Global
Speedbase Presentation Manual
Version 8.1

MPMMV8.1/60

TABLE OF CONTENTS

APPENDICES

0. Foreword

This manual details the operation of the Speedbase Presentation Manager and supplements the extensive on-screen help displays.

Chapter One serves as an introduction to Speedbase, its database management system, window controller and the use of the keyboard and screen.

All Speedbase applications handle the screen and keyboard in the same easy-to-use way. So once you are familiar with one Speedbase product you should be able to use any other without difficulty, whoever it was written by. Chapter Two of this manual describes the use of the screen and keyboard in detail.

Chapters Three to Seven describe how to use the Speedbase utilities. You will need to know how to do a backup of your data and how to re-generate your database should this be required if, for example, you need more space than anticipated for your data. You may also need to rebuild the database from time to time, to improve performance or following an error in an application program. The status utility shows how much space is available for each record type on the database.

Speedbase will from time to time detect an error condition while you are running an application or one of the utilities. For example, you might attempt to delete a record that may not be deleted for some reason. Or you might try to backup the database while another user is updating it, which is not allowed. When such a situation occurs you will see an error message on the last line of the screen, known as the baseline. These messages are often self-explanatory and are documented in Appendix A.

Speedbase may also detect an error condition which should have been detected and handled by your application program. When this happens an EXIT code is displayed at the baseline. Speedbase may also detect an error from which your application program cannot recover and in this case displays a STOP code at the baseline. These codes are documented in Appendix B and should be used to assist your software supplier in resolving the error condition for you.

The Speedbase Presentation Manager is one of the most advanced products of its type available, yet is easy to use. Most of the information you need to operate your Speedbase application may be displayed on your screen at any time by keying Help. We do urge you to make good use of the Help key, particularly when learning to use a Speedbase application for the first time.

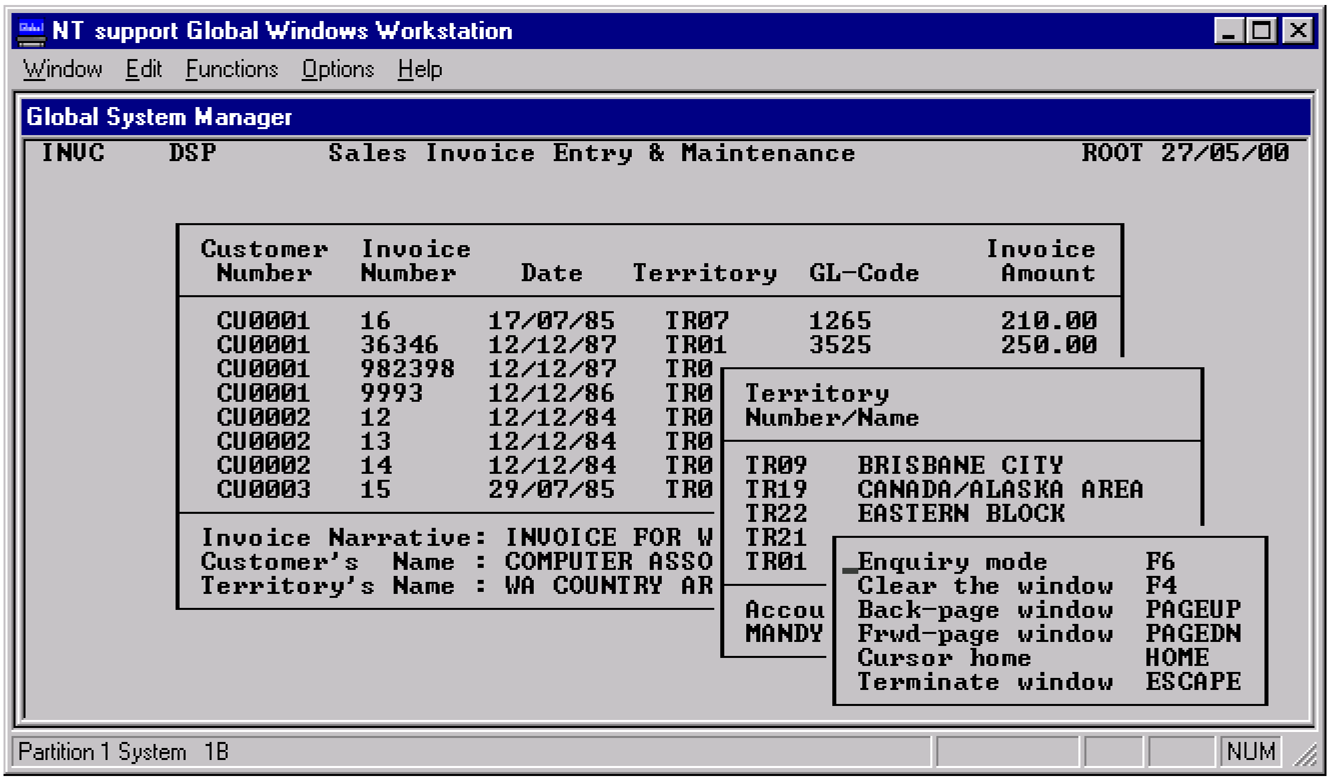
1. Presentation Manager Overview

The Speedbase Presentation Manager is the environment in which all Speedbase application programs operate. It provides what has come to be known as the user interface to the applications and controls every aspect of your interaction with them. It includes a highly sophisticated multi-user database management system to manage the storage of your data on disk.

1.1 The Screen and Keyboard

The most important aspect of the Speedbase Presentation Manager As far as the user is concerned is the use of the screen and keyboard. In its use of both PC monitors, such as the IBM PC, and of the serial terminals often used in Unix systems, such as the Wyse-60, the Speedbase Presentation Manager makes use of every available facility to provide the most sophisticated use of these diverse resources of any system currently available. Because of its great importance this topic is covered in full in the next chapter of this manual.

Figure 1.1a is an example of a typical Speedbase Presentation Manager window with an on-screen help window.

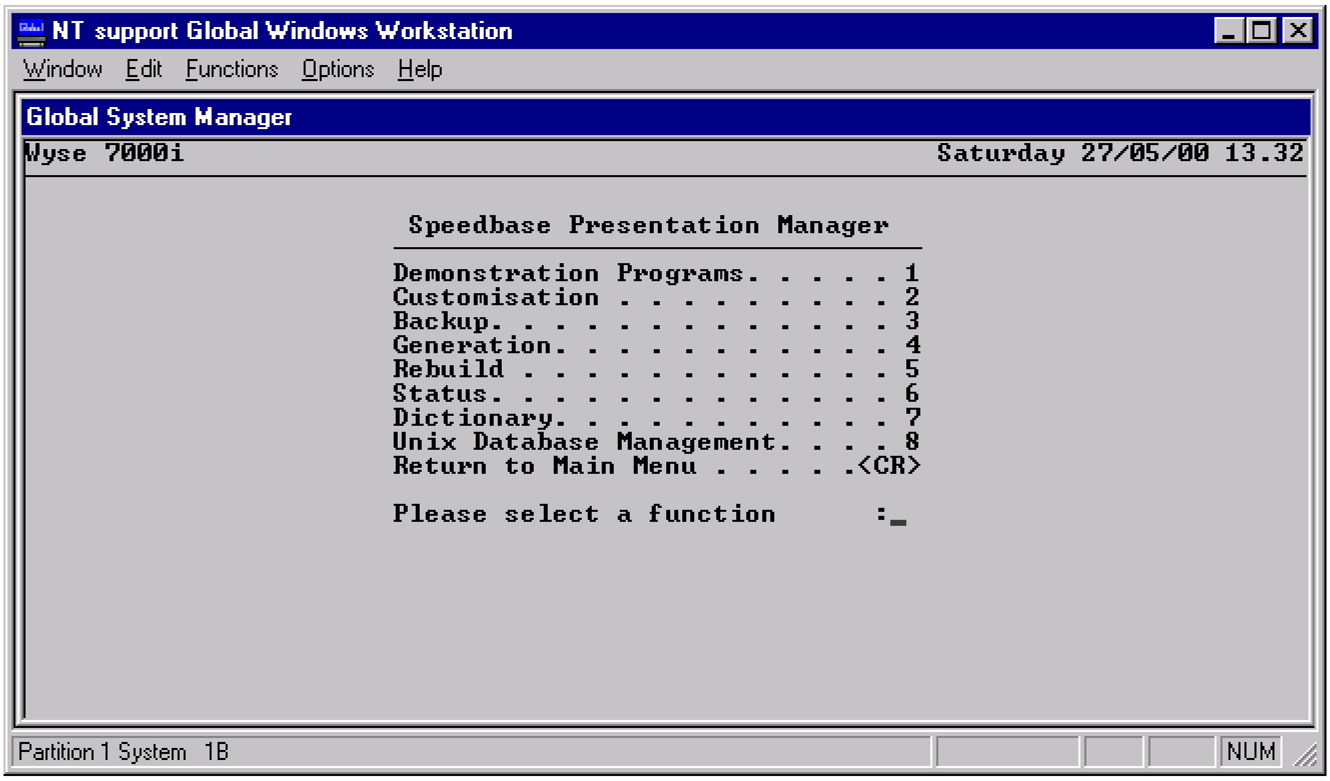

NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
NUM

```
 INVC     DSP         Sales Invoice Entry & Maintenance            ROOT 27/05/00


            Customer  Invoice                                    Invoice
             Number   Number     Date    Territory  GL-Code       Amount

             CU0001   16        17/07/85   TR07      1265          210.00
             CU0001   36346     12/12/87   TR01      3525          250.00
             CU0001   982398    12/12/87   TR0
             CU0001   9993      12/12/86   TR0  Territory
             CU0002   12        12/12/84   TR0  Number/Name
             CU0002   13        12/12/84   TR0
             CU0002   14        12/12/84   TR0  TR09   BRISBANE CITY
             CU0003   15        29/07/85   TR0  TR19   CANADA/ALASKA AREA
                                                TR22   EASTERN BLOCK
            Invoice Narrative: INVOICE FOR W    TR21
            Customer's  Name  : COMPUTER ASSO   TR01  Enquiry mode       F6
            Territory's Name  : WA COUNTRY AR         Clear the window   F4
                                                Accou Back-page window   PAGEUP
                                                MANDY Frwd-page window   PAGEDN
                                                      Cursor home        HOME
                                                      Terminate window   ESCAPE
```

Figure 1.1a - A Typical Window

1.2 Running the Speedbase Presentation Manager

The Presentation Manager installation job automatically creates a menu entry for Speedbase in your main system menu. When you key that option the Presentation Manager menu is displayed:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
NUM

```
Wyse 7000i                                             Saturday 27/05/00 13.32

                    Speedbase Presentation Manager

                    Demonstration Programs. . . . . 1
                    Customisation . . . . . . . . . 2
                    Backup. . . . . . . . . . . . . 3
                    Generation. . . . . . . . . . . 4
                    Rebuild . . . . . . . . . . . . 5
                    Status. . . . . . . . . . . . . 6
                    Dictionary. . . . . . . . . . . 7
                    Unix Database Management. . . . 8
                    Return to Main Menu . . . . .<CR>

                    Please select a function      :_
```

Figure 1.2a - The Presentation Manager Menu

The Presentation Manager menu allows you to run the demonstration programs and the utilities. The demonstration programs act as a useful introduction to the screen and keyboard facilities of the Presentation Manager, see section 2.1. The utilities are used to perform important functions such as database backup and are documented in later chapters of this manual. The chapter number for each utility is the same as the menu option number, so that, for example, the status utility is documented in chapter six.

Note that, most of the time when you are using the Presentation Manager it is acting as the user interface to your application programs. In this case you will actually use the menus and programs provided by your application program supplier. It may still be useful to refer to this manual from time to time, particularly with reference to the customisation and database utilities.

1.3 Printing Reports

Reports printed by Speedbase application programs may be directed to a disk unit for spooling, or may be printed directly, depending on the assignment of the logical printer unit, $PR. For example, if $PR is assigned to unit 500, your reports will be printed directly, on the printer with that unit-id. If on the other hand, $PR is assigned for example to unit 207, a spool unit, the report is written to that disk unit for later spooling.

Note that the assignment of the print unit, $PR, and of your logical disk units, is usually done in your various menus. When a report is written to disk rather than being printed directly it is assigned a name of the form:

```
    name-a
```

where name is the name of the program generating the report and a is a letter from A to Z. If $PR is assigned to a disk spool unit the letter used is always A. On other disk units the letter is changed each time the report is printed so that you can print several versions of the same report.

