIT 15702	The RENA\$ routine has been unable to find the file to be renamed, and the program has failed to trap the resulting exception.
EXIT 16302	The file to be fixed by the FIX\$ system routine is not a template file, and the program has failed to trap the resulting exception.

Appendix D - EXIT Codes

EXIT 16701	A program using a LOCK statement has failed to trap the exception indicating that the lock could not be obtained because all 16 possible locks were in use.
EXIT 16702	A program using a LOCK statement has failed to trap the exception indicating that the lock could not be obtained because it was already in use by another user.
EXIT 16803	The area provided to QBLD\$ system routine is too small to contain the index.
EXIT 16902	A program using the QLOAD\$ routine has failed to trap the exception which indicates that the requested program-id is not part of the quick index.
EXIT 16903	A program using the QLOAD\$ routine has failed to trap the exception signalled when there is insufficient memory available to load the program. See note 4.
EXIT 17001	An overflow has occurred in the MULTI\$ or DIVI\$ subroutines.
EXIT 19001	Either the ASCII string supplied to the AS-RL\$ routine contains a character that has no RADIX-50 equivalent, or the RADIX-50 string supplied to the RL-AS\$ routine contains a RADIX-50 value greater than 63999 (decimal). In either case your program failed to trap the resulting exception.
EXIT 19101	The octal ASCII string supplied to the OC-BI\$ routine either contains a character which is not ASCII 0-7 or contains a value which is greater than octal 177777. In either case your program failed to trap the resulting exception.
EXIT 19301	The binary ASCII string supplied to the BS-BI\$ routine contains a character which is not ASCII 0 or ASCII 1, but your program failed to trap the resulting exception.
EXIT 19401	A program using the LOAD\$ routine failed to trap the exception signalled when an irrecoverable I/O error occurs. See notes 2 and 4.
EXIT 19402	A program using the LOAD\$ routine failed to trap the exception signalled when the requested relocatable program could not be found on the device assigned to \$P. See note 4.
EXIT 19403	A program using the LOAD\$ or SDATA\$ routine failed to trap the exception signalled when there was insufficient stack space available.
EXIT 19501	The stack entry to be unloaded by the UNLO\$ system routine was not currently loaded by this user, and the program has failed to trap the resulting exception.
EXIT 19601	A program using the TEST\$ routine has failed to trap the exception signalled when the operator has keyed <CTRL G>.

Appendix D - EXIT Codes

EXIT 19902	A call on MENU\$ with \$\$ATIM set up has timed out, and the program has failed to trap the resulting exception.
EXIT 20101	A program has called BOX\$ without a default graphics table, but the version of Global System Manager in use does not contain graphics characters for this screen, and the program has failed to trap the resulting exception.
EXIT 20201	Either the WIDE\$ routine has been called to set up a wide screen on a pre-V6.0 system, or it has been called to enable wide mode on a screen that does not support a physical width change, and the program has failed to trap the resulting exception.
EXIT 20301	RESID\$ has indicated that it is not possible to fix the current partition in memory, and the program has failed to trap the resulting exception.
EXIT 20401	A program has failed to trap the exception returned by SCRNS\$ to indicate that the partition does not currently have direct access to the screen.
EXIT 20501	A program has called MSG\$ on a pre-V6.0 system, and has failed to trap the resulting exception.
EXIT 20601	Either a program has called TYP\$ on a pre-V6.0 system or it has not been possible to pass type-ahead to the required partition, and has failed to trap the resulting exception.
EXIT 20602	A program has called TYP\$ but the type-ahead buffer is shorter than the number of characters paused to TYP\$, and has failed to trap the resulting error.
EXIT 20603	A program has called TYP\$ but there is insufficient space to insert the characters into the type-ahead buffer, and has failed to trap the resulting error.
EXIT 20701	A program has called CMND\$ on a pre-V6.0 system and has failed to trap the resulting exception.
EXIT 20801	An I/O error has occurred on the user file during an access by the OPID\$ or USER\$ subroutines.
EXIT 20802	The USER\$ subroutine has been called for an operator not signed on to the system or the specified user number for an OPID\$ call is out of range for the system.
EXIT 20803	Either the OPID\$ or the USER\$ subroutine has been called on a pre-V5.1 system and the user program has failed to trap the resulting exception. Alternatively, OPID\$ has been called with a computer letter of LOW-VALUES (#00).
EXIT 20901	The D-CID\$ routine has been called with an invalid display, format or computer-id, and the program has failed to trap the resulting error.

