Global
Operating Manual
Unix
Version 8.1

TABLE OF CONTENTS

Section Description Page Number

APPENDICES

Appendix Description Page Number

1. Introduction

This manual is a user's guide to Global System Manager for Unix. It should be read in conjunction with the Global System Manager Manual and the Global Configuration Notes.

In this manual we describe the implementation of Global System Manager on Unix together with those aspects of Global System Manager that are specific to Unix.

Global System Manager is responsible for running the Global range of application software.

This chapter explains what Global System Manager is, and briefly describes the facilities that it offers.

1.1 What is Global System Manager?

In order to run on a particular computer, a computer program must be able to receive data from, and send it to, the computer's *peripheral devices*: screen and keyboard, disk drives, printers etc.

For example, the program has to interpret what the operator types at the keyboard and display the appropriate characters on the screen. Similarly, it has to be able to 'read' data files on a disk, process the data, and output it to the screen or printer.

People recognised early on in the development of computers that it was neither necessary, nor efficient, for every program, designed to run on a particular computer, to duplicate these 'housekeeping' routines. Reading a file of sales figures or a word processing document are identical operations at the hardware level.

Accordingly, operating systems were developed. These are sophisticated computer programs which are loaded first into the memory of a computer, and which provide an environment for other programs - *applications* which perform actual tasks - to run in. The application programs contain only task-specific routines, and call on the resources of the operating system when they need to perform common, low-level operations, such as reading from a disk.

As the computer industry developed, manufacturers began to produce ranges of computers, each available in a variety of configurations: that is to say, with various combinations of peripherals. A standard model could be bought with varying amounts of memory, one or many screens, different printers and different disk options. Operating systems had to be developed which could handle different configurations of the same computer, or of a range of computers from the same manufacturer.

Global System Manager represents the next stage of development. Standard operating systems are specific to computers from a particular manufacturer or, at best, to different makes of computer built around the same processor type. Because non-Global application programs are designed to run under a particular operating system, they cannot usually be transferred from one computer to another. If you want to move on to another make of computer (because you needed a bigger or better machine) you have to scrap your current applications, usually together with all your accumulated data.

Global System Manager, on the other hand, has been designed from the outset to be portable across a wide range of computers, processor types and operating systems.

This means that a company running Global applications can progress from, say, a single-user microcomputer from one manufacturer to a more powerful multi-user minicomputer from another, or to a mixed network of personal or departmental computers. Data files are easily transferred from the old machine to the new, and because Global applications run identically on different computers, so are staff skills.

Standard operating systems have another disadvantage. Their origins lie either in mainframe computing, which is oriented towards very large organizations with specialist data processing departments, or (in the case of early microprocessor operating systems) in the hobbyist market. In both cases, the consequence is complexity and user-unfriendliness.

Global System Manager, in contrast, is targeted towards small to medium-sized organizations running business applications on a wide range of equipment, but which will not normally have their own specialist data processing staff. Global System Manager can be installed, modified and run by staff with a minimum of training because, although it offers sophisticated facilities, these are presented to the user in a simple and clear fashion. All screen dialogue uses ordinary English terms in preference to jargon, and the more complex operations can be made invisible to those who use the system on a daily basis. Extensive on-screen help facilities are available at every point, and (we hope) the manuals are concise but clear when you need to use them.

1.2 An Overview of Global System Manager

So far, we have described Global System Manager in the most general terms. Let's now examine what Global System Manager does in a little more detail. This can be broken down into four main areas: hardware set-up, controlling operator access, running application programs, and providing system utilities.

1.2.1 Hardware set-up - the bootstrap process

When you switch your computer on, initially it is just an empty machine. There may be programs and data held on the hard disk, and various other peripherals attached, but the computer's memory and processor are empty and thus the machine cannot access them. By performing the bootstrap process (sometimes this involves putting a diskette into one of the drives), you cause the operating system to be loaded into memory. Once the Unix operating system has been loaded you can run Global System Manager (see Chapters 2 and 3).

The Global System Manager start-up process falls into two main stages:

- Firstly, Global System Manager loads those parts of itself, so-called 'resident' routines, which stay in memory as long as the Global System Manager session lasts. These enable Global System Manager to control peripheral devices and load the non-resident elements (such as Global System Manager utilities which the operator may ask to run intermittently) when required.

- Next, Global System Manager allocates some of the computer's memory for use as *buffers* and *memory banks*.

  Buffers are areas of memory used as temporary stores for data being moved from one peripheral to another.

  Memory banks, or *partitions*, are areas of memory into which application programs or utilities are loaded in order to be executed. Depending on how much memory is available,

Global System Manager can allocate up to nine partitions for each screen attached to the computer, thus allowing each operator to run up to nine programs concurrently.

1.2.2 Controlling operator access - the sign-on process

The whole process of loading Global System Manager, although internally very complex, takes only a few seconds on most computers and is wholly invisible to the user. The first sight the user has of Global System Manager is when the main menu is displayed. This menu lists the application programs installed on the computer, and you can select a particular application, such as Global Payroll, by keying the appropriate number.

Under some circumstances (described later in this manual) the first sight the user has of Global System Manager **may** be when asked for confirmation that the date and time it has obtained from Unix are correct. Once confirmed, the date and time information is held internally by Global System Manager. Global System Manager can, in turn, pass the date and time on to application programs and utilities which may need to display them or print them in reports.

Under other circumstances (described later in this manual) Global System Manager **may** ask the user of each screen for an operator-id: This too is held internally by Global System Manager and may be passed on to applications and utilities, but its main importance lies in the fact that it can be used to control access to the computer and hence to your data. You can set up a table of authorized users, with associated passwords and authorization levels. Global System Manager will ensure that only these operators can access programs and data. You can also get Global System Manager to direct specified users to particular menus.

Under other circumstances (also described later in this manual) Global System Manager **may** ask the user for their *terminal type*: A number (between one and four digits) which tells Global System Manager what type of screen and keyboard is being used. Global System Manager can then interpret input from that keyboard, and format any displays to the size of the screen.

Operator details are held in the *User File*. The contents of this file can be inspected using the $STATUS command. It also serves as the basis for message passing between different users.

1.3 Global System Manager on Unix: An Operational Overview

Global System Manager provides Unix with the ability to run the Global range of software and hundreds of industry-specific packages. These applications are designed for serious business use, and utilise a simple yet powerful system of menus, clear English commands and a consistent user interface to ensure maximum productivity.

The main features provided by Global System Manager are described in the following sections.

1.3.1 Password and authorisation

Multi-level password and authorisation checking guard against unauthorised access to the system as a whole, to individual modules or to sensitive functions within modules. Individual operators can be directed to particular functions at startup or to personalised menus of their own.

1.3.2 Security

You can set up a table of authorised users, each with their own operator code and password which are checked whenever they run Global System Manager. Use of individual menu entries

can be restricted to operators with certain authorisation codes. Data files can also be password protected, and can either be private or accessible by all users.

Security copies of data and programs can be created on diskettes or tapes, either manually or under job control.

1.3.3 The Menu Handler

Global System Manager introduces a powerful yet easy to use menu handler to Unix. It can be used as a user friendly and consistent interface for **all** your Unix applications.

System menus are automatically updated as new modules are installed, and can be customised to suit each installation. Menu entries can be protected by passwords, and can include a pre-defined sequence of keystrokes to drive the application program. Operators can be restricted to certain functions, or directed to particular submenus.

1.3.4 Virtual screen facility

Each user can be configured to use up to nine virtual screens from which they can run up to nine concurrent Global applications, with the ability to switch between each screen at will. These virtual screens continue processing even when not currently selected (displaying).

In addition, non-Global applications can be executed concurrently. However, the ability to switch between virtual screens is disabled while a non-Global application is executing. Finally, Global System Manager provides the ability to create a Unix shell at **any** point within a Global application (see section 5.4).

1.3.5 Global utilities

Global System Manager acts as a common resource manager for modules that run under it, providing an extensive set of utilities, windowing and pop-up facilities. Global utilities include a powerful print spooler, message passing between screens, and system customisation programs. Pop-up windows provide on-screen calculator and calendar, data transfer between tasks and a standard help system.

Global applications use a common data file format across the whole range of operating systems upon which they run. This enables data transfer between a whole range of different computers, including data transfer between Unix systems of differing architectures. Global System Manager provides a comprehensive set of file management utilities which allow Global files to be created, modified, inspected, repaired and deleted.

1.3.6 System requests

A powerful set of 'system requests' can be invoked at any time with a minimum of keystrokes. These include:

- On-screen calculator able to take numbers from the screen and feed back results into the application program;

- Calendar - shows a month at a time on the screen and can return a selected date to the application program;

- Help - displays help windows to the current application;

- Print - prints the current screen contents;

- Transfer - transfers data between concurrent tasks.

1.3.7 The Global Cobol interpreter
The Global Cobol interpreter has been implemented on Unix. This allows any programs written in Global Cobol to be run on Unix. Each user executes a copy of the Global Cobol interpreter to run their Global applications.

1.3.8 The Global Spooler
Global System Manager provides a sophisticated yet user friendly spooler which allows printers to be shared by all users, with features such as priority sequencing, multiple copies and line-up patterns for pre-printed forms. The spooler can run as a background task, and can be controlled from any screen.

1.3.9 The Global Mailing System
Global System Manager provides a user friendly electronic mail facility which allows operators to send mail messages to each other and take telephone messages electronically.

1.3.10 A file import and export utility
A batch file import and export utility, FILECONV, is available which converts data, text and Informix C-ISAM files to (and from) the Global data format. This facility allows Global applications to interface to other Unix applications.

1.4 Global System Manager on Unix: A Technical Overview
The implementation of Global System Manager on Unix involves the following:

- Global System Manager is started by running special Unix programs or scripts;

- By default, each user is allocated a unique Global System Manager SYSTEM identifier (usually a number between 0x1b and 0xff). SYSTEM's "A" to "Z" are used to identify different Global System Manager filing systems;

- Global System Manager does not access the hard disk directly, the Global System Manager domain is simulated inside a collection of Unix files. Thus, instead of accessing the sectors of a physical disk, Global System Manager accesses the records of a Unix file. These 'disks-within-a-file' are referred to as *simulated volumes*;

- Because Global System Manager and Unix are both resident within the same computer at the same time, Global System Manager is able to access Unix files directly. The Unix functions are made available to Global System Manager programs through a special calling mechanism. Although a version of the Global File Converter product is available using these facilities, usually it would be necessary to write a program to take full advantage of them. This interface is documented in the Global File Converters Manual.

1.5 Global System Manager (Unix) Version and Variant Numbers
Global System Manager consists of several software layers. Each layer has an independent software version (or variant).

1.5.1 Global System Manager (Unix) GSM Versions

The highest level in the Global System Manager (Unix) software hierarchy consists of the Global System Manager "$ programs" (e.g. $F) and related files (e.g. $MONITOR). The *version* of this software layer is referred to as the "Global System Manager version" (e.g. V7.0, V8.0, V8.1). This version number is displayed, together with the Contract Protection Message, at start-up time and also on the first line of the $S report (see Chapter 5 of the Global System Manager Manual).

1.5.2 Global System Manager (Unix) Executive Versions

The next level in the software hierarchy consists of the executives. For Global System Manager (Unix) configurations the executive library (i.e. +.C0) is empty because the code for the executives is included within the BACNAT software (see section 1.5.4). Thus, the version of the executive library, which is meaningless, is not displayed by $S.

1.5.3 Global System Manager (Unix) Controller Variants

The controllers represent the next level in the software hierarchy. For Global System Manager (Unix) configurations the controller library (i.e. +.C2) is empty because the code for the controllers is included within the BACNAT software (see section 1.5.4). Thus, the variant of the controller library, which is meaningless, is not displayed by $S.

1.5.4 Global System Manager (Unix) BACNAT Variants

The BACNAT software (i.e. "native" software: Unix programs, scripts and text files etc.) represents the lowest level in the software hierarchy. Because both the executives (see section 1.5.2) and controllers (see section 1.5.3) are contained within the BACNAT software, the BACNAT *variant* number is the crucial parameter when describing the Global System Manager software revision.

The BACNAT variant number is displayed when the first user loads Global System Manager. The BACNAT variant number is also displayed by $S (see Chapter 5 of the Global System Manager Manual).

The BACNAT variant number is held within the following Unix text files:

$GLDIR/sys/variant
$GLDIR/sys/version
$GLDIR/sys/machine

2. Installing Global System Manager

This chapter describes how to install Global System Manager and other Global software onto your computer. The purpose of installation is to copy the software you want to use for day-to-day running.

The installation process involves using the distribution media (either tape or diskettes), with which you have been supplied, to initialise the files containing the installed software. The distribution media should then be kept in a safe place so that they can be used to reinstall the software if this should ever become necessary.

THIS CHAPTER (ESPECIALLY SECTION 2.1.8) SHOULD BE READ IN CONJUNCTION WITH APPENDIX A.

Sections 2.1 to 2.6 provide a step-by-step guide to installing Global System Manager (Unix). Section 2.7 describes the various options available to **upgrade** an existing Global System Manager (Unix) installation.

2.1 Installing Global System Manager

You must install Global System Manager and (optionally) Speedbase Presentation Manager before installing any other Global software.

Important note: If you are upgrading an existing Global System Manager (Unix) installation, please go directly to section 2.7.

2.1.1 The Distribution Media

Global System Manager together with Speedbase Presentation Manager are always distributed on high-capacity 1.4Mb 3½" diskettes or a variety of magnetic tape formats (e.g. QIC-120, 4mm DAT). Please consult your Global Configuration Notes for further information. If Global System Manager is distributed on diskette, the diskettes will be labelled as follows:

BACNAT	Unix tar format diskette (see Chapter 6);
BACRES	Global System Manager starter system (1st diskette). The BACRES diskette contains Global System Manager startup data;
BEA	Global System Manager starter system (1st extension diskette);
HAA	Global System Manager starter system (2nd extension diskette);
EPA	Speedbase Presentation Manager installation diskette (only present for Global System Manager PM configurations);
CFA	Global Configurator installation diskette (not required during the installation of Global System Manager). See the Global Configurator Manual for further details.

THE GLOBAL FORMAT DISTRIBUTION DISKETTES ARE UPDATED DURING THE INSTALLATION PROCESS, AND THEREFORE MUST NOT BE WRITE-PROTECTED.

If Global System Manager is distributed on magnetic tape, all the above volumes (e.g. BACRES, BEA etc.) will be included on the tape which will be labelled BACRES.

2.1.2 Installation Pre-requisites

PLEASE READ THIS SECTION CAREFULLY BEFORE YOU COMMENCE THE INSTALLATION OF GLOBAL SYSTEM MANAGER.

Global System Manager can only be installed by superuser (root).

The installation procedure requires that you first set up a new Unix user and group (see section 2.1.3).

You must be fully conversant with the System Administration programs and files required to maintain and tailor your Unix system.

In the case of an installation via diskette, ensure that you are close to the computer as you will be asked to mount and remount the diskettes.

2.1.3 Adding a New Unix User and Group

All Global software runs as the Unix group *global*. This convention allows applications run by different users to access the same data files. Because of this requirement, a new user and group must be added to the current list. Both the user and the group must be called *global*.

BEFORE PROCEEDING WITH THE INSTALLATION OF GLOBAL SYSTEM MANAGER, ADD A USER *global* AND A GROUP *global* TO YOUR UNIX SYSTEM, USING THE APPROPRIATE SYSTEM ADMINISTRATION PROCEDURE FOR YOUR VERSION OF UNIX.

2.1.4 The global Directory

Global System Manager is installed into a directory called *global* which is created during installation. Before proceeding with the installation you must decide where this directory is to reside. The initial installation requires at least 8 Mbytes of space available on that mounted filing system, although in general more space will be required as other Global modules will also be installed on the same filing system.

The Global System Manager data files may be anywhere within the filing system. However, the installation software assumes that the system data will be installed into the *global* directory.

2.1.5 Extracting the Installation Script

Extract the installation script glinstall from the distribution media as described in your Global Configuration Notes. This is normally achieved using the Unix tar command. For example:

```
tar xvf /dev/tape glinstall
```

Once the glinstall script has been extracted from the distribution media, the installation can commence.

2.1.6 Running the Installation Script

The options used when running the installation script depend if you are installing Global System Manager (Unix) for the first time or if you a upgrading an existing Global System

Manager (Unix) installation. If you are upgrading an existing Global System Manager (Unix) installation please go directly to section 2.7.

To install Global System Manager from diskette key the following Unix command line:

```
# ./glinstall -f
```

To install Global System Manager from tape key the following Unix command line:

```
# ./glinstall -t
```

You will be asked for the name of a directory into which Global System Manager is to be installed. The Unix shell variables that must be established for a user before they can run Global System Manager are then displayed. **Make a note of these shell variables so that they may be added to each user's login script.**

Do not be disturbed by the WARNING messages that may appear when loading Global System Manager from the distribution diskettes or tape. They indicate that Global System Manager has re-configured itself dynamically to match your computer set-up more closely. The messages are suppressed once Global System Manager has been installed.

The glinstall script, which includes several other options in addition to the -f and -t options described above, is fully described in section 6.2.

2.1.7 The Serial Number Prompt

When you load the starter system for the first time you may get a prompt of the form:

```
Please key serial number (number and letters):
```

You must key in the serial number which appears in the middle of the label on the BACRES diskette or tape. You may then be prompted for details such as the name of the company that supplied you with your Global System Manager and the address of the site where it will be used. When you have keyed these in, the following prompt will appear:

```
Key line number to amend, or A to accept:
```

Check what you have keyed carefully, and if there are any errors key in the corresponding line number to change that line. When all the lines are correct, key A to continue. Note that if you make a mistake after accepting the changes, you can apply up to 10 further amendments using the $CUS System Maintenance option (see section 6.1.4.9 of the Global System Manager Manual).

2.1.8 The Installation Process

The starter system runs the installation program automatically when you load Global System Manager. This program displays a sequence of explanatory text and prompts to determine precisely how you want Global System Manager installed. If you cannot understand a particular prompt, check its reference in the section of Appendix A which explains what you need to do in more detail. Each prompt is prefixed by a reference to Appendix A. For example, [A.17] refers to section 17 of Appendix A.

2.1.9 The Installation Itself

Once you have replied to the first set of prompts, Global System Manager is installed from the distribution diskettes, or in the case of a tape distribution, from the previously extracted data files. Section 2.2 describes problems that can arise during this process and suggests recovery actions. The installation process creates a data volume called SYSRES.

The installation may create more than just the SYSRES data volume. Optionally, a spool unit (SPOOL), a work unit (SYWORK), a log unit (SYSLOG) and a mail unit (SYSML) may be created during installation.

Once the software has been installed, there are further prompts allowing you to customise Global System Manager. For example, you must select the date format you want to use and the type of printer. Again, these prompts are explained more fully in Appendix A.

It is possible to quit the installation in order to run utility commands (e.g. $S, $U, $F, $V) in order to check the new system **BEFORE** installing onto the hard disk. To quit the installation, reply Q to the following prompt:

```
Key <CR> to continue:
```

To continue with the installation, provided no unit assignments have been altered, key INSGSM to the GSM READY: prompt.

To abort Global System Manager without installing, run the $BYE utility from the P.MIN library. For example:

```
GSM READY:P.MIN
V8.1 MINIMUM SYSTEM LIBRARY
GSM READY:*BYE
```

Note the use of both the "*" to "$" aliasing, necessary to run the $BYE command (typed as *BYE) from the $P unit; and the library index load (of P.MIN) prior to running the command program. See Appendix F of the Global Utilities Manual for further details.

2.1.10 Restoring the pre-V8.1 customisation

As explained in Appendix A (sections A.40 and A.42) when installing Global System Manager V8.1 you are given the option of saving the customisation from an existing SYSRES. If you select this option, a copy of the existing SYSRES will be copied to the BACSAV sub-volume before Global System Manager V8.1 is installed, overwriting the existing SYSRES. Once the installation has completed you are given the option of restoring the existing customisation.

If this option is selected, the following files (if present) are copied from the BACSAV sub-volume to the newly installed SYSRES:

$RP*nnn*	Pre-V8.1 printer control files (renamed to $$P*nnn*)
$$P*nnn*	V8.1 printer control files
$$RS*xxxx*	Screen reset sequence files
$FK*xxxxx*	Pre-V8.1 function key definition files (renamed to $$FK*xxxx*)
$$FK*xxxx*	V8.1 Function key definition files
$TR*xxxxx*	Pre-V8.1 input key translation files (renamed to $$TR*xxxx*)
$$TR*xxxx*	V8.1 input key translation files
$t*xxxxxx*	$TAPE catalogue files
$$DR*xxxx*	$DIRP personal menu selection files

$$SC*xxxx*	Screen specific sequence files
$.*xxxxxx*	Global System Manager TAP's
T>*xxxxxx*	Speedbase Presentation Manager TAP's
$MENU*xxx*	User-defined menu files
$$DOM*xxx*	$VOLSAV domain layout files
$$AUTH	$AUTH authorisation file
$$UREQM	End-user system request menu file
$$OPID	Operator-id file
$$GROUP	Group file
$$MENUS	System menu file
$$MPARAM	Menu parameter file
$$UREQ	End-user system request data library
$$CDES	$STATUS computer description file
$$DEBUG	$DEBUG template file
++*xxxxxx*	Configuration file
$DIAL	Comms Support Pack component
$DIALX	Comms Support Pack component
$FCOMM	Comms Support Pack component
P.QG	Comms Support Pack component
P.BC	Comms Support Pack component
$TAPE	$TAPE component
$TAPED	$TAPE component
P.$TAPE	$TAPE component
TACUS	$TAPE component
$OZ	Global Organiser component
$$$JOB	End-user restore customisation job
S.$$$LST	End-user restore customisation list

It is possible to add extra, site-specific files to the above list. This is achieved by creating a text file, S.$$$LST, containing a simple list of the extra files (one file per line) to be copied back to SYSRES after Global System Manager has been installed. If a S.$$$LST file is required it must be present on the existing SYSRES before the update installation is attempted.

Important note: Each file listed in the S.$$$LST file is simply copied, from BACSAV to the newly installed V8.1 SYSRES, using the $F COP command. If a file, listed in the S.$$$LST text file, is not present on the BACSAV volume, the restoration of the pre-V8.1 customisation will fail leaving the newly installed V8.1 Global System Manager in an unpredictable state. The BACSAV volume is created, during the V8.1 installation, by copying all the files from the existing SYSRES volume. Ensure that only files actually present on the existing SYSRES volume are included in the S.$$$LST text file before starting the installation.

When the components in the above list (and in the optional S.$$$LST file) have been restored to the newly installed SYSRES volume an optional, end-user defined, post-installation customisation job is invoked. If this job, $$$JOB, is required it must be present on the existing SYSRES before the update installation is attempted.

If the restore customisation phase of the installation restores a $$AUTH authorization file it will be automatically upgraded to operate with Global System Manager V8.1 (see section 6.3.13 of the Global System Manager Manual). Similarly, if the restore customisation phase of the installation restores any Printer Control Files they will be automatically upgraded to V8.1 standards.

If the save/restore existing customisations option is selected then it is not possible to apply the new V8.1 customisation options (e.g. $MAIL, $GROUP customisations) to the freshly installed Global System Manager. These customisations must be applied using the "Install Extra Facilities" option of $CUS as explained in section 6.1.4.13 of the Global System Manager Manual.

IMPORTANT NOTE: ALL PRE-V8.1 PRINTER CONTROL FILES MUST BE RUN THROUGH V8.1 $CUS TO UPGRADE THEM TO THE V8.1 STANDARD. IF YOU ATTEMPT TO USE A PRE-V8.1 PRINTER CONTROL FILE UNDER GLOBAL SYSTEM MANAGER V8.1, A STOP CODE 405 WILL OCCUR.

IMPORTANT NOTE: $AUTH V8.1 WILL CONTINUE TO WORK WITH PRE-V8.1 $$AUTH AUTHORISATION FILES BUT WITH REDUCED FUNCTIONALITY. SEE SECTION 6.3.13 OF THE GLOBAL SYSTEM MANAGER MANUAL FOR FULL DETAILS.

2.1.11 Returning to Unix

When customisation is complete exit Global System Manager and return to Unix by keying:

```
Installation of Global System Manager is now complete.
You should now restart Global System Manager as described
in the Global Operating Manual.
Key <CR> then run $BYE to terminate Global System Manager:
GSM READY:$BYE
```

2.1.12 Starting the Installed Global System Manager

When the installation of Global System Manager is complete you are recommended to check the installation as described in sections 2.3 and 2.4, then to install your other Global software modules as described in 2.5. Finally, take backup copies of the installed Global System Manager as described in 2.6.

Note that there is little point taking a backup of Global System Manager before installing other software, as the menu (which is held on the SYSRES disk) will be updated as you install further software.

Before running Global System Manager you must first set up the environment variable GLDIR. This must be assigned to the path of the *global* directory, as displayed during the installation procedure. You must also add the path $GLDIR/bin to your path variable (usually PATH). These obligatory Unix shell variables are fully described in section 8.1.

It is usually convenient to establish the shell environment variables in a script file (see section 8.7).

If you are in the shell csh remember to run the following Unix command to add the Global System Manager programs to your list of current commands:

```
# rehash
```

Run the *gladmin* script, as described in section 6.4, to set up the authorisation parameters required to allow you to run Global System Manager:

```
# gladmin
```

To start Global System Manager run the *global* command as described in section 6.3:

```
      # global
or:   # gl
or:   # GLOBAL
or:   # GL
```

2.2 Errors During Installation

This section describes the error conditions most likely to arise during installation. If you get an error message that is not described here consult Appendix C of this manual and Appendix A of the Global System Manager Manual.

Group "global" not defined in *systemfile*
User "global" not defined in *systemfile*

The group and user *global* cannot be found in the Unix system administration files. Ensure that you have added global as both group and user to your Unix system, as described in your Unix system administrator's guide.

***directoryname* does not exist or is not a directory**

The path specified by you as the directory into which Global System Manager is to be installed is not a valid directory. The Global System Manager installation automatically creates a directory called global in the directory specified by you, so, for example, if you specify **/usr** Global System Manager will install into **/usr/global**.

Error whilst extracting files from *installationdevice*

This message can appear under a variety of conditions:

1. You have not specified the correct flag (-f or -t) when running glinstall.

2. The distribution media is not in the drive.

3. The specified *installationdevice* does not exist or is not the correct device. You must change the name of the installation device by editing the installation script **glinstall** and modifying the value of the FLOPPY or TAPE shell variable appropriately; or use the glinstall -d option as described in section 6.2.1. This will only be necessary if your system has been set up in a non-standard way.

4. There is an error on the distribution media. Any media error will be reported on the system console. The software will have to be regenerated by TIS Software.

glintd: Error 1209 - SYSDATA device not found.

The INSDEV device specified in your Systems file does not specify the correct installation device. For tape distribution the INSDEV device should be A60.dir. For diskette distribution this error may occur if the BACRES diskette is not currently in the specified drive.

glintd: Error 1212 - No configuration information on SYSDATA.

For diskette distribution the BACRES diskette is not currently in the specified drive, or INSDEV is not the installation device or file.

PLEASE MOUNT *name* ON *description - uuu* AND KEY <CR>

If this message is repeated when you key <CR>, despite the correct diskette being mounted, then check that you have put the disk in the drive correctly. Also check the unit number (*uuu*) against your Global Configuration Notes to make sure that you are using the appropriate type of diskette. If everything appears to be correct, treat it as a READ error, as described later on in this section.

$57 *message*
$66 *message*
$99 *message*

These error messages are explained in Appendix B.

*** READ ERROR ON *description - uuu***
*** WRITE ERROR ON *description - uuu***
*** H ERROR ON *description - uuu***

Key <CR> to retry, as these errors sometimes are transient. If it is a diskette which has the error, try taking it out of the drive, reinserting it and trying again. If this does not work then proceed as described below.

If the disk in the indicated drive is BACRES, BEA, HAA or EPA then your computer is unable to read the distribution diskette. The most likely reason is that it has been damaged in some way (e.g. by a fingerprint on the recording surface). Contact your supplier for a replacement. If this also fails then the diskette drive on your computer is probably out of alignment, and needs servicing.

*** HARDWARE PROTECTION ON *description - uuu***

The installation process needs to write to the diskette in the drive described, but cannot do so because it has been 'write protected'. Remove the write protection and key <CR> to continue.

2.3 Checking Your New System

You should now have a correctly installed Global System Manager. Load Global System Manager as described in Chapter 3.

If all is well the main menu should appear. There should be **no** error or warning messages beginning with $57 or $99. If there are consult Appendix B for an explanation and suggested recovery action.

If the installed software includes a Speedbase Presentation Manager Run-Time Licence the following message and prompt will appear:

```
The annual rental fee password is shown on the advice note.

Please key password:
```

To use the installed software a rental fee password, consisting of a single digit followed by 13 letters, **MUST** be supplied.

A menu ending with a selection prompt will now be displayed (or a GSM READY: prompt if you elected not to use a menu). You can run command programs by keying their names to either prompt.

If the correct terminal mapping has not been defined in the Systems file (see section 7.3.4) the following terminal code prompt will be displayed:

```
PLEASE KEY TERMINAL CODE (code):
```

If you are unsure of the terminal code key LIS to list those available.

It is very unlikely that the default terminal-type will be correct for your particular screen: Key LIS to list those terminal types that are available.

For serially-connected screens you should perform further tests to make sure that the option switches on the terminal are set correctly, and that the correct terminal code is being used. (Quite often it is necessary to disable "auto-wrap" and "auto line-feed".) To test the terminal, run command $T and key TEST to its first prompt. This will give you a menu of test options. You should try tests 1 (dimensions), 6 (clear screen and cursor positioning) and 7 (extended control functions).

In the **dimensions screen** test, every character position, except one, of the available display area should be filled with a digit. The only exception is the rightmost character of the bottom line, which should be a colon to indicate that the screen serves as a prompt. If the screen display is not as described, but contains lots of blank lines, this is probably because an erroneous option switch setting is causing the terminal to automatically generate a new line sequence when the rightmost character of each line is displayed. Correct the switch setting, key <CR> to the colon prompt to return to the menu, and select the dimensions test again by keying 1.

The **clear screen and cursor positioning** test should result in a display where the screen is bordered by a continuous frame of digits. The cursor should be located to the right of a colon prompt, which appears in the top left hand corner, just inside the framing digits. If the screen does not appear like this you have probably specified the wrong terminal type during installation.

The **extended control functions** test checks that special keys on the keyboard generate the expected characters. You will be prompted with the names of 17 special keys in turn. Reply to each one by pressing the corresponding key. If the result is not as expected, ERROR will be displayed and the prompt repeated. If you are unable to find the correct key, reply N to go to the next one. If any of the keys are incorrect, but the previous two tests worked, then you have probably used the terminal code for a different version of the keyboard.

When you have completed the tests key <ESCAPE> to exit.

2.4 Checking the Printer

You should now check that your printer is working correctly. If you have installed the Global System Manager spooler, you will first need to reassign the standard printer ($PR) to the real printer ($RP) by running command $A:

```
PLEASE SELECT A FUNCTION:$A
..........
.......... (list of current assignments)
..........
$69 UNIT:$PR ADDRESS:$RP
$69 UNIT:<ESCAPE>
```

Run command $T (screen description) and key P to its prompt to print out a description of your screen on the printer. Check that it is printed correctly, with no missing characters or lines. If you cannot print anything check the printer functions correctly under Unix by running standard Unix commands. For example:

```
# cal > /dev/tty1a
```

If you have more than one printer then use $CUS to set up the characteristics of the other printers as described in section 6.1 of the Global System Manager Manual, and then test these printers as described above.

2.5 Installing Other Global Software ($INSOFT)

You are now ready to start installing other Global software modules such as Global Sales Ledger, Global Writer or Global Cobol. If you have chosen to use a menu (the default option) then select function 1, "Install Global Software:" (otherwise run the command $INSOFT from the GSM READY: prompt).

The installation process for software distributed on diskettes differs from the installation process for software distributed on tape.

2.5.1 Installing Global Software from Diskette

You will be asked to supply the name of the first distribution disk (two letters followed by "A", appearing in the top left corner of the label). You are then asked for the diskette format code (3 to 6 characters which appear in the bottom right corner of the label).

$INSOFT then loads and invokes an installation job called *xx*INS from the distribution diskette. If you have mounted the distribution diskette in the wrong drive (or used the wrong diskette) $INSOFT displays a mount message of the following form:

```
PLEASE MOUNT xxA ON diskette-drive - nnn AND KEY <CR>
```

$INSOFT will also accept the unit number of the diskette drive (e.g. a40) as an alternative to the format code (e.g. O2A).

Each software module has a section in its user manual describing installation. You are advised to check all these sections before starting, to see whether the modules need to be installed in a specific order. In particular, some of the accounting modules require other modules to be installed first.

Normally you would delete the $INSOFT menu entry after the application software as been installed so as to avoid confusion. The $INSOFT command can still be used, of course, by invoking it by name from the GSM READY: prompt or menu.

2.5.2 Installing Global Software from Tape

If the Global applications were distributed with your Global System Manager then the software will be distributed as data volumes in the data file A60.

If the Global applications were distributed separately from the Global System Manager tape then you must extract the application software from the tape and modify your Systems file to make Global System Manager aware of the distribution volumes. The simplest technique is to remove (or rename) your current A60.dir before extracting the data files within A60.dir from the tape. **You must ensure that this data file has read/write permissions for the group *global*.** Your Global Configuration Notes explain how to extract files from the distribution tape.

Subsequently, proceed as for diskettes, supplying the data volume unit instead of diskette unit. Alternatively, if the data volume unit of the domain (e.g. A60) is specified, $INSOFT will search for the appropriate data volume (e.g. EDA for the Global Speedbase Development System) on the domain.

2.5.3 Installing Global Software - Important Note

The installation jobs for some of the more mature Global applications (e.g. Global Finder, Nominal Ledger) require the presence of a local file server SYSTEM. That is, the installation job will fail if a disk unit in the range 200-299 is unavailable (the available disk units are displayed by the $U command - see section 4.9). The default, and recommended, SYSTEM's configuration does not include a SYSTEM that is both a file server and includes USER definitions. In order to install applications such as Global Finder and Nominal Ledger, the SYSTEM's configuration must be **temporarily** changed to include a SYSTEM with both DATA and USER definitions (see section F.2.9). Once the modified configuration has been used for product installation, the original SYSTEM's configuration should be restored. This technique does not require any modification of the distributed Global Configuration File (i.e. there is no need to use Global Configurator). Chapter 7 explains how to modify the Systems file in order to change the SYSTEM's configuration.

This installation problem does not apply to the recent Global 2000 and Global 3000 applications.

2.6 Global System Manager Backup and Reinstallation

Once you have installed Global System Manager and all your Global software, you should take a backup copy of Global System Manager and its customisation. **This precaution is absolutely vital**. If you ever need to reinstall Global System Manager you can avoid repeating the customisation of the main menu and table of authorized users by restoring the backup copy of the customisation.

2.6.1 Backup Copy of Global System Manager

Taking a backup copy of Global System Manager is simply achieved by using a suitable Unix command (e.g. tar or cpio) to make a backup of the directory *global* and all its sub-directories, together with any Global System Manager data files that you have added in other directories. For example:

```
# tar cvf /dev/tape $GLDIR
```

2.6.2 Backup Copy of the Global System Manager Customisation

You can take a copy of the customisation by running $CUS, selecting the "System Maintenance" function and the "Save current customisation" sub-function. You are given the option to save the customisation on diskette or on a subvolume on the hard-disk.

If you select to save the customisation onto diskette, a new pre-formatted (or pre-initialised) diskette called BACSAV is required. The "Save current customisation" sub-function will copy the customisation files to the BACSAV diskette (note that versions of Global System Manager prior to V8.0 automatically initialised a diskette as BACSAV). You should then put this diskette away safely with the BACRES, BEA, HAA (and optional EPA) diskettes. You are recommended to save the current customisation in this way every month, so that the BACSAV disk contains the latest menus and list of users.

If you select to save the customisation on a subvolume on the hard-disk, $CUS will create a new sub-volume called BACSAV (or use an existing one if available) and copy all the files from SYSRES to the BACSAV subvolume.

2.6.3 Re-installing the Global System Manager BACNAT components

To re-install the software distributed on the BACNAT volume, key:

```
# ./glinstall -u
```

This will only update the files in the **global/bin** and **global/sys** directories. The **global/data** files will not be updated. The current **Systems** file (see section 7.1) will be renamed to **Systems.old** before the re-installation takes place. When the re-installation has completed you are given the option of restoring the renamed **Systems** file.

The glinstall command is fully described in section 6.2.

2.6.4 Re-installing the Global System Manager BACRES components

If you need to reinstall Global System Manager because of I/O errors on the SYSRES volume, for example, you should login to Unix as super-user (i.e. root) and ensure that Global System Manager is not currently being used. Place the BACRES diskette into the diskette drive, or for tape installations ensure that the A60.dir data file contains the sub-volumes extracted from the Global System Manager distribution tape labelled BACRES. To re-install Global System Manager key the following Unix command:

```
# global -i
```

The global command is fully described in section 6.3.

Once the installation has completed and you have reloaded the newly installed Global System Manager, run the $CUS command, select the "System Maintenance" function, and then select the "Restore saved customisation" function. This will copy the saved menus, table of authorized users and system customisation from a BACSAV diskette to the freshly installed system. Note that this option **cannot** be used to restore customisation from a BACSAV sub-volume.

It is also advisable to create an "emergency SYSRES diskette" by running $CUS, selecting the "System Maintenance" function and then selecting the "Create SYSRES diskette" function.

2.7 Upgrading Global System Manager (Unix)

The steps required to upgrade Global System Manager depend on two factors:

- the software layer, or layers, that are to be upgraded (see section 1.5);
- the distribution media (i.e. tape or diskettes).

2.7.1 Upgrading the Variant of the BACNAT Software

To upgrade ONLY the *variant* of the BACNAT software (see section 1.5.4) distributed on the BACNAT volume (i.e. diskette or tape) key:

```
# ./glinstall -u
```

For example, this option is used to upgrade from BACNAT variant 3.190 to variant 3.192.

Note that the same category of glinstall option is used regardless of the actual media (i.e. the same type of option is used for diskettes, QIC tapes and DAT tapes).

The glinstall -u option, and related options, are fully described in section 6.2.2.

2.7.2 Upgrading the Global System Manager Version

The steps required to upgrade the version (e.g. from V8.0 to V8.1) of an installed Global System Manager (Unix) configuration (see section 1.5.1) depend on the distribution media (i.e. diskettes of tape).

2.7.2.1 Upgrading the Global System Manager Version from Diskette

The upgrade of an existing Global System Manager (Unix) installation from diskette is a two-stage process. Firstly, upgrade the BACNAT software using the glinstall option described on section 2.7.1. This first step is necessary because a new BACNAT variant almost always accompanies a new version of Global System Manager (in any case it is always prudent to upgrade to the most recent BACNAT variant).

Secondly, upgrade the BACRES software using the global -i option described in section 2.6.4. The steps described in sections 2.1.7 to 2.1.12 should be followed to complete the upgrade installation. When performing an upgrade installation you will probably want to select the option to save and restore the existing Global System Manager customisation as described in section 2.1.10.

2.7.2.2 Upgrading the Global System Manager Version from Tape

This section applies to both QIC and DAT tapes.

Because the upgrade of Global System Manager from a tape involves copying data from the tape to a data file on the disk, the two steps described in section 2.7.2.1 for upgrading from diskettes are not sufficient. The glinstall -b option, which is fully described in section 6.2.3.2, must be used to upgrade an existing Global System Manager (Unix) configuration from tape. For example:

```
# ./glinstall -b
```

This option performs the following:

- extracts the BACNAT software from the distribution tape;

- extracts the Global System Manager starter system (i.e. BACRES, BEA etc.) encapsulated in the A60.dir directory, from the distribution tape;

- initiates the global -i command to re-install Global System Manager.

The steps described in sections 2.1.7 to 2.1.12 should be followed to complete the upgrade installation. When performing an upgrade installation you will probably want to select the option to save and restore the existing Global System Manager customisation as described in section 2.1.10.

2.7.3 Upgrading the Global System Manager Revision

The steps required to upgrade the *revision* (from Global System Manager V8.1, revision V8.1d to Global System Manager V8.1, revision V8.1e) of an installed Global System Manager V8.1 (Unix) configuration depend on the distribution media (i.e. diskettes of tape).

2.7.3.1 Upgrading the V8.1 Revision from Diskette

The upgrade of an existing Global System Manager (Unix) installation from diskette is a two-stage process. Firstly, upgrade the BACNAT software using the glinstall option described on section 2.7.1. This first step is necessary because it is always prudent to upgrade to the most recent BACNAT variant.

To complete the revision upgrade, reload Global System Manager from the installed system (i.e. the global command without any parameters in the normal way) and use $CUS to upgrade the Global System Manager revision from the BACRES, BEA etc. diskettes as described in the Global System Manager V8.1 Notes (document MSMNV8.1).

2.7.3.2 Upgrading the V8.1 Revision from Tape

This section applies to both QIC and DAT tapes.

Because the upgrade of Global System Manager from a tape involves copying data from the tape to a data file on the disk, the two steps described in section 2.7.3.1 for upgrading from diskettes are not sufficient. The glinstall -r option, which is fully described in section 6.2.5.2, must be used to upgrade the revision of an existing Global System Manager (Unix) configuration from tape. For example:

```
# ./glinstall -r
```

This option performs the following:

- extracts the BACNAT software from the distribution tape;

- extracts the Global System Manager starter system (i.e. BACRES, BEA etc.) encapsulated in the A60.dir directory, from the distribution tape;

To complete the revision upgrade, reload Global System Manager from the installed system (i.e. the global command without any parameters in the normal way) and use $CUS to upgrade the Global System Manager revision from the BACRES, BEA etc. diskettes as described in the Global System Manager V8.1 Notes (document MSMNV8.1).

3. Running Global System Manager

This chapter explains how to run Global System Manager under the Unix operating system.

Once Global System Manager has been started it is then responsible for running the Global range of application software.

3.1 Starting Global System Manager

Before attempting to use Global System Manager you must have the appropriate authorisation (see section 7.3.1) in order to run any of the Global applications. The appropriate shell variables must also have been set up (see section 8.1). Your system administrator should have done this when Global System Manager was installed.

3.1.1 Starting Global System Manager from Unix

Use the following Unix command to load Global System Manager from the Unix shell prompt:

```
# global
```

The global utility is fully explained in section 6.3.

The following message will be displayed if you are the only person currently using Global System Manager:

```
Configuring Global System Manager (Bnnnn Vn.nnn); please wait...
```

See section G.1 for a description of the numbers contained in this message.

After a short delay, a screen containing your Global System Manager serial number, SYSTEM number, user number and the *contract description* will be displayed briefly (see section 3.2). After a period of two seconds this introductory screen will be replaced by the main Global System Manager menu.

3.1.2 The Unix Date and Time Information

Global System Manager obtains the date and time information from Unix. If the date-time information supplied by Unix is more than an hour earlier than the last time Global System Manager was used, or more than a week later, a warning message will appear. For example:

```
WARNING - DATE IS MORE THAN 6 DAYS LATER THAN PREVIOUS DATE
```

or:

```
WARNING - DATE/TIME IS EARLIER THAN PREVIOUS
```

3.1.3 Speedbase Presentation Manager Licence Fee Password

If the installed software includes a Speedbase Presentation Manager Run-Time Licence the following warning message will appear within 30 days of the expiry date:

```
Your Presentation Manager rental is due by dd/mm/yyyy.

Please key password, <CR> to continue:
```

The rental fee password, obtained from your software supplier, consists of a single digit followed by 13 letters. Note that either upper-case or lower-case letters are acceptable.

If the installed software includes a Speedbase Presentation Manager Run-Time Licence the following warning message will appear when the licence expires:

```
You are in danger of infringing your licencing agreement.
This software expired on dd/mm/yyyy.

Please contact your software supplier to obtain a new password.

In emergency contact the "Emergency Rental Review
Department" on phone number (international phone number)
before 16th April 1995 or phone number
(international phone number) after 16th April 1995.

Please key password, <CR> to continue.
```

The rental fee password, obtained from your software supplier, consists of a single digit followed by 13 letters. Note that either upper-case or lower-case letters are acceptable.

If the installed software includes a Speedbase Presentation Manager Run-Time Licence the following warning message will appear 14 days after the licence expiry date:

```
You have infringed your licencing agreement.
This software expired on dd/mm/yyyy.

Please contact your software supplier to obtain a new password.

In emergency contact the "Emergency Rental Review
Department" on phone number (international phone number)
before 16th April 1995 or phone number
(international phone number) after 16th April 1995.

Please key password:
```

A rental fee password, obtained from your software supplier, consisting of a single digit followed by 13 letters **MUST** be applied. Note that either upper-case or lower-case letters are acceptable.

3.2 Global System Manager Start-Up

The Global System Manager start-up is normally performed invisibly. Under some circumstances, additional prompts may appear before the main menu is displayed.

3.2.1 The Contract Protection Message

Global System Manager commences by displaying a title line containing your Global System Manager serial number, SYSTEM number and user number. Appearing below the title line is the *contract description*, a short summary of your licensing agreement. **PLEASE ENSURE THAT YOU AND YOUR COLLEAGUES HONOUR THIS AGREEMENT AT ALL TIMES**.

Note that it is possible to customise Global System Manager so that the contract protection message is suppressed (see section 2.4.9 of the Global Configurator Manual for more details).

3.2.2 The Operator-id Prompt

Normally, your Global System Manager operator-id will be derived from your Unix login name by a mapping within the Systems file (see section 7.3.5). If your Global System Manager operator-id cannot be determined from the information within the Systems file, the following prompt will appear immediately below the contract description:

```
PLEASE KEY YOUR OPERATOR-ID:
```

Reply with your operator-id, a code up to four characters in length. Normally, you just use your initials, but sometimes special codes will be allocated by your system supervisor.

3.2.3 The Terminal Type Prompt

Normally, the information that Global System Manager requires to operate on your terminal will be derived from your Unix TERM shell variable by a mapping within the Systems file (see section 7.3.4). If the Global System Manager terminal information cannot be determined from the information within the Systems file, a prompt of the following form appears:

```
PLEASE KEY TERMINAL CODE (code):
```

requesting you to supply the terminal code identifying the device. The default terminal code, displayed in brackets, is the code used for the previous Global System Manager session on the terminal. Usually, the default terminal code will be correct. However, the default terminal code will NOT be correct if this is the first time your screen has been used to run Global System Manager, if you have changed terminals (e.g. replaced a Wyse-50 screen by a Wyse-60) or if $STATUS has been used to purge the User File. You can, if necessary, list the available terminal codes by keying LIS. For example:

```
PLEASE KEY TERMINAL CODE (116 ):LIS
0     BASIC TELETYPE SUPPORT ONLY
161   MICROCOLOUR M2200
163   WYSE WY-50+
187   WYSE WY-370 (ANSI KEYBOARD)
197   TCL NyCE COLOUR TERMINAL
PLEASE KEY TERMINAL CODE (163 ):187
```

In this example, Global System Manager continues, knowing you are working at a Wyse WY-370.

If the terminal information derived from your Unix TERM shell variable by the mapping within the Systems file does not correspond to a valid Global System Manager terminal code, the following messages will appear:

```
TERMINAL TYPE nnnn NOT FOUND
$57 INITIATION WARNING 450 - INVALID TERMINAL TYPE
```

The warning message will be followed by the PLEASE KEY TERMINAL CODE: prompt described above.

Important note: If this is the first time your screen has been used, or if Global System Manager has been reloaded after the $STATUS PUR command (see the Global Utilities Manual) has been used to purge the $$USER file, the default terminal type (in brackets) may be incorrect or inappropriate. If in doubt, use the LIS command (as above).

The numeric terminal code may be prefixed by one of the following letters:

E The screen is being used by a terminal emulator package. This option is obsolete - do not use;

W Start the Global System Manager session in "wide-mode" if the terminal supports wide-mode working.

3.2.4 The Password Prompt

If your installation has decided to employ authorization checking, you will now be prompted for your password:

```
PLEASE KEY PASSWORD:
```

You should enter the password, which for security reasons will not be displayed on the terminal. If you want to change your password then key it as usual but terminate the input with <CTRL B> rather than <CR>. You will then be prompted for a new password. The option to change the password at sign-on is only allowed if the password is less than 8 characters. If the password is 8 characters in length then it can only be changed using $AUTH as documented in section 6.3 of the Global System Manager Manual.

If you have supplied an invalid operator-id or password the following message will be displayed:

```
$94 YOU ARE NOT AUTHORIZED TO SIGN-ON - PLEASE CONTACT A SUPERVISOR
```

This message is followed by the initial sign-on screen. The above warning message is also displayed if you key <ESCAPE> to either the operator-id or password prompts.

3.3 Global System Manager

The following sections in this chapter briefly summarize the information described in the Global System Manager Manual.

3.4 The Main Menu and Ready Prompt

Once the start-up procedure is complete, the main Global System Manager menu will be displayed, unless your installation has decided not to use menus. You will be able to select any one of up to 16 functions by simply keying the appropriate number (terminated by <CR>, of course) in response to the selection prompt which appears at the bottom of the screen. Once the selected function completes, the main menu will be redisplayed so you can continue with other work.

You should note that some menu functions may be restricted to operators with special authority codes. If you are not allowed to use a particular function, an asterisk will be displayed instead of the function number.

Provided you have a sufficiently high authority code, you can run any Global System Manager command (or indeed, any other program) by keying its name in response to the selection prompt, instead of a function number. For example, you can key $T to run the screen information command which displays or prints a page of information about your screen and keyboard. You can also get out of the main menu to the GSM P*n* READY: prompt by keying the word READY in response to the selection prompt.

If you have chosen not to use menus, then instead of a menu the GSM P*n* READY: prompt will be displayed by Global System Manager when you sign on. You can key the name of any command or program in response to this prompt to cause it to be run.

3.5 Concurrent Screen Handling

Concurrency allows you to run several programs from one screen. Associated with each physical screen are up to 9 partitions (usually 4) each of which can be running a different program. At any time the screen will be displaying one of the partitions, as indicated by the *Status Line* at the top (or sometimes the bottom) of the screen showing the partition number. For example:

```
Partition 1
```

On some screens, the writable portion of the status line does not extend across the entire width of the screen. On such screens the partition number information will be condensed. For example:

```
P1
```

If your screen does not have space to display the status line permanently, you can use the <SYSREQ> M system request to cause it to be displayed temporarily over the top line of the screen, and <SYSREQ> Z to remove it. Chapter 4 of the Global System Manager Manual fully describes system requests.

The partition number is followed by the SYSTEM identification letter or number. For example:

```
Partition 1 System A
```

or:

```
Partition 3 System 30
```

Again, this message may be condensed if the writable portion of the status line is shorter than the screen width. For example:

```
P2 S1B
```

The partition number will also appear in the Global System Manager GSM P*n* READY: prompt (e.g. GSM P3 READY:). You can swap to another partition by entering a special keystroke, usually <SYSREQ> 1 for the first partition, <SYSREQ> 2 for the second, etc. (If some key other than <SYSREQ> is used the screen information command $T will tell you which.)

A program will continue running, and writing messages to the screen, even if its partition is not the one selected, but you will only be able to read the messages when you swap back to its partition. However, if a program needs a response keying it will be halted until you swap to its partition, since the keyboard always sends characters to the currently selected screen.

Concurrency enables you to leave a lengthy, automatic process (such as printing a report) running in one partition and use another partition to do something else which requires the use of the screen and keyboard. You can also break off from what you are doing in one partition in order to run an enquiry program in another, and then continue in the original partition when you have your information. You can, if you wish, leave an enquiry program permanently loaded in one partition so that you do not have to wait for it to be loaded when you need it.

3.6 Creating a New Shell

Keying <SYSREQ> . whilst running Global System Manager causes the Global application in the current partition to suspend whilst a new Unix shell is created:

<SYSREQ> . (system request full stop)

The shell prompt should appear at the beginning of the next line below the cursor (unless you have redefined the SHELL variable to execute another command).

Exiting from this shell will redisplay the Global System Manager partition.

Note that you are not allowed to run Global System Manager from the new shell. This is to prevent a user from nesting multiple copies of Global System Manager.

3.7 Exiting from Global System Manager

If you do not wish to run any further Global applications at the end of a session you can exit from Global System Manager and return to Unix by selecting the appropriate function from the main menu or by running the $E utility. If you have completed your session with Global software for the day you should exit Global System Manager by running the $BYE utility (see section 3.8). $BYE may be selectable from your menu. $E temporarily suspends Global System Manager whereas $BYE terminates the process (it is therefore quicker to run *global* after using $E than after using $BYE).

If your screen has concurrency you must be in partition 1 to sign off, and all other partitions must be at a suitable point, for example at a menu or GSM P*n* READY: prompt. If they are not, an error message will be displayed, and you will be automatically swapped into the lowest numbered active partition. If you sign off in a partition other than 1, that partition is rendered inactive and you are automatically swapped into partition 1.

$E is fully described in section 4.3.

3.8 Terminating Global System Manager

If you do not wish to run any further Global applications you can exit from Global System Manager and return to Unix by running either the $BYE command or use the equivalent TER instruction in $STATUS (see Chapter 5 of the Global System Manager Manual). Before returning you back to Unix, if you are using a non-standard Global System Manager configuration, $BYE performs two additional functions. Neither of these functions apply to a standard configuration. Firstly, if your SYSTEM has been configured for more than one USER (see section 7.4.1) $BYE checks that no other users are using your SYSTEM. Secondly, if your SYSTEM has been configured to be a file-server (see section 7.4.3), $BYE ensures that no users on other SYSTEM's have files open on volumes on your file-server. If either of these checks fails, an error message will be displayed, and you should take the appropriate action recommended in Appendix A of the Global System Manager Manual.

The following message, displayed by $BYE will appear briefly:

```
GLOBAL SYSTEM MANAGER TERMINATED
```

followed by a shell prompt.

If you were the only person using Global System Manager then the following message will also appear:

```
Terminating processes and tidying Global System Manager resources
```

Note that these messages are displayed by daemon processes so may appear after the shell prompt has been displayed. The shell prompt may even be intermingled with these messages. It may be necessary to key <CR> to the Unix shell to obtain the shell prompt on a newline (see section G.76).

$BYE is fully described in section 4.1.

4. Additional Utilities

In this chapter we give detailed accounts of some important Global System Manager commands that are only available when the host operating system is Unix. These commands are not fully described in the Global System Manager Manual.

In addition, this chapter also describes those Global System Manager commands which execute in a slightly non-standard way when the host operating system is Unix.

4.1 $BYE - Terminate Global System Manager

$BYE should be used if you want to terminate Global System Manager and return to Unix. For those non-standard configurations that include more than 1 USER per SYSTEM, $BYE checks that no other users are running Global System Manager on your SYSTEM. Furthermore, if your SYSTEM is a file-server (another non-standard configuration) $BYE also ensures that no users on other SYSTEM's have files open on the simulated volumes accessed via your file-server SYSTEM.

If your screen is configured with a number of concurrent partitions, $BYE **must** be run from partition 1. All other partitions must be at either a menu prompt or the GSM READY: prompt. If an attempt is made to run $BYE from a partition other than P1, the following message will appear:

```
GSM P4 READY:$BYE
CAN ONLY TERMINATE GLOBAL SYSTEM MANAGER FROM PARTITION 1
```

$BYE does not prompt for any further information. If all is satisfactory, the following message is displayed:

```
GLOBAL SYSTEM MANAGER TERMINATED
```

You will be returned to the Unix shell prompt.

YOU SHOULD NOT SIMPLY SWITCH OFF YOUR COMPUTER ONCE YOU HAVE RUN $BYE. USE WHATEVER SHUTDOWN PROCEDURES ARE APPROPRIATE FOR YOUR UNIX OPERATING SYSTEM.

If you run $BYE when other users are running Global System Manager only the GLOBAL SYSTEM MANAGER TERMINATED message will be displayed. If you are the last user to exit Global System Manager the GLOBAL SYSTEM MANAGER TERMINATED message will be followed by the following message:

```
Terminating processes and tidying Global System Manager resources
```

If other operators on your SYSTEM are still active $BYE displays the following warning message and will not terminate Global System Manager:

```
OTHER USERS STILL USING THIS SYSTEM MANAGER
Key <CR> to retry, <ESC> to abandon
```

If other operators are accessing files via your SYSTEM (which must be configured as a file-server), $BYE 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 $BYE again once these users have signed off (by using $E - see section 4.3), or closed the necessary files (usually by exiting from an application).

If $BYE prevents the termination of the Global System Manager session because other operators are accessing files via your SYSTEM, you can key <CTRL B> to the retry prompt to terminate Global System Manager **WITHOUT** performing the open file checking. **THIS OPTION MUST BE USED WITH GREAT CARE BECAUSE TERMINATING GLOBAL SYSTEM MANAGER WHILE FILES ARE GENUINELY IN-USE MAY CAUSE DATA CORRUPTION.**

Note that the <CTRL B> option does **NOT** allow $BYE to ignore other operators that are signed on to your SYSTEM.

$BYE is equivalent to the $STATUS terminate (TER) command (see Chapter 5 of the Global System Manager Manual).

4.2 $CUS - Modify Installed Global System Manager

$CUS is documented in section 6.1 of the Global System Manager Manual. This section just describes those aspects of Global System Manager customisation that are not available when the host operating system is Unix.

4.2.1 Sign-on Customisation

In the Sign-On Customisation section of $CUS, the "Automatic Start" option is not available when the host operating system is Unix.

4.2.2 Printer Characteristics Customisation

When Global System Manager is installed on Unix, only the following prompts in the "Printer Characteristics" section of $CUS are issued:

```
Set up device attributes for a printer unit.
PRINTER UNIT:501 Direct Printer Output
The printer timeout period is rounded to multiples of 10 seconds

Printer timeout period (currently 20 sec):<CR>
Does the printer automatically provide LF after CR (N):<CR>
Do you wish the printer to throw a page at startup (N):<CR>
Do you wish the printer to throw a page at end of file (Y):<CR>

Customize printer immediately (N):Y
```

4.2.3 System Maintenance Customisation

In the System Maintenance section of $CUS, the "Apply nucleus update" and "Alter master node address" options are NOT available for Global System Manager (Unix) configurations. Note that most of the options in the System Maintenance menu expect the $BA logical unit assignment to be accurate. Note also that the special technique described in section 2.7.3.2 must be used in order to upgrade the revision of Global System Manager distributed on tape.

4.2.4 Configuration Maintenance

The "LAN Buffers", "RAM Disk/Cache", "Network Control Block" and "Extended customisation" options from the $CUS Configuration Maintenance menu are NOT available for Global System Manager (Unix) configurations.

4.2.5 Printer Control Files

As explained in section 6.2.5 of the Global System Manager Manual, the Printer Control Files for Global System Manager (Unix) configurations are named $$P*nnn* and are allocated on unit $M.

When the host operating system is Unix, the Global System Manager start-up processing does not flag the Printer Control Files to indicate the "undefined" printer status. If the printers were flagged as "undefined" on each occasion a user invoked Global System Manager, the Start Sequences would be sent every time. To send a Start Sequence to a printer you must use either $P (refer to section 5.30 of the Global System Manager Manual) or the I instruction of $SP (refer to Chapter 8 of the Global System Manager Manual).

4.3 $E and $END - Operator Sign Off

$E should be used to exit once you have finished using Global System Manager utilities or Global applications. When $E is used, your session with Global System Manager is temporarily suspended and you are returned to the Unix shell prompt from which you ran the program *global*. It is particularly important to exit using $E when away from your screen if you have access to sensitive data which must be kept secure. Running $E forces anyone wishing to use your screen to go through the System Global Manager authorisation mechanism (providing $AUTH has been used to set up a table of authorised users).

In addition, $E leaves the screen in a state which can be readily displayed by the system supervisor using $STATUS. Thus the supervisor can check to see if anyone is using the Global applications before reconfiguring Global System Manager. Note that $BYE and the $STATUS TER command demand that $E is run on all screens associated with this SYSTEM before it allows Global System Manager to terminate.

If your screen has a number of concurrent (virtual) partitions you must be in partition 1 to exit, and all other partitions must be inactive. If you try to exit from a partition other than partition 1, $E will display the following message:

```
PARTITION INACTIVE
```

and you will be automatically be swapped into partition 1. If you try to exit from partition 1 while another partition is still in use, Global System Manager will display the active partition. You will have to come out of whatever application is being run in that partition and return to the main menu.

Running $E will clear the assignment table of all temporary unit assignments. It should also be used if you want to specify a different TERM type or short user name (operator ID).

Note that you can set up automatic logging off (which has the same effect as $E) to occur if a screen has not been used for some time by using the functions of the menu system (see Chapter 7 of the Global System Manager Manual).

4.3.1 $END - Operator Sign Off

$END should be used if you want to end your current Global session and immediately restart another one. $END may be useful if you want to change your operator-id or terminal type, if the mapping in the Systems file (see section 7.3.4) is disabled, without returning to Unix.

4.4 $REMOTE - Non-networked File Transfer Utility

$REMOTE allows you to transfer files between two computers running Global System Manager which do not have compatible direct access volumes and are not linked by a network. For example, you can transfer files from one computer that only supports 3½" diskettes to another that only supports 5¼" diskettes. The computers must be connected using a serial RS232 cable, and both must support $REMOTE and be capable of using the same baud rate.

At a baud rate of 9600, $REMOTE can transfer about 900 bytes per second, which means that 1Mb of data will take approximately 20 minutes to transfer.

IMPORTANT NOTE: THIS SECTION EXPLAINS $REMOTE IN GENERAL TERMS AND SHOULD BE READ IN CONJUNCTION WITH SECTION G.71 WHICH DESCRIBES THE NUMEROUS SPECIAL FEATURES THAT MUST BE CONSIDERED WHEN USING $REMOTE ON A GLOBAL SYSTEM MANAGER (UNIX) CONFIGURATION.

4.4.1 Using $REMOTE

When $REMOTE is used to connect two computers, the direct access units of the remote computer, 100-299, can be accessed as if they are units 800-999 respectively of your local computer. Although typically you would simply wish to copy files between the computers (using $F), any program or command can be used and can access the remote devices as normal direct access units.

Note that it is not possible to use $V to format a diskette or hard-disk on a remote computer. Formatting can only be performed locally.

Note also that only numeric unit addresses (e.g. 201) on the remote computer can be accessed as if they are units on your local computer (e.g. 901). Units on other files servers (e.g. A01) are NOT accessible when $REMOTE is used.

The computer whose screen you are using is termed the *master* and the remote computer is termed the *slave*. Before you can access files on the remote computer the following steps must be performed:

1. Connect up the RS232 link between the computers;

2. Run $REMOTE on the slave computer and reply S to the OPTION: prompt;

3. Run $REMOTE on the master computer and reply M to the OPTION: prompt.

The direct access units on the slave should now be accessible from the master. While the link is in use the slave computer cannot be employed for any other purpose, whereas the master appears as normal, except that the user area is decreased by about 2 Kbytes and units in the range 800-999 are available.

When you have finished using the link, terminate $REMOTE on the master, by re-running $REMOTE (which will release the 2 Kbytes of memory occupied) then on the slave by keying <CTRL W>. If you physically disconnect the link (i.e. disconnect the serial cable) but fail to

terminate $REMOTE on the master, you will notice a long delay each time you return to the ready prompt. This delay will also occur if you terminate $REMOTE on the slave before terminating on the master.

$REMOTE can be used on multi-user SYSTEMs, but since it takes exclusive control of the process while transferring data it is best to exclude other users when you run it.

4.4.2 The Option Prompt

Normally you need only key two commands in response to the option prompt: S to put $REMOTE into slave mode, or M to put $REMOTE into master mode. There are, however, a number of other options that can be used to control the way in which $REMOTE works. The defaults for these options are taken from the configuration file.

4.4.2.1 The H Option

The H option is used to specify the name of the software module which is used to control the serial port during a $REMOTE session. The name of the module for Global System Manager (Unix) configurations is always %.C2R (see section G.71.1.1). The necessary option would be keyed as H=%.C2R. **You must not change this option**.

4.4.2.2 The D Option

The D option is used to specify the device address to be employed by $REMOTE. For Global System Manager (Unix) configurations the device address is specified by a Unix shell variable (see section G.71.1.2) so this option can be ignored.

4.4.2.3 The B Option

Use the B option to specify the baud rate. The baud rate can be specified in decimal (normally), hexadecimal or octal. For example, the two commands B=#2580 and B=9600 both set the baud rate to 9600 (see section G.71.1.3).

4.4.2.4 The T Option

The T option is used to specify a time-out for communication error conditions (see section 4.4.6). On other Global System Manager configurations (e.g. Global System Manager (BOS) configurations) a time-out of zero allows $REMOTE to calculate its own time-out. Because of the large amount of buffering employed by Unix it is not, in general, possible for the $REMOTE handler to calculate a time-out value. Consequently, the T option should always be set to a non-zero value when $REMOTE is used on a Global System Manager (Unix) configuration. If a value of 0 is specified, the default value of 250 is used. The time-out value specified is not in seconds, but in terms of executions of an internal processing loop.