1.3.1 Using Special Stationery

Your Speedbase application programs may well make use of special stationery. An example would be an invoice form on which your company name and address and other details are pre-printed. When you do printing of this sort you will be prompted to mount the special stationery by a baseline prompt:

```
UNIT unit form-description
```

where unit is the printer unit-id followed by the description of the particular special stationery. For example, an invoice program might issue the message:

```
UNIT 500 Please mount invoice stationery
```

Once you have mounted the appropriate stationery, key <RET>. If you are unable to do so for any reason, key N. So that you can check the alignment, an alignment pattern is usually printed, followed by the prompt:

```
UNIT unit IS ALIGNMENT OK?
```

Key Y to continue, or adjust the stationery and key N so that another pattern is printed. Once the print using special stationery is complete you should replace your usual stationery and reply <RET> to the prompt:

```
UNIT unit REPLACE STANDARD STATIONERY
```

1.3.2 Dealing with Printing Problems

You may interrupt printing to deal with problems that may arise, such as paper jams, etc. When printing first commences the following prompt is usually displayed:

```
TYPE <CTRL G> TO HALT PRINTING:
```

Keying <CTRL G> will cause printing to be suspended and the following prompt is displayed:

```
RE-ASSIGN PRINTER?:
```

If you can resolve the problem simply key N and printing re-commences. Alternatively, you may re-assign the printer unit to another printer, or a disk or spool unit. To do so, key Y and the following prompt is displayed:

```
ENTER NEW PRINT UNIT ID:
```

You may now enter a new printer unit-id or key <BCK> to suppress further printing. In this case the following message is displayed:

```
OUTPUT WILL BE SUPPRESSED
```

Note that although the output is suppressed the print program still proceeds. This is to ensure that any other processing carried out by the same program, database updates for example, are completed correctly.

1.4 Database Structure and Capabilities

The Speedbase Presentation Manager includes a sophisticated data- base management system which controls the storage of all your data on disk. It is a truly multi-user system which means that it allows many users to display **and update** the database information at the same time without interfering with one another. It looks after such functions as indexing, which allows you to view your data in many different ways. It ensures that the relationships that exist between different record types are strictly maintained. For example, it would ensure that invoice records could not be added to the database without being assigned to a valid customer record.

Each database may contain up to thirty-six record types, each with up to eight million records. As you add, change or delete records Speedbase automatically updates the indexes involved, as each change takes place, thus ensuring that your database is always kept fully up to date.

The layout of each individual record type and the relationship of each to the others is defined in a data dictionary. This also contains details of the indexes used to access each record type. The data dictionary therefore specifies the form of the database, but not its size. If your Speedbase database is stored in the Global Speedbase format its size is fixed when it is created or modified using the generation utility described in chapter four.

If your Speedbase database is installed on a Unix system you have the option of storing your data in the Unix file structure rather than as a Global format Speedbase database. In this case the size of the database is adjusted automatically as data is added.

1.4.1 Global Database Structure

A Global format Speedbase database (i.e. one stored in the Global System Manager file structure rather than a Unix file structure) consists of a data dictionary file, a main index file and one, two, or three datafiles. The names of the files are as follows, where xxxxx is the database name:

File-id	**Description**
Dixxxxx	Data Dictionary
DBxxxxx	Main Index File
DBxxxxx 1	Data-file 1
DBxxxxx 2	Data-file 2 (optional)
DBxxxxx 3	Data-file 3 (optional)

Table 1.4.1 - Global Speedbase Database Files

The data dictionary is created by the application developer using the dictionary utility of the Speedbase Development System. The five character database name xxxxx is specified when the database is created or modified using the generation utility.

The main index file contains the indexes for all records on the database and is created automatically by the rebuild utility following database creation or modification by the generation utility. It consists mostly of index blocks, known as IDBs. These are initially all free when the database is first created and are allocated as records are added to the database. The status utility described in chapter six displays information about how many index blocks are free.

The three datafiles contain the data records. All the records of a particular record type are stored in one datafile. You may have only one datafile, in which case all the records of all your record types are stored in it. If you have two or three datafiles each stores all the records of particular record types.

For example, you might have all your customer records in datafile 1 and all your product records in datafile 2. The usual reason for having multiple datafiles is that each may be stored on a different disk volume, and these may be attached to different computers in a networked configuration. The data dictionary and main index file are stored together but may be on a separate disk volume from the datafiles.

1.4.2 Unix Database Structure

If your Speedbase database is stored in a Unix file system it has a special file structure. In this case the Speedbase database consists of a data dictionary file and a schema file, both stored in the usual Global file structure, and in addition, a special index file and several datafiles and index files stored in the Unix file structure. The names of the files are as follows, where xxxxx is the database name and rt1 to rtn are up to thirty-six record types:

File-id	Description	Format
DIxxxxx	Data Dictionary	Global
DBxxxxx	Schema File	Global
DBxxxxx	Special Index File	Unix
DBxxxxxrt1 .dat	C-ISAM Data File for record rt1	Unix
DBxxxxxrt1 .idx	C-ISAM Index File for record rt1	Unix
DBxxxxxrt2 .dat	C-ISAM Data File for record rt2	Unix
DBxxxxxrt2 .idx	C-ISAM Index File for record rt2	Unix
DBxxxxxrtn .dat	C-ISAM Data File for record rtn	Unix
DBxxxxxrtn .idx	C-ISAM Index File for record rtn	Unix

Table 1.4.2 - Unix Speedbase Database Files

The data dictionary is created by the application developer using the dictionary utility of the Speedbase Development System. The five character database name xxxxx is specified when the database is created using the Unix database management utility.

1.4.3 Database Generations

When a data dictionary is created using the dictionary utility of the Speedbase Development System it is assigned a generation number. Each time the dictionary structure is changed this number is incremented.

When your database is created from the data dictionary by the generation utility, the generation number is embedded in it. When your application programs run they check to see that your database has the correct generation number.

When a new version of your application programs is released it may require a new version of the database too. This is produced from the new generation of the data dictionary supplied as part of the new release, by the conversion option of the generation utility. If your database is stored in the Unix file structure your data is then transferred using the Unix database management utility.

1.4.4 Index Management

If your database is stored as a Global format Speedbase database, rather than the Unix file structure, it is important to ensure that there are always some index blocks free, to allow re- indexing to take place during normal database activity. Should the database manager run out of free index blocks it will cause your application program to be terminated and you will have to use the rebuild utility to correct the problem before you can proceed. To avoid this happening Speedbase warns you when less than twenty index blocks are free:

```
WARNING: INDEX BLOCKS FREE: n
```

where n is the actual number free. You should exit from your application program and use one of the utilities to increase the number of free index blocks. The quickest way to do this is to do a partial rebuild of the database with the rebuild utility. This optimises index block use and will usually free up sufficient for you to proceed. Alternatively you can use the generation utility to increase the size of the main index file. The status utility, described in chapter six, displays the number of free index blocks.

If your database is held in the Unix file structure this process is carried out automatically.

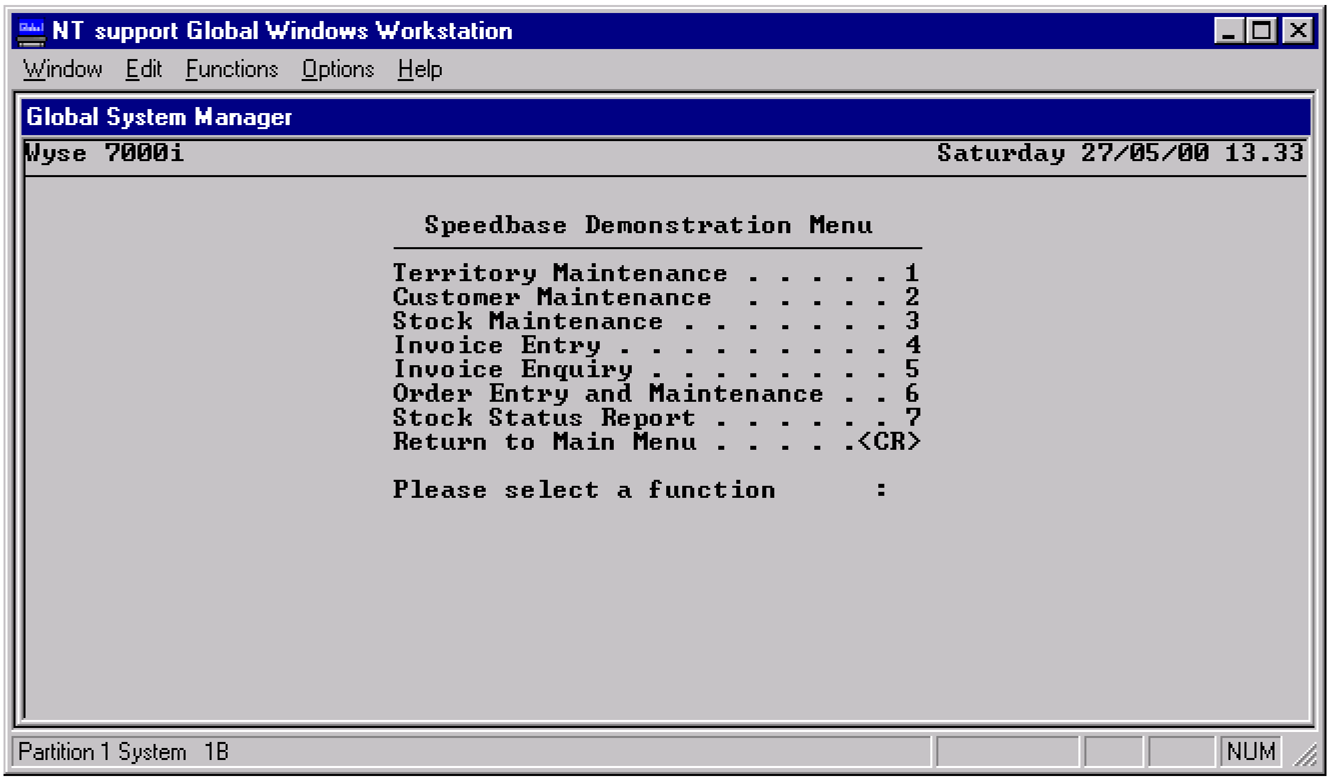
2. Using the Screen and Keyboard

Your Speedbase application programs make extensive use of screen and keyboard facilities provided by the Speedbase Presentation Manager. These facilities are what has come to be known as the User Interface to your application programs and affect virtually all the activities you are involved in running your computer. All applications have their unique features of course, but handle the screen and keyboard in the same efficient and consistent fashion.

This chapter describes the screen and keyboard facilities of Speedbase. You should find it helpful to use the examples given to assist you in becoming familiar with them. One of the major benefits of the Speedbase Presentation Manager is that it provides extensive on-screen help facilities. There are two levels of help available at all times and the first of these tells you what keyboard options you have whenever you are required to enter information. To see the help window, use your help key. See section 2.2 if you do not know which key to use for help.

2.1 Windows

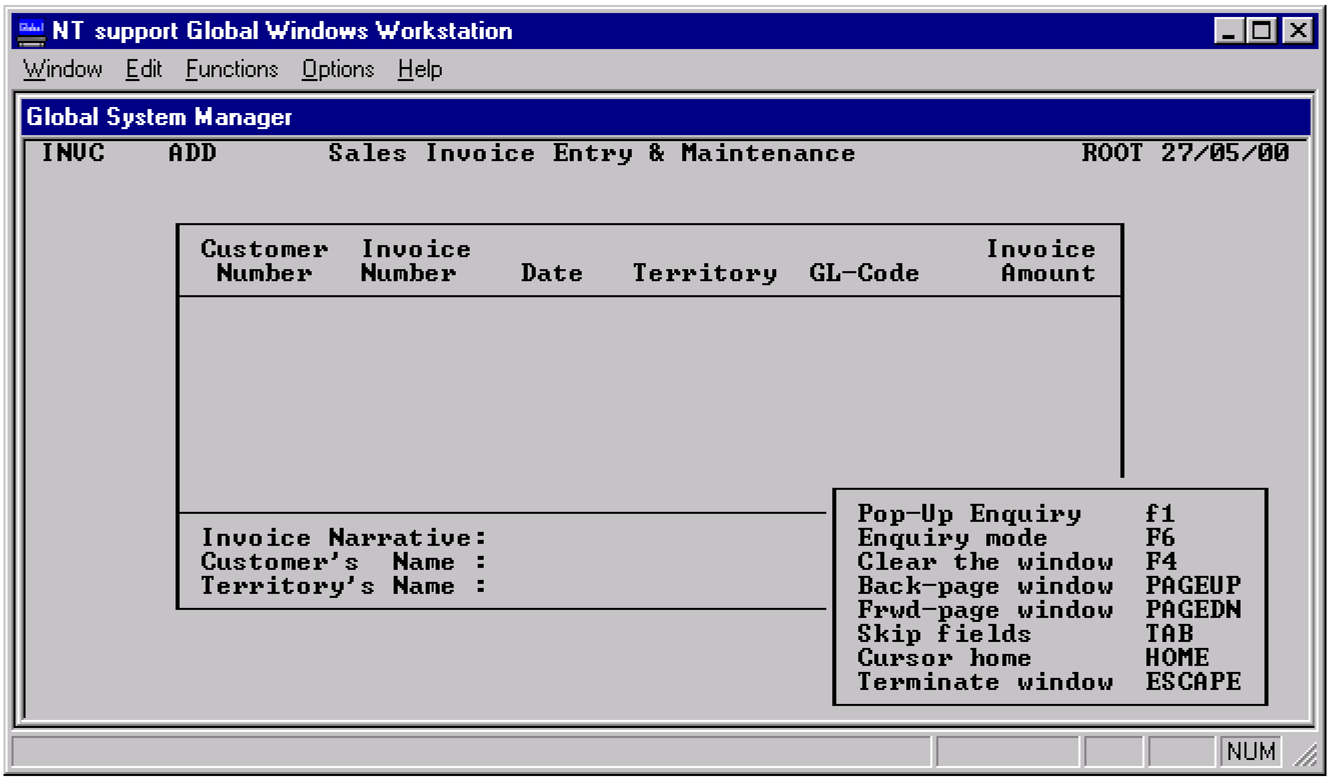
When Speedbase is installed it creates an entry in your main menu for the sample application. When you enter this menu option the following menu is displayed:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
NUM

```
Wyse 7000i                                       Saturday 27/05/00 13.33

                    Speedbase Demonstration Menu

                    Territory Maintenance . . . . . 1
                    Customer Maintenance  . . . . . 2
                    Stock Maintenance . . . . . . . 3
                    Invoice Entry . . . . . . . . . 4
                    Invoice Enquiry . . . . . . . . 5
                    Order Entry and Maintenance . . 6
                    Stock Status Report . . . . . . 7
                    Return to Main Menu . . . . .<CR>

                    Please select a function      :
```