EXIT 21001	The DBSET\$ routine has determined that it cannot load the key build routine, either because it is running on a V5.0 system, or because it cannot find a key build routine on SYSRES (routines are named "%a0V63 ", where a is the architecture code for the computer), and the user program has failed to trap the resulting exception.
EXIT 21101	Overflow has occurred during the calculations on database size performed by DBSTA\$ or DBCLC\$, and the user program has failed to trap the resulting exception.
EXIT 21401	An I/O error has occurred when attempting to view a file using VIEW\$ which the calling program has failed to trap.
EXIT 21402	An attempt has been made to run VIEW\$ on a pre-V6.0 system or an attempt has been made to run VIEW\$ on a printer and not on a spool unit.
EXIT 21501	ZAP\$ has suffered a hard error while trying to read or write the file to be zapped.
EXIT 21502	ZAP\$ has not found the file or module to be zapped or it was of the wrong type.
EXIT 21503	ZAP\$ has found the patch to be out of bounds. It either starts outside the extent of the file, or extends too far.
EXIT 21504	ZAP\$ has found a mis-match with the previous file contents.
EXIT 21505	ZAP\$ has found the ZAP to be already applied.
EXIT 21601	Is signalled by \$MH to indicate that a background activity may take place, while awaiting operator input.
EXIT 21602	Is signalled by \$MH if the menu handler detects data corruption in the menu file.
EXIT 21603	Is signalled by \$MH if a serious program error has been detected by the menu handler, which will not be helped by restoring the data.
EXIT 21801	An irrecoverable I/O error has arisen when using MHRW\$ to process the menu file. The same error will be reported if the menu file cannot be found.
EXIT 21802	Is signalled if the menu file is locked by another user when MHRW\$ is called to read and lock a menu.
EXIT 21901	Indicates that on a call to MHPAS\$ the password was not correct (and that a password is required to gain access to the identified function).
EXIT 21902	When a call to MHPAS\$ has been made this exit code indicates that access to the indicated function is denied, and that no password can be

Appendix D - EXIT Codes

keyed which will grant access (the line would be 'starred out' of the menu for this operator).

EXIT 22001	LOG\$ has detected an irrecoverable I/O error on the log file \$\$LOG.
EXIT 22002	There are less than 50 available records available on the log file.
EXIT 22003	LOG\$ has found that the log file \$\$LOG is full.
EXIT 22101	A byte from the BCD string passed to the BCD-X\$ or X-BCD\$ subroutines is not in the range ASCII 0-9.
EXIT 22102	The length passed to the BCD-X\$ or X-BCD\$ subroutines is either zero or odd.
EXIT 22601	The DBUF\$ routine has been run on a pre-V8.0 system and the user program has failed to trap the exception.
EXIT 22602	The DBUF\$ routine has found that there are still characters in the display buffer for the target partition.
EXIT 22701	The GROUP\$ subroutine has detected an irrecoverable I/O error on the \$\$GROUP file or the \$\$GROUP file is not present on the \$M volume.
EXIT 22702	The GROUP\$ routine has found that there are no next or prior groups to return, or that the group to be found does not exist.
EXIT 22703	The group record to be returned by the GROUP\$ subroutine is in use.
EXIT 22704	An attempt has been made to run GROUP\$ on a pre-V8.1 version of Global System Manager.
EXIT 22801	The OPNM\$ routine has not found the operator file, \$\$OPID, or the file is in use. Alternatively, an unexpected I/O error has occurred while accessing the operator file.
EXIT 22802	The OPNM\$ routine has not found the required operator in the operator list file.
EXIT 22803	An attempt has been made to run the OPNM\$ subroutine on a pre-V8.1 version of Global System Manager.
EXIT 22901	The N-ORG\$ subroutine was passed an invalid file-type and the program has failed to trap the resulting exception.
EXIT 23001	The SUBS\$ routine has been called on an unsupported unit.
EXIT 23002	The SUBS\$ routine has been called with the unit number of a domain.
EXIT 23101	The GUI\$ routine has been called on a pre-V8.1 system and the program has failed to trap the exception.