4.4.2.5 Changing the Default Options

On Global System Manager (Unix) configurations, $REMOTE standardly transmits and receives at 9600 baud. When running $REMOTE it is possible to modify the baud rate and the time-out period. In the following example dialogue, the baud rate is changed from 9600 to 19200 and the time-out is changed from 250 to 500:

```
GSM READY:$REMOTE
$39 OPTION:H=%.C2R          Name of $REMOTE module
$39 OPTION:T=#00FA          Time-out value (# = hexadecimal)
$39 OPTION:D=#0000          Port address (not used)
$39 OPTION:B=#2580          Baud rate (# = hexadecimal)
$39 OPTION:T=500            Change time-out value
```

```
$39 OPTION:B=19200          Change baud rate
$39 OPTION:
```

Note that the port address can only be changed by modifying a Unix shell variable (see section G.71.2) and reloading Global System Manager. Note also that the $REMOTE handler on Global System Manager (Unix) configurations recognises the "hidden" Y and Z options (see section G.71.3).

4.4.3 Communication Errors

If the RS232 link has not been properly established then accessing the slave with $F will probably result in the following dialogue:

```
GSM READY:$F
$66 INPUT DEVICE:901 INVALID - REINPUT
```

If an irrecoverable error is detected during data transfer an error message of the following form will be displayed:

```
* NETWORK ERROR ON description - unit
* RETRY?:
```

If these errors occur when you try to access the remote units then see below.

4.4.4 Troubleshooting a $REMOTE Link - RS232 Wiring

The two computers must be connected using an RS232 cable. $REMOTE does not make use of the modem control lines: RTS, DCD, or CTS etc, and normally only needs wires connected between the TRANSMIT DATA, RECEIVE DATA and GROUND pins of the RS232 connector (i.e. normally pins 2, 3 and 7). The pin connections on a simple cable would be:

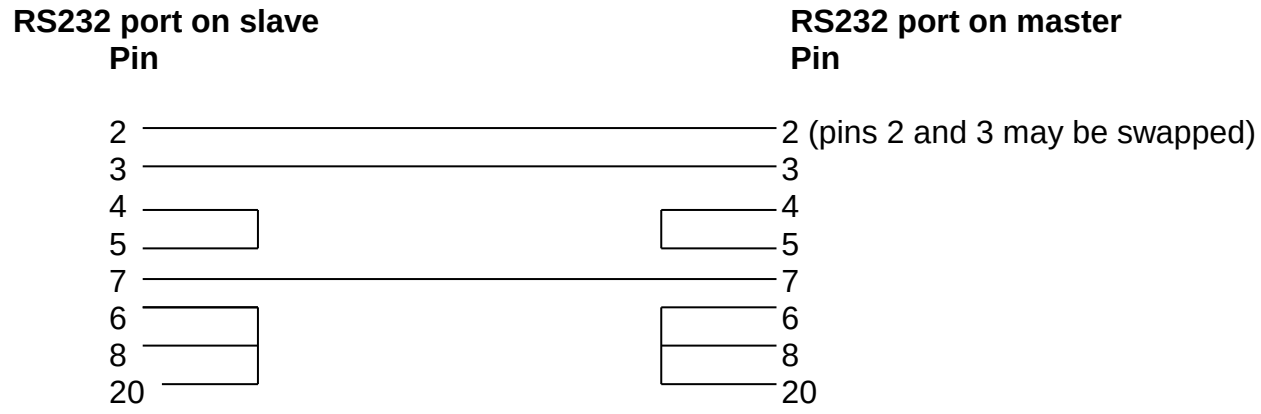
RS232 port on slave Pin		**RS232 port on master Pin**
2	———	2
3	———	3
7	———	7

This represents the ideal situation, but in practice it will probably be affected by two factors.

Firstly each computer will have its own idea of which pin is TRANSMIT DATA and which RECEIVE DATA. They may both be transmitting on pin 2 and receiving on pin 3. To overcome this try the following cable connection:

RS232 port on slave Pin		**RS232 port on master Pin**
2	———	3
3	———	2
7	———	7

The second factor relates to the modem control lines. Although $REMOTE does not use these, some computer hardware, and Unix device driver, demand that they be set up. In some cases this will need very special wiring but most situations can be solved with the following cable:



Pins 4 and 5 are connected together at each end of the cable as are pins 6, 8 and 20.

4.4.5 Troubleshooting a $REMOTE Link - Checking Device Address

Before connecting the two computers together it is wise to check that the two separate ends of the link perform correctly. To do this, isolate each computer from the other, then connect a "breakout box" to each computer in turn while running in slave mode. While in slave mode, the transmit line should be continuously active so you should see the indicator for either pin 2 or pin 3 illuminated on the box.

The same test can be carried out using a serial terminal instead of a breakout box, but the terminal must support "transparent" or "monitor" mode so that it can display characters in the range #00-1F. When running in slave mode $REMOTE sends a continuous stream of "ENQ" characters (#05) which will be visible on a suitably configured terminal.

If none of the indicators on the box light up, or "ENQ" characters are not being transmitted, then it is for one of the following reasons:

- You have plugged the cable into the wrong RS232 port or used the wrong device address with the D option;
- You need to connect some of the modem control lines;
- There is a broken or missing connection in one of the wires of the cable.

Check that both computers transmit in slave mode. If they don't there is no point in connecting the two computers together, they are unlikely to communicate with each other.

If both computers transmit successfully in slave mode then connect the two computers together using the cable, and try one in slave mode and one in master mode. If things don't work at this stage it is probably for one of the following reasons:

- Pins 2 and 3 at **one** end of the cable need to be swapped;
- The computers are not using the same baud rates;
- The modem lines are incorrectly wired and are not operating the same way as when connected to the "breakout box";

- You may need to establish a time-out. See next section.

4.4.6 Timeout Handling

The T option is used to set up time-outs on both the master and slave so that one computer does not hang if it gets no response from the other. The time-out handling for Global System Manager (Unix) configurations is something of a "black-art". Please refer to section G.71.3 for further details.

4.5 $REORG - Reorganise/Reallocate Data File Subvolumes

The $REORG command reorganizes the (subvolumes) volumes of a Global data file (domain), increasing or decreasing their sizes as required. The command can also be used to reorganize files within a single volume in a similar way. However, most Global software modules contain their own facilities to change file allocations, and if available these must be used rather than $REORG.

Before starting to reorganize a domain or volume, you should make sure you have recent backups of all the data it contains, since if the reorganization fails some or all the data will be lost. You should also verify the disk by using the $F VER instruction, since a read error on the disk will cause one or more volumes or files to be lost.

Important note: There is very little point in using $REORG to simply condense the sub-volumes of a Discrete Data File domain (aka Separated Subunit Domain) as the domain layout is automatically reconstructed each time Global System Manager is invoked. Furthermore, the Unix utility *glreorg* (see section 6.10) can be used to alter the size of a subvolume within a Discrete Data File domain. This utility is more efficient than $REORG. However, $REORG (or the $F CON instruction) are still required in order to condense the Global System Manager files within a subvolume of a Discrete Data File domain.

If there are volumes allocated on a domain that are no longer required these should be de-allocated, using $V, before $REORG is used (see section 4.10.5).

A domain or volume cannot be reorganized while files on it are in use. You should run the command when no one else is using the computer as it will lock out other users while it is in progress.

When you run $REORG you will first be asked to specify the unit you want to reorganize. The name of the domain or volume will be displayed for confirmation. If it is not the one you want key <ESCAPE> to the subsequent prompt.

```
Please select a function:$REORG
........
(explanatory text)
........
REORGANIZE UNIT:200 SYSDOM
```

Next you assemble a list of the new sizes the volumes (if you specified a domain address) or the files (if you gave the address of a subvolume) are to have after the reorganization. A series of prompts of the form:

```
KEY UNIT ADDRESS TO CHANGE SIZE, ? TO LIST, R TO REORGANIZE:
```

or:

```
KEY FILE NAME TO CHANGE SIZE, ? TO LIST, R TO REORGANIZE:
```

will appear (the unit or file prompts).

If you want to change the size of any volume on the domain, or of any file on a volume, key its unit address or name respectively. The current size will be displayed, and you can key in a new size in bytes, Kbytes or Mbytes. For example:

```
CURRENT SIZE 395264 (386.0K)
KEY NEW SIZE, OR M FOR MAXIMUM:
```

DO NOT ATTEMPT TO USE $REORG TO TRUNCATE DMAM FILES.

If the new size you specify is smaller than the current size, the volume or file will be checked to make sure that there will remain sufficient space for its current contents. If there is not, a message will be displayed indicating the minimum possible size the volume or file's contents require. If the reduction in size requires a volume to be reorganized so that all the files are at the beginning, the following prompt will appear:

```
UNIT uuu REQUIRES REORGANIZING - CONTINUE ? (Y):
```

Key <CR> to continue and reorganize this volume so that all the free space is at the end, or N to return to the unit prompt and leave the current volume the same size. Note that the reorganization of the volume takes place as soon as you key <CR>.

The special reply of M to the new size prompt causes the volume or file to be given all the available free space on the domain or volume. You can, of course, specify this option for only one volume on any one domain, or for one file on a volume.

To check what new sizes you have specified, and how much space is free, key ? to the unit or file prompt. Note that the new sizes shown may be slightly greater than you specified, as space is required for an index, and all sizes are rounded up to a whole number of logical tracks (for efficiency reasons the Global data format is normally coupled with the hard-disk geometry for the computer - on Unix this feature is relaxed to use multiples of the Unix kernel buffer size).

Up to this point none of the volumes or files will have been moved or had their sizes changed (though they may have been internally reorganized). If you want to abandon the reorganization of the domain or volume, key <ESCAPE> to quit and it will be left unchanged.

When you are happy with the new size allocations key R to start the reorganization process. The command displays messages indicating its progress and a "Reorganization complete" message when it has finished. If a read or write error occurs, and persists when you retry, reply N to the retry prompt. The address of the volume or file affected will then be displayed: You will need to restore this volume or file later from a backup as at least one of its files or some of its data will be corrupt. The reorganization will then continue: **DO NOT ATTEMPT TO ABANDON THE REORGANIZATION OR YOU WILL LOSE ALL YOUR DATA.** If there is an error on the domain it is likely to be reported twice: Once on the original volume containing the error, and once on the volume being moved to that part of the disk.

4.6 $S - Obtain System Information

$S is documented in Chapter 5 of the Global System Manager Manual. This section just describes those aspects of the System Information report that are dependent on the host operating system (i.e. Unix).

In addition to the information described in Chapter 5 of the Global System Manager Manual, $S displays the BACNAT variant of the Global System Manager (Unix) nucleus (see section 1.5.4).

$S does not display any details of the executive library (i.e. +.C0) or the controller library (i.e. +.C2) which are both empty in Global System Manager (Unix) configurations (see sections 1.5.2 and 1.5.3).

4.7 $STATUS - Control Multi-user Configuration

$STATUS is documented in Chapter 5 of the Global System Manager Manual. This section just describes those aspects of status-reporting that are dependent on the host operating system (i.e. Unix).

The $STATUS LIL (List only IN-USE computers/systems) is not supported on Global System Manager (Unix) configurations. This command is not necessary because the generic List Systems command (i.e. <CR>) lists only details of those SYSTEM's that are currently IN-USE.

In addition to reporting all the operators that are currently using Global System Manager, the $STATUS list command displays details of all file-server SYSTEMs. In addition to displaying the file-server identifier (e.g. "A", "B" etc.), the $STATUS report includes the description, if any, associated with each file-server. $STATUS descriptions are established using the DES command - see Chapter 5 of the Global System Manager Manual for full details.

The $STATUS DIS command is functionally equivalent to the RES command on Global System Manager (Unix) configurations. However, whereas the RES command restarts the user at the copyright message (as if the user had run the $END command, see section 4.3.1), the DIS command ends the Global System Manager session and returns the user to the Unix shell (as if the user had run the $E command, see section 4.3).

Important note 1: The $STATUS RES, CAN and DIS commands cannot be used to REStart, CANcel or DISconnect users on other SYSTEM's. Thus, for most Global System Manager (Unix) configurations, which include only a single user per SYSTEM, these commands are effectively useless.

Important note 2: The $STATUS GET and REL commands only affect other users on the same SYSTEM. Thus, for most Global System Manager (Unix) configurations with 1 user per SYSTEM, the GET and REL commands are effectively useless. However, the QUI command, which affects all Global System Manager users, does operate as documented in Chapter 5 of the Global System Manager Manual.

The $STATUS MON command is not supported on Global System Manager (Unix) configurations.

The remaining $STATUS commands work as documented in Chapter 5 of the Global System Manager Manual.

4.8 $TDUMP - Tape Backup/Restore Utility

The cartridge tape dump/restore utility, $TDUMP, is not available on Global System Manager (Unix) configurations.

Note that $TAPE, the Streamer Tape utility, which provides a much faster alternative to $TDUMP and is not restricted to the $TDUMP size limitation of 127.9Mb, is available for most Global System Manager (Unix) configurations. Please consult your Global Configuration Notes and section G.64 for further details.

4.9 $U - Display/Print Unit Information

$U is documented in Chapter 5 of the Global System Manager Manual. This section just describes those aspects of the unit information that are dependent on the host operating system (i.e. Unix).

The general format of the information displayed by $U is as follows:

Unit range	*Description*
100 - 199	Private diskettes (optional)
200 - 299	Private data file descriptions (optional)
500 - 599	**Printers**
600 - 699	**Data file descriptions on the master SYSTEM (usually file server SYSTEM A)**
a00 - a99	**Diskettes on file server SYSTEM A (optional)**
A00 - A99	**Data file descriptions on file server SYSTEM A**
b00 - b99	Diskettes on file server SYSTEM B (optional)
B00 - B99	**Data file descriptions on file server SYSTEM B (optional)**
c00 - c99	Diskettes on file server SYSTEM C (optional)
C00 - C99	**Data file descriptions on file server SYSTEM C (optional)**

etc.

z00 - z99	Diskettes on file server SYSTEM Z (optional)
Z00 - Z99	Data file descriptions on file server SYSTEM Z (optional)

On most configurations only the units printed in emboldened text will be available.

Note that $U automatically displays details of all available file-server SYSTEMs.

To display the data file descriptions associated with a single, specific file server SYSTEM, key S to the following prompt:

```
Key P to print, S for different system, <CR> to page, <ESC> to exit:
```

and specify the file server SYSTEM letter.

For example, to display the data file descriptions associated with file server X, use the following dialogue:

```
Key P to print, S for different system, <CR> to page, <ESC> to exit:S
Specify System-id (A):X
```

The information displayed by $U describing the diskette formats supported on your system refers to the characteristics of the physical devices. For example:

```
a40   O2D         1ST DISKETTE DRIVE CAPACITY 1428.0K ( 63 FILES)
                  2 SURFACE(S) OF 80 TRACKS WITH 18 512-BYTE SECTORS
```

The "O2" diskettes are physically double-sided, double-tracking with 18 512-byte sectors.

Normally, the 25 character diskette drive descriptions will be derived from information held in the Global configuration file. However, if the technique described in section 9.2.3 has been used the diskette drive description will be the Unix device name as defined in the Systems file.

Note that $U reports on the virtual geometry of any virtual diskette volumes that may be configured (see section G.52).

The information displayed by $U describing Discrete Data Files (i.e. Separated Subunit Domains, format codes T151Z) refers to the mounted Unix file system. For example:

```
A00   T151Z       Discrete Data Files CAPACITY 337.4M (250 FILES)
                  4 SURFACE(S) OF 10798 TRACKS WITH 16 512-BYTE SECTORS
```

The CAPACITY refers to the size of the mounted file system. The SURFACE, TRACK and SECTOR information describe a virtual disk and do not relate, in any way, to the geometry of the physical disk. Note also that the SPARE SPACE value displayed by the $F or $V list of a Separated Subunit Domain refers to the total amount of spare space on the mounted Unix file system (rounded down to a multiple of the virtual track size).

The FORMAT is specified by an ANA code. An ANA code is made up of an Alphabetic part followed by a Numeric part and trailing Alphabetic, for example "O2D". The preceding AN part ("O2" in the example) describes the physical (diskette) geometry and the trailing A part ("D" in the example) describes the location of the Global directory.

Normally, the 25 character disk descriptions will be derived from information held in the Global configuration file. However, if the technique described in section 9.2.1 has been used the disk description will be the Unix directory pathname as defined in the Systems file.

Note that V8.1 $U displays the maximum number of files for diskette and hard disk formats.

The information displayed by $U also describes the printer outputs which have been defined (see section 7.3.6). For example:

```
500   Direct printer output

501   Direct printer output

510   Spooled printer output

511   Spooled printer output
```

Normally, the 25 character printer descriptions will be derived from information held in the Global configuration file. However, if the technique described in sections 9.5.1 and 9.5.2 has been used the printer description will be either the Unix printer device (for a direct printer) or the Unix spooler command (for a spooled printer) as defined in the Systems file.

$U displays no useful information regarding the capacity of any tape drives connected to your computer.

4.10 $V - Volume Maintenance Utility

$V, the volume maintenance utility, is used to prepare diskette and data file volumes for use with Global System Manager. Before a volume can be used, it must be initialised and allocated a volume name by which it is known.

When you run $V it displays a main menu and prompts for a function. To allow software compatibility with Global System Manager on other host operating systems, some options that appear are not available with Global System Manager (Unix). The functions listed are as follows:

Initialise is used to set up data file volumes (or subunits). You are prompted for the unit address, volume description, volume name and size. This option can also be used to initialise diskettes - you are prompted for the unit address and volume name.

Initialise and Verify performs the same function, but also verifies the volume (diskette or hard disk subunit) after initialising it, to check that each track is readable.

Format is not available for Global System Manager (Unix) configurations.

Format and verify is not available for Global System Manager (Unix) configurations.

Scratch domain is used to wipe all current data from a specified hard disk domain. **THIS FUNCTION SHOULD ONLY BE USED BY SYSTEM SUPERVISORS.**

Reformat faulty track is not available for Global System Manager (Unix) configurations.

Allocate alternate track is not available for Global System Manager (Unix) configurations.

Amend domain error map is not available for Global System Manager (Unix) configurations.

Set up special access option is for use by development programmers only.

List Directory is identical in effect to the $F LIS option. It lists the files on a specified hard disk or diskette volume, or the volumes on a specified hard disk domain. You might want to do this before you perform one of the other operations, to avoid destroying valuable data.

Description Maintenance allows you to amend the Volume Description associated with a hard disk domain or sub unit.

To exit from the $V command back to the ready prompt or main menu, key <CR> or <ESCAPE>.

Note that some of the $V functions require exclusive access to a domain. On a multi-user SYSTEM the Swap File ($$SWAP*xx* on logical unit $SW) is permanently "IN USE" unless the $STATUS GET instruction is used to gain exclusive control (see Chapter 5 of the Global System Manager Manual). The user File ($$USER on logical unit $M, usually A01) is permanently "SHARED". In order to perform some of the above $V functions on SYSTEM A, it is necessary to reload the Starter System from the BACRES diskette (see section 6.3.2).

There is some overlap between the $F and $V system commands. For example, both the $F INI instruction and option 1 from the $V menu can be used to initialise diskettes. Similarly, both the $F DES instruction and option 11 from the $V menu can be used to amend Volume Descriptions. $F is described in Chapter 5 of the Global System Manager Manual.

4.10.1 Preparing New Diskettes

Only rarely are new diskettes already formatted correctly, needing simply to be initialized. Usually, factory-fresh diskettes will need to be formatted. In particular, you should format a diskette if you are re-using one which has developed a bad sector. To do this, use the appropriate Unix command (e.g. *format*(C) for SCO Unix) to format the diskette using the same Unix device name defined in the Systems file (see section 7.3.3) for the diskette type you are about to initialise.

If you choose to verify the diskette, using function 2, it will be re-read, track-by-track, once it has been initialised, to check that every sector is readable.

When the unit prompt ($78 UNIT:) appears you should mount the diskette to be processed and then key the unit address (or unit-id) of the drive the new diskette occupies, together with its volume-id (which can be up to 6 characters).

While the diskette is being initialised and verified, messages indicate the progress of the operation track by track. When both processes have finished the unit prompt re-appears so that you can create another volume if you want, or return to the $V menu by keying <CR>. If you want to create several volumes with the same name (e.g. several backup volumes), you can reply <CR> to the volume prompt to use the volume-id you last specified.

NOTE THAT DISKETTE FORMATTING IS NOT SUPPORTED ON GLOBAL SYSTEM MANAGER (UNIX) CONFIGURATIONS.

4.10.2 Re-using Old Diskettes

You proceed in much the same way when you wish to re-use an old diskette. The only difference is that Global System Manager detects that there is already a valid volume present, and prompts to make sure that you really do intend to destroy the data it currently contains (which, of course, will be the effect of an initialize operation).

For example, it is possible to initialise and verify a diskette called WORK1, which was previously initialized as SAPROG. The special reply of "?" to the destroy prompt allows you to obtain a listing of the files occupying the volume to see whether it really should be overwritten. You should reply N to the prompt if you do not want to proceed.

On a configuration which includes several diskette formats (as displayed by the $U command), it can take a considerable time for $V to check that the target diskette does not contain a valid Global System Manager directory. $V attempts to read the directory once for each diskette format that is supported on the selected drive. For example, if the configuration file includes the following formats on drive-0:

a50 G1A
a90 C10A
a96 C24A

$V will attempt to read the diskette using the volume format parameters for G1A, C10A and C24A. For example:

```
$78 UNIT:a50 VOLUME ID:TEST
;
; $V normally attempts to read the diskette to check for
; a valid Global System Manager directory.
;
$78 DESTROY DISK-1 ON a96?:Y
```

This exhaustive checking can be bypassed by terminating the reply to the unit number prompt with <CTRL A> instead of the normal <CR>. For example:

```
$78 UNIT:a50<CTRL A> VOLUME ID:TEST
$78 CONFIRM CORRECT DISK MOUNTED ON 1ST DISKETTE DRIVE (A:) - a50?:Y
```

DO NOT USE THIS OPTION UNLESS YOU ARE ABSOLUTELY SURE THAT THE CORRECT DISKETTE IS MOUNTED IN THE SELECTED DRIVE.

4.10.3 Preparing Data File Domains

You normally use function 1, Initialize, to prepare a new data file domain, since verification is usually not needed. You will have to set up a number of units on the disk starting with the lowest unit number, the domain unit. To find the unit addresses involved consult the unit description report produced by the $U system command (see section 4.9).

The following example assumes that you want to set up three volumes named XXWORK, XXDATA and XXPROG on subunits B01, B02 and B03 of the data volume domain covered by unit B00. You must first establish the domain itself. You decide to name it XXDOM. The dialogue is the same as for initializing a diskette:

```
Please select a function:$V
........
........ (Menu. Function 1 is Initialize)
........
PLEASE SELECT A FUNCTION:1
$78 UNIT:B00 VOLUME:XXDOM INITIALISED
$78 UNIT:
```

Now you must *allocate* the three volumes. You have to specify how large they are so that Global System Manager can provide the correct amount of space for them from the Unix filing system.

The following example dialogue will allocate 2 Mbytes for XXWORK, 250 Kbytes for XXDATA, and 128000 bytes for XXPROG (where Mbyte = 1 Megabyte, 1048576 bytes; Kbyte = 1 Kilobyte, 1024 bytes.):

```
$78 UNIT:B01 VOLUME DESCRIPTION:EXAMPLE 2M WORK VOLUME
VOLUME:XXWORK SIZE:2M 2098176 INITIALISED
$78 UNIT:B02 VOLUME DESCRIPTION:EXAMPLE 250K DATA VOLUME
VOLUME:XXDATA SIZE:250K 257024 INITIALISED
$78 UNIT:B03 VOLUME DESCRIPTION:EXAMPLE PROGRAM VOLUME
VOLUME:XXPROG SIZE:128000 129024 INITIALISED
$78 UNIT:<CR>
........
........ (Menu)
........
Please select a function:
```

The volumes are now set up and can be used like any other data volumes. The value displayed before the word INITIALISED is the size actually allocated for the volume - usually slightly larger than the size requested, since each volume will occupy a whole number of virtual tracks.

Volume Descriptions are described in section 4.10.10.

Note that you can key <CR> to the SIZE: prompt, in which case the size last specified will be used. The special reply of 0 (zero) causes the maximum amount of contiguous space remaining on the Integrated Data File domain (e.g. volume format P259Z) to be allocated to the volume. **A SIZE OF 0 IS NOT ALLOWED FOR A DISCRETE DATA FILE DOMAIN (e.g. volume format T151Z).**

When allocating a new subvolume on a Discrete Data File domain, the number of each logical track is displayed as it is being initialised. For example:

```
Allocating track nnnn INITIALISED
```

If the subvolume initialisation does not complete successfully (e.g. if <CTRL W> is used to abort $V) a dummy Unix subvolume data file may exist in the domain directory (e.g. A00.dir) occupying un-reusable hard-disk space. To reclaim the space, delete this file using the Unix *rm* command.

4.10.4 Allocating Additional Volumes

You can use $V to set up new volumes, providing that some remain to be allocated and there is spare data space available. You can check the state of the domain either by using the List Directory function on the $V menu or by keying its address in response to the unit prompt, and ? to the subsequent volume or volume description prompts. The resulting display shows the units and volumes already allocated, the number of subunits available, and the amount of free space remaining.

The allocation of a new volume takes place just as explained in the previous section. For example, to set up XXNEW with 1.5 Mbytes of storage:

```
$78 UNIT:B04 VOLUME DESCRIPTION:2ND WORK VOLUME (1.5MB)
VOLUME:XXNEW SIZE:1.5M 1573888 INITIALISED
$78 UNIT:
etc, etc.
```

If you mistakenly key the address of a volume which has already been allocated the destroy prompt appears. You should key N to return to the unit prompt:

```
$78 UNIT:B03 VOLUME DESCRIPTION:2ND WORK VOLUME (1.5MB)
```

```
VOLUME:XXNEW SIZE:1.5m
$78 DESTROY XXPROG?:N
$78 UNIT:
```

Note that you can use the ? response to the VOLUME DESCRIPTION or VOLUME prompts to examine the contents of a diskette or hard disk volume, as well as a domain. The display produced is the same as that which appears when you key ? in response to the destroy prompt.

If you want to allocate an additional spool unit you must prefix the size with an S to indicate its status. You will then be prompted for the "allocation". You would normally reply <CR> to accept the default shown. For example:

```
$78 UNIT:B10 VOLUME DESCRIPTION:SECONDARY SPOOL UNIT
VOLUME ID:SPOOL2 SIZE:S3M
$78 ALLOCATION ( 61K):<CR> 3150336 INITIALISED
```

4.10.5 Deallocating a Hard Disk Volume

To deallocate a volume previously allocated use the initialize function as above, but reply <CTRL A> to the volume description or volume prompts. The volume will erase all the data on it. Key Y to continue, or N or <CR> to avoid deallocating the volume and losing data you want to keep. For example:

```
$78 UNIT:B03 VOLUME DESCRIPTION:<CTRL A>
$78 DEALLOCATE XXNEW?:Y DEALLOCATED
$78 UNIT:
```

Note that you can use the ? response to the DEALLOCATE prompt to examine the contents of a hard disk volume as a final check before deallocating it. The display produced is the same as that which appears when you key ? in response to the destroy prompt.

4.10.6 Recovering a Faulty Track on a Hard Disk

This option is not available for Global System Manager (Unix) configurations.

4.10.7 Preparing Work Volumes

Many software modules need to use special volumes, known as *work volumes*, when reorganizing data etc. These volumes, which can be held on any disk of sufficient capacity, have names of the form *xx*WORK, where *xx* represents the 2 character abbreviated form of the program name. Although most Global software modules automatically set up such volumes when required, in some cases you will be required to do this manually.

Work volumes are usually named $$WORK. The $ characters have the special property that they will match the request for any two other characters. This means that this volume can be used as a work volume by any Global software module without having to be renamed. For example, a Sales Ledger module might require a work volume called SLWORK. If the unit assignment in its menu entry tells it to look for this volume at the $$WORK unit address it will not signal an error when it finds the volume $$WORK rather than SLWORK.

If you want to allocate a work unit you must prefix the size with a W to indicate its status as a work unit. You will then be prompted for the "allocation". You would normally reply <CR> to accept the default shown. For example:

```
$78 UNIT:B10 VOLUME DESCRIPTION:A WORK UNIT
VOLUME ID:$$WORK SIZE:W3M
```

```
$78 ALLOCATION ( 61K):<CR> 3150336 INITIALISED
```

4.10.8 Amending a Domain Error Map

This option is not available for Global System Manager (Unix) configurations.

4.10.9 Listing a Directory

The $V List function displays a volume or domain listing in the same format as the listing produced by the $F LIS instruction.

The format of a domain listing produced by the List function in versions 8.0, and later, of Global System Manager differs from that produced by earlier versions. The Volume Description (see section 4.10.10) of each subvolume is displayed instead of the "start, size and protection" information. To produce a domain listing which includes the "start, size and protection" information for each subunit, terminate the unit number with <CTRL A> instead of the usual <CR>.

For example:

```
$78 UNIT:600
VOLUME SYSDOM ON 600 ACCESS OPTION 1 11/12/90 11.53.49
FIRST DOMAIN (GLOBAL SYSTEM MANAGER DEVELOPMENT)
UNIT  VOL-ID      DATASIZE    DESCRIPTION
601   SYSRES      3150336     V8.0 GLOBAL SYSTEM MANAGER VOLUME
602   SYSDEV      1577472     V8.0 GLOBAL COBOL PROGRAMMING KIT
603   PSDATA      2101760     V8.0 GLOBAL COBOL PRODUCT SUPPORT
604   SYSLOG      791040      V8.0 SYSTEM EVENT LOGGING VOLUME
etc.

$78 UNIT:600<CTRL A>
VOLUME SYSDOM ON 600 ACCESS OPTION 1 11/12/90 11.57.14
FIRST DOMAIN (GLOBAL SYSTEM MANAGER DEVELOPMENT)
UNIT  VOL-ID      DATASIZE    TOTAL       START
601   SYSRES      3150336     3162112     49152
602   SYSDEV      1577472     1589248     3211264
603   PSDATA      2101760     2113536     4800512 PROTECTED
604   SYSLOG      791040      802816      6914048
etc.
```

4.10.10 Volume Description Maintenance

Global System Manager V7.0, and later, allows a 50 character *Volume Description* to be associated with each data file domain and subvolume. An area of the index header (see section 4.10.11) is utilised to hold the table of Volume Descriptions.

The Description Maintenance function 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 has 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. For example:

```
Please select a function :11
```

```
$78 UNIT:610 DESCRIPTION:AN EXAMPLE VOLUME DESCRIPTION
$78 UNIT:699 NO ROOM FOR DESCRIPTION - RE-INITIALIZE DOMAIN
```

The behaviour of the $V Initialise function(s) may also depend on the volume number for domains that have been initialised prior to Global System Manager V7.0:

```
$78 UNIT:610 VOLUME DESCRIPTION:EXAMPLE VOLUME DESCRIPTION
VOLUME ID:TEST1 SIZE:1M 1053184 INITIALISED
$78 UNIT:699 VOLUME ID:TEST2 SIZE:1M 1053184 INITIALISED
```

Note that if the $V Initialise function is used under Job Management and there is insufficient space in the volume description table for a particular subunit the VOLUME DESCRIPTION prompt always appears, although the reply is subsequently ignored. This is necessary to ensure a predictable and consistent Job Management dialogue.

The Initialise function contains another special feature which is only enabled when $V is used under Job Management. If the reply to the VOLUME DESCRIPTION prompt is six characters or less and $V is executing under Job Management, the reply (supplied by the job) is used as the volume-id and the subsequent VOLUME prompt, which normally appears, is suppressed. This feature has been incorporated into $V to ensure a compatible dialogue for jobs developed prior to Global System Manager V7.0.

Note also that Volume Descriptions are only recognised by versions 7.0, and later, of Global System Manager. Any program generating job dialogue must check the Global System Manager version before attempting to amend a Volume Description.

Finally, the first example shown in section 4.10.3 over-simplified the use of $V when initialising domains. If a domain is being initialised for the first time, $V prompts for the volume i.d. only. If however, a domain is being re-initialised, $V prompts for both a volume i.d. and a volume description. For example:

```
Please select a function :1

$78 UNIT:250 VOLUME ID:DISK-2 INITIALISED
$78 UNIT:250 VOLUME DESCRIPTION:2ND 2.5GB HARD DISK
VOLUME ID:BIGDSK
$78 DESTROY DISK-2 ?:
```

Important note: The Volume Descriptions for Discrete Data File domains (e.g. volume format T151Z) are held in the domain header file rather than being associated with individual subvolume data files. If a subvolume data file is copied to a data directory using an Unix command (e.g. cp) it will automatically inherit the Volume Description, if any, associated with the corresponding subvolume number. The same effect will occur if a subvolume data file is "renumbered" using the Unix *mv* command.

4.10.11 Global System Manager (Unix) Data File Format

A data file domain is a Unix directory specified by the corresponding Systems file definition (see section 7.3.2), within which the subvolumes are created. For example:

```
SYSTEM A                                    (within the Systems file)
DATA        0     A00.dir
```

The unit addresses for subvolumes of this data file are A00 upwards. The name of the directory is free format, the example above shows the suggested convention only. The unit

addresses A*nn* are actually determined by the data definitions in the configuration file. Each subvolume of associated data is contained within a separate file, the internal structure of the file being in Global format. The name of each file has a fixed part and a variable part. For example, the contents of the data file directory created in section 4.10.3 would look like this:

```
$ ls B00.dir
 SVL00_XXDOM
 SVL01_XXWORK
 SVL02_XXDATA
 SVL03_XXPROG
 $
```

The prefix SVL is mandatory and is used to recognise Global data files. The number (00 to 03 here) gives the unit address offset from the domain unit address. The underscore _ is again mandatory. The rest of the file name corresponds to the name specified for the unit by $V or $F.

Note that there is a file with the same unit address as the data file domain itself (SVL00_XXDOM above). This is the index header file which MUST exist. It contains initialisation data and the long volume descriptions displayed by $F.

When Global System Manager is NOT in use it is possible to rename or copy these subvolumes using standard shell commands. The new unit address and volume name will be used the next time Global System Manager is run. The unit addresses must be unique otherwise the later volume will be ignored (and it may not be clear which file is ignored). Similarly subvolumes can be copied or moved to different data file domains (or archived), but the same rules apply. If a subvolume is given a different unit address, however, the long volume description is not moved to the new address, as this is stored in the index header SVL00_*vvvvv* (SVL00_XXDOM above).

5. Unix Specific Utilities and System Requests

In this chapter we give detailed accounts of some important Global System Manager commands and system requests that are only available when the host operating system is Unix. These commands and system requests are not described in the Global System Manager Manual.

5.1 %.SHCMD and %.SHCMDR - Execute Shell Command From Keyboard

The %.SHCMD utility executes a Unix shell command specified from text typed in from the keyboard. The command is executed once <CR> is keyed to terminate the command string. For example:

```
Please select a function :%.SHCMD
Shell command:vi $GLDIR/sys/Systems
```

This example command enters the Unix editor *vi* and edits the Global System Manager Systems file.

During the period the Unix command is executing, Global System Manager displays are suspended in the currently active partition, but all background partitions continue processing. Once the Unix shell command terminates, in this example by exiting from the vi editor, the following prompt is issued:

```
[Key <CR> to continue]
```

When <CR> is keyed Global System Manager displays continue as normal.

If the terminator to the shell command string is <CTRL A>, rather than <CR>, the "[Key <CR> to continue]" prompt is suppressed so that Global System Manager continues as normal without any operator intervention immediately after the termination of the Unix shell command.

The maximum command string length is currently 128 characters.

The %.SHCMDR utility is similar to %.SHCMD expect that when <CR> is keyed to the following prompt:

```
[Key <CR> to continue]
```

the screen is refreshed with the currently active partition, thus overwriting any information displayed by the Unix command. After the screen refresh, Global System Manager displays continue as normal. As for %.SHCMD, if the terminator to the shell command string is <CTRL A>, rather than <CR>, the "[Key <CR> to continue]" prompt is suppressed so that the screen is refreshed without any operator intervention immediately after the termination of the Unix shell command.

The %.SHCMD (or %.SHCMDR) utility, and the Unix command string, may be executed from the type-ahead dialogue of a Global System Manager menu (see Chapter 7 of the Global System Manager Manual) or from job management dialogue (see the Global Development Job Management Manual).

The %.SHCMD and %.SHCMDR commands uses the SVC 70 interface described in Appendix H.

5.2 %.SHELL - Execute Shell Command

The %.SHELL utility executes a Unix shell command. Like the %.SHCMD utility (see section 5.1) it accepts type-ahead characters but unlike %.SHCMD it doesn't display a prompt, nor does it echo any characters. The Unix shell command string is terminated by the <ESC> character hex 0x1B (or the <ESCAPE> key). When %.SHELL detects the <ESC> character in the text string, the Unix command is executed.

Thus, use of %.SHELL allows <CR> characters to be included in the Unix command string. However, this particular feature is expected to be of limited use because the Unix shell will probably (mis)interpret the <CR> characters. If multiple commands are to be executed by the shell, they should be separated by semi-colons. For example, to list directories $GLDIR/sys and $GLDIR/bin to Unix files /tmp/dir1 and /tmp/dir2 with a single invocation of %.SHELL use the following command string, terminated with an <ESC> character:

```
ls $GLDIR/sys>/tmp/dir1;ls $GLDIR/bin>/tmp/dir2
```

During the period the Unix command is executing, Global System Manager displays are suspended in the currently active partition, but all background partitions continue processing. Once the Unix shell command terminates the screen is refreshed with the currently active partition and displays continue as normal.

The maximum command string length is currently 128 characters.

The %.SHELL utility, and the Unix command string, may be executed from the type-ahead dialogue of a Global System Manager menu (see Chapter 7 of the Global System Manager Manual). Because %.SHELL neither echoes nor displays any text, it is well suited to the addition of shell commands or other Unix applications to a Global System Manager menu.

Important note: The %.SHELL utility is not appropriate for running a Unix command specified directly by an operator. If you wish to run a Unix command specified from text typed in from the keyboard the %.SHCMD or %.SHCMDR command (see section 5.1) should be used in preference to %.SHELL.

The %.SHELL command uses the SVC 70 interface described in Appendix H.

5.3 <SYSREQ> S - Specify Screen Reset Sequence

In addition to the System Requests described in Chapter 4 of the Global System Manager Manual, an additional System Request, <SYSREQ> S, is available when the host operating system is Unix. Only supervisors are allowed to use this system request. <SYSREQ> S allows you to define a "Screen Reset Sequence" for your terminal. A reset sequence is often necessary because the Global System Manager Terminal Attribute Programs (TAP's) for some terminals contain "Screen Start Sequences" that program the function keys. Although the re-programmed function keys are suitable for Global applications they are often very unsuitable for other Unix applications.

<SYSREQ> S allows you to define a terminal-specific "Reset Sequence" which is sent to the terminal whenever Global System Manager relinquishes control back to Unix. When Global

System Manager is reloaded the "Start Sequences" are sent thus configuring the terminal for a Global System Manager session.

The "Reset Sequences" are held in a file, $$RS*tttt*, on unit $DP (where *tttt* is the TAP number (see section 7.3.4) of the terminal you are using). When <SYSREQ> S is keyed, the screen is cleared and the following prompt is displayed on the baseline:

```
Program terminal reset sequences - Key C to continue, <ESC> to exit:
```

Because of the serious consequences that may result from accidental misuse of this system request you must key C to continue. Any other reply (e.g. <CR> or <ESC> will abort the system request immediately.

When C has been keyed to continue with the system request, the $TED editor (see Chapter 5 of the Global System Manager Manual) is invoked using $$RSTttt as the name of the text-file to be edited. If a new file is being created, the first line of the text will contain the comment line:

```
* Terminal tttt reset sequences
```

If an existing file is being updated, the first screen of text is displayed.

Using the standard $TED editor commands you can specify the Reset Sequences for your particular terminal.

The sequences must be supplied as hexadecimal strings. Comments, indicated by the "*" character, may be inserted anywhere in the text. For example:

```
* Terminal 163 reset sequences
* The sequences in this file program the function keys
* for the Wyse-50 to a state suitable for use by "Labyrinth"
*
1B7A401B607F * Program <F1> to 1B60
1B7A411B617F * Program <F2> to 1B61
etc.
```

When the edit is complete, the text is validated to ensure all the strings represent valid hexadecimal quantities. If an invalid line is detected, one of the following errors is displayed allowing you to return to the editor:

```
INVALID LENGTH HEX STRING LINE llll Key <CR>:

INVALID HEX STRING LINE llll Key <CR>:
```

Key <CR> to return to the editor, you will be positioned at the same place in the file that you were when the edit completed.

If the text file is valid, the following message is displayed:

```
Key A to abandon, <CR> to update terminal reset sequences
```

Key <CR> to update the terminal reset sequences. The following baseline prompt is displayed:

```
Terminal reset sequences updated - Key <CR>
```

Key <CR> to exit from the system request.

The maximum total length of the Reset Sequences is 1024 bytes (i.e. the same as the maximum total length of the Start Sequences defined in the TAP). If this limit is exceeded, the following error message is displayed:

```
SEQUENCE IS LONGER THAN 1024 BYTES Key <CR>:
```

The maximum length of any line is 70 bytes. Individual escape sequences may be split between lines. For example:

```
* Terminal 163 reset sequences
* The sequences in this file program the function keys
* for the Wyse-50 to a state suitable for use by "Labyrinth"
*
1B          * Program <F1> to 1B60
7A          * but split the
40          * sequence into
1B          * several
607F        * lines!
```

Use the following technique to determine the values of the function keys generated by the terminal in "native mode" (i.e. before the Global System Manager Start Sequences are sent):

- Remove the Systems file terminal mapping (see section 7.3.4) for the particular terminal type.

- Load Global System Manager. You will be presented with the following prompt:

  ```
  PLEASE KEY TERMINAL CODE (nnnn ):
  ```

  A reply of 0 to the Terminal Code prompt will cause Global System Manager to use a TAP that includes basic teletype support only.

- Run option 2 from the $T/DIAG menu to determine the byte values generated by the various function keys. $T, and TAP's in general, are fully described in the Global Cobol Screen Support Manual.

As soon as the function key sequences have been determined, exit from Global System Manager and edit the Systems file to restore the original Terminal Type mapping.

Important note: Do not be confused by the functionality of <SYSREQ> S and <SYSREQ> * (see section 4.31 of the Global System Manager Manual). <SYSREQ> * allows an operator to define a pair of **operator-specific** sequences, referred to as the "in sequence" and "out sequence", which are sent explicitly by the operator by invoking <SYSREQ> *. <SYSREQ> S, on the other hand, allows an operator to define a **terminal-specific** "reset sequence" which is sent automatically each time Global System Manager returns to, or invokes, a Unix shell.

Thus the "reset screen" sequence established using <SYSREQ> S alleviates the problems that can occur when control is passed from Global System Manager to Unix using a internal software switch (e.g. $BYE or <SYSREQ> . - see section 5.4). However, this technique can't possibly cope with the situation that occurs when a screen is swapped between Global System Manager and Unix using an external switch (e.g. a T-switch or switching between multiple hosts on a serial terminal). <SYSREQ> * is available to send a specific sequence before an operator switches between Global System Manager and Unix, for example.

The "start sequences" defined in the TAP and the "reset sequences" defined using <SYSREQ> S are sent when the following events occur:

- When the *global* command (see section 6.3.1) is used to load Global System Manager, the "start sequences" are sent to the terminal;

- When $E (see section 4.3) or $BYE (see section 4.1) are used, the "reset sequences" are sent to the terminal;

- When <SYSREQ> . (see section 5.4) is used to create a new shell, the "reset sequences" are sent to the terminal before the new shell is opened. When the new shell is closed (e.g. by keying <CTRL D>) and control returns to Global System Manager the "start sequences" are sent to the terminal;

- When SVC 70 is invoked (see Appendix H), for example, by use of the %.SHCMD (see section 5.1) or %.SHELL (see section 5.2) commands, the "reset sequences" are sent to the terminal before the new shell is opened. When the new shell is closed and control returns to Global System Manager the "start sequences" are sent to the terminal.

5.4 <SYSREQ> . - Create New Unix Shell

Keying <SYSREQ> . at any time while running Global System Manager causes the Global application in the current partition to suspend whilst a new Unix shell is created.

The Unix shell prompt should appear at the beginning of the next line below the cursor (unless you have redefined the SHELL variable (see section 8.6.1) to execute another Unix command).

Exiting from this shell will redisplay the Global System Manager partition.

Section 8.6.1 describes a technique that can be used to prevent an operator from using <SYSREQ> . to create a new Unix shell.

Note that you are not allowed to run Global System Manager from the new Unix shell. This restriction has been imposed to prevent a user from nesting multiple copies of Global System Manager.

6. Unix Utilities

This chapter describes the Unix utilities distributed with Global System Manager (Unix) on the BACNAT diskette or tape. In addition to explaining the various utilities, this chapter also describes the Unix directory hierarchy of a Global System Manager installation.

6.1 Global System Manager Unix File and Directory Organisation

Global System Manager (Unix) is installed into a Unix directory called *global* (see Figure 6.1). The following sections describe the contents of the various sub-directories within the global directory hierarchy (see section G.12 for a more detailed description of the global directory). In order to use Global System Manager, the GLDIR Unix shell variable must be set to the pathname of the global directory (see section 8.1.1).

6.1.1 The $GLDIR/sys directory

The $GLDIR/sys directory contains most of the Global System Manager customisation and authorisation files.

6.1.1.1 The $GLDIR/sys/Systems file

This text file contains the user authorisations, data file descriptions, printer device descriptions and various other configuration parameters. See Chapter 7 and Appendix F for a full description of the Systems file. If Global System Manager has been re-installed, backup copies of the "live" Systems file, renamed as Systems.old and Systmp.*nnnn* (where *nnnn* is a numeric string, actually the process-id of the glinstall that created the file) may also exist in the $GLDIR/sys directory.

6.1.1.2 The $GLDIR/sys/machine file

This text file contains the computer type and Unix operating system version on which the installed Global System Manager will operate with. This file actually contains the computer type and Unix operating system on which the BACNAT software was produced.

6.1.1.3 The $GLDIR/sys/messages file

This text file contains a log of significant Global System Manager events. Normally, each invocation of Global System Manager will write several records to the messages file so its size will rapidly increase. Use of the global -m command line argument (see section 6.3.3.3) will prevent Global System Manager writing to the messages file. If the messages file becomes too large it may be deleted (the next invocation of Global System Manager will create a new one). **The $GLDIR/sys/messages file is the only file that can be deleted from the $GLDIR/sys directory.**

6.1.1.4 The $GLDIR/sys/version file

This text file contains the version number for every software module within the Global System Manager (Unix) nucleus. The version (*sic*) file also has the BACNAT variant number near the beginning of the file (see section G.1). **THIS VARIANT NUMBER SHOULD ALWAYS BE QUOTED WHEN REPORTING PROBLEMS TO THE SERVICE CENTRE.**

6.1.1.5 The $GLDIR/sys/variant file

This text file contains a single line which just contains the BACNAT variant number (see section G.1). **THIS VARIANT NUMBER SHOULD ALWAYS BE QUOTED WHEN REPORTING PROBLEMS TO THE SERVICE CENTRE.**

6.1.1.6 The $GLDIR/sys/data directory

This directory contains the compiled Systems file data produced by running the glconfig utility (see section 6.5). The various data files within this directory, the formats of which are beyond the scope of this manual, may be inspected using the glsysdump utility (see section 6.8).

6.1.1.7 The $GLDIR/sys/errmsg directory

All the error messages displayed by the Global System Manager (Unix) nucleus (see Appendix C) are held in two text files within this directory. The $GLDIR/sys/errmsg/txt file contains the main text of each message. The $GLDIR/sys/errmsg/hdr file contains the message header text.

6.1.1.8 The $GLDIR/sys/spool directory

This directory, which is empty when Global System Manager is installed, contains temporary files created by the SPOOLED printer processes (see Appendix E). **DO NOT DELETE THIS DIRECTORY**.

6.1.1.8.1 The $GLDIR/sys/spool/queue file

This file is used to pass spooled printer requests between the *glintd* process and the *glspod* daemon (see section G.103.2).

6.1.1.8.2 The $GLDIR/sys/spool/log file

This file is used to log significant events (e.g. failed spooled print commands) that occur while the *glspod* spooled printer daemon is running (see section G.103.3).

6.1.2 The $GLDIR/bin directory

This directory contains the Global System Manager binary executables and daemon processes. This directory MUST form part of the Unix PATH shell variable for any user who wishes to run Global System Manager or any of the Global range of software (see section 8.1.2). The Unix utilities, that may be run from the Unix shell prompt, are described in sections 6.2 to 6.11 of this chapter). The daemon processes, indicated by a * in Figure 6.1, are described in Appendix D.

6.1.3 The $GLDIR/data directory

This directory contains, by default, the Global data files required by Global System Manager and the Global applications. Note that there is no absolute requirement to hold the Global data files within this directory. They can reside anywhere within the filing system as their locations are defined within the Systems file (see section 7.3.2).

Each "data file" consists of a directory (e.g. A00.dir). Within that directory there is an index file called SVL00_*xxxxxx* and data volumes (or subvolumes) called SVL*nn*_*xxxxxx* where *nn* is a number between 01 and the number of subvolumes allowed for this data file (usually 99), and *xxxxxx* is the name of the subvolume (see section 4.10.11).

6.1.4 The $GLDIR/spool directory

This directory, which is empty when Global System Manager is installed, is the default directory for print files created by the Global System Manager. Please refer to Appendix E for further information. **DO NOT DELETE THIS DIRECTORY**.

6.1.5 The $GLDIR/tmp directory

This directory, which is empty when Global System Manager is installed, is used to hold temporary files and FIFO's that are required when Global System Manager is running (see Appendix D). **DO NOT DELETE THIS DIRECTORY**.

```
glinstall                                          (section 6.2)
global\sys\
global\sys\Systems
global\sys\Systems.old                             (optional)
global\sys\Systmp.nnnn                             (optional)
global\sys\machine
global\sys\messages
global\sys\version
global\sys\variant
global\sys\data\
global\sys\data\screen
global\sys\data\shm
global\sys\data\shmsiz
global\sys\data\tmap
global\sys\data\tty
global\sys\data\umap
global\sys\data\user
global\sys\errmsg\
global\sys\errmsg\hdr
global\sys\errmsg\txt
global\sys\spool\                                  (installed empty directory)
global\bin\
global\bin\global                                  (section 6.3)
global\bin\gl                                      (section 6.3)
global\bin\GLOBAL                                  (section 6.3)
global\bin\GL                                      (section 6.3)
global\bin\gladmin                                 (section 6.4)
global\bin\GLADMIN                                 (section 6.4)
global\bin\glconfig                                (section 6.5)
global\bin\glmkdat                                 (section 6.6)
global\bin\glclean                                 (section 6.7)
global\bin\glintd*                                 (Appendix D)
global\bin\glprid*                                 (Appendix D)
global\bin\glspad*                                 (Appendix D)
global\bin\glspod*                                 (Appendix D)
global\bin\gltimd*                                 (Appendix D)
global\bin\glsysdump                               (section 6.8)
global\bin\glshmdump                               (section 6.9)
global\bin\glreorg                                 (section 6.10)
global\bin\glinfo                                  (section 6.11)
global\data\
global\data\A00.dir\
global\data\A00.dir\SVL00_SYSDOM
global\data\A60.dir\                                (tape distribution only)
global\data\A60.dir\SVL00_DISTBN
global\data\A60.dir\SVL01_BACRES
global\data\A60.dir\SVL02_BEA
global\data\A60.dir\SVL03_HAA
global\data\A60.dir\SVLnn_CFA
global\data\A60.dir\SVLnn_EPA                      (optional)
global\data\A60.dir\SVLnn_xxA                      (optional)
global\spool\                                      (installed empty directory)
global\tmp\                                        (installed empty directory)
global\pkg\
global\pkg\gpprod
```

Figure 6.1 - global directory structure

6.1.6 The $GLDIR/pkg directory

This directory contains the various parameter files that were used when the software was generated, the contents of which are beyond the scope of this manual.

6.2 glinstall - Install Global System Manager

This glinstall installation script is used for a variety of purposes:

- to install Global System Manager for the first time. See section 6.2.1);
- to upgrade the Global System Manager (Unix) "nucleus" software (i.e. the BACNAT software). See section 6.2.2;
- to upgrade the Global System Manager (Unix) "nucleus" software AND the version of Global System Manager from tape. See section 6.2.3;
- to reset the file permissions and ownerships of the files in the *global* directory. See section 6.2.4;
- to upgrade the Global System Manager (Unix) "nucleus" software AND restore the Global System Manager starter system (i.e. BACRES, BEA etc.) from tape before upgrading the Global System Manager revision. See section 6.2.5.

The installation script can only be invoked by super-user (root).

The directory into which Global System Manager is going to be installed (e.g. /usr/gsm) is not normally part of the PATH shell variable (see section 8.1.2) so it is normally necessary to change to the relevant directory (e.g. /usr/gsm) and prefix the glinstall command name with a "./". For example:

```
# cd /usr/gsm
# ./glinstall
```

The following glinstall command line options are available:

glinstall	perform fresh installation from the default installation device (see section 6.2.1);
glinstall -b	upgrade Global System Manager from the default tape device (see section 6.2.3);
glinstall -c *devname*	upgrade Global System Manager from the *devname* tape device (see section 6.2.3);
glinstall -d *devname*	perform fresh installation from the *devname* device (see section 6.2.1);
glinstall -f	perform fresh installation from the default (floppy) diskette device (see section 6.2.1);
glinstall -h	display help information;
glinstall -p	reset ownerships and permissions of global files without extracting from BACNAT device (see section 6.2.4);

glinstall -r	restore Global System Manager from the default tape device before upgrading the Global System Manager revision (see section 6.2.5);
glinstall -s *devname*	restore Global System Manager from the *devname* tape device before upgrading the Global System Manager revision (see section 6.2.5);
glinstall -t	perform fresh installation from the default tape device (see section 6.2.1);
glinstall -u	update BACNAT software from the default device (see section 6.2.2);
glinstall -v *devname*	update BACNAT software from the *devname* device (see section 6.2.2).

The command line options listed above are mutually exclusive. Only one glinstall option can be specified.

6.2.1 Installing Global System Manager

The installation of Global System Manager is started by invoking glinstall:

```
# ./glinstall
```

If glinstall is executed with no command line arguments, it will attempt a fresh installation from the default device. The default device is a diskette for "diskette configurations" or a tape for "tape configurations". Please consult your Global Configuration Notes for further details. To override the default installation device class (i.e. diskette or tape) use one of the following command line arguments:

-f	install Global System Manager from the default (floppy) diskette device;
-t	install Global System Manager from the default tape device.

The **exact** Unix device name of the installation device (i.e. floppy diskette, tape or even a "BACNAT image" file, created from a BACNAT diskette or tape using the Unix cp command) can be specified by the following command line argument:

-d *devname*	where *devname* is full device name of the installation device (e.g. /dev/rfd0135ds18 or /dev/rmt/ctape0).

When operating in "fresh installation mode" glinstall prompts for the following information:

- confirmation that the installation mode is correct;
- the directory into which Global System Manager is to be installed;
- confirmation that the BACNAT volume is correctly mounted.

and performs the following functions:

- checks that a *global* group and a *global* user exist on the Unix system (see section 2.1.3). This test is performed by attempting to chown and chgrp a temporary file, created for this purpose (note the earlier versions of glinstall searched for user *global* and group *global* in the /etc/passwd and /etc/group files, respectively);

- ensures that Global System Manager is not running by testing global/bin/glintd, global/bin/gltimd, global/bin/glspad and global/bin/glspod to check they are all writable files;

- extracts the contents of the global directory from the installation device, using the Unix tar command. **This operation will overwrite the contents of an existing Global System Manager**;

- sets the Unix file permissions and ownerships (see section G.12);

- invokes the global command using the -i (install) option (see section 6.3.2).

6.2.2 Upgrading the BACNAT Software

The glinstall script can also be employed to upgrade the variant of the BACNAT software, using either of the following command line arguments:

-u	upgrade the BACNAT software from the default installation device (i.e. diskette or tape);
-v *devname*	upgrade the BACNAT software from *devname* where *devname* is full device name of the installation device (e.g. /dev/rfd0135ds18 or /dev/rmt/ctape0).

When operating in "BACNAT upgrade mode" glinstall prompts for the following information:

- confirmation that the installation mode is correct;

- the directory into which Global System Manager is already installed;

- confirmation that the BACNAT volume is correctly mounted;

- an option to restore and recompile the original Systems file.

and performs the following functions:

- copies the existing $GLDIR/sys/Systems file (see section 6.1.1.1) to both Systmp.old AND Systems.*nnnn* (where *nnnn* is the Unix process-id of the executing glinstall script);

- ensures that Global System Manager is not running by testing global/bin/glintd, global/bin/gltimd, global/bin/glspad and global/bin/glspod to check they are all writable files;

- extracts the contents of the global/sys and global/bin directories from the installation device, using the Unix tar command. In addition, to upgrading the binary executable files, this extraction process also restores the default Systems files and data files within the $GLDIR/sys/data directory;

- sets the Unix file permissions and ownerships of the $GLDIR/sys and $GLDIR/bin directories (see section G.12);

- optionally, renames $GLDIR/sys/Systmp.old to $GLDIR/sys/Systems and invokes the glconfig utility to recompile the Systems file (see section 6.5).

6.2.3 Upgrading the Global System Manager Version

In order to upgrade the version of Global System Manager (e.g. to upgrade from V8.0 to V8.1) it is normally necessary to upgrade both the BACNAT software AND perform a re-installation of Global System Manager. In general, a new variant of the BACNAT software will accompany a new version, or revision, of Global System Manager.

If the technique described in section G.46 has been used to move the DISKETTE definitions from SYSTEM A the Systems file must modified to ensure that a DISKETTE definition is included for SYSTEM A (see section G.59).

6.2.3.1 Upgrading Global System Manager from Diskette

Upgrading Global System Manager from diskette is performed in two stages:

- upgrade the BACNAT software using glinstall as described in section 6.2.2;

- re-install Global System Manager as described in section 6.3.2.

6.2.3.2 Upgrading Global System Manager from Tape

Although upgrading Global System Manager from diskette is relatively easy (see section 6.2.3.1), upgrading from tape is relatively complicated because the BACRES, BEA, HAA etc. volumes must be extracted from the distribution tape. glinstall recognises the following command line arguments to effect a Global System Manager upgrade from a distribution tape:

-b	upgrade the BACNAT software and re-install Global System Manager from the default installation tape;
-c *devname*	upgrade the BACNAT software and re-install Global System Manager from *devname* where *devname* is full device name of the installation tape (e.g. /dev/rmt/ctape0).

When operating in "Global System Manager version upgrade mode" glinstall prompts for the following information:

- confirmation that the installation mode is correct;

- the directory into which Global System Manager is already installed;

- confirmation that the BACNAT tape is correctly mounted.

and performs the following functions:

- copies the existing $GLDIR/sys/Systems file (see section 6.1.1.1) to both Systmp.old AND Systems.*nnnn* (where *nnnn* is the Unix process-id of the executing glinstall script);

- ensures that Global System Manager is not running by testing global/bin/glintd, global/bin/gltimd, global/bin/glspad and global/bin/glspod to check they are all writable files;

- extracts the contents of the global/sys and global/bin directories from the installation device, using the Unix tar command. In addition, to upgrading the binary executable files, this extraction process also restores the default Systems file and data files within the $GLDIR/sys/data directory;

- extracts the contents of the global/data/A60.dir directory from the installation device, using the Unix tar command. Before extracting the A60.dir directory from the tape, the current A60.dir is renamed to A60.old (if an A60.old directory is already present, it is renamed to A60.$$, where $$ is the process id of the executing glinstall script - essentially a random number). This rename is necessary to prevent possible multiple copies of subvolume files appearing the A60.dir directory (see section G.62);

- sets the Unix file permissions and ownerships of the $GLDIR/sys and $GLDIR/bin directories (see section G.12);

- renames $GLDIR/sys/Systmp.old to $GLDIR/sys/Systems and invokes the glconfig utility to recompile the Systems file (see section 6.5). Unlike the options described in section 6.2.2, this rename and recompilation stage is automatic;

- invokes the global command using the -i (install) option (see section 6.3.2).

6.2.4 Resetting Global System Manager File Ownerships/Permissions

In order for Global System Manager (Unix) to function correctly, the file ownerships and permissions created by glinstall (see section 6.2.1) when Global System Manager is installed **MUST NOT BE ALTERED IN ANY WAY**. If the global directory (see Figure 6.1) is moved from one filing system to another, or from one Unix computer to another, the transfer utility (e.g. cpio, tar etc.) may not restore the original ownerships and permissions. Although the original ownerships and permissions can be restored using the Unix *chown*, *chgrp* and *chown* commands, this process is tedious and error-prone.

The glinstall script recognises the following command line arguments to automatically reset file ownerships and permissions of all files and directories within the global directory:

-p	reset file ownerships and permissions WITHOUT extracting files from the BACNAT media.

When operating in "Global System Manager upgrade mode" glinstall prompts for the following information:

- confirmation that the installation mode is correct;

- the directory into which Global System Manager is installed;

and performs the following functions:

- sets the Unix file permissions and ownerships of the $GLDIR/sys and $GLDIR/bin directories (see section G.12).

6.2.5 Upgrading the Global System Manager Revision

In order to upgrade the revision of Global System Manager V8.1 (e.g. to upgrade from GSM V8.1 revision V8.1d to revision V8.1e) it is normally necessary to upgrade both the BACNAT software AND use $CUS to upgrade the GSM revision by extracting the new components from the BACRES, BEA, HAA etc. diskettes. It is always prudent to upgrade to the latest variant of the BACNAT software.

6.2.5.1 Upgrading the GSM Revision from Diskette

Upgrading the Global System Manager revision from diskette is performed in two stages:

- upgrade the BACNAT software using glinstall as described in section 6.2.2;
- use $CUS to perform the Update GSM Revision customisation from the BACRES, BEA, HAA etc. diskettes.

6.2.5.2 Upgrading the GSM Revision from Tape

Although upgrading the Global System Manager revision from diskette is relatively easy (see section 6.2.5.1), upgrading from tape is relatively complicated because the BACRES, BEA, HAA etc. volumes must be extracted from the distribution tape. glinstall recognises the following command line arguments to restore the Global System Manager starter system (i.e. BACRES, BEA, HAA etc.) prior to running $CUS:

-r	upgrade the BACNAT software and restore Global System Manager from the default installation tape;
-s *devname*	upgrade the BACNAT software and restore Global System Manager from *devname* where *devname* is full device name of the installation tape (e.g. /dev/rmt/ctape0).

When operating in "Global System Manager revision upgrade mode" glinstall prompts for the following information:

- confirmation that the installation mode is correct;
- the directory into which Global System Manager is already installed;
- confirmation that the BACNAT tape is correctly mounted.

and performs the following functions:

- copies the existing $GLDIR/sys/Systems file (see section 6.1.1.1) to both Systmp.old AND Systems.*nnnn* (where *nnnn* is the Unix process-id of the executing glinstall script);
- ensures that Global System Manager is not running by testing global/bin/glintd, global/bin/gltimd, global/bin/glspad and global/bin/glspod to check they are all writable files;
- extracts the contents of the global/sys and global/bin directories from the installation device, using the Unix tar command. In addition, to upgrading the binary executable files, this extraction process also restores the default Systems file and data files within the $GLDIR/sys/data directory;

- extracts the contents of the global/data/A60.dir directory from the installation device, using the Unix tar command. Before extracting the A60.dir directory from the tape, the current A60.dir is renamed to A60.old (if an A60.old directory is already present, it is renamed to A60.$$, where $$ is the process id of the executing glinstall script - essentially a random number). This rename is necessary to prevent possible multiple copies of subvolume files appearing the A60.dir directory (see section G.62);

- sets the Unix file permissions and ownerships of the $GLDIR/sys and $GLDIR/bin directories (see section G.12);

- renames $GLDIR/sys/Systmp.old to $GLDIR/sys/Systems and invokes the glconfig utility to recompile the Systems file (see section 6.5). Unlike the options described in section 6.2.2, this rename and recompilation stage is automatic.

6.3 global, gl, GLOBAL, GL - Run Global System Manager

The global utility is used for two purposes. Firstly, global is the program which operators use, on a day-to-day basis, to start a Global System Manager session. global may be included in a login script to cause Global System Manager to be initiated automatically when a particular operator logs in to Unix. Secondly, global is used by the System Administrator to install, or re-install, Global System Manager (see section 2.6.4).

For convenience, the program global is linked to:

gl
GLOBAL
GL

so that any of these commands can be run in place of global.