Figure 2.1a - The Demonstration Programs

Select option 4 and the first invoice entry window is displayed. Key <HLP> and your screen will look similar to Figure 2.1b.


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
NUM

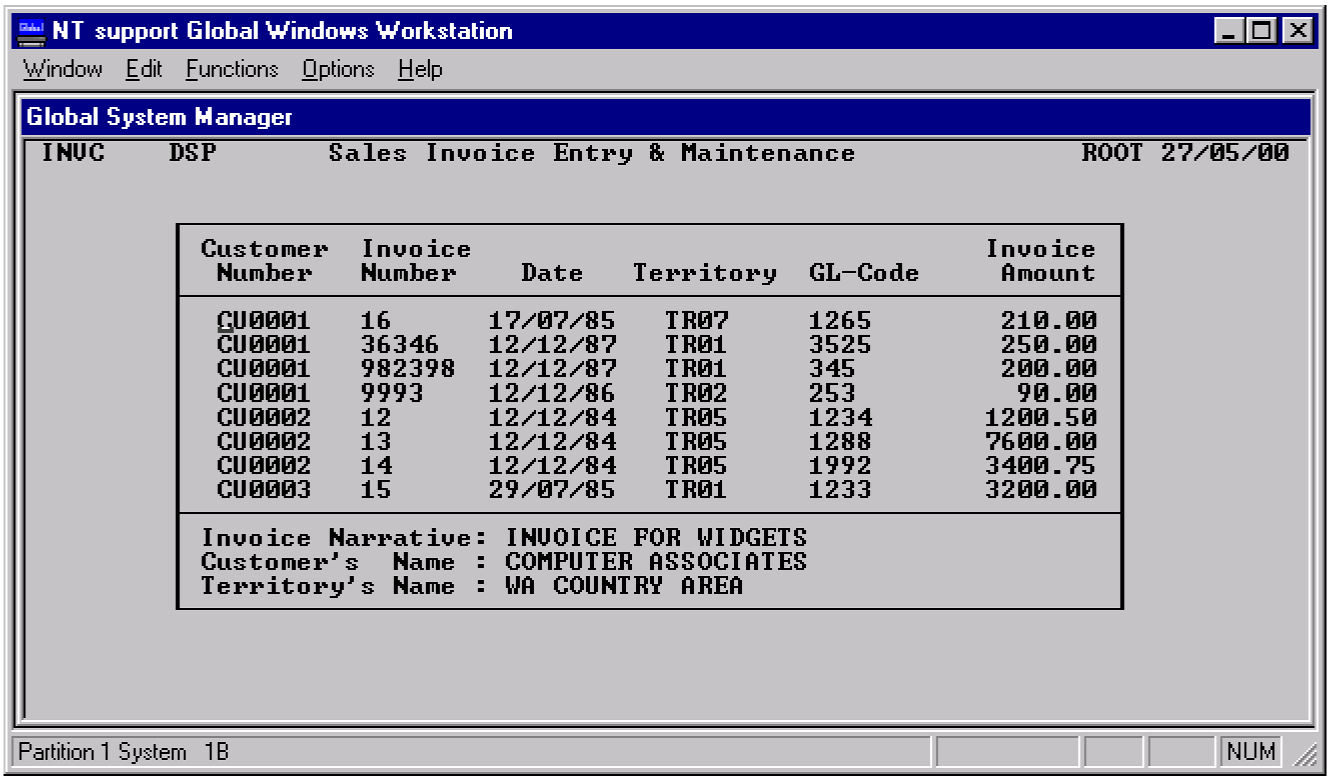
```
INVC    ADD       Sales Invoice Entry & Maintenance              ROOT 27/05/00

          Customer  Invoice                                  Invoice
           Number   Number    Date    Territory  GL-Code      Amount

          Invoice Narrative:
          Customer's  Name :
          Territory's Name :

                                          Pop-Up Enquiry     f1
                                          Enquiry mode       F6
                                          Clear the window   F4
                                          Back-page window   PAGEUP
                                          Frwd-page window   PAGEDN
                                          Skip fields        TAB
                                          Cursor home        HOME
                                          Terminate window   ESCAPE
```

Figure 2.1b - Invoice Entry Screen with Help

The small help window lists the options you have in entering the customer number. On your computer you will probably have different keys listed against each option. This is because there are many varieties of keyboard, each with different keys and key labels. The key labels listed in this and every other help window you will see should match the keys and key labels on your own keyboard. These are set up using the customisation utility described in section 2.2. You can change them at any time should you wish to do so.

The screens shown in this section are taken from an IBM PC computer so your keyboard may have been set up differently. You should find that the options are the same in each case, only the key labels will differ. You should of course use the keys listed in your on-screen help window rather than those documented in this manual, which are used as an example only. It is a feature of the Speedbase Presentation Manager that the on-screen help is customised for each user's keyboard, and is therefore more useful than any written documentation as a guide to using the computer.

As a guide to using Speedbase Presentation Manager windows the following sections take you through the sales invoice entry and maintenance sample program. Start by displaying a list of customer invoices. As you have the help window displayed, move the cursor down to the fifth line, Frwd-page window, and key <RET>. The window fills with customer invoice records, see Figure 2.1c.


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
INVC     DSP        Sales Invoice Entry & Maintenance        ROOT 27/05/00

      Customer  Invoice                                         Invoice
       Number   Number     Date    Territory  GL-Code           Amount

       CU0001   16        17/07/85   TR07     1265              210.00
       CU0001   36346     12/12/87   TR01     3525              250.00
       CU0001   982398    12/12/87   TR01     345               200.00
       CU0001   9993      12/12/86   TR02     253                90.00
       CU0002   12        12/12/84   TR05     1234             1200.50
       CU0002   13        12/12/84   TR05     1288             7600.00
       CU0002   14        12/12/84   TR05     1992             3400.75
       CU0003   15        29/07/85   TR01     1233             3200.00

      Invoice Narrative: INVOICE FOR WIDGETS
      Customer's  Name : COMPUTER ASSOCIATES
      Territory's Name : WA COUNTRY AREA
```

Figure 2.1c - Sales Invoice Window

The window is divided into three areas. At the top are the headings for the main area where the customer details are displayed. The main area is capable of being scrolled. This means that you have several records on the screen at once and you can move forwards and back through them, using the cursor keys. At the bottom of the window the third area shows further details of the current record. This is known as the non-scrolled area.

You will see the current record highlighted as you move down the window using the cursor-down key. You also see the details in the non-scrolled area changing as you change current record. When you get to the bottom of the window it scrolls to reveal the next record, and so on. If you use the cursor-up key to move up through the window, once you reach the top the window scrolls to reveal the previous record.

Alternatively, and to move more quickly through the records, you can display a whole new page at once. To do this you could display the help window as before, move to line 5 and key <RET>. Or you could simply use the page key (listed as <PGE> in Table 2.1), which on the IBM PC keyboard is usually the key labeled PageDown. To go back a page use the back-page key (listed as <BPG> in Table 2.1). On a PC keyboard this is usually the key labeled PageUp. Table 2.1 lists all the function keys that may be customised, together with their mnemonics.

Function keys are referred to by their mnemonics throughout this manual so it may be helpful to write yours into Table 2.1 for reference.

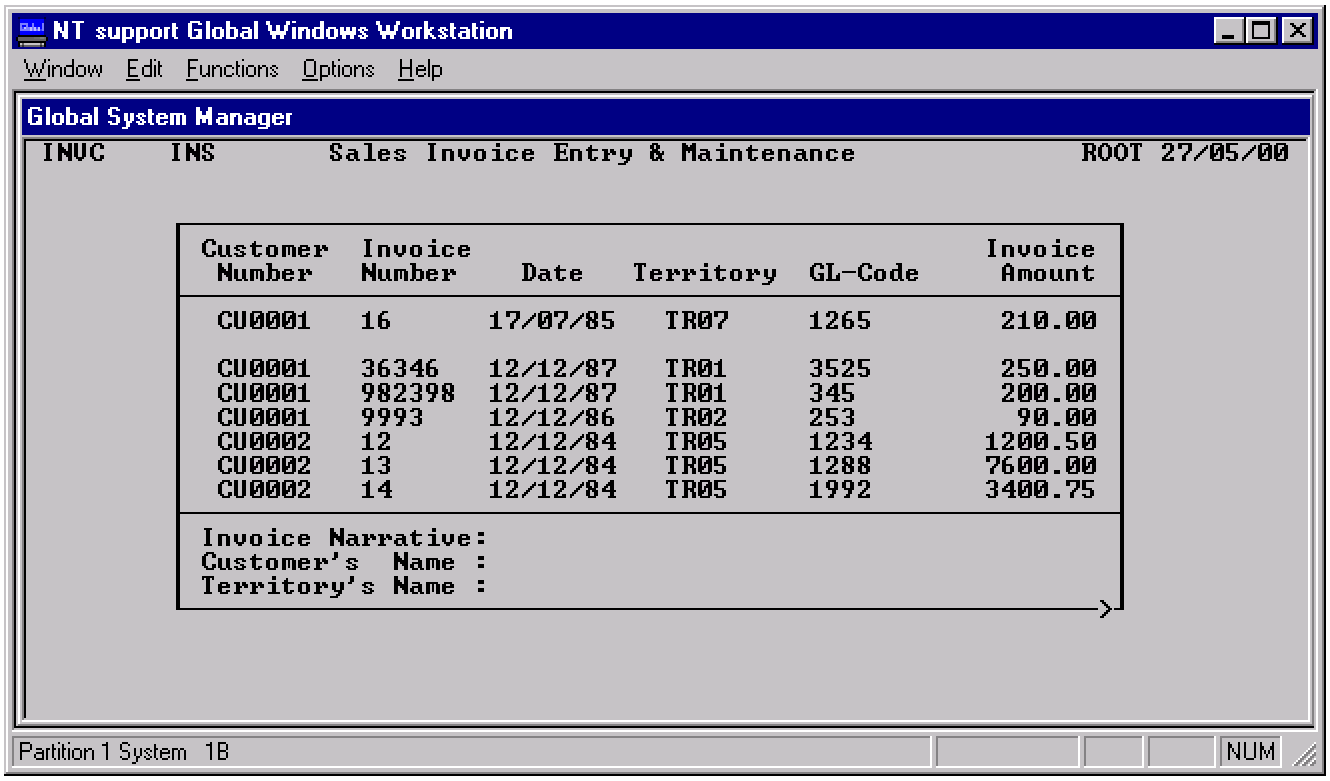
Function Description	**Function Mnemonic**	**Example Key Label**	**Your Key Label**
Enter field or select record	RET	Enter	
User function 1	UF1	F1	
User function 2	UF2	F2	
User function 3	UF3	F3	
Go to next window or process	NXT	End	
Display next page of records	PGE	PageDown	
Display prior page of records	BPG	PageUp	
Up to prior record	UP	Up Arrow	
Down to next record	DWN	Down Arrow	
Skip past fields	SKP	Tab	
Abort this program	ABO	Escape	
Back to prior window	BCK	F9	
Clear record or window	CLR	F4	
Delete the current record	DTE	F7	
Go to first or last record	HME	Home	
Step back to prior field	BFL	^Tab	
Enter ENQ mode, change index	ENQ	F6	
Insert record here	INS	F5	
Undelete record	UDL	<CTRL D>	
Cursor left one character	LFT	Left Arrow	
Cursor right one character	RGT	Right Arrow	
Insert a space at this character	ISP	Insert	
Delete character at cursor	DEL	Delete	
Change case of character	CCC	<CTRL C>	
Clear field from cursor on	CFL	<CTRL F>	
Restore field contents	RFL	<CTRL R>	
Display help messages	HLP	F8	

Table 2.1 - Function Key Customisation

With a page of invoice records on the screen, key <HLP> and then, with the help window displayed, key <HLP> again. A second help window is displayed which explains the function of the sales invoice entry and maintenance program. This second level of help is also available at all times for all Speedbase application programs and utilities. If function keys, as listed in Table 2.1 are included in the help window they are described by their correct key label as set up using the customisation utility.

As described in the help window there are four different orders in which records may be displayed in the window we are using. Key anything to return to the window, then key <ENQ>. The window clears and the cursor is on the first index field, customer number. Key <ENQ> again and the cursor moves to the territory number field. Key <ENQ> again and it moves to the invoice number field. If you now key <PGE> the records are displayed in invoice number order. Key <ENQ> twice more, followed by <PGE> and they are displayed in date order. This illustrates the powerful indexing facilities of the Speedbase Presentation Manager. You should experiment with the <ENQ> facility as you will find it very useful in all the Speedbase applications you use.

With a page of records displayed, key <INS>. The window scrolls to allow you to insert a new record:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
INVC     INS        Sales Invoice Entry & Maintenance              ROOT 27/05/00

 Customer  Invoice                                          Invoice
  Number   Number     Date    Territory   GL-Code           Amount

  CU0001   16        17/07/85   TR07      1265              210.00

  CU0001   36346     12/12/87   TR01      3525              250.00
  CU0001   982398    12/12/87   TR01      345               200.00
  CU0001   9993      12/12/86   TR02      253                90.00
  CU0002   12        12/12/84   TR05      1234             1200.50
  CU0002   13        12/12/84   TR05      1288             7600.00
  CU0002   14        12/12/84   TR05      1992             3400.75

 Invoice Narrative:
 Customer's   Name  :
 Territory's  Name  :
                                                                  >
```