- EXIT 23102 The GUI\$ routine has detected that the screen is not running the GSM Windows option and the program has failed to trap the exception.
- EXIT 23201 The BUFF\$ routine has been run on a pre-V8.1 system and the user program has failed to trap the exception.
- EXIT 25301 A program calling a scientific subroutine (e.g. ROOT\$, SOLVE\$) has failed to trap the exception that the routine signalled on detecting a fatal error (such as an invalid parameter) which prevented it from continuing. The scientific extension of the debug report will contain an error message naming the failing routine and describing the reason for the error. This will be followed by the list of floating point variables with the values set up by the user program when the routine was called.
- Alternatively, Speedbase Presentation Manager has detected a window that was terminated by the <BCK> function.
- EXIT 25302 A program calling a scientific subroutine (e.g. ROOT\$, SOLVE\$) has failed to trap the exception that the routine signalled on detecting a recoverable error (from which a properly coded calling program might be able to continue). The scientific extension of the debug report will contain an error message naming the failing routine and describing the reason for the error. This will be followed by the list of floating point variables with the values set up by the user program when the routine was called.
- Alternatively, Speedbase has detected a window (or series of windows) terminated by the use of the <ABO> function.
- EXIT 253xx An exception condition was signalled by a Window process Routine in order to terminate the window.
- EXIT 2540x A program executing a COMPUTE statement has failed to detect the exception signalled when the scientific program supplied as the first or only parameter was found to be in error. The scientific extension of the debug report will display the failing program, the program error line, and then the list of floating point variables at the point of error. The program error line will either consist of an uparrow pointing at the first invalid character, in the case of a syntax error, or will identify the failing statement and iteration together with the reason for the error. The reasons (which are described in detail in section 9.4 of the Global Development System Subroutines Manual), correspond to the possible exit codes for this type of error:
- | | |
|-------|--|
| 25401 | Syntax error. |
| 25402 | Attempted to divide by 0. |
| 25403 | Attempted non integral power of -ve number, or 0 to -ve power. |
| 25404 | Overflow in indicated operation. |

- 25405 A non-valid argument supplied to indicated function (e.g. SQR -1).
- 25406 Undefined variable referenced.
- 25407 Indexed vector entry outside vector dimensions.
- 25408 Indicated vector entry or user variable contains an improper floating point value.

Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.

EXIT 25410 A program using the FP-SN\$ routine has failed to trap the exception signalled when the floating point number supplied as the routine's first parameter contains an improper (e.g. too large) value which cannot be converted to standard numeric form.

Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.

EXIT 25411 A program using the DFP\$ routine has failed to trap the exception signalled by the routine when it is requested to display an improper (e.g. too large) floating point number.

Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.

EXIT 25412 A program using the IN-FP\$ routine has failed to trap the exception signalled when the 17 byte string supplied as the routine's first parameter does not contain valid floating point input.

Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.

EXIT 25413 A program using the AFP\$ routine has failed to trap the exception signalled when the operator's response is the null string (i.e. <CR> by itself).

Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.

EXIT 25414 A program using the FP-AC\$ routine has failed to trap the exception signalled when the floating point number supplied as the routine's parameter is greater than or equal, in magnitude, to 1011.

Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.