6.3.1 Running Global System Manager

To invoke Global System Manager run the program global without the -i command line argument (see section 6.3.2):

```
# global
```

On running global you will either be presented with the Global System Manager copyright message followed by (ignoring any password authorisation) the main Global System Manager menu (see Chapter 3); or you will informed that you are not authorised to run Global System Manager for some reason. For example:

```
global: Error 1009 - Authorisation failed. Inform your System Administrator.
```

The first user that runs global will be presented with the following message:

```
Configuring Global System Manager (BACNAT Vn.nnn); please wait
```

or:

```
Configuring Global System Manager (Bnnnn Vn.nnn); please wait
```

The BACNAT variant number should always be quoted when reporting problems to the Service Centre.

6.3.2 Installing Global System Manager

If your Global System Manager SYSRES volume becomes corrupt, and you have to reinstall the software (e.g. because you are unable to restore from a backup), there is no need to reinstall Global System Manager completely. Ensure that the INSDEV command word in the Systems file (see section F.2.8) specifies the correct installation device or data file and run:

```
# global -i
```

This will start the installation of Global System Manager onto SYSRES only. If you are installing from diskette ensure that the BACRES diskette is in the correct diskette drive. If you are installing from tape ensure that the BACRES volume (normally SVL01_BACRES) is in the specified data file directory (normally A60.dir).

No other users must be running Global System Manager during the installation. You can ensure this by removing the execute permissions for "others" from the directory $GLDIR/bin before you install or reinstall (after checking, of course, that no users are currently running Global System Manager):

```
# chmod o-x $GLDIR/bin
```

When installing you are automatically given authorisation for Global System Manager as SYSTEM A. This pseudo-authorisation overrides any authorisations that may be established in the Systems file. It is as though the Systems file includes the following lines (where *logname* is your current logname):

```
SYSTEM      A
DATA        0     A00.dir
USER        1     LOGNAME     logname
```

6.3.3 global Command Line Arguments

In addition to the -i command line argument global recognises several other parameters. The complete list, in alphabetic rather than functional order, is as follows:

-a Re-attach to running glintd (see section 6.3.3.13);

-b Do not execute the background glintd (see section 6.3.3.14);

-c Allows a core dump to be generated (see section 6.3.3.12);

-d display diagnostic information (see section 6.3.3.1);

-e Disables 32 bit option (see section 6.3.3.15);

-f Use when logging on using GX (see section 6.3.3.16);

-F Use when logging on using OneOffice 3000 WorkSpace (see section 6.3.3.17);

-g Leaves glspad running (see section 6.3.3.18);

-h hibernate during File Executive locks (see section 6.3.3.2);

-i install Global System Manager from BACRES (see section 6.3.2);

-j Disables the attempted re-open of the standard TTY device (see section 6.3.3.19);

-k Ignores the results of statfs and always assume that the filing system has 2Gb of free space (see section 6.3.3.20);

-l Turns off most of the writes to the spooler log file (see section 6.3.3.21);

-m Suppress writing messages to the messages file (see section 6.3.3.3);

-n Disable automatic date/time handling (see section 6.3.3.4);

-o Allows a name on the global command line to override the Unix user name (see section 6.3.3.22);

-p Display perror diagnostic information (see section 6.3.3.5);

-q Allows a term name on the global command line to override the Unix term shell variable (see section 6.3.3.23);

-r Run glintd as root user (see section 6.3.3.6);

-s Ignore SIGHUP signal rather then terminating (see section 6.3.3.24);

-t*string* Pass type-ahead string to Global System Manager (see section 6.3.3.7);

-u Override new default *umask* value (see section 6.3.3.25);

-v*nn* Diskette timeout delay (see section 6.3.3.26);

-w*nn* Wait *nn* seconds for glintd process (see section 6.3.3.8);

-x Disable XON/XOFF handshaking (see section 6.3.3.9);

-X Enable console XON/XOFF flag (see section 6.3.3.27);

-y Clear I0XLAN flag (see section 6.3.3.10);

-z Suppress *glintd* hibernation (see section 6.3.3.11).

The -i argument is described in section 6.3.2. The other command line arguments are described below. Note that unlike the glinstall command line options (see section 6.2) multiple global command line options can be specified. For example:

```
# global -p -d -r
```

6.3.3.1 global -d Display Diagnostics

The -d command line argument causes global to display various diagnostic information during the Global System Manager initialisation process. This option also results in more detailed diagnostic messages written to the $GLDIR/sys/messages file.

6.3.3.2 global -h Hibernate During Locks

Use of the -h command line argument is a specialised technique which is beyond the scope of this manual and is documented here for completeness only. **DO NOT ATTEMPT TO USE THIS OPTION.**

6.3.3.3 global -m Suppress Messages

The -m option prevents global and glintd from writing messages to the $GLDIR/sys/messages file (see section 6.1.1.3). It should not normally be necessary to use this option as messages are written so infrequently that the effect on the performance of Global System Manager is negligible. However, on large systems the message file may rapidly increase in size. If the messages file is not deleted on a regular basis by the System Administrator, the -m option can be used to prevent the file from occupying too much space in the filing system.

6.3.3.4 global -n Disable Automatic Date/Time

The -n command line argument is of little general use. It may be used to override the date/time information that is normally supplied automatically by Unix. If the -n option is used Global System Manager will prompt the user for the date and time.

6.3.3.5 global -p Display perror Diagnostics

The -p command line argument causes global to display various diagnostic information, using the Unix *perror* function during the Global System Manager initiation process.

6.3.3.6 global -r Run glintd With root Permissions

The -r command line argument allows the glintd daemon to run with root privileges. This may be necessary under some circumstances (e.g. if $REMOTE (see section G.71) or SPD (see section G.72) are to be used). The -r option may also be useful in diagnosing some classes of problems.

6.3.3.7 global -t Pass Type-Ahead Text

The -t command line argument allows you to pass text to Global System Manager (via the type-ahead buffer - see Chapter 2 of the Global Configurator Manual). This technique can be used to execute any of the Global utilities, applications or menu entries directly from the Unix shell. For example, if you wish to run $STATUS then you can do so by keying:

```
# global -t'$STATUS:'
```

Where -t informs global that you wish to use the rest of the parameter as a command to execute. The colon character ":" is used to represent a <CR> as it is not normally possible to insert a <CR> into a Unix command line argument. Note that the parameter has been surrounded by single quotes because the characters "$" and ":" would be treated as special characters by the Unix shell. This shell command will run the Global utility $STATUS. On exit from $STATUS you will be returned to the Global System Manager main menu.

6.3.3.8 global -w Specify Wait Period

During Global System Manager initialisation, the global process spawns the *glintd* process and waits for it to respond. Normally, the delay is minimal (i.e. between 1 to 2 seconds) but on a sizeable configuration, with a large number of file-server SYSTEMs defined in the Systems file, it may take a relatively long time for the first *glintd* process to initialise all the file-servers. If the time-out period expires the Global System Manager initialisation will be aborted. By default, the wait period is 20 seconds. The global -w option may be used to modify this delay. For example, to increase the delay to 40 seconds:

```
# global -w40
```

6.3.3.9 global -x Disable XON/XOFF Handshaking

Use of the -x command line argument is a specialised technique which is fully described in section G.70. **Important note**: This option is **not** supported by BACNAT software version V3.313, or later.

6.3.3.10 global -y Clear I0XLAN Flag

Use of the -y command line argument is a specialised technique which is beyond the scope of this manual and is documented here for completeness only. **DO NOT ATTEMPT TO USE THIS OPTION**.

6.3.3.11 global -z Suppress glintd Hibernation

Use of the -z command line argument is a specialised technique which is beyond the scope of this manual and is documented here for completeness only. **DO NOT ATTEMPT TO USE THIS OPTION.**

6.3.3.12 global -c Allow Creation of Unix core File (V3.183, and later)

Use of the -c command line argument is a specialised technique which is fully described in section G.104.

6.3.3.13 global -a Re-attach to running glintd (V3.207 and later)

The –a option can be used to allow global to attach to an existing glintd background process, rather than initiating a new glintd. **Warning:** This option must be used with great care because glintd has not direct reference its parent global task. Using this option under the wrong circumstances may inadvertently “take over” an existing GSM (Unix) session.

6.3.3.14 global -b Suppress execution of background glintd (V3.227, and later)

Use of the -b command line argument, which suppresses execution of the background glintd process, thus allowing a debug version to be executed, is a specialised technique which is beyond the scope of this manual and is documented here for completeness only. **DO NOT ATTEMPT TO USE THIS OPTION**.

6.3.3.15 global -e Disable 32-bit option (V3.317, and later)

The –e option can be used to disable the GSM 32-bit run-time option. It may be required for some non-PM GSM configurations that have not installed a 32-bit software licence.

6.3.3.16 global -f Indicate GX connection (V3.317, and later)

The –f option **MUST** be used if GSM (Unix) is started from the telnet emulator within the GX.EXE (or AX.EXE) thin-client. Note that GX/AX connections to GSM (Unix) are normally made directly from the GX Connection Window, which bypasses the initial telnet session, so this option should be considered as specialised.

6.3.3.17 global -F Indicate OneOffice 3000 Workspace connection (V3.348, and later)

The –F option **MUST** be used if GSM (Unix) is started from the telnet emulator within the OneOffice 3000 WorkSpace (OO3KWS) client. Note that OO3KWS connections to GSM (Unix) are normally made directly from the OO3KWS Connection Window, which bypasses the initial telnet session, so this option should be considered as specialised.

6.3.3.18 global -g Leave glspad running (V3.206, and later)

Use of the -g command line argument is a specialised technique which is fully described in section G.107.

6.3.3.19 global -j Suppress re-open of standard TTY device (V3.340, and later)

The -j option disables the re-opening of the standard TTY device for stdout and stdin. This option can be used when GSM (Unix) does not have sufficient permissions to open these devices. For example when GSM (Unix) is being ruin from a cron job.

6.3.3.20 global -k Assume filing system size is 2Gb, or higher (V3.333, and later)

For various reasons, and for some versions of Unix, the algorithm that is used to calculate the free-space on filing systems that are much larger than 2Gb can produce inaccurate results. Typically, the free space displayed by $F and used by $V when allocating new subvolumes can be dramatically less than the GSM domain limit of 2Gb, even though the actual free space on the Unix filing system exceeds 2Gb. If the free space on the Unix filing system(s) that hold the GSM domain directories is larger than 2Gb, the –k option can be used to inform GSM (Unix) to ignore the results of the *statfs* function call and to use a “free space” value of 2Gb

6.3.3.21 global -l Suppress writes to spooler log file (V3.206, and later)

This option disables most updates to the $GLDIR/sys/spool/log log file from the *glspod* daemon process. Note that some messages, which are written before *glspod* has attached to the GSM shared memory, will still appear in this log file. Nevertheless, the –l option will drastically reduce the size of $GLDIR/sys/spool/log.

6.3.3.22 global -o Override Unix user name (V3.336, and later)

The -o option allows a NAME on the global command line to override the Unix User name. This specialized option may be useful when GSM (Unix) is running as cron job.

6.3.3.23 global -q Override Unix TERM shell variable (V3.336, and later)

The -q option allows a TERM name on the global command line to override the Unix TERM shell variable. This specialized option may be useful when GSM (Unix) is running as cron job.

6.3.3.24 global -s Ignore SIGHUP signal (V3.266, and later)

By default global will terminate if it receives a SIGHUP message from Unix. The –s option can be used to ignore the SIGHUP message.

6.3.3.25 global -u Override new default *umask* value (V3.332, and later)

For BACNAT V3.331, and earlier, the value of the umask variable is 0003. For BACNAT V3.332, and later, the value of the umask variable is 0000. This change was to allow applications to specify the **precise** permissions of a Unix file created using the Open BDAM Access Method or SVC-61 File Create functions. The –u option, without any further parameters, can be used to set the umask variable to 0003 for backwards compatibility with BACNAT V3.331, and earlier.

6.3.3.26 global -v Diskette timeout delay (V3.254, and later)

This option can be used to override the default delay period (1 second) that is normally required before the first read/write operation after an open on a diskette device. If this delay period is too short Buffer Write errors may be reported on some diskette operations. For example, to increase the delay to 5 seconds:

```
# global -v5
```

The special value of 0 will suppress all close operations on the diskette device thus keeping it

permanently open.

6.3.3.27 global -X Enable XON/XOFF Handshaking (V3.313, and later)

For BACNAT version V3.313, and later, the XON/XOFF hand-shaking protocol with the TTY device driver is disabled by default. Note that prior to V3.313 the “global –x” option (see section 6.3.3.9) was required to disable XON/XOFF hand-shaking. This change to the default behaviour at the release of BACNAT V3.313 was implemented to support the interface to GX **which requires XON/XOFF hand-shaking to be disabled**. If it is required to enable the XON/XOFF hand-shaking protocol with the TTY device driver for BACNAT V3.313, or later, the -X option must be used to explicitly enable the option.

6.4 gladmin, GLADMIN - Systems File Edit

gladmin is a simple shell script which uses the vi editor to allow the Systems file to be edited, then, if modifications have been made, runs glconfig (the Systems file compiler - see section 6.5) to produce the Global System Manager authorisation and initialisation data files (see section 6.1.1.6). This shell script is provided because **any changes made to the Systems file (an ASCII text file) using a Unix editor (e.g. vi) are not effective until glconfig is used to compile the text file.**

To run gladmin simply key:

```
# gladmin
```

gladmin invokes the vi editor using the following command:

```
vi $GLDIR/sys/Systems
```

For convenience, the gladmin script is linked to GLADMIN.

The gladmin script can only be invoked by super-user (root).

6.5 glconfig - Systems File Compiler

The glconfig program is a special compiler written for Global System Manager which converts the authorisation and configuration file, $GLDIR/sys/Systems (see section 6.1.1.1), into the data files used by each user when running Global System Manager. The various data files are held in the directory $GLDIR/sys/data (see section 6.1.1.6). The Systems file is a text file which contains a sophisticated and flexible dialogue describing the environment from which Global applications can be run (see Chapter 7 and Appendix F). Whenever the Systems file is modified, glconfig MUST be run to update $GLDIR/sys/data. Until glconfig has completed successfully the changes will not take effect. **It is not possible to run glconfig while Global System Manager is in use**. If an attempt is made to run glconfig while Global System Manager is in use, the following error will be reported:

```
glconfig: Error 1826 - Can't recompile Systems file. Global System Manager in use.
Configuration file not created.
```

To run glconfig simply key:

```
# glconfig
```

If there are no errors or inconsistencies in the Systems file then the files in the $GLDIR/sys/data directory are updated and the command returns immediately to the shell

without displaying any messages. If any fatal errors are encountered then an **error** message will be displayed in a similar form to that shown above and the contents of the $GLDIR/sys/data will NOT be updated. There is no danger of glconfig generating inconsistent files within the $GLDIR/sys/data directory when a fatal error occurs because it creates the new files in a temporary directory. The nascent files in the temporary directory are only copied to the $GLDIR/sys/data directory if no fatal errors are detected.

If a **warning** message is displayed, to indicate a non-fatal condition that can be ignored, then the files in the $GLDIR/sys/data directory will still be updated.

Error and Warning messages always come one of the following formats:

```
glconfig: Error number at line linenumber - error message.
```

or:

```
glconfig: Warning number at line linenumber - warning message.
```

Where *number* is an error code which can be used to find a description of the error message in Appendix C, *linenumber* is the line number of the error in the Systems file, and *error message* or *warning message* is a description of the error or warning.

The glconfig utility can only be run by super-user (root).

6.6 glmkdat - Data File Create Program

Before Global System Manager and the other Global applications can create or access any data units or subvolumes, the directories and associated header index files must be created. glmkdat provides an automatic mechanism for creating these files. After the files have been created they must be initialised to the Global data format before they can be used.

The glmkdat utility can only be run by super-user (root).

6.6.1 Create a New Discrete Data File Directory

To create a new Global data file run glmkdat. For example:

```
# glmkdat B00.dir
```

The above dialogue illustrates the creation of a Global data file, B00.dir.

The above example dialogue would result in the creation of the following files:

permissions	*owner*	*group*	*size*	*file*
drwxrwxr-x	global	global	N/A	B00.dir
-rw-rw----	global	global	32768	B00.dir/SVL00_DOMAIN

Note that the files are already owned by global and that the group *global* has read and write permissions to the files. To initialise this data volume, which must be defined in the Systems file (see section 7.3.2), run global to load Global System Manager then use the $V utility (see section 4.10.3).

If the name you specify starts with a slash / character, it is assumed to be an absolute path. Otherwise the file is created within the directory $GLDIR/data. The $GLDIR is replaced by the

value of this variable when you run the command. Note that there is no absolute requirement to create data files in the $GLDIR/data directory. They can be held anywhere in the Unix filing system.

glmkdat allows you to create multiple Global data files in a single invocation. For example:

```
# glmkdat B00.dir C00.dir D00.dir
```

will create 3 data files in the $GLDIR/data directory.

6.6.2 Obsolete glmkdat Dialogue

glmkdat was originally conceived as a menu-driven utility. This usage has been superseded by the recognition of command line arguments which results in a much simpler dialogue (see section 6.6.1). However, if an argument(s) is not supplied on the glmkdat command line it will revert to the obsolete menu-driven dialogue:

```
# glmkdat

Create Global System Manager data file:

      1 - Discrete data file
      2 - Integrated data file
      3 - Quit

Choose one : 1
Enter directory name for discrete data file : B00.dir

$GLDIR/data/B00.dir created.
```

Option 2 from the glmkdat menu (2 - Integrated data file) is only provided for backward compatibility with Global System Manager V6.2, or earlier, data files. Discrete data files should always be used in preference to Integrated data files.

6.7 glclean - Global System Manager Termination Program

glclean is used to terminate a user's Global System Manager process, or to terminate all users' Global System Manager processes and remove all Global System Manager resources from the system. It is used by Global System Manager itself when a user runs $BYE, but is also available to the System Administrator to remove any processes that have locked up and cannot be terminated using a standard Global System Manager utility such as $STATUS. glclean sends the signal SIGTERM to the appropriate processes and clears various flags in Unix shared memory to indicate the state of that SYSTEM. **It should only be used as a last resort to remove or restart a user as any Global applications currently in use will not have the chance to perform any recovery processing before the process terminates.**

To terminate a single SYSTEM:

```
# glclean -ssystemnumber
```

The -s flag indicates that a single SYSTEM is to be removed, and *systemnumber* is the number for the SYSTEM. This number can be specified in decimal, hexadecimal or octal. Hexadecimal numbers are prefixed by 0x and octal numbers are prefixed by 0. Note that there is NO space between the -s and the *systemnumber*. The user (or users) on this SYSTEM will be returned to the shell from which they ran the program global, together with error messages reporting the termination of their processes. If after removing the target user(s) no further Global System

Manager users remain, then the background Global System Manager resources are also removed and the following message is displayed:

```
Terminating processes and tidying Global System Manager resources
```

If glclean completes successfully, no other messages are displayed. Note also that no error message is displayed if the specified SYSTEM is not running Global System Manager.

For example, to terminate SYSTEM 27 key:

```
# glclean -s27
```

Note that it is NOT possible to clean multiple SYSTEMs with a single invocation of glclean (i.e. only one -s*systemnumber* command argument is allowed).

To terminate all Global System Manager users:

```
# glclean all
```

The word 'all' causes all Global System Manager processes to receive the signal SIGTERM, and for all Global System Manager resources to be removed from the system. The following message is displayed:

```
Terminating processes and tidying Global System Manager resources
```

SYSTEMs A to Z are specified by the *systemnumbers* 1 to 26 respectively.

Because of the potentially disastrous consequences of accidental use, if glclean is invoked without a parameter, no action is taken and the following error message is displayed:

```
glclean: Error 1604 - No system specified.
```

In addition to the mandatory -s*systemnumber* or "all" command line arguments, glclean recognises the -m option. This option suppresses the messages that are normally written to the $GLDIR/sys/messages file (see section 6.1.1.3) when glclean is employed to clean a SYSTEM or all SYSTEMs. For example:

```
# glclean -s27 -m
```

To summarize the glclean command line arguments:

-s*systemnumber*	clean the specified SYSTEM
all	clean all SYSTEMs
-m	suppress writes to the messages file

6.8 glsysdump - Dump $GLDIR/sys/data Data Files

The glsysdump utility can be used to produce a formatted dump of the various data files in the $GLDIR/sys/data directory (see section 6.1.1) that are produced by the glconfig utility (see section 6.5).

By default, glsysdump produces a summary report of all the files in the $GLDIR/sys/data directory, writing to the standard output device. The behaviour of glsysdump can be modified by any combination of the following command line arguments:

-f	generate full memory dump
-o	generate file offset information
shm	selectively dump the $GLDIR/sys/data/shm file
shmsize	selectively dump the $GLDIR/sys/data/shmsize file
tmap	selectively dump the $GLDIR/sys/data/tmap file
umap	selectively dump the $GLDIR/sys/data/umap file
tty	selectively dump the $GLDIR/sys/data/tty file
user	selectively dump the $GLDIR/sys/data/user file
screen	selectively dump the $GLDIR/sys/data/screen file

or the standard Unix redirection and piping techniques. For example, to produce a full report, with offset information, to the file /tmp/dump:

```
# glsysdump -f -o > /tmp/dump
```

or to produce a summary report of just the $GLDIR/sys/data/tmap and $GLDIR/sys/data/umap files, piped to the Unix more command:

```
# glsysdump tmap umap | more
```

6.9 glshmdump - Dump Shared Memory

The glshmdump utility can be used to produce a formatted dump of the Unix Shared Memory area (see Appendix D). Unix Shared Memory is absolutely essential to the implementation of Global System Manager (Unix). Although the detailed structure of shared memory is beyond the scope of this manual, the glshmdump utility may be a useful tool in problem diagnosis.

Unlike glsysdump (see section 6.8), which dumps the contents of the data files in the $GLDIR/sys/data directory, glshmdump displays the control blocks that govern the operation of a running Global System Manager configuration. Consequently, glshmdump can only be used when Global System Manager is in use. If an attempt is made to run glshmdump when Global System Manager is not executing, the following error message will appear (see Appendix C):

```
# glshmdump
IPCBASE = 47534d30
glshmdump: Error 1902 - Cannot attach to shared memory (IPC inconsistency)
```

The most useful parameter displayed by glshmdump is the actual amount of shared memory being used by a running Global System Manager configuration. Thus, glshmdump can be used to optimise the value of the SHMSIZE option in the Systems file (see section 7.4.4).

By default, glshmdump produces a summary report of all the control blocks held in shared memory, writing to the standard output device. The behaviour of glshmdump can be modified by any combination of the following command line arguments:

header	selectively dump the "header" information;
lanflags	selectively dump the Global System Manager lan (*sic*) flags;
printers	selectively dump the information that controls the Global System Manager printer handling
screens	selectively dump the information that controls the Global System Manager screen handling and tty buffering;
sysinfo	selectively dump the information that describes each SYSTEM;
-h	short-hand form of "header" option;
-i	short-hand form of "sysinfo" option;
-l	short-hand form of "lanflags" option;
-p	short-hand form of "printers" option;
-s	short-hand form of "screens" option.

or the standard Unix redirection and piping techniques. For example, to produce a dump of the printer information to the file /tmp/dump:

```
# glshmdump -p > /tmp/dump
```

or to produce a report of just the header and sysinfo information piped to the Unix more command:

```
# glshmdump header -i | more
```

6.10 glreorg - Reorganise Data File Domains

The glreorg utility allows a data file domain to be reorganised in situ without the need to run $REORG (see section 4.5). However, unlike $REORG, glreorg only allows the size of a subvolume file to be altered - it does not perform any internal reorganisation.

The glreorg utility can only be run by super-user (root).

The glreorg utility cannot be used when Global System Manager is executing.

The glreorg utility can only be used on volume format T224Z or T151Z (i.e. volume formats with a virtual track size of 8Kb). Furthermore, the maximum number of files in the directory (see section 9.2.1) MUST be 250.

DO NOT ATTEMPT TO USE glreorg ON ANY VOLUME OTHER THAN A FORMAT T224Z/T151Z VOLUME WITH 250 FILES PER DIRECTORY OTHERWISE DATA CORRUPTION MAY OCCUR.

BEFORE STARTING TO REORGANISE A VOLUME, YOU SHOULD MAKE SURE YOU HAVE RECENT BACKUPS OF ALL THE DATA IT CONTAINS, SINCE IF THE REORGANISATION FAILS THE ENTIRE DATA VOLUME MAY BE LOST.

The syntax of the glreorg command is:

```
glreorg [-d] [-i] [-s size] [-t tmpfile] file [files]
```

The various command line arguments are as follows:

-d enable diagnostic mode (see section 6.10.6). This argument is optional;

-I prompt to confirm each file (see section 6.10.5). This argument is optional;

-s specify new filesize. If this argument is supplied it MUST be followed by a filesize parameter (see section 6.10.1). If a size argument is NOT supplied all the files specified will be rounded up to an integral number of 8Kb tracks (if necessary) and no further action will be taken. If a size argument is supplied all the files specified will be rounded up to an integral number of 8Kb tracks (if necessary) before the re-sizing is performed;

-t specify the name of a temporary file to be used if a subvolume file is reduced in size (see section 6.10.4). If this argument is supplied it MUST be followed by the name of a temporary work-file;

Any number of files may follow the last command line argument. Every file specified MUST be named SVL*nn_vvvvvv*, where *nn* are any numbers between 01 and 99; and *vvvvvv* are any characters. It is NOT possible to alter the size of the SVL00_*nnnnnn* header file. Normally, the files will reside within a Discrete Data Volume directory (e.g. A00.dir, B00.dir etc.).

6.10.1 The size argument

The size argument (i.e. the argument immediately following the -s command line option) can be specified in any of the following formats:

nnnn change size of file(s) to *nnnn* Mbytes

*nnnn*K change size of file(s) to *nnnn* Kbytes

*nnnn*M change size of file(s) to *nnnn* Mbytes

+*nnnn* increase size of file(s) by *nnnn* Mbytes

+*nnnn*K increase size of file(s) by *nnnn* Kbytes

+*nnnn*M increase size of file(s) by *nnnn* Mbytes

-*nnnn* decrease size of file(s) by *nnnn* Mbytes

-*nnnn*K decrease size of file(s) by *nnnn* Kbytes

-*nnnn*M decrease size of file(s) by *nnnn* Mbytes

x*nnnn* multiply size of file(s) by *nnnn*

/*nnnn* divide size of file(s) by *nnnn*

%*nn* increase size of file by *nn* percent (where *nn* is between 1 and 100)

Note that the default unit of file size is Mbytes (i.e. not bytes). All file-sizes are rounded **up** to an exact multiple of the virtual track size (i.e. 8Kb) unless the "/nnnn" filesize option is specified, in which case the resultant file-size is rounded **down** to an exact multiple of 8Kb. If any filesize rounding is performed a warning message is displayed.

The Global System Manager sub-volume data-size is always 11.5Kb less than the size of the corresponding Unix file (to allow for the Global System Manager directory which always occupies 23 512-byte sectors). For example, if you specify a Unix file size of 1Mb (i.e. 1048576 bytes), the data-size of the corresponding Global System Manager sub-volume will be 1036800 bytes).

6.10.2 Using glreorg to Round-up Filesizes

All subvolume files (e.g. SVL01_SYSRES) MUST be a multiple of 8Kb in size. If the size of a subvolume file is not an exact multiple of 8Kb it will be ignored by Global System Manager (see section G.62). Under no circumstances will any Global System Manager function (e.g. $V or $REORG) change the size of a Unix subvolume files to any value other than an exact multiple of 8Kb). The size of a subvolume file can be only be set to an invalid size by the unauthorized use of a Unix utility or if the Unix filing system has become corrupted (e.g. if the computer has been switched off without running the necessary Unix close-down procedures).

If the size of subvolume file is not a multiple of 8Kb, treat the contents of the corresponding sub-volume with suspicion as an invalid filesize may be a symptom of a more insidious data corruption problem.

glreorg will automatically round-up the size of any file, or files, specified to an exact multiple of 8Kb. If no size argument is specified, glreorg will only perform the automatic rounding. If a Unix file is already an exact multiple of 8Kb, the file will be unaltered and a warning message will be displayed. For example:

```
# glreorg $GLDIR/data/A00.dir/SVL01_SYSRES
glreorg: Error 2129 - No need to round filesize:
/u/gsm81/global/data/A00.dir/SVL01_SYSRES
```

Note how the $GLDIR shell variable has been expanded by the Unix shell.

Rounding-up is performed by appending the requisite number of bytes containing binary-zeroes to the end of the subvolume file.

Errors will be reported if there is insufficient space on the Unix filing system to increase the size of a file. In the extremely unlikely event of this error occurring (the maximum possible amount of data written will be only 8191 bytes) the size of the subvolume filesize will be unpredictable.

After the filing system has been reorganised to create more free space, glreorg will have to be re-employed to perform another rounding-up operation.

6.10.3 Using glreorg to Increase Filesizes

If a size argument is specified and the new Unix filesize (after possible rounding up) is larger than the current filesize (after possible rounding up) the filesize will be increased by appending the requisite number of bytes containing binary-zeroes to the end of the subvolume file.

glreorg will automatically round-up the size of any file, or files, specified to an exact multiple of 8Kb. If a Unix file is already an exact multiple of 8Kb (i.e. the normal case), no automatic rounding is necessary.

The new filesize can be specified in any of the formats described in section 6.10.1. For example, to set the Unix filesize to 1Mb:

```
# glreorg -s 1m $GLDIR/data/A00.dir/SVL10_WORK
```

For example, to increase the size of all subvolume files by 16Kb:

```
# glreorg -s +16K $GLDIR/data/A00.dir/SVL*
glreorg: Error 2105 - Cannot change size of domain header file:
/u/gsm81/global/data/A00.dir/SVL00_SYSDOM
```

Note how the Unix shell has expanded the "SVL*" file specification.

For example, to double the size of the subvolume files that correspond to units x20 and X21:

```
# glreorg -s x2 $GLDIR/data/A00.dir/SVL2[01]*
```

Note that "x" characters is used as the multiply operator rather than more obvious "*" (which would be (incorrectly) interpreted by the Unix shell).

An error will be reported if there is insufficient space on the Unix filing system to increase the size of a file. If this error occurs the size of the subvolume filesize will be unpredictable. After the filing system has been reorganised to create more free space, glreorg will have to be re-employed to perform another rounding-up operation.

6.10.4 Using glreorg to Decrease Filesizes

If a size argument is specified and the new Unix filesize (after possible rounding up) is smaller than the current filesize (after possible rounding) the filesize will be decreased. Because Unix does not provide a file truncation function, the filesize reduction is performed by copying only the required portion of the existing file to a work file before renaming (or copying - see below) to recreate the original file name.

glreorg will automatically round the size of any file, or files, specified to an exact multiple of 8Kb. If a Unix file is already an exact multiple of 8Kb (i.e. the normal case), no automatic rounding is necessary.

Before attempting the file truncation, glreorg checks to ensure the required filesize is not less than the extent of last Global System Manager file on the corresponding Global System Manager volume. If an attempt is made to truncate a subvolume file below the extent of the last

Global file, which would result in file and directory corruption, a warning message is displayed and the request is increased to minimum size allowed.

The new filesize can be specified in any of the formats described in section 6.10.1. For example, to set the Unix filesize to 1Mb:

```
# glreorg -s 1m $GLDIR/data/A00.dir/SVL10_WORK
```

For example, to decrease the size of all subvolume files by 16Kb:

```
# glreorg -s -16K $GLDIR/data/A00.dir/SVL*
glreorg: Error 2105 - Cannot change size of domain header file:
/u/gsm81/global/data/A00.dir/SVL00_SYSDOM
```

Note how the Unix shell has expanded the "SVL*" file specification.

For example, to halve the size of the subvolume files that correspond to units x20 and X21:

```
# glreorg -s /2 $GLDIR/data/A00.dir/SVL2[01]*
```

A special size of 0 indicates the filesize is to be reduced to the minimum size allowed, given the structure of the Global System Manager directory (i.e. glreorg will truncate the Unix file up to the extent of the last Global file, rounding the size up to an exact multiple of 8Kb if necessary).

For example, to minimise the size of the subvolume file SVL10_WORK:

```
# glreorg -s 0 $GLDIR/data/A00.dir/SVL10_WORK
```

No internal reorganisation of the Global System Manager directory is attempted. Any gaps in the Global System Manager volume will remain after the truncation. It is necessary to condense the volume using the $F CON command or $REORG to gain the most benefit from a glreorg file truncation.

By default, the file truncation is achieved by a "copy and rename" combination corresponding to the following Unix commands:

cps SVL*nn_vvvvvv* TMP*nn_vvvvvv*
mv TMP*nn_vvvvvv* SVL*nn_vvvvvv*

where "cps" is an (unavailable) Unix command (the complicated "dd" command provides some of the functionality) that copies a file to one of a smaller size, ignoring any data after the required file extent; and "mv" is the standard Unix file rename command.

Note that the temporary file is created in the same Unix directory (and hence the same Unix filing system) as the original space. The corollary is that sufficient free space is required in the filing system to allow the temporary file to be created.

An error will be reported if there is insufficient space on the Unix filing system to create the temporary file. If this error occurs the original subvolume file will be unaltered but the temporary file will remain taking up valuable disk space.

6.10.4.1 Using glreorg to Decrease Filesizes (in Special Mode)

The "-t" command line argument can be used to specify the name of the temporary file thus allowing the work file to be created in another filing system (i.e. normally a filing system with more free space than the filing system which contains the Global System Manager data directory).

If the -t option is specified, the file truncation is achieved by a "copy, delete and copy-back" combination corresponding to the following Unix commands:

cps SVL*nn_vvvvvv tmpfile*
rm SVL*nn_vvvvvv*
cp *tmpfile* SVL*nn_vvvvvv*

where *tmpfile* is the temporary file specified after the -t argument on the glreorg command line; "cps" is an (unavailable) Unix command (the complicated "dd" command provides some of the functionality) that copies a file to one of a smaller size, ignoring any data after the required file extent; "rm" is the standard Unix file delete command; and "cp" is the standard Unix copy command.

Note that the temporary file can be created in any Unix directory (and hence in any Unix filing system). The corollary is that there is no requirement for sufficient free space in the filing system to allow the temporary file to be created, provided, of course, that there is sufficient free space in at least one other Unix filing system.

An error will be reported if there is insufficient space on the Unix filing system to create the temporary file. If this error occurs the original subvolume file will be unaltered but the temporary file will remain taking up valuable disk space. An error will also be reported if there is insufficient space on the Unix filing system to copy back to the original filename. If this error occurs the size of the subvolume filesize will be unpredictable (although the temporary file will provide a backup). After the filing system has been reorganised to create more free space, the standard Unix cp command can be used to copy the temporary file back to the Global System Manager data directory.

6.10.5 The glreorg "-i" option

If the "-i" option is specified, glreorg will ask you confirm each file that matches the filename argument(s). For example:

```
# glreorg -i -s /2 $GLDIR/data/A00.dir/SVL2[01]*
```

will result in the following dialogue:

```
Reorganise SVL20_G3PROG ? y<CR>
Reorganise SVL21_G3DATA ? y<CR>
```

To proceed with a file reorganisation reply "y" or "Y" to the above prompt. A reply of <CR> or any reply that does not start with a "y" or "Y" will indicate that the file is not to be reorganised.

6.10.6 The glreorg "-d" option

If the "-d" option is specified, glreorg will display various diagnostic information as the reorganisation proceeds. In particular, glreorg will use the Unix perror function to display the full details of any error returned by a Unix function. For example, full details of read and write errors will be displayed if the -d option is specified.

This option should not be required in normal circumstances.

6.11 glinfo - Capture Global System Manager and Unix Information

glinfo is a simple Bourne shell script that captures most of the important information regarding the status of an executing Global System Manager (Unix) configuration. The information logged by glinfo may also be useful if Global System Manager (Unix) is not running. The output of every Unix command executed by glinfo is redirected to a text file, $GLDIR/tmp/info. This text file may be required to investigate some types of problems with Global System Manager (Unix).

The glinfo script is held in the $GLDIR/bin directory (see section 6.1.2) and is invoked in the same way as the other commands documented in this chapter. For example:

```
# glinfo
```

The glinfo script contains the following Unix commands:

Command	*Description*
date	Get Unix current date and time
id	Get current user id and group name
uname	Print the name of the current Unix system
cat GLDIR/sys/version	Take a copy of the current version file (see section 6.1.1.4)
cat $GLDIR/sys/Systems	Make a copy of the current Systems file (see section 6.1.1.1)
ps	Obtain a full Unix process status report
ls $GLDIR	Obtain a full listing of the Global System Manager directory
ipcs	Report the status of the Unix IPC facilities
glsysdump	Obtain a dump of the various compiled Systems information (see section 6.8)
glshmdump	Obtains a dump of the Global System Manager shared memory (see section 6.9)

Other Unix commands may be added to this script.

The actual parameters associated with each Unix command depend on the version of Unix. For example, for most versions of Unix the "uname -a" command is used to print the name of the current Unix system. However, for SCO Unix, the "uname -X" command, which returns additional information, is employed.

7. Configuring Global System Manager

This chapter describes how to modify the *Systems* file to configure Global System Manager for your Unix system. It covers the user authorisation procedure which restricts the use of Global System Manager to a group of users. This chapter also describes the definition of data files, access to diskettes, and setting up the Global System Manager terminal handling.

Other techniques that may be used to configure Global System Manager on Unix (e.g. the use of Unix shell variables, modifying the Global configuration file) are described in the following two chapters.

7.1 An Overview of the Systems File

The *Systems* file (i.e. $GLDIR/sys/Systems - see section 6.1.1.1) is the main configuration and authorisation database for Global System Manager. It is a text file which **must be compiled after editing** to form the parameter files used by Global System Manager. The *Systems* file is used to define the following parameters:

- who, or from where Global System Manager can be run;
- the names of the Global data files;
- which diskette devices to access;
- the table of TERM and corresponding TAP definitions;
- where the print files are to be sent;
- the short user names for Global System Manager.

Appendix F contains several examples of Systems files describing various configurations. Appendix F (section F.1) also includes an example of the standard Systems file distributed with Global System Manager.

The Systems file is composed of a set of command words (in upper-case letters) each being followed by one or more parameters. The text-file is free format but there are associations between command words. This association depends upon their relative position in the file.

The general format of the Systems file is as follows:

SYSTEM *parameter* The command word SYSTEM must precede any of the next set of command words and MUST be unique.

The following command words are associated with a particular SYSTEM and may come in any order in the Systems file:

CONTINUE *parameter* Continue with a previously defined SYSTEM. Listed for completeness only.

SYSDATA *parameter* Normally only set up for SYSTEM A.

DATA *parameters* Global data file for this SYSTEM.

DISKETTE *parameters* Diskette formats for this SYSTEM.

USER *parameters* User authorisation, defines this SYSTEM as specified user.

The following command words affect all SYSTEMs and may be placed anywhere within the Systems file:

INSDEV *parameter* Only one specified installation device for Global System Manager.

SHMSIZE *parameter* Shared memory size, shared by all Global System Manager users.

TERM *parameters* TERM to TAP map used by all SYSTEMs.

OPIDMAP *parameters* Short user name (operator ID) used by any SYSTEM for that user.

PRINTER *parameters* Printer output used by all SYSTEMs.

All SYSTEM definitions for SYSTEMs B to Z that include the command word DATA are defined as dedicated file servers and cannot include the USER command word. A SYSTEM which is followed by the DATA command word is merely a way of defining Global data files, accessible by all other SYSTEMs. These virtual "file-server" SYSTEMs do not run as Unix processes and thus are note strictly file servers. Conversely, SYSTEMs B to Z that include the command word USER cannot also include the command word DATA and therefore cannot be file servers.

7.2 Modifying and Recompiling the Systems File

The Systems file is a Unix text file that can be modified using any of the standard Unix editors. Once the modifications have been made the Systems file must be recompiled to generate the binary data files used by Global System Manager (see section 6.1.1.6). **This recompilation cannot take place whilst Global System Manager is in use**. The compiler is called *glconfig* (see section 6.5) and can only be run by the Unix users *global* and *root*.

To simplify the Systems file modification process a simple Unix script, *gladmin*, is provided which allows you to edit the Systems file using vi, then automatically recompiles the file if any modifications have been made. The gladmin script is fully described in section 6.4.

7.3 A Detailed Description of the Systems File

This section describes each aspect of the Systems file and should be read in conjunction with Appendix F.

7.3.1 User Authorisation

This section should be read in conjunction with section F.2.5. To authorise a user for Global System Manager simply add a new SYSTEM number to the file followed by a USER description.

The SYSTEM definition might look like this:

SYSTEM 27

This must be followed by the USER description which can be supplied in one of two formats. The first format is as follows:

USER *number* LOGNAME *username*

where *number* refers to the User number defined in the USER DISPLAY ATTRIBUTES section of the configuration file (see section 9.3). The LOGNAME command word causes the Global System Manager SYSTEM to be assigned to the user whose user name is *username*. Whenever this user loads Global System Manager they will have this SYSTEM configuration.

The alternative format is:

USER *number* TTY *ttydevice*

where *number* is as above, and TTY assigns this Global System Manager SYSTEM to the tty device *ttydevice*. Whenever a user who has not been authorised using the previous method loads Global System Manager on this *ttydevice*, the user will have this SYSTEM configuration.

The *username* and *ttydevice* must be uniquely defined within the Systems file. Any duplication will cause the glconfig compiler to fail with an appropriate error message.

The SYSTEM identifier (i.e. 27 in this example) can be specified in one of two ways. Firstly, as a decimal number between 27 and 255. Note that, for internal reasons, a SYSTEM number of 192 is forbidden. Note also that SYSTEM numbers 1 to 26 are aliases for file-server SYSTEMs "A" to "Z". A SYSTEM identifier can be specified as a hexadecimal number by prefixing with "0x". Hexadecimal numbers 0x1b to 0xff are allowed. Note that, for internal reasons, a SYSTEM number of 0xc0 is forbidden.

7.3.2 Configuring Data Files

This section should be read in conjunction with section F.2.4. A data file is a directory which contains the Global data format files (or subvolumes). The subvolumes are created and deleted dynamically by the Global applications as required, but the directory must be defined within the Systems file. This is achieved by adding the definition DATA to an existing file server SYSTEM or preferably by adding a new file server SYSTEM and adding the DATA definition to this.

A file server SYSTEM is simply a way of assigning data files in the format expected by Global System Manager. A file server SYSTEM is a data area for those data files defined in the Systems file. A file server SYSTEM must have a SYSTEM letter A to Z. Any SYSTEM B to Z that has a DATA definition is a **dedicated file server** and thus USER authorisation can not be set up for such a SYSTEM. **Only SYSTEM A can be both a file server and have USER authorizations**.

Within the Global configuration file there is a database of DATA FILE DEFINITIONS from which a description can be selected using the DRIVE *number* (see section 9.2). This *number* must correspond to the DATA *number* defined in the Systems file. For a particular SYSTEM these numbers must be unique although the same description may be used by different SYSTEMs.

To add a data file to a new file server SYSTEM add:

SYSTEM *letter*

to the Systems file, where *letter* is in the range A to Z and is not currently defined. Then add the DATA definition:

DATA *number directory_name*

or:

DATA *number full_path_name*

where the *number* is as defined above, *directory_name* refers to a directory within the directory $GLDIR/data ($GLDIR is the shell variable defining the path of the global directory - see section 8.1.1) and *full_path_name* refers to the full pathname of a directory that is NOT within the $GLDIR/*data directory*.. **Important note:** Shell variables cannot be used within the Systems file) unless *filename* starts with a forward slash ('/').

Before Global System Manager, and Global application software, can use this new data file, the directory and index file must be created using the *glmkdat* command as described in section 6.6. When using *glmkdat*, you will normally specify the directory name as a command line argument (see section 6.6.1). The *glmkdat* command will create the new data file for you with the correct permissions for use by Global System Manager. Note that there is no requirement to add the DATA definition to the Systems before running *glmkdat* - there is no cross reference to the Systems file by *glmkdat*.

Important note: The data files used by Global System Manager must not be linked to other Unix files (i.e. the only link should be to the parent directory).

7.3.3 Configuring Diskettes

This section should be read in conjunction with section F.2.2. Global System Manager is capable of accessing diskettes that are in Global data format. Both the Global AN code and the corresponding Unix device name must be set up in the Systems file. For each diskette drive on your system you should set up a list of these formats within the Systems file.

DISKETTE *number* AN *devicename*

The configuration file acts as a database of formats. Each associated set of formats (e.g. those formats used on 3½" diskettes) should have a unique DRIVE *number* associated with them (see section 9.2.3). This *number* can then be used in the Systems file when defining the formats you require. If a format is not in the Global configuration file then defining it in the Systems file will have no effect. Diskette definitions are added to file server SYSTEMs in the same way as DATA definitions. There is no checking performed on the diskette device name to ensure its uniqueness, so you must ensure that you do not define the same physical diskette drive in more than one DISKETTE *number* description, as all file locking will be lost. The Systems file as distributed defines the diskettes on SYSTEM A. This is required to allow Global System Manager to be installed from diskette.

To add diskette formats to an existing drive description, add the DISKETTE description to the SYSTEM for which that DISKETTE is already defined:

DISKETTE *number uniqueANcode devicename*

where *number* is the same as that previously defined for this drive, *uniqueANcode* is a new AN code (e.g. O2 for O2A, or B3 for B3B) that has not previously been defined for this drive, and *devicename* is the device name for this diskette geometry on this drive. For example:

DISKETTE 0 O2 /dev/rfd0135ds18

adds the format O2 to DISKETTE 0 for this SYSTEM (defined by the previous SYSTEM command in the file).

Important note: It is only necessary to define the AN code for Global diskette ANA formats. The last alphabetic character (e.g. "D" for O2D formats) must be dropped when adding DISKETTE definitions to the Systems file.

7.3.4 TERM Handling

This section should be read in conjunction with section F.2.7. Global System Manager uses its own TERM files to define the terminal attributes. These files are referred to as Terminal Attribute Programs (TAPs). Global System Manager reads the value of the shell variable TERM and attempts to use the corresponding TAP number from the Systems file. If there is no corresponding TAP then the user will be prompted for a TAP number when they run global (see section 3.2.3). To add a TERM to TAP map to the Systems file use the following construct:

TERM *termtype tapnumber*

where *termtype* is the value of $TERM and *tapnumber* is the corresponding Global System Manager TAP number. For example, to add an entry for a Wyse-50 terminal ($TERM value of 'wyse50' and a TAP number of 163) simply add:

TERM wyse50 163

You can add as many entries as you like in this way.

The TAPs are stored in Global data format. You must have the appropriate TAPs installed on your Global SYSRES subvolume before they can be used (see section A.15).

7.3.5 Defining Short User Names

This section should be read in conjunction with section F.2.6. Global System Manager maintains a short user name (operator-id) extensively to keep a record of transactions and operations performed by each user. This operator-id allows the system administrator to determine information such as who printed out a report, who created a specific record, etc. These short user names can be defined either in the Systems file or as a Unix shell variable (see section 8.3.2). The short user name is defined in Systems file either as:

OPIDMAP *username operator_id*

or:

USER *number* LOGNAME *username* OPID *operator_id*

where *username* is the Unix user name and *operator_id* is the short user name used by Global System Manager. The second description shows a standard USER authorisation to which is appended (shown as underlined) the short user name to be used for this user.

The short user name (operator-id) can be up to four characters in length.

7.3.6 Printers

This section should be read in conjunction with section F.2.3. Global System Manager and Global applications print via printer units. Each unit is given a unique number. Valid unit numbers are 500 to 599 inclusive. All printer output is performed via one of these unit numbers. The Systems file contains a description for each unique printer unit describing either the name of the Unix printer device to define a "direct printer" or the Unix spooler command to be executed to define a "spooled printer".

Appendix E contains a complete description of the Global System Manager printer interface.

7.3.6.1 Direct printers

The Systems file format for "direct printers" is as follows:

PRINTER *number devicename*

This construct defines all printer output sent to unit *number* to be written directly to device *devicename*. The number must correspond to a UNIT NUMBER in the PRINTER ATTRIBUTE section of the Global configuration file, and the CONTROLLER type for this definition must be DIRECT (see section 9.5.1).

For example:

PRINTER 500 /dev/ttypc01

will cause all printer output to the printer unit 500 to be sent directly to the printer attached to the device /dev/ttypc01.

7.3.6.2 Spooled printers

The format for "spooled printers" is as follows:

PRINTER *number* "*shellcommand*"

which defines that the Global print files sent to the printer unit number are printed by the command *shellcommand*. The *shellcommand* as its name implies could be any valid shell command. It is usually a Unix spooler command or a shell script which calls the Unix spooler. The quotes (" or ') are required around the whole *shellcommand* if it contains spaces. The *number* must correspond to a UNIT NUMBER in the PRINTER ATTRIBUTE section of the Global configuration file, and the CONTROLLER type for this definition must be SPOOLED (see section 9.5.2).

For example:

PRINTER 510 "lpr -r"

will cause the Unix spooler *lpr* to be invoked with the -r flag, which, in this example, informs the *lpr* command to remove the file after printing. Appended to the Unix command will be the Global print file name. For example the following Unix command would be executed if SYSTEM 0x27 performed a screen dump to printer unit 510:

```
$ lpr -r D.SCREEN.27.01
```

If it is inconvenient to have the Global print file name appended to the end of the Unix command string a string substitution mechanism is available. All occurrences of the string "$FILE" within the Unix command string will be replaced by the name of the Global print file. For example:

```
PRINTER 510 "cat $FILE >/dev/tty1a;rm $FILE"
```

will be expanded into the following Unix command string if, for example, SYSTEM 0x27 performed a screen dump to printer unit 510:

```
cat D.SCREEN.27.01 > /dev/tty1a;rm D.SCREEN.27.01
```

In this example, the string substitution mechanism has been used to ensure that the Global print file is deleted from the spool directory. Note in this example, the $FILE string is required at the end of the command string as the filename is not automatically appended if the command string includes "$FILE".

7.4 Advanced Topics

This section describes some advanced topics that are of specialised use.

7.4.1 Multiple USER SYSTEMs

This section should be read in conjunction with section F.2.11. It is possible to set up the user authorisation in such a way that there is more than one USER per SYSTEM. This is required if you require more than the maximum 255 users available with a single USER per SYSTEM. To do this simply add more USER definitions to the list for that SYSTEM ensuring that the USER *number* is unique and corresponds to a set of USER DISPLAY ATTRIBUTES in the configuration file.

The maximum number of USERs per SYSTEM is 99.

7.4.2 Shortcuts

Most of the command words within the Systems file remain active until a new command word is specified. For example defining data files for a particular SYSTEM might involve the following text:

```
DATA        0       A00.dir
DATA        1       A60.dir
DATA        2       A80.dir
```

This could more easily be defined as follows:

```
DATA
            0       A00.dir
```

```
1       A60.dir
2       A80.dir
```

The descriptions for the same command do not have to be together. For example, if the above list were followed by DISKETTE definitions, these DISKETTE definitions could then be followed by more DATA definitions:

```
DISKETTE 0 G1 /dev/fd096ds15

DATA 3 A90.dir
```

These rules apply to all the command words in the Systems file.

The CONTINUE command word allows descriptions for a particular SYSTEM to be defined in different sections of the Systems file. For example:

```
# Define file-servers first
SYSTEM     A      DISK  0      A00.dir
SYSTEM     B      DISK  0      B00.dir
SYSTEM     C      DISK  0      C00.dir

# Define all USERs together
SYSTEM     0x1b   USER 1       LOGNAME   global1
CONTINUE   A      USER 1       LOGNAME   root
SYSTEM     0x1c   USER 1       LOGNAME   global2
```

This technique is not recommended as it can result in Systems files that are difficult to maintain!

7.4.3 Private data files

This section should be read in conjunction with section F.2.10. Non file server SYSTEM definitions (SYSTEMs outside the range A to Z) can also have private data files defined in the Systems file. These data files are defined in the same way as the file server DATA definitions but are associated with a particular user (or set of users; see above) as one SYSTEM. Only USERs defined with this SYSTEM can access these data files. This configuration might be used to protect sensitive information (e.g. Payroll data files).

7.4.4 Changing the size of shared memory

This section should be read in conjunction with section F.2.12. Shared memory is used to hold the control data for any shared resources (see section D.1.1). On large configurations it may be necessary to increase the size of shared memory allocated by Global System Manager. If the shared memory is insufficient for the configured resources, users may experience memory errors (displayed as error "M") or may be prevented from using some of the resources (including Global System Manager itself). The amount of Unix shared memory used by an executing Global System Manager configuration can be determined by running the *glshmdump* utility (see section 6.9).

To change (usually increase) the amount of shared memory allocated by Global System Manager the Systems file command SHMSIZE is used:

```
SHMSIZE size
```

where *size* is in Kbytes (i.e. 1024 bytes).

For example, to increase the shared memory allocation to 256Kb add the following line to the Systems file:

```
SHMSIZE 256
```

Each file server SYSTEM requires approximately 50K bytes of shared memory. Each USER requires approximately 300 bytes of shared memory. These estimates are very approximate and depend on the values of various Nucleus Options defined in the Global Configuration file. The glshmdump utility (see section 6.9) can be used to determine the actual amount of shared memory utilised when Global System Manager is running.

You must ensure that the value specified is within the system imposed limits. If not you may need to tune or rebuild your kernel.

8. Unix Shell Variables

This chapter describes those Unix shell variables that may be used to modify the behaviour of Global System Manager (Unix).

Before Global System Manager can be used, two obligatory shell variables MUST be established. It is usually most convenient for the System Administrator to set up these variables in each user's login script (e.g. .profile or .login) when authorising users for Global System Manager. In addition to the obligatory shell variables, many others may be defined to override Global System Manager defaults or enable levels of diagnostics.

8.1 Obligatory Unix Shell Variables

The following Unix shell variables MUST be defined before Global System Manager can be used.

8.1.1 GLDIR

This shell variable MUST be set to the *global* directory into which Global System Manager is installed. This variable is used extensively to determine the location within the Unix filing system of many of the files used to configure Global System Manager. For example:

```
GLDIR=/usr/global
```

If the GLDIR shell variable is not defined, the following error message (see Appendix C) will appear when an attempt is made to invoke Global System Manager:

```
Global System Manager: Error 1005
(can't open error header file "sys/errmsg/hdr")
```

8.1.2 PATH

The *global/bin* directory contains all the commands and utility programs required by Global System Manager. To run these programs the directory $GLDIR/bin must be added to the PATH variable. For example:

```
PATH=$PATH:$GLDIR/bin
```

Consult your Unix documentation for a full description of the PATH shell variable.

8.2 Unix Shell Variables for Terminal Mapping

This section describes those Unix shell variables that affect the Global System Manager terminal type mapping.

8.2.1 TERM

The general-purpose TERM shell variable is used to select the correct Terminal Attribute Program (TAP) from the Global TAP library. The mapping between the TERM and the TAP is defined in the Systems file (see section 7.3.4). For ease of operation it is recommended that all the TERM types used on the Unix system are mapped to TAPs in the Systems file and that these TAPs are in the Global TAP library.

Consult your Unix documentation for a full description of the TERM shell variable.

Note that the TERM shell variable may be overridden by the GLTERM (section 8.2.2) and GLTT (section 8.2.3) shell variables.

8.2.2 GLTERM

The special-purpose GLTERM shell variable is recognised by Global System Manager to override the general-purpose TERM variable (see section 8.2.1). For example, if TERM is set to "vt100" and GLTERM is set to "vt200", Global System Manager will use the TERM mapping for the "vt200" as defined in the Systems file (see section 7.3.4).

Note that both the TERM and GLTERM shell variables may be overridden by the GLTT shell variable (see section 8.2.3).

8.2.3 GLTT

The special-purpose GLTT shell variable is recognised by Global System Manager to override the default terminal mapping, if any, in the Systems file (see section 7.3.4). For example, if TERM is set to "vt100" and GLTT is set to "163", terminal type 163 (i.e. TAP $.163) will be used when Global System Manager is invoked.

Note that whereas both the TERM and GLTERM shell variables are set to the **name** of a Unix terminal description (e.g. "wyse50"), the GLTT shell variable must be set to the **number** of a Global System Manager TAP (e.g. "163").

8.2.4 Order of Precedence of Terminal Mapping Shell Variables

The order of precedence of the TERM, GLTERM and GLTT shell variables is as follows:

```
IF shell variable GLTT is defined THEN
        Use GLTT directly
ELSE
        IF shell variable GLTERM is defined THEN
                Map GLTERM shell variable to terminal type via the Systems file
        ELSE
                Map TERM shell variable to terminal type via the Systems file
        ENDIF
ENDIF
```

If the Systems file mapping fails (e.g. TERM=ansi but "ansi" is not defined in a Systems file TERM entry) Global System Manager will prompt for a TERMINAL TYPE (see section 3.2.3).

8.3 Unix Shell Variables for Operator-id Mapping

This section describes those Unix shell variables that affect the Global System Manager operator-id mapping.

8.3.1 GLNAME

The special-purpose GLNAME shell variable is recognised by Global System Manager to override the Unix login name when converting a Unix name to a Global System Manager operator-id using the System file mapping (see section 7.3.5).

Note that both the Unix login name and the GLNAME shell variable may be overridden by the GLID shell variable (see section 8.3.2).

8.3.2 GLID

The special-purpose GLID shell variable is recognised by Global System Manager to override the default operator-id mapping, if any, in the Systems file (see section 7.3.5).

Note that whereas the GLNAME shell variable is set to a "long" Unix login name (e.g. "global"), the GLID shell variable must be set to a 4 character Global System Manager operator-id (e.g. GLOB).

8.3.3 Order of Precedence of Operator-id Mapping Shell Variables

The order of precedence of the GLNAME and GLID shell variables is as follows:

```
If shell variable GLID is defined THEN
        Use GLID directly
ELSE
        IF shell variable GLNAME is defined THEN
                Map GLNAME shell variable to short user name via the Systems file
        ELSE
                Map Unix login name to short user name via the the Systems file
        ENDIF
ENDIF
```

If the Systems file mapping fails (e.g. an OPID or OPIDMAP equivalence is not established) Global System Manager will prompt for an OPERATOR-ID (see section 3.2.2).

8.4 Unix Shell Variables for Printer Handling

This section describes those Unix shell variables that affect Global System Manager printing.

8.4.1 GLSP*nnn*

This Unix shell variable (where *nnn* is a printer unit number from 500 to 599) defines the Unix directory into which the spooled output for the relevant printer unit, for the current SYSTEM is to be sent. If this variable is not defined then the default $GLDIR/spool directory is used instead. It may be useful to keep confidential print files (e.g. reports produced by Payroll) separate from other reports. For example:

```
GLSP510=/usr/payroll/spool510
```

8.5 Unix Shell Variables for $TAPE

This section describes those Unix shell variables that are recognised by the Global System Manager $TAPE controller.

8.5.1 GLTAPE

The special-purpose GLTAPE shell variable MUST be defined in order to use $TAPE on your SYSTEM (see section G.64.1 for further information on $TAPE). For example:

```
GLTAPE=/dev/rmt0
```

Note that the name of the tape device is NOT specified in the Systems file.

8.5.2 GLTAPE1K

The use of the special-purpose GLTAPE1K shell variable is fully described in section G.88.

8.6 Other Unix Shell Variables

This section describes the other Unix shell variables recognised by Global System Manager that are of general use. Further Unix shell variables that may be defined for diagnostic purposes are described in the Troubleshooting Guide (Appendix G).

8.6.1 SHELL

When a <SYSREQ> . is keyed from within Global System Manager a new shell is created. This shell is defined by the shell variable SHELL. If this variable is not defined then the default /bin/sh is used instead. For example, the following use of the SHELL variable will create the C-shell when <SYSREQ> . is keyed:

```
SHELL=/bin/csh
```

The SHELL variable may be set to run any program. For example:

```
SHELL=sysadmsh          Run Unix sysadmsh command
SHELL=gladmin           Run gladmin script
```

Furthermore, this variable may be set to cause an immediate return to Global System Manager if the Unix System Administrator wishes to prevent certain users from executing commands from the shell. For example:

```
SHELL=exit              Return immediately
SHELL=true              Return immediately without error
SHELL=false             Return immediately with error
```

8.6.2 GLIPCBASE

The use of the special-purpose GLIPCBASE shell variable is fully described in section G.53.

8.6.3 GLCONFIG*x*

The use of the special-purpose GLCONFIG*x* shell variable is fully described in section G.41.

8.6.4 GLREMOTE

The use of the special-purpose GLREMOTE shell variable is fully described in section G.71.2.

8.6.5 GLSPD

The use of the special-purpose GLSPD shell variable is fully described in section G.72.1.3.

8.6.6 GSM_*xxxxxx*

The use of the special-purpose GSM_*xxxxxx* diagnostic shell variables is fully described in section G.86.

8.7 Shell scripts

If the Unix shell variables described in this chapter are not defined in a login script (e.g. .profile or .login) it is often convenient to create a small script file which may be executed in order to establish them.

8.7.1 Example Bourne Shell Script

In the Bourne or Korn shells, create a shell script, **gle**, of the form:

```
GLDIR=/usr/global
PATH=$PATH:$GLDIR/bin
export GLDIR PATH
```

The following command:

```
$ . ./gle
```

will establish the variables in the current shell.

8.7.2 Example C-Shell Script

In the C-shell, create a shell script, **gle**, of the form:

```
setenv GLDIR /usr/global
setenv PATH $GLDIR/bin:$PATH
```

The following command:

```
$ source gle
```

will establish the variables in the current shell.

9. Global System Manager Configuration Files

This chapter describes the special considerations that apply when Global Configurator is used to update a Global System Manager (Unix) configuration file.

In general, the controller entries that appear within a Global System Manager (Unix) configuration file refer to corresponding entries in the Systems file (see Chapter 7 and Appendix F). Typically, each entry in the Global configuration file corresponds to a command word in the Unix Systems file.

By default, a single configuration file is used for all the SYSTEM's on a Global System Manager (Unix) installation. Those controllers defined in the configuration file that have no equivalent entry for a particular SYSTEM in the Systems file are dynamically removed from that SYSTEM.

This chapter should be read in conjunction with the Global Configurator Manual and Chapter 7 and Appendix F of this manual. All references to the Global Configurator Manual are of the form CF-*n.n*.

9.1 Machine Name and Bootstrap Messages [CF-3.2]

All Global System Manager V8.1 (Unix) configurations are level 9 (i.e. BOS/XLAN (*sic*) or LEVEL9). The same computer architecture code (C), computer machine code (C2) and computer subcode (1) are used for all the current Global System Manager (Unix) implementations. The MACHINE NAME and BOOTSTRAP MESSAGE (*sic*) are as described in CF-3.2.

9.2 Data File Definitions [CF-3.3]

The DATA FILE DEFINITIONS section of a Global System Manager (Unix) configuration file corresponds to the DIRECT ACCESS CONTROLLERS section of the example configuration file described in the Global Configurator Manual.

The following controller names are allowed:

DATA	Discrete data files
DDATA	Integrated data file
DISKETTE	Diskette drive
RAM DISK	Reserved for future use (do not use)

9.2.1 DATA - Discrete data files (Separated Subunit Domain)

The DRIVE number corresponds to the DATA *number* in the Systems file (see section 7.3.2). The DESCRIPTION is explained in CF-3.3. The VOLUME FORMAT, MAXIMUM NUMBER OF FILES, NUMBER OF SUBUNITS and UNIT NUMBERS are also described in CF-3.3. The VOLUME FORMAT **must** be T151Z or the isometric, but obsolete, T224Z (see section G.63). **The MAXIMUM NUMBER OF FILES must be left at the default value of 250.** The DRIVE parameter must match a DATA entry in the Systems file otherwise the UNIT NUMBER will not be available. The Unix *directory_name* from the corresponding entry in the Systems file will be used to hold the discrete data files. For example, the following entry in the configuration file:

```
CONTROLLER (DATA ) : Discrete data files
      DRIVE ( 0):
```

```
      DESCRIPTION (Discrete data files ) :

VOLUME FORMAT (T151Z ) : SEPARATED SUBUNIT DOMAIN
MAXIMUM NUMBER OF FILES ( 250) :
NUMBER OF SUBUNITS ( 99) :
      UNIT NUMBER ( 200) :
```

corresponds to an entry of the following format in the Systems file:

```
DATA 0 A00.dir
```

Note that although the domain appears as 2*xx* in the configuration file it will be standardly addressed as %*xx*, where % is the corresponding file-server SYSTEM letter (e.g. A00 for SYSTEM A).

If the DESCRIPTION is set to a single "?", "<" or ">" character the corresponding *directory_name* as defined in the Systems file, truncated to 25 characters (if necessary), will appear in $U reports (see section 4.9). Furthermore, the description extracted from the Systems file will appear in all Global System Manager error messages that report a problem with the simulated hard disk. The algorithm used when truncating the string to 25 characters depends on the actual character specified in the DESCRIPTION field in the Global configuration file:

Character	*Truncation algorithm*
<	Display first 25 characters from the text string defined in the Systems file;
>	Display last 25 characters from the text string defined in the Systems file;
?	Display first 12 characters from the text string defined in the Systems file, followed by a "+" character, followed by the last 12 characters from the text string defined in the Systems file.