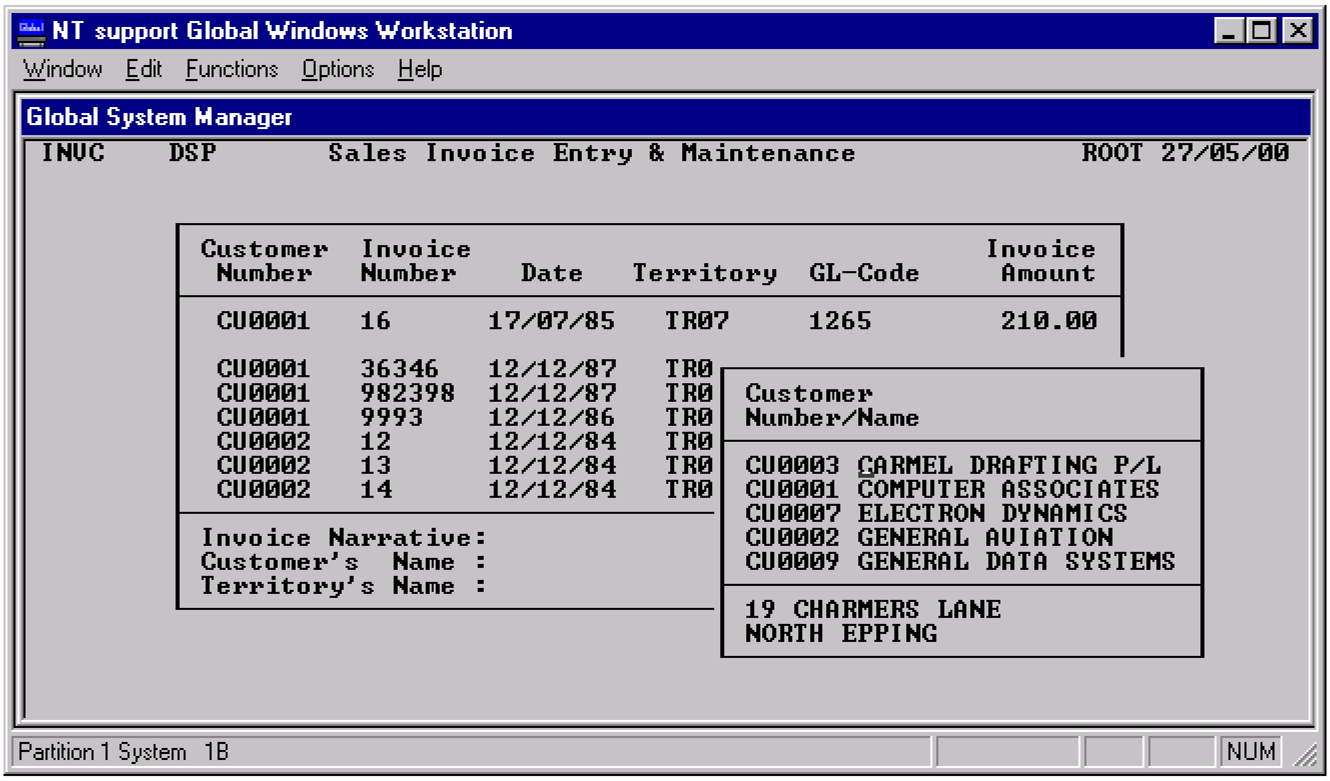
Figure 2.1d - Inserting a New Record

Key in the details for the new record, using customer number CU0007 and territory TR04. When you redisplay the window, by keying <PGE> or <BPG>, you will notice that your new record is displayed in its correct position relative to the other records. If you place the cursor on your new record and key <DTE> it will be deleted.

It is often the case when adding records in an application like this, that you are required to enter a field and you find you do not know what to key. For example, if you key <INS> to insert a new invoice record, you are required to enter a customer number. You may know all your customer numbers, but you may not. To cater for this common situation the Speedbase Presentation Manager uses pop-up enquiry windows.

With the cursor on the empty customer number field you will notice that a > character has appeared at the bottom right-hand corner of the window. This indicates that a pop-up window is available to help you.

Key <UF1>, usually set up as function key F1 on the IBM PC keyboard. The customer pop-up window is displayed. Key <PGE> to display the first five customer records:

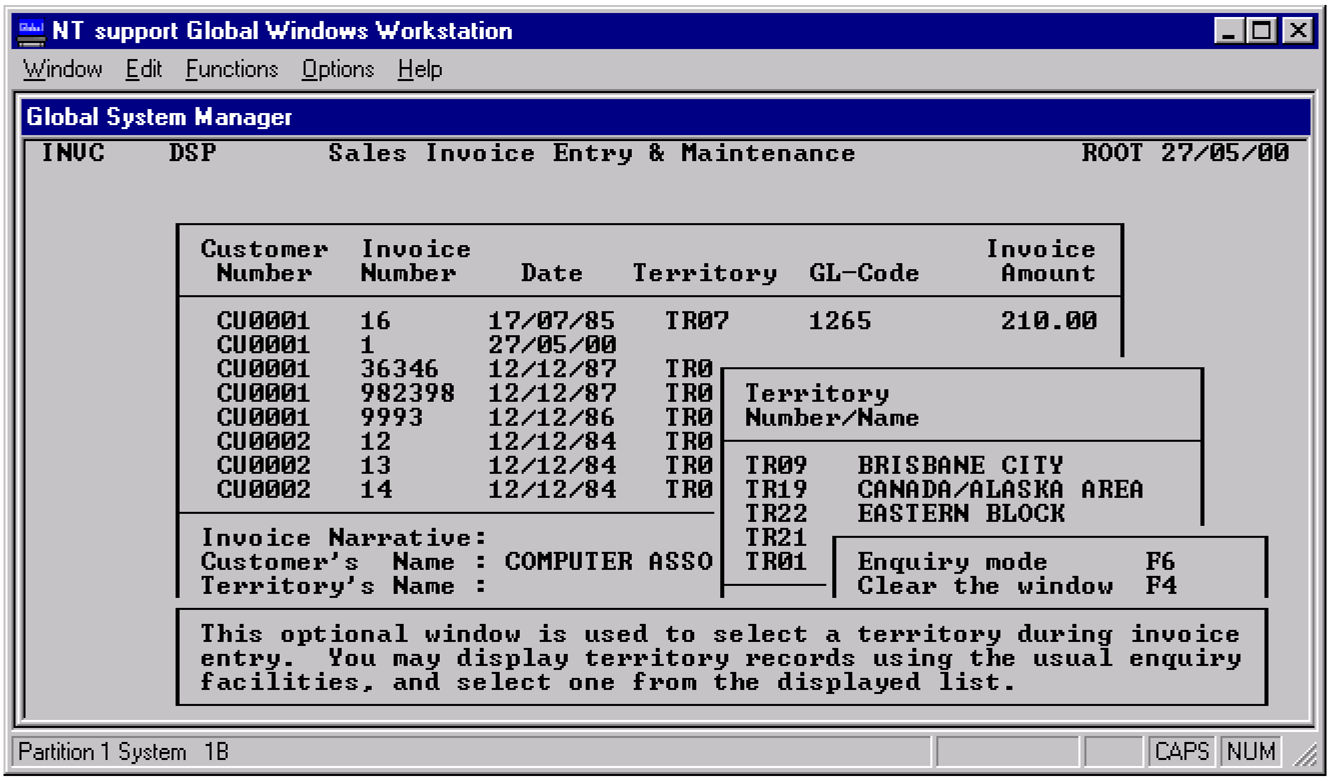

NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
NUM

```
INVC     DSP        Sales Invoice Entry & Maintenance              ROOT 27/05/00

 Customer  Invoice                                         Invoice
  Number   Number      Date    Territory   GL-Code          Amount

  CU0001   16        17/07/85    TR07      1265             210.00

  CU0001   36346     12/12/87    TR0
  CU0001   982398    12/12/87    TR0   Customer
  CU0001   9993      12/12/86    TR0   Number/Name
  CU0002   12        12/12/84    TR0
  CU0002   13        12/12/84    TR0   CU0003 CARMEL DRAFTING P/L
  CU0002   14        12/12/84    TR0   CU0001 COMPUTER ASSOCIATES
                                       CU0007 ELECTRON DYNAMICS
 Invoice Narrative:                    CU0002 GENERAL AVIATION
 Customer's  Name :                    CU0009 GENERAL DATA SYSTEMS
 Territory's Name :
                                       19 CHARMERS LANE
                                       NORTH EPPING
```

Figure 2.1e - The Customer Pop-Up

Move the cursor to the customer you require and key <RET>. The pop-up disappears and the number of your chosen customer appears in the main window. You then proceed as before.

When entering the territory number you may key <UF1> to display the territory pop-up. Note that the pop-up windows have all the usual help windows, see Figure 2.1f for example. Note also that the records in these pop-ups also make use of the enquiry facility described above so that while in the pop-up you may key <ENQ> to display territories in number or name order.