Appendix D - EXIT Codes

EXIT 25415	<p>A program using the AC-FP\$ routine has failed to trap the exception signalled when the computational value supplied in the accumulator when the routine is called is not less than 9999999999.5 in magnitude.</p> <p>Alternatively, a Speedbase accept operation was terminated using a function other than <RET>.</p>
EXIT 25416	<p>A Speedbase accept operation was terminated by a using a function other than <RET>.</p>
EXIT 254xx	<p>A Speedbase accept operation was terminated by using a function other than <RET>.</p>
EXIT 255xx	<p>There has been an internal error within one of the components of Global Cobol SCF (i.e. programs named TD\$, TF\$, TI\$ or TT\$ on the link map). See note 3.</p>
EXIT 25501	<p>A record locked exception was signalled from Speedbase Presentation Manager following a GET, READ, or FETCH statement coded without the NOLOCK clause.</p>
EXIT 25502	<p>A requested record key in Speedbase was not found. The index key specified in a READ or FETCH statement was not found.</p>
EXIT 25503	<p>The Speedbase exception conditions 25501 and 25502 have occurred simultaneously using the READ or FETCH FIRST, NEXT, LAST or PRIOR statements.</p>
EXIT 25504	<p>An end of file or start of file exception has occurred using a Speedbase READ or FETCH FIRST, NEXT, LAST or PRIOR statement. This condition can also occur using the GET statement, when the requested RRN key is beyond the logical end-of-file of the target record type.</p>
EXIT 25505	<p>An attempt has been made to DELETE a Speedbase record with an active (i.e. non-zero) servant record group.</p>
EXIT 25506	<p>The target of a Speedbase WRITE statement contained a primary index key which already existed on the database.</p>
EXIT 25507	<p>A Speedbase WRITE statement could not be completed because no free data records remain in the database for the target record type.</p>
EXIT 25508	<p>The Speedbase RRN key coded for the GET statement specified a deleted record.</p>
EXIT 25511	<p>A Speedbase date conversion from display to computational format using a MOVE statement failed because the display format date was invalid.</p>
EXIT 25513	<p>A Speedbase MOUNT statement could not be fulfilled by the operator.</p>

Appendix D - EXIT Codes

EXIT 25520	The Speedbase system area \$BASYS was signalled as not loaded using the test routine B\$CHK.
EXIT 25521	An attempt was made in Speedbase to open the same database twice using database open routine B\$OPN.
EXIT 25522	The Speedbase database open routine B\$OPN was unable to open the specified database either because it was not present on the specified unit, or the file-type was incorrect.
EXIT 25523	The Speedbase data-file(s) belonging to a database could not be opened by the database open routine B\$OPN. This problem is usually caused by incorrect unit assignments.
EXIT 25524	A Speedbase database could not be opened because it was already exclusively opened by another partition.
EXIT 25525	An I/O error occurred during a Speedbase database open.
EXIT 25526	There is insufficient memory on the user stack to open a Speedbase database.
EXIT 25527	A call on B\$OPN could not complete because the DB index file could not be opened within the Unix directory, and the resulting error was not trapped.
EXIT 25528	A call on B\$OPN could not complete because a C-ISAM channel could not be initialised, and the resulting error was not trapped. This may occur when insufficient memory is available.
EXIT 25529	A duplicate key condition has been detected within a C-ISAM file and the resulting exception was not trapped. The database should be rebuilt.
EXIT 25530	The Speedbase system area \$BASYS could not be loaded prior to executing a frame. This occurs if the program \$BASYS is not found, an I/O error occurs, or if there is insufficient room on the user stack to load this module.
EXIT 25531	Invalid display formatting codes have been passed to the Speedbase qualifier definition routine B\$QLN.
EXIT 25532	The display format codes passed to the Speedbase B\$QLN specify positive or negative value highlighting (<>DC+-) although the target operand's format is un-signed.
EXIT 25542	An attempt has been made to run a frame compiled using the V4.0 Speedbase development system within the V3.0 Speedbase Presentation Manager.
EXIT 25543	An attempt has been made to execute a Speedbase frame which is not at the anticipated depth in an overlay structure. For example, this will happen

when using the SEQUENCE statement to transfer control between frames at different levels in an overlay structure.