Discrete data files (with every sub-volume corresponding to a Unix file in the directory specified in the Systems file - see section 7.3.2) should be used in preference to integrated data files (with the entire domain corresponding to the Unix file specified in the Systems file).

The standard configuration files supplied with Global System Manager (Unix) for systems distributed on diskette include a single DATA controller (format T151Z, unit 200, 99 subvolumes, 250 files/subvolume). The standard configuration files supplied with Global System Manager (Unix) for systems distributed on tape include two DATA controllers (format T151Z, unit 200, 59 subvolumes, 250 files/subvolume; format T151Z, unit 260, 39 subvolumes, 250 files/subvolume).

9.2.2 DDATA - Integrated data file (Virtual hard disk domain)

The DRIVE number corresponds to the DATA *number* in the Systems file (see section F.2.14). The DESCRIPTION is explained in CF-3.3. The VOLUME FORMAT, MAXIMUM NUMBER OF FILES, NUMBER OF SUBUNITS and UNIT NUMBERS are also described in CF-3.3. The VOLUME FORMAT **must** be P224Z. The DRIVE parameter must match a DATA entry in the Systems file otherwise the UNIT NUMBER will not be available. The Unix *file_name* from the corresponding entry in the Systems file will be used as the integrated data file. For example, the following entry in the configuration file:

```
CONTROLLER (DDATA ) : Integrated data file
      DRIVE ( 0):
      DESCRIPTION (Integrated data file ) :

VOLUME FORMAT (P224Z ) : UNIX VARIABLE DOMAIN
MAXIMUM NUMBER OF FILES ( 99) :
NUMBER OF SUBUNITS ( 49) :
      UNIT NUMBER ( 200) :
```

corresponds to an entry of the following format in the Systems file:

```
DATA 0 gsm200.vol
```

Note that although the domain appears as 2*xx* in the configuration file it will be standardly addressed as %*xx*, where % is the corresponding SYSTEM letter (e.g. B00 for SYSTEM B).

If the DESCRIPTION is set to a single "?", "<" or ">" character the corresponding *file_name* as defined in the Systems file, truncated to 25 characters (if necessary), will appear in $U reports (see section 4.9). Furthermore, the description extracted from the Systems file will appear in all Global System Manager error messages that report a problem with the simulated hard disk. The algorithm used when truncating the string to 25 characters depends on the actual character specified in the DESCRIPTION field in the Global configuration file:

Character	*Truncation algorithm*
<	Display first 25 characters from the text string defined in the Systems file;
>	Display last 25 characters from the text string defined in the Systems file;
?	Display first 12 characters from the text string defined in the Systems file, followed by a "+" character, followed by the last 12 characters from the text string defined in the Systems file.

INTEGRATED DATA FILES SHOULD BE CONSIDERED OBSOLETE. A DDATA controller should only be added to a Global System Manager (Unix) configuration file in order to transfer data to discrete data files from earlier implementations of Global System Manager (Unix) - see section F.2.14.

9.2.3 DISKETTE - Diskette drive

The DRIVE number corresponds to the DISKETTE number in the Systems file (see section 7.3.3). The DESCRIPTION is explained in CF-3.3. The VOLUME FORMAT and UNIT NUMBERS are also described in CF-3.3. The DRIVE and VOLUME FORMAT parameters must match a DISKETTE entry in the Systems file otherwise the UNIT NUMBER will not be available. The Unix *device_name* from the corresponding entry in the Systems file will be used to access the diskette. For example, the following entry in the configuration file:

```
CONTROLLER (DISKETTE) : Diskette drive
      DRIVE ( 0):
      DESCRIPTION (Diskette drive 0 ) :

VOLUME FORMAT ( ) :O2A 3" IBM HIGH CAPACITY
      UNIT NUMBER ( 140) :

VOLUME FORMAT ( ) :B3B 3" APRICOT DS,DD (80T)
      UNIT NUMBER ( 170) :
```

corresponds to an entry of the following format in the Systems file:

```
DISKETTE
    0     O2    /dev/rfd0135ds18
    0     B3    /dev/rfd0135ds9
```

Note that a full "ANA" code must be supplied as the configurator VOLUME FORMAT whereas only the "AN" part of the code appears in the Systems file. Volume formats with the same diskette geometry that differ only in directory layout (e.g. O2A and O2B) are accessed using the same Unix raw device. Note also that although the diskette format appears as 1*xx* in the configuration file it will be standardly addressed as %*xx*, where % is the lower-case equivalent of the upper-case file-server SYSTEM letter (e.g. c40 for SYSTEM C).

All devices for the same physical drive must be defined in the Systems file as the same DISKETTE number on the same SYSTEM otherwise the Global System Manager resource locking mechanism may be compromised.

If the DESCRIPTION is set to a single "?", "<" or ">" character the corresponding *device_name* as defined in the Systems file, truncated to 25 characters (if necessary), will appear in $U reports (see section 4.9). Furthermore, the description extracted from the Systems file will appear in all Global System Manager error messages that report a problem with the diskette. The algorithm used when truncating the string to 25 characters depends on the actual character specified in the DESCRIPTION field in the Global configuration file:

Character	*Truncation algorithm*
<	Display first 25 characters from the text string defined in the Systems file;
>	Display last 25 characters from the text string defined in the Systems file;
?	Display first 12 characters from the text string defined in the Systems file, followed by a "+" character, followed by the last 12 characters from the text string defined in the Systems file.

The standard configuration files supplied with Global System Manager (Unix) for systems distributed on 5¼" diskettes include a single DISKETTE controller with the following formats:

```
C24B
G1A
G1B
```

The standard configuration files supplied with Global System Manager (Unix) for systems distributed on 3½" diskettes include a single DISKETTE controller with the following formats:

```
B3B
B3C
O2B
O2D
```

The standard configuration files supplied with Global System Manager (Unix) for systems distributed on QIC tape also include a single DISKETTE controller with the following formats:

B3B
B3C
O2B
O2D

The standard configuration files supplied with Global System Manager (Unix) for systems distributed on DAT tape also include a single DISKETTE controller with the following formats:

B3B
B3C
O2B
O2D

9.3 User Display Attributes [CF-3.4]

The USER DISPLAY ATTRIBUTES section of a Global System Manager (Unix) configuration file corresponds to the CONSOLE CONTROLLERS section of the example configuration file described in the Global Configurator Manual.

Only the following mandatory controller name is allowed:

USER Display attributes

9.3.1 USER - Display Attributes

The TYPE AHEAD BUFFER LENGTH, DISPLAY BUFFER LENGTH, FUNCTION KEY BUFFER LENGTH, SCREEN IMAGE WIDTH, SCREEN IMAGE DEPTH, NUMBER OF STORED ATTR' BYTES, NUMBER OF VIRTUAL PARTITIONS, CHARACTER TRANSLATION ENABLED and CONSOLE EXECUTIVE FLAG BYTE are all described in CF-2.4 (*sic*).

By default, "Direct displays" are **enabled** (i.e. so that the *glintd* process writes characters directly to the Unix device driver). Note that for versions of Global System Manager prior to V8.1, "Direct displays" are disabled (i.e. so the *glintd* process uses the *global* process to display characters). Enabling "Direct displays" normally improves the screen performance when running Global System Manager (Unix). Direct displays may be disabled by setting this flag to N. Direct displays are automatically disabled, over-ridding the configuration file setting, on multi-USER SYSTEM's or if the USER is attached to SYSTEM-A.

The "User number" must match a USER number for a SYSTEM in the Systems file (see section 7.3.1). For example, the following entry in the configuration file:

```
CONTROLLER (USER ) : Display attributes
      TYPE AHEAD BUFFER LENGTH      ( 100):
      DISPLAY BUFFER LENGTH         ( 500):
      FUNCTION KEY BUFFER LENGTH    ( 0):
      SCREEN IMAGE WIDTH            ( 132):
      SCREEN IMAGE DEPTH            ( 25):
      NUMBER OF STORED ATTR' BYTES  ( 1):
      NUMBER OF VIRTUAL PARTITIONS  ( 4):
      CHARACTER TRANSLATION ENABLED (Y):
      CONSOLE EXECUTIVE FLAG BYTE   (#00):
      Direct displays               (Y):
      User number                   ( 1):
```

corresponds to an entry of the following format in the Systems file:

```
USER  1     LOGNAME     aju
```

The LOGNAME command word in the Systems file is used to map a Unix login-name to a system-letter/user-number combination (see section 7.3.1). The TTY command word in the Systems file is used to map a Unix console-device to a system-letter/user-number combination (see section 7.3.1).

Adding extra users to the configuration file in order to increase the number of users on a Global System Manager (Unix) implementation is NOT recommended (see section F.2.11). The number of users should be increased by adding extra SYSTEM's to the Systems file (see section F.2.5).

However, it is necessary to add extra USER's to the configuration file in order to obtain SYSTEM's with different "user attributes" (e.g. if different users require different number of partitions).

For example, if the configuration file contains the following entries:

User number	*Number of virtual partitions*
1	4
2	3
3	2
4	1

and the Systems file contains the following entries:

```
SYSTEM    0x1b  USER 1     LOGNAME   global4
SYSTEM    0x1c  USER 2     LOGNAME   global3
SYSTEM    0x1d  USER 3     LOGNAME   global2
SYSTEM    0x1e  USER 4     LOGNAME   global1
```

then the following Unix users would be allocated the following numbers of virtual partitions:

Login name	*System number*	*Number of virtual partitions*
global4	0x1b	4
global3	0x1c	3
global2	0x1d	2
global1	0x1e	1

The standard configuration files supplied with Global System Manager (Unix) include a single USER controller.

9.4 Tape Controller [CF-3.5]

The TAPE CONTROLLER section of a Global System Manager (Unix) configuration file corresponds to the TAPE CONTROLLER section of the example configuration file described in the Global Configurator Manual.

Only the following controller name is allowed:

TAPE	Tape backup/restore

No controller options are required. The TAPE controller entry in the configuration file is optional - it is not used by $TAPE (see section G.64). **Important note:** $TDUMP is NOT supported with Global System Manager (Unix).

The standard configuration files supplied with Global System Manager (Unix) include a single TAPE controller.

Note that the TAPE CONTROLLER section of the configuration file will not appear when using Global Configurator with a version of A.C2 dated 27-Nov-91, or earlier.

9.5 Printer Attributes [CF-3.6]

The PRINTER ATTRIBUTES section of a Global System Manager (Unix) configuration file corresponds to the PRINTER CONTROLLERS section of the example configuration file described in the Global Configurator Manual.

The following controller names are allowed:

DIRECT	Direct printer output
SPOOLED	Spooled printer output

9.5.1 DIRECT - Direct Printer Output

The UNIT NUMBER, DESCRIPTION, HARDWARE FORM FEED, MAXIMUM PAGE DEPTH, TIME-OUT IN TENS OF SECONDS and PRINTER EXECUTIVE FLAG BYTE are all described in CF-3.6. The SPOOLER CONTROL BITS and DEVICE CHARACTERISTICS documented in CF-3.6 are not appropriate for Global System Manager (Unix) printer handling and will not appear when running Global Configurator.

The UNIT NUMBER parameter must match a PRINTER number in the Systems file otherwise the printer unit will not be available. The corresponding PRINTER entry in the Systems file **must** include a *device_name* (see section 7.3.6.1). This *device_name* will be used to access the printer device. For example, the following entry in the configuration file:

```
CONTROLLER (DIRECT ) : Direct printer output
      UNIT NUMBER ( 500):
      DESCRIPTION (? ) :
      HARDWARE FORM FEED            (Y):
      MAXIMUM PAGE WIDTH            ( 132):
      TIME-OUT IN TENS OF SECONDS   ( 2):
      SPOOLER CONTROL BITS          (#00):
      PRINTER EXECUTIVE FLAG BYTE   (#FF):
      Maximum FIFO buffering/256    ( 0):
      Print direct without a FIFO?  (N):
```

corresponds to the following entry in the Systems file:

```
PRINTER     500   /dev/tty1a
```

If the DESCRIPTION is set to a single "?", "<" or ">" character the corresponding *device_name* as defined in the Systems file, truncated to 25 characters (if necessary), will appear in $U reports (see section 4.9). Furthermore, the description extracted from the Systems file will

appear in all Global System Manager error messages that report a problem with the printer (e.g. NOT READY ERROR). The algorithm used when truncating the string to 25 characters depends on the actual character specified in the DESCRIPTION field in the Global configuration file:

Character	*Truncation algorithm*
<	Display first 25 characters from the text string defined in the Systems file;
>	Display last 25 characters from the text string defined in the Systems file;
?	Display first 12 characters from the text string defined in the Systems file, followed by a "+" character, followed by the last 12 characters from the text string defined in the Systems file.

The "Maximum FIFO buffering/256" parameter can be used to limit the number of characters placed in the printer FIFO (see section G.92). If this value is left at 0 the default Unix FIFO size, normally 8Kb, will be used.

The "Print direct without a FIFO?" flag can be used to modify the Global System Manager Direct printer interface as described in section G.93.

The standard configuration files distributed with Global System Manager (Unix) include 5 DIRECT printer controllers (units 500 to 504). Up to 50 DIRECT printers can be included in a single configuration file.

9.5.2 SPOOLED - Spooled Printer Output

The UNIT NUMBER, DESCRIPTION, HARDWARE FORM FEED, MAXIMUM PAGE DEPTH, TIME-OUT IN TENS OF SECONDS and PRINTER EXECUTIVE FLAG BYTE are all described in CF-3.6. The SPOOLER CONTROL BITS (*sic*) and DEVICE CHARACTERISTICS documented in CF-3.6 are not appropriate for Global System Manager (Unix) printer handling and will not appear when running Global Configurator.

The UNIT NUMBER parameter must match a PRINTER number in the Systems file otherwise the printer unit will not be available. The corresponding PRINTER entry in the Systems file **must** include a *shell_command* enclosed in double-quotes (see section 7.3.6.2). This *shell_command* will be executed in order to expedite the print.

For example, the following entry in the configuration file:

```
CONTROLLER (SPOOLED ) : Spooled printer output
      UNIT NUMBER ( 510):
      DESCRIPTION (? ) :
      HARDWARE FORM FEED            (Y):
      MAXIMUM PAGE WIDTH            ( 132):
      TIME-OUT IN TENS OF SECONDS   ( 2):
      SPOOLER CONTROL BITS          (#00):
      PRINTER EXECUTIVE FLAG BYTE   (#FF):
```

corresponds to the following entry in the Systems file:

```
PRINTER     510   "lp -rxv"
```

If the DESCRIPTION is set to a single "?", "<" or ">" character the corresponding *shell_command* as defined in the Systems file, truncated to 25 characters (if necessary), will appear in $U reports (see section 4.9). Furthermore, the description extracted from the Systems file will appear in all Global System Manager error messages that report a problem with the printer (e.g. NOT READY ERROR). The algorithm used when truncating the string to 25 characters depends on the actual character specified in the DESCRIPTION field in the Global configuration file:

Character	*Truncation algorithm*
<	Display first 25 characters from the text string defined in the Systems file;
>	Display last 25 characters from the text string defined in the Systems file;
?	Display first 12 characters from the text string defined in the Systems file, followed by a "+" character, followed by the last 12 characters from the text string defined in the Systems file.

The standard configuration files supplied with Global System Manager (Unix) include 5 SPOOLED printer controllers (units 510 to 514). Up to 50 SPOOLED printers can be included in a single configuration file.

9.6 Transport Layer [CF-3.7]

The TRANSPORT LAYER section of a Global System Manager (Unix) configuration file corresponds to the LAN CONTROLLERS section of the example configuration file described in the Global Configurator Manual.

Only the following mandatory controller name is allowed:

SM System Manager

No controller options are required. All Global System Manager (Unix) configuration files **MUST** include a single SM controller.

9.7 Nucleus Options [CF-2.9]

The NUCLEUS OPTION section of a Global System Manager (Unix) configuration file corresponds to the NUCLEUS OPTION section of the example configuration file described in the Global Configurator Manual.

The following prompts are described in CF-2.9:

```
SET BTFLAG TO #72 FOR Global System Manager V8.1(#72)
DYNAMIC DC/DF-BLOCK ALLOCATION(Y)
DYNAMIC LOCK TABLE ALLOCATION (Y)
LARGEST SECTOR SIZE
NUMBER OF FILE CHANNELS
NUMBER OF FILE BUFFERS
NUMBER OF FILE BLOCKS
NUMBER OF LOCK TABLE ENTRIES
NUMBER OF PRINT BUFFERS
LENGTH OF PRINT BUFFERS
NUMBER OF PRINT XLATION BUFFERS
MAXIMUM MEMORY ALLOCATION
TARGET STARTUP STRATEGY
```

The NUMBER OF PRINT BUFFERS is set to 20 in all standard configuration files distributed with Global System Manager V8.1. Note that earlier versions of Global System Manager were distributed with configuration files containing fewer (i.e. 6) printer buffers.

The NUMBER OF XLATION BUFFERS parameter is not used by Global System Manager (Unix) configurations. The number of printer translation buffers allocated is always equal to the total number of printers defined in the configuration file (i.e. 10 in all standard configuration files distributed with Global System Manager).

The MAXIMUM MEMORY ALLOCATION should be set to 0.

The $REMOTE SUPPORTED? flag should be left at Y. The $REMOTE options are fully described in section G.71.1. Note that versions of Global System Manager (Unix) prior to V8.1 were distributed with configuration files that did not support the $REMOTE option.

The "System Manager" and "Auto date/time sign-on" flags **MUST** be set to Y.

The "Operations between displays" count should be set to 256.

The following 4 parameters are only relevant if the Unix Universal Channel Interface (UCI) is to be used. The Unix UCI, which is required to access C-ISAM format Speedbase databases; C-ISAM format RSAM, ISAM and DMAM files and by the Global File Converter product, is fully described in Chapter 7 of the Global File Converters Manual.

```
Maximum C-ISAM record size    ( 6000):
Number of UCI channels        ( 200):
Max. no. of UCI Unix opens    ( 10):
Max. no. of UCI/RS Unix opens ( 10):
```

As explained in Chapter 7 of the Global File Converters Manual, one of the functions of the UCI is to map a large number of UCI channels to a smaller number of open C-ISAM files. For example, a single C-ISAM format Speedbase database will actually consist of several C-ISAM files. This UCI functionality is necessary because, typically, a single Unix process can only open a relatively small number of Unix files - far fewer than the number of open C-ISAM files required by a *glintd* process that is executing a Global Speedbase application (e.g. Global 3000) when the Speedbase database is held as a collection of C-ISAM files.

The "Maximum C-ISAM record size" must be at least as large as the size of the largest record in any C-ISAM file accessed via the UCI. The default value of 6000 should be adequate for most requirements.

The "Number of UCI channels" is the number of "logical channels" allocated by the UCI for each SYSTEM. For example, consider a SYSTEM with a single User Definition configured for 9 concurrent partitons. If each partition on the SYSTEM is running a Speedbase application that accesses a C-ISAM format Speedbase database with 80 record sets, a total of 720 (i.e. 9 * 80) UCI channels will be required.

The "Max. no. of UCI Unix opens" determines the maximum number of Unix files kept open simultaneously by the UCI for "normal" UCI operations. Increasing this value may increase the performance of C-ISAM format Speedbase database access, especially database rebuilds, because fewer Unix open/close operations are required. However, because several Global System Manager modules, other than the UCI, maintain a pool of open Unix files (e.g. SVC-61,

the data file SSD controller, the direct printer controller, the spooled printer controller) the actual number of files opened by Global System Manager is very difficult to predict. **Great care must be exercised when increasing this value.** Severe problems can occur if any module within the *glintd* process, especially the UCI, causes the Unix limit of the "maximum number of open files per process" to be exceeded. See section G.101 for a full description of these problems.

The "Max. no. of UCI/RS Unix opens" determines the maximum number of Unix files kept open simultaneously by the UCI for "special" RS file operations (see Chapter 7 of the Global File Converters Manual for further details). The same considerations, described above, regarding the number of simultaneously open Unix files, apply to this seldom-used parameter.

9.8 Distribution Options [CF-3.9]

The DISTRIBUTION OPTION section of a Global System Manager (Unix) configuration file corresponds to the DISTRIBUTION OPTION section of the example configuration file described in the Global Configurator Manual.

You should not attempt to make any changes to this section. The following configuration file parameters are only used when the software is being generated:

BACNAT format

Number of native diskettes

C-ISAM supported

10. Speedbase C-ISAM Database Utility

This chapter describes the extension to Speedbase Presentation Manager that is available on Global System Manager (Unix) configurations. In particular, it describes the utility that allows you to hold a Speedbase database as a collection of Informix C-ISAM files.

Note that the "C-ISAM format Speedbase database" option is not available on all Global System Manager (Unix) configurations. Please refer to your Global Configuration Notes for further details regarding the availability of C-ISAM on your configuration.

10.1 Introduction to the Speedbase C-ISAM Database Utility

The Speedbase C-ISAM Database Management Utility is used to perform the following operations:

- creation of C-ISAM format Speedbase databases;
- rebuilding the indexes of C-ISAM format Speedbase databases;
- transferring data from a "traditional" Global format Speedbase database to a C-ISAM format Speedbase database;
- dumping and regenerating C-ISAM format Speedbase databases;
- deletion of unwanted C-ISAM format Speedbase databases.

A C-ISAM format Speedbase database consists of a data dictionary file and a schema file, both stored as Global format files, and in addition, a special index file and several data-files and index files stored in the Unix filing system. The names of the files are described in Table 10.1, where *xxxxx* is the database name and *rt1* to *rtn* are up to thirty-six record types.

The data dictionary is created by the application developer using the Speedbase Development System dictionary utility. The five character database name *xxxxx* is specified when the database is created using the Speedbase C-ISAM database management utility (see section 10.2). The C-ISAM data-files, which each store the data of a specific record type, are adjusted in size automatically. There is no need to re-allocate the data-files to provide additional space, as is occasionally required for "traditional" Global format Speedbase databases.

To run the Speedbase C-ISAM database utility select option 8 from the main Speedbase Presentation Manager menu. The Speedbase C-ISAM database utility may be run directly, by keying $BADB to the main menu or GSM READY: prompt.

The six menu options are described in the following sections.

10.2 Create C-ISAM Format Speedbase Database

To create a new C-ISAM format Speedbase database, select option 1 from the main menu. The C-ISAM Database Creation Window will be displayed.

Enter the dictionary name and its unit number. The utility displays the generation number, creation date and title of the dictionary. You should now enter the five-character database

name and the unit on which you wish to store the Global format schema file and new Global format dictionary file.

Important note: The name of the new C-ISAM format Speedbase database MUST be 5 characters.

Two warnings may be displayed at this point in the dialogue. If a database of the same name already exists on the specified unit, a warning pop-up will alert you to this fact. The pop-up will ask you to confirm that you want to delete this pre-existing database before proceeding. If you confirm this prompt then the database will be deleted in its entirety (including all associated C-ISAM files).

File-id	Description	Type
DI*xxxxx*	Data Dictionary	Global
DB*xxxxx*	Schema File	Global
DB*xxxxx*	Special Index File	Unix
DB*xxxxxrt1*.dat	C-ISAM Datafile *rt1*	Unix
DB*xxxxxrt1*.idx	C-ISAM Index file *rt1*	Unix
DB*xxxxxrt2*.dat	C-ISAM Datafile *rt2*	Unix
DB*xxxxxrt2*.idx	C-ISAM Index file *rt2*	Unix
DB*xxxxxrtn*.dat	C-ISAM Datafile *rtn*	Unix
DB*xxxxxrtn*.idx	C-ISAM Index file *rtn*	Unix

Table 10.1 - Speedbase C-ISAM Database Files

If a dictionary of the same name exists on the specified unit, then a pop-up will again alert you to this. You will be asked to confirm deletion of the dictionary before proceeding.

You then enter the Unix directory path for the database data-files (e.g. /usr/cisam). The current working directory is shown, and you may specify the required directory path relative to this. The utility then checks the Unix directory to ensure that there are no files that would be overwritten by the creation of the database. If files are detected, the following prompt will appear:

```
Database already exists on this directory - Delete?
```

If you reply Y any Unix files having the same file names as the new database will be over-written. Once the above warnings, if any, have been dealt with, the Informix Selection window will be displayed.

This window allows you to update a set of optional Informix files, which must have been generated earlier using the appropriate Informix tools. This facility automatically transfers information about the structure and contents of the database files so that the data stored in these files may be accessed directly by Informix.

If you wish to use Informix to access your data, you should confirm the prompt, and supply the Unix path to the Informix database (i.e. the directory on which the Informix dictionary files are stored). Note that the use of relative path names is NOT permitted in this window, and that the path name must end with the suffix ".dbs".

In the event that any of the Informix database files are missing or unable to be opened, then an error pop-up will alert you of the problem. In this event you should correct the specified error, and re-run the creation operation.

Assuming that all of the required Informix files are correctly present on the designated directory, the Speedbase C-ISAM Database utility will then update these, thus completing the database creation process.

10.3 Rebuild C-ISAM Format Speedbase Database

The Speedbase C-ISAM database utility rebuild option is used to perform two main functions. It may be used to re-organise the special index file, known as a **partial rebuild**. Alternatively it may be used to re-establish all linkages within the database, known as a **total rebuild**, which also re-organises the special index file.

10.3.1 Partial Database Rebuild

A partial database rebuild is essentially a reorganisation function. It causes all indexes stored in the special index file to be re-created, which guarantees index accuracy. It usually results in a small but noticeable performance improvement, and increases the size of the free index area reserved for the addition of records to the database.

This function may therefore be used to increase the number of free index blocks as an alternative to permanently expanding the size of the database.

10.3.2 Total Database Rebuild

A total database rebuild is generally used as a recovery procedure following either hardware or software failure. A total rebuild guarantees the integrity of the information stored on the database by reconstructing both the special index file and the linkages that connect related records. For example, the total rebuild of a database containing customer and invoice records would firstly recreate all indexes to both of these record types. Each invoice record would then be checked to ensure its linkage with the correct customer record.

A total rebuild will generally also recalculate accumulators as part of this process. In the above example, the customer account balance would be recalculated to ensure that it is equal to the value of all related invoices.

This process goes beyond a simple checking. In the event that an error is detected (e.g. an invalid account balance, or an invoice referencing to the wrong customer account) it is automatically corrected. The operator is only notified when the error cannot be resolved, such as an invoice record referencing a non-existent customer.

The type of errors discussed above can only arise from certain hardware malfunctions (e.g. a power loss) or from extremely serious application software problems. The utility guarantees the internal integrity of the database following such errors, but some application programs may require further processing to be performed.

The rebuild option operates by examining the data records physically stored in the database files. This means that all information up to and including the last successfully completed database update can normally be rebuilt. The option cannot be used, however, if the disk on which the database resides is lost or develops permanent errors. In this event the only option is to restore a backup copy of the database.

During normal use, Speedbase Presentation Manager checks the internal integrity of the database, and will notify you if any inconsistencies are found. If this occurs, a total rebuild should be performed. Note that the Speedbase C-ISAM Database utility rebuild option does not carry out any operations on the C-ISAM index files associated with each datafile in the database. These index files are provided in order to facilitate access to the data-files by Unix programs other than Speedbase applications.

10.3.3 Rebuild C-ISAM Database

To rebuild a C-ISAM format Speedbase database select option 2 from the main menu. Specify the name and unit of the schema file. The C-ISAM Database Rebuild window will be displayed.

Reply to the total rebuild prompt as required. The Speedbase C-ISAM database utility displays each record number as the rebuild proceeds.

10.4 Load C-ISAM Format Speedbase Database

A "traditional" Global format Speedbase database may be loaded into a C-ISAM format Speedbase database. To load a newly created C-ISAM format Speedbase database with data from a Global format Speedbase database, select option 3 from the main menu. The Speedbase C-ISAM database utility displays the Source Database Details window.

Specify the name and unit of the Global format Speedbase database. The Target C-ISAM Database window will be displayed.

Reply with the name and unit of the schema file you specified in the database creation phase (see section 10.2) and confirm that the transfer is to proceed. The Speedbase C-ISAM database utility displays the transfer details pop-up as records are transferred from the Global format Speedbase database to the new C-ISAM format Speedbase database.

Important note: You may only transfer data to a newly-created C-ISAM format Speedbase database. You cannot use this function to transfer data to a database containing any data records. If you wish to re-transfer data you should create the C-ISAM format Speedbase database afresh, deleting the previous version, and transfer data from the Global format Speedbase database to it. **Any data present in the C-ISAM format Speedbase database but not present in the Global format Speedbase database would be lost in this process**.

10.5 Dump C-ISAM Format Speedbase Database

This option is used to write the data contained in a C-ISAM format Speedbase database to a number of "Dump" files. These files are created in an industry standard format which allows you to move the data contained in your database from version of Unix to another. For example, from an RS/6000 running AIX 3.2.5 to SCO Unix. The Dump option is also required as the first step when converting a database from one generation to another, and may also be used to move the database from one Unix directory to another.

To dump a C-ISAM format Speedbase database, select option 4 from the main menu. You must then enter the Schema file name and unit, after which the C-ISAM Database Dump window is displayed.

The window includes the Unix directory path of the database, as well as the current working directory. You must now specify the Unix directory path on which the dump files are to be created.

Having specified this, the utility then displays the Delete Option pop-up.

This option allows you to delete the database files **during the course of the dump process**. If you confirm this prompt, the C-ISAM files will be progressively deleted as each file is dumped. Note that this option causes considerably less disk to be used during the dumping process. However, should the utility fail to complete normally, it will be necessary to restore the database. **A current backup is therefore essential before using the delete option.**

Once a reply to the "delete" prompt has been given, the Speedbase C-ISAM database utility proceeds to dump the designated database. This results in creation of a number of Unix files in the specified directory. These files are:

DI*xxxxx*	Dump of Database Dictionary File
DB*xxxxx*RT*n*	Dump of Unix C-ISAM file record type *n*.

Note that the dump files are all inter-related, and must therefore be restored as a set. **Under no circumstances can the files from one dump be interchanged with files from another later or earlier dump.**

Once the dump process has completed the utility returns to the main menu.

10.6 Regenerate C-ISAM Format Database

This option is used to reload and/or convert a database that was previously dumped using the facility described in section 10.5. To regenerate your database select option 5 from the main menu. The C-ISAM Database Regeneration Source window is displayed.

You should now enter the database name and directory location of the previously dumped database. The current working directory is displayed, and you may specify the dump file directory relative from this. The C-ISAM Database Regeneration Destination window is displayed.

You now enter the database name and unit where the database is to be generated.

Two warnings are possible at this point. If a database of the same name already exists on the specified unit, a warning pop-up will alert you to this fact. The pop-up will ask you to confirm that you want to delete this pre-existing database before proceeding. If you confirm this prompt then this database will be deleted in its entirety (including all associated C-ISAM files).

If a dictionary of the same name exists on the specified unit, then a pop-up will again alert you to this. You will be asked to confirm deletion of the dictionary before proceeding.

You then enter the Unix directory path for the data files of the new database (e.g. /usr/cisam). The current working directory is shown, and you may specify the required directory path as a relative path to this.

The utility then checks the specified Unix directory to ensure that there are no files that would be overwritten by the creation of the database. If files are detected, the following prompt will appear:

```
Database already exists on this directory - Delete?
```

If you reply Y to this prompt any Unix files having the same file names as the new database will be over-written by the creation phase.

If the database is to be converted to a new generation (version), then you may now specify the dictionary to be used in the conversion process. Enter "Y" to the Convert prompt, and enter the name and unit of the dictionary to which the database is to be converted. Having done this the previously dumped data will be restructured during the regeneration process.

The utility then creates the necessary Unix C-ISAM files on the specified directory. During this process the Informix Selection Window previously described in section 10.2 is displayed. This window optionally allows you to update Informix dictionary tables to allow direct access to the new database.

The utility then displays the C-ISAM Regeneration Delete Option pop-up.

This option allows you to delete the dump files **during the course of regeneration**. If you confirm this prompt, the dump files will be progressively deleted as each file is reloaded. Note that this option causes considerably less disk to be used during the dumping process. However, should the utility fail to complete normally, it will be necessary to restore the dump files. A current backup of the dump files is therefore essential before using this option.

The utility then asks you to confirm that you wish to proceed with the regeneration, after which the data is reloaded from the dump files.

On completion, the utility returns to the main menu.

10.7 Delete C-ISAM Format Speedbase Database

This option allows you to delete a C-ISAM format Speedbase database. To delete a database, enter 6 at the main menu. The C-ISAM Database Deletion window is displayed.

Enter the database name and unit of the database schema. The utility displays the generation number creation date and title of the database.

To delete the database respond "Y" to the confirmation prompt. The utility will then delete all Global format and Unix files associated with the database.

On completion the utility returns to the main menu.

10.8 $BADB User Notes

This section contains some miscellaneous points that should be considered when using $BADB to create or load C-ISAM format Speedbase databases.

10.8.1 Database Name Must be 5 Characters

When creating a C-ISAM format Speedbase database, the name of the database MUST be exactly 5 characters. This can lead to problems when converting some of the Global-3000 databases (e.g. DBCL). In order to avoid the problem, use a temporary name (e.g. DBCLTMP) when creating and loading the database on the schema unit. Rename the resultant schema file (and the Data Dictionary, see section 10.8.2) back to the name expected by Global-3000 (e.g. DBCL) after the data has been transferred.

10.8.2 Dictionary Copied

The Data Dictionary is now automatically copied to the same unit as the Global format schema file when $BADB is used to create a new database. Earlier versions of $BADB, did not attempt to copy the Data Dictionary.

10.8.3 Converting a C-ISAM format Speedbase Database to Global

The V8.1 version of $BADGN (documented in the Global System Manager Manual), unlike earlier versions, can be used to generate a Global format Speedbase database from a C-ISAM format Speedbase database.

10.9 C-ISAM Format Speedbase Databases - Special Considerations

This section contains some miscellaneous points that should be considered when using C-ISAM or Informix utilities to access the C-ISAM files that constitute a C-ISAM format Speedbase database.

10.9.1 Updating Speedbase C-ISAM Files

The C-ISAM files that constitute a C-ISAM format Speedbase database can be accessed by programs using C-ISAM or Informix SQL, subject to some restrictions. For reading data there is no restriction on access. However, if you wish to update the files, the recommended method is to use one of the batch posting interfaces available with the particular application. If no suitable posting interface is available, a Speedbase program can be written to effect the updates. It is however, possible to apply the updates directly using C-ISAM or Informix, subject to the following restrictions:

- records can only be updated, not added or deleted;
- no changes may be made to fields that are part of any Speedbase index;
- no changes may be made to any GVF (i.e. to any field which is accumulated into a GVA);
- no changes may be made to any of the fields maintained by Speedbase (i.e. a GVA or one of the $ fields);
- no updates may be done while the database is also being updated by a Speedbase application (i.e. the Unix utility should gain exclusive access to the C-ISAM files);

For bulk data loading or updating, it is possible to create or amend the files using a C-ISAM utility and then use the Speedbase database rebuild option (see section 10.3.2) to reconstruct all the indexes, inter-record linkages and GVA values. However, use of a Speedbase program to update the data is strongly recommended since it avoids all the above restrictions.

10.9.2 Indexes on Speedbase C-ISAM Files

Speedbase creates and uses C-ISAM databases with a primary index in record number sequence. Speedbase creates no other C-ISAM indexes but does not prevent the addition of other indexes by non-Speedbase applications. Speedbase provides no method for creating such indexes. The Speedbase indexes are held in a non C-ISAM format file in the same Unix directory as the C-ISAM database files. The Speedbase indexes are more sophisticated than the C-ISAM indexes and contain some relational information. The Speedbase index file must not be corrupted. It should only be updated by the Speedbase Database Manager from within Global System Manager.

The C-ISAM format Speedbase database is available for access by non-Global applications. Indexes of any type may be added to the database using the C-ISAM *isaddindex* function, and records may be read on these indexes. Note that it is also possible to add and/or delete records as well, but if you do, the Speedbase index will not be updated and the database will be corrupted. This can be remedied by performing a full index rebuild using $BADB (see section 10.3.2) before attempting any database access via Speedbase. If records are updated, then if any field indexed by Speedbase is changed, an index rebuild will be required.

Note that records added to the C-ISAM database other than via Speedbase, could have perfectly valid indexes but contain data which is not correct for the application concerned. This would not cause any problems with the index rebuild, but could cause problems later when running applications which access the relevant database**. It is therefore the responsibility of the user to check such data before adding it to a C-ISAM format Speedbase database**.

Appendix A - Installation Notes

A.1 Introdcution

These notes refer to the prompts and messages produced during the installation of Global System Manager, as indicated by references of the form [A.17], for example. Note that messages which start $57, $66, $78 or $99 are listed separately in Appendix B. Note also that messages of the form:

program_name: number and error message

are listed separately in Appendix C.

These notes should be used for reference only, as not all of them will apply to your installation. Chapter 2 describes the installation of Global System Manager in detail.

A.2 Standard Installation

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.3 Type of Installation

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.4 No Valid SYSRES Unit

This message indicates that Global System Manager cannot be installed because there is no suitable simulated volume. At least one hard disk simulated volume is required to install Global System Manager.

This message will appear if the $GLDIR/data/A00.dir directory is inaccessible or missing.

A.5 Selecting a Unit

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.6 Disk Requires Formatting

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.7 Disk Will be Formatted - Is This OK?

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.8 Overwrite Existing Global System?

There is already a copy of Global System Manager installed on the simulated volume. If you reply Y the existing SYSRES will be overwritten, otherwise Global System Manager will be installed onto a separate sub-volume on the selected simulated volume.

Note that if you want to have two loadable copies of Global System Manager you must use separate discrete data file simulated volumes.

A.9 Destroy Volume-Name?

You are attempting to overwrite a simulated volume on a hard disk which is being used for some other purpose. You must key Y to confirm that this is what you intend.

A.10 Specify SYSRES Unit

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.11 Which Disk To Bootstrap From

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.12 Domain *xxxxxx* will be initialised

This message indicates that the simulated volume requires initialising before it can be used. Initialising the simulated domain will destroy any data currently on the disk. **Note that the default reply to the following prompt is N to prevent accidental deletion of Global System Manager data:**

```
This will DESTROY any existing data
Key Y if it is OK to continue (N):
```

A.13 Computer Identification Letter

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.14 System Unit On Master

Global System Manager must have a designated 'master' SYSTEM, where details of users (i.e. operators) currently using Global System Manager can be held. **This system must always be SYSTEM A.**

You must specify the unit address upon which Global System Manager occupies (SYSRES). This is usually A01.

A.15 Terminal Code

If your computer has serially connected screens then you must specify what type they are, so that the appropriate information can be installed. Each type of screen is identified by a number of up to three digits, and these can be listed by keying L to the prompt. You can specify up to four terminal types. When you have specified all you need key E. Note that it is easy to install further terminal types when the system has been installed by running the $CUS System Maintenance option.

A.16 Memory Allocation

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.17 Insufficient Space

There is insufficient contiguous free space in your filing system for the discrete data file to install Global System Manager. Normally this can only happen if you're re-installing Global System Manager. Increase the amount of free space available in the filing system. The error message indicates how much space is needed.

If there is enough space for the command programs but not for all the user partitions required, then you will be given the option of installing a restricted system, which will support fewer partitions. Because the size calculations are slight overestimates it is possible that this system may be completely satisfactory. If, however, there is not enough space you will be warned that the number of users is being reduced when you run Global System Manager. If you then use $REORG to increase the size of the SYSRES unit by a suitable amount it will be possible for all users to run Global System Manager.

A.18 Existing Unit Too Large

There is more space allocated to the unit onto which Global System Manager is to be installed than is needed. If you reply Y the unit will be re-allocated to release the excess space, otherwise it will be left at its current size. You may want to have a larger than standard SYSRES if extra menus are required.

A.19 Diskettes Required

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.20 Key size of SYSRES in K (minimum of *xxxx*K)

The size of the SYSRES unit may be increased from the default to allow for extra menus, or to provide room for additional swap files.

Note that the reply is automatically multiplied by 1024 to convert to Kbytes so that, for example, a reply of 6000 would be converted to 6144000 bytes (i.e. 6000 * 1024). Replies of the form 6000K or 6M are both deemed invalid.

A.21 Date Format

Global System Manager supports two date formats: European, dd/mm/yy; and American, mm/dd/yy. For example, the date 6th February 1956 is represented as 06/02/56 (European format) or 02/06/56 (American format). Whichever format you chose will be used automatically by any Global software to input and display dates.

A.22 Password Checking

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.23 Menu

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

The default menu contains the following entries:

```
Install Global Software . . . . 1
GSM Utilities . . . . . 2
```

The "GSM Utilities" sub-menu contains the following options:

```
Volume Maintenance. . . . . 1
Directory Maintenance . . . . . 2
System Status . . . . . 3
File Inspection . . . . . 4
Exit. . . . . .<CR>
```

A.24 The Spooler

The Spooler (see Chapter 8 of the Global System Manager Manual) allows a printer to be shared between several users. This is achieved by writing all reports to a special disk unit, the *spool unit,* rather than directly to the printer. The reports are printed off by a program called the spooler (command $SP) which typically is running all the time in a background partition. Note that it is still possible to print reports directly if the spooler is not printing.

As well as allowing you to share the printer, the spooler gives you more control over the printing of reports. You can 'hold' some reports for printing later, give others high priority, or print multiple copies. There is also a spooler status command, $SPS, that allows you to re-schedule printing in this way from screens other than the one running the spooler.

You must specify the unit address and size of the spool unit to be allocated. The default unit is the next free unit on the disk after SYSRES (e.g. if SYSRES is unit A01, the default spool unit will be A02). This is usually satisfactory, but on a network you may want to use a spool unit on another computer, in which case you should supply its address to the prompt. The size must be sufficient to hold all the reports waiting to be printed. As a guide, a typical report of 50 pages will occupy about 400 Kbytes. Clearly, if you intend holding a lot of reports without printing them you will need a larger unit than if you intend to print everything immediately.

Important note to users of the Global Cobol Development system: The size of the spool unit for Global System Manager V8.1 should be larger than for earlier versions of Global System Manager to accommodate the larger listing files produced by the V8.1 Global Cobol compiler.

The following prompt allows you to specify the size of the spool unit:

```
[A.24] Size of spool unit ( nnnnK):
```

The reply may be suffixed by K (to specify a size in Kilobytes), M (to specify a size in Megabytes) or G (to specify a size in Gigabytes).

If you asked for a menu to be entered after sign-on then two extra entries will be added to the "GSM Utilities" sub-menu (see section A.23) for the spooler. These entries will be:

```
Start spooler . . . . . 5
Check spooler status. . . . . 6
```

A.25 Selecting a Printer Unit

If you intend attaching more than one printer, you should specify the unit to be used for plain paper printing, as this unit will be used (unless overridden) for any report not on special stationery. If you have installed the spooler, then it will print reports to this printer.

A.26 Printer Busy Handling

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.27 Printer Baud Rate

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.28 Domain Error Map

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.29 Standard Printer Control Files

This prompt will not appear during the installation of Global System Manager V8.1. No Printer Control Files (see section 6.1.3.3 of the Global System Manager Manual) are installed.

A.30 Menu Entry For Menu Maintenance

You are asked whether you want an entry in the main menu for running the menu maintenance software. The software itself is installed onto the SYSRES unit.

If this option is selected, the following extra entry will be added to the default menu (see section A.23):

```
Menu maintenance. . . . . 3
```

A.31 Event Logging Installation

Event logging is fully described in the Global Utilities Manual. If you answer Y you will be prompted for the size of the master event log file, given a default of 200 Kbytes. The installation will allocate a unit on the computer called SYSLOG which will contain the event logging programs (about 160 Kbytes), a master event log file $$MLOG of the size given and leave free space of 1/5 of the size of $$MLOG for the event logging file $$LOG. An insufficient space message will be given if a unit cannot be allocated of the required size.

The following prompt allows you to specify the size of the SYSLOG volume:

```
[A.31] Specify size of master event log file in Kbytes (200):
```

Note that the reply is automatically multiplied by 1024 to convert to Kbytes so that, for example, a reply of 200 would be converted to 204800 bytes (i.e. 200 * 1024). Replies of the form 200K are deemed invalid.

To de-activate Event Logging at any time after Global System Manager has been installed, simply remove the $LG logical unit assignment using the $CUS "Permanent Unit Assignment" Customisation.

A.32 Century Start Year

This prompt gives you the option to change the year before which dates are regarded as being in the 21st century rather than the 20th (see section 6.1.2.12 of the Global System Manager Manual for further details).

A.33 Work Space Allocation

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.34 Install Operator Groups

The operator group option is fully described in the Global Utilities Manual. If you answer Y, the installation will create an empty group file ($$GROUP) of size 200Kb on the SYSRES sub-volume (i.e. on logical unit $M).

A.35 Is the Mailing System Installed on the Master Computer

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.36 Disk Requires Partitioning

This message is documented for completeness only. It should not appear during the installation of Global System Manager (Unix).

A.37 Install Mailing System

You are asked whether you want the Global System Manager mailing software installed. The Global System Manager mailing system ($MAIL) is fully described in the Global Utilities Manual.

If you answer Y you will be prompted for the size of the Mail unit, given a default of 1 Mbyte. The installation will allocate a unit on the computer called SYSML which will contain an empty $$MAIL file of maximum size (i.e. default 1Mb). An insufficient space message will be given if a unit cannot be allocated of the required size.

The following prompt allows you to specify the size of the mail unit:

```
[A.37] Size of mail unit ( nnnnK):
```

The reply may be suffixed by K (to specify a size in Kilobytes), M (to specify a size in Megabytes) or G (to specify a size in Gigabytes).

A.38 Install Speedbase Presentation Manager

Speedbase Presentation Manager V8.1 MUST be installed during the installation of Global System Manager**. Unlike pre-V8.1 versions of Speedbase Presentation Manager, it is NOT possible to use $INSOFT or run EPINS from the EPA distribution diskette.** If your software is for a Global System Manager PM configuration, key Y to install Speedbase Presentation Manager.

If Speedbase Presentation Manager is installed, the default menu (see section A.23) will contain the following options:

```
Install Global Software . . . . 1
GSM Utilities . . . . . . . . . 2
Presentation Manager. . . . . . 3
```

or, if a menu entry for Menu Maintenance has been requested (see section A.30), the default menu will contain the following options:

```
Install Global Software . . . . 1
GSM Utilities . . . . . . . . . 2
Menu maintenance. . . . . . . . 3
Presentation Manager. . . . . . 4
```

A.39 Install Speedbase Demonstration System

The installation of the Speedbase Presentation Manager demonstration software is optional. If you select to install the demonstration software you will be prompted for a sub-volume on which to install the demonstration data. **Important note**: The sub-volume selected must already be allocated on the hard-disk. The installation of the Speedbase Demonstration System will NOT allocate a new sub-volume automatically.

A.40 Save Current Global System Manager Customisation

If you are upgrading to Global System Manager V8.1 from Global System Manager V7.0 or V8.0 you have the option of saving the current Global System Manager customisation before installing. If this option is selected a BACSAV sub-volume will be allocated of the same size as the existing SYSRES. All the files on the existing SYSRES will be copied to BACSAV before Global System Manager is installed, overwriting the existing SYSRES.

A.41 Insufficient Space for a Save Unit

There is insufficient space to allocate a BACSAV sub-volume to save the contents of the existing SYSRES (see A.40, above). If you choose to continue with the installation the current Global System Manager customisation will NOT be saved.

A.42 Existing Customisation will be restored

This option only appears if you selected to save the existing Global System Manager customisation prior to installing Global System Manager V8.1 (see section A.40).

If you choose to restore the customisation, the files listed in section 2.1.10 will be copied from the BACSAV sub-volume to the new SYSRES subvolume.

Furthermore, if the save/restore existing customisations option is selected then it is NOT possible to apply the new V8.1 customisation options (e.g. $GROUP - see A.34; $MAIL - see A.37) during the installation of Global System Manager. These customisations must be applied using the "Install Extra Facilities" option of $CUS as explained in section 6.1.4.13 of the Global System Manager Manual.

Appendix B - Installation Error and Warning Messages

The messages listed in this appendix appear when you attempt to startup, sign on to or use Global System Manager which has been invalidly customized, has had its configuration file wrongly updated by Global Configurator, or has been incorrectly generated. Those messages designated as errors are fatal and prevent Global System Manager from being loaded. Warnings are less severe and normally allow a degraded system with missing features to operate so that you can use Global System Manager itself to analyse the problem. A warning message should never be ignored unless the documentation indicates it is to be expected.

All messages of the form:

program_name: number and error message

are listed separately in Appendix C.

The following notes are referred to from the message descriptions overleaf:

Note 1 This condition is documented for completeness only to assist the TIS Software staff responsible for software generation and porting Global System Manager to new operating systems. If it does occur it indicates that the system has been incorrectly or incompletely generated, or there is an internal error in the installation jobs.

Note 2 When there is insufficient space to load a program this may be due to you specifying too large a system stack or too small a memory bank. You may need to re-install Global System Manager.

Note 3 If an error occurs on a distribution volume (i.e. BACRES, BEA or HAA) you will normally have to obtain a new set of diskettes before you can proceed. If the error is on a system volume you should either restore the volume from a backup, or re-install Global System Manager.

$57 INITIATION WARNING 1 - FILE *name* NOT FOUND
$57 INITIATION WARNING 2 - FILE *name* HARD ERROR H
$57 INITIATION WARNING 3 - FILE *name* CORRUPT
$57 INITIATION WARNING 4 - FILE *name* READ ERROR

These messages appear if a problem occurs during the startup process when attempting to load the named file. Note 1 applies to warning 1. Warning 2 usually indicates a drive electronics error rather than a media error, and if it persists after retrying the startup process several times the drive should be serviced. Note 3 applies to warnings 3 and 4. If a target system volume is affected it will be the SYSRES volume.

$57 INITIATION WARNING 5 - FILE *name* MISSING DEVICE

The physical device or simulated volume associated with a particular nucleus file is not present on this computer. This usually indicates that the software has been supplied with a configuration file which describes a configuration which is a superset of the current one. For example, the configuration file might support both 3½" and 5¼" diskette drives whereas the computer you are working with possesses just one diskette drive.

The message appears only when the starter system is loaded and is suppressed during the loading of the target system to avoid an unnecessary warning when the reason for it is understood. The file name indicates the type of missing device:

+C2CA*xx*	direct access device
+C2CB*xx*	screen
+C2CE*xx*	printer

$57 INITIATION WARNING 6 - FILE *name* ERROR *x*

Note 1 applies to this message. An unexpected problem, not covered by the previous 5 warnings, has occurred when accessing the indicated file. The error code is a single letter whose meaning is given under "$99 ERROR *x* ..." see below.

$57 INITIATION ERROR 10 - UNABLE TO LOAD P.$MON *uuu e*

Note 1 applies to this message. It was not possible to open the monitor overlay library. This may be because of illegal use of the $F INS or PIP instructions. The attempt to open P.$MON from unit *uuu* failed with an error of type *e*.

$57 INITIATION ERROR 11 - UNABLE TO LOAD P.$MON INDEX *e*

Note 1 applies to this message. It was not possible to read the monitor overlay library index. This probably means that the library is corrupt. The attempt to read the library index failed with an error of type *e.*

$57 INITIATION ERROR 12 - UNABLE TO READ MONITOR OVERLAY *e*

Note 1 applies to this message. It was not possible to read the monitor overlay from the P.$MON library. This probably means that the library is corrupt or incomplete. The attempt to read the library failed with an error of type *e*.

$57 INITIATION ERROR 13 - UNABLE TO CREATE MONITOR PAGE

Note 1 applies to this message. It was not possible to load an overlay from P.$MON into a monitor page. This error will occur if the number of monitor pages has been reduced by injudicious use of the $F PAM instruction.

$57 INITIATION ERROR 14 - UNABLE TO LOAD P.$PAGES
e

Note 1 applies to this message. It not possible to read the monitor overlay library P.$PAGES. This maybe because of illegal use of the $F INS or PIP instructions. The attempt to open P.$PAGES failed with an error type *e*. It is still possible to initiate Global System Manager but attempts to run some command programs will result in PGM CHK-8.

$57 INITIATION ERROR 55 - STOP -98 ON ODD BOUNDARY

Note 1 applies to this message. An error has occurred when loading the monitor file ($MONITOR).

$57 INITIATION ERROR 60 - MONITOR AND COMMAND LIBRARY INCOMPATIBLE

This error indicates that the version code of the monitor is not the same as that associated with Global System Manager command library. See Note 1.

$57 INITIATION ERROR 61 - INVALID GENERATION

The serial code keyed in response to the "please key serial code" prompt, though a valid code, did not match the software on the disk. Try again in case you mis-keyed the number. If the error occurs again, it probably indicates an error in the generation of Global System Manager, and you will need to obtain replacement disks.

$57 INITIATION ERROR 101 - $STARC I/O ERROR
$57 INITIATION ERROR 102 - $STARC NOT FOUND

The startup process has been unable to load the main sign-on procedure overlay, $STARC, from the command library on BACRES (starter system) or SYSRES (target system). If the error is on SYSRES either restore it from a backup or re-install Global System Manager. See Note 3.

$57 INITIATION ERROR 103 - $STARC TOO LARGE

This error terminates the startup process because there is insufficient space to load the main sign-on procedure overlay. See Note 2.

$57 INITIATION ERROR 151 - INITIATION INCOMPLETE

This error, which terminates the startup process, will occur if you key <CTRL W> to interrupt it before the process is complete. The message will not appear immediately you key <CTRL W>, but will occur at the point at which the ready prompt would normally be output.

$57 INITIATION WARNING 201 - TOO MUCH MONITOR IS SWAPPED
SWAP UNIT WOULD HAVE BEEN *nnn*

Note 1 applies to this message. The initiation of memory by the system has not been properly completed.

$57 INITIATION WARNING 203 -
NOT ENOUGH ROOM FOR ALL USERS ON SWAP FILE OF SIZE *ssssss*
SYSTEM OF *pp* PARTITIONS POSSIBLE
IS THIS OK?:

There is insufficient room on the unit assigned to $SW (normally SYSRES) for a swap file large enough to hold information for all the users of this system. This may occur if you change the configuration using $CUS (Configuration Maintenance) or Global Configurator.

If you reply Y or <CR> to the outstanding prompt Global System Manager will continue by initiating a restricted system as indicated by the message. A response of N also

causes the startup process to continue, but in this case only a single-user system will be initiated.

If the system as loaded is satisfactory but you wish to prevent this prompt appearing in the future, you should run $CUS and decrease the number of users as described in Chapter 6 of the Global System Manager Manual.

To support the full number of users you must use $REORG to increase the size of SYSRES by at least (*memory bank size * number of extra users needed*) bytes. An alternative is to establish a separate subvolume for swap files (by convention called $$SWAP) of the required size, and then use the $CUS Permanent Unit assignments option to reassign $SW to the address of this unit.

$57 INITIATION WARNING 205 - IRRECOVERABLE ERROR ON $$SWAP FILE - *x*

An irrecoverable error has prevented the swap file from being used. The single character at the end of the message indicates the type of error: R or W indicate an I/O error on the disk; S indicates that the disk is full. The startup process continues, but only a single user system will be initiated. See Note 3.

$57 INITIATION WARNING 206 - SWAP FILE ERROR 2
$57 INITIATION WARNING 207 - SWAP FILE ERROR 3
$57 INITIATION WARNING 208 - COMMAND $STARN OR $STARO CANNOT BE LOADED

These warnings indicate that internal consistency checks within the startup process have failed. Note 1 applies. The startup process continues but only a single-user system will result.

$57 INITIATION WARNING 209 - NO TIMER AVAILABLE
SWAP UNIT WOULD HAVE BEEN *nnn*

This message should not occur and is documented for completeness only.

$57 INITIATION WARNING 210 - TOTAL NUMBER OF PARTITIONS EXCEEDS 99

The configuration file has become corrupt.

$57 INITIATION WARNING 251 - *program* CANNOT BE LOADED

This warning appears if you have used customization to include the indicated program in the system stack, but this is not possible. If an I/O error prevented the program from being loaded the warning will be preceded by an explanatory message. Otherwise the problem is either due to the program (or command) not being present on the device assigned to $P (or $CP), or to there being insufficient room in the system stack for it. Global System Manager will continue as though the customization had not taken place.

$57 INITIATION WARNING 302 - $MONITOR VERSION *x*; $STARC VERSION *y*

This warning message indicates that the version code of the monitor is not the same as that associated with Global System Manager command library. See Note 1. The startup process continues unaffected.

$57 INITIATION ERROR 303 - CANNOT BOOTSTRAP FROM V.R.K

This message should not occur and is documented for completeness only.

$57 INITIATION ERROR 35x - USER FILE ERROR

This indicates that an error has occurred on the user file when initiating a multi-user system. If it is an I/O error affecting the file itself the warning will be preceded by an error message. Otherwise SYSRES is probably corrupt and should be restored. See Note 3.

$57 INITIATION ERROR 36x USER FILE ERROR

This indicates that an error has occurred on the user file when initiating a multi-user system. If it is an I/O error affecting the file itself the warning will be preceded by an error message. Otherwise SYSRES is probably corrupt and should be restored. See Note 3.

$57 INITIATION ERROR 360 - NO ROOM FOR SYSTEM IN USER FILE

The user file has become fragmented so that there is not a contiguous range of user numbers available, and in consequence it is not possible to add a new entry for this SYSTEM. The user file has been marked as requiring reorganization, but this cannot be done until Global System Manager is restarted, when it will happen automatically.

$57 INITIATION ERROR 363 - INITIALISING MORE THAN MAXIMUM NUMBER OF USERS

See error 369.

$57 INITIATION WARNING 364 - CANNOT INITIALISE ALL USERS

There is insufficient space in the system to initialise all the users so a reduced number of users will have been initialised. This may happen because the user file is too small. The user file can be extended by using the $STATUS command to purge the user file, allocating a larger size, and then restarting the system again.

$57 INITIATION ERROR 368 - INCOMPATIBLE CONFIGURATION

The software level defined by the configuration file is incompatible with the software level of Global System Manager. This is probably caused by changing your configuration file to a higher level of Global System Manager.

$57 INITIATION ERROR 369 - MORE THAN PERMITTED NUMBER OF USERS

Global System Manager has been generated for a maximum number of users. This error will occur if more than the maximum number of users have been configured. This may occur if more screens have been added to the configuration file, or if more SYSTEM's have been added to the Systems file. If you require more screens you must obtain an upgraded Global System Manager.

$57 INITIATION WARNING 370 - LOG FILE IS NEARLY FULL

The log file, $$LOG on unit $LG, is nearly full (i.e. there is space for less than 50 records in the logfile). When this message appears, it is prudent to use the $LOG command to purge the log-file.

To de-activate Event Logging at any time, simply remove the $LG logical unit assignment using the $CUS "Permanent Unit Assignment" Customisation.

$57 INITIATION WARNING 371 - LOG FILE IS FULL

The log file, $$LOG on unit $LG, is full. No more events will be logged until the $LOG command is used to purge the log-file.

To de-activate Event Logging at any time, simply remove the $LG logical unit assignment using the $CUS "Permanent Unit Assignment" Customisation.

$57 INITIATION WARNING 372 - UNABLE TO LOCK USER FILE
DO YOU WISH TO TRY AGAIN (Y)
$57 INITIATION WARNING 373 - UNABLE TO UNLOCK USER FILE
DO YOU WISH TO CONTINUE (Y)

The system is unable to lock or unlock the user file. This can occur if another SYSTEM is in the process of initiating Global System Manager or a utility (e.g. $STATUS or $BYE) is being run which temporarily requires exclusive use of the user file. If you key "Y" in reply to the prompt the lock or unlock operation will be retried. A reply of "N" will cause initiation to be abandoned and the system will need to be restarted. If either of these warnings occur check that no other operators are using $STATUS or $BYE.

$57 INITIATION ERROR 374 - SYSTEM START PROCESS ABANDONED

This message appears if "N" has been keyed to warnings 372 and 373.

$57 INITIATION WARNING 375 - UNIT ALIASING FAILURE

An error has occurred during the calculation of the aliases for units 100 - 108. This problem may occur if the 100 unit stored in $MONITOR has not been patched correctly using the $F PAM instruction.

$57 INITIATION WARNING 401 - CUSTOMISATION HAS INCREASED COUNT OF SCREENS
$57 INITIATION WARNING 402 - CUSTOMISATION HAS INCREASED COUNT OF USERS

The user details customization instruction has either increased the number of normal users beyond the maximum number of screens the configuration supports, in which case warning 401 appears, or has increased the total number of users beyond the maximum defined in the configuration file. This error usually results from changing the configuration file to contain fewer screens or users. A single user system will be initiated so that you can change the erroneous user details customization.

$57 INITIATION WARNING 403 - S-USER CUSTOMISATION OF M-USER NUCLEUS

This warning appears if user details customization has reduced the number of users of a multi-user system to just one. In this case the startup process will continue but only a single-user system will be initiated.

$57 INITIATION WARNING 404 - M-USER CUSTOMISATION OF S-USER NUCLEUS

This warning appears, together with either warning 401 or 402, if you employ user details customization to set the number of users of a single-user system to more than 1. See the notes accompanying warnings 401 and 402.

$57 INITIATION WARNING 405 - BANK CUSTOMISATION INVALID IN S-USER BOS

This message should not occur and is documented for completeness only.

$57 INITIATION WARNING 407 - TINT CUSTOMISATION INVALID IN S-USER BOS

This message should not occur and is documented for completeness only.

$57 INITIATION WARNING 409 - WIDTH OF PRINTER 5*xx* NOW *nnn*

This warning appears if you have attempted to increase the line width of the indicated printer past the maximum (*nnn*) established when your system was generated. Your request will be ignored and the width will be set to the maximum allowable, as indicated by the message.

$57 INITIATION WARNING 410 - CUSTOMISATION OF NON-EXISTENT PRINTER 5*xx*

You have mistakenly used customization to modify the printer attributes of the unit whose address appears in the message, but that unit is not supported as a printer on your configuration. The spurious customization is ignored.

$57 INITIATION WARNING 411 - CUSTOMISATION INCREASE OF CBOS SCREENS
$57 INITIATION WARNING 412 - CUSTOMISATION INCREASE OF CBOS PARTITIONS

These messages should not occur and are documented for completeness only.

These are equivalent to messages 401 and 402, but for a single-screen configuration.

$57 INITIATION ERROR 413 - CANNOT BOOT WITH THIS VERSION OF USER FILE

The user file present on SYSRES is incompatible with the present system.

$57 INITIATION WARNING 414 - INAPPROPRIATE VERSION OF USER FILE

The user file present on the master is inappropriate but not incompatible. It is possible for initiation to proceed.

$57 INITIATION WARNING 450 - INVALID TERMINAL TYPE

The TAP supplied from the host operating system does not exist or has not been installed.

$57 INITIATION WARNING 460 - NO GUI LICENCE
(PLEASE CONTACT YOUR SOFTWARE SUPPLIER)

You have attempted to use the Global Windows Workstation on a Global System Manager configuration that has not been licenced to use this software. Please contact your software supplier to obtain a Global Windows Workstation licence.

$57 INITIATION WARNING 461 - GUI LICENCE EXCEEDED
(PLEASE CONTACT YOUR SOFTWARE SUPPLIER)

You have attempted to exceed your Global Windows Workstation licence. Please contact your software supplier to upgrade your Global Windows Workstation licence.

$57 INITIATION ERROR 462 - INCOMPATIBLE NUCLEUS

You have attempted to use the Global Windows Workstation with a pre-V8.1 nucleus. This is not allowed.

$57 INITIATION WARNING 5*xx* - USER FILE ERROR DURING SIGN-ON

This indicates that an error has occurred on the user file when initiating a multi-user system. The *xx* number in the warning code indicate the File Executive operation that suffered the error. If it is an I/O error affecting the file itself the warning will be preceded by an error message.

$57 INITIATION WARNING 501 - USER FILE ERROR DURING SIGN-ON

This indicates that an error has occurred on the user file when initiating a multi-user system. The error has occurred in the OPEN operation. If it is an I/O error affecting the file itself the warning will be preceded by an error message.

$57 INITIATION WARNING 510 - USER FILE ERROR DURING SIGN-ON

This indicates that an error has occurred on the user file when initiating a multi-user system. The error has occurred in the CLOSE operation. If it is an I/O error affecting the file itself the warning will be preceded by an error message.

$57 INITIATION ERROR 55*x* - MEMORY BANK INITIATION, DISK ERROR

There has been an I/O error on the swap file during the initiation of the memory banks. Error codes 551, 552 and 553 indicate an error opening the swap file. Error codes 554, 555 and 556 indicate an error on the first read of the swap file. Error codes 557, 558 and 559 indicate an error on subsequent reads of the swap file.