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

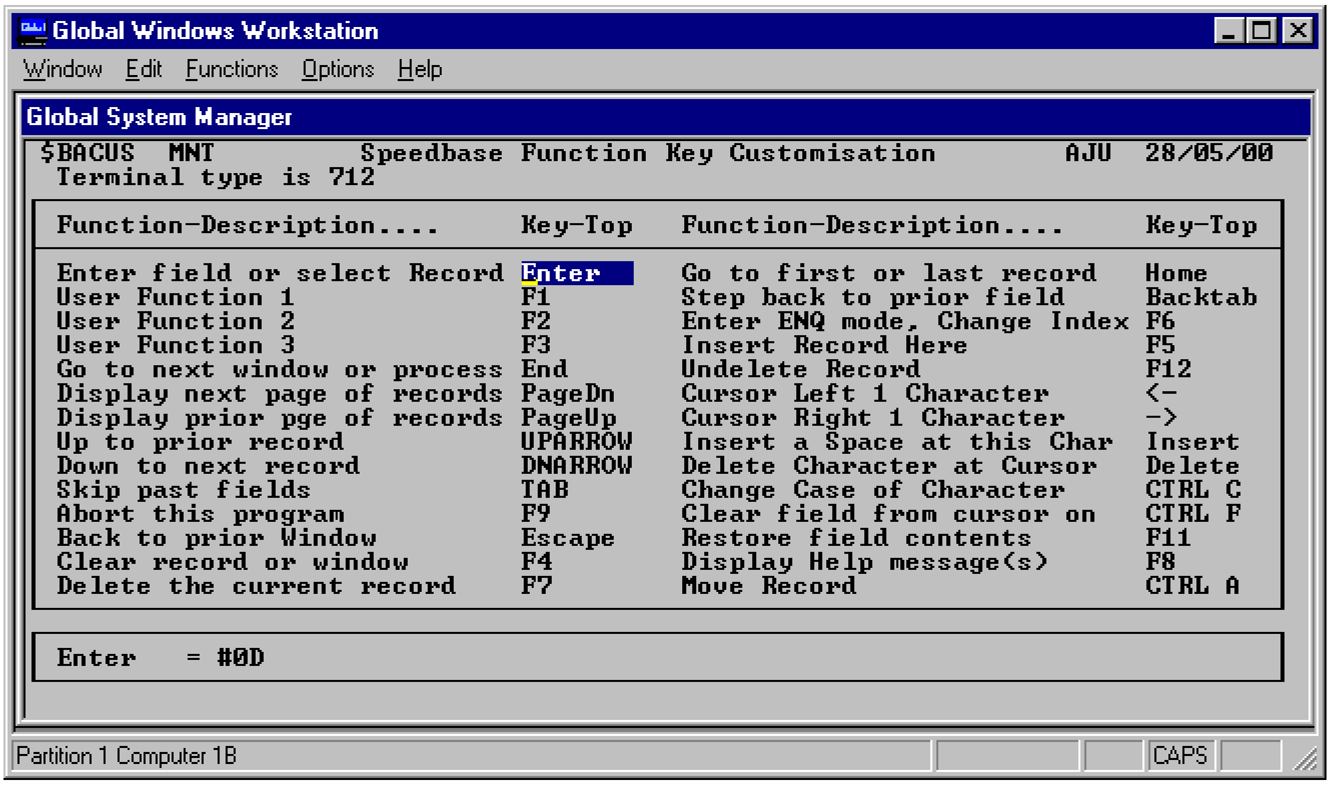
```
INVC     DSP        Sales Invoice Entry & Maintenance              ROOT 27/05/00

          Customer  Invoice                                      Invoice
           Number   Number     Date    Territory  GL-Code         Amount

           CU0001   16        17/07/85   TR07     1265            210.00
           CU0001   1         27/05/00
           CU0001   36346     12/12/87   TR0
           CU0001   982398    12/12/87   TR0   Territory
           CU0001   9993      12/12/86   TR0   Number/Name
           CU0002   12        12/12/84   TR0
           CU0002   13        12/12/84   TR0   TR09   BRISBANE CITY
           CU0002   14        12/12/84   TR0   TR19   CANADA/ALASKA AREA
                                               TR22   EASTERN BLOCK
          Invoice Narrative:                   TR21
          Customer's  Name : COMPUTER ASSO     TR01   Enquiry mode      F6
          Territory's Name :                          Clear the window  F4

          This optional window is used to select a territory during invoice
          entry.  You may display territory records using the usual enquiry
          facilities, and select one from the displayed list.
```

Figure 2.1f - Help Windows for the Territory Pop-Up

You should now make use of the programs in the demonstration system to become thoroughly familiar with the screen and keyboard facilities of the Speedbase Presentation Manager. Remember that all Speedbase applications, no matter what their function, use the screen and keyboard in exactly the same way, so that once you are familiar with one application you should be able to run any other without difficulty.

2.2 Screen and Keyboard Customisation

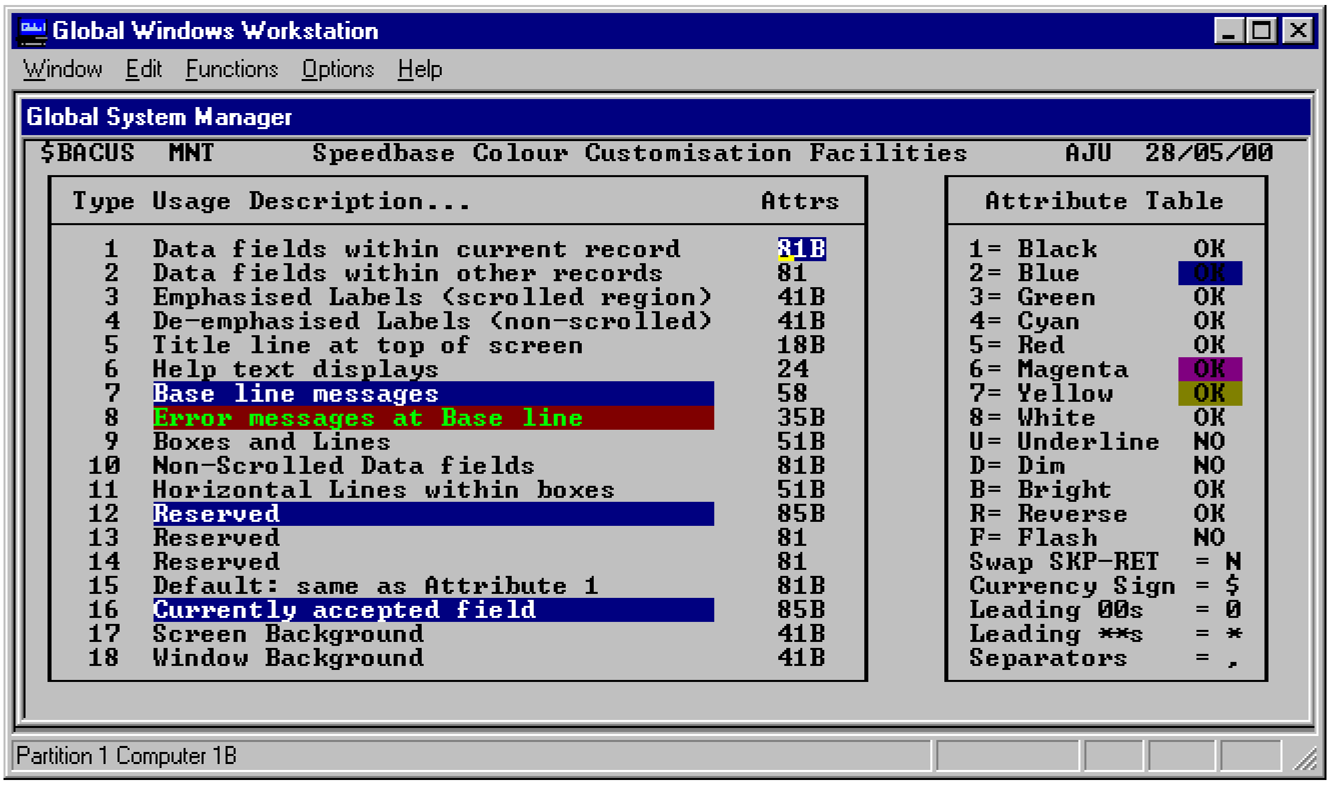
The $BACUS customisation utility is used to set up or amend the function keys and screen attributes of your terminal. The information about these features is stored on the SYSRES volume in a customisation file named T>xxx where xxx is the number of your terminal as entered when you log on. Run $BACUS. The following window is displayed:


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
$BACUS   MNT          Speedbase Function Key Customisation          AJU   28/05/00
 Terminal type is 712

 Function-Description....     Key-Top   Function-Description....     Key-Top

 Enter field or select Record Enter     Go to first or last record   Home
 User Function 1              F1        Step back to prior field     Backtab
 User Function 2              F2        Enter ENQ mode, Change Index F6
 User Function 3              F3        Insert Record Here           F5
 Go to next window or process End       Undelete Record              F12
 Display next page of records PageDn    Cursor Left 1 Character      <-
 Display prior pge of records PageUp    Cursor Right 1 Character     ->
 Up to prior record           UPARROW   Insert a Space at this Char  Insert
 Down to next record          DNARROW   Delete Character at Cursor   Delete
 Skip past fields             TAB       Change Case of Character     CTRL C
 Abort this program           F9        Clear field from cursor on   CTRL F
 Back to prior Window         Escape    Restore field contents       F11
 Clear record or window       F4        Display Help message(s)      F8
 Delete the current record    F7        Move Record                  CTRL A

 Enter    = #0D
```

Figure 2.2a - Function Key Customisation

The screen shown in Figure 2.2a is for a typical keyboard as set up for the examples used in section 2.1. Your terminal will probably have different keys assigned to each function, and different key-tops (key labels). To assign a different key to a function simply move the cursor to it and enter a different key label. The utility will prompt you to press the actual keyboard key you wish to use. If there is a function for which you do not wish to assign a function key simply enter a label of spaces. When you have completed the required changes, key <NXT>. The screen customisation window is displayed, see Figure 2.2b.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B

```
$BACUS  MNT      Speedbase Colour Customisation Facilities      AJU   28/05/00

 Type Usage Description...                      Attrs     Attribute Table

   1  Data fields within current record          81B      1= Black       OK
   2  Data fields within other records           81       2= Blue        OK
   3  Emphasised Labels (scrolled region)        41B      3= Green       OK
   4  De-emphasised Labels (non-scrolled)        41B      4= Cyan        OK
   5  Title line at top of screen                18B      5= Red         OK
   6  Help text displays                         24       6= Magenta     OK
   7  Base line messages                         58       7= Yellow      OK
   8  Error messages at Base line                35B      8= White       OK
   9  Boxes and Lines                            51B      U= Underline   NO
  10  Non-Scrolled Data fields                   81B      D= Dim         NO
  11  Horizontal Lines within boxes              51B      B= Bright      OK
  12  Reserved                                   85B      R= Reverse     OK
  13  Reserved                                   81       F= Flash       NO
  14  Reserved                                   81       Swap SKP-RET   = N
  15  Default: same as Attribute 1               81B      Currency Sign  = $
  16  Currently accepted field                   85B      Leading 00s    = 0
  17  Screen Background                          41B      Leading **s    = *
  18  Window Background                          41B      Separators     = ,
```

Figure 2.2b - Colour Screen Customisation

The window of the left of the screen lists the eighteen screen functions and the attributes assigned to each. Figure 2.2b shows the attributes set up for the IBM PC screen examples used in section 2.1. The window to the right of the screen lists the attributes available on your terminal. This window also shows the numeric editing symbols, which may also be customised. The example of Figure 2.2b, which is a colour screen, all the colours are available, i.e. OK, as are the attributes bright, reverse and flash.

Figure 2.2c shows the same customisation window for a monochrome screen, the Wyse-60. None of the colours is available, of course, but the attributes underline, dim, reverse and flash are all OK.

To change the colour and attributes to be used for a screen function, move the cursor to it and enter the appropriate attribute codes. For example, on a colour screen, to display error messages in blue on a magenta background and flashing, move the cursor to function line eight and key 26F or, on the Wyse-60, for flashing and reverse key RF. Key <NXT> to exit, saving the customisation you have made.

Terminal attribute choice is a matter of personal taste and you will probably wish to experiment to find the combinations that suit you. Note that, having chosen these combinations, they will be used by all Speedbase applications on your terminal, no matter who the supplier of the software is. This ensures you get the screen and keyboard facilities you require in an efficient and consistent fashion.

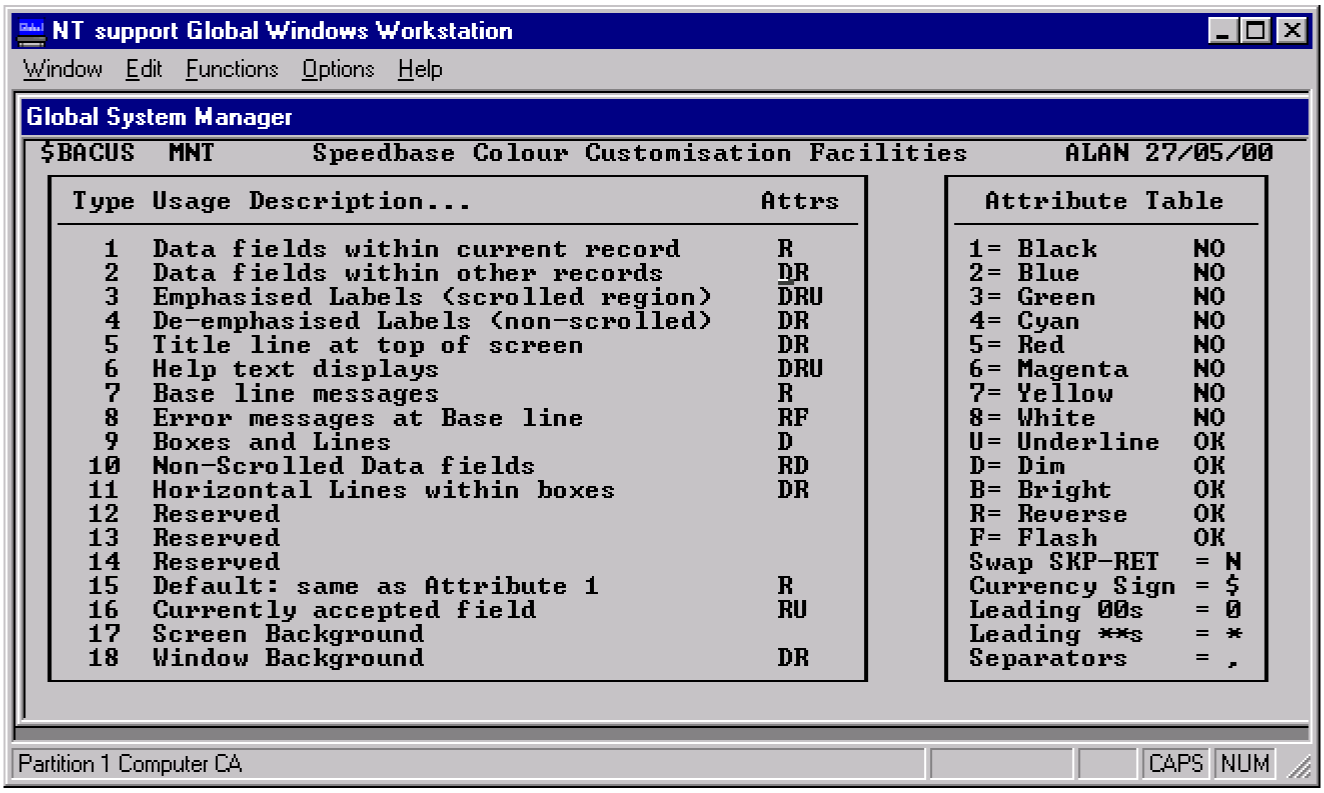
The Currency Sign field contains the leading currency sign to be printed (where requested) in money fields. Speedbase is supplied as standard with this symbol set to "$". This should be changed to the local currency symbol where appropriate.

The leading 00 field contains the symbol to be used when zero fill is requested on a numeric field. This will normally be left unchanged.

The leading * field contains the symbol to be used when asterisk fill is requested on a numeric field. This will normally be left unchanged.

The Separator field contains the symbol to be inserted between thousands in numeric fields (e.g. the comma in 999,999,999). This may be changed to a different symbol (such as ".") in appropriate countries.

A sample screen customisation for a monochrome screen is shown below:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer CA
CAPS NUM