- EXIT 25544 An attempt has been made to execute a dependent Speedbase frame which was not compiled with the currently loaded controlling frame.
- EXIT 25546 The Speedbase system area \$BASYS could not be loaded prior to executing a frame. This occurs if the program \$BASYS is not found, an I/O error occurs, or if there is insufficient room on the user stack to load this module.
- EXIT 25547 A Speedbase frame load has failed because the required program was not found, or was too large to fit into the available memory. This exit condition will also result if an I/O error occurs during loading.
- EXIT 25548 The field name coded in a call on the B\$WRJ right justification routine was not found in the coded window-id.
- EXIT 25549 A call on B\$XCL failed because the routine could not provide exclusive access, and the resulting exception was not trapped.
- EXIT 25550 A call on B\$STA or B\$ST2 failed because the required database was not open, and the resulting exception was not trapped.
- EXIT 25551 XA\$FAM exception condition caused by an unusual Unix file processing condition. \$\$RES holds corresponding result code. This error should not occur in application programs.

Appendix E - Special Global System Manager Files

This appendix lists the various files that may appear on the SYSRES volume after Global System Manager has been installed:

Name	Type	Description
\$AUWK\$	RS	Temporary authorization file created by \$AUTH. If this appears spuriously on the SYSRES volume of the master computer on a network configuration, it should be deleted before attempting to run \$AUTH.
\$\$AUTH	RS	The authorization file. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$CDES	RS	Table of computer descriptions. This file is created and amended during the \$STATUS DES command. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$DEBUG	DL	The debugger template data library, created and amended by \$DEBUG. (Refer to the Global Cobol User Manual for more details).
\$\$DOM <u>uuu</u>	RS	Domain layout file (<u>uuu</u> is a domain unit number). This file is created and used by \$VOLSAV.
\$\$DR <u>oooo</u>	SY	The \$DIRP personal menu selection file (where <u>oooo</u> is the operator-id). This file is created using \$DIRCUS for subsequent use by \$DIRP.
\$\$FK	SY	Function key definition file. This file is created by the "General Save" option of <SYSREQ> K. Please note that prior to V8.1, this file was named \$FK. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$FK <u>oooo</u>	SY	Function key definition file (where <u>oooo</u> is the operator-id). This file is created by the "Personal Save" option of <SYSREQ> K. Please note that prior to V8.1, this file was named \$FK <u>oooo</u> . This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$GROUP	DM	The group customisation file. This file holds information about operator groups and is maintained using \$GROUP. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$MENUS	VL	The initial system menu file created during installation and amended by menu maintenance.
\$\$MPARAM	VL	The menu display parameter file, created during installation and amended by menu maintenance.
\$\$OPID	SY	The operator name file. This file holds operator-ids and operator long names and is created and maintained using \$OPID. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$P <u>pppxx</u>	SY	Printer control file (<u>ppp</u> is the printer number, <u>xx</u> is the optional computer node-id for network systems). Please note that prior to V8.1, these files were named \$RP <u>pppxx</u> . See section 6.2.5 of the Global System Manager Manual for full details of the Printer Control File naming convention.
\$\$RS <u>ttt</u>	SY	The screen reset sequence file (where <u>ttt</u> is the TAP number). This file is created and maintained using <SYSREQ> S. This file should