$57 KEY NEW SYSRES UNIT:

It is not possible to access the SYSRES unit. You should supply the address of SYSRES on another file-server SYSTEM (if there is one configured) which your

SYSTEM can access to initiate Global System Manager from. If there isn't one then Global System Manager will have to be re-installed.

$57 MASTER COMPUTER UNAVAILABLE
$57 KEY UNIT OF SYSRES ON NEW MASTER COMPUTER:

This message should not occur and is documented for completeness only. If this message does appear, it implies that the SYSTEM which contains the master SYSRES has failed to configure.

$57 SYSTEM IS QUIESCED - PLEASE DO NOT ATTEMPT TO SIGN ON

A supervisor has used the $STATUS QUI command to quiesce Global System Manager. New users are not allowed to sign on to Global System Manager.

$57 SWAP FILE NOT FOUND

You have attempted to sign off using $E but have failed to load the volume containing the swap file, although requested to do so by a previous mount prompt. In this circumstance control returns to the monitor and you remain signed on to the system.

$57 $STAR*xx* MUST NOT BE RUN

You have attempted to run from the menu or ready prompt one of the command program overlays used by the sign-on procedure. This is not allowed. The overlay returns control immediately to the ready prompt.

$66 INSTALLATION ERROR 1

There is insufficient free memory for the installation program to install the startup initialisation data. The installation or transfer operation can only take place on a system with a larger user area.

$66 INSTALLATION ERROR 2

The startup unit is invalid. See Note 1.

$66 INSTALLATION ERROR 3
$66 INSTALLATION ERROR 4

The part 1 startup data file (error 3) or the part 2 startup data file (error 4) cannot be found on the input volume. The missing file is neither present as a stand-alone file, nor is it a member of startup data library #.*nnnn*, where *nnnn* is the set number (the numeric part of the configuration file name). Note 1 applies.

$66 INSTALLATION ERROR 5

The BH record on the part 1 startup data file is inconsistent with information held on the configuration file. See Note 1.

$66 INSTALLATION ERROR 6

The INS instruction is unable to allocate a +BOOT file because the output volume already contains other data. It should be empty. See Note 1.

$66 INSTALLATION ERROR 7

The volume type of the output volume is not the same as the volume type required for the startup data, as held in the configuration file. See Note 1.

$66 INSTALLATION ERROR 8

The assignment tables are full and it is impossible for the INS instruction to assign $P to the input unit in order to access the startup data library. See Note 1.

$66 INSTALLATION ERROR 9

The output diskette is software interleaved, but the startup process cannot use interleaving. The volume should have been initialised (or formatted) with access option 1. See Note 1.

$78 DESTROY *xxxxxx*?
$78 DESTROY *xxxxxx* ON UNIT *xxx*?

The SYSIPL diskette you have mounted has previously been initialized by Global System Manager with the name shown. The second form of the message indicates that it is also formatted differently. If you intend overwriting an old diskette reply Y to continue, otherwise key N and then mount the correct diskette.

$94 *message*

All messages prefixed by "$94" are documented in Appendix A of the Global System Manager Manual.

$99 ERROR *c* LOADING *component*

There has been a fatal error during the startup process which prevented the loading of the identified component. The startup process cannot continue, and it is impossible to initiate Global System Manager. The quantity *c* is a single character error code defining the problem:

3 The required component is missing. Normally Note 1 applies.

H There was a hardware error (associated with the drive electronics). Retry the startup process several times and if the error persists have the drive serviced.

M The resident part of Global System Manager is too large to be loaded into the available memory. Normally Note 1 or Note 2 applies, but if you have updated the configuration file using Global Configurator this means you will have to reduce its size by allocating fewer or smaller buffers, or by removing support for some devices.

P The configuration file version is incompatible with the nucleus version. Note 1 applies.

R There was an I/O error (associated with the magnetic medium) when reading the component. See Note 3.

V The Global System Manager directory of the volume containing the nucleus files and monitors has become corrupt. See Note 3.

Z Sequence number fields in the records of the component itself are in error, indicating that it has become corrupt. See Note 3.

Appendix C - Error Messages From Unix Utilities

The messages described in this appendix are generated during the Global System Manager startup process, or by one of the Global System Manager utilities in $GLDIR/bin. All the error messages described in this appendix are generated from two files, $GLDIR/sys/errmsg/hdr and $GLDIR/sys/errmsg/txt. These text files list errors by error number.

For each error message a description of the possible error is given and where possible a solution. The errors are listed by program name.

C.1 Messages from "global"

global: Error 1001 - Invalid option *option*
usage: global [-i] [-ttypeahead] [-wdelay]
or:
global: Error 1003 - Invalid parameter *parameter*
usage: global [-i] [-ttypeahead] [-wdelay]

When running global you have specified an illegal flag on the command line. The valid flags are described in section 6.3.3. For the -t option, if the typeahead contains spaces or illegal characters then you must quote it (preferably in single quotes).

global: Error 1005 - GLDIR not defined.

You do not have the shell variable GLDIR defined **and exported**. This variable must be set to the path of the global directory.

Note that this message will normally appear as:

Global System Manager: Error 1005
(can't open error header file "sys/errmsg/hdr")

global: Error 1006 - Can't create lock file.

The lock file required by this SYSTEM to startup cannot be created or accessed. The lock file is called $GLDIR/tmp/gl_LCK_001. Ensure that the directory exists and that if the file already exists that it has read/write permissions for group *global*.

global: Error 1007 - Can't lock lock file.

This error should never occur. Execute the command again.

global: Error 1008 - User not authorised to install Global System Manager.

To run global with the -i (install) option you must be either the superuser (root) or the user *global*.

global: Error 1009 - Authorisation failed. Inform your System Administrator.

The Systems file does not contain authorisation for your user name, nor does it contain authorisation for this terminal.

global: Error 1010 - Can't allocate shared memory.

The shared memory required by Global System Manager could not be created due to some unknown system restriction. For example the system limits for shared memory size have been exceeded (see section 7.4.4). Alternatively your Unix system may not have the System V IPC facilities installed.

global: Error 1011 - Can't create message queue.

The message queue required by Global System Manager could not be created due to some unknown system restriction. The message queue may have remained in the system after Global System Manager resources were removed.

global: Error 1012 - Can't load shared memory image.

Could not find, open or read the Global System Manager data file $GLDIR/sys/data/shm. Check that this file exists and that it has the read permissions for group *global*.

global: Error 1013 - Can't reinstall whilst Global System Manager is in use.

Global System Manager and the Global applications are currently being used by (possibly) other users on the system. Before and during installation you must ensure that other users are prevented from using Global System Manager. If Global System Manager had previously crashed it is possible that some of the resources required were left in the system. These can be removed by running glclean (see section 6.7).

global: Error 1014 - Can't start timer daemon. Inform your System Administrator.

Either the program gltimd can not be run from $GLDIR/bin or the new process can not be created due to insufficient system resources or the system imposed limit of processes has been reached for that user.

Configuring Global System Manager (BACNAT V*v.vvv*); please wait...

This message is always displayed while Global System Manager is loading. **The BACNAT variant number, *v.vvv*, should always be quoted when reporting problems to the TIS Software Service Centre.**

Configuring Global System Manager (B*nnnn* V*v.vvv*); please wait...

This message is always displayed while Global System Manager is loading (where *nnnn* is the configuration code). **The BACNAT variant number, *v.vvv*, should always be quoted when reporting problems to the TIS Software Service Centre.**

global: Error 1016 - No install device defined in Systems file.

There is no INSDEV command in the Systems file. Without this information Global System Manager does not know which device to install from.

global: Error 1017 - Fileserver A has failed to configure; system terminating.

Whilst attempting to install Global System Manager, SYSTEM A can not startup for one of many reasons. For example:

- The install device does not contain the correct installation software;
- Insufficient system resources to fully configure this process;
- The configuration file on the distribution media is not compatible with the current Systems file.

global: Error 1018 - Fileserver *letter* has failed to configure; system terminating.

Whilst configuring Global System Manager resources the SYSTEM *letter* file server could not initialise the data required by the Global applications to access these files:

- SYSDATA for this SYSTEM does not contain a configuration file compatible with the current Systems file;
- Insufficient system resources to fully configure this process;
- The configuration file on the distribution media is not compatible with the current Systems file.

global: Error 1020 - System *systemid* has failed to configure.
Inform your System Administrator.

A fatal error occurred during the initialisation of this user's SYSTEM *systemid*:

- SYSDATA for this SYSTEM does not contain a configuration file compatible with the current Systems file;
- Insufficient system resources to fully configure this process.

global: Error 1021 - Can't access *$GLDIR*/sys/data/user to validate user.

Could not find, open or read the specified file. Ensure that this file exists (created by glconfig) and has read permissions for the group *global*.

global: Error 1022 - Can't access *$GLDIR*/sys/data/tty to validate tty.

Could not find, open or read the specified file. Ensure that this file exists (created by glconfig) and has read permissions for the group *global*.

global: Error 1023 - Can't read *filename*.

Could not find, open or read the specified file. Ensure that this file exists and has read permissions for the group *global*.

global: Error 1024 - Died due to signal SIG*name*.

A signal was caught which could not be ignored. The signal SIGTERM implies that the process was killed deliberately, either by glclean or by the use of the kill command.

global: Error 1025 - Error returned from new shell.

One of the following has occurred:

- A new shell (or the command specified by $SHELL) could not be created;
- The new shell exited with a non zero exit status. This will occur if the previously executed shell command returned an error to the shell. The message is for information only.

global: Error 1026 - Error in option; -w*number*.
The delay must be a non-negative integer.

The wait period for a Global System Manager SYSTEM to initialise, given by *number*, is not a valid positive number. The default is 20 seconds.

global: Error 1027 - System *letter* has failed to configure.
Inform your System Administrator.

See error 1020.

global: Error 1028 - Screen not contained in configuration file.

The USER *number* defined for this user does not have a corresponding set of USER DISPLAY ATTRIBUTES defined in the configuration file. The *number* specified must correspond to User *number* in the configuration file (see section 7.3.1).

global: Error 1029 - User already running Global System Manager.

This user is already running Global System Manager and is thus prevented from running it again. Ensure that the user has not created a new shell and is trying to run global. Alternatively the user has run Global System Manager from another terminal and has not exited. As a last resort glclean can be run to restart this user - however any Global applications that were running for this user will terminate possibly leaving the files in an inconsistent state.

global: Error 1030 - System imposed limit for shared memory size exceeded.

The system kernel limit on the size of shared memory available to a process is smaller than that used by Global System Manager. The default used is 128K bytes. This can be overridden by the Systems file command SHMSIZE (see section 7.4.4). Values smaller than 128K are not recommended. It may be necessary to change the system kernel limit before continuing.

global: Error 1031 - Could not create semaphore.

The semaphore required by Global System Manager could not be created due to some unknown system restriction.

global: Error 1032 - Could not create lock semaphores.

One lock semaphore is required for each file server. Either:

- The semaphores required by Global System Manager could not be created due to some unknown system restriction. Ensure that the system imposed limit for the number of semaphores on your system is sufficient. Global System Manager requires *n* + 1 semaphores where *n* is the number or file server SYSTEMs;

- The system Kernel limit on the number of semaphores available for either this process or the system would be exceeded. It may be necessary to change the system Kernel limit before continuing.

global: Error 1033 - System no longer running.

Your SYSTEM daemon has terminated for some reason whilst you were executing a command in a new shell. This message appears on attempting to return to the nonexistent process.

global: Error 1035 - Global System Manager expiry date reached.

You are attempting to use pre-release or demonstration software after the agreed expiry date. Contact TIS Software to obtain an updated version of the demonstration software.

global: Error 1036 - Timed out on process synchronisation.
Inform your System Administrator.

Your glintd daemon has terminated for some reason while you were attempting to initiate Global System Manager. This message will normally be preceded by an error from the glintd process (see section C.2).

global: Warning 1037 - Can't open system log file.

It was not possible to open the messages file (see section 6.1.1.3) probably because the permissions on the file $GLDIR/sys/messages have been changed. No permanent record will be kept of the current Global System Manager session.

global: Message 1038 - Input buffer full; waiting...

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1039 - Input buffer now available.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1040 - Unexpected error from keyboard read; errno = *number*.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1041 - TTY output interrupted by a signal. Retrying.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1042 - Disconnected from GSM.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1043 - Unable to restore TTY. Controlling terminal lost.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1044 - Entering GSM session.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1045 - Message queue error. errno = *number*

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1046 - Child (pid *number*) died.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1047 - Died due signal *string*.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Message 1048 - Entering subshell.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

global: Error 1049 - Invalid tty device name (Unknown or too long).

The name of the tty device name is invalid. Inform your System Administrator.

global: Error 1050 - glintd can't open tty device.

An attempted open on your tty device (*tty_device_name*) has failed unexpectedly. Check the permissions on the file /dev/*tty_device_name*.

global: Error 1051 - GLNAME variable too long.

The name of the GLNAME shell variable (see section 8.3.1) is too long.

global: Diagnostics 1052 - Made it to *diag_message*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

C.2 Messages from "glintd"

glintd: Error 1200 - Invalid parameters.
or:
glintd: Error 1201 - Invalid parameters.

An invalid parameter has been passed to glintd. This error should never occur as glintd is only executed by the global program.

glintd: Error 1202 - Invalid system id.

A SYSTEM *id* outside the range 1 to 255 has been passed to glintd. This error should never occur as glintd is only executed by the global program.

glintd: Error 1203 - GLDIR not defined.

The shell variable GLDIR does not exist. This error should never occur as glintd is only executed by the global program which checks for the existence of GLDIR.

Note that this message will normally appear as:

Global System Manager: Error 1203
(can't open error header file "sys/errmsg/hdr")

glintd: Error 1204 - IPC inconsistency.

The IPC facilities (shared memory, semaphores and message queue) that have been set up by the program global have been removed or are not accessible by this process. Ensure that the permissions and ownership of the programs in $GLDIR/bin are exactly as set up by glinstall during installation.

glintd: Error 1205 - This program can only be run by Global System Manager.

The program was executed other than by running the program global. This is not permitted.

glintd: Error 1206 - SHM mapped to different address.

This error should not occur. Please contact TIS Software immediately.

glintd: Error 1207 - SYSDATA device not found.

Can't open the specified directory to read its entries. Ensure that the group *global* has read, write and execute permissions for this directory.

glintd: Error 1208 - SYSDATA device not found.

Can't find the file which contains the initialisation data. This file is called SVL00_*name* and must exist in the SYSDATA directory. This file is created automatically by running glmkdat.

glintd: Error 1209 - SYSDATA device not found.

Can't open the SYSDATA file or SVL00_*name* file to read the initialisation data. Ensure that this file has read, write permissions for the group *global*.

glintd: Error 1210 - Can't allocate space for configure information.

A request for workspace failed either due to the system imposed memory limit per process, or the resource being temporarily unavailable. Run global again; if the problem still persists, ensure that this user's memory limit is sufficient (at least 1Mbyte).

glintd: Error 1211 - Can't read configure information from SYSDATA.

The initialisation data could not be retrieved from the SYSDATA *device*. Suspect a media error.

glintd: Error 1212 - No configure information on SYSDATA.

The SYSDATA *device* has no initialisation information installed. The device must have an installed SYSRES or BACRES.

glintd: Error 1213 - Can't create glspad.

The glspad daemon could not be created either because the system/user imposed limit for number of processes would have been exceeded, the system currently has insufficient resources or the program $GLDIR/bin/glspad could not be executed. For the latter check the permissions and ownership of all the files within $GLDIR/bin to ensure that they are the same as those set up after installation.

glintd: Error 1217 - Process terminated due to signal SIG*name*.

The program has been sent a signal that it cannot ignore.

The signal SIGTERM could be sent by glclean, or by the use of the kill command. It is important to check the integrity of any files that were being accessed by Global modules at the time.

glintd: Error 1218 - Insufficient shared memory for screen information.

The shared memory area defined for Global System Manager is too small (default 128K): Use SHMSIZE in the Systems file to increase the size (see section 7.4.4). It may be necessary to increase the system imposed maximum size before continuing.

glintd: Error 1219 - Malloc failed.

See error 1210.

glintd: Error 1220 - System data file missing.

Could not find, open or read the $GLDIR/sys/data/screen file. Ensure that this file exists (created by glconfig) and has read permissions for the group *global*.

glintd: Error 1221 - Unable to load screen information.

Can't read data from the $GLDIR/sys/data/screen file due to some unknown system error. See error 1220.

glintd: Error 1222 - Insufficient shared memory for screen information.

The shared memory area defined for Global System Manager is too small (default 128K): Use SHMSIZE in the Systems file to increase the size (see section 7.4.4). It may be necessary to increase the system imposed maximum size before continuing.

glintd: Error 1223 - Could not terminate processes.
Inform your System Administrator.

The execution of glclean failed. Ensure that $GLDIR/bin/glclean has execute permission for group *global*. When this error message occurs you may need to run glclean from the shell, to remove this SYSTEM.

glintd: Warning 1224 - Unable to create or open lock file.

Insufficient permissions to create or access the lock file $GLDIR/tmp/gl_LCK_001 (or the lock file has been deleted).

glintd: Warning 1225 - Can't open system log file.

It was not possible to open the messages file (see section 6.1.1.3) probably because the permissions on the file $GLDIR/sys/messages have been changed. No permanent record will be kept of the current Global System Manager session.

glintd: Message 1226 - PROCESS TERMINATED

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Warning 1227 - Using config variable GLCONFIGx, config file *++NNNNxx*

This warning message will only appear if the technique described in section G.41 has been used to define a SYSTEM-specific configuration file.

glintd: Warning 1228 - Domain contains multiple SVL00*_xxxxxx* files *directory_path/SVL00_xxxxxx*

As explained in section G.62, the SSD directory (e.g. A00.dir) MUST contain only one "header file" with the name SVL00_* (where * is the Unix shell wildcard character). Any SSD directory that contains multiple header-files is considered invalid by Global System Manager. If an "invalid directory" of this type is detected the domain will be ignored and the above warning message will be displayed.

The warning message will be followed by the full pathname of the unexpected file(s).

glintd: Warning 1229 - Unix file not multiple of Global System Manager track size *directory_path/SVLnn_xxxxxx*

As explained in section G.62, the size of every file in the SSD directory (e.g. A00.dir) MUST be an exact multiple of the Global System Manager virtual track size (i.e. 8Kb for format T224Z). The domain "header file" created by *glmkdat* is always 32Kb. When a "subvolume" file is created by $V, the size specified by the operator is always rounded up to be a multiple of 8Kb. Any files with an SSD directory that are not multiples of the track size are considered invalid. If an attempt is made to run global with an "invalid file" within an SSD directory defined in the Systems file, the file will be ignored and the above warning message will be displayed.

The warning message will be followed by the full pathname of the invalid file.

glintd: Warning 1230 - Directory contains multiple SVL*nn_xxxxxx* files *directory_path/SVLnn_xxxxxx*

As explained in section G.62, the SSD directory (e.g. A00.dir) MUST contain only one "subvolume file" for each numeric subvolume. For example, only one SVL01_* file, only one SVL02_* file etc. (where * is the Unix shell wildcard character). Any SSD directory that contains multiple subvolume-files is considered invalid by Global System Manager. If an "invalid directory" of this type is detected all the SVL*nn_xxxxxx* files will be ignored and the above warning message will be displayed.

The warning message will be followed by the full pathname of the unexpected file(s).

glintd: Diagnostics 1231 - $BYE: Made it to *diag_message*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Diagnostics 1232 - glprid has died, printer *print_device*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Diagnostics 1233 - Can't create FIFO, printer *print_device*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Diagnostics 1234 - Can't open FIFO, printer *print_device*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Diagnostics 1235 - Can't open direct printer *print_device*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Diagnostics 1236 - Start-up: System *xx*, pid *nnnnn* lanflag = *x*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

glintd: Diagnostics 1237 - Coredump mode enabled

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3) if the global "-c" option (see section 6.3.3.12) has been used. It will never appear on the screen.

C.3 Messages from "glmkdat"

glmkdat: Error 1400 - Data file already exists.

The file name you have specified already exists.

glmkdat: Error 1401 - Size must be in K or M.

You must specify the size of your integrated data file in kilobytes or megabytes. The size must be a whole number, postfixed with either K or M (these can be in lower case). For example:

```
Enter size (K/M) : 100m
```

will attempt to create a file one hundred megabytes in size.

glmkdat: Error 1402 - Size must not be smaller than 256K bytes.

It makes little sense to have an integrated data file smaller than this size.

glmkdat: Error 1403 - Cannot create data file.

The specified file could not be created. Either the file already exists, or the user *global* does not have write permissions for the directory into which the file is to be created. This includes read-only file systems.

glmkdat: Error 1404 - Cannot allocate requested size to data file.

When creating an integrated data file, the size of that file is guaranteed by writing to every byte of the file. This error indicates that this write failed before the requested size was met. Due to either ulimit restrictions for this user, or insufficient space within this mounted file system.

glmkdat: Error 1405 - *filename* already exists.

filename specified for the discrete data file you are attempting to create already exists.

glmkdat: Error 1406 - Can't create *filename*.

The user *global* does not have write permissions for the directory into which the file is to be created. This includes read-only file systems.

glmkdat: Error 1407 - Cannot create discrete data files.

The index file SVL00_DOMAIN within the specified *filename* directory could not be created. Ensure that umask is set to allow writes for this user (and group).

glmkdat: Error 1408 - Cannot allocate index header.

Could not get 32K bytes from the file system to allocate space for this file. Check the free space for this file system, and the file quota for the user *global*.

glmkdat: Error 1409 - GLDIR not defined.

You do not have the shell variable GLDIR defined and exported. This variable must be set to the path of the global directory.

Note that this message will normally appear as:

Global System Manager: Error 1409
(can't open error header file "sys/errmsg/hdr")

C.4 Messages from "glclean"

glclean: Error 1600 - Variable GLDIR not defined, can't tidy Global System Manager.

You do not have the shell variable GLDIR defined and exported. This variable is required to locate the Global System Manager resources.

Note that this message will normally appear as:

Global System Manager: Error 1600
(can't open error header file "sys/errmsg/hdr")

Terminating processes and tidying Global System Manager resources

This message is always displayed while glclean is executing.

glclean: Error 1602 - Invalid system number specified.

The value specified after the -s is not a number. This value must be a number, it can be specified in decimal, or in hexadecimal (by prefixing 0x), or in octal (by prefixing 0). For SYSTEMs A to Z you should specify 1 to 26 respectively.

glclean: Error 1603 - Invalid option.

You have specified an invalid flag parameter. Valid flags are:

-s specify single SYSTEM only

glclean: Error 1604 - No system specified.

No parameters have been specified. Valid parameters are:

-s specify single SYSTEM only
all all SYSTEMs

**glclean: Error 1605 - Unable to clean up - can't get to shared memory.
Inform your System Administrator.**

The shared memory required by Global System Manager has been removed. Without this the Global System Manager programs will fail to run. The Global System Manager processes, IPC facilities and temporary files (in $GLDIR/tmp) must be removed by hand. Alternatively the ownership and permissions for glclean have been changed. These must remain as they were set up by glinstall. Ensure that you are running glclean in the directory $GLDIR/bin and not some copy within a directory previously specified by your path variable (usually $PATH), and that the ownership/permissions are correct.

glclean: Error 1606 - Invalid Parameter.

An unknown parameter has been specified. See error 1604.

glclean: Warning 1607 - Unable to lock lock file.

This lock file is used to ensure Global System Manager integrity when users are entering and exiting Global System Manager. It is a warning that another user may be affected by your operation, if it results in the removal of all Global System Manager resources. The other user will be returned to the shell with an error message before the Global System Manager copyright message appears.

glclean: Warning 1608 - Unable to open lock file.

Insufficient permissions to create or access the lock file $GLDIR/tmp/gl_LCK_001.

glclean: Warning 1609 - Can't open system log file.

It was not possible to open the messages file (see section 6.1.1.3) probably because the permissions on the file $GLDIR/sys/messages have been changed. No permanent record will be kept of the current invocation of glclean.

glclean: Message 1610 - Terminating System(s) *system numbers*.

This message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3). It will never appear on the screen.

C.5 Messages from "glconfig"

glconfig: Error 1800 at line *linenumber* - SYSTEM 0x*number* doubly defined.

You are trying to define the same SYSTEM *number* more than once. All SYSTEM definitions must be unique. If you wish to add definitions to a SYSTEM that has already been defined, you should add the details directly below that SYSTEM *number* definition. Alternatively use the CONTINUE *number* command, and add the extra details for this SYSTEM *number* below this.

glconfig: Error 1801 at line *linenumber* - SYSTEM *letter* doubly defined.

See error 1800, but for SYSTEMs A to Z.

glconfig: Error 1802 at line *linenumber* - Can't CONTINUE; SYSTEM 0x*number* nonexistent.

A CONTINUE *number* statement has been encountered for which there has been no previous SYSTEM *number* definition.

glconfig: Error 1803 at line *linenumber* - Can't CONTINUE; SYSTEM *letter* nonexistent.

See error 1802, but for SYSTEMs A to Z.

glconfig: Error 1804 at line *linenumber* - *id* not a valid SYSTEM id.

SYSTEM *id* is not in the range A to Z.

glconfig: Error 1805 at line *linenumber* - 0xC0 not a valid SYSTEM id.

SYSTEM 0xC0 (decimal 192) is an illegal SYSTEM number. For internal reasons this number must not be used.

glconfig: Error 1806 at line *linenumber* - SYSDATA doubly defined for SYSTEM 0x*number*.

You are attempting to define SYSDATA for this SYSTEM more than once. There can only be one SYSDATA per SYSTEM, and in most cases the Systems file will be set up defining SYSDATA for SYSTEM A only.

glconfig: Error 1807 at line *linenumber* - SYSDATA doubly defined for SYSTEM *letter*.

See error 1806, but for SYSTEM *letter*.

glconfig: Error 1808 at line *linenumber* - INSDEV doubly defined.

INSDEV *filename* has been defined more than once. There can only be one installation device.

glconfig: Error 1809 at line *linenumber* - DATA *filename* allocated more than once.

You are attempting to allocate the data file *filename* in more than one DATA definition. Each data file must be unique, otherwise all locking will be lost. Do not attempt to link data files.

glconfig: Error 1810 at line *linenumber* - DATA *number* doubly defined.

For a given SYSTEM the DATA *number* must be unique.

glconfig: Error 1811 at line *linenumber* - ANA code *ANcode* for DISKETTE *number* doubly defined.

For this DISKETTE *number* on this SYSTEM the *ANcode* code (i.e. the Global diskette geometry code) has been specified more than once. If you have a second physical device with the same *ANcode* code, add this as a new DISKETTE *number*.

glconfig: Error 1812 at line *linenumber* - Invalid ANA code *ANcode*; too long.

An *ANcode* code cannot be more than seven characters in length. If you specify the full ANA code, then this cannot be more than eight characters in length.

glconfig: Error 1813 at line *linenumber* - TTY device name *devicename* too long.

Global System Manager only allows tty device names of up to 30 characters. If a device name exceeds this length then it must be abbreviated (possibly by linking) to a total length no longer than 30 characters.

glconfig: Error 1814 at line *linenumber* - TTY device *devicename* allocated to more than one USER.

The specified TTY *devicename* has been defined by a previous TTY description. Each *devicename* must be unique as Global System Manager uses this tty name as authorisation for a particular SYSTEM.

glconfig: Warning 1815 at line *linenumber* - Ignoring OPID *shortname* for LOGNAME *user*.
Already defined.

The user *user* has more than one short user name (operator ID) defined. The latter short user name (*shortname*) is ignored. Short user names may be defined by appending OPID *shortname* to a USER definition, or by the OPIDMAP table; but not by both, for the same user.

glconfig: Error 1816 at line *linenumber* - LOGNAME *user* too long.

The specified user name *user* is too long. Global System Manager only accepts user names of up to 14 characters in length.

glconfig: Error 1817 at line *linenumber* - LOGNAME *user* allocated to more than one USER.

More than one USER authorisation for the user *user* has been set up. Each user must have only one authorisation definition in the Systems file.

glconfig: Error 1818 at line *linenumber* - USER *number* for SYSTEM 0x*number* doubly defined.

If more than one USER is defined for a particular SYSTEM, the USER *number* must be unique.

glconfig: Error 1819 at line *linenumber* - USER *user* for SYSTEM *letter* doubly defined.

See error 1818, but for SYSTEMs A to Z.

glconfig: Error 1820 at line *linenumber* - Invalid PRINTER unit *number*.

PRINTER *number* numbers must be in the range 500 to 599.

glconfig: Error 1821 at line *linenumber* - PRINTER *number* doubly defined.

This PRINTER *number* has previously been defined. There can only be one description for each PRINTER *number*.

glconfig: Warning 1822 at line *linenumber* - TERM *term* already defined.
Ignoring TAP number *number*.

A TERM entry for this *term* already exists, the latter definition and corresponding TAP number (*number*) will be ignored.

glconfig: Error 1823 at line *linenumber* - Invalid TAP number *number*

The specified TAP number *number* is not in the valid TAP range 0 to 9999.

glconfig: Warning 1824 at line *linenumber* - Ignoring OPID *shortname* for LOGNAME *user*

Already defined.

See error 1815.

glconfig: Error 1825 at line *linenumber* - OPID *shortname* too long.

The specified short user name *shortname* is too long. It must be no more than four characters in length.

glconfig: Error 1826 - Can't recompile Systems file. Global System Manager in use.

Configuration file not created.

The Global System Manager initialisation data files contained within the directory $GLDIR/sys/data must not be modified whilst users are running Global System Manager and the Global software. Note that Global System Manager may still be in use even if all users are logged out. This is because Global System Manager allows users to run the Global applications concurrently with other non Global software (in batch mode). The application continues running until the user explicitly requests its termination ($BYE).

If all users are logged out, run Global System Manager and use the $STATUS utility to ensure that no files are in use, before removing any other Global System Manager processes via glclean. Using glclean whilst files are in use may result in corrupt Global files.

glconfig: Error 1827 - Can't read directory *$GLDIR*/sys.
Configuration file not created.

You do not have permissions to access the directory *$GLDIR*/sys. glconfig can only be executed by the users global and root. Ensure the permissions are set up for access by both these users.

glconfig: Error 1828 - No Systems file.
Configuration file not created.

The file *$GLDIR*/sys/Systems could not be found. Ensure that this file exists, and has read permissions set for this user.

glconfig: Error 1829 - No data directory.
Configuration file not created.

This user does not have read, write and execute permissions for the directory *$GLDIR*/sys/data, or the directory does not exist. This directory contains the compiled Systems file data and must exist.

glconfig: Error 1830 - SYSTEM A has no SYSDATA.
Configuration file not created.

There is no SYSDATA *datafile* definition for SYSTEM A. SYSTEM A must always contain a SYSDATA definition. This is the default YSDATA for all SYSTEMs.

glconfig: Error 1831 - SYSTEM A must have a DATA or DISKETTE definition.
Configuration file not created.

SYSTEM A must have a DATA definition as the Global System Manager initialisation data and Global utilities are set up on these data files.

glconfig: Error 1832 - SYSTEM *letter* must have a DATA or USER definition.
Configuration file not created.

This SYSTEM definition has no definitions for either DISK, DISKETTE, or USER. This SYSTEM must be either a file server having only DISK or DISKETTE definitions, or a user having only USER definitions.

glconfig: Error 1833 - SYSTEM *letter* cannot be a non dedicated file server.
Configuration file not created.

The SYSTEM has both DATA (or DISKETTE) definitions, and USER definitions. It must only have one or the other. SYSTEMs A to Z are usually reserved for data file allocation.

glconfig: Error 1834 - SYSTEM 0x*number* must have at least one USER definition.
Configuration file not created.

Non file server SYSTEMs (those outside the range A to Z) are used to describe the Global System Manager configuration for a specified USER. Without a USER definition this SYSTEM description would do nothing.

glconfig: Error 1835 - Can't overwrite data/*filename*.
Configuration file not created.

Insufficient permissions to delete the file data/*filename*.

glconfig: Error 1836 - Can't create data/*filenam*e.
Configuration file not created.

Insufficient permissions to create the file data/*filename*.

glconfig: Error 1837 - Can't delete temporary file datatmp/*filename*.
Configuration file not created.

In fact the configuration files in $GLDIR/sys/data have been updated. For some reason (lack of permissions) the temporary file $GLDIR/sys/datatmp/*filename* could not be removed. Remove the directory datatmp and all its contents, by hand.

glconfig: Error 1838 at line *linenumber* - *error message*.

The specified *error message* occurred at the specified *linenumber*. The most common form is syntax error. The line has either an unknown command or incorrect parameters.

glconfig: Error 1839 - GLDIR not defined.
Configuration file not created.

You do not have the shell variable GLDIR defined and exported. This variable is required by all Global System Manager programs, and must point to the global directory (see section 8.1.1).

Note that this message will normally appear as:

Global System Manager: Error 1839
(can't open error header file "sys/errmsg/hdr")

glconfig: Error 1840 at line *linenumber* - SHMSIZE already defined.

SHMSIZE has been defined more than once.

glconfig: Warning 1841 at line *linenumber* - SHMSIZE should be at least *number*.

You have defined a value for SHMSIZE that is smaller than the suggested minimum. If you only have a small Global System Manager configuration, a smaller size may be sufficient. However you may find that Global System Manager returns a memory error when configuring Global System Manager resources.

Each file server SYSTEM requires approximately 10K of shared memory. Each USER requires approximately 300 bytes.

C.6 Messages from "glsysdump" and "glshmdump"

glsysdump: Error 1900 - GLDIR not defined.

You do not have the shell variable GLDIR defined and exported. This variable must be set to the path of the global directory.

Note that this message will normally appear as:

Global System Manager: Error 1900
(can't open error header file "sys/errmsg/hdr")

glshmdump: Error 1901 - GLDIR not defined.

You do not have the shell variable GLDIR defined and exported. This variable must be set to the path of the global directory.

Note that this message will normally appear as:

Global System Manager: Error 1901
(can't open error header file "sys/errmsg/hdr")

glshmdump: Error 1902 - Cannot attach to shared memory (IPC inconsistency)

Either Global System Manager is not running or an internal error has occurred.

glshmdump: Error 1903 - SHM mapped to different address

An internal error has occurred. You will not be able to use glshmdump until Global System Manager has been re-loaded.

C.7 Messages from "glspod"

Most of the following messages will not appear on the screen but are written to the $GLDIR/sys/spool/log log file (see section 6.1.1.8.2).

glspod: Warning 2000 - *date/time* Error spooling print file (*retval*). Retrying command:

spooler command string

A "soft error" was returned from the shell when attempting to execute the spooler shell command string (see section E.1). *retval* is the result returned from the Unix system function. glspod will wait for 3 seconds before retrying the command. The errors (in errno) that are treated as "soft errors" are:

EAGAIN No more processes available
ENOMEM Not enough memory available

glspod: Error 2001 - *date/time* Failed execution (*retval*) of command:
spooler command string

A fatal error was returned from the shell when attempting to execute the spooler shell command string (see section E.1). *retval* is the result returned from the Unix system function.

glspod: Diagnostics 2002 - Initiating glspod at *date/time*

This message is written to the spooler log file to indicate that the glspod daemon is running.

glspod: Error 2003 - Unable to open spooler lock file

The glspod daemon has been unable to open the $GLDIR/sys/spool/gl_LCK_002 lock file.

glspod: Error 2004 - Unable to open spooler queue file

The glspod daemon has been unable to open the $GLDIR/sys/spool/queue file.

glspod: Error 2005 - Unable to open spooler FIFO

The glspod daemon has been unable to open the $GLDIR/tmp/gl_FIFO_5*xx* FIFO.

glspod: Error 2006 - Unable to attach to shared memory

The glspod daemon has been unable to attach to the shared memory created by the glintd process.

glspod: Error 2007 - Unable to create new instantiation

An internal error has occurred within the glspod daemon process. Re-install the Global System Manager BACNAT software.

glspod: Diagnostics 2008 - Exiting glspod normally

The glspod process has detected a "closedown" signal sent by glclean.

glspod: Error 2009 - spoolnext can't lock the queue file

An internal error has occurred within the glspod daemon process. Re-install the Global System Manager BACNAT software.

glspod: Error 2010 - spoolnext has detected new glspod running

An internal error has occurred within the glspod daemon process. Re-install the Global System Manager BACNAT software.

glspod: Warning 2011 - spoolnext has detected NULL command string

An internal error has occurred within the glspod daemon process. This warning message is included for information purposes only and can be ignored.

glspod: Diagnostics 2012 - *date/time* spoolnext command:
spooler command string

The spoolnext function within the glspod daemon has been called to process the command string.

glspod: Error 2013 - Unable to stat spooler lock file

The glspod daemon has been unable to determine the size of the $GLDIR/sys/spool/gl_LCK_002 lock file.

glspod: Error 2014 - Spooler lock file is wrong size

The size of the $GLDIR/sys/spool/gl_LCK_002 lock file is incorrect. Unload Global System Manager and delete this file.

glspod: Diagnostics 2020 - spooler process is running...

This message is displayed on the screen to indicate the execution path within the glspod daemon process if the GSM_VC2GG shell variable is defined (see section G.86). This message will not appear in the $GLDIR/sys/spool/log log file.

glspod: Diagnostics 2021 - Can't get *filenames*

This message is displayed on the screen to indicate the execution path within the glspod daemon process if the GSM_VC2GG shell variable is defined (see section G.86). This message will not appear in the $GLDIR/sys/spool/log log file.

glspod: Diagnostics 2022 - Can't chdir to spool directory

This message is displayed on the screen to indicate the execution path within the glspod daemon process if the GSM_VC2GG shell variable is defined (see section G.86). This message will not appear in the $GLDIR/sys/spool/log log file.

glspod: Diagnostics 2023 - Can't open or dup to log file

This message is displayed on the screen to indicate the execution path within the glspod daemon process if the GSM_VC2GG shell variable is defined (see section G.86). This message will not appear in the $GLDIR/sys/spool/log log file.

C.8 Messages from "glreorg"

The following messages may appear on the screen when glreorg is used.

glreorg: Error 2100 - GLDIR not defined.

You do not have the shell variable GLDIR defined and exported. This variable must be set to the path of the global directory.

Note that this message will normally appear as:

Global System Manager: Error 2100
(can't open error header file "sys/errmsg/hdr")

glreorg: Error 2101 - Can't reorganise whilst Global System Manager is in use

The Global System Manager data files containing within $GLDIR/data/A00.dir, for example, must not be reorganised whilst users are running Global System Manager and the Global software. Note that Global System Manager may still be in use even if all users are logged out. This is because Global System Manager allows users to run the Global applications concurrently with other non Global software (in batch mode). The application continues running until the user explicitly requests its termination ($BYE).

If all users are logged out, run Global System Manager and use the $STATUS utility to ensure that no files are in use, before removing any other Global System Manager processes via glclean. Using glclean whilst files are in use may result in corrupt Global files.

glreorg: Error 2102 - Invalid size argument

The size argument specified on the command line is invalid. It must be specified in one of the formats described in section 6.10.

glreorg: Warning 2103 - Cannot stat file:
filename

The Unix stat function has returned an error when glreorg attempted to determine the size of file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2104 - Illegal file name for SVL file:
filename

glreorg can only reorganise Global System Manager sub-volume files with a file name of the following general form:

SVL*nn_vvvvvv*

where "SVL" and "_" are those characters, *nn* is any pair of numeric characters (excluding "00") and *vvvvvv* represents any character string.

glreorg: Warning 2105 - Cannot change size of domain header file:
filename

An attempt has been made to change the size of the domain header file:

SVL00_*dddddd*

The size of the domain header file must not be modified.

glreorg: Diagnostics 2106 - Expanded absolute filename:
filename

This message only appears if the glreorg "-d" command line argument has been specified. The full pathname of the file currently being reorganised is displayed.

glreorg: Diagnostics 2107 - Relative filename:
filename

This message only appears if the glreorg "-d" command line argument has been specified. The relative name of the file currently being reorganised is displayed.

glreorg: Warning 2108 - Old filesize not multiple of GSM track size:
filename

This message will appear if the current size of the sub-volume file *filename* is not a multiple of the Global System Manager virtual track-size (i.e. 8Kb). The filesize will be automatically rounded up to an exact multiple of 8Kb.

glreorg: Warning 2109 - New filesize not multiple of GSM track size:
filename

This message will appear if the requested size of the sub-volume file *filename* is not a multiple of the Global System Manager virtual track-size (i.e. 8Kb). The filesize will be automatically rounded up to an exact multiple of 8Kb. This message will appear if the new (invalid) filesize has been specified as an absolute value (e.g. 1k), an incremented value (e.g. +1k) or a decremented value (e.g. -1k).

glreorg: Warning 2110 - Unexpected default from case statement:
filename

An internal logic error has occurred in glreorg. This message is included for completeness only.

glreorg: Warning 2111 - New filesize cannot be negative or zero:
filename

This message will appear if the requested size of the sub-volume file *filename* is nonsensical.

glreorg: Warning 2112 - New filesize not multiple of GSM track size:
filename

This message will appear if the requested size of the sub-volume file *filename* is not a multiple of the Global System Manager virtual track-size (i.e. 8Kb). The filesize will be automatically rounded down to an exact multiple of 8Kb. This message will appear if the new (invalid) filesize has been calculated as a result of a filesize division (e.g. the new filesize was specified as /2).

glreorg: Warning 2113 - Illegal percentage:
filename

The value of the requested file size, when specified as a percentage, must be an integer between 1 and 100, inclusive.

glreorg: Warning 2114 - Unexpected default from case statement:
filename

An internal logic error has occurred in glreorg. This message is included for completeness only.

glreorg: Warning 2115 - Unable to open file for extending:
filename

The Unix open function has returned an error when glreorg attempted to open file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2116 - Unable to write file for extending:
filename

The Unix write function has returned an error when glreorg attempted to write to file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2117 - Unable to open file for rounding up:
filename

The Unix open function has returned an error when glreorg attempted to open file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2118 - Unable to write file for rounding up:
filename

The Unix write function has returned an error when glreorg attempted to write to file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Error 2119 - Can't malloc 8Kb track buffer

glreorg has been unable to allocate 8Kb of free memory.

glreorg: Warning 2120 - Unable to open file for directory scan:
filename

The Unix open function has returned an error when glreorg attempted to open file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2121 - Unable to read file for directory scan:
filename

The Unix read function has returned an error when glreorg attempted to read from file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2122 - New filesize same as old filesize:
filename

The requested file size is the same as the current file size (after possible rounding up). No further action will be taken.

glreorg: Warning 2123 - Cannot truncate volume below extent of last GSM file:
filename

A request has been made to truncate a sub-volume file below the extent of the last GSM file on the volume. Such an event, which is forbidden by glreorg, would corrupt the GSM files and directory on the sub-volume.

glreorg: Error 2124 - Illegal command string

The syntax of the glreorg command string is incorrect. A size argument does not follow immediately after the "-s" argument; or a *filename* argument does not follow immediately after the "-t" argument.

glreorg: Error 2125 - Invalid command line argument

The command string specified to the glreorg command contains an invalid argument. Only the arguments described in section 6.10 are allowed.

glreorg: Error 2126 - Reserved for future use

This message is reserved for future use.

glreorg: Error 2127 - Reserved for future use

This message is reserved for future use.

glreorg: Error 2128 - No filename argument(s) specified

The command line string must contain at least one *filename* argument.

glreorg: Error 2129 - No need to round filesize:
filename

glreorg has determined that there is no need to round up the size of file *filename*. This message will only appear if the "rounding only" option has been selected.

glreorg: Error 2130 - Invalid volume type:
filename

glreorg has attempted to read the GSM directory of the sub-volume simulated by file *filename* but has detected an invalid GSM directory.

glreorg: Warning 2131 - Unable to open old file for copying:
filename

The Unix open function has returned an error when glreorg attempted to open file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2132 - Unable to create temp file for copying:
filename

The Unix creat (*sic*) function has returned an error when glreorg attempted to create file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2133 - Unable to read old file for copying:
filename

The Unix read function has returned an error when glreorg attempted to read from file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2134 - Unable to write temp file for copying:
filename

The Unix write function has returned an error when glreorg attempted to write to file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2135 - Unable to unlink old file after copying:
filename

The Unix unlink function has returned an error when glreorg attempted to delete file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2136 - Unable to link old file to temp file after copying:
filename

The Unix link function has returned an error when glreorg attempted to rename file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2137 - Unable to unlink temp file after copying:
filename

The Unix unlink function has returned an error when glreorg attempted to delete file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2138 - Unable to open old file for moving:
filename

The Unix open function has returned an error when glreorg attempted to open file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2139 - Unable to create temp file for moving:
filename

The Unix creat (*sic*) function has returned an error when glreorg attempted to create file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2140 - Unable to read old file for moving:
filename

The Unix read function has returned an error when glreorg attempted to read from file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2141 - Unable to write temp file for moving:
filename

The Unix write function has returned an error when glreorg attempted to write to file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2142 - Unable to unlink old file after moving:
filename

The Unix unlink function has returned an error when glreorg attempted to delete file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2143 - Unable to reopen temp file after moving:
filename

The Unix open function has returned an error when glreorg attempted to re-open file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2144 - Unable to recreate new file after moving:
filename

The Unix creat (*sic*) function has returned an error when glreorg attempted to create file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2145 - Unable to read temp file for moving:
filename

The Unix read function has returned an error when glreorg attempted to read from file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2146 - Unable to write new file for moving:
filename

The Unix write function has returned an error when glreorg attempted to write to file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

glreorg: Warning 2147 - Unable to unlink temp file after moving:
filename

The Unix unlink function has returned an error when glreorg attempted to delete file *filename*. If the glreorg "-d" command line argument has been specified further diagnostic information will accompany this message.

Appendix D - Global System Manager Processes

This appendix describes the various Unix processes that run while Global System Manager is in use. It describes the general function of each process, their interaction and any files they may use.

The Global System Manager (Unix) "nucleus" can be considered as two sections:

- Global System Manager "core section" consisting of printer and timing daemons plus a shared memory area that contains control data and lock tables;

- Global System Manager "user section" consisting of each user's Global System Manager back-end and front-end process pair.

D.1 Global System Manager Core

This section describes the Unix processes and other resources that are not related to a specific user.

D.1.1 Shared Memory

Unix shared memory is used to hold all the control data for the shared Global System Manager resources. This data is held in shared memory rather than temporary files for efficiency reasons. It contains information about each Global user and their processes, but is mainly used to hold the Global data file and Global lock tables. See section 7.4.4.

D.1.2 glspod - Spooled Printer Output

This daemon process accepts command requests from the print queue that is used by all "spooled" printers (i.e. those printer defined as controller SPOOLED in the Global configuration file - see section 9.5.2). The *glspod* process executes the specified command on the specified file. Each printer unit may have a different command, and each user may have a different directory from which files are printed.

Each queued request is executed in turn. The command executed is specified for the printer unit in the Systems file. For example:

```
PRINTER 510 "lp -c -r"
```

The file to be printed is passed as the last parameter of this command unless the $FILE keyword is included in the string (see section E.1). The *glspod* daemon waits for the command to complete (i.e. the new process to terminate) before taking the next request from the queue. The example above invokes the Unix spooler to print the specified file, copying it first (usually via a link) and then removing the original copy. It is important that the command you specify removes the print file after printing, as the nature of some Unix spoolers make it impossible for *glspod* to know when the file can be deleted.

The command specified in the PRINTER definition would usually be a shell script created by the System Administrator (see section 7.3.6.2).

The glspod daemon writes records to the $GLDIR/sys/spool/log file (see section G.103.3) and includes diagnostics that may be enabled by setting a shell variable (see section G.86).

D.1.3 glprid - Direct Printer Output

This daemon process controls printing to a specific "direct" printer (i.e. a printer defined as controller DIRECT in the Global configuration file - see section 9.5.1). The *glprid* process opens and locks the device specified in the Systems file for the printer unit, then prints directly to the device. Once the print has completed the device is closed and released. This technique allows printers to be shared between Global System Manager (and Global applications) and other Unix applications, including the Unix spooler, that open the printer device exclusively during printing.

If the printer is being used heavily by a Unix spooler, the Unix spooler is likely to have priority over the Global prints, as *glprid* only polls the printer device once a second waiting for it to become free. **Global System Manager never reconfigures the printer port settings**. If the printer is not set up as a Unix spooler printer then the device characteristics (e.g. baud-rate, handshaking protocol etc.) must be configured using the facilities provided by your Unix administration system.

The *glprid* process controls Global System Manager printing for a single "direct" printer. Thus, in a Global System Manager (Unix) configuration that includes *N* DIRECT printers, there will be *N glprid* daemons executing (one for each DIRECT printer). Contrast the multiplicity of *glprid* daemons with the single *glspod* daemon (see section D.1.2) that controls printing to all the SPOOLED printers defined in the Global configuration file.

D.1.4 gltimd - Timing and Control Daemon

This daemon process handles various timing aspects of Global System Manager which include process coordination via signals.

D.1.5 glspad - Spawner Daemon (temporary)

This daemon process is used by *glintd* to load and initiate (i.e. spawn) the other daemon processes (i.e. *gltimd* and *glspod*). Once the *glspad* process has completed its task, it is killed and thus, should only appear fleetingly in a ps report (section D.3).

D.1.6 Lock file - $GLDIR/tmp/gl_LCK_001

This file is used when SYSTEMs are started by running *global*, and when SYSTEMs are terminated by running $BYE. It is required to ensure efficient shared memory usage.

D.1.7 Permanent Spooler Printer FIFO - $GLDIR/tmp/gl_FIFO_5*xx*

This FIFO is used by the *glintd* daemon to pass print lines to the *glspod* daemon (see section D.1.2).

D.1.8 Temporary Direct Printer FIFOs - $GLDIR/tmp/gl_FIFO_5*nn*

Where 5*nn* is the unit number of the direct printer. This FIFO is used by the *glintd* daemon to pass print lines to the *glspod* daemon (see section D.1.2).

D.1.9 Spooler Lock file - $GLDIR/sys/spool/gl_LCK_002

This file is used to control the spawning and use of the *glspod* spooled printer daemon (see section D.1.2).

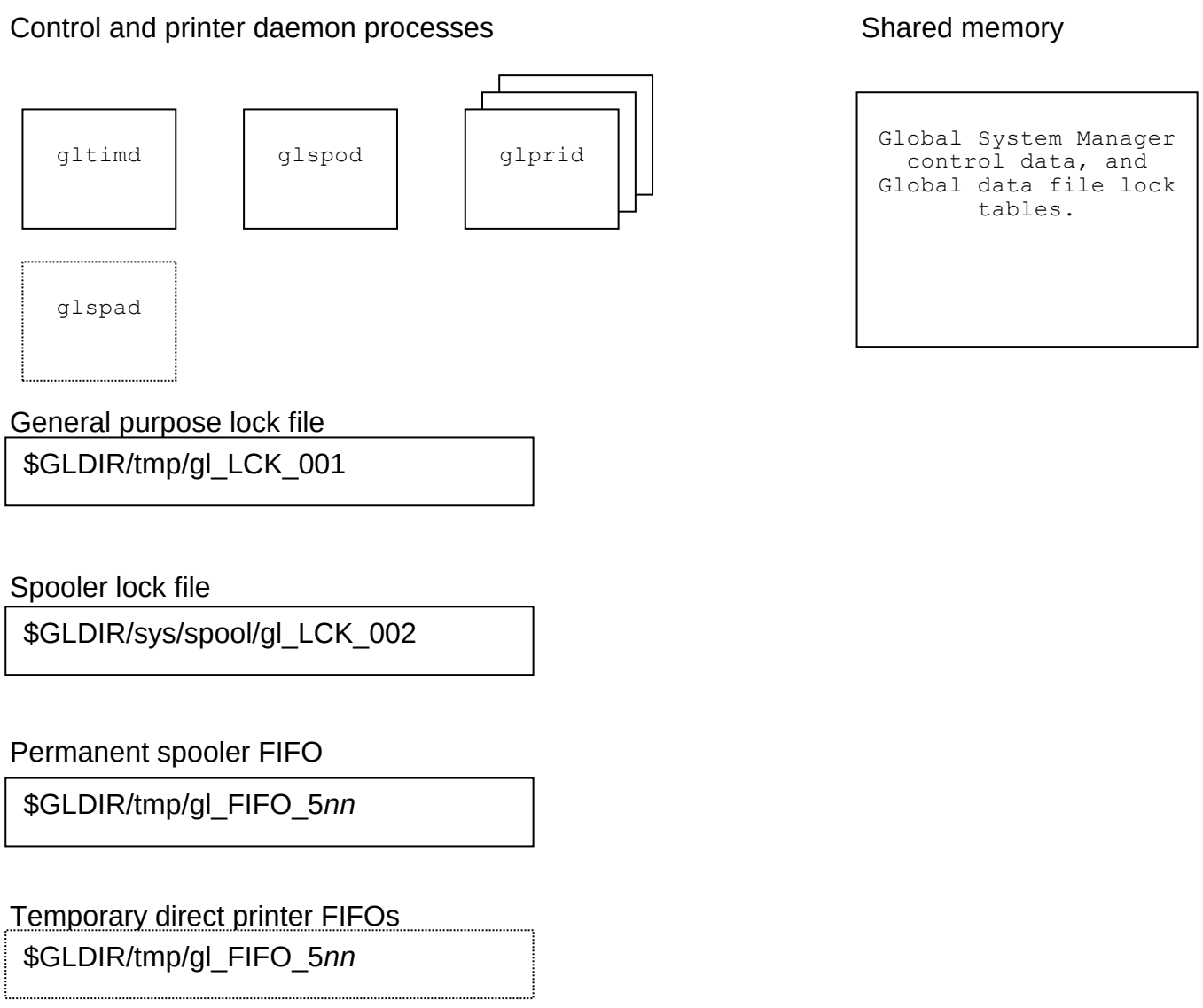


Figure D.1 - Global System Manager core processes and data

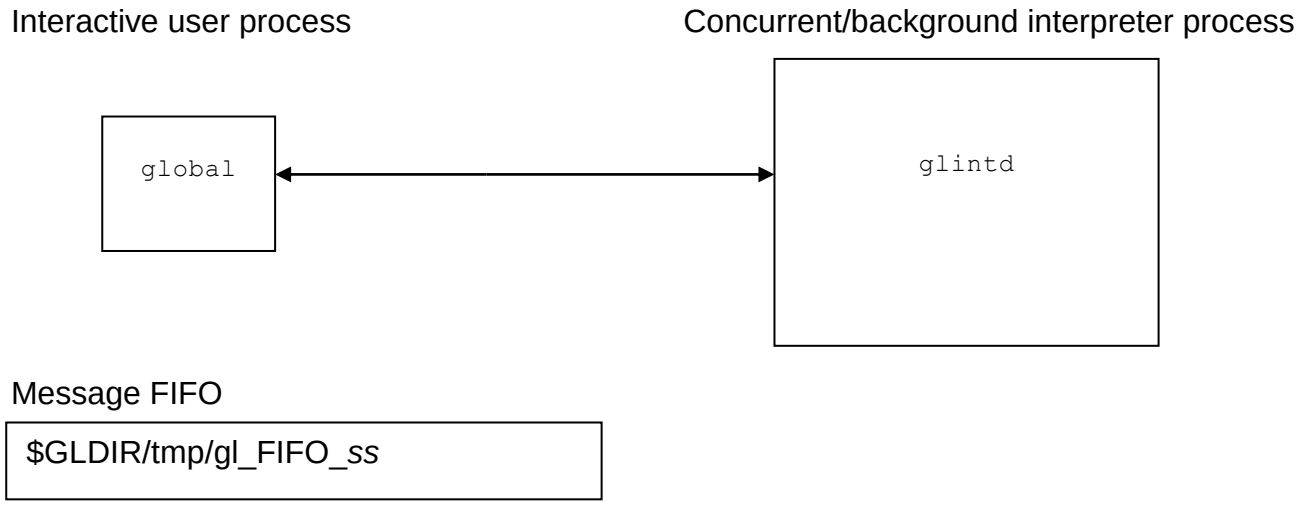


Figure D.2 - Global System Manager user processes and data

D.2 Global System Manager User

This section describes the Unix processes and other resources that are related to a specific Global user.

D.2.1 glintd - Global Interpreter Daemon

This daemon process runs the Global Cobol interpreter, Global System Manager menu handler and controls the virtual screen partitions. To enable virtual screen handling, and hence, concurrent processing, this process must run in the background. The advantage of this technique is to allow Global applications to execute (in a kind of batch mode), whilst the user runs other, possibly non-Global, applications. Any screen output performed by the Global application is stored in the virtual partition and upon exiting the other application, the Global application is redisplayed.

This process should be removed by running $BYE before logging out of Unix, or by glclean once all users are logged out.

D.2.2 global - Run Global System Manager

global is the front-end process which performs the Global System Manager authorisation through the $GLDIR/sys/data files, initiates the *glintd* interpreter daemon, performs input for Global System Manager and performs the shell execution for commands such as <SYSREQ> . (see section 5.4) and utilities such as %.SHCMD (see section 5.1) and %.SHELL (see section 5.2).

While executing a shell command or just creating a new shell from Global System Manager, this process suspends after creating the new shell command or shell, and resumes when the new shell exits. The *global* process terminates when the utility $E is run even though *glintd* is still running (although it is likely to be hibernating). When *global* is next run it simply wakes *glintd* and resumes execution.

D.2.3 Message FIFO - $GLDIR/tmp/gl_FIFO_ss

Where *ss* is the SYSTEM number of the *glintd* process that created this file. This FIFO is used to allow the *glintd* daemons to communicate with each other. For example to send messages to other users from SYSTEM D:

```
GSM READY:$STATUS
...
...
$47 TASK CONTROL:MES
SEND MESSAGE:PLEASE LOGOUT NOW
TO USER:^B SYSTEM-ID, <CTRL B> FOR ALL (D ):^B
TO USER:
```

This FIFO is removed by $BYE or glclean.

D.3 Example ps reports

The Unix *ps* utility displays the status of current processes. A description of *ps* or Unix process handling is beyond the scope of this manual (refer to your Unix documentation for further information) but this section contains some example "ps reports" obtained while Global System Manager was "in use". The actual example was obtained on "SCO Unix System V/386 release 3.2 version 2" but similar results would be obtained on all the versions of Unix currently supported by Global System Manager.

The Systems file used to obtain these results included the following lines:

```
SYSTEM A
                  DATA  0     A00.dir
SYSTEM 0x1b       USER  1     LOGNAME     global1
SYSTEM 0x1c       USER  1     LOGNAME     global2

PRINTER
      500   /dev/null
      501   /dev/null
      510   "cat > /dev/null"
      511   "cat > /dev/null"
```

D.3.1 ps report (1 active SYSTEM)

This report was obtained immediately after user *global1* had invoked Global System Manager (using the *gl* command):

UID	*PID*	*PPID*	*COMMAND*
global1	370	260	gl
global1	371	370	gltimd
global1	377	370	glintd 27
global1	374	1	glprid 500 /usr/global/tmp/gl_FIFO_500
global1	375	1	glprid 501 /usr/global/tmp/gl_FIFO_501
global1	376	1	glspod

D.3.2 ps report (2 active SYSTEM's)

This report was obtained immediately after user *global2* invoked Global System Manager (using the *GL* command):

UID	*PID*	*PPID*	*COMMAND*
global1	370	260	gl
global1	371	370	gltimd
global1	377	370	glintd 27
global1	374	1	glprid 500 /usr/global/tmp/gl_FIFO_500
global1	375	1	glprid 501 /usr/global/tmp/gl_FIFO_501
global1	376	1	glspod
global2	**387**	**261**	**GL**
global2	**388**	**387**	**glintd 28**

Appendix E - Global System Manager Printer Interfaces

Global System Manager and the Global applications include various techniques for printing documents and reports. This appendix describes each technique together with the type of print files for which the technique is best suited.

Note the printer device (port) must be configured (for correct baud rate etc.) by your System Administrator using the appropriate Unix utilities. Global System Manager expects the printer to be configured correctly by Unix.

E.1 The Unix Spooler

Print files can be sent to the Unix spooler. However, as most Unix spoolers do not prompt for changes of stationery (or font cartridges etc.) the only print files that should be sent to this spooler are those that require no special sequences to select a particular printer mode. There is also no guarantee when the file will be printed, so this mechanism should not be used if you need to know exactly when the file is printed.

This section describes how to configure Global System Manager to print via a Unix spooler. Global System Manager can be configured to print via many Unix spoolers/printers. However, the procedure for each is the same as that described below.

The printer unit number that will be used to send files to the Unix spooler must be defined in the configuration file PRINTER ATTRIBUTES section as CONTROLLER type SPOOLED (see section 9.5.2). The spooler command, or Unix shell script, which is required to print files to this spooler must then be set up in the Systems file (see section 7.3.6.2). For example, if you wish to set up the Global printer unit 510 to output to the Unix spooler, the configuration and Systems file entries would look like this:

Configuration file:

```
PRINTER ATTRIBUTES

CONTROLLER (SPOOLED ) : Spooled printer output
UNIT NUMBER ( 510)
DESCRIPTION (Spooled printer output )
HARDWARE FORM FEED (Y)
MAXIMUM PAGE WIDTH ( 132)
TIME-OUT IN TENS OF SECONDS ( 2)
```

Systems file:

```
PRINTER 510 "lpr -r"
```

All print files sent to printer unit 510 will now be printed by the shell command:

```
$ lpr -r printfile
```

where *printfile* is the file to be printed. lpr -r is an example of a Unix spooler command which, in this example, prints the file to the system specified printer and removes the file afterwards.

When the Global application prints to a printer unit of type SPOOLED, by default the print file is created in the directory $GLDIR/spool. However, if the variable $GLSP*nnn* (where *nnn* is the printer unit number) is defined (see section 8.4.1) the print file is created in the directory

specified by this variable. Note that the value of the variable is only read once when the *glintd* (see section D.2.1) process is created. This variable allows users to have private spool directories.

Global System Manager cannot remove the print files as it does not know when the file has been printed. **You must therefore ensure that the print file is removed by the Unix command executed for this printer unit (defined in the Systems file).**

The format of the print file name is as follows:

aaaaaaaa.c.nn

or:

aaaaaaaa.xx.nn

where *aaaaaaaa* is the Global print file name (up to eight characters) with any illegal characters removed (the only legal characters are alphanumeric and full stop), *c* is a SYSTEM letter A to Z, *xx* is a SYSTEM number in hexadecimal and *nn* is a sequence number from 01 to 99. The sequence number is to allow multiple copies of the same print file name to be created. The sequence number is restricted to 99 to prevent the spool directory becoming cluttered with print files (which may occupy a large amount of disk space). If an attempt is made to create a file with a sequence number greater than 99, a DIRECTORY FULL error will be reported to the user.

If your Unix spooler does not have a facility to remove a file after printing then you should create a Unix shell script to be used in place of the spooler command in the Systems file. You can then tailor this script to invoke the Unix spooler and subsequently remove the print file (or perform any other operation you wish). For example:

1. Define a shell script for the printer unit you require (in the Systems file), for example unit 510:

```
PRINTER 510 "gllp510"
```

2. Now create the executable shell script called gllp510 in $GLDIR/bin:

```
# vi $GLDIR/bin/gllp510
```

3. Enter the commands:

```
lp -c -ddevice $1          (copy then print the file)
rm $1                      (delete file from work directory)
```

4. Exit from the editor and then make the file executable:

```
# chmod u+x $GLDIR/bin/gllp510
```

This script will cause any files to be printed (at the leisure of lp), but the file will be immediately removed from your directory, and glspod will service the next request on the queue.

An alternative technique, using the $FILE string substitution mechanism, which does not require the creation of shell script, is described in section 7.3.6.2.

Use of SPOOLED output via a Unix spooler from the Global System Manager spooler, $SP, is not normally practical. It is NOT recommended as any mount stationery requests will not necessarily coincide with the immediate state of the physical printer.

Appendix G includes further information regarding the SPOOLED printer interface.

E.2 Direct Unix Printing

This section describes how to configure Global System Manager to print directly to a Unix print device. Global System Manager can be configured to print to many Unix print devices. However, the procedure for each is the same as that described below.

When setting up a direct printer, the printer number that will be used for the printer must be defined in the configuration file PRINTER ATTRIBUTES section as CONTROLLER type DIRECT (see section 9.5.1). The device to which the printer is connected must then be set up in the Systems file (see section 7.3.6.1). For example, to set up printer unit 500 as a printer to be used by the Global spooler, the configuration and Systems file entries would look like this:

Configuration file:

```
PRINTER ATTRIBUTES

CONTROLLER (DIRECT ) : Direct printer output
UNIT NUMBER ( 500)
DESCRIPTION (Direct printer output )
HARDWARE FORM FEED (Y)
MAXIMUM PAGE WIDTH ( 132)
TIME-OUT IN TENS OF SECONDS ( 2)
```

Systems file:

```
PRINTER 500 /dev/ttypc01
```

The printer is connected to device /dev/ttypc01. Any Global System Manager spooler or Global application that prints to the printer unit 500 will print to this printer.

Appendix G includes further information regarding the DIRECT printer interface.

E.3 The Global System Manager Spooler ($SP)

The Global System Manager spooler, $SP, is a sophisticated, yet user-friendly, spooler which prompts the operator for changes of stationery (or font cartridges etc.) and waits until the required change has been made before attempting to print the file. This technique requires that the spooler "knows" the state of the printer at all times. Therefore, to make sensible use of the Global System Manager spooler it should be used in conjunction with a DIRECT printer dedicated for its use only. It is possible to use the Global System Manager spooler with a shared Unix printer or SPOOLED printer but this combination is NOT recommended.

Ideally, the Global System Manager spooler should be used with printers that require the mounting of special stationery. For example, invoices or documents that require printing on logo headed paper. Several Global System Manager spoolers can be used simultaneously. One spooler is required for each printer.

The suggested set up for Global System Manager spooler use is as follows:

1. Configure each printer unit required, as described previously. Create a special Unix user called glspool. The login should be set up to immediately run global. It may be desirable to prevent this user from using the <SYSREQ> . facility: This can be achieved by setting their shell variable $SHELL to be exit:

```
SHELL=exit
```

2. Add the following line to the Systems file:

```
SYSTEM S USER 1 LOGNAME glspool GLID GLSP
```

   This line adds the user glspool as SYSTEM S. Alternatively use the TTY command word if you require the spoolers to be run from a fixed terminal.

3. Use $CUS to set up the printer control files for this SYSTEM (see Chapter 6 of the Global System Manager Manual).

4. Set up a menu for this SYSTEM which can start the printers automatically (see Chapter 7 of the Global System Manager Manual).

5. Finally, use the automatic menu selection option of $AUTH to start each Global spooler when glspool signs on (see Chapter 6 of the Global System Manager Manual).

All the Global spoolers are now running from the same terminal. If the printers are geographically remote (e.g. in different rooms or buildings) it may be more sensible to set up the user/terminal nearest the printer to run the Global spooler for that printer. This can be done without disturbing the user by taking advantage of Global System Manager's virtual screen partitions.

E.4 Global System Manager Direct Printing

Global System Manager direct printing is only recommended for printers that have no special stationery requirements or control sequences. This method of printing might be preferred over the Unix spooler if the user wishes to receive the printout immediately. The printer can be shared with other Unix applications but these applications must not re-program the printer or keep the printer open whilst they are not using it.

To configure Global System Manager to print directly the Global configuration file and Systems file should be set up as described in section E.2.

E.5 Printer Control Files

Printer Control Files, in general, are described in Chapter 6 of the Global System Manager Manual. Section 4.2.5 of this manual describes the special considerations that must be applied to Printer Control Files on a Global System Manager (Unix) configuration. Although the combination of a Printer Control File with a SPOOLED printer is not recommended, their use is sometimes unavoidable. A problem occurs when stationery mount messages are displayed by Global System Manager. These messages are displayed before the print report is sent to the Unix spooler and NOT when the Unix spooler is about to print the report to the physical printer.

If a physical printer is shared between Global System Manager and other Unix applications, it may be necessary to configure Global System Manager to reset the printer format characteristics (e.g. font, paper orientation etc.) before printing each report. This is necessary

because Global System Manager can never "know" the state of the printer if another Unix printer has used it.

Under these circumstances, $CUS should be employed to set the "Stationery Sequences Sent Always" flag in the Printer Control File to Y (see Chapter 6 of the Global System Manager Manual). It may also be desirable to set the "Mount Messages Never Displayed" flag to Y also if only one type of paper is being used.

Appendix F - Example Systems Files

This appendix should be read in conjunction with Chapter 7.

F.1 The Standard Systems File

The actual Systems file ($GLDIR/sys/Systems) distributed with Global System Manager is dependent on the version of Unix. For example, the DISKETTE definitions will vary according to the configuration code (see the Global Configuration Notes for further details). The following Systems file is typical:

```
#/*************************************************************/
#/*
#/* Copyright TIS SOFTWARE LTD. (c) 1990
#/*
#/* Module vc2go.tm
#/*
#/* Version 3.1
#/*
#/* Last change 91/11/25 Time 11:54:03
#/*
#/* Current date 95/09/08 Time 16:09:01
#/*
#/*************************************************************/
# Systems - text file containing the configuration data
# for Global System Manager.

#
# On some large configurations the default size used to allocate the
# shared memory may be insufficient. If this is the case remake the
# kernel and define SHMSIZE to the appropriate value. SHMSIZE is in
# K bytes.
# The default is 128K.
# SHMSIZE n

# When the system is installed it uses the device defined by INSDEV.
#
# INSDEV devicename
INSDEV /dev/rfd0135ds18

# For each system we have a SYSTEM heading followed by the system id.
#
# SYSTEM id
SYSTEM A

# Each system requires a SYSDATA file.
# This defines the location of the Global System Manager
# configuration data.
# Generally all systems use the SYSDATA defined for SYSTEM A, and
# therefore do not explicitly define SYSDATA.
#
# SYSDATA datafile
SYSDATA A00.dir

# For each file server SYSTEM there is a list of DATA files which
# are assigned to that system. These are numbered from 0 and have
# their data format defined by the Global System Manager
# configuration file # on SYSDATA.
# It is MORE EFFICIENT to add new data files as new file servers
# rather than adding them to existing file servers.
#
# DATA n pathname
DATA
 0 A00.dir

# For each file server SYSTEM there is a list of DISKETTE's which
# are assigned to that system.
# Each AN code (or ANA code) is listed with a separate device name.
```

```
# The AN code describes the geometry of the diskette.
#
# DISKETTE n ancode devicename
DISKETTE
 0 B3 /dev/rfd0135ds9
 0 O2 /dev/rfd0135ds18

# If a SYSTEM is not a file server then the Global System Manager
# USER's are listed in terms of either user name or tty device.
# A screen cannot be associated with both a user name and a tty
# device.
# File servers must be dedicated (must not have USER definitions),
# except file server SYSTEM A.
#
# USER n (TTY ttydevice | LOGNAME username [OPID tag])
# Where tag is a short user description for Global System Manager.
# n defines the entry in the Global System Manager configuration
# file on SYSDATA which describes certain screen parameters for
# example, the number of virtual screens a user may have.
# USER 1 LOGNAME root

# End of descriptions for this SYSTEM

# The PRINTER definitions define the printers that are available to
# Global System Manager. If an argument contains spaces it must be
# quoted (placed between inverted commas).
#
# For spooled printers the directory that the files are spooled to
# can be defined by a shell variable GLSPnnn where nnn is the printer
# unit number. The default is $GLDIR/spool.
#
# PRINTER n devicename
PRINTER
 # direct printers
 500 /dev/ttya
 501 /dev/ttyb
 # spooled printers
 510 "laserlp"
 511 "lpr -r"

# The TERM entries describe the map between the Unix terminal name
# and Global System Manager TAP code. Global System Manager uses its
# own terminal information programs.
#
# TERM termname tap
TERM
      w5      163
      wy-50   163
      wyse50  163
      wy-60   169
      wyse60  169
      sun     163
      sun-cmd 0
      vt100   19
      vt200   182
      ansi    606

# For each user that is allowed to use Global System Manager we
# specify a tag to be used by Global System Manager to uniquely
# identify each user.
# This tag can be up to 4 characters in length. This tag can also be
# defined in the USER definition.
#
# OPIDMAP username tag
# Where username is the Unix user name
# This name is only user if the shell variable GLID does not exist.
OPIDMAP
 root ROOT

# A dedicated file server.
SYSTEM B DATA 0 B00.dir
# End of descriptions for this SYSTEM
```

```
# Examples of users defined by TTY or by LOGNAME. It is more usual
# to define users by LOGNAME especially when terminals are
# connected via an ethernet terminal server.
# SYSTEM 27 USER 1 TTY /dev/tty01
SYSTEM 0x1b USER 1 LOGNAME peterb OPID CAT
SYSTEM 0x1c USER 1 LOGNAME ronh OPID CHOP
SYSTEM 0x1d USER 1 LOGNAME eddiemac OPID MAC
SYSTEM 0x1e USER 2 LOGNAME johnh OPID HOLY
SYSTEM 0x1f USER 1 LOGNAME johnd OPID DEMP
SYSTEM 0x20 USER 1 LOGNAME davew OPID WEBO
SYSTEM 0x21 USER 2 LOGNAME tommyb OPID LOTT
SYSTEM 0x22 USER 1 LOGNAME charlie OPID KING
SYSTEM 0x23 USER 2 LOGNAME petero OPID OSSY
SYSTEM 0x24 USER 2 LOGNAME ianh OPID IIH
SYSTEM 0x25 USER 2 LOGNAME peterh OPID MARY
SYSTEM 0x26 USER 2 LOGNAME ruud OPID RUDI
SYSTEM 0x27 USER 1 TTY /dev/ttyd
SYSTEM 0x28 USER 1 TTY /dev/ttye
```

F.2 Example Systems Files

This section includes examples of various Systems files and describes the set up defined by each example. The examples illustrate the simple forms of each command.

F.2.1 A Minimum Systems File

A minimum configuration consists of SYSTEM A, its SYSDATA and a DATA file, and a USER SYSTEM:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir

SYSTEM 27 USER 1 LOGNAME root
```

The first three lines are obligatory. The second line:

```
SYSDATA A00.dir
```

specifies A00.dir as the data file that contains the SYSRES. This must be a data file defined for SYSTEM A. The third line:

```
DATA 0 A00.dir
```

defines this data file for SYSTEM A. The file (directory) is called A00.dir and its definition (volume format etc) is defined by the configuration file DATA FILE DEFINITION section as CONTROLLER type DATA, DRIVE 0. The unit number for this data file will be A*nn* where *nn* is defined by the configuration file definition.

Users are added as separate SYSTEM's. In the above example:

```
SYSTEM 27
```

is included, which defines:

```
USER 1
```

a user with the configuration file USER DISPLAY ATTRIBUTES defined by CONTROLLER type USER, User number 1:

```
LOGNAME root
```

defines this USER as the Unix user name *root*.

When the user root runs:

```
# global
```

Global System Manager will configure the file server A using the SYSRES on A00.dir and then run the user root as SYSTEM 27. The only Global data files available will be the units on data file A00. There will be no printers or diskettes.

F.2.2 Adding Diskettes

Diskettes must be added to a specified SYSTEM. This System will normally be a file server SYSTEM so that all users are able to access the diskettes. The following example adds to the Systems file described in F.1, one diskette drive definition with two diskette formats:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

SYSTEM 27 USER 1 LOGNAME root
```

The diskette formats O2 (for example O2A and O2B) and B3 have been defined. These have been included for file server A because:

```
SYSTEM A
```

is the previous SYSTEM definition in the file. These diskette units will be addressed as units *ann* (lower case A followed by a two digit number). The diskette geometry and unit numbers are defined in the configuration file DATA FILE DEFINITIONS section, CONTROLLER type DISKETTE, DRIVE 0 (the 0 corresponds to DISKETTE 0 in the Systems file definition above). The formats (O2 etc) must be defined in the configuration file for DRIVE 0 otherwise that format will not be available to the Global applications. The device name included after each format, for example:

```
O2 /dev/rfd0135ds18
```

specifies the device that will be accessed for this unit (format on this drive). Note that any devices for the same physical drive must be defined in the Systems file as the same DISKETTE *number* on the same SYSTEM. In the above example both formats are defined as DISKETTE 0 and should therefore be on the same physical drive.

F.2.3 Adding Printers

The printers are not defined in terms of a SYSTEM and so their definitions can be placed anywhere within the Systems file. The following example adds to the Systems file described in F.2, three printer outputs, two of which are direct to a printer device and one to the Unix spooler:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9
```

```
PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

SYSTEM 27 USER 1 LOGNAME root
```

Three printers have been added. The first two:

```
500 /dev/ttypc01
501 /dev/ttypc02
```

both define output directed to a printer device. The printer unit numbers 500 and 501 must correspond to a CONTROLLER type DIRECT in the PRINTER ATTRIBUTES section of the configuration file. Units 500 to 509 are by convention used as DIRECT printer outputs. The third printer definition:

```
510 "lpr -r"
```

defines printer output directed to the Unix spooler. The print file has the shell command (specified), and executes the command:

```
lpr -r printfile
```

Note that the quotes are required as the command contains a space character. The printer unit number 510 must correspond to a CONTROLLER type SPOOLED in the PRINTER ATTRIBUTES section of the configuration file.

All the printer units defined in the Systems file must be unique and in the range 500 to 599.

F.2.4 Adding Another Data File

If another data file is required it must be defined in the Systems file (see section 7.3.2 for the other factors that must be considered). The recommended method is to add a new file server system:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

SYSTEM B
DATA 0 B00.dir

PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

SYSTEM 27 USER 1 LOGNAME root
```

Here the SYSTEM:

```
SYSTEM B
```

has been added, and for this SYSTEM the data file:

```
DATA 0 B00.dir
```

B00.dir has been defined. The file (directory) is called B00.dir and its definition (volume format etc) is defined by the configuration file DATA FILE DEFINITION section as CONTROLLER type DATA, DRIVE 0. The unit number for this data file will be B*nn* where *nn* is defined by the configuration file definition.

It is possible to add another data file to an existing file server SYSTEM but only if the configuration file has a corresponding CONTROLLER type DATA definition (where the DATA *number* in the Systems file corresponds to DRIVE *number* in the configuration file). The Systems file entry for a second data file for file server SYSTEM A:

```
DATA 1 A60.dir
```

after the DATA 0 entry, would define a second data file A60.dir. Note that the DATA *number* must be unique.

F.2.5 Adding More Users

It is strongly recommended to add Users as separate SYSTEM's. Users can be added either via their user name or as a specified terminal. The example below shows both the authorisation for a user *john* and for any user from the terminal connected to device /dev/ttypc05:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

SYSTEM B
DATA 0 B00.dir

PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

SYSTEM 27 USER 1 LOGNAME root
SYSTEM 28 USER 1 LOGNAME john
SYSTEM 29 USER 1 TTY /dev/ttypc05
```

The user *john* can now run Global System Manager as SYSTEM 28 and with the USER DISPLAY ATTRIBUTES defined by User number 1. Note that all the user names must be unique.

Any user (excluding those explicitly defined by LOGNAME) who runs Global System Manager from the tty device /dev/ttypc05 will run as SYSTEM 29.

If a different set of USER DISPLAY ATTRIBUTES are required for a particular user (for example, if a different number of virtual partitions are required) then a configuration file CONTROLLER type USER with a different User number should be used. For example:

```
SYSTEM 28 USER 1 LOGNAME john
```

becomes:

```
SYSTEM 28 USER 2 LOGNAME john
```

The user *john* can now run Global System Manager as SYSTEM 28 and with the USER DISPLAY ATTRIBUTES defined by User number 2. This description must be defined in the configuration file otherwise it will be ignored preventing the user *john* from running Global System Manager.

F.2.6 Setting Up Short User Names

Default short user names can be set up in the Systems file. The name for each user can be set up in one of two ways:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

SYSTEM B
DATA 0 B00.dir

PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

OPIDMAP
john JOHN

SYSTEM 27 USER 1 LOGNAME root OPID ROOT
SYSTEM 28 USER 2 LOGNAME john
SYSTEM 29 USER 1 TTY /dev/ttypc05
```

The command:

```
OPIDMAP
```

is added which defines a list of Unix user name to short user name maps. The only one in this example is:

```
john JOHN
```

which defines the user *john* as having the Global System Manager short user name JOHN (short user names can be up to four characters in length).

Alternatively, the short user name can be defined by appending it to the USER definition:

```
USER 1 LOGNAME root OPID ROOT
```

where:

```
OPID ROOT
```

is appended to define the short user name for the user root as ROOT. OPID is the required keyword here. Each user must have only one short user name defined. Note that although more than one user can have the same short user name, Global System Manager will prevent them from running at the same time.

F.2.7 Terminal Types

Global System Manager uses its own terminal types referred to as Terminal Attribute Programs (TAP's). Global System Manager must know which of its TAP's corresponds to the TERM types defined in the Unix system. This mapping is achieved by adding a list to the Systems file specifying the TERM name followed by the TAP number:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

SYSTEM B
DATA 0 B00.dir

PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

OPIDMAP
john JOHN

TERM
vt100  19
wyse50 163
vt200  181
vt300  182

SYSTEM 27 USER 1 LOGNAME root OPID ROOT
SYSTEM 28 USER 2 LOGNAME john
SYSTEM 29 USER 1 TTY /dev/ttypc05
```

The command:

```
TERM
```

is followed by a list of TERM to TAP conversions:

```
vt100  19
wyse50 163
vt200  181
vt300  182
```

The TERM name in the left hand column is the name given by:

```
$ echo $TERM
vt100
$
```

in this case vt100. The number in the right hand column is the TAP number. This TAP must be included in the TAP library P.$TAP on SYSRES. Note that the TAP numbers are specified during Global System Manager installation (see section A.15).

F.2.8 Installation Device

In order to install Global System Manager, the Systems file must include an entry describing the installation device (or file in the case of tape distribution):

```
INSDEV /dev/rfd0135ds18

SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

SYSTEM B
DATA 0 B00.dir
```

```
PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

OPIDMAP
john JOHN

TERM
vt100 19
wyse50 163
vt200 181
vt300 182

SYSTEM 27 USER 1 LOGNAME root OPID ROOT
SYSTEM 28 USER 2 LOGNAME john
SYSTEM 29 USER 1 TTY /dev/ttypc05
```

The command:

```
INSDEV
```

defines the installation device as:

```
/dev/rfd0135ds18
```

which is a O2 format diskette device, as shown by the line:

```
DISKETTE 0 O2 /dev/rfd0135ds18
```

For tape distribution the installation device must be defined as:

```
INSDEV A60.dir
```

In both cases file server SYSTEM A must have the specified device defined. For tape distribution this would require the addition of the following line to the Systems file:

```
DATA 1 A60.dir
```

The Global configuration file must include a corresponding DATA type CONTROLLER definition with a DRIVE number of 1.

F.2.9 Adding Users to SYSTEM A

Although, not a recommended technique, it may be necessary to add a User to SYSTEM A in order to access data units in the range 200 to 299. The following example changes the Systems file described in F.1 to move the User to SYSTEM A:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
USER 1 LOGNAME root
```

Note that SYSTEMs B to Z are dedicated file servers and cannot include USER definitions.

F.2.10 Private Data Files

Non file server SYSTEM definitions (SYSTEMs outside the range A to Z) can have private data files defined in the Systems file. Only USERs defined within the particular SYSTEM can access these data files. For example:

```
SYSTEM 0x1b
DATA 0 1b00.dir
USER 1 LOGNAME private
```

Only user *private* will be able to access the data files within directory 1b00.dir (as units in the range 200-299).

F.2.11 Adding multiple Users to a SYSTEM

Although, not a recommended technique, it is possible to include more than one USER per SYSTEM. To achieve this, simply add more USER definitions to the list for a SYSTEM ensuring that the USER number is unique and corresponds to a set of USER DISPLAY ATTRIBUTES in the configuration file. For example:

```
SYSTEM 0x1b
USER 1 LOGNAME global1
SYSTEM 0x1c
USER 1 LOGNAME global2
SYSTEM 0x1d
USER 1 LOGNAME global3
USER 2 LOGNAME global4
USER 3 LOGNAME global5
```

This technique is not recommended for performance reasons. In this example 3 "interpreter processes" (see Appendix D) will be active:

Process name	*Users*
glintd 27	global1
glintd 28	global2
glintd 29	global3, global4 & global5

In this example, users *global3*, *global4* and *global5* will be competing for the resources of a single Unix process.

F.2.12 Shared Memory

For large configurations it may be necessary to increase the amount of Unix shared memory allocated by Global System Manager. For example, to increase the shared memory allocation to 512K bytes, add the following line to the Systems file:

```
SHMSIZE 512
```

F.2.13 Readability

For readability it is recommended that the Systems file layout match those in the previous examples. However the layout can be changed to suit your system administrator (see Chapter 7). It is often useful to comment certain commands within the configuration file. Comments are indicated by a # character, all text up to the next <CR> is ignored. The following example indents commands to show their association with each other:

```
# O2A diskette on left hand 3½" diskette drive.
INSDEV /dev/rfd0135ds18
```

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir

# Left hand 3½" diskette drive.
DISKETTE    0 O2 /dev/rfd0135ds18
            0 B3 /dev/rfd0135ds9

# End of details for SYSTEM A.

# Office software data file.
SYSTEM B
DATA 0 B00.dir

# End of details for SYSTEM B.

PRINTER
500 /dev/ttypc01
501 /dev/ttypc02
510 "lpr -r"

OPIDMAP
john JOHN

TERM
vt100 19
wyse50 163
vt200 181
vt300 182

SYSTEM 27 USER 1 LOGNAME root OPID ROOT
# John Smith's authorisation.
SYSTEM 28 USER 2 LOGNAME john
# Terminal in the board room.
SYSTEM 29 USER 1 TTY /dev/ttypc05
```

F.2.14 Compatibility

Global System Manager V7.0, and later, data file formats are upwards compatible with the data formats of Global System Manager V6.2 and earlier releases.

If you are upgrading from V6.2 to V7.0, or later, and your Global data is in the form of separated subunit domains (compatible with discrete data files), then the only consideration is the MAXIMUM NUMBER OF FILES specified in the configuration file for this data file. This value **MUST BE THE SAME** for both systems.

If you are upgrading from V6.2 (or earlier) and your Global data is in the form of simulated volumes then you must configure these data files as integrated data file's under Global System Manager V8.0. If the number of subunits per data file is small (the default was 29) then these data files could be added in the following way:

Global configuration file:

```
CONTROLLER (DATA ) : Discrete data file
      DRIVE ( 0)
      DESCRIPTION (Discrete data file )

VOLUME FORMAT (T224Z ) SEPARATED SUBUNIT DOMAIN
MAXIMUM NUMBER OF FILES ( 250)
NUMBER OF SUBUNITS ( 59)
      UNIT NUMBER ( 200)

VOLUME FORMAT ( )

CONTROLLER (DDATA ) : Integrated data file
      DRIVE ( 1)
```

```
      DESCRIPTION (Integrated data file )

VOLUME FORMAT (P224Z ) UNIX VARIABLE DOMAIN
MAXIMUM NUMBER OF FILES ( 99)
NUMBER OF SUBUNITS ( 39)
      UNIT NUMBER ( 260)
```

Unix Systems file:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir
DATA 1 bos200.vol
```

Your old bos200.vol will now appear as units A60 to A99. bos230.vol could then be added to SYSTEM B thus:

```
SYSTEM B
DATA 1 bos230.vol
```

which would appear as units B60 to B99. If these data files have a larger number of subunits then the data file could be defined as UNIT NUMBER (110) with 89 subunits (any diskette definitions would have to be altered to refer to unit numbers lower than 100).

This would result in the data file being accessed by Global System Manager as units a10 to a99 for SYSTEM A. Alternatively, having previously installed onto data file A00.dir, Global System Manager could be installed onto an integrated data file (by changing the BACRES configuration file appropriately). A file server SYSTEM (e.g. SYSTEM B) could then be set up to configure itself from this data file by defining a special SYSDATA for it. For example, having installed onto an integrated data file called bos200.vol (previously defined as SYSTEM A DATA 0 bos200.vol), $F PIP $MONITOR to B01. Next, modify the Systems file to specify the correct DATA 0 for SYSTEM A:

```
SYSTEM A
SYSDATA A00.dir
DATA 0 A00.dir # previously installed onto
```

then add the newly installed system as SYSTEM B:

```
SYSTEM B
SYSDATA bos200.vol # should rename to B00.dat
DATA 0 bos200.vol # ditto
```

When Global System Manager is reconfigured B00 will be an integrated data file and A00 will be a discrete data file.

Note this is a general mechanism for setting up heterogeneous file servers and must only be attempted by experienced system administrators.

Appendix G - Troubleshooting Guide

This appendix contains a list of commonly encountered problems and misunderstandings that have been reported with Global System Manager (Unix) together with the recommended solutions.

Since the inception of this trouble-shooting guide, which was produced immediately after the release of Global System Manager (Unix) V8.0, several sections have been incorporated into the main body of this manual. However, to preserve consistent section numbering between this version of Appendix G and earlier versions, the section numbers of those topics moved to other chapters in this manual have been maintained.

This appendix contains sections on the following topics:

G.1 What is the BACNAT variant number and why is it so important?

This section is a combination of sections G.1, G.45 and G.65 from previous versions of the Global System Manager (Unix) Troubleshooting Guide.

On other implementations of Global System Manager (e.g. Global System Manager (BOS) and Global System Manager (Novell)) the "nucleus" components are distributed in a collection of libraries and stand-alone files on the Global format BACRES volume.

With Global System Manager (Unix), the situation is different. The libraries (i.e. +.C0 and +.C2) and stand-alone files (e.g. %.C2D) on the Global format BACRES volume are dummy components present on the volume so that standard generation and installation jobs may be

used for **all** versions of Global System Manager. The various executable, binary files that comprise the "nucleus" are distributed on a Unix format BACNAT volume. The BACNAT volume is either a tape, a single tar format diskette or, for some obsolete configurations, a series of diskettes. Thus, in general, it is not possible to provide zaps to fix problems with the Global System Manager (Unix) nucleus. The only mechanism to distribute upgraded software is to provide a new BACNAT volume(s). The "glinstall -u" option is provided to update the "BACNAT components" (see section 6.2.2).

The packaging process that creates the BACNAT volume automatically generates a unique "version file" which contains the version numbers of all the source files compiled to produce the BACNAT components. The source files are held in a Unix Source Code Control System (SCCS) directory. Each version of the "version file" is also kept (as a SCCS file). Thus, given a BACNAT variant number TIS Software support staff can immediately identify which sources, and which versions of those sources, were compiled to produce the BACNAT components. This self-policing mechanism is less error-prone than a manual system (see also section G.102).

The "version file", **$GLDIR/sys/version**, (see section 6.1.1.4) is an ASCII text file that may be inspected with any Unix text-file inspect/edit command (e.g. cat, more, pg, vi, view, head, tail etc.). For example:

```
      # head -16 $GLDIR/sys/version

/*************************************************
/*
/* Copyright TIS Software ...
/*
/* Module ...
/*
/* Version ...
/*
/* Last change ...
/*
/* Current date ...
/*
/*************************************************
/* This version file created while packaging: SCO */
/* Sun June 28 18:14:13 BST 1992 */
--> s.vc2gm.ver Version 3.67
```

The BACNAT variant number appears in the message displayed by *global* if you are the only person using Global System Manager:

```
      Configuring Global System Manager (Bnnnn Vv.vvv); please wait...
```

where *v.vvv* is the variant number (e.g. 3.190) and *nnnn* is the configuration code for the BACNAT software (e.g. 5527 for SCO Unix). Note that early versions of the BACNAT software just displayed the variant number information:

```
      Configuring Global System Manager (BACNAT Vv.vvv); please wait...
```

Furthermore, if the BACNAT version is V3.113, or later, the **$GLDIR/sys/variant** file (see section 6.1.1.5) will be present in the $GLDIR directory. This file is used by the V8.1 version of $S to display the BACNAT variant (see section 4.6).

BACNAT variants uniquely identify a particular revision of the Global System Manager nucleus sources AND the version of Unix for which the sources were compiled. For example, if a

BACNAT repackage involves 3 versions of Unix (e.g. SCO Unix, AIX and HP-UX) **without any intervening changes to the source code** then 3 different BACNAT versions will result:

V3.200	B5522	SCO Unix
V3.201	B5539	AIX
V3.202	B5540	HP-UX

In general, the variant number of the BACNAT software is not tied to the version number of Global System Manager. For example, BACNAT version V3.88 will operate with both Global System Manager V7.0 and V8.0. Thus, every Global System Manager (Unix) system may be referred to by two different version numbers (i.e. Global System Manager version: V7.0, V8.0 etc. and BACNAT version: V3.88, V3.89 etc.). In order to prevent possible confusion when referring the "the version number", the term "variant number" is used in Global Technical Bulletins when referring to the BACNAT version number.

G.2 What is the messages file?

The "messages file", **$GLDIR/sys/messages** (see section 6.1.1.3), contains a log of all significant Global System Manager events and hence can be used for problem diagnosis. The following example messages file was obtained by running Global System Manager and exiting using $BYE:

```
root, pid 6361: Wed Oct 11 09:42:36 1995
Wed Oct 11 09:42:36 1995
root, pid 6361: Wed Oct 11 09:42:36 1995
Configuring GSM (BACNAT V3.167); please wait...
root, pid 6361: Wed Oct 11 09:42:37 1995
global: Message 1044 - Entering GSM session.
System 0x1b, pid 6363: Wed Oct 11 09:42:42 1995
glintd: Message 1226 - PROCESS TERMINATED
root, ppid 6363, pid 6371: Wed Oct 11 09:42:43 1995
glclean: Message 1610 - Terminating System(s) 0x1b.
root, pid 6361: Wed Oct 11 09:42:43 1995
global: Message 1047 - Died due to signal SIGTERM.
root, ppid 6363, pid 6371: Wed Oct 11 09:42:43 1995
Terminating processes and tidying GSM resources
```

In general, the messages file contains records of the type (see Appendix C):

```
program_name: Message nnnn - Description of message
```

The logging of events in the messages file is optional, it can be switched off using the global -m option (see section 6.3.3.3), but the performance loss is negligible because records are written to the messages file so infrequently.

Note that since the release of BACNAT V3.163, all messages in the $GLDIR/sys/messages file are preceded by the current Unix system date and time.

Records are never removed from the messages file thus it may grow rapidly. We suggest the System Administrator regularly deletes the messages file, for example, by adding an entry to the /etc/rc script (or a related script) to be executed each time Unix is bootstrapped.

G.3 GLDIR shell variable must be defined

This topic has been moved to section 8.1.1.

G.4 glinstall must be run by superuser

The glinstall installation script (see sections 2.1.2 and 6.2) can only be run by a user with superuser privileges (i.e. root).

G.5 User and group *global* required in order to install GSM

A user *global* and a group *global* must both be defined before you attempt to install Global System Manager (see section 2.1.3). The techniques used to establish new users and groups vary between versions of Unix - consult your System Administrator's Guide (or equivalent manual).

G.6 What to do if the installation device is incorrect

The name of the Global System Manager installation device is given in the Global Configuration Notes. For example, the Global Configuration Notes for SCO Unix V3.2.x distributed on 3½" diskettes (document C5527) state that the following command should be used to extract the installation script from the BACNAT media:

```
tar xvf /dev/rfd0135ds18
```

If this device name is inappropriate for a particular computer configuration then the following steps must be taken to tailor the installation procedure. This example describes the changes required in order to install Global System Manager (Unix) for SCO Unix distributed on 3½" diskettes when the 3½" diskette drive is configured as drive-1, instead of drive-0.

1. Use the correct device name (i.e. rather than the device name documented in the Configuration Notes) to extract the installation script from the BACNAT media. In this example:

   ```
   tar xvf /dev/rfd1135ds18
   ```

2. If an attempt is made to use the standard glinstall script, the following message is displayed:

   ```
   The distribution is on diskette,using device /dev/rfd0135ds18
   ```

   Either use the glinstall -d option (see section 6.2.1) to specify the name of the installation device:

   ```
   # ./glinstall -d /dev/rfd1135ds18
   ```

   or edit the glinstall script to change the line from:

   ```
   FLOPPY='/dev/rfd0135ds18'
   ```

   to:

   ```
   FLOPPY='/dev/rfd1135ds18'
   ```

   The modified glinstall script can be used to extract the remaining Unix files from the BACNAT media. The message:

   ```
   Please mount BACNAT on /dev/rfd1135ds18
   ```

   will confirm that glinstall is using the correct device.

3. When the following message appears:

```
Please mount BACRES on /dev/rfd1135ds18
```

key q to quit the installation. The Systems file must be amended to indicate to the "BACRES software" that the device name has been changed.

Define the GLDIR and PATH shell variables as described by the *glinstall* dialogue then use *gladmin* to make two modifications to the Systems file. Firstly, change the name of the installation device from:

```
INSDEV /dev/rfd0135ds18
```

to:

```
INSDEV /dev/rfd1135ds18
```

Secondly, change the device names of the default diskette devices from:

```
DISKETTE
      0     B3    /dev/rfd0135ds9
      0     O2    /dev/rfd0135ds18
```

to:

```
DISKETTE
      0     B3    /dev/rfd1135ds9
      0     O2    /dev/rfd1135ds18
```

4. Lastly, use the following command (see section 6.3.2) to complete the interrupted installation:

```
global -i
```

Note that shell scripts may be run in echo, non-suppress mode. This technique may be used to diagnose problems in the glinstall script. For example:

Bourne shell: `sh -x glinstall`

Korn shell: `ksh -x glinstall`

G.7 Problems with software installation (part-1)

The default Systems file does not include any USER's on file-server SYSTEM A. Consequently, no local diskettes (i.e. units in the range 100 - 199) or data files (i.e. units in the range 200 - 299) appear when the default system is installed. The only "numeric" direct access units are those in the range 600 - 699 which are aliased to data file units on the "master" SYSTEM (usually SYSTEM A).

The lack of local diskettes causes a problem with V7.0 $INSOFT. If the format code of the installation diskette is specified, V7.0 $INSOFT only searches local units for the corresponding unit number. For example:

```
Please key format code:G1A UNSUPPORTED FORMAT
```

If the G1A diskette is addressed as unit a50, for example, the unit number must be specified explicitly:

```
Please key format code:a50
```

The V8.0, and later, versions of $INSOFT scan units on SYSTEM A in addition to local units when searching for the specified format code. For example:

```
Please key format code or unit-id:G1A
```

As explained in section 2.5.3 some of the more mature Global applications (e.g. Global Finder) require the presence of a local data file (virtual domain). If a data file unit in the range 200 - 299 is not available the installation jobs may fail unexpectedly (usually accompanied by a message of the form "No hard disk available" or "No space on hard disk"). If the installation of a Global application fails because of the absence of a local data file, the steps outlined in section 2.5.3 should be followed.

G.8 No diskette units appear in the $U report.

The default Systems file does not include any USERS on file-server SYSTEM A. All USERS are attached to non file-server SYSTEMS (e.g. 0x1b). Consequently, by default, the only diskette units that are available appear in the range a00 - a99. There are no units in the range 100 - 199. If it is ever necessary to access diskettes as units 1*xx*, the technique described in section F.2.9 should be used. Note that whereas the V7.0 version of $U only displays details of the local SYSTEM, the V8.0, and later, versions of $U display details of all the available file-servers SYSTEM's as well as the local SYSTEM (see section 4.9).

G.9 No data definitions appear in the $U report.

The default Systems file does not include any USERS on file-server SYSTEM A. All USERS are attached to non file-server SYSTEMS (e.g. 0x1b). Consequently, by default, the only data file (virtual hard disk) units that are available appear in the range A00 - A99. There are no units in the range 200 - 299. If it is ever necessary to access data files as units 2*xx*, the technique described in section F.2.9 should be used. Note that whereas the V7.0 version of $U only displays details of the local SYSTEM, the V8.0, and later, versions of $U display details of all the available file-servers SYSTEM's as well as the local SYSTEM (see section 4.9).

G.10 BACNAT upgrades (always extract glinstall from BACNAT media)

The "glinstall -u" command (see section 6.2.2) is used to upgrade an existing Global System Manager installation. Although changes to the *glinstall* script are made very infrequently, **you are recommended to always extract the *glinstall* file from the BACNAT diskette**, using the command described in the Global Configuration Notes, before attempting the upgrade installation.

G.11 Can't re-install while Global System Manager files in-use

If an attempt is made to re-install Global System Manager (i.e. using glinstall) while Global System Manager is in-use one of the following warning messages will be displayed:

```
tar global/bin/xxxxxx - cannot create
```

or:

```
tar global/bin/xxxxxx - text file busy
```

Any attempts to use the partially re-installed system will result in unpredictable results.

To check the consistency of the files in the $GLDIR/bin directory, use the "ls -l" command. The last date of modification should be the same for all binary files extracted from a BACNAT volume (see section G.12 below).

G.12 File permissions of Global System Manager files and directories

The following Unix directory list was obtained from a freshly installed Global System Manager V8.1 system on SCO Unix using the following command:

```
# ls -lR $GLDIR

/usr/global
drwxrwxr-x  2     global      global  256     May 17 11:19  bin
drwxrwxr-x  3     global      global  48      May 17 11:19  data
drwxrwxr-x  2     global      global  48      May 17 11:19  pkg
drwxrwxr-x  2     global      global  32      May 17 11:19  spool
drwxrwxr-x  5     global      global  144     May 17 11:20  sys
drwxrwxr-x  2     global      global  80      May 17 11:38  tmp

/usr/global/bin:
-rws--s--x  4     root        global  64125   May 09 09:41  GL
-rws--S---  2     global      global  561     May 09 09:41  GLADMIN
-rws--s--x  4     root        global  64125   May 09 09:41  GLOBAL
-rws--s--x  4     root        global  64125   May 09 09:41  gl
-rws--S---  2     global      global  561     May 09 09:41  gladmin
-rws--s---  1     root        global  37660   May 09 09:41  glclean
-rws--S---  1     global      global  59104   May 09 09:41  glconfig
-rws--s---  1     root        global  319894  May 09 09:41  glintd
-rwxr-x---  1     global      global  3194    May 09 09:41  glinfo
-rws--S---  1     global      global  34545   May 09 09:41  glmkdat
-rws--s--x  4     root        global  64125   May 09 09:41  global
-rws--s---  1     root        global  24288   May 09 09:41  glprid
-rws--S---  1     global      global  87659   May 09 09:41  glreorg
-rwx--x---  1     global      global  145683  May 09 09:41  glshmdump
-rws--s---  1     global      global  6936    May 09 09:41  glspad
-rws--s---  1     root        global  45226   May 09 09:41  glspod
-rwx--x---  1     global      global  50717   May 09 09:41  glsysdump
-rws--s---  1     root        global  5672    May 09 09:41  gltimd

/usr/global/data:
drwxrwxr-x  2     global      global  96      May 17 11:38  A00.dir

/usr/global/data/A00.dir:
-rw-rw----  1     global      global  32768   May 17 11:38  SVL00_SYSDOM
-rw-rw----  1     root        global  629632  May 17 11:38  SVL01_SYSRES
-rw-rw----  1     global      global  114688  May 17 11:38  SVL02_TEST

/usr/global/pkg:
-r--r--r--  1     global      global  56      May 09 09:41  gpprod

/usr/global/spool:
<empty>

/usr/global/sys:
-rw-r--r--  1     global      global  4152    May 17 11:37  Systems
drwxrwxr-x  2     global      global  256     May 17 11:37  data
drwxrwxr-x  2     global      global  64      May 17 11:19  errmsg
-r--r--r--  1     global      global  99      May 09 09:41  machine
-rw-rw----  1     root        global  2592    May 17 11:38  messages
drwxrwxr-x  2     global      global  80      May 17 11:37  spool
-r--r--r--  1     global      global  6       May 09 09:41  variant
-r--r--r--  1     global      global  3429    May 09 09:41  version

/usr/global/sys/data:
-rw-r--r--  1     global      global  216     May 17 11:37  screen
-rw-r--r--  1     global      global  2406    May 17 11:37  shm
-rw-r--r--  1     global      global  4       May 17 11:37  shmsiz
```

```
-rw-r--r--  1     global      global  160     May 17 11:37  tmap
-rw-r--r--  1     global      global  231     May 17 11:37  tty
-rw-r--r--  1     global      global  19      May 17 11:37  umap
-rw-r--r--  1     global      global  34      May 17 11:37  user

/usr/global/sys/errmsg:
-r--r--r--  1     global      global  2457    May 09 09:41  hdr
-r--r--r--  1     global      global  7839    May 09 09:41  txt

/usr/global/sys/spool:
-rw-rw----  1     global      global  4       May 17 11:38  gl_LCK_002
-rw-rw----  1     root        global  2783    May 17 11:23  log
-rw-rw----  1     global      global  0       May 17 11:37  queue

/usr/global/tmp:
<empty>
```

The file sizes and creation dates are illustrative.

THE OWNERSHIPS AND PERMISSIONS OF FILES WITHIN THE $GLDIR DIRECTORY, OR ANY SUB-DIRECTORY, MUST NEVER BE CHANGED (E.G. USING chown, chgrp OR chmod) OTHERWISE UNPREDICTABLE RESULTS MAY OCCUR.

G.13 Copying data files between Unix computers (file permissions)

When copying Global data directories from one Unix computer to another (e.g. using *tar* or *cpio*) the ownerships of the directories and files must be updated to reflect the group and passwd files of the destination computer. Although a description of Unix file permissions and ownerships is beyond the scope of this appendix it should be remembered that the group-id for group *global* and user-id for user *global* are held in the /etc/group and etc/passwd files, respectively. Unless the entries for global in the /etc/group and /etc/passwd files on the source and destination computers match, or unless special actions are taken when performing the Unix file copying process, the files/directories will not have the correct ownerships. If a data directory (e.g. B00.dir) does not possess the correct ownership and permissions it may not be possible to access the Global System Manager data files (i.e. ASSIGNMENT ERRORS will be reported when attempting to access the virtual domain). If a data file (e.g. SVL01_SYSRES) within a data directory (e.g. B00.dir) does not possess the correct ownership and permissions it may not be possible to access the corresponding subvolume (i.e. the unit will not appear in a $F LIS of the domain). The glinstall -p option (see section 6.2.4) can be used to reset the ownerships and permissions. Alternatively, the Unix *chgrp* and *chmod* commands should be used by the super-user to reset the file/directory ownerships and permissions to the values shown in the above Unix directory listing (see section G.12).

This problem is most likely to affect data files (e.g. SVL01_SYSRES) within data directories (e.g. A00.dir) as these files are frequently copied from one computer to another. When copying Unix date files between computers ensure that the group is set to global (using the chgrp command) and that "group read and write" permissions are allowed for the file (using the chmod command). For example:

```
chgrp global SVL20_NEWVOL

chmod g+rw SVL20_NEWVOL
```

G.14 The Systems file must be re-compiled to effect changes

The Systems file is an ASCII text file which can be amended using any standard Unix editor (e.g. ed or vi). Rather than performing the time-consuming Systems file syntax validation during

every Global System Manager invocation, it is more expedient to validate the Systems file before it is used. The glconfig utility (see section 6.5) performs two functions. Firstly, it validates the syntax of the Systems file, preventing attempts to specify inconsistent SYSTEM configurations. Secondly, *glconfig* compiles the Systems file into a collection of data files in the $GLDIR/sys/data directory (see section 6.1.1.6). When Global System Manager is invoked the data files, rather than the Systems text file, are used to build the SYSTEM configuration. Thus, the Systems file must be re-compiled before any changes become effective. The *gladmin* script (see section 6.4) runs the vi editor on the Systems file and automatically invokes *glconfig* if any amendments were made to the text file.

G.15 Systems file DATA, DISKETTE, USER numbers start from 0 & 1

The DATA and DISKETTE numbers in the Systems file (that correspond to DRIVE NUMBER's in the configuration file - see sections 7.3.2 and 7.3.3) are numbered from 0. However, the USER numbers in the Systems file (that correspond to "User numbers" in the configuration file - see section 7.3.1) are numbered from 1.

G.16 When to make changes to the Global Configuration file

The Systems file is intimately linked with the Global configuration file (see Chapters 7 and 9). Because every SYSTEM, file-server and non-file-server alike, share the same Configuration file, the simple addition of extra SYSTEM's to the Systems file does not require changes be made to the Global configuration file. However, the introduction of additional DATA numbers, DISKETTE numbers, DISKETTE volume-types, USER numbers and PRINTER numbers does require corresponding changes to the Global configuration file (see Chapter 9).

G.17 Errors when attempting to load large systems

If the amount of shared memory is insufficient for a particular systems configuration (see section 7.4.4) the following fatal error message is usually displayed:

```
$99 ERROR M LOADING ++nnnnxx
global: Error 1018 - Fileserver x has failed to configure: system terminating
```

However, if the addition of extra users to the Systems file causes the Shared Memory limit to be exceeded it may not be possible for Global System Manager to fail so gracefully. If the simple addition of extra SYSTEM's to the Systems file results in crashes or unexpected results (e.g. the inability to access all data files on all file server SYSTEM's), immediately suspect the Shared Memory size. The size of Shared Memory may be specified using the SHMSIZE command word in the Systems file (see section 7.4.4).

G.18 $STATUS limitations

This topic has been moved to section 4.7.

G.19 glclean is not $BYE

Before $BYE terminates a Global System Manager session it checks that no other users are running Global software on your SYSTEM (e.g. $BYE checks that no files are left open). However, glclean, performs no checking of this sort (see section 6.7) and should only be used as a last resort to remove or restart a user.

G.20 Spare space reported by $F

The amount of spare space displayed by the $F (or $V) list of a T151Z domain directory is slightly misleading. The spare space does not refer to the amount of space reserved for the particular domain, but rather, it refers to the spare space available on the Unix filing system. Thus, if two Discrete Data directories (e.g. A00.dir and B00.dir) are present on the same filing system, a $F list of both will display the same amount of spare space. Don't be misled into thinking that each domain can be increased in size by the amount displayed!

G.21 Capacity reported by $U

The data file CAPACITY displayed by $U (see section 4.9) is slightly misleading. The CAPACITY does not refer to the amount of size of the particular domain, but rather, it refers to the total capacity of the Unix filing system. Thus, if two Discrete Data directories (e.g. A00.dir and B00.dir) are present on the same filing system, a $U list of both will display the same capacity. Don't be misled into thinking that each domain can be increased in size to the amount displayed!

G.22 Spurious messages appear on the master console when diskettes are being accessed

Unix system messages are displayed directly on the system console. On most versions of Unix there is no way to switch off this feature. Error messages from device drivers fall into the category of Unix system messages and as such will appear on the system console whenever an error with the hardware or media occurs. The most likely hardware/media error to occur on a reliable computer is a read error on a faulty diskette.

The following example dialogue was observed on SCO Unix when an attempt was made to read an un-formatted diskette:

```
$66 INPUT DEVICE:a96
$66 OUTPUT DEVICE:<CR>
$66 FILE MAINTENANCE
:LIS
WARNING error on floppy (2/36),blk=18 cmd=0003,status= 0002
WARNING error on floppy (2/36),blk=18 cmd=0003,status= 0002
WARNING error on floppy (2/36),blk=18 cmd=0003,status= 0002
WARNING error on floppy (2/36),blk=18 cmd=0003,status= 0002

* READ ERROR ON Diskette drive 0 - a96
* RETRY?:
```

The Unix system messages are not stored in Global System Manager's screen image and are removed if <SYSREQ> 0 is used to refresh the screen.

G.23 Using $BYE when spoolers are in-use

$BYE always ensures that no other users are active on the target SYSTEM. In addition, $BYE also checks that no files are currently in-use if the target SYSTEM is a file-server. However, if an attempt is made to invoke $BYE while another partition (or partitions) on the SYSTEM are suspended within the Global System Manager spooler (i.e. $SP), with $PR assigned to a unit on another SYSTEM, a problem will occur. $BYE will not detect the "shared" $$SP file on the remote SYSTEM and will exit Global System Manager leaving the $$SP file open. Subsequent attempts to use the spooler will result in a "FILE IN-USE" error until the SYSTEM that hosts the spooler unit is reloaded.

Note that this general problem is not restricted to the Global System Manager spooler ($SP).

G.24 Sharing Global System Manager printers with other Unix applications

Printers may be shared between Global System Manager and other Unix applications (see Appendix E). If a Unix application prints to a Global System Manager "direct" printer the print mode, paper type etc. may be altered. Obviously, Global System Manager will not be aware of the change in the state of the printer and will continue to assume the printer status immediately after the last Global System Manager print. In order to reset the printer for subsequent use by Global System Manager use the $P command or the I instruction within the Global System Manager spooler ($SP or $SPS). $P, $SP and $SPS are all documented in the Global System Manager Manual.

G.25 Printer Buffer length must be set to 250

In order to use the escape sequences defined in a Printer Control File (see Chapter 6 of the Global System Manager Manual) with Global System Manager V8.0, the printer buffers must be 250 bytes in length. Some early versions of Global System Manager V7.0 were distributed with configuration files that included printer buffers of only 132 bytes. $CUS (see Chapter 6 of the Global System Manager Manual) can be used to increase the size of the printer buffers. Note that the dependency on the printer buffer length has been removed for Global System Manager V8.1.

G.26 Global System Manager printers don't alter serial line characteristics

Global System Manager does not attempt to alter the serial line characteristics (e.g. baud-rate, parity, handshaking) for either serial screens or, more importantly, serial printers. The direct printer handler merely writes characters to the Unix device. Any serial line characteristics must be correctly initialised (e.g. using *stty*), either directly or via a start-up script, before Global System Manager writes to the device. Future versions of Global System Manager **may** include an option to initialise the serial line characteristics for direct printers.

G.27 Persistent NOT READY errors on direct printers

The direct printer interface creates a temporary FIFO, gl_FIFO_5*xx*, in the $GLDIR/tmp directory. When allocated, this FIFO (aka "named pipe") special file has the following permissions and ownerships:

```
prw-------  global      global      gl_FIFO_500
```

If it not possible to create the FIFO, because a real file with the same name already exists in the $GLDIR/tmp directory, for example, all attempts to use the corresponding printer will fail with a NOT READY error. If this error occurs, delete the offending file from the $GLDIR/tmp directory.

G.28 Common problems with spooled printers

The Spooled Printer handling is one of the most difficult areas of Global System Manager. Several problem areas must be considered.

As explained in Appendix E the spooled printer interface involves the creation of a temporary file in the Global spool directory (e.g. $GLDIR/spool). The filename includes a sequence

number (between 1 and 99). If an attempt is made to create a file with a sequence number greater than 99 a DIRECTORY FULL error will be reported. Note that a NOT READY error was reported by early variants of the BACNAT software. The sequence number limit only applies to the members of a series of files of the type *aaaaaaaa.c.nn*, so that, for example, it is possible to create 99 files of the form D.F01.1B.*nn*, 99 files of the form D.S.1B.*nn*, 99 files of the form D.F01.1C.*nn* etc. Notwithstanding, it is strongly recommended that the Unix spooler command (or script) deletes the file after printing (see Appendix E).

The spooled printer handling includes several potentially error-prone stages (e.g. the temporary file must be created on the spool directory; the Unix spooler command or script must be invoked passing it the name of the temporary file). If any problems occur (usually manifested by the spool files not appearing on the physical printer) it is difficult to determine the exact cause of the failure. Future versions of Global System Manager **may** include diagnostic code in the spooled printer handler (e.g. writing records to the log file when errors are returned from the Unix spooler command/script).

In the meantime, in order to investigate problems with the spooled printer interface, we recommend modifying the Systems file to replace the Unix spooler command by a "null" command (e.g. include the line: PRINTER 510 "true"). Any attempts to print to unit 510, in this example, should result in the creation of a temporary file in the spool directory, but no further action will be taken. If the temporary file is created successfully, the Unix spooler command (e.g. gllp510) can be run directly from the shell prompt. This technique will normally isolate problems with file permissions etc. (see below).

Early versions of the BACNAT software (i.e. incorporating the V3.1 or V3.2 versions of vc2eb.c - you can check the version of this source file by examining the $GLDIR/sys/version file) created the temporary spool files with the following permissions: -rw-r--r--. Creating files that are "readable" by others exposed a security loophole (i.e. sensitive print reports created by Global System Manager could be inspected by other, non-global users). The V3.3 version of vc2eb.c (which is incorporated into all BACNAT variants 3.35, and later) creates temporary spool files with the following permissions: -rw-rw----. This change closes the security loophole and, for most versions of Unix, introduces no undesirable side effects because the Unix spooler command (e.g. lp) is usually simply executable. If however, the Unix "suid" bit in the permissions of the Unix spooler command is set, the command will not have sufficient permissions to read the temporary file.

Two solutions are available. A "sledge-hammer" technique that is not recommended is to change the permissions of the Unix spooler command to remove the "suid" bit. For example:

```
chmod u-s lp
```

The preferred solution is to change the permissions of the temporary file from within the Unix spooler script. For example, include the chmod command at the start of the script:

```
chmod o+r $1
lp -c -ddevice $1
rm $1
```

Alternatively, this script could be replaced the following command in the Systems file (note the use of the $FILE variable - see section 7.3.6.2):

```
"chmod o+r $FILE;lp -c -ddevice $FILE; rm $FILE"
```

Note that early versions of Global System Manager (Unix) didn't require the presence of the $GLDIR/sys/spool directory. Consequently, this directory was not created by early versions of the glinstall installation script. The printer handling now requires the presence of the $GLDIR/sys/spool directory otherwise no printers will be available (e.g. the $U report will not include details of any printers). The $GLDIR/sys/spool directory is created by versions of the BACNAT software V3.35, or later - see section G.1. **Thus, it is vital to use the current version of the glinstall installation script (extracted from the BACNAT volume) when using the "glinstall -u" command to upgrade the BACNAT software (see section G.10 above).**

The following table describes the symptoms that will occur if either of the /spool directories are missing or have incorrect permissions:

$GLDIR/sys/spool	Spooled printers are dynamically removed and do not appear in the $U report;
$GLDIR/spool	Attempts to print to spooled printers result in persistent DIRECTORY FULL errors.

Note that there is an error on page 5-2 of the V8.0 Global Operating Manual (Unix), which has been corrected in this manual: The important **global/sys/spool** directory is not shown on the global directory structure.

G.29 Mixing direct and spooled printers

When extra printers are added to the Systems file, great care must be taken to ensure that direct printers and spooled printers are not mixed (see sections 7.3.6.1 and 7.3.6.2) otherwise unexpected results will occur when printing is attempted. For example, on a system that included the following printers:

Unit	*Legal?*	*Config' file*	*Systems file*
500	Yes	DIRECT	/dev/tty1a
501	Yes	DIRECT	/dev/tty2a
502	No	DIRECT	"cat > /dev/null"
503	Yes	DIRECT	/dev/null
504	Yes	DIRECT	/dev/null
510	Yes	SPOOLED	"gllp510"
511	Yes	SPOOLED	"gllp511"
512	No	SPOOLED	/dev/null
513	Yes	SPOOLED	"cat > /dev/tty1a"
514	Yes	SPOOLED	"cp $FILE /dev/tty1a"

The following results were obtained when attempting to use the "illegal printers":

502	* ASSIGNMENT ERROR ON 502 (i.e. unit 502 does not appear in the $U report)
512	Printing appeared to work! (i.e. a temporary file was written to the $GLDIR/spool directory)

G.30 After using Global System Manager the function keys are un-usable by other Unix applications

Most Global System Manager TAP's re-program the function keys for subsequent use by Global applications. By default, when $BYE, $E or <SYSREQ> . are used to return to a Unix shell the function keys are not reset. A special System Request, <SYSREQ> S, is distributed with Global System Manager V8.0 (Unix) which allows you to define "Screen Reset" sequences (see section 5.3). We have investigated integrating the Global System Manager screen and keyboard handling more closely with the information held in the Unix terminfo database (or similar databases) but this approach is fraught with difficulties. Future versions of Global System Manager (Unix) **may** include alternative solutions to the "function key" problem.

G.31 Attempts to use separately installed Global System Managers will fail

Because of the Unix hierarchical directory structure it is possible to install two, or more, Global System Manager systems on a single Unix computer. However, it is not possible to use the separately installed Global System Managers simultaneously. Attempts to do so will produce unpredictable results. For an update, see section G.53.

G.32 Inter-Process Communication facilities

Global System Manager makes extensive use of the Inter-Process Communication (IPC) facilities as defined in the System-V Interface Definition (SVID). On some versions of Unix that are based on BSD Unix (e.g. SunOS) the IPC facilities are optional. The IPC facilities **must** be installed in order to use Global System Manager.

When installing SunOS 4.1, if the option to install System V files and libraries is not taken then the IPC facilities will not be added to the kernel. To determine whether the facilities are installed, examine the kernel configuration file. If the following lines:

```
options     IPCMESSAGE
options     IPCSEMAPHORE
options     IPCSHMEM
```

are absent or commented out the System V IPC facilities are not installed. The kernel must be rebuilt with these facilities included.

If you are unsure whether a particular Unix system includes the Interprocess Communication Facilities, the **ipcs** command will return a definitive result. For example:

```
# ipcs
IPC status from /dev/kmem as of Fri Oct 13 16:11:17 1995
T     ID    KEY         MODE        OWNER       GROUP
Message queues:
q     3000  0x47534d32  -Rrw-rw---- root        global
Shared memory:
M     5900  0x47534d30  --rw-rw---- root        global
Semaphores:
s     590   0x47534d33  --ra-ra---- root        global
s     591   0x47534d34  --ra-ra---- root        global
```

G.33 <SYSREQ> . fails after displaying a brief message

The error message:

```
global: Error 1025 - Error returned from new shell
```

is displayed briefly when a <SYSREQ> . is attempted if it is not possible to create a new shell (or the command specified by $SHELL) or if the new shell exited with a non zero status. For example, setting the shell variable:

```
SHELL=false
```

will always result in this error message appearing. See section 5.4 for further details.

G.34 Incorrect system time on SCO Unix

There have been reports of a problem with the date/time handling on SCO Unix. The Global System Manager system date/time (i.e. the information displayed by $D) does not agree with the Unix system date/time (i.e. the information displayed by the date command). The Global System Manager date may appear to be 14 hours ahead of the Unix date. This problem is caused by the format of the TZ shell variable (defined in the /etc/TIMEZONE script file by /etc/timezone). When the problem occurred the TZ variable was defined as follows:

```
TZ=GMT00BST,W13,W39
```

A more correct value for the TZ variable is:

```
TZ=GMT0BST;M3.5.0/02:00:00M10.5.0/02:00:00
```

The /etc/TIMEZONE script should be amended to include the line:

```
TZ="GMT0BST;M3.5.0/02:00:00M10.5.0/02:00:00"
```

Note that the string must be quoted because it includes the ";" character which would be interpreted by the shell.

After amending /etc/TIMEZONE, log back in to Unix. You will have to reset the Unix system date using the date command.

Refer to SCO documentation for further information.

G.35 Global System Manager changes umask value

While Global System Manager is executing the umask value is set to 007. When either $E or $BYE are used to terminate a Global System Manager session the original umask value is restored. However, if <SYSREQ> . is used to create a new shell, the umask value remains at 007.

G.36 Long volume descriptions are persistent

All the 50 character Long Volume Descriptions for a data directory are held in the SVL00_*dddddd* domain file rather than being associated with individual data files (e.g. SVL*nn_vvvvvv*). If a data file is copied to a data directory using a Unix command (e.g. cp, cpio or tar) it will automatically inherit the Long Volume Description associated with the corresponding sub-volume number.

G.37 Non-alphanumeric characters in Unix data file names ($F CHA)

As explained in section 4.10.11, Global System Manager subvolumes are mapped to Unix files as follows:

Description	*Format*	*Example*
GSM unit:	A*nn*	A10
GSM volume name:	*vvvvvv*	SLDATA
Unix filename:	SVL*nn_vvvvvv*	SVL10_SLDATA

The $F CHA instruction effectively performs the same file-level processing as the Unix mv command. For example, to change the volume name of SLDATA on unit A10 to ABCDEF would require the following Unix file rename:

```
mv SVL10_SLDATA SVL10_ABCDEF
```

Of course Global System Manager does not use the *mv* command directly but it does issue the same Unix system calls (link and unlink) that are used by the *mv* command. Thus, the $F CHA instruction is subject to the same Unix file naming restrictions as the mv command, so that, for example, it is not possible to include the "/" character in the volume name. Attempts to use this special character (e.g. to rename a volume to AB/DEF) are equivalent to the following Unix command:

```
mv SVL10_SLDATA SVL10_AB/DEF
```

The "/" character is interpreted as a directory indicator and in the extremely likely case that a directory $GLDIR/A00.dir/AB does not exist the following error will appear:

```
rename: No such file or directory
```

If an attempt to rename a subvolume involves an illegal Unix filename, Global System Manager interprets the error returned by the link system call as a write error which results in the following message appearing:

```
:CHA AAAAAA TO:AB/DEF
* WRITE ERROR ON Discrete data file - A10
* RETRY?:
```

G.38 global command line options

This topic has been moved to section 6.3.3.

G.39 Software licensing problems after installing from tape

The following problem may occur when attempting to run Menu Maintenance, MN, on Global System Manager V8.0 configurations installed from tape:

```
SORRY - YOU SEEM TO BE IN DANGER OF INFRINGING
        YOUR BOS SOFTWARE LICENSE.

PLEASE DO NOT TRY TO RUN THE SOFTWARE
UNDER A MORE POWERFUL LEVEL OF BOS OPERATING SYSTEM
THAN THE ONE FOR WHICH IT WAS SUPPLIED.
```

This problem, which has been fixed for Global System Manager V8.1, was caused by a bug in the job that installs Global System Manager from the BACRES (661) and BEA (662) volumes. For Global System Manager V8.0, the following work-around solution is available:

1. Use $BYE to exit Global System Manager;

2. From the Unix shell, key the following command to reload the starter system (from unit 261):

```
global -i
```

   Key <CR> to the introductory prompts.

3. Key "Q" to the following prompt:

```
[A.8] Overwrite existing Global System on 201 (Y):Q
```

4. Key <ESCAPE> to the following prompt to obtain the "GSM READY" prompt:

```
[A.14] Specify address of system unit on master system (A01):<ESC>
```

5. Run the SERIAL program "by hand" using the following dialogue:

```
GSM READY:SERIAL<CTRL A>
PLEASE ASSIGN $P:262
SOFTWARE SERIALISED AS nnnnnnn
```

6. Use $F to copy files MN and P.MN from 262 to 201, overwriting the existing versions.

7. Run $BYE from 201 to exit the starter system:

```
GSM READY:$BYE<CTRL A>
PLEASE ASSIGN $P:201
```

G.40 All systems use the SYSDATA defined for SYSTEM A

All SYSTEM's use the SYSDATA defined for SYSTEM A, and therefore do not explicitly define SYSDATA. **LOADING FROM MULTIPLE SYSDATA's IS A HIGHLY SPECIALISED TECHNIQUE WHICH IS NOT RECOMMENDED BECAUSE IT CAN EASILY RESULT IN INCONSISTENT CONFIGURATIONS.** For example, in addition to merely copying a SYSRES volume file, it is also necessary to use the $F utility to INStall a bootstrap and to PAM and PIP the $MONITOR file (i.e. specialised techniques that are beyond the scope of this manual). This technique complements the SYSTEM-specific configuration mechanism described in section G.41. For example to load Global System Manager for SYSTEM 0x22 from the bootstrap sub-volume on Discrete Data File B00.dir:

```
SYSTEM      0x22
SYSDATA B00.dir
USER 1 LOGNAME aju OPID AJU
```

G.41 SYSTEM-specific configuration files (GLCONFIG*x* Variable)

By default, all SYSTEM's use the same configuration file from the shared SYSDATA unit (e.g. ++5522AA on unit A01). Under exceptional circumstances it may be necessary to use a special configuration file for a particular file-server system. This may be achieved by defining the shell

variable GLCONFIGx (where *x* is the system identifier "A" to "Z") to be the 8-character file-name of the special configuration file. For example:

```
GLCONFIGX=++552XAA
```

Note that the first 6 characters of the system-specific configuration filename (e.g. ++552XAA) differ from the first 6 characters of the generic configuration filename (e.g. ++5522AA). This renaming convention (i.e. ensuring that the first 6 characters of the filenames are different) is vital in order to prevent the system-specific configuration from being loaded in preference to the generic configuration file if the former happens to occupy a lower slot in the SYSRES directory (if a GLCONFIGx variable is not defined for a SYSTEM, the generic configuration file, ++*NNNN*?? (where ? are wildcard characters), loaded).

If a SYSTEM-specific configuration file has been defined, the following warning message is displayed when Global System Manager is loaded:

```
glintd: Warning 1227 - Using config variable GLCONFIGX, config file ++552XAA
```

The SYSTEM-specific configuration file should only differ from the standard configuration file in the DATA FILE DEFINITIONS section (see section 9.2.1) otherwise inconsistent configurations may result. SYSTEM-specific configuration files may be required for the following, albeit slightly artificial, reasons:

1. To allow a mixture of volumes with 250 files/directory and 99 files/directory. For example:

SYSTEM A	A00:	A01 - A99	250 files/directory
SYSTEM X	X00:	X01 - X97	99 files/directory

2. To allow a mixture of discrete and integrated data volumes. For example:

SYSTEM A	A00:	A01 - A99	Format T151Z
SYSTEM X	X00:	X01 - X99	Format P224Z

3. To allow a particular SYSTEM(s) to access more than 1 data volume. For example:

SYSTEM A	DATA 0	A00:	A01 - A99
SYSTEM X	DATA 0	X00:	X01 - X49
	DATA 1	X50:	X51 - X99

SYSTEM-specific configuration files should only be used in exceptional circumstances. This advanced technique may not be available in future versions of Global System Manager (Unix).