		only appear on the SYSRES volume of the master computer on a network configuration.
\$\$SCoooo	SY	The screen sequence file (where oooo is the operator-id). This file is created and maintained using <SYSREQ> *.
\$\$SWAPxx	SW	Swap file (xx is the optional node-id). The swap file is created on logical unit \$SW.
\$\$TLoooo	SY	Temporary work-file created by <SYSREQ> T. This file should only appear on the SYSRES volume of the master computer on a network configuration when <SYSREQ> T is active.
\$\$TRttt	SY	Table of input key translations (where tttt is the TAP number). This file is created and maintained using <SYSREQ> B. Please note that prior to V8.1, this file was named \$TRttt.
\$\$UREQ	DL	Data library holding the table of End User system requests. This file should only appear on SYSRES on the master computer of a network. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$UREQM	SY	Additional system request menu. This file is maintained using \$UMMAIN. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$\$USER	US	User file which contains details of the network. This file should only appear on the SYSRES volume of the master computer on a network configuration.
\$txxxxxx	SY	\$TAPE catalogue file, created and amended by \$TAPE. (Refer to the Global Tape Utility (\$TAPE) Manual for more details).
P.\$OV\$nn	RS	<p>Temporary file opened during the processing of a system request (<i>nn</i> is the label number of the file in the directory). These files should not be confused with the P.\$OV\$ system request library. Any similarity in name is purely coincidental.</p> <p>The P.\$OV\$<i>nn</i> files will be created on unit \$OV if a permanent assignment for logical unit \$OV has been customised (otherwise the work files are created on unit \$DP).</p>

Appendix F - Running System Commands

For most Global System Manager installations, the powerful and flexible menu facility (see Chapter 7 of the Global Operating Manual) allows all operations to be selected from a menu entry. However, for some tasks it may be necessary to run programs that do not have a corresponding menu entry. This appendix provides a full explanation of the techniques that may be used to run Global System Manager system commands, and other programs, directly when a corresponding menu entry is not available.

F.1 The GSM Pn READY: Prompt

If Global System Manager was installed using the standard installation option (see sections A.2 and A.23 of the Global Operating Manual), the main menu will be displayed once the sign-on procedure has been completed. Provided you have a sufficiently high authorisation code, you can quit the Global System Manager Menu Handler to produce the GSM Pn READY: prompt by replying READY to the menu prompt. For example, in partition 3, this prompt will appear as:

```
GSM P3 READY:
```

To reload the Global System Manager Menu Handler, key \$MH (an example of a system command) to this prompt. For example:

```
GSM P3 READY: $MH
```

Note that for some non-standard installations, if the menu option was not selected, the GSM Pn READY: prompt will appear instead of the menu when Global System Manager is loaded.

F.2 Running System Commands from the GSM Pn READY: Prompt

To run a system command from the GSM Pn READY: prompt, simply key the name of the command. A system command is a program with a filename that starts with the "\$" character. For example, to run the \$A system command:

```
GSM P2 READY: $A
```

The Global System Manager Program Loader searches for the command program in the P.\$CMLB0 command library on the \$CP unit (i.e. SYSRES). If the program file, or a library stub (see section 21.16) with the program name, is not found in the P.\$CMLB0 library, the loader continues to search in the optional P.\$CMLB1 and P.\$CMLB2 command libraries. Note that a library stub may cause the loader to re-assign the \$CP logical unit.

If the search through the 3 libraries is unsuccessful, the loader attempts to load the program file from the \$CP unit directly (i.e. without considering any command libraries). If this search is unsuccessful, the following prompt will be displayed:

```
PROGRAM $xxxxxxx IN P.$CMLB0 REQUIRED ON device desc - uuu:
```

F.3 Running Other Programs from the GSM Pn READY: Prompt

To run a program that is not a system command (i.e. a program with a filename that does not start with either the "\$" or "*" characters) from the GSM Pn READY: prompt, simply key the name of the program. For example, to run the program EXAMPLE:

```
GSM P2 READY:EXAMPLE
```

The Global System Manager Program Loader searches for the command program on the \$P unit. To indicate that a program resides in a library, the program library must be attached by keying its name to the GSM Pn READY: prompt. For example, to attach library P.TEST on the \$P unit:

```
GSM P2 READY:P.TEST
V8.1 TEST PROGRAMS
```

The loader displays the library title to confirm that it is attached. If a program library is attached, the Global System Manager loader will search for the program in the attached library on the \$P unit before searching for the file in the volume directory.