G.42 Data file directories do not have to be in $GLDIR/data

Although the recommended default directory for all data file directories (e.g. A00.dir, B00.dir etc.) is $GLDIR/data (see section 6.1.1.6), this default may be over-ridden. For example, to allow access to other installed Global System Manager data files, a Systems file may contain entries of the form:

```
SYSTEM A
#
```

```
# Data files for SYSTEM A
#
DATA 0 A00.dir
#
SYSTEM B
#
# Data files for SYSTEM B
#
DATA 0 B00.dir
#
SYSTEM C
#
# Data files for SYSTEM C allow access to the data from
# another installed Global System Manager.
#
DATA 0 /usr/aju/gsm70/global/data/A00.dir
#
SYSTEM D
#
# Data files for SYSTEM D allow access to the data from
# yet another installed Global System Manager.
#
DATA 0 /usr/aju/test/global/data/A00.dir
```

Important Note: The file permissions and ownerships of extra data file directories *x*00.dir (e.g. D00.dir), and the .SVL files within those directories, must be the same as for the A00.dir directory (see section G.12). If the file permissions and ownerships of extra data file directories, and the .SVL files within those directories are **not** the same as for the A00.dir directory then problems will be encountered when accessing sub-volumes in the corresponding domain (e.g. D00).

G.43 Problems with multiple SVL00_*XXXXXX* files in an SSD directory

When a Discrete Data file directory (e.g. A00.dir) is created by *glmkdat* (see section 6.6) a 32Kb "domain header" file, SVL00_DOMAIN, is created in that directory. The "domain header" file is renamed to SVL00_*xxxxxx* (where *xxxxxx* is the domain volume name e.g. SYSDOM) when $V is used to initialise the domain.

If a Unix utility (e.g. cp, cpio, tar) is subsequently used to copy subvolume files to an SSD directory be extremely careful not to copy the SVL00_*xxxxxx* "header file". If an SSD directory contains more than one file with a filename of SVL00_* (where * is the Unix wildcard character), Global System Manager will not recognise the Separated Subunit Domain and, if the domain contains the SYSRES directory, fail to load.

See also sections G.61 and G.62.

G.44 Problems moving the global directory after installing

In addition to the potential "file permissions" problems described in sections G.12 and G.13, another problem can occur if the $GLDIR directory is moved (using Unix utilities) immediately after installing Global System Manager. During the Global System Manager installation a number of empty directories are created. Some Unix backup utilities (e.g. SCO tar) do not save/restore empty directories. For example, after extracting directories and files from the BACNAT media, the glinstall script uses the Unix *mkdir* command to explicitly create several additional, empty directories. If the $GLDIR directory (and all sub-directories) are moved from one computer/filing-system to another immediately after installing Global System Manager, it is necessary to recreate the empty directories.

The two empty directories that may not move when using Unix backup/restore utilities (e.g. tar) are $GLDIR/tmp (see section 6.1.5) and $GLDIR/spool (see section 6.1.4). If the $GLDIR/tmp directory is absent, the following message will appear when an attempt is made to run Global System Manager:

```
global: Error 1006 - Can't create lock file
```

If the $GLDIR/spool directory is absent, the following message will appear when an attempt is made to print to a SPOOLED printer:

```
* DIRECTORY FULL ON Spooled printer output - 510
* RETRY?:
```

Section G.12 describes the correct file permissions for these empty directories and Appendix D describes the temporary files that reside in these directories when Global System Manager is running.

G.45 Why do BACNAT versions differ so much?

This topic has been moved to section G.1.

G.46 Move heavily used diskette drives to separate systems

The standard, distributed Systems file contains both DISKETTE and DATA definitions for SYSTEM A (see section F.1). Because of the poor diskette performance under most versions of Unix, diskette access will degrade the performance of those users who are accessing data files on SYSTEM A while diskette operations are in progress. On those configurations that involve regular diskette access, we strongly recommend creating a separate SYSTEM definition(s) for DISKETTE units. For example:

```
SYSTEM A DATA A00.dir # SYSTEM A for hard-disk data files
SYSTEM D DISKETTE # SYSTEM D for diskette drive-0
0 B3 /dev/rfd0c
0 O2 /dev/rfd0h
SYSTEM E DISKETTE # SYSTEM E for diskette drive-1
1 G1 /dev/rfd1h
```

However, see section G.59 for potential problems that may occur if this technique is used.

G.47 Systems file and $STATUS USER numbers are different

A USER number in the Systems file is merely the key to a particular "console channel" in the configuration file (see section 9.3). The USER number in the Systems file must match a "User number" in the Global configuration file. The Systems file/configuration file "User" numbers are better described as "Console" numbers and bear no relationship to the Global System Manager USER numbers displayed by $STATUS, for example. See also, section G.54.

G.48 Using "glinstall -u" overwrites Systems file

This topic has been moved to section 6.2.2.

G.49 glinstall command line options

This topic has been moved to section 6.2.5.

G.50 glmkdat command line options

This topic has been moved to section 6.6.1.

G.51 More shell variables: GLTT and GLTERM

This topic has been moved to section 8.2.

G.52 Diskette image files on hard disk

An obscure technique that may be of general use involves creating a Unix file that is the image of a Global System Manager diskette and then subsequently accessing that "diskette image" using Global System Manager (Unix).

For example, use the Unix cp command to take an image backup of a O2A diskette:

```
cp /dev/rfd0135ds18 /usr/global/data/o2a_image
```

then modify the Systems file to allow the "fixed diskette image" to be accessed by Global System Manager:

```
SYSTEM X
DISKETTE    0 O2 /usr/global/data/o2a_image
```

Obviously, this technique is of limited use if the Unix computer includes a real diskette drive. However, see section G.90 for an enhancement to this technique.

G.53 Use of the GLIPCBASE variable for separate Systems

For Systems with a BACNAT version of V3.84 (or later), it is possible to load two, or more, separate Global System Manager "systems" on the same Unix computer (section G.31 describes the situation for pre V3.84 BACNAT's). **THIS TECHNIQUE SHOULD ONLY BE USED FOR TEST/EVALUATION SYSTEMS AND NEVER ON A "LIVE" SITE.**

Central to the implementation of Global System Manager (Unix) is the use of the Unix Interprocess Communication facilities (see section G.32). Prior to BACNAT V3.84, the IPC base key was hard-coded as "GSM0". If an attempt was made to run two separately installed Global System Managers, a clash of IPC facilities would result causing the second invocation of Global System Manager to fail with the following message:

```
global: Error 1029 - User already running Global System Manager
```

BACNAT V3.84, and later, allows the default IPC base to be overridden by the value of the optional GLIPCBASE shell variable. In order to run several independent Global System Managers concurrently on the same computer the GLIPCBASE variable should be set to a different 4 character string for each independent system. For example:

```
GLIPCBASE=AJU0;export GLIPCBASE
```

THIS TECHNIQUE SHOULD BE USED WITH GREAT CARE. In particular, the separate Global System Managers should never be allowed to access the same data files.

G.54 A discussion of $STATUS and Systems User Numbers

In a standard Global System Manager configuration, each SYSTEM contains a single USER definition. Consequently, when Global System Manager is loaded each user runs with a unique System number (e.g. $$LNID = 0x1b, 0x1c etc.) but with partition P1 always $STATUS USER

1, P2 always $STATUS USER 2 etc. By default, it is never possible to arrange for each user to run with a unique $STATUS USER number. The following technique explains how this slightly perverse arrangement may be achieved.

In order to arrange for unique $STATUS USER numbers, the USER DISPLAY ATTRIBUTES section of the configuration file must be amended to include several USER definitions. In the example that follows, this configuration file is assumed:

User number	*NUMBER OF VIRTUAL PARTITIONS*
1	4
2	3
3	2
4	1

It is not merely sufficient to modify the Global configuration file as above and change the Systems file as follows:

```
SYSTEM 0x1b USER 1 LOGNAME test1 OPID TST1
SYSTEM 0x1c USER 2 LOGNAME test2 OPID TST2
SYSTEM 0x1d USER 3 LOGNAME test3 OPID TST3
SYSTEM 0x1e USER 4 LOGNAME test4 OPID TST4
```

In this example, user TST1 will be running Global System Manager as SYSTEM 0x1b, USER 1 (as expected) but user TST2 will also be running as USER 1 (SYSTEM 0x1c), rather than, as naively predicted, USER 5. This unexpected result occurs because the Global System Manager start-up code includes logic to optimise the allocation of Unix memory resources by removing "un-usable" entries from the configuration data. In this example, the optimisation will result in the removal of USERS 2,3,4 from SYSTEM 0x1b (because these USERS do not appear in the Systems file for this SYSTEM) and the removal of USERS 1,3,4 from SYSTEM 0x1c (because these USERS do not appear in the Systems file for this SYSTEM). This optimisation code has the effect of re-basing all $STATUS USER numbers from 1.

To thwart the optimisation code, dummy USER's must be added to the Systems file. For example:

```
SYSTEM 0x1b USER 1 LOGNAME test1 OPID TST1
SYSTEM 0x1c
USER 1 LOGNAME dummy1c1
USER 2 LOGNAME test2 OPID TST2
SYSTEM 0x1d
USER 1 LOGNAME dummy1d1
USER 2 LOGNAME dummy1d2
USER 3 LOGNAME test3 OPID TST3
SYSTEM 0x1e
USER 1 LOGNAME dummy1e1
USER 2 LOGNAME dummy1e2
USER 3 LOGNAME dummy1e3
USER 4 LOGNAME test4 OPID TST4
```

In this revised example the following results will be obtained:

User name	*System number*	*User number*
TST1	0x1b	1
TST2	0x1c	5

TST3	0x1d	8
TST4	0x1e	10

By judicious manipulation of configuration file and Systems file entries it is possible to build large, separate process Systems with unique $STATUS USER numbers. However, this technique does involve by-passing the aforementioned optimisation code which results in Global System Manager making **extremely heavy** demands on the Unix memory resources. Consequently, **THIS TECHNIQUE IS NOT RECOMMENDED.**

G.55 Complications when a USER is defined for SYSTEM A

The default Systems file does not include a USER on SYSTEM A. Consequently, SYSTEM A behaves as a normal file server system: When global is invoked by the first Global System Manager user, no processes are started for the file server systems. The file server definitions in the Systems file just cause shared memory to be allocated for the relevant File Executive control blocks. When the last Global System Manager uses $BYE to quit a Global System Manager session, the following familiar message is displayed and all "global" processes are killed:

```
Terminating processes and tidying Global System Manager resources
```

If however, a USER is defined for SYSTEM A, when global is invoked by the first Global System Manager user, a glintd process is started for SYSTEM A and appears in the process table as "glintd 1". This process will be left running and shared memory will remain allocated, even after all other Global System Manager users have quit to provide the **potential** for a user to attach to SYSTEM A. This feature can sometimes lead to unexpected, although strictly accurate, "Global System Manager in use" messages when attempting to use glconfig, for example.

The "Terminating process..." message will only appear, and shared memory will only be de-allocated, if the last user to quit Global System Manager is the one who is attached to SYSTEM A.

G.56 More Problems DIRECT Printers (part 1) - Large Prints

Printing very large reports using the DIRECT printer interface may result in seemingly persistent NOT READY errors part-way through the print. This situation can be alleviated by using $CUS, or Global Configurator, to increase the Printer Timeout period (e.g. from 20 to 40 seconds). Alternatively, setting the timeout period to zero will cure the problem but this solution is not recommended because any permanent errors on the printer will result in a "hang". This problem is currently unavoidable because of the relatively large time taken to flush the large Unix buffers.

The generalised DIRECT printer interface:

```
PRINTER 500 /dev/ttynnn
```

may be emulated by the following SPOOLED printer interface, which will not suffer from the "buffer flush" problem:

```
PRINTER 510 "cat $FILE > /dev/ttynnn;rm $FILE"
```

However, see sections G.92 and G.93 for better solutions to this problem.

G.57 Display buffering introduces message latency

Because most versions of Unix are not truly real-time operating systems, a latency can occur on displays that are issued immediately before an access to a slow-device. This can result in slightly unusual displays when accessing diskettes or tape devices. The effect is most striking when verifying a diskette (the track number increments in a jerky fashion) or when running $TAPE (some messages appear fleetingly). There is no simple solution to this problem that will not introduce a severe performance degradation.

The display buffering also results in another effect that can occur when a Global System Manager "interrupt key" (e.g. <CTRL U> or <CTRL G>) is used. Because the "non-flushable" Unix display buffers may contain the last several hundred characters displayed by Global System Manager they will continue to be displayed even after a <CTRL U> or <CTRL G> keystroke has been used to prevent any further Global System Manager displays.

G.58 Modifying the Console "Direct Display" flag

This topic has been moved to section 9.3.1.

G.59 Diskettes on SYSTEM A required for installing

Section G.46 describes a technique which involves moving DISKETTE definitions to separate SYSTEM's. This technique is recommended on those computers where diskettes are frequently accessed.

However, if the technique described in section G.46 is used, it must be remembered that the Global System Manager installation process (i.e. global -i) requires, for a diskette installation, both a DISKETTE definition (to access the BACRES, BEA, HAA etc. diskettes) and a DATA definition on SYSTEM A (to access the hard disk); or, for a tape installation, both a DATA definition for drive 1 on SYSTEM A (to access the BACRES, BEA, HAA etc. volumes) and a DATA definition for drive 0 on SYSTEM A (to access the hard disk).

When the global "-i" option is specified, global ignores the Systems file OPID mapping for root and always executes as SYSTEM A (in order to access 'local' devices e.g. 140 and 200; or 260 and 200) - see section 6.3.2.

If an attempt is made to run "global -i" without a DISKETTE defined for SYSTEM A in the Systems file the following messages will appear:

```
# global -i
Configuring Global System Manager (BACNAT V3.nn); please wait...

Terminating processes and tidying Global System Manager resources
global: Error 1036 - Timed out on process synchronisation.
Inform your System Administrator.
```

Therefore, before any changes affecting the DISKETTE definition for SYSTEM A are made to the Systems file, a copy of the original file (with DISKETTE and DATA definitions for SYSTEM A) should be kept. This backup must be restored (and recompiled using glconfig) in order to reinstall Global System Manager using the "global -i" command.

G.60 Persistent error 1050 when loading Global System Manager

When a user attempts to login to Global System Manager (i.e. by keying global) a connection (using Unix signals, semaphores and shared memory) is established between the *global* foreground process and the *glintd* background process. During the initialisation of the *global*-to-*glintd* link, the glintd process attempts to re-open the standard Unix file channels associated with the stdin and stdout file descriptors (using the Unix fropen system call).

The file descriptor re-open is necessary to prevent problems from occurring if a user attempts to login to Global System Manager after logging out of Unix while leaving a background glintd process executing. The following fatal error message is displayed if the Unix *fropen* system call fails (which should never happen under normal circumstances):

```
global: Error 1050 - glintd can't open tty device.
```

If this error occurs a background, "zombie" *glintd* process will be left running (use the Unix ps command to confirm this).

To avoid the error 1050, run *glclean* to kill the zombie *glintd* process (if necessary) then use the "-r" option of global (see section 6.3.3.6) to overcome any problems with the fropen system call.

G.61 Re-installing Global System Manager on top of an installed system

To upgrade the BACNAT software without affecting the Global System Manager BACRES, BEA software, the "-u" option of *glinstall* must be used (see section 6.2.2). To upgrade the Global System Manager BACRES, BEA software without affecting the BACNAT software, the "-i" option of *global* must be used (see section 6.3.2). **However, the problem described in sections G.43 and G.62 (i.e. multiple SVL00_xxxxxx files in an SSD directory) will occur if an attempt is made to re-install both the BACNAT and the Global System Manager software in a single operation**.

Running *glinstall* (i.e. without the "-u", or related, option) will result in the creation of a SVL00_DOMAIN file in the A00.dir directory. For an installed system, this directory will already contain a file with a name of the form SVL00_*xxxxxx* (e.g. SVL00_SYSDOM). When *glinstall* invokes "global -i" to commence the second phase of the two-part installation, the problems described in section G.43 and G.62 (including Global System Manager crashing with SIGSEGV errors if the BACNAT variant is earlier than V3.90) may occur.

If this problem occurs, simply use rm to delete the SVL00_DOMAIN file from the A00.dir directory. For example:

```
# cd $GLDIR/data/A00.dir
# rm SVL00_DOMAIN
```

It is also important to note that the accidental use of *glinstall* (i.e. without the "-u", or related, option) will also overwrite the existing Systems file ($GLDIR/sys/Systems) WITHOUT creating a backup copy. If this occurs, the Systems file must be restored from your Unix backup copy.

G.62 Various problems with Discrete Data Files

Sections G.43 and G.61 touch on some of the problems that may occur with Discrete Data Volumes (aka Separated Subunit Domains). This section provides a more comprehensive list

of these problems. Note that sections G.12 and G.13 describe further problems that may occur if the file-permissions and ownerships are incorrect.

This section should be read in conjunction with section 4.10.11.

The SSD directory (e.g. A00.dir) MUST contain only one "header file" with the name SVL00_* (where * is the Unix shell wildcard character). Any SSD directory that contains multiple header-files is considered invalid by Global System Manager. If an attempt is made to run global, using BACNAT software prior to V3.90, with an "invalid directory" defined in the Systems file, the results will be unpredictable (normally the domain will be ignored, although system crashes may also occur!). For BACNAT versions V3.90, or later, if an "invalid directory" of this type is detected the domain will be ignored and the following warning message displayed:

```
glintd: Warning 1228 - Domain contains multiple SVL00_xxxxxx files
```

This message will be followed by the full pathname of the unexpected file. For example:

```
/usr/aju/global/data/A00.dir/SVL00_DOMAIN
```

Similarly, the SSD directory (e.g. A00.dir) MUST contain only one "subvolume file" for each numeric subvolume. For example, only one SVL01_* file, only one SVL02_* file etc. (where * is the Unix shell wildcard character). Any SSD directory that contains multiple subvolume-files is considered invalid by Global System Manager. If an attempt is made to run global, using BACNAT software prior to V3.90, with an "invalid directory" defined in the Systems file, the results will be unpredictable (normally the subvolume will be ignored, although system crashes may also occur!). For BACNAT versions V3.90, or later, if an "invalid directory" of this type is detected all but the first SVL*nn_xxxxxx* file will be ignored and the following warning message displayed:

```
glintd: Warning 1230 - Domain contains multiple SVLnn_xxxxxx files
```

This message will be followed by the full pathname of the unexpected file(s). For example:

```
/usr/aju/global/data/A00.dir/SVL01_COPRES
```

Finally, the size of every file in the SSD directory (e.g. A00.dir) MUST be an exact multiple of the Global System Manager virtual track size (i.e. 8Kb for formats T224Z and T151Z). The domain "header file" created by *glmkdat* (see section 6.6) is always 32Kb. When a "subvolume" file is created by $V, the size specified by the operator is always rounded up to be a multiple of 8Kb. Any files with an SSD directory that are not multiples of the track size are considered invalid. If an attempt is made to run global, using BACNAT software prior to V3.90, with an "invalid file" within an SSD directory defined in the Systems file, the file will be ignored without any warning message (the subvolume will not appear in a $F or $V list of the domain). For BACNAT versions V3.90, or later, if an "invalid file" of this type is detected it will be ignored and the following warning message displayed:

```
glintd: Warning 1229 - Unix file not a multiple of Global System Manager track size
```

This message will be followed by the full pathname of the file with the incorrect size.

```
/usr/aju/global/data/A00.dir/SVL02_BADSIZ
```

If the above message appears, the erroneous file should be restored from a Unix backup copy. If the filesize of the version on the backup copy is also incorrect, the *glreorg* utility (see section 6.10) or Unix *dd* command may be used, AS A LAST RESORT, to create a copy with a valid size. For example on AIX V3.2.4, use the following Unix command:

```
dd ifs=SVL02_BADSIZ ofs=SVL03_GOODSZ ibs=8k conv=sync
```

The Global data within the recovered subvolume (i.e. SVL03_GOODSZ in this example) should be treated with deep suspicion as the unexpected change in filesize may be symptomatic of more insidious data corruption.

G.63 Files per subvolume and Discrete Data Files

When the maximum number of files per directory was increased from 99 to 250 (with the introduction of Global System Manager V6.2) checking was included in the File Executive to ensure that the maximum number of files per subvolume on the initialised domain(s) matches the value in the configuration file. If the value on disk differs from that in the configuration data, the File Executive will return an error "V" (WRONG VOLUME FORMAT) when an attempt is made to access the domain. This checking is necessary to prevent possible data corruption caused by the inconsistent directory sizes (the size of the domain directory is a function of the maximum number of subvolumes).

The checking that applies to real domains (e.g. P88Z) and integrated data volumes (e.g. P224Z) has always been included in Global System Manager. The equivalent checking for discrete data volumes (e.g. T224Z and T151Z) has only been included in the 2.37 vc0a.c module (check your $GLDIR/sys/version file). If a directory size mismatch is detected by the new checking algorithm an error "v" (i.e. NOT error "V") will be returned.

It should not normally be necessary to amend the DATA controller section of the Global configuration file (see section 9.2.1) because extra discrete data volumes (SSD's) should be added by amending the Systems file. **If the DATA controller section of the configuration file is changed be extremely careful to ensure that the number of files per subvolume in the configuration file agrees with the number of files per subvolume on the existing discrete data volumes.**

For historical reasons, the default number of files per subvolume for volume format T224Z is 99. However, all distributed Global System Manager (Unix) V7.0 and V8.0 configuration files include a single T224Z format domain with 250 files per subvolume. This potential problem has been removed for Global System Manager V8.1. All V8.1 configuration files include volume format T151Z, which is isometric with volume format T224Z. The default number of files per directory for volume format T151Z is 250.

G.64 $TAPE and Global System Manager (Unix)

$TAPE V7.0A, and later, includes support for Global System Manager (Unix) computers (see Global Technical Bulletin GT571, 13-Nov-1992). This section describes some special factors that must be considered when using $TAPE on Global System Manager (Unix).

For Global System Manager (BOS) computers, the "$TAPE software" consists of three components:

$TAPE	$TAPE Cobol front-end;

TAPE*nn* Various Cobol overlays in P.$TAPE library;

*am*CT*nn* Assembler tape controllers (interface to hardware)
am = Machine-code (e.g. J5 for the IBM PC)
nn = Tape controller number (e.g. 03 = AHA SCSI)

For Global System Manager (Unix) computers, the "$TAPE software" consists of four components:

$TAPE $TAPE Cobol front-end (as above);

TAPE*nn* Various Cobol overlays in P.$TAPE library (as above);

C2CT01 Dummy tape controller;

glintd Real tape controller (uses Unix system calls) within the *glintd* module. Note that *glintd* is distributed on the BACNAT diskette/tape.

$TAPE loads the relevant tape controller onto the Global Cobol User Stack within the Cobol User Area. On Global System Manager (BOS) it is perfectly permissible to execute assembler code loaded within a Cobol User Area (i.e. on the Cobol stack). However, on Global System Manager (Unix) the Unix kernel treats the Cobol User Area as a "data segment" so that it is not possible for $TAPE to invoke a C tape controller on the Cobol User Stack. To circumvent this problem, the dummy C2CT01 module supplied with $TAPE V7.0A contains a Cobol ESCAPE instruction followed by a data area that redirects the C version of the Commercial Code Interpreter (CCI) to the real tape controller within *glintd* (via a jump-table, also in *glintd*). This technique is used for the other C "assembler assist modules" (e.g. %.C2D, BW999C).

If the entry-point to the real tape controller is missing from the jump-table the CCI will return a "STOP with 772". This will be displayed as:

```
$91 TERMINATED - STOP CODE 772
```

The first version of the BACNAT software that includes a real tape controller is V3.84 (see GT571). Consequently, if an attempt is made to use $TAPE V7.0A with BACNAT software prior to V3.84, the above STOP CODE will result. Unfortunately, due to a "feature" in the C version of the CCI, this STOP CODE will be persistent. The only method available to free up a partition stuck with a STOP CODE 772 is to use *glclean* to clean the particular SYSTEM (see section 6.7).

Note that versions of $TAPE earlier than V7.0A (i.e. V7.0, V6.0A and V6.0) did not include the C2CT01 module in P.$TAPE so will fail with the following error message if run on a Global System Manager (Unix) configuration:

```
ERROR LOADING CONTROLLER C2CT01,
```

When $TAPE is used with Global System Manager (BOS), a tape controller entry must be present in the configuration file (e.g. ADAPTEC, AHA-TAPE). This entry creates a TA-block within the configuration data which is used to pass configuration parameters (e.g. port address) and $TAPE operations to the tape controller. When $TAPE is used with Global System Manager (Unix) the TA-block, which does not contain any configuration parameters, is part of the DATA DIVISION of $TAPE itself. Consequently, an entry within the TAPE CONTROLLER

section of the configuration file is not strictly necessary for Global System Manager (Unix) configurations. Older Global System Manager (Unix) configuration files (pre 9-Dec-92) will not contain a TAPE controller. More recent configuration files (9-Dec-92, and later) will include a dummy TAPE controller.

A problem has been reported in V7.0A $TAPE (packaging date 1/10/92) which results in a PGM CHK-11 AT 8C06 when an attempt is made to save or restore a tape from any SYSTEM other than "A", 0x1b, 0x1c, 0x1d, 0x1e or 0x1f. The problem can be avoided by running $TAPE from a System lower than 0x20 (NB. System 0x20 == System 32).

The following Unix specific sections have been extracted from the V7.0 Global Tape Utility ($TAPE) Manual.

G.64.1 GLTAPE shell variable

The GLTAPE shell variable specifies the device name which the $TAPE controller will use. It should be added to the user's .profile script (or wherever else the GLDIR and PATH variables are set) as follows:

```
GLTAPE=device_name;export GLTAPE
```

If this shell variable is not established then an error D (missing device error) will be reported when attempting to run $TAPE. Note that this name must specify the device which executes a rewind when the device is closed. For example, if you are using a SCSI tape drive on SCO Unix then you should use the /dev/rStp0 device rather than /dev/nrStp0.

G.64.2 $TAPE must not be run by more than one user

Currently, there is no way to lock the tape drive when it is running under Global System Manager. However, if GLTAPE is set up for use by only one user then there will never be a conflict because all other users will obtain an error D when attempting to use $TAPE (as described above).

G.64.3 Options screen settings

Since Global System Manager (Unix) comprises a collection of separate systems which emulate a network of computers $TAPE will always attempt to use a staging unit. This is unnecessary since all the files reside on a single machine so the "Use staging unit" option should be set to "N". The only advantage in using a staging unit is to minimise the size of the units being saved.

G.64.4 Tape must be in drive

The $TAPE controller determines whether the device specified by the GLTAPE shell variable is valid by attempting to open it. If a tape is not present in the drive then the Unix open operation will fail, which is reported as an error D by $TAPE.

G.64.5 Read/write privileges on tape drives

Devices are usually owned by the superuser but Global System Manager does not normally have superuser privileges. For this reason please ensure that all Global users have read/write privileges on the selected tape device. For most versions of Unix, the required privileges are set up by default. For example, the SCSI tape device on SCO Unix has the following privileges:

```
# ls -l /dev/rStp0
crw-rw-rw-  3     root  other 46, 0 Oct 30 1992 rStp0
```

G.64.6 $TAPE status displays

$TAPE displays various helpful messages during save, restore, etc. However, these displays are buffered by the Unix drivers so may not appear in full before the $TAPE controller takes exclusive control of the process, in which case the screen may appear to hang until the outstanding tape operation completes.

G.64.6 Data transfer between Unix and non-Unix configurations

Tapes produced by $TAPE running on Global System Manager (BOS) are incompatible with those produced by $TAPE on Global System Manager (Unix). All tapes produced by BOS $TAPE controllers (e.g. J5CT03) contain file marks to separate header and data blocks. The incompatibility comes about because the Unix $TAPE controller is unable to write file marks. **The absence of file marks on tapes produced on Unix systems means that such tapes cannot be read by the standard BOS controllers.**

However, it is possible to read tapes produced by Global System Manager (BOS) systems on Global System Manager (Unix) systems by using one of the following methods:

G.64.6.1 Using the standard Unix tape device

Certain versions of Unix allow the $TAPE controller to read BOS tapes (i.e. file marks can be skipped). This represents the simplest option since there is no need to change the GLTAPE shell variable to specify a different device (see section G.64.6.2). Please refer to the Global Configuration Notes to determine whether this facility is appropriate for your particular version of Unix.

G.64.6.2 Using the "no rewind" Unix tape device

If the standard Unix tape device will not skip file marks then you will need to use the equivalent "no rewind" tape device. The "no rewind" device is usually identified by the addition of an "n" to the standard device name. For example on SCO Unix:

/dev/rStp0	(standard tape device)
/dev/nrStp0	("no rewind" tape device)

Note that this is only a Unix naming convention, the *major* and *minor* device numbers associated with the device name actually specify the true device characteristics. Please refer to your System Administrator's Guide for the meaning of the major and minor device numbers. For example, the standard and "no rewind" devices for SCSI tape drives on SCO Unix are listed as follows:

```
# ls -l /dev/rStp0
crw-rw-rw- 3 root other 46, 0 Oct 30 1992 rStp0

# ls -l /dev/nrStp0
crw-rw-rw- 3 root other 46, 8 Oct 30 1992 nrStp0
```

Note how the minor device number differs, 0 for the standard device and 8 for the "no rewind" device. The "no rewind" device does not rewind the tape when the tape device is closed, so $TAPE can use a special technique to skip file marks.

IMPORTANT NOTE: The "no rewind" device must only be used when transferring data from a Global System Manager (BOS) computer to a Global System Manager (Unix)

computer. In particular, save operations will not complete correctly if a Unix "no rewind" device is used.

Using the "no-rewind" tape device can introduce some side-effects because the tape can't be rewound. For example, if a restore is attempted immediately after a list, the operation will fail with a READ ERROR or "Invalid tape header" message because the media cannot be rewound back to the tape header. To execute a tape rewind on some drives it is necessary to remove and re-insert the tape in the drive to initiate a "tape load".

G.64.6.3 Preparing a Unix style tape with a special BOS controller

The two techniques described above should enable $TAPE on most versions of Unix to read tapes created when running $TAPE on Global System Manager (BOS). If neither of the above techniques work a special BOS controller (for use with Adaptec SCSI tape drives only) must be used to produce a tape with no file marks. Use the following procedure to achieve this:

1. Run $TAPE and exit immediately, which will convert the $TDUMP style TA block into a $TAPE style TA block.

2a. For versions of $TAPE before the V7.0C release (see Global Technical Bulletin GT765, 4-July-1995), use TACUS to modify the controller code as follows:

```
Restore TA block, Customise parameters (C):<CTRL A>
Controller code: 6 (For QIC tape drives)
                 7 (For Exabyte tape drives)
TAFLAG value:<CR>
SCSI i.d. (0 - 7):<CR>
```

2b. For versions of $TAPE V7.0C, and later, use TACUS to modify the controller code as follows:

```
Restore TA block, Customise parameters (C):<CTRL A>
SCSI i.d. (0 - 7):<CR>
No filemark mode required?Y
```

3, Run $TAPE to save data.

4. Restore the original controller code using TACUS, as described above, or re-boot the machine. IMPORTANT NOTE: This final stage must be performed otherwise normal backup tapes may be inadvertently created using the special "transfer controller".

The special BOS "transfer controller" will also read tapes produced by Global System Manager (Unix) configurations. Thus it is possible to use $TAPE as a mechanism to transfer data from Global System Manager (Unix) to Global System Manager (BOS) configurations.

G.65 What is the BACNAT variant number mentioned in Global Technical Bulletins

The information formerly in this section has been moved to section G.1

G.66 More problems with DIRECT printers (part 2)

In addition to the problem that may occur with direct printers described in section G.27, another potential problem has been recently brought to our attention.

If the owner of the *glprid* printer daemon is not root (see section G.12) then unpredictable results may occur when attempting to use a direct printer. For example, if *glprid* is chown'ed to user "other" and global run by a non-root user then direct printing will fail with either a "NOT READY" or "ERROR H" message appearing on the screen. In addition to this explicit warning message, the *glprid* 5*xx* process (displayed by the Unix ps command) will be killed and thus will not appear in the process status report. All further attempts to use the direct printer will also fail.

It should be stressed that this problem will only occur if the ownership or permissions of glprid are changed, contravening the advice given in section G.12.

For further problems with the DIRECT printer interface, see sections G.92, G.93 and G.94.

G.67 Changes to Printer Control File naming convention

The Printer Control File naming convention differs between Global System Manager (Unix) V7.0 and Global System Manager (Unix) V8.0, and later.

For Global System Manager V7.0, Printer Control Files are named $RP*pppss* (where *ppp* is the printer unit (e.g. 500) and *ss* is the System identifier (e.g. "B" for System B, "1B" for System 27)). The V7.0 naming convention is perfectly adequate for Global System Manager (BOS) and Global System Manager (MS-DOS, Windows and Novell) but problems occur with Global System Manager (Unix) because the printers in a Global System Manager (Unix) implementation are shared by all the Systems.

In order for the Printer Control File to be recognised when a System (i.e. user) uses a printer, multiple copies of the Printer Control File for a particular printer must be created on the SYSRES unit. The Printer Control Files must be suitably renamed for every System that will send "direct prints" (i.e. not via the Global System Manager spooler) to that printer. For example, multiples copies of the Printer Control File for printer unit 500 must be named as $RP500A, $RP5001B, $RP5001C etc.

However, this "solution" is normally unworkable because changes to a Printer Control File for one particular System are not recording in the Printer Control Files (for the same printer) for all other Systems. The best "work-around" solution for V7.0 is to always print via the Global System Manager spooler and allocate a dedicated System (e.g. System "S") to control all the printer units, printing from the Global System Manager spooler(s). Thus, the Printer Control Files are named $RP500S, $RP501S, etc.

For Global System Manager V8.0 (BOS), the Printer Control Files naming convention is the same as that for V7.0 (see above). This is acceptable because, in a true BOS network configuration, only the local computer can print to a particular printer. However, for Global System Manager V8.0 (Unix), the Printer Control Files are named $RP*ppp* (where *ppp* is the printer unit (e.g. 500)) - the System identifier has been dropped from the file name. This name reflects the nature of the Global System Manager (Unix) printer handling - any SYSTEM can access any printer.

Please refer to section 4.2.5 for a discussion of the problems that may occur if printers are shared between Global System Manager and other Unix applications. Also, refer to Chapter 6 of the Global System Manager Manual for a full description of the V8.1 Printer Control File naming convention. The main difference between the V8.0 and V8.1 Printer Control File

naming conventions is that the V8.0 files are named $RP*ppp*, whereas the V8.1 files are named $$P*ppp*.

G.68 Insufficient Print Buffers in configuration files

All V7.0/V8.0 Global System Manager (Unix) configuration files dated before 9-Dec-92 include a minor flaw: Although the standard configuration files include 10 printers (i.e. 5 DIRECT printers and 5 SPOOLED printers) only 6 Printer Buffers are defined. If Printer Control Files are being used, an insufficient number of Printer Buffers may result in an ERROR Z when attempting to print. The number of Printer Buffers can be increased using $CUS (or CFUPDATE). As with all modifications to the configuration file, Global System Manager must be reloaded for the change to become effective.

G.69 Direct SCSI hard-disk access (for SCO Unix only)

This section describes a technique that allows the direct access of a "raw" BOS P246Z format disk from Global System Manager (Unix). This technique will only work with limited combinations of BOS disks and Unix computers. The BOS disk must be a P246Z domain (i.e. a SCSI disk on an IBM PC accessed by the "ADAPTEC" hard disk controller). **This technique will not work with P88Z disks.** The bad-track and alternate track-tables must be empty (i.e. $V options 7 (Allocate alternate track) and 8 (Amend Domain error map) must not be used on the source BOS disk). The "perfect disk" restriction may be removed in a future implementation, although this is not expected to be a serious limitation.

The Unix computer must include a SCSI bus onto which an extra hard disk can be attached. In the test implementation, the Unix computer was a 386-based PC running SCO Unix with a single disk on the SCSI bus at SCSI-ID 0.

The following steps should be taken to complete the data transfer:

1. Run the SCSISZ disk utility (from the Porting-Kit P.PK4 library) to write the size of the disk (DVBYTE and DAFIL) to a reserved sector. This utility writes a marker text-string followed by the two disk size parameters to logical sector 31 (counting from 1) on the disk. The BOS ADAPTEC controller uses the SCSI "Read Capacity" command to determine the size of the disk - sending commands directly on the SCSI interface is impossible for a Global System Manager (Unix) controller.

2. Physically connect the BOS hard disk to the Unix computer, setting the SCSI ID to 1.

3. Use the Unix *mkdev* command to add a 2nd disk to the Unix kernel. Use the following command:

   ```
   mkdev hd
   ```

   Follow the instructions to add another SCSI disk, using Adaptec host adapter 0; target-id 1; LUN 0. Rebuild the kernel, the kernel environment and reboot Unix.

   DO NOT ATTEMPT TO USE mkdev AGAIN ON THIS DEVICE. IF *mkdev* IS USED TO PARTITION THE EXTRA DISK, ALL DATA WILL BE DESTROYED!!!

4. Use the Unix *mknod* command to make a raw device driver for the 2nd disk. Use the following command:

```
mknod /dev/rhd10 c 1 64
```

5. To check that the reconfigured Unix system is accessing the new disk correctly, compare the details displayed by the following disk dumps:

   BOS:
   ```
   GSM READY:$L
   $63 INPUT FILE NAME:<CTRL B> UNIT:200
   $63 LISTING OPTION:33
   $63 LISTING OPTION:<CR>
   ```

   Unix:
   ```
   hd -s 16*k -n 512 /dev/rhd10 | more
   ```

6. Using gladmin/glconfig, add a new SYSTEM (e.g. X) to the Global System Manager Systems file:

   ```
   SYSTEM X DISKETTE 2 P246Z /dev/rhd10
   ```

7. Use Global Configurator to create a special Global System Manager Unix configuration file using the existing configuration file as the template. For example:

++5522AA	Standard configuration file
++5522XA	Special configuration file for System-X

   Change only the DIRECT ACCESS CONTROLLER SECTION, replacing the DISKETTE and DISK definitions with a single DISKETTE definition:

   ```
   CONTROLLER:             DISKETTE
   DRIVE:                  2
   DESCRIPTION:            BOS raw hard disk
   VOLUME FORMAT:          P246Z
   MAX. NO. OF FILES:      Same as BOS disk (e.g. 250)
   NUMBER OF SUBUNITS:     Same as BOS disk (e.g. 99)
   ```

8. Define the following shell variable:

   ```
   GLCONFIGX=++5522XA;export GLCONFIGX
   ```

9. Reload Global System Manager (as user *root* or with the **global -r** option), the BOS disk should be accessible as device X00. The following warning message should be displayed:

   ```
   glintd: Warning 1227 - Using config variable GLCONFIGX, config file ++5522XA
   ```

   The following message may also be displayed on the Unix system console:

   ```
   %disk - - - type=S ha=0 id=1 lun=0
   Sdisk - - cyls=nnnn hds=64 secs=32
   ```

G.70 Serial console XON/XOFF hand-shaking

By default, all console XON/XOFF hand-shaking is performed by the Unix device driver. Global explicitly enables XON/XOFF when a Global System Manager session is started on a terminal. The Console Executive included on BACNAT's up to, and including V3.90, does not contain any code to recognise the ^S and ^Q control characters. This simplistic approach leads to two **potential** problems. Firstly, if Global System Manager is configured in a non-standard fashion (i.e. when two or more screens are attached to a single SYSTEM), $STATUS (which calls the

Console Executive to determine whether a ^S character has been received by Global System Manager) will never report the "WAITING FOR <CTRL Q>" condition on a line because the ^S is intercepted by the Unix device driver. Note that $STATUS does not report on the state of the serial line for "remote" SYSTEM's. Secondly, some Global System Manager File Transfer utilities (e.g. the FTRAN utility distributed with PCWS (see section G.73)) expect to be able to temporarily disable the ^Q/^S hand-shaking on the line by switching the Console Executive into a special "transparent mode". This switch to "transparent mode" fails if the Unix tty device driver is performing the XON/XOFF hand-shaking. Section G.73 describes how this problem by can avoiding within the PC Workstation product. Note that this problem does not occur with the $FCOMM File Transfer utility, distributed with the Comms Support Pack, which uses a more sophisticated datagram structure.

BACNAT versions V3.91, and later, include an option to disable the XON/XOFF hand-shaking in the Unix device driver thus allowing the Console Executive to handle ^S/^Q characters. The fundamental XON/XOFF flag recognised by the Console Executive is included in the Terminal Attribute Program (TAP) - the flag is set to "XON/XOFF enabled" for most serial screens. By default, the version of global distributed on BACNAT V3.91, and later, continues to explicitly enable the XON/XOFF hand-shaking in the Unix device driver. However, global now recognises a new command line option to disable XON/XOFF hand-shaking in the Unix tty device driver:

```
global -x
```

Important note: The global –x option is **not** supported by BACNAT software version V3.313, or later.

G.71 $REMOTE and Global System Manager (Unix)

This section supplements the description of $REMOTE in section 4.4 and describes some special factors that must be considered when using $REMOTE on Global System Manager (Unix).

Note that $REMOTE is not available on every Global System Manager (Unix) platform although the dummy %.C2R module (see below) is distributed on all Global System Manager (Unix) BACRES volumes. In particular, $REMOTE has only been tested on SCO Unix.

For Global System Manager (BOS) computers, the "$REMOTE software" consists of three components:

$REMOTE	$REMOTE Cobol front-end and customization;
$REMEXEC	Relocatable $REMOTE executive intercept module;
%.*am*R	Assembler $REMOTE controller (interface to hardware) *am* = Machine-code (e.g. J5 for the IBM PC).

For Global System Manager (Unix) computers, the "$REMOTE software" consists of four components:

$REMOTE	$REMOTE Cobol front-end and customization (as above);

$REMEXEC Relocatable $REMOTE executive intercept module (as above);

%.C2R Dummy $REMOTE controller;

glintd Real $REMOTE controller (uses Unix system calls) within the *glintd* module. Note that *glintd* is distributed on the BACNAT diskette/tape.

$REMOTE loads the relevant $REMOTE controller onto the Global Cobol User Stack within the Cobol User Area. On Global System Manager (BOS) it is perfectly permissible to execute assembler code loaded within a Cobol User Area (i.e. on the Cobol stack). However, on Global System Manager (Unix) the Unix kernel treats the Cobol User Area as a "data segment" so that it is not possible for $REMOTE to invoke a C $REMOTE controller on the Cobol User Stack. To circumvent this problem, the dummy %.C2R module supplied with Global System Manager V7.0 and V8.0 contains a Cobol ESCAPE instruction followed by a data area that redirects the C version of the Commercial Code Interpreter (CCI) to the real $REMOTE controller within *glintd* (via a jump-table, also in *glintd*). This technique is used for the other C "assembler assist modules" (e.g. %.C2D, BW999C).

If the entry-point to the real $REMOTE controller is missing from the jump-table the CCI will return a "STOP with 772". This will be displayed as either:

```
$91 TERMINATED - STOP 772
```

or:

```
SORRY - YOU ARE ATTEMPTING TO USE A SERVICE THAT IS NOT PART OF
        YOUR OPERATING SYSTEM.
```

The first version of the BACNAT software that includes a real $REMOTE controller is V3.98. Consequently, if an attempt is made to use $REMOTE with BACNAT software prior to V3.98, the above STOP CODE will result. Unfortunately, due to a "feature" in the C version of the CCI, this STOP CODE will be persistent. The only method available to free up a partition stuck with a STOP CODE 772 is to use *glclean* to clean the particular SYSTEM (see section 6.7).

Because of the special nature of the Unix $REMOTE controller (e.g. it does not access the serial port directly but uses Unix read/write system calls to perform all i/o) several factors, described in the following sections, must be remembered when using $REMOTE on a Global System Manager (Unix) computer.

G.71.1 $REMOTE run-time options

As documented in section 4.4.2, a number of options are available which modify the way in which $REMOTE operates. Not all of the standard options are necessary when $REMOTE is used on a Global System Manager (Unix) computer:

G.71.1.1 The $REMOTE H option

The H option is used to specify the name of the assembler driver (*sic*) which is used to control the serial port during a $REMOTE session. The name of the dummy $REMOTE driver for all Global System Manager (Unix) computers is %.C2R. Thus, the necessary option would be keyed as H=%.C2R. **You must not change this option**.

G.71.1.2 The $REMOTE D option

The D option is normally used to specify the device address to be employed by $REMOTE. Since the port address for $REMOTE on a Global System Manager (Unix) computer is specified by a shell variable (see section G.71.2), this option is ignored.

G.71.1.3 The $REMOTE B option

Use the B option to specify the baud rate. The baud rate can be specified in decimal (normally), hexadecimal or octal. The *ioctl* Unix system call is used to change the baud rate on the selected serial port.

G.71.1.4 The $REMOTE T option

The T option is used to specify a time-out for communication error conditions. On Global System Manager (BOS) computers a time-out value of zero allows $REMOTE to calculate its own time-out. Because of the large amount of buffering employed by Unix it is not, in general, possible for the $REMOTE handler to calculate a time-out value. Consequently, the T option should always be set to a non-zero value when $REMOTE is used on a Global System Manager (Unix) computer (if a value of 0 is specified the default value of 250 is used). The time-out is not specified in seconds, but in terms of executions of an internal processing loop (see section G.71.3).

The B, D, H and T options are customized into the $REMOTE program file so are 'remembered' for the next invocation of $REMOTE. In addition to the four visible options, $REMOTE also allows another two controller options to be supplied by the user. These special options (Y and Z, see section G.71.3) are not displayed by $REMOTE but are nevertheless customized into the program file and passed to the controller. Note that the Y and Z options are not normally required when $REMOTE is used on a Global System Manager (BOS) computer.

G.71.2 $REMOTE shell variable

As explained in section G.71.1 the $REMOTE serial port is not specified using the D option. Instead, the GLREMOTE shell variable specifies the name of the serial device which will be used by the $REMOTE controller. The GLREMOTE shell variable must be added to the user's .profile script (or wherever else the GLDIR and PATH variables are set) as follows:

```
GLREMOTE=device_name;export GLREMOTE
```

For example:

```
GLREMOTE=/dev/tty1a;export GLREMOTE
```

If this shell variable is not established then the following message will be reported when attempting to run $REMOTE:

```
$91 TERMINATED - EXIT 400
```

G.71.3 $REMOTE timing parameters

The additional character buffering imposed by the Unix kernel causes severe problems for the simplistic $REMOTE protocol. Consequently, relatively large time-out values are required at both ends of the link - **a great deal of experimentation is required to obtain the optimum values**.

In the tests that were performed between a 20Mhz 80836 computer running Global System Manager (SCO Unix) and a 16MHz 80386 computer running Global System Manager (BOS) the following time-out values were used:

Global System Manager (BOS) computer: T=8000

Global System Manager (Unix) computer: T=400

In addition the normal "T" time-out option, the Unix $REMOTE handler recognises two additional customizable options:

Y The Y option specifies the time (in arbitrary processor loops) to wait for an ENQ character (0x05) before the start of a packet transmission. The default value is 250.

Z The Z option specifies the time (in arbitrary processor loops) to wait for the ACK character (0x06) reply after a complete packet has been sent to the remote computer. The default value is 250.

The "Y" and "Z" options should only be modified by experienced users of $REMOTE.

G.71.4 $REMOTE master and slave options

It is anticipated that $REMOTE will only be used on a Global System Manager (Unix) computer to transfer data to/from a Global System Manager (BOS) computer with an incompatible diskette or tape drive. Consequently, $REMOTE (Unix) has only been tested using the following configuration:

Master	Global System Manager (SCO Unix) computer
Slave	Global System Manager (BOS) computer

If a Global System Manager (Unix) computer is used as the "slave computer" of a $REMOTE link, two special factors must be taken into consideration. Firstly, the unit number mapping performed by $REMOTE/$REMEXEC does not allow for LAN (or pseudo-LAN) addresses:

800 - 899	mapped to "remote" 100 - 199
900 - 999	mapped to "remote" 200 - 299

Since the default Systems file does not include users with local direct access devices there will be no devices with unit numbers in the range 100 - 299 that can be mapped to 800 - 999 units! This problem can be avoided by adding a user to SYSTEM A (see section F.2.9) to allow remote access, using $REMOTE, to the Integrated Data and diskette units on a Global System Manager (Unix) computer.

The $REMEXEC relocatable intercept routine performs some of the functions of the nucleus File Executive and performs the unit mapping described above. The V7.0 and V8.0 versions of $REMEXEC do not recognise the special File Executive operations that are required in order to access Discrete Data Volumes (i.e. Separated Subunit Domains e.g. format T151Z). If an attempt is made to access a Separated Subunit Domain via $REMOTE the following error message may occur:

```
* ERROR O ON Discrete Data Files - 900
```

G.71.5 $REMOTE use of the serial port

The serial port selected for use by $REMOTE using the GLREMOTE variable (see section G.71.2) must be "disabled". For example, use the following command to disable the serial port before loading Global System Manager:

```
disable $GLREMOTE
```

The $REMOTE controller uses the isatty Unix system call to ensure that the selected device is suitable for serial data transfer. When the $REMOTE controller is loaded the *ioctl* Unix system call is used to initialise the serial port to:

8 data bits
1 stop bit
No parity
Baud-rate selected by the B option (or 9600 if B option is invalid)

When the $REMOTE controller is unloaded the *ioctl* Unix system call is used to reset the serial port to its original settings.

G.71.6 $REMOTE diagnostic shell variable

In addition to the mandatory GLREMOTE shell variable (see section G.71.2), the Unix $REMOTE handler also recognises another shell variable that may be "set" to enable detailed diagnostics. The name of the optional, diagnostic shell variable is GSM_VC2RA. To enable diagnostics define the GSM_VC2RA variable as follows:

```
GSM_VC2RA="YES";export GSM_VC2RA
```

All the diagnostics displayed by the $REMOTE handler when the GSM_VC2RA variable is defined are sent to the /dev/stdout device.

The diagnostic information is very detailed and should normally only be enabled to determine the cause of the open error that may result in the following message immediately the "M" option is specified:

```
$91 TERMINATED - EXIT 400
```

If diagnostics is enabled when $REMOTE is used to access a remote computer several character strings, indicating the progress through the packet transmission/reception routines, are displayed every few seconds.

G.71.7 $REMOTE and non-superusers

Serial devices that are not terminal devices are normally set with the following permissions:

```
crw-------  owner=bin   group=terminal
```

Therefore non-superusers do not have read/write permissions for the serial device and therefore are unable to open the serial channel. Consequently, it is not normally possible for any user other than *root* to use $REMOTE on Global System Manager (Unix). If diagnostics are enabled (see section G.71.6) the following message will accompany the "EXIT 400" that results when a non-root user attempts to run $REMOTE:

```
Unable to open device!
```

There are three techniques available to avoid this problem:

1. Use the Unix chmod command to change permissions of the /dev/tty*nnn* device. This technique is not recommended.

2. Use the Unix *mknod* command to create a "local" version of the serial device, then use the *chmod* command to change the permissions on the "local" version of the device. For example:

```
mkdir $GLDIR/dev
mknod $GLDIR/dev/tty1a c major minor
chmod +rw $GLDIR/dev/tty1a
```

   This technique is not recommended.

3. The recommended technique to allow non-root users to run $REMOTE is to use the global "-r" command line option to grant that user root permissions when using Global System Manager (see section 6.3.3).

G.72 Serial Port Driver and Global System Manager (Unix)

This section describes some special factors that must be considered when using the Serial Port Driver (SPD) on Global System Manager (Unix).

Note that SPD is not available on every Global System Manager (Unix) platform although the dummy %.C2S module (see below) is distributed on all Global System Manager (Unix) BACRES volumes. In particular, SPD has only been tested on SCO Unix.

For Global System Manager (BOS) computers, the "SPD software" consists of two components:

SPD_PROG End-user Cobol program that uses the SPD handler. For example SPTEST;

%.*amSnn* Assembler SPD handler (interface to hardware)
am = Machine-code (e.g. J5 for the IBM PC).

For Global System Manager (Unix) computers, the "SPD software" consists of three components:

SPD_PROG End-user Cobol program (as above);

%.C2S Dummy SPD handler;

glintd Real SPD handler (uses Unix system calls) within the *glintd* module. Note that *glintd* is distributed on the BACNAT diskette/tape.

The end-user Cobol program (i.e. *SPD_PROG*) loads the relevant SPD handler onto the Global Cobol User Stack within the Cobol User Area. On Global System Manager (BOS) it is perfectly permissible to execute assembler code loaded within a Cobol User Area (i.e. on the Cobol stack). However, on Global System Manager (Unix) the Unix kernel treats the Cobol User Area as a "data segment" so that it is not possible for *SPD_PROG* to invoke a C SPD handler on the Cobol User Stack. To circumvent this problem, the dummy %.C2S module supplied with Global System Manager V7.0 and V8.0 contains a Cobol ESCAPE instruction followed by a data area that redirects the C version of the Commercial Code Interpreter (CCI) to the real SPD handler within *glintd* (via a jump-table, also in *glintd*). This technique is used for the other C "assembler assist modules" (e.g. %.C2D, BW999C).

If the entry-point to the real SPD controller is missing from the jump-table the CCI will return a "STOP with 772". This will be displayed as either:

```
$91 TERMINATED - STOP 772
```

or:

```
SORRY - YOU ARE ATTEMPTING TO USE A SERVICE THAT IS NOT PART OF
        YOUR OPERATING SYSTEM.
```

The first version of the BACNAT software that includes a real SPD handler is V3.103. Consequently, if an attempt is made to use SPD with BACNAT software prior to V3.103, the above STOP CODE will result. Unfortunately, due to a "feature" in the C version of the CCI, this STOP CODE will be persistent. The only method available to free up a partition stuck with a STOP CODE 772 is to use *glclean* to clean the particular SYSTEM (see section 6.7).

Because of the special nature of the Unix SPD handler (e.g. it does not access the serial port directly but uses Unix read/write system calls to perform all i/o) several factors must be considered when using SPD on a Global System Manager (Unix) computer.

G.72.1 SPD restrictions

The Unix SPD handler suffers from several restrictions. Some of these restrictions MAY be removed in future versions of the BACNAT software but others are completely intractable.

G.72.1.1 Restriction on number of SPD users

The Unix SPD handler must only be loaded by one partition per SYSTEM otherwise unpredictable results will occur. This restriction MAY be removed in future versions of the BACNAT software.

SPD may be used simultaneously by users on different SYSTEMs provided, of course, that different serial ports are being used.

G.72.1.2 Use of SP-block fields by the Unix SPD handler

The Unix version of the SPD handler does not require all the fields in the SP-block to be established. The following table describes the use of the SP-block (as described in section 5.1.2 (page 5-1) of the Global Cobol Assembler Interface Manual V6.1):

SPRES — Result code.

The following result codes are returned by the Unix SPD handler:

0	Successful operation
1	Unable to open port
2	Not used
3	Illegal baud-rate
4	Illegal number of stop bits
5	Illegal number of data bits
6	Illegal parity
7	Not used
8	Not used
9	Time-out elapsed
10	Incorrect number of parameters
11	Illegal op-code
12	Not used

13 Not used
14 Not used
-1 Awaiting completion

SPOPC Operation code.
See sections G.72.1.4 to G.72.1.12.

SPRBLN Length of receive buffer.
Not used (see section G.72.1.5).

SPTBLN Length of transmit buffer.
Not used (see section G.72.1.6).

SPRTO Receive time-out.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.5)

SPTTO Transmit time-out.
Not used (see section G.72.1.6).

SPRTER Up to 9 receive terminators.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.5)

SPEOF Terminating character.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.5)

SPLENG Length of Tx/Rx operation.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see sections G.72.1.5 and G.72.1.6)

SPMRTS Set RTS modem control.
Not used (see section G.72.1.8).

SPMDTR Set DTR modem control.
Not used (see section G.72.1.8).

SPMCTS CTS modem control status.
Not used (see sections G.72.1.9 and G.72.1.10).

SPMDSR DSR modem control status.
Not used (see sections G.72.1.9 and G.72.1.10).

SPMDCD DCD modem control status.
Not used (see sections G.72.1.9 and G.72.1.10).

SPWORK SPD harness work area.

SPBAUD Required baud rate.

Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.4)

SPBITS Required number of data bits.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.4)

SPSTOP Required number of stop bits.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.4)

SPPAR Required parity.
Used as described in section 5.2.2 of the Global Cobol Assembler Interface Manual V6.1 (see section G.72.1.4)

SPFLOW Required flow control.
The following bits in the SPFLOW byte are recognised:

Bit-0	#01	0 = Disable IXON
		1 = Enable IXON
Bit-1	#02	0 = Disable IXOFF
		1 = Enable IXOFF
Bit-2	#04	0 = Disable CTSFLOW
		1 = Enable CTSFLOW
Bit-3	#08	0 = Disable RTSFLOW
		1 = Enable RTSFLOW
Bit-4	#10	0 = Enable CLOCAL
		1 = Disable CLOCAL
Bit-5	#20	Unused
Bit-6	#40	Unused
Bit-7	#80	Unused

Note that the use of the SPFLOW byte for SPD with Global System Manager (Unix) is different from the use of the SPFLOW byte for SPD with Global System Manager (BOS). For a description of the bits in SPFLOW please refer to the termio(M) section of your Unix manual.

SPDEV Hardware device address.
Not used (see section G.72.1.3).

SPVECT Hardware interrupt vector.
Not used (see section G.72.1.3).

SPFULL Space left in Rx buffer when busy is signalled.
Not used (see section G.72.1.5).

SPFREE Space left in Rx buffer when not busy is signalled.
Not used (see section G.72.1.5).

G.72.1.3 SPD shell variable

The SPD serial port is not specified using the SPDEV and SPVECT parameters. Instead, the GLSPD shell variable specifies the name of the serial device which will be used by the SPD handler. The GLSPD shell variable must be added to the user's .profile script (or wherever else the GLDIR and PATH variables are set) as follows:

```
GLSPD=device_name;export GLSPD
```

For example:

```
GLSPD=/dev/tty1a;export GLSPD
```

If this shell variable is not established then an error 1 will be returned when attempting to OPEN the serial port.

Future versions of the BACNAT software MAY allow a selection of SPD devices to be specified in the Systems file. For example:

```
SPD  0     /dev/tty1a
SPD  1     /dev/tty2a
SPD  2     /dev/tty3a
```

If this change is implemented, the SPDEV field must contain the SPD device number, as defined in the Systems file, that corresponds to the required physical device.

G.72.1.4 Restrictions on the SPD Open operation (SPOPC=0)

The serial port selected for use by SPD using the GLSPD variable (see section G.72.2) must be "disabled". For example, use the following command to disable the serial port before loading Global System Manager:

```
disable $GLSPD
```

The SPD handler uses the isatty Unix system call to ensure that the selected device is suitable for serial data transfer. When the SPD Open operation is issued, the *ioctl* Unix system call is used to initialise the serial port to the protocol defined in SPBAUD, SPBITS, SPSTOP and SPPAR. The flow control defined in SPFLOW is also established by the Open operation.

The Unix SPD handler does not require user-supplied receive or transmit buffers (although, for compatibility with SPD on BOS, the Open operation does expect 3 parameters:

CALL *entry_point* USING SP *receive_buffer transmit_buffer*

the *receive_buffer* and *transmit_buffer* parameters are not used). Consequently, the SPRBLN, SPTBLN, SPFREE and SPFULL fields are not used.

G.72.1.5 Restrictions on the SPD Receive operation (SPOPC=1)

The Receive operation accepts bytes from the serial port. If the characters required are already in the Unix kernel receive buffer, the operation will complete immediately, otherwise it will set SPRES to -1 and signal an exception. Your program MUST wait for completion. The use of SPRTO, SPRTER, SPLENG and SPEOF are described in section 5.2.2 (page 5-5) of the Global Cobol Assembler Interface Manual V6.1.

G.72.1.6 Restrictions on the SPD Transmit operation (SPOPC=2)

The Transmit operation sends a block of up to SPTBLN bytes to the serial port. It does this by writing the bytes to the Unix kernel transmit buffer. The process will be blocked until the write operation completes. The SPTTO field is ignored by the Transmit operation.

A Transmit operation with SPLENG set to 0 is ignored.

G.72.1.7 Restrictions on the SPD Close operation (SPOPC=3)

When the SPD Close operation is issued, the *ioctl* Unix system call is used to reset the serial port to its original settings.

G.72.1.8 Restrictions on the SPD Set modem operation (SPOPC=4)

This operation is not supported by the Unix SPD handler and will result in error 11 (illegal opcode) if attempted. It is not possible to manipulate the modem control lines using the *ioctl* Unix system call.

G.72.1.9 Restrictions on the SPD Read modem operation (SPOPC=5)

This operation is not supported by the Unix SPD handler and will result in error 11 (illegal opcode) if attempted. It is not possible to read the modem control lines using the *ioctl* Unix system call.

G.72.1.10 Restrictions on the SPD Modem change operation (SPOPC=6)

This operation is not supported by the Unix SPD handler and will result in error 11 (illegal opcode) if attempted. It is not possible to read the modem control lines using the *ioctl* Unix system call.

G.72.1.11 Restrictions on the SPD Check Tx operation (SPOPC=7)

This operation is not supported by the Unix SPD handler and will result in error 11 (illegal opcode) if attempted. All Tx data is placed immediately into a Unix kernel buffer (using a *write* Unix system call).

G.72.1.12 Restrictions on the SPD Clear Tx operation (SPOPC=8)

This operation is not supported by the Unix SPD handler and will result in error 11 (illegal opcode) if attempted. All Tx data is placed immediately into a Unix kernel buffer (using a *write* Unix system call).

G.72.2 SPD diagnostic shell variable

In addition to the mandatory GLSPD shell variable (see section G.72.1.3), the Unix SPD handler also recognises another shell variable that may be "set" to enable detailed diagnostics. The name of the optional, diagnostic shell variable is GSM_VC2PA. To enable diagnostics define the GSM_VC2PA variable as follows:

```
GSM_VC2PA="YES";export GSM_VC2PA
```

All the diagnostics displayed by the SPD handler when the GSM_VC2PA variable is defined are sent to the /dev/stdout device.

The diagnostic information is very detailed and should normally only be enabled to determine the cause of an error returned from the Open operation (SPOPC=0).

G.72.3 SPD and non-superusers

Serial devices that are not terminal devices are normally set with the following permissions:

```
crw-------  owner=bin   group=terminal
```

Therefore non-superusers do not have read/write permissions for the serial device and therefore are unable to open the serial channel. Consequently, it is not normally possible for any user other than *root* to use SPD on Global System Manager (Unix). If diagnostics are enabled (see section G.72.2) the following message will accompany the "Non-existent port" error (SPRES=1) returned by the Open operation (SPOPC=0) if a non-root user attempts to use the SPD handler:

```
Unable to open device!
```

There are three techniques available to avoid this problem:

1. Use the Unix chmod command to change permissions of the */dev/ttynnn* device. This technique is not recommended.

2. Use the Unix *mknod* command to create a "local" version of the serial device, then use the *chmod* command to change the permissions on the "local" version of the device. For example:

   ```
   mkdir $GLDIR/dev
   mknod $GLDIR/dev/tty1a c major minor
   chmod +rw $GLDIR/dev/tty1a
   ```

   This technique is not recommended.

3. The recommended technique to allow non-root users to use the SPD handler is to use the global "-r" command line option to grant that user root permissions when using Global System Manager (see section 6.3.3.6).

G.73 PCWS File Transfer and Global System Manager (Unix)

PC Workstation V8.0 includes support for Global System Manager (Unix) computers. The Global System Manager TAP mechanism ensures that the PCWS terminal emulator is largely independent of the host operating system thus no changes are required to the BACNAT software to allow PCWS to emulate a screen running Global System Manager (Unix). However, the PCWS File Transfer utility, FTRAN, is not so straightforward - this section describes some special factors that must be considered when using FTRAN on Global System Manager (Unix).

The terminfo entries for 4 different PCWS terminal emulations are described in the Global PC Workstation V8.0 Notes (document MQXNV8.0). Due to restrictions in the PCWS terminal emulator, the terminfo entries are incomplete (e.g. reverse scroll is not supported) so that, for example, vi does not function in full screen mode when TERM=pcws. However, the PCWS terminfo entries can be used to map the TERM shell variable to a Global System Manager by including the following entries in the Systems file:

```
TERM
      pcws 706          # 80-column, monochrome
      pcws-c 707        # 80-column, colour
      pcws-w 704        # 132-column (i.e. wide), monochrome
      pcws-cw 705       # 132-column (i.e. wide), colour
```

For Global System Manager (BOS) computers, the "FTRAN software" consists of two components:

FTRAN File transfer utility written in Global Cobol;

*a*KLOW Kermit packet conversion routine (in P.PCWS library), written in assembler (where *a* = Architecture code (e.g. J for 8086 family based computers))

For Global System Manager (Unix) computers, the "FTRAN software" consists of three components:

FTRAN File transfer utility written in Global Cobol;

CKLOW Dummy Kermit Packet Conversion routine;

glintd Real Kermit Packet Conversion routine within the *glintd* module.