Only one program library can be attached in any partition. An attempt to attach another library will automatically detach the current program library. To detach the program library explicitly, key P. to the GSM Pn READY: prompt. For example:

```
GSM P3 READY:P.
LIBRARY DETACHED
```

F.4 Changing the \$P Assignment Using the <CTRL A> Terminator

If the terminator to the program name or program library is <CTRL A> instead of the normal <CR>, the Global System Manager Loader will allow you to re-assign the \$P unit without recourse to \$A or <SYSREQ> I. For example, to run the program TEST from unit 210:

```
GSM P4 READY:TEST<CTRL A>
PLEASE ASSIGN $P:210
```

This second example, runs the program TEST2 from the P.TEST library on the TST unit:

```
GSM P4 READY:P.TEST<CTRL A>
PLEASE ASSIGN $P:TST
V8.1 TEST PROGRAMS
GSM P4 READY:TEST2
```

F.5 Changing the \$CP Assignment Using the <CTRL A> Terminator

Although the technique of terminating the name of a system command by <CTRL A> can be used to re-assign the \$CP logical unit, it is not recommended. The \$CP assignment should never be changed explicitly.

F.6 The "*" to "\$" Aliasing Convention

Occasionally, usually when loading Global System Manager from the BACRES starter system, it may be necessary to run a system command that is not present in a P.\$CMLBn command library. For example, to abort the Global System Manager installation process and return to the host operating system, it is necessary to run \$BYE from the P.MIN library on BACRES.

To allow system commands to be run from the \$P unit instead of the normal \$CP unit (either directly or within an attached library) the Global System Manager Loader treats programs with a filename that starts with the "*" character specially. The Loader automatically replaces the "*" with the "\$" character.

character by a "\$". Because of this aliasing technique, the name of a program file should never start with the "*" character.

For example, to run the \$BYE system command from the P.MIN library (i.e. the library present on the BACRES volume), use the following dialogue:

```
GSM P1 READY: P.MIN
V8.1 MINIMUM SYSTEM LIBRARY
GSM P1 READY: *BYE
```

F.7 Running Programs from a Menu

The methods used to run programs (both system commands and non-command programs) from a menu depend on the menu display style (i.e. SAA or non-SAA) that is being employed.

F.7.1 Running Programs from a non-SAA Style Menu

Provided your menu has been customised to allow commands to be run (see section 7.2.6.2 of the Global System Manager Manual) and you have sufficient authorisation, the method used to execute a program directly from the menu prompt is identical that used to run it from the GSM Pn READY: prompt. For example, to run \$F from a non-SAA style menu:

```
Please select a function: $F
```

Non-command programs can also be run from the menu prompt, libraries can be attached and the \$P assignment can be changed "on-the-fly". For example:

```
Please select a function: P.TEST<CTRL A>
PLEASE ASSIGN $P: 210
;
; (the library attached message will appear briefly and
; the menu will be redisplayed)
;
Please select a function: TEST
```

Note that entries selected from the menu may change both the \$P assignment and the attached library.

F.7.2 Running Programs from an SAA Style Menu

Running a program directly from an SAA style menu is relatively complicated. Provided your menu has been customised to allow command programs to be run (see section 7.2.6.2 of the Global System Manager Manual) and you have sufficient authorisation, key <F1> to produce the menu Command Box. This small dialogue box is usually equivalent to the GSM Pn READY: prompt, allowing system commands and non system programs to be executed, program libraries to be attached and the \$A unit to be re-assigned. If the SAA-style menu has been customised to forbid the running of command programs, the <F1> key will have no effect.

If command programs are allowed, the SAA-style Menu Handler detects when a "\$" character has been keyed and displays the menu Command Box, automatically supplying the first character (i.e. a "\$") of the command program name. Furthermore, if the mnemonic selection length is 0 (see section 7.4.1.10 of the Global System Manager Manual), keying any non-numeric character will result in the automatic display of the Command Box.

Important note: Use of the Global Windows Workstation modifies the behaviour of the Menu Handler. Please refer to the Global Windows Workstation Notes for further details.