FTRAN loads the relevant Kermit Packet Conversion routine onto the Global Cobol User Stack within the Cobol User Area. On Global System Manager (BOS) it is perfectly permissible to execute assembler code loaded within a Cobol User Area (i.e. on the Cobol stack). However, on Global System Manager (Unix) the Unix kernel treats the Cobol User Area as a "data segment" so that it is not possible for FTRAN to invoke a C assist file on the Cobol User Stack. To circumvent this problem, the dummy CKLOW module supplied with PCWS V8.0 contains a Cobol ESCAPE instruction followed by a data area that redirects the C version of the Commercial Code Interpreter (CCI) to the real Kermit Packet Conversion routine within *glintd* (via a jump-table, also in *glintd*). This technique is used for the other C "assembler assist modules" (e.g. %.C2D, BW999C).

If the entry-point to the real Kermit Packet Conversion routine is missing from the jump-table the CCI will return a "STOP with 772". This will be displayed as either:

```
$91 TERMINATED - STOP 772
```

or:

```
SORRY - YOU ARE ATTEMPTING TO USE A SERVICE THAT IS NOT PART OF
        YOUR OPERATING SYSTEM.
```

The first version of the BACNAT software that includes a real Kermit Packet Conversion routine is V3.91. Consequently, if an attempt is made to use FTRAN with BACNAT software prior to V3.91, the above STOP CODE will result. Unfortunately, due to a "feature" in the C version of the CCI, this STOP CODE will be persistent. The only method available to free up a partition stuck with a STOP CODE 772 is to use *glclean* to clean the particular SYSTEM (see section 6.7). **At the time of writing, the "C" version of the PCWS packet conversion routine is only supported on SCO Unix (see document C5522), AIX V3.2.5 (see document C5539) and Motorola Unix V/88 SVR3 (see document C5535).**

Note that versions of PCWS earlier than V8.0 (i.e. V7.0 and V6.0) did not include the CKLOW module in P.PCWS so will fail with the following error message if run on a Global System Manager (Unix) implementation:

```
PROGRAM CKLOW IN P.PCWS REQUIRED ON ...
```

There are two potential problems when using FTRAN to transfer files to/from a Global System Manager (Unix) computer. Firstly, the data-blocking within the Unix kernel may result in relatively long delays between consecutive characters transmitted down the serial line. To

prevent this effect from causing packet-loss we recommend that the "File Transfer Timing Parameter" in FTRAN should be set to a nonzero value. During the testing of the "C" Kermit Packet Conversion, the FTRAN "File Transfer Timing Parameter" was set to 3, the FTRAN "Initialisation Retry Count" was set to 4000 and the PCWS "Initialisation Count" was set to 1000 but some experimentation may be required.

The second problem that may occur with FTRAN on Global System Manager (Unix) is caused by the Unix device driver intercepting ^Q (XON, 0x11) and ^S (OFF, 0x13) characters. Although the FTRAN protocol prefixes both 0x11 and 0x13, both 0x91 (i.e. 0x11 with top bit set) and 0x93 (i.e. 0x13 with top bit set) are transmitted as raw characters. These characters may be intercepted by the Unix device driver resulting in an irrecoverable lost packet. There are two methods available to circumvent this problem. Firstly, PCWSCUS can be used to enable 8-bit prefixing during FTRAN data transfer (see document MQXNV8.0). Alternatively, the "-x" option in *global* (see section G.70) can be used to disable XON/XOFF flow control in the Unix device driver.

The following table describes the possible options:

global option	**8-bit prefixing**	**FTRAN results**
None	Disabled	Fails if received data includes 0x91 or 0x93
None	Enabled	OK for all data patterns
-x	Disabled	OK for all data patterns
-x	Enabled	OK for all data patterns

G.74 $COBOL and Global System Manager (Unix)

As explained in Technical Bulletin GT591 (24-Feb-1993), the Global Cobol compiler, $COBOL, is now supported on most Global System Manager (Unix) configurations. Please refer to your Global Configuration Notes for full details regarding the availability of $COBOL on a particular version of Unix.

This section describes some special factors that must be considered when using $COBOL on Global System Manager (Unix).

For Global System Manager (BOS) computers, the "$COBOL software" consists of four sets of components:

$COBOL $COBOL Cobol front-end;

$COB*nn* Various compiler overlays written in Cobol;

$COB*nn* Various compiler overlays written in the TRAMS language (The TRAMS language is a proprietary language designed for writing compilers);

$*a*TRAM TRAMS language interpreter (written in assembler), *a* = Architecture-code (e.g. J for 80x86-based computers);

For Global System Manager (Unix) computers, the "$COBOL software" consists of five sets of components:

$COBOL $COBOL Cobol front-end;

$COB*nn*	Various compiler overlays written in Cobol;
$COB*nn*	Various compiler overlays written in the TRAMS language (The TRAMS language is a proprietary language designed for writing compilers);
$CTRAM	Dummy TRAMS language interpreter;
glintd	Real TRAMS language interpreter (written in C) within the *glintd* module. Note that *glintd* is distributed on the BACNAT diskette/tape.

$COBOL loads the TRAMS interpreter onto the Global Cobol User Stack within the Cobol User Area. On Global System Manager (BOS) it is perfectly permissible to execute assembler code loaded within a Cobol User Area (i.e. on the Cobol stack). However, on Global System Manager (Unix) the Unix kernel treats the Cobol User Area as a "data segment" so that it is not permissible for $COBOL to invoke the C version of the TRAMS interpreter on the Cobol User Stack. To circumvent this problem, the dummy $CTRAM module supplied via Autozap MK6220 contains a Cobol ESCAPE instruction followed by a data area that redirects the C version of the Commercial Code Interpreter (CCI) to the real TRAMS interpreter within *glintd* (via a jump-table, also in *glintd*). This technique is used for the other C "assembler assist modules" (e.g. %.C2D, BW999C).

If the entry-point to the real TRAMS interpreter is missing from the jump-table the CCI will return a "STOP with 772". This will be displayed as either:

```
$91 TERMINATED - STOP 772
```

or:

```
SORRY - YOU ARE ATTEMPTING TO USE A SERVICE THAT IS NOT PART OF
        YOUR OPERATING SYSTEM.
```

The first version of the BACNAT software that includes the TRAMS interpreter is V3.94 (see GT591). Consequently, if an attempt is made to use $COBOL with BACNAT software prior to V3.94, the above STOP CODE will result. Unfortunately, due to a "feature" in the C version of the CCI, this STOP CODE will be persistent. The only method available to free up a partition stuck with a STOP CODE 772 is to use *glclean* to clean the particular SYSTEM (see section 6.7).

G.75 Screen clear and/or reset after $BYE

$BYE does not clear the screen when closing down Global System Manager. If it is required to clear the screen as part of the "$BYE processing", follow the steps outlined below:

1. Use your favourite Unix editor to create the following script in the $GLDIR/bin directory. For example:

   ```
   $ vi $GLDIR/bin/glclr
   ```

   The script should contain the following two lines:

   ```
   global
   tput clear
   ```

2. Make the script executable:

```
$ chmod +x $GLDIR/bin/glclr
```

3. If *glclr*, instead of *global*, is used to invoke a Global System Manager session, when control is returned to Unix (e.g. after running $BYE), the screen will be cleared.

This technique may also be used to reset the screen after a Global System Manager session. For example, if $BYE is keyed while a Global System Manager SAA-style menu is displayed, the screen may left in "reverse video". To reset the screen back to "normal video" include the following line in the glclr script:

```
tput rmso
```

For a full description of the tput(C) command, refer to your Unix manuals.

G.76 No shell prompt after exiting Global System Manager

The following message is displayed when Global System Manager is terminated when no other SYSTEM's are active:

```
Terminating processes and tidying Global System Manager resources
```

The display of this message can interfere with the display of the shell prompt. The absence of a shell prompt may be incorrectly interpreted as a system crash. If a shell prompt does not appear immediately after exiting Global System Manager, key <CR> to produce a new prompt. If, for cosmetic reasons, an immediate shell prompt is required, create a script to cause a small delay after exiting Global System Manager. For example:

1. Use your favourite Unix editor to create the following script in the $GLDIR/bin directory. For example:

   ```
   $ vi $GLDIR/bin/gldelay
   ```

   The script should contain the following two lines:

   ```
   global
   sleep 2
   ```

2. Make the script executable:

   ```
   $ chmod +x $GLDIR/bin/gldelay
   ```

3. Use *gldelay*, instead of *global*, to invoke Global System Manager.

G.77 Dynamic printer device descriptions from Systems file

This topic has been moved to sections 9.5.1 and 9.5.2.

G.78 Another shell variable: GLNAME

This topic has been moved to section 8.3.1.

G.79 Problems with software installation (part-2)

This section describes a problem that may occur when installing Global products from tape. This problem only affects those products that are artificially split into 2 or more product codes. For example, Global 2000 Nominal Ledger is distributed on two volumes, NLA and SKA.

During software product installation, the installation job expects the first product volume (i.e. NLA in this example) to be demounted and replaced by the second product volume (i.e. SKA in this example). When the software is distributed on diskettes replacing the NLA volume by the SKA volume is possible. However, when the software is distributed on tape, the NLA and SKA units are on different subvolumes (e.g. NLA on 264, SKA on 263 - assuming the steps outlined in section 2.5.3 have been followed).

Consequently, when the installation job expects the SKA volume to replace the NLA volume, the following message will be displayed:

```
:IO :SKA :NLDATA
PLEASE MOUNT SKA ON Discrete data file - 264 AND KEY <CR>:
```

If a message of this form appears when installing a Global product from tape, the following steps must be taken:

1. Switch to another partition;

2. Allocate a work unit (e.g. 265) to hold a copy of unit 264;

3. Copy all the files on unit 264 (i.e. NLA) to 265;

4. Copy all the files on unit 263 (i.e. SKA) to 264 (i.e. NLA);

5. Rename unit 264 to SKA;

6. Return to the original partition to continue the installation.

If the installation job requires the original contents of 264 to be available, the following message will appear:

```
:IO :NLA :NLDATA
PLEASE MOUNT NLA ON Discrete data file - 264 AND KEY <CR>:
```

The contents of 264 must be restored from the work unit.

G.80 File permissions of the Global System Manager $GLDIR directory

Section G.12 describes the expected file permissions of all files and directories within the $GLDIR directory. If the ownerships or permissions of any files or directories beneath the $GLDIR directory are altered Global System Manager will behave unpredictably. If you suspect a problem with Global System Manager (e.g. printers not working) is caused by incorrect Unix file permissions try running Global System Manager as super-user (i.e. root) or use the global "-r" option (see section 6.3.3.6). If the problem disappears when Global System Manager is running with root privileges, it is almost certainly caused by incorrect file permissions.

The file permissions on the $GLDIR directory itself are important and must be such that all global users are granted "read" and "execute" (i.e. search) permissions on this directory. This

can be accomplished either by arranging for $GLDIR to be owned by group "global" with restricted "other" permissions or, if $GLDIR is not owned by group "global", by changing the "other" permissions to "r-x".

G.81 Using rlogin to allocate unique Unix login names

This section describes how a particular site-specific "problem" was overcome. The solution may be of general use.

The site in question consists of a front-end Unix computer running as a terminal emulator connected (via TCP/IP) to a back-end Unix computer which is hosting Global System Manager. All users log into the front-end computer via serial consoles then use TCP/IP *rlogin* to log into the Global System Manager computer. For various reasons, all users must sign into the front-end computer with the same login name.

The problem was to find a technique that could be used to associate a unique login-name to each operator so that the Systems file LOGNAME mapping could be used to allocate a consistent SYSTEM number to each operator. The TTY mapping could not be exploited because each operator is associated a virtual, and unpredictable, ttyname when using *rlogin*. A solution, exploiting the GLNAME shell variable (see section 8.3.1), was also untenable because all users share the same home directory - the overhead of a separate script for each user, to establish the GLNAME variable, was deemed unworkable.

The following solution, which depends on all operators being allocated unique, fixed ttynames on the terminal emulator computer (which is the situation because all terminals are connected via serial tty devices with fixed device names), is currently in use at the site:

1. The following script, on the front-end computer, is used instead of the "bare" *rlogin* command to commence a terminal emulation session:

```
TEMP1=`tty`
TEMP2=`basename $TEMP1`
echo "Logging on to host computer as login-id $TEMP2"
rlogin hostname -l $TEMP2
```

2. The $GLDIR/sys/Systems file, on the back-end computer, includes entries of the form:

```
SYSTEM nn LOGNAME ttyname
```

   Where *ttyname* is a unique ttyname associated with a particular user on the front-end computer.

For example, if a user logs on to the "front-end" computer using terminal device /dev/tty1a, their login name on the "back-end" computer will be tty1a.

G.82 Removing the terminating FORM-FEED from Global System Manager reports

This section describes how a particular site-specific "problem" was overcome. The solution may be of general use.

The Global System Manager Printer Executive always appends a FORM-FEED character (i.e. 0x0c) to all reports. The form-feed ensures that the paper is correctly aligned when the next

report is printed. Unfortunately, the extra FORM-FEED inserted by Global System Manager can cause spurious page throws, resulting in wasted paper, when the Unix spooler also insists on inserting FORM-FEED characters at the end of each report.

Two trivial solutions were proposed:

1. Print directly to the printer rather than using the Unix spooler;

2. Modify the Unix spooler script to remove the insertion of the FORM-FEED. For example, on SCO Unix change the script:

```
/usr/spool/lp/admins/lp/interfaces/tty1a_spool
```

from:

```
do
for file in $files
      do
            cat "$file" 2>&1
            echo "\014\c"
      done
      i = `expr $i + 1`
done
```

to:

```
do
for file in $files
      do
            cat "$file" 2>&1
      done
      i = `expr $i + 1`
done
```

At the site in question, neither of these proposed solutions were acceptable, so the following script, $GLDIR/bin/glfilter, was created to remove the last two characters (i.e. 0x0c, 0x0d) from every Global System Manager report:

```
TEMP=`cat $1 | wc -c`
TEMP=`expr $TEMP - 2`
dd if=$1 count=$TEMP ibs=1 obs=1|lp -d tty1a_spool
```

In the Systems file, the line:

```
510 "lp -d tty1a_spool"
```

was replaced by:

```
510 "glfilter"
```

Note that with the release of Global System Manager V8.1 a new $CUS option is available to avoid this problem (see Chapter 6 of the Global System Manager Manual).

G.83 Problems with O2A BACRES

As explained in Global Technical Bulletin GT619 (8-July-1993), the distribution format for software on 3½" diskettes was changed from B3B to O2A/O2D at the end of August 1993. This change introduced a problem with some Global System Manager (Unix) configurations because the BACNAT software was not amended to alter the expected BACRES format from B3B to O2A/O2D.

The problem occurs during the execution of the glinstall script immediately after the BACRES diskette has been accessed:

```
Please mount BACRES on /dev/.....

Key <CR> to continue, q to quit

Configuring Global System Manager (BACNAT V3.nnn); please wait

WARNING: floppy: Read error (Unix specific error messages)
glintd: Error 1211 - Can't read configure information from SYSDATA
global: Error 1017 - Fileserver A has failed to configure; system terminating.

Terminating processes and tidying Global System Manager resources
```

The problem will only affect the following configurations:

5523	Sun SPARC
5524	IBM RS/6000 AIX V3.1.x
5527	SCO Unix
5539	IBM RS/6000 AIX V3.2.x
5540	HP-9000 (HP-UX V8)

To avoid the problem, quit the glinstall installation script when the prompt to mount the BACRES diskette is displayed (see above) and use glconfig (after defining the GLDIR and PATH shell variables - see sections 8.1.1 and 8.1.2) to change the name of the INSDEV device from "B3" to "O2". For example, for SCO Unix (5527) the following change is required:

```
INSDEV /dev/rfd096ds9 ; INSDEV set to "B3"
```

to:

```
INSDEV /dev/rfd0135ds18 ; INSDEV set to "O2"
```

Similarly, for AIX V3.2.x (5539) the following change is required:

```
INSDEV /dev/rfd0l ; INSDEV set to "B3"
```

to:

```
INSDEV /dev/rfd0h ; INSDEV set to "O2"
```

Use the "global -i" command (see section 6.3.2) to re-load the Global System Manager starter system from the O2D BACRES diskette.

Note that format O2D is isometric with O2A. O2D should be considered as the "Unix version" of O2A. An O2A diskette can be accessed using the O2D unit address, and vice versa. Thus, if a configuration only includes the O2D volume format, the corresponding unit address (e.g. a40) can be used to access both O2D and O2A format diskettes.

G.84 New BACNAT software required for Global System Manager V8.1

Global System Manager V8.1 expects several facilities, the details of which are beyond the scope of this manual, to be available in the BACNAT software. These facilities are not required for Global System Manager V8.0. In order for Global System Manager V8.1 to run on a Unix configuration the BACNAT software must be variant V3.163, or later. Note that for the various pre-releases of Global System Manager V8.1, BACNAT variants V3.111, or later, were required.

When upgrading Global System Manager V8.0 (or V7.0) to V8.1, the new BACNAT software MUST be re-installed prior to the installation of Global System Manager V8.1.

G.85 How to detect if Global System Manager is running

Several methods are available to determine if Global System Manager is running on a Unix computer. The most reliable technique involves testing for the presence of the Global shared memory (see section G.32). The following example script may be useful:

```
if ipcs|grep "0x47534d30" > /dev/null
then
      echo "GSM is already running"
else
      echo "About to run GSM for the first time"
      global
fi
```

Note that the string that is "grep'ed" will have to be changed if the GLIPCBASE Unix shell variable is defined (see section G.53).

G.86 Diagnostic Unix shell variables

In addition to the Unix shell variables described in Chapter 8, Global System Manager (Unix) recognises several diagnostic Unix shell variables. These shell variables should not be defined (i.e. set) unless you are advised to do so by the TIS Software Service Centre:

GSM_VC0B	Enables Console Executive diagnostics. The first character of the text string associated with this shell variable specifies the control character that can be used to toggle up to 10 diagnostic flags (e.g. set GSM_VC0B=X to define <CTRL X> for this function). When <CTRL X>, in this example, is keyed followed by a number between 0 and 9 a diagnostic flag, which may be used by other modules, is toggled zero/nonzero;
GSM_VC0A	Enables File Executive stdout diagnostics;
GSM_FTRACE	Enables File Executive trace diagnostics;
GSM_VC0SH	Enables SVC-61 & UCI stdout diagnostics;
GSM_VC0SH_XLAT	Enables "stream i/o" to "raw i/o" operation mapping for **some** (obsolete) BACNAT variants (see section G.100);
GSM_VC0SH_NOXL	Disables "stream i/o" to "raw i/o" operation mapping for current BACNAT variants (see section G.100);
GSM_VC2AA	Enables SSD controller stdout diagnostics;
GSM_VC2CA	Enables $TAPE controller stdout diagnostics;
GSM_VC2GG	Enables glspod stdout diagnostics (in addition to the standard messages written to the $GLDIR/sys/spool/log file);

GSM_VC2KA	Enables PCWS/FTRAN stdout diagnostics;
GSM_VC2PA	Enables SPD controller stdout diagnostics;
GSM_VC2RA	Enables $REMOTE controller stdout diagnostics.

G.87 Possible warning message on the Sun SPARC with SunOS V4.1.x

Please refer to the Global Configuration Notes for the Sun SPARC (document C5523) for the details of an error message of the following form that may occur when loading Global System Manager (Unix) on Sun OS V4.1.1:

```
id.so: warning /usr/5lib/lib has older version than expected
```

where *lib* is the name of a SunOS demand paged shared library.

G.88 Problem with Exabyte tapes on AIX V3.2.5 on the IBM RS/6000

Please refer to the Global Configuration Notes for AIX V3.2.5 on the IBM RS/6000 (document C5539) for full details of a Unix shell variable that can be used to change the block size of the Unix $TAPE controller from the default value of 512 bytes to 1Kb. This shell variable, GLTAPE1K, must be defined in order to restore, on an RS/6000 running AIX V3.2.5, an 8mm Exabyte tape created on a computer running Global System Manager (BOS). For example:

```
GLTAPE1K=YES;export GLTAPE1K
```

G.89 Diskette device names on AIX V3.2.5 on the IBM RS/6000

As explained in the Global Configuration Notes for AIX V3.2.5 on the IBM RS/6000 (document C5539) the default Unix device names for 3½" diskettes in the glinstall script and $GLDIR/sys/Systems files have undergone several revisions. For all V8.1 compliant BACNAT variants the device name for all 3½" formats supported on the RS/6000 (i.e. B3, O2 and V2) is "/dev/rfd0".

G.90 Simulated hard-disk volumes

Section G.52 describes a technique that allows a diskette volume to be simulated on a Unix hard-disk. This facility has been used successfully at many sites. However, under some circumstances, the capacity of the available diskette formats is too small to be generally useful. To alleviate this problem, the following Discrete Logic Volumes (DLV's) have been created, for use with Global Configurator V8.1, to provide a series of virtual units of varying capacities:

Format	*Description*
Z151Z	4Mb simulated volume
Z152Z	8Mb simulated volume
Z153Z	16Mb simulated volume
Z154Z	32Mb simulated volume
Z155Z	64Mb simulated volume

To add a simulated volume to a Global System Manager (Unix) configuration:

1. Use CFUPDATE (see section 9.2.3) to add a diskette (*sic*) controller to the existing configuration file. For example:

```
CONTROLLER:         DISKETTE
DRIVE:              1
VOLUME FORMAT:      Z155Z
UNIT NUMBER:        151
```

2. Use gladmin to add an entry to the Systems file of the following form (see section 7.3.3):

```
DISKETTE 1 Z155 /usr/global/data/z155z
```

   Where the "1" must match the drive number in the configuration file and "Z155" must match the "Z155Z" in the configuration file. Note that, as usual, the last letter of the configuration file AnA code (i.e. the "Z") is dropped from the format code in the Systems file.

3. Use glmkdat, option 2 (i.e. Integrated Data File - see section 6.6.2) to create /usr/global/data/z155z of the appropriate size (i.e. 64Mb).

G.91 Potential problem configuring 26 file server SYSTEM's

If all 26 possible file-server SYSTEM's (i.e. "A" to "Z") are defined in the $GLDIR/sys/Systems file, then an error type 1032 (see Appendix C) may occur on some versions of Unix. For example, the problem occurs on SCO Unix V3.2.x. The problem arises because the maximum number of Semaphore Identifiers defined in the Unix kernel is less than the number of file-server SYSTEM's defined in the Systems file. Global System Manager (Unix) requires 1 Semaphore Identifier for each file-server SYSTEM. The problem can be avoided by increasing the number of Semaphore Identifiers in the Unix kernel. For SCO Unix, this is achieved by increasing the value of the SEMMSL parameter. Please consult your Unix System Administrator's Guide for further information.

G.92 Possible problem with DIRECT printers (part 1)

The Global System Manager (Unix) DIRECT printer interface is relatively complex (see Appendix D). The printer controller module within the *glintd* process (see Appendix E) passes print lines to the glprid printer daemon via a Unix named pipe (aka FIFO). The glprid daemon writes the print lines to the Unix device specified in the PRINTER section of the $GLDIR/sys/Systems file. If glintd were to write data to the Unix device directory, Global System Manager on the SYSTEM performing the printing would hang whenever a full output queue caused the Unix write system call to block (see section G.93).

BACNAT V3.115, and later, includes a fix for a problem that causes spurious NOT READY errors to be reported during DIRECT printing. This problem is caused by the glprid daemon occasionally failing to inform the printer controller, within *glintd*, that a new print line is required. If a request for a new print line is not received by the glintd printer controller within the Time-Out period specified in the configuration file (see section 9.5.1) a NOT READY error is reported.

However, even with the bug fixed (i.e. in BACNAT software V3.115, or later) a second problem that results in spurious NOT READY errors during DIRECT printing has been identified. The cause of the second problem is more involved and only occurs when an attempt is made to start a new print while a previous file is being printed (e.g. when the Global spooler, $SP, is

being used to print several reports). The problem only occurs if the time taken by glprid to flush the contents of the immediate Unix FIFO exceeds the Time-Out period specified in the Global configuration file. This problem is more likely to occur on a slow printer when a large FIFO is created. For most versions of Unix, the size of the FIFO is fixed (e.g. for SCO Unix, the size is 8Kb). In order to overcome this problem the DIRECT printer controller within *glintd* recognises a new prompt in the DIRECT printer controller section of the Global configuration file:

```
Maximum FIFO buffering/256 (0):
```

Set this value to *N* to limit the maximum number of characters placed in the FIFO to *N**256.

For example, to limit the number of characters to 1024 bytes, use Global Configurator to set the value to 4). If this value is left at the default of 0, the default Unix FIFO size (normally 8Kb, but refer to the PIPE_BUF definition in the */usr/include/limits.h* file) will be used.

G.93 Possible problem with DIRECT printers (part 2)

As explained in Appendix D, the printer controller module within the *glintd* process passes print lines to the glprid printer daemon via a Unix named pipe (aka FIFO). The glprid daemon writes the print lines to the Unix device specified in the PRINTER section of the $GLDIR/sys/Systems file. In some circumstances, typically to diagnose problems, it may be necessary to print to the Unix device directly from within the *glintd* process, by-passing the glprid daemon process entirely. This non-standard interface is enabled by setting the following flag in the DIRECT printer controller section of the Global configuration file to "Y":

```
Print direct without a FIFO? (N)
```

This extremely specialised technique may have several side-effects:

1. Normal users may not have sufficient permissions to open the Unix print device. The global -r option (see section 6.3.3.6) may have to be used by all non-super-users. If the open of the Unix print device fails, a Global System Manager SOFTWARE PROTECTION will be reported.

2. The direct DIRECT printer interface does not include any locking on the Unix print device so that several users may print to the same Unix device simultaneously, leading to garbled reports. The standard DIRECT printer interface uses the existence of the FIFO to prevent simultaneous prints by several SYSTEM's.

3. Using the direct DIRECT printer interface in one partition can seriously impair the performance of other partitions/users on the same SYSTEM.

G.94 Possible problem with DIRECT printers (part 3)

In order to provide diagnostic information to assist in the analysis of Global System Manager (Unix) printer problems, messages are written to the $GLDIR/sys/messages file (see section 6.1.1.3) if glintd is unable to create or open the FIFO or if the glprid process has been killed unexpectedly. For example, see messages 1232, 1233 and 1234 in Appendix C.

These diagnostics have been useful in tracking down several problems in the DIRECT printer handling. For example, the DIRECT printing interface will fail, with a variety of errors, if there are insufficient Unix file channels for the open operations to complete. Similarly, problems will

occur if the ownerships or permissions of the glprid daemon is altered in any way (see section G.66).

G.95 SCO Unix TAPs and the Autowrap feature

As explained in the Global Configuration Notes for SCO Unix V3.2.*x* (document C5522) a large number of Terminal Attribute Programs (TAPs) are available for the integral screen on an SCO Unix computer. These TAP's, a full list of which is included in the Global Configuration Notes, have evolved as new features have been implemented.

All SCO Unix TAP's in the range $.554 to $.608 include the ANSI "disable autowrap" escape sequence (0x1b5b3f376c) in the START SEQUENCES section of the TAP.

Conversely, the TAP's in the range $.690 to $.698 do not disable the "autowrap option" which, by default, is enabled when SCO Unix is initiated. Consequently, these TAP's do not function correctly with some Global System Manager utilities and fail with most Global applications. These specialised TAP's have been created to allow the operation of other SCO Unix applications that rely on the autowrap option being enabled (e.g. sysadmsh). As an alternative to using the special-purpose $.69n TAP's it may be more convenient to use <SYSREQ> S (see section 5.3) to define the "enable autowrap" sequence:

    1B5B3F3768

as the "screen reset" sequence.

G.96 Removal of D-ISAM code from BACNAT software

Prior to March-1995, all BACNAT's included either the C-ISAM sub-routines, as supplied by Informix Inc., or compiled D-ISAM sources, as supplied by Byte Designs Ltd. Since March-1995, coinciding with a proposed change to the C-ISAM licencing arrangement, all BACNAT's include either the C-ISAM sub-routines, as supplied by Informix Inc., or no Unix ISAM functions whatsoever.

To determine the availability, and version, of C-ISAM on a particular configuration please refer to your Global Configuration Notes.

G.97 De-allocate does not always release space to filing system

Due to a feature of the Separated Subvolume Domain (SSD) controller within the *glintd* process, disk space that is logically returned back to the Unix filing system by a $V de-allocate operation (see section 4.10.5) is not always immediately available for subsequent re-allocation. This can lead to unexpected errors from $V (e.g. STOP CODE 6604) if a large sub-volume is de-allocated and another allocated immediately without reloading Global System Manager between the operations.

G.98 Copying Data while Global System Manager is running

This section describes some difficulties that may occur when attempting to import data into a running Global System Manager (Unix) system. Obviously, if the computer supports a diskette drive, data may be copied using $F etc. via diskette. Similarly, if $TAPE is supported on the computer/hardware configuration $TAPE may be used to import data. For small amounts of data, $REMOTE may be used under some circumstances.

If Global System Manager is unable to access any suitable peripheral devices (i.e. diskette drive or tape drive) then data, encapsulated within Global System Manager volumes, may be imported using Unix utilities either directly (e.g. cpio, tar etc.) or remotely (e.g. rcp, ftp, uucp etc.). Note that "native" Unix data files may be imported into Global System Manager using Global File Converters. This section explains the precautions, in addition to the file ownerships and permissions considerations (see sections G.12 and G.13), that must be taken when transferring data within virtual diskettes (see section G.52), virtual DLV's (see section G.90), Integrated Data Files (see section 9.2.2) and Discrete Data Files (see section 9.2.1).

G.98.1 Copying Virtual Diskettes and DLV's

If data is copied from one Unix computer to another via a "virtual diskette" unit (e.g. Unix file $GLDIR/data/o2a_image - see section G.52) or a virtual DLV (e.g. Unix file $GLDIR/data/z155z - see section G.90), in both cases overwriting an existing "virtual file" of the same size, then Global System Manager can be left running while the transfer takes place. Obviously, if Global System Manager attempts to access the volume while the transfer/overwrite is in progress than data corruption may occur. However, once the Unix data transfer has completed Global System Manager will be able to access the freshly transferred data immediately (i.e. without the need to reload Global System Manager). If a file of the correct name (as defined in the $GLDIR/sys/Systems file) and size does not exist, to be overwritten, Global System Manager will not recognise the corresponding unit, without a reload. If the file is not of the correct size for the virtual diskette or virtual DLV, Global System Manager will report a INVALID VOLUME error message.

G.98.2 Copying Integrated Data Files

If data is copied from one Unix computer to another via an Integrated Data File (e.g. Unix file gsm200.vol), overwriting an existing Integrated Data File of the same size, or a different size, Global System Manager MUST be reloaded completely for the new volume to be recognised. If Global System Manager is not reloaded after the transfer, any attempt to access the new volume may result in data corruption.

The Integrated Data File controller, within the *glintd* process, maintains a table of open Unix files and is unaware when a new file has been imported. Thus, the only safe technique to copy data to a running Global System Manager system is via a virtual diskette or a virtual DLV (see section G.98.1).

G.98.3 Copying Discrete Data Files

If a Discrete Data File subvolume file (e.g. SVL10_G3DATA) is copied from one Unix computer to another, Global System Manager MUST be reloaded completely for the new subvolume to be recognised. If the freshly imported subvolume file does not overwrite a file of the same name, the corresponding sub-volume won't be recognised until Global System Manager is completely reloaded. However, if the new subvolume does overwrite an existing file of the same name and size, the new data may **appear** to be immediately accessible. If Global System Manager is not reloaded after the transfer, any attempt to access the new volume may result in data corruption.

The Discrete Data File controller (i.e. SSD controller), within the *glintd* process, establishes a table of Unix sub-volume files when Global System Manager is initiated. The SSD controller is unaware that a new file has been imported into the SSD directory until Global System Manager is restarted. Thus, the only safe technique to copy data to a running Global System Manager system is via a virtual diskette or a virtual DLV (see section G.98.1). Future versions of Global

System Manager **may** include a facility to allow Discrete Data Files domains to be reconstructed "on the fly".

G.99 Copying Data from MS-DOS to Unix

This section, which is related to section G.98, explains how Integrated Data Files and Discrete Data Files may be transferred from a Global System Manager (MS-DOS) configuration to a Global System Manager (Unix) configuration. Obviously, if both computers support a compatible diskette drive, data may be copied using $F, $SAVE etc. via diskette. Similarly, if both computers support a compatible tape drive, $TAPE may be used to transfer data. For small amounts of data, $REMOTE may be used under some circumstances.

This section describes the techniques, in addition to the precautions described in section G.98, that are available to transfer Global System Manager data from an MS-DOS computer to a Unix computer.

G.99.1 Copying Virtual Diskettes and DLV's

Unlike, Global System Manager (Unix), Global System Manager (MS-DOS) doesn't support virtual diskettes. However, the same techniques as described for IDF's (see section G.99.2) can be used to transfer data within a (seldom used) Global System Manager (MS-DOS) virtual DLV.

G.99.2 Copying Integrated Data Files

Any of the standard Global System Manager (MS-DOS) Integrated Data File domain volume formats (i.e. P151Z, P224Z, P246Z or P259Z) can be transferred to Global System Manager (Unix). For example:

- copy an MS-DOS IDF domain file (e.g. gsm200.vol) to Unix using a "DOS-to-Unix" file-transfer utility (e.g. the *doscp* command available on SCO Unix). The file may have to be renamed to obey the Unix file naming conventions;

- change the Unix file permissions of the newly imported file (see sections G.12 and G.13);

- modify the $GLDIR/sys/Systems file to include the new volume (see section F.2.2);

- modify the Global configuration file to include the new volume (see section 9.2.2);

- reload Global System Manager to recognise the new volume (see section G.98.2).

G.99.3 Copying Discrete Data Files

Any sub-volume file within either of the standard Global System Manager (MS-DOS) Discrete Data File domain volume formats (i.e. T151Z or T259Z) can be transferred to Global System Manager (Unix) **provided both volumes have been configured with the same number of files per directory** (almost invariably 250). For example:

- copy an MS-DOS DDF sub-volume file (e.g. 10G3DATA.SVL) to Unix using a "DOS-to-Unix" file-transfer utility (e.g. *ftp* or the *doscp* command available on SCO Unix);

- the file must be copied into a domain directory (e.g. B00.dir) and must be renamed to obey the Global System Manager (Unix) naming convention (e.g. SVL10_G3DATA). Check that the sub-volume number of the new file is unique (see section G.62);

- change the Unix file permissions of the newly imported file (see sections G.12 and G.13);

- reload Global System Manager to recognise the new sub-volume (see section G.98.3).

G.100 Unix stream file i/o vs. Unix raw file i/o

As explained in the Global File Converters Manual, SVC-61 supports both Unix stream i/o operations (e.g. fopen, fread, fwrite, fclose) and Unix raw i/o operations (e.g. open, read, write, close). The resilience of stream i/o operations in a multi-user Unix environment on some versions of Unix is questionable. There appears to be no guarantee that a stream write operation will complete immediately. This failing has caused a problem in the implementation of Speedbase C-ISAM (see Chapter 10), the details of which are beyond the scope of this manual. During the course of the Speedbase problem investigation, several versions of SVC-61 have been produced that internally mapped "stream i/o" operations into "raw i/o" operations. The SVC-61 "stream i/o" opcodes affected (see the Global File Converters Manual for full details) are:

19 fopen
22 fclose
28 fread
29 fwrite
30 fseek
56 fopen (*sic*)

The c language source for SVC-61 is module vc0sh.c. The following versions of the vc0sh.c source include the "stream i/o" to "raw i/o" mapping by default:

2.48
2.49
2.50
2.53

The following versions of the vc0sh.c source only perform the "stream i/o" to "raw i/o" mapping if the GSM_VC0SH_XLAT Unix shell variable is defined:

2.54

The following versions of the vc0sh.c source include the "stream i/o" to "raw i/o" mapping by default, unless the GSM_VC0SH_NOXL Unix shell variable is defined:

2.55 (and later)

If "stream i/o" to "raw i/o" mapping is enabled, the stream i/o operations are still available by adding 200 to the relevant opcode listed above (e.g. DSFUNC = 219 will always execute the stream *fopen* operation).

In all cases a special set of "hybrid" SVC-61 operations are available. These hybrid operations execute the relevant "raw Unix i/o" functions but expect the "stream i/o" parameters in the DS-block. The opcodes (i.e. DSFUNC values) for the hybrid operations are formed by adding 100 to any of the opcodes listed above (e.g. DSFUNC = 119 will always execute the raw *open* operation but expects the DS-block parameters to be set up for the stream *fopen* operation).

The version of *vc0sh.c*, or indeed any Global System Manager (Unix) source file, may be determined by inspecting the $GLDIR/sys/version file (see section 6.1.1.4).

G.101 Unix resources (open files and lock entries)

In common with other Unix applications, Global System Manager is very dependent on the availability of sufficient Unix resources for its requirements. This section describes some problems that have been encountered with one particular Unix resource, namely "open files", with particular reference to SCO Unix and the Universal Channel Interface (UCI) used by Speedbase Presentation Manager to access C-ISAM format Speedbase databases.

The Unix kernel parameter NOFILES specifies the maximum number of open files per process. On most versions of Unix (e.g. SCO Unix V3.2.x) the value of this "Tunable System Parameter" can range between 60 and 100 (although on SCO Unix, attempts to increase the value beyond the default of 60 have not always been successful - a problem that is under investigation). We have also discovered that the error returned by Unix functions (e.g. open) when the open file limit has been reached is not always the expected EMFILE (too many open files) error (this problem is also under investigation). On SCO Unix, the value of the NOFILES (and all the Tunable System Parameters can be displayed by the *sysdef*(ADM) command. For example:

```
# sysdef | qrep "NOFILES"
```

The implications of this restriction for Global System Manager (Unix) is that a single process (e.g. *global*, *glintd* etc.) cannot open more than 60 files. This limit is effectively 57 (i.e. 60 - 3) allowing for the 3 standard Unix channels (stdin, stdout, stderr). Of all the Global System Manager processes (see Appendix D) only the *glintd* process opens more than a few Unix files simultaneously (e.g. the Discrete Data File controller within *glintd* maintains a pool of 10 open files), both the SPOOLED and DIRECT printer controllers keep files open while printing is in progress.

In general, the number of files currently opened by a glintd process is very difficult to predict (diagnostic software **may** be included in a future version of Global System Manager to allow the actual number of Unix files opened by a particular glintd process to be determined).

The UCI also maintains a number of open files (see Chapter 7 of the Global File Converters Manual). To restrict the maximum number of files opened simultaneously by the UCI, two parameters (i.e. "Max. no. of UCI opens" and "Max. no. of UCI/RS open") may be defined in the Global configuration file (see section 9.7). Although increasing the value of the "Max. no. of UCI opens" may increase some aspects of Speedbase C-ISAM processing (e.g. rebuilding C-ISAM format Speedbase databases), you are advised against increasing the value of this parameter as unpredictable results, including database corruption, may occur if any UCI open operations fail because of insufficient Unix resources.

In addition to the NOFILES kernel parameter, the NFILE parameter specifies the maximum number of open files for the entire Unix system (see your Unix System Administrator's Guide for further details). On SCO Unix, the *sar*(ADM) command can be used to obtains details of the resource usage on your system. For example:

```
# sar -v
```

The FLCKREC kernel parameter specifies the number of records that can be locked by the system. Although the Global System Manager file locking mechanism (e.g. the Global Cobol LOCK verb) does not use the Unix lock tables, the Speedbase UCI does make extensive use of Unix locks. Note also that a few Unix lock table entries are used for internal sequencing during Global System Manager initiation. Experience has shown that with the FLCKREC set to 100, approximately 6 users can access a C-ISAM format Speedbase database concurrently. Thus, if more than 6 users are accessing a C-ISAM format Speedbase database simultaneously, the FLCKREC parameters must be greater than 100.

The sar(ADM) displays details of the number of Unix locks that have been used and the total number available. For example:

```
# sar -v
```

If a Global System Manager (Unix) installation is suffering from obscure problems, especially if C-ISAM format Speedbase databases are being accessed, the three kernel parameters described in this section (i.e. NOFILES, NFILE and FLCKREC) should be examined to ensure that sufficient Unix resources are available for Global System Manager.

G.102 Details of Global System Manager (Unix) source versions

Brief details of all the revisions to the various source files that comprise the Global System Manager (Unix) BACNAT software are available on the BBS, as file UNIXSRCV.ZIP.

G.103 Contents of the $GLDIR/sys/spool Directory

The $GLDIR/sys/spool directory, which is briefly described in section 6.1.1.8, contains the files described in the following sections when the Global System Manager configuration includes one, or more, SPOOLED printers.

G.103.1 The $GLDIR/sys/spool/gl_LCK_002 File

This file is used to lock resources between the glintd and glspod daemon processes. The structure of this lock file is beyond the scope of this manual.

G.103.2 The $GLDIR/sys/spool/queue File

This text file is used to pass commands from the glintd daemon to the glspod daemon. Each record in this file contains the following structure:

```
number print_file
```

where *number* is a printer unit number (e.g. 510) and *print_file* is the name of a temporary print file (normally) in the $GLDIR/spool directory.

When the glspod spooler daemon has expedited the relevant Unix command, using the number to index an entry in the Systems file data and the *print_file* name as an argument to the command, the first character of the record in the queue file is set to a "p" to mark the record as being printed.

G.103.3 The $GLDIR/sys/spool/log File

This text file is the equivalent of the $GLDIR/sys/messages file (see section 6.1.1.3) for the glspod spooler daemon. This background process writes messages (error, warning and diagnostic messages) to the log file whenever a significant event occurs that affects the Global

System Manager spooler sub-system. For example, a record is written to this log file every time a Unix spooler command is executed. Each message includes the current Unix date and time. This text file should be examined if you are experiencing problems with a Global System Manager (Unix) SPOOLED printer.

Unlike the $GLDIR/sys/messages file, it is NOT possible to suppress writes to the $GLDIR/sys/spool/log file. Consequently, this log file should be purged (i.e. deleted) on a regular basis by the Unix System Administrator. **Warning:** If the spooler log file is deleted whilst the *glspod* process is running, all further writes to the log file will be suppressed until Global System Manager is unloaded then restarted (i.e. until a new instance of the glspod daemon is running).

G.104 Producing a Unix core Dump

By default, the Global System Manager glintd process captures all Unix signals. Those signals that are unexpected and cause glintd to die are displayed in a message of the form:

```
glintd: Error 1217 - Process terminated due to signal SIGxxxx
```

Occasionally, in order to investigate a problem that results in an Error 1217, a special BACNAT will be produced by compiling the various sources that comprise Global System Manager (Unix) with the debug option of the Unix c compiler. The special BACNAT will be recognised by the inclusion of the phrase "DEBUG" in the familiar message:

```
Configuring Global System Manager (Bnnnn Vn.nnn); please wait
```

For example:

```
Configuring Global System Manager (SCODEBUG V3.188); please wait
```

The global -c option must be used to run Global System Manager in a manner that will result in the production of a Unix core dump with a special "debug BACNAT" under the conditions that would normally result in an Error 1217 with a standard BACNAT.

THIS OPTION SHOULD ONLY BE USED WITH A SPECIAL DEBUG BACNAT AND ONLY IF YOU BEEN INSTRUCTED TO DO SO BY THE SERVICE CENTRE.

In order to check the "debug BACNAT" has been installed correctly and the global -c option has been recognised a special (and extremely dangerous) program, CRASH, is available (from the Service Centre) which will cause a SIGSEGV with a standard BACNAT; or the production of a Unix core dump, if the global -c option has been used with a "debug BACNAT". To protect against injudicious use, the CRASH utility will only produce the desired effect (i.e. a SIGSEGV or a Unix core dump) if the special GSM_CRASH Unix shell variable is set, to any value, and exported. For example, if you are using the Bourne shell:

```
GSM_CRASH="YES";export GSM_CRASH
```

THIS SHELL VARIABLE SHOULD ONLY BE SET "BY HAND" IN ORDER TO TEST THE CRASH UTILITY AND SHOULD NEVER APPEAR IN A LOGIN SCRIPT.

G.105 GLDIR must be an absolute pathname

The Unix directory pathname defined by the GLDIR shell variable (see section 8.1.1) MUST be an absolute pathname. for example:

```
GLDIR=/usr/gsm81/global
```

The GLDIR shell variable must NOT be a relative pathname. For example:

```
GLDIR=usr/gsm81/global
```

If GLDIR is defined as a relative pathname, then it will only be possible to run Global System Manager when the current directory is the root directory. However, although Global System Manager may appear to work when invoked from the root directory, some facilities (e.g. glconfig) may not function correctly and may produce unexpected messages of the form:

```
specific error message
(can't open error header file "sys/errmsg/hdr")
```

G.106 New global -l option to switch off glspod messages

The global program distributed with BACNAT variant V3.206, and later, recognises the "-l" command line option. When this option is specified, the majority of the messages written to the $GLDIR/sys/spool/log file by the glspod daemon are suppressed. Without this new option the $GLDIR/sys/spool/log file may grow very large if the spooled printer interface is used.

G.107 New global -g option to keep glspad running

The *global* program distributed with BACNAT variant V3.206, and later, recognises the "-g" command line option. When this option is specified, the *glspad* daemon process, which is only used when the first user runs Global System Manager, is not killed once it has been used to initiate the other daemon processes (e.g. *gltimd*, *glspod*, *glprid*). This option is required to enable the special *glspod* retry mechanism described in section G.109.

G.108 glshmdump displays global command line arguments

The glshmdump utility distributed with BACNAT variant V3.206, and later, displays the "Run-time flags" as a hexadecimal value. One bit in this flag is allocated for each of the possible global command line options (e.g. -a = 0x01; -b = 0x02 etc.). For example, if the "global -d" option is used, glshmdump will report the following:

```
Run-time flags = 0x8
```

G.109 New glspod retry mechanism

On some AIX V4.1 configurations, the Global System Manager spooled printers were found to be stopping for no apparent reason. An investigation showed that the problem was caused by the Unix *system*() function returning an error when invoked by the *glspod* daemon to execute the spooler command derived from the Systems file. The cause of the problem could not be found and the simplistic retry mechanism in *glspod* (i.e. retry the command a number of times) did not provide a work-around.

The *glspod* daemon distributed with BACNAT variant V3.206, and later, contains a more sophisticated retry mechanism which appears to provide a work-around for this intractable problem. If the simple retry mechanism fails to initiate the *system*() function successfully, *glspod* will save the current Unix command (e.g. lp *filename*) in the file $GLDIR/sys/spool/retry. The *glspod* daemon will then invoke another copy of itself (using the *glspad* spawner daemon - see section G.107). The new invocation of glspod will commence by retrying the original command before considering any more outstanding spooled printer requests. This new retry

mechanism is enabled automatically but can only function if the global -g option (see section G.107) has been used to leave the glspad daemon running (i.e. in the global -g option is the switch that enables this new retry mechanism). The implementation of this new retry mechanism has resulted in the creation of several new diagnostic and error messages (see section G.113).

G.110 New global -a option to re-attach to existing glintd

If an attempt is made to invoke global while the required glintd process is already running a fatal error 1029 (see section C.1) will appear. This problem will occur, for example, if a terminal emulator session has crashed resulting in another login with a different pseudo-tty name.

The global program distributed with BACNAT variant V3.207, and later, recognises the "-a" command line option. When this option is specified, global will attempt to re-attach to the required glintd process instead of return the error code 1029.

Note that this option must be used with great care: The back-end glintd daemon has no reference to the front-end global that is attached. Consequently, the global -a option can be used to take over an existing user, leaving the global process for that user unattached to any background glintd process.

G.111 Unix result code when global exits

The global program distributed with BACNAT variant V3.207, and later, returns to the Unix shell using an "exit(1)" whenever a fatal error is detected and an "error(0)" whenever exiting normally (e.g. after $BYE has been run). Earlier versions of global returned an unpredictable result code to the Unix shell. This new consistent behaviour can be used to write scripts of the following form:

```
if (global)
then
      echo "Global exited normally"
else
      # Retry the global command with more options, for example
      echo "First attempt to load global failed, retrying..."
      global -a- -d
fi
```

G.112 More diagnostic messages written to $GLDIR/sys/messages

The following new glintd error messages have been created (see section C.2):

glintd: Diagnostics 1238 - Diskette seek error, address *x*, offset *y*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3) if the global "-d" option (see section 6.3.3.12) has been used. It indicates that the lseek() function has returned an error to the diskette controller.

glintd: Diagnostics 1239 - Diskette read/write hard error, address *x*, reqlen *y*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3) if the global "-d" option (see section 6.3.3.12) has been used. It indicates that either the read() or write() function has returned an error to the diskette controller.

glintd: Diagnostics 1240 - Diskette read/write soft error, address *x*, reqlen *y*, actlen *y*

This diagnostic message will only appear as a log-record written to the file $GLDIR/sys/messages (see section 6.1.1.3) if the global "-d" option (see section 6.3.3.12) has been used. It indicates that either the read() or write() function has returned an actual transfer length that is different from the required transfer length.

G.113 More diagnostic messages written to $GLDIR/sys/spool/log

The following new glspod error messages have been created (see section C.7):

glspod: Diagnostics 2024 - Command retry mode enabled

This message confirms that glspod has detected the "global -g" option. The spooler command retry mode has been enabled (see sections G.107 and G.109).

glspod: Diagnostics 2025 - Retry file detected, previous command will be retried

The glspod initialisation code has detected the presence of the spooler retry file:

```
$GLDIR/sys/spool/retry
```

The previous spooler command (that failed when a previous instance of glspod was running), obtained from the retry file, will be retried immediately.

glspod: Diagnostics 2026 - Retry file opened OK

See message 2025. The $GLDIR/sys/spool/retry file has been opened successfully.

glspod: Diagnostics 2027 - Retry file read OK, command = %s, retry count = %s

See message 2025. The previous (failed) spooler command has been extracted from the $GLDIR/sys/spool/retry file.

glspod: Diagnostics 2028 - Retry file read failed (command retry aborted)

See message 2025. The $GLDIR/sys/spool/retry file has been detected but the previous (failed) spooler command could not be read from the file. The command retry mode has been aborted - glspod will continue normally.

glspod: Diagnostics 2029 - Retry file open failed (command retry aborted)

See message 2025. The $GLDIR/sys/spool/retry file has been detected but could not be opened. The command retry mode has been aborted - glspod will continue normally.

glspod: Diagnostics 2030 - Command failed, retry mode in progress

The current spooler command has failed (even after a small number of internal retries). The command will be written to the $GLDIR/sys/spool/retry file, then glspod will terminate, invoking another copy of itself (i.e. retry mode will be enabled).

glspod: Diagnostics 2031 - Retry file created OK

See message 2031. The $GLDIR/sys/spool/retry file has been created successfully.

glspod: Diagnostics 2032 - Retry file create failed (command retry aborted)

See message 2031. The $GLDIR/sys/spool/retry file could not be created. Command retry mode has been aborted.

glspod: Diagnostics 2033 - Exiting glspod, awaiting new invocation

See message 2031. The current glspod is about to invoke a new copy of itself (before closing down).

glspod: Diagnostics 2034 - New glspod exec failed [%s,%s] (command retry aborted)

This message is reserved for future use.

glspod: Diagnostics 2035 - %s spoolnext command was successful:\n%s

The spooler command was successful. For a correctly functioning spooler sub-system, this message should appear at the conclusion of every print command (i.e. the spooler log file should consist of matching pairs of 2012 and 2035 messages).

glspod: Diagnostics 2036 - %s spoolnext retry counter exhausted

The $GLDIR/sys/spool/retry file includes an internal retry counter. The spooler command has been retried a number of times but has failed on every attempt. There is no point continuing with the command - retry mode (for the current printer command only) has been abandoned.

glspod: Diagnostics 2037 - %s spoolnext retry not possible (no glspad running)

The previous spooler command failed and should be retried using the $GLDIR/sys/spool/retry file. However, the *glspad* daemon is not running because the "global -g" option was not used to invoke Global System Manager. The absence of the *glspad* (spawner daemon) has prevented *glspod* from invoking another copy of itself.

G.114 GSM (Unix) loader uses the installed subvolume number

When the $F INS instruction is used during the installation of GSM (Unix) the subvolume number of the SYSRES volume (i.e. normally 201 aka A01) is patched into the Domain Header File (e.g. SVL00_SYSDOM). This subvolume number is used by the GSM (Unix) loader when initiating the installed system. Unlike some other implementations of Global System Manager (e.g. GSM (MS-DOS)) it is NOT possible to copy another SYSRES subvolume to an SSD directory and load Global System Manager from it, normally for testing purposes, unless the subvolume number matches the number that was patched into the SVL00_DOMAIN file when Global System Manager was installed. For example, assuming the following files in the A00.dir directory:

```
SVL00_SYSDOM                    * Domain Header File
SVL01_SYSRES                    * Installed SYSRES
```

```
SVL02_SYSRES                          * Test SYSRES
```

the following technique (involving just a single file rename), to attempt to load from the SVL02_SYSRES volume, will not work:

```
cd $GLDIR/data/A00.dir                * Change to SSD directory
mv SVL01_SYSRES XXX01_SYSRES          * Deactivate 201/A01 SYSRES
```

Instead, the following additional rename must be performed:

```
mv SVL02_SYSRES SVL01_SYSRES          * Rename to "bootstrap volume"
```

G.115 GSM and Unix filing systems 2Gb, and larger

The size of a Global System Manager Discrete Data File simulated volume is limited to 31-bits (i.e. 2Gb-1). If a Global System Manager (Unix) configuration, with a BACNAT variant less than V3.212, is installed on a Unix filing system that is 2Gb, or larger, various commands will fail with a variety of symptoms. For example, $U will fail with a PGM CHK-11; $F will return a "WRONG FORMAT VOLUME" error.

The Discrete Data File controller in BACNAT variants V3.212, or later, will truncate the effective size of the Unix filing system (i.e. the size returned to Global System Manager, not the real size!) to 2Gb (actually 2Gb - 8Kb). During the initiation of Global System Manager, for every domain that is truncated to 2Gb approx, the following warning message will be displayed on the screen and in the $GLDIR/sys/messages file:

```
glintd: Warning 1241 - Unix filing system is larger than 2Gb
pathname_of_Unix_domain_header_file
```

If the amount of free space on a Global System Manager Discrete Data File simulated volume is 2Gb, or larger, the amount of free space returned by the Discrete Data File controller in the BACNAT nucleus will be 0 rather then the true amount. This will result in utilities such as $F and $V reporting a SPARE SPACE of 0 bytes. This effect will also make sub-volume allocation, and thus installation, impossible because any attempt to allocate a sub-volume will result in an immediate INSUFFICIENT SPACE error.

G.116 Importing Data while Global System Manager is running

Section G.98 describes the problems that may be encountered when attempting to import data using a non Global utility (e.g. tar, cpio, cp, rcp, ftp) "on the fly" while Global System Manager is running. To summarise the recommendations in section G.98:

Copy data into	**Action needed to ensure data consistency**
Virtual diskette	Wait 1 second: The Unix device or file is automatically closed after 1 second of inactivity.
DLV	Wait 1 second: The Unix file is automatically closed after 1 second of inactivity.
Integrated Data File	Reload Global System Manager completely otherwise an attempt to access the domain may result in data corruption.
Discrete Data File	Reload Global System Manager completely otherwise an attempt to access the sub-

	volume may result in data corruption.

It is not always convenient to restore data into a specially prepared DLV (e.g. if the data to be restored is a series of sub-volume files from an end-users computer). This section describes a technique that can be used to import sub-volume files from a customer's tape, for example, to a running, multi-user Global System Manager (Unix) configuration without the requirement to re-load Global System Manager.

In addition to the standard GSM configuration which, for the purposes of this example, is assumed to be installed in directory /usr/gsm, a separate "feeder" configuration should be installed into a completely different directory (e.g. /usr/gsmio). The GLIPCBASE variable (see section G.53) for the user on the "feeder system" should be defined, to a value other than "GSM0" (e.g. "AAA0"). The Systems file for the "feeder system" should contain an entry for a single user, a Discrete Data File directory within the /usr/gsmio directory and a DLV (or DLV's) (see section G.90) that is shared with the "master system". For example:

```
SYSTEM A
DATA 0 /usr/gsmio/A00.dir
DISKETTE 1 Z155 /usr/gsm/data/Z155
```

The Systems file for the "master system" should also include anentry for the shared DLV. For example:

```
SYSTEM X
DISKETTE 1 Z155 /usr/gsm/data/Z155
```

DO NOT BE TEMPTED TO SHARE "DATA" ENTRIES BETWEEN THE TWO SystemsFILES OTHERWISE DATA CORRUPTION WILL OCCUR.

Perform the following steps to import a sub-volume file into the "master system" from a customer's tape, for example:

1. Exit GSM from the "feeder system". This will only affect the "special user" with a GLIPCBASE of "AAA0";

2. Restore the sub-volume(s) into the /usr/gsmio/data/A00.dir subvolume, taking the normal precautions (see section G.13 and G.62);

3. Reload GSM on the "feeder system" from the "special user" with a GLIPCBASE of "AAA0". This user should be able to access the freshly imported sub-volume(s);

4. Use $F, for example, to copy the Global files from required subvolumes to the shared DLV;

5. Once the $F copy has been completed wait 1 second before accessing the data in the DLV from a "normal user" on the multi-user configuration.

G.117 The Unix nice command and Global System Manager

Although it is not possible to alter the Unix priority of a running *glintd* process, the Unix *nice* command can be used to alter the priority of a Global System Manager system when the global command is invoked. For example:

```
nice -5 global
```

Please consult your Unix manual for a full description of the *nice* command. Note that the extended options of the ps command detail the nice value for a running process.

The following illustrative Bourne shell script may be used to give individual operators the chance to set the nice value for their global process:

```
# script glnice - run global with optional nice value
# usage: glnice - run global without a nice option
# glnice nn - run global with nice option nn
# no parameter validation is performed
#
if [ $# -eq 0 ]
then
      echo "Calling global without a nice value..."
      global
else
      if [ $# -eq 1 ]
      then
            echo "Calling global with a nice value of $1..."
            nice -$1 global
      else
            echo "usage: glnice [nn]"
      fi
fi
```

G.118 New glshmdump -x and -z options

The glshmdump utility distributed with BACNAT variant V3.218, and later, includes the following new features:

- both the physical and logical addresses of Global System Manager Shared Memory are displayed;

- the -z option is recognised. This option informs glshmdump to attempt to dump Shared Memory even if there is an IPC inconsistency;

- the -x option is recognised. This option produces a complete hexadecimal dump of Shared Memory.

G.119 Ownership of spooler files

If the Owner and Group of the $GLDIR/sys/spool/gl_LCK_002 are not both "global" (e.g. if the Owner is "root"; and the Group as "other") then the glintd process will suffer a SIGSEGV error when an attempt is made to use the spooled printer interface.

G.120 New glreorg -c option to create a Sub-Volume (SVL) file

The glreorg utility distributed with BACNAT variant V3.219, or later, includes the following new features:

- the -c option is now recognised. This option allows a new Sub-Volume file to be created with an empty Global directory. For example:

  ```
  glreorg -c -s 10m SVL01_SYSRES
  ```

  will create a 10Mb sub-volume SYSRES.

This option is expected to be particularly useful in those circumstances where it is not possible to create a new sub-volume using the standard method (i.e. $V Initialise). For example, when the amount of free space on the filing system is 2Gb, or more (see section G.115).

- the size argument following the -s option can either be separated by a space (i.e. follow as the next command line argument) or appear immediately after the "-s" string. For example:

```
glreorg -s 10M SVL02_WORK
```
or:
```
glreorg -s10M SVL02_WORK
```

G.121 SCO Unix tape device minor numbers

The minor numbering scheme for SCO Unix SCSI tape devices is as follows:

7 6 5 4 3 2 1 0	*Description*	*Minor number*
x x x	Unused (reserved)	
x	High density (6250 BPI)	16
x	No rewind on close	8
x	No unload on close	4
x x	Tape unit number	0 - 3

G.122 Preventing tape ejection on close on SCO Unix

Some users have experienced a problem when using a SCSI DAT (4mm) or SCSI Exabyte (8mm) tape drive: the tape is ejected after each command.

In order to prevent the SCSI DAT or SCSI Exabyte tape drive from ejecting the tape cartridge after each tape command, the "No unload on close" device must be used (see section G.121). This is achieved as follows:

1. Determine the major number of the tape device by doing a long listing on /dev/rct0:

```
# l /dev/rct0
crw-rw-rw 2 root root 46, 0 Dec 9 9:40 /dev/rct0
```

   In this example, 46 is the major device number; 0 is the minor device number.

   The major number for the device might be different because it is dynamically allocated when the tape drive is installed.

2. Create a new tape device that will not eject the tape:

```
# mknod /dev/noeject c 46 4
```

   In this example, "noeject" is the new device name; 46 is the major device number; 4 is the minor device number.

   The major number should be the same as /dev/rct0.

3. Either make the no-eject device the default device. Perform this by logging in as root and executing the following commands:

```
# mv /dev/rct0 /dev/eject
# ln /dev/noeject /dev/rct0
```

or simply set the GLTAPE variable to the "no eject" device:

```
GLTAPE=/dev/noeject; export GLTAPE
```

G.123 Disk Cacheing and Global System Manager (Unix)

Disk cacheing is NOT supported on Global System Manager (Unix). There is no benefit supporting Global System Manager disk cacheing on a virtual memory operating system such as Unix. Firstly, the Unix cacheing algorithm is considerably more efficient than Global System Manager cacheing. Secondly, the virtual memory allocated by a Global System Manager disk cache may be swapped to hard-disk by the Unix virtual memory handler!

G.124 RAM Disk and Global System Manager (Unix)

A RAM Disk is NOT supported on Global System Manager (Unix). There is no benefit supporting a RAM Disk on a virtual memory operating system such as Unix: The virtual memory allocated by a Global System Manager RAM Disk may be swapped to hard-disk by the Unix virtual memory handler!

G.125 No =.55*nn* supplied with Global System Manager (Unix)

The =.55*nn* customisation utility supplied with GSM (BOS), GSM (MS-DOS and Windows) and GSM (Novell NetWare) configurations is used mainly to update controller specific information in the Global configuration file without recourse to Global Configurator. The configuration file on Global System Manager (Unix) configurations contains considerably less controller specific information than in any of the "real-mode" configurations listed above because most of these details are held in the Unix Systems file (see Chapter 7). This gradual move away from the Global configuration file to the Unix Systems file is likely to continue. The heavy use of the Systems file makes a =.55*nn* customisation utility (i.e. =.5527) inappropriate for Global System Manager (Unix) configurations.

Appendix H - SVC 70 Interface

Global System Manager (Unix) contains a special interface, SVC 70, that may be used by application programmers to execute Unix shell commands from within Global System Manager. The %.SHELL (see section 5.2) and %.SHCMD (see section 5.1) utilities both use this SVC.

The call to SVC 70 is coded as follows:

SVC 70 USING $$SCNN *buff prompt result* $$SUSP *suspflag*

where: $$SCNN is the Screen Number as described in the Global Development System Subroutines Manual;

buff is a character buffer containing the command string which **must be terminated by a byte of binary-zero;**

prompt is a PIC 9(2) COMP variable which can be set to the following values:

0 display "[Key return to continue]" before refreshing screen when the Unix shell returns control to Global System Manager;

1 refresh the screen immediately when the Unix shell returns control to Global System Manager;

result is a PIC 9(2) COMP variable which reflects the result of the Unix shell command;

$$SUSP is the Suspend Flag as described in the Global Development System Subroutines Manual;

suspflag is a PIC 9(2) COMP variable that MUST contain LOW-VALUES when SVC 70 is called. It will be set to 1 by SVC 70 when the Unix command completes.

SVC 70 will ALWAYS return an exception. $$COND must be tested to determine the result of the call:

0 Obsolete version of SVC 70 used (this should never occur);

1 Wrong number of parameters;

2 SVC 70 already in use by another partition on this SYSTEM;

3 Success. The Unix shell command has been executed. Wait for the suspflag to be set;

4 The command string was too long.

The call to SVC 70 should be coded as follows:

```
*
* Establish the desired value in prompt
* Establish the Unix command string in buff
*
MOVE 0 to suspflag
SVC 70 USING $$SCNN buff prompt result $$SUSP suspflag
ON NO EXCEPTION
*
* Obsolete version of SVC 70
*
ELSE
      IF $$COND = 1
*
* Wrong number of parameters
*
      END
      IF $$COND = 2
*
* SVC 70 in use by another partition on the SYSTEM
*
      END
      IF $$COND = 4
*
* The Unix command string was too long
*
      END
      IF $$COND = 3 * Unix command in progress
            DO UNTIL suspflag NOT ZERO
                  SUSPEND 60
            ENDDO
            IF result ZERO
*
* The Unix command completed successfully
*
            ELSE
*
* The Unix command failed
*
            END
      END
END
```