```
$BACUS   MNT        Speedbase Colour Customisation Facilities        ALAN 27/05/00

 Type Usage Description...                         Attrs      Attribute Table

   1  Data fields within current record            R          1= Black        NO
   2  Data fields within other records             DR         2= Blue         NO
   3  Emphasised Labels (scrolled region)          DRU        3= Green        NO
   4  De-emphasised Labels (non-scrolled)          DR         4= Cyan         NO
   5  Title line at top of screen                  DR         5= Red          NO
   6  Help text displays                           DRU        6= Magenta      NO
   7  Base line messages                           R          7= Yellow       NO
   8  Error messages at Base line                  RF         8= White        NO
   9  Boxes and Lines                              D          U= Underline    OK
  10  Non-Scrolled Data fields                     RD         D= Dim          OK
  11  Horizontal Lines within boxes                DR         B= Bright       OK
  12  Reserved                                                R= Reverse      OK
  13  Reserved                                                F= Flash        OK
  14  Reserved                                                Swap SKP-RET   = N
  15  Default: same as Attribute 1                 R          Currency Sign  = $
  16  Currently accepted field                     RU         Leading 00s    = 0
  17  Screen Background                                       Leading **s    = *
  18  Window Background                            DR         Separators     = ,
```

Figure 2.2c - Monochrome Screen Customisation

3. Database Backup Utility ($BASAV)

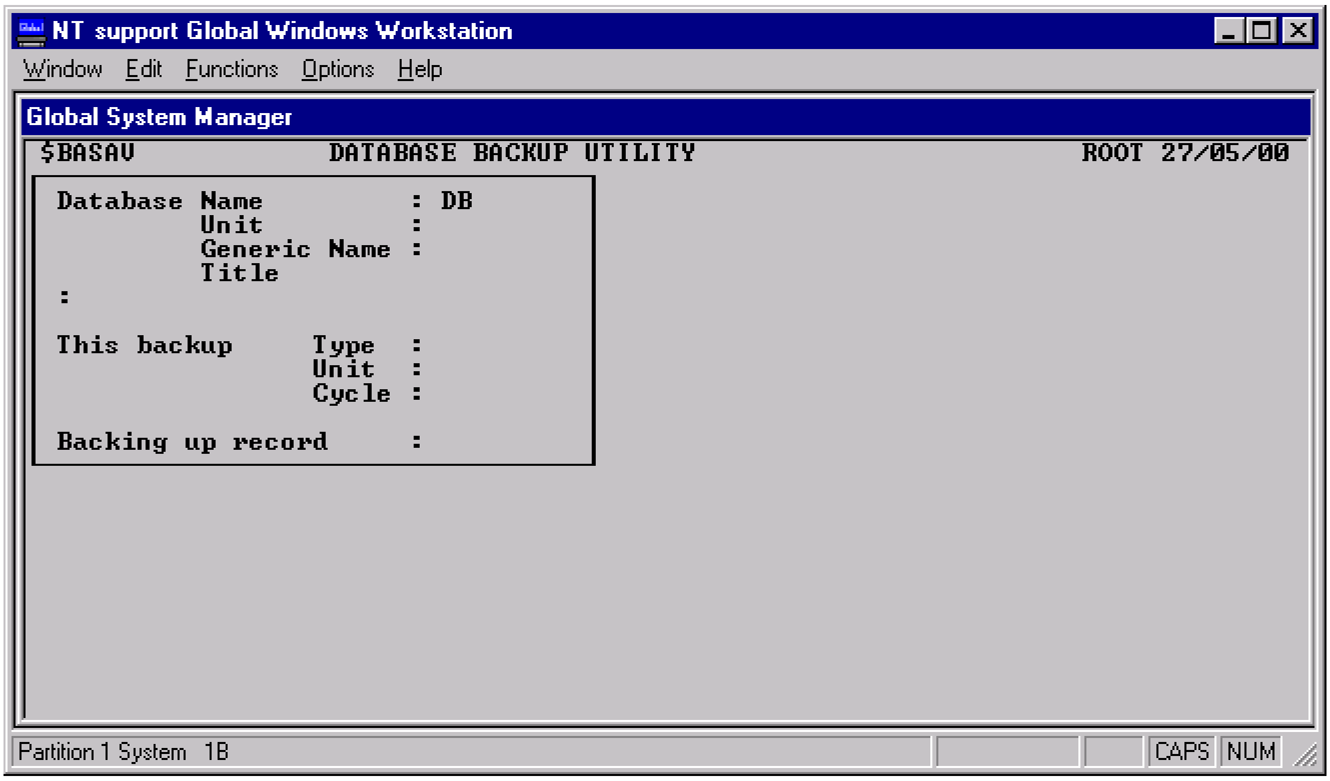
If your Speedbase database is stored in the Global file structure the backup utility is used to create backup copies of your data. These backup copies may later be used to restore your database following corruption or hardware failure. You might also wish, for some operational reason, to use a backup taken at an earlier time in order to restore the database to its earlier state. The backup utility performs the backup function and the database generation utility described in chapter four performs the restoration and re-creation function.

The backup utility has been designed to cater for all Speedbase users, including those with large databases and those wishing to do backups onto diskettes. You may choose to do incremental backups in which only the data that has changed since the last full backup is backed up. This has the great advantage that the backup is usually much quicker and takes up much less space on the backup diskettes.

In order to ensure the integrity of your backups and to assist in the restoration process, Speedbase stores the history of your backups in the database itself, up to a maximum of thirty-six cycles. This information is displayed in a window (Figure 3.1a) and is itself backed up.

3.1 Running the Backup Utility

To run the database backup utility, key 3 at the main menu. The following window is displayed:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS NUM

```
$BASAV              DATABASE BACKUP UTILITY              ROOT 27/05/00

 Database Name          : DB
          Unit          :
          Generic Name  :
          Title
 :

 This backup      Type  :
                  Unit  :
                  Cycle :

 Backing up record      :
```

Figure 3.1 - The Backup Input Window

Enter the name of the database and its unit-id (e.g. DEMON and FLS). Then enter the generic name you wish to use for the database, if this is the first time a backup has been done. You may choose any two letters for the generic name although it is best if all the databases on your computer have different generic names. If this is not the first backup of this database the generic name is displayed for you. The database title is always displayed so that you can check that you a doing a backup of the correct database.

The history of the last thirty-six backup cycles is displayed for your information. You then enter the type of backup you want to do. This can be F, I, or P depending on whether you require to do a full, incremental, or part incremental backup. The unit-id of the backup device, usually a diskette, is entered next (e.g. $B, 100, etc.).

The cycle-id is displayed and you may override this if you are doing a full backup. If the backup is incremental or part incremental the utility will automatically use the cycle-id displayed and you key <RET> to continue.

3.1.1 The Backup Process

The backup utility will now begin the backup process. You may be requested to mount a volume on the backup unit (see section 3.1.2). If you are doing an incremental backup the previous backups of the cycle will be validated to ensure continuity. If you are doing a full backup the baseline message:

```
Backing up Dictionary ...
```

indicates that the data dictionary is being copied to the backup volume.

The name of a record is displayed as it is added to the backup. Once this operation is complete, the baseline display:

```
Updating Backup History ......
```

indicates that information about the backup is about to be included in the history table and on the database itself. This display may be followed by a mount prompt (see section 3.1.2) requesting you to mount the first backup diskette of the cycle. Finally, the baseline display:

```
Backup Completed ........
```

informs you that the backup was successful. You should write the volume-id on the label of any new diskettes used in the backup.

3.1.2 The MOUNT VOLUME Prompt

This baseline prompt is as follows:

Load Backup Volume *xxannn* on Unit *uuu* for *yyyyy* cycle *a*..

where xx is the database generic name, a is the cycle-id, nnn is the diskette number, uuu is the unit and yyyyy is the database name. You should mount a backup diskette with an appropriate volume-id. If you are doing a full backup, you may mount a diskette with the volume-id BACKUP rather than the requested volume-id. In any other circumstances, the actual volume requested must be loaded. Enter <RET> when you have mounted the required diskette or <ESC> to exit.

For example, if you do a backup of the DEMON sample database on unit 100, using cycle B, when it is time to mount the second diskette the mount prompt is displayed as follows:

```
      Load Backup Volume DEB 2 on Unit 100 for DEMON cycle B..
```

At the end of the backup the mount prompt:

```
      Load Backup Volume DEB 1 on Unit 100 for DEMON cycle B..
```

is displayed and you mount the first diskette of the cycle so that the backup history may be updated.

3.2 Backup Types and Strategies

You may choose to do one of three types of backup:

- **Full Backup**: All records on the database are copied;
- **Full Incremental Backup**: Only records which have been updated since the last full backup are copied;
- **Part Incremental Backup**: Only records which have been updated since the last backup of any type are copied.

A full incremental backup supersedes all the part incremental backups done since the last full or full incremental backup. When you use the generation utility to restore your database from a backup it uses the last full backup, then the most recent full incremental backup and any subsequent part incremental backups.

Backups are maintained in cycles. A backup cycle consists of a full backup followed by any number of full incremental backups and part incremental backups. Each of these cycles of backups is assigned a backup cycle-id (e.g. A). The next full backup is assigned the next backup cycle-id (e.g. B etc.). Many variations of backup strategy are available to you, here are two examples:

3.2.1 Backup Example A

Let us assume you have a large database and a moderate level of activity. Your application involves monthly accounting periods and you decide to keep three sets of backup volumes, cycles A, B and C. Each cycle represents one month, so you can restore to any point in the last two or three months.

On the first day of each accounting period you do a full backup using the next backup cycle-id in sequence (A, B, C, then A again etc). Part incremental backups are then performed during the first week until, at the end of the week, a full incremental is done. This pattern continues each week until the next accounting period when you put aside the diskettes for this cycle and do a full backup to commence the next monthly cycle.

3.2.2 Backup Example B

You have a database of medium activity which occasionally receives additional updates. You decide to maintain weekly cycles during which a full backup is performed at the start of the week and part incremental backups are done each night. On days when many database updates are performed you do a full incremental backup instead of the part incremental backup, to store in one file on diskette all the database updates made so far this week.

Whilst there is no obligation to follow a specific backup strategy, it does assist you in maintaining the backup diskettes themselves. This is an important consideration when dealing with large databases. It is not advisable to keep cycles of an excessively large number of incremental backups. Although up to 99 incremental backup files are possible for any one cycle, the risk of a damaged file increases, as does the restore time.

3.3 Backup Volumes and Files

When initialising backup diskettes use the volume-id BACKUP. The backup utility expects the volume-id to be either BACKUP or of the form xxannn where xx is the database generic name, a is the backup cycle-id and nnn is the diskette number. For example, the first diskette of the cycle A backup of the sample database DEMON has the volume-id DEA 1. If the volume-id is initially BACKUP the utility changes it to a volume-id of the form xxannn.

The two-character generic name is entered when the first database backup is done, and should preferably be unique for each database on your computer. The prompts requesting you to mount backup diskettes specify the volume-id which should therefore be written on the diskette label. Orderly diskette labeling simplifies the handling of diskettes during the restoration process and the manual storage of backup cycles for each database.

As the backup utility will not overwrite a volume-id other than BACKUP, any spare diskettes should be initialised with the volume-id BACKUP to permit their immediate use when another diskette is required.

At the end of a backup involving two or more diskettes, the first diskette of the cycle must be put back in the drive so that the information on backup history can be updated. You will be prompted to re-load the first volume (see section 3.1.2) and labeling the diskette with the volume-id will simplify this procedure.

When an incremental backup is requested, the backup utility searches for the last backup file produced in the cycle. This file must be present in order to maintain contiguous and valid backup files throughout a backup cycle.

The file types of backup files shown in directory listings are as follows:

90	Full backup
91	Full incremental backup
92	Part incremental backup

The first file of any backup cycle is the backup header file, named BUxxxxxa where xxxxx is the database name and a is the backup cycle-id. The backup header file contains the history table and must be updated following a successful backup. The second file is the data dictionary (DIxxxxx) which is identical to the original dictionary, of file type 98. Following this is the full backup file, type 90 and any number of incremental backup files of types 91 and 92.

3.4 Considerations and Restraints

The backup utility requires exclusive use of the dictionary and all database files. None of these files may be in use by another user or screen while a backup is being made.

The data dictionary, which is backed up during the full backup process, must be copied complete to the first backup volume. In other words, it may not span multiple backup volumes. This should not be a restriction as even a diskette of minimal storage capacity would be able to hold an unusually large dictionary.

The maximum number of full or part incremental backups which may be performed in any one backup cycle is 99. It is not recommended that you perform anywhere near this many as you increase the possibility of one file being damaged and therefore corrupting the whole backup cycle.

Thirty-six backup history entries are displayed on the backup screen. This constitutes the last 36 backups performed over one or more backup cycles. Although not displayed, all backups performed prior to these are valid and may indeed be used to restore a database using the database generation utility.

The backup utility must have enough memory to load at least one complete data record at a time. The memory space available for loading of data records in a minimum user area of 31Kb is at least 8Kb which should be larger than the length of any record type.

3.5 Backing up to Hard Disk

The backup utility may be used to produce a condensed version of the database on your hard disk. This can be useful as a first step in regenerating or converting your database in order to reduce the total needed disk space.

To do this you must rename a unit on your hard disk using the Volume ID "BACKUP". The volume may contain other files unrelated to the backup (i.e. a unit also used for other purposes). In order to perform the backup to disk, rename an appropriate hard disk unit to the volume ID "BACKUP" using the System Manager file utility $F CHA instruction. Then specify this as the backup unit to $BASAV.

If, during the backup, other files exist on this unit, then $BASAV will issue the following prompt:

```
File(s) exist on Backup Volume - Delete?
```

Assuming that you do not want these files to be deleted you should enter "N" to this prompt. In this event any existing files will be remain on the unit, and the volume will not be re-labeled. Note that when you want to restore from such a backup using the $BADGN utility, it will prompt you for the cycle ID used during the backup process. You should therefore take a note of this while backing up.

4. Database Generation Utility ($BADGN)

If your Speedbase database is stored in the Global file structure, the generation utility is used to create, restore, convert, and modify its size. Where the input to the generation utility is an existing database then that database may be Global or Unix C-ISAM format.

Whenever it is run, the following happens:

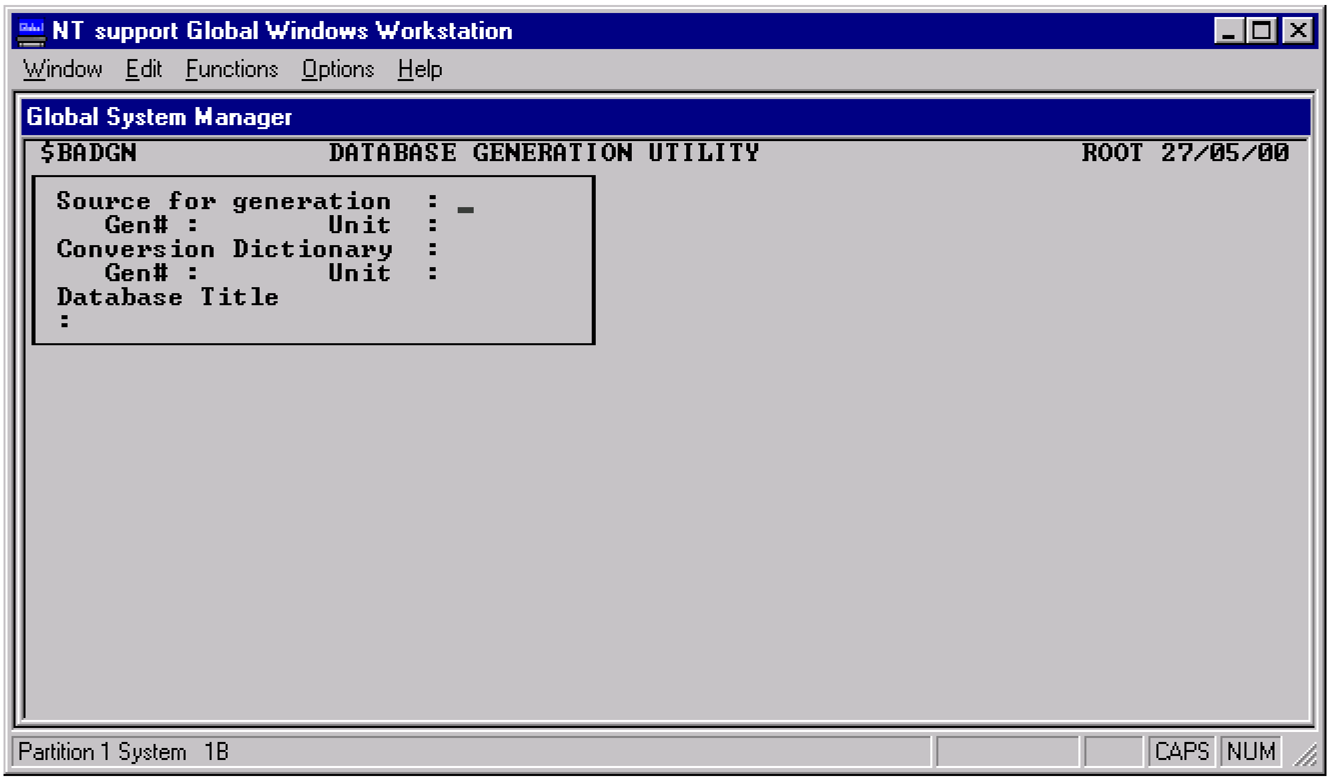
- A new database is generated;
- Its data dictionary is copied to the main database unit;
- Control passes to the rebuild utility which builds the index structure of the new database.

The most common use of the generation utility is for the restoration of your database from a backup copy produced by the backup utility, documented in chapter three. The generation utility is also used to create a new database and to modify existing databases. Such modifications include the expansion or contraction of the size of the database, data concentration, database conversion, transferring a Unix C-ISAM format Speedbase database to a Global format Speedbase database, or simply the redistribution of the database files to several different disk units.

4.1 Restoring a Database from Backup

One of the principal uses of the generation utility is to restore your database from a backup copy previously taken using the backup utility as described in chapter three. The backup, usually on diskette, consists of a number of files: a backup history table, data dictionary and a full backup data file followed by any number of incremental backup data files. This section covers the straightforward restore process, and the advanced facilities are documented in section 4.2.

To run the generation utility and restore from a backup, key 4 at the main menu. The utility displays the window:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS NUM

```
$BADGN              DATABASE GENERATION UTILITY                    ROOT 27/05/00
 Source for generation   : _
    Gen# :          Unit :
 Conversion Dictionary   :
    Gen# :          Unit :
 Database Title
 :
```

Figure 4.1a - The Generation Input Window

Enter the name of the source for the generation. Since you are doing a restore from backup you should reply:

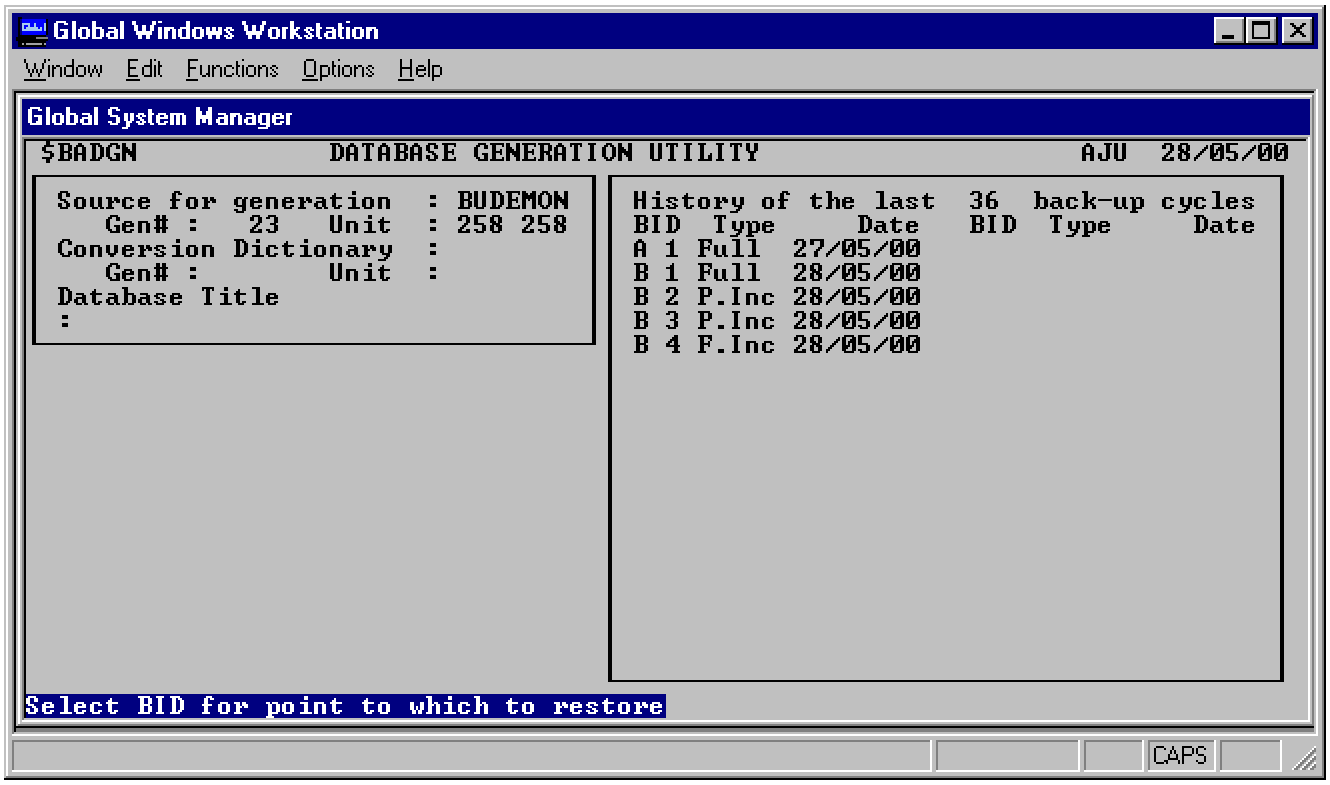
```
BUxxxxx
```

where xxxxx is the name of your database. You next specify the disk unit holding the backup (e.g. $B, 100, etc.). The utility displays a window showing the history of the last thirty-six backups and places the cursor on the most recent backup, see Figure 4.1b.

If the back-up volume is a hard disk volume labeled "BACKUP", then the generation utility will need to know the Cycle ID of the backup. In this event the following prompt will be displayed:

```
No Cycle ID available, Please enter:
```

You should now enter the Cycle ID letter (e.g. "A", "B" etc.) specified when the database was last backed up to this unit.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
CAPS

```
$BADGN              DATABASE GENERATION UTILITY                     AJU   28/05/00

 Source for generation   : BUDEMON     History of the last  36  back-up cycles
    Gen# :   23   Unit   : 258 258     BID  Type      Date    BID  Type      Date
 Conversion Dictionary   :             A 1 Full   27/05/00
    Gen# :        Unit   :             B 1 Full   28/05/00
 Database Title                        B 2 P.Inc  28/05/00
 :                                     B 3 P.Inc  28/05/00
                                       B 4 F.Inc  28/05/00

Select BID for point to which to restore
```

Figure 4.1b - The Backup History Window

In this window, BID is the backup-id, which is a combination of major and minor backup cycles, and can be seen to have incremented for each backup performed within any one backup cycle. Type is the type of backup performed, either Full, F.Inc (full incremental), or P.Inc (part incremental). The date on which the backup was performed is also displayed.

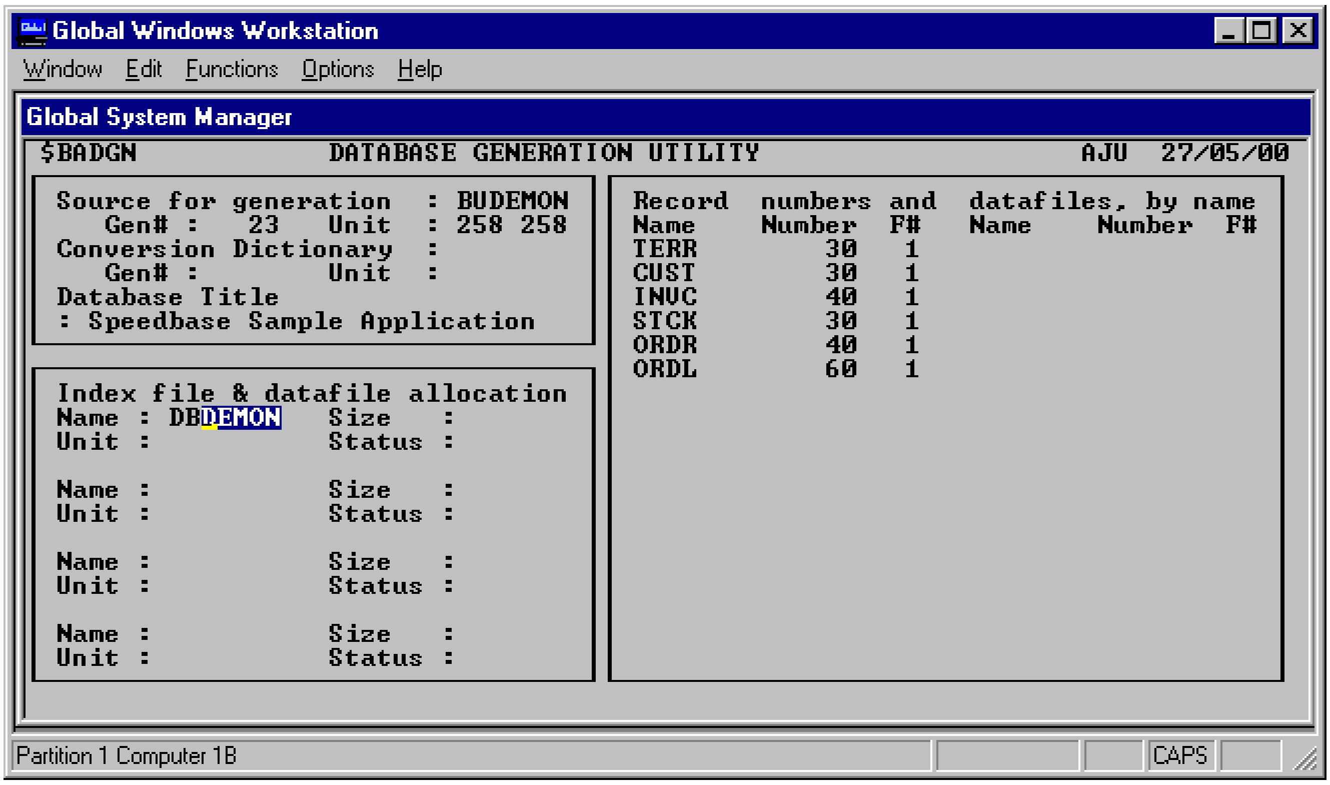
If you wish to restore from the most recent backup displayed, simply key <RET>. If you wish to use any other backup in the window, place the cursor on it, using the cursor keys, and key <RET>. If the backup you wish to use is earlier than the first one displayed in the window (i.e. is more than 36 backups ago) key <UF1>. The procedure in this case is described in section 4.2.

Unless you require to convert your database to a new format as part of the restore process you should key <RET> to the Conversion Dictionary prompt. The conversion process is covered in detail in section 4.2. You should now enter the database title, which has a maximum of thirty characters. If you wish to use the database title entered when the database was created, simply key <RET>. If you are restoring from a full backup, you then have the opportunity to concentrate the data. The utility displays the baseline prompt:

```
Data Concentration required ? Y/N :
```

The normal reply is N, which is the default. Data concentration is covered in detail in section 4.2.

The generation utility now replaces the backup history with a window showing a list of records on your database. Key <NXT> to use the number of records and datafile as displayed, for each record type. The file allocation window is then displayed, see Figure 4.1c.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
$BADGN              DATABASE GENERATION UTILITY                  AJU   27/05/00

 Source for generation   : BUDEMON      Record   numbers  and   datafiles, by name
    Gen# :   23   Unit   : 258 258      Name     Number   F#    Name      Number  F#
 Conversion Dictionary   :              TERR         30    1
    Gen# :        Unit   :              CUST         30    1
 Database Title                         INVC         40    1
 : Speedbase Sample Application         STCK         30    1
                                        ORDR         40    1
                                        ORDL         60    1
 Index file & datafile allocation
 Name : DBDEMON    Size   :
 Unit :            Status :

 Name :            Size   :
 Unit :            Status :

 Name :            Size   :
 Unit :            Status :

 Name :            Size   :
 Unit :            Status :
```

Partition 1 Computer 1B   CAPS

Figure 4.1c - The Record and File Allocation Windows

Key <RET> to accept the default database name, file size and file units. If you wish to change any of these details, see section 4.2. The baseline message:

```
Updating new Database ....
```

Is displayed while the data transfer is performed. If your backup cycle spans multiple volumes, you will be asked to mount further disks, as documented in section 3.1.2. When data transfer is complete, the baseline message:

```
New database created ...
```

is displayed and control passes to the database rebuild utility which automatically completes the restore process by building the database index file.

4.2 Advanced Database Restore Facilities

While restoring your database from backup you may carry out a number of modifications to its structure. The ability to use the backup and restore process to make database modifications without using large amounts of disk space is a powerful facility of the Speedbase Presentation Manager.

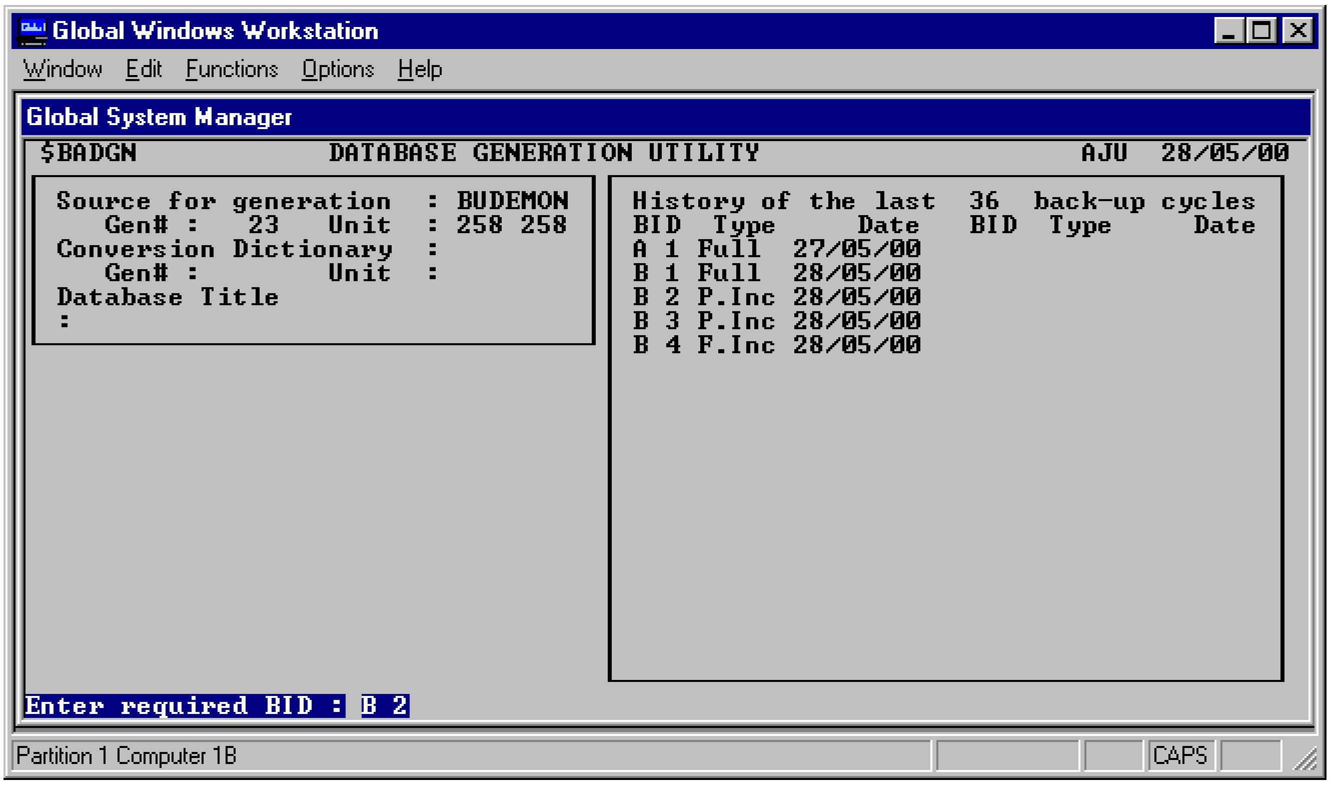
This section describes how to restore from a backup cycle earlier than those displayed in the backup history window, database conversion and data concentration, and how to modify the number of records allocated to a record type, the datafile in which it is stored, the database name, file sizes and disk units.

4.2.1 Restoring from an Early Backup Cycle

To restore from a backup cycle earlier than any displayed in the backup history window, key <UF1> for the baseline prompt (see figure 4.2):

```
Enter required BID :
```

Enter the backup-id you wish to use (e.g. A1). Its backup-id must be the same as the disk cycle you are using. For example, with a BID of A1 you should be using disk cycle A for this restore. The backup utility then proceeds as documented in section 4.1.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BADGN              DATABASE GENERATION UTILITY              AJU   28/05/00

 Source for generation     : BUDEMON     History of the last  36  back-up cycles
     Gen# :    23   Unit   : 258 258     BID  Type       Date    BID  Type      Date
 Conversion Dictionary     :             A 1  Full   27/05/00
     Gen# :         Unit   :             B 1  Full   28/05/00
 Database Title                          B 2  P.Inc  28/05/00
 :                                       B 3  P.Inc  28/05/00
                                         B 4  F.Inc  28/05/00

Enter required BID : B 2
```

Figure 4.2 - The Backup History Window

4.2.2 Database Conversion

Having chosen a backup-id for the restore you are prompted to enter the name of a conversion dictionary and its disk unit. This dictionary becomes the dictionary for your database and is used to convert your database to its new structure. The restore process then proceeds as documented in section 4.1 except that, following the entry of the database title, the baseline message:

```
Conversion processing ...
```

is displayed while a temporary work file of data conversion records is produced on the main database unit. The baseline message:

```
Updating new database ...
```

is then displayed and the restore process proceeds as documented in section 4.1. Database conversion causes the restore process to proceed more slowly than normal as the data in modified record types is transferred record by record rather than in blocks. Note that if you have arranged to convert your old database to the new format on the same disk unit there must be sufficient disk space on the unit to hold both databases.

The technical details of database conversion are documented in section 4.5 of this chapter.

4.2.3 Database Data Concentration

If you are restoring from a full backup, you have the opportunity to specify data concentration. Once you have replied to the Conversion Dictionary prompt the following baseline prompt is displayed:

```
    Data Concentration required ? Y/N :
```

Reply Y to carry out data concentration. In this process records are transferred to the new database on a record by record basis to ensure that the new database is as compact as possible.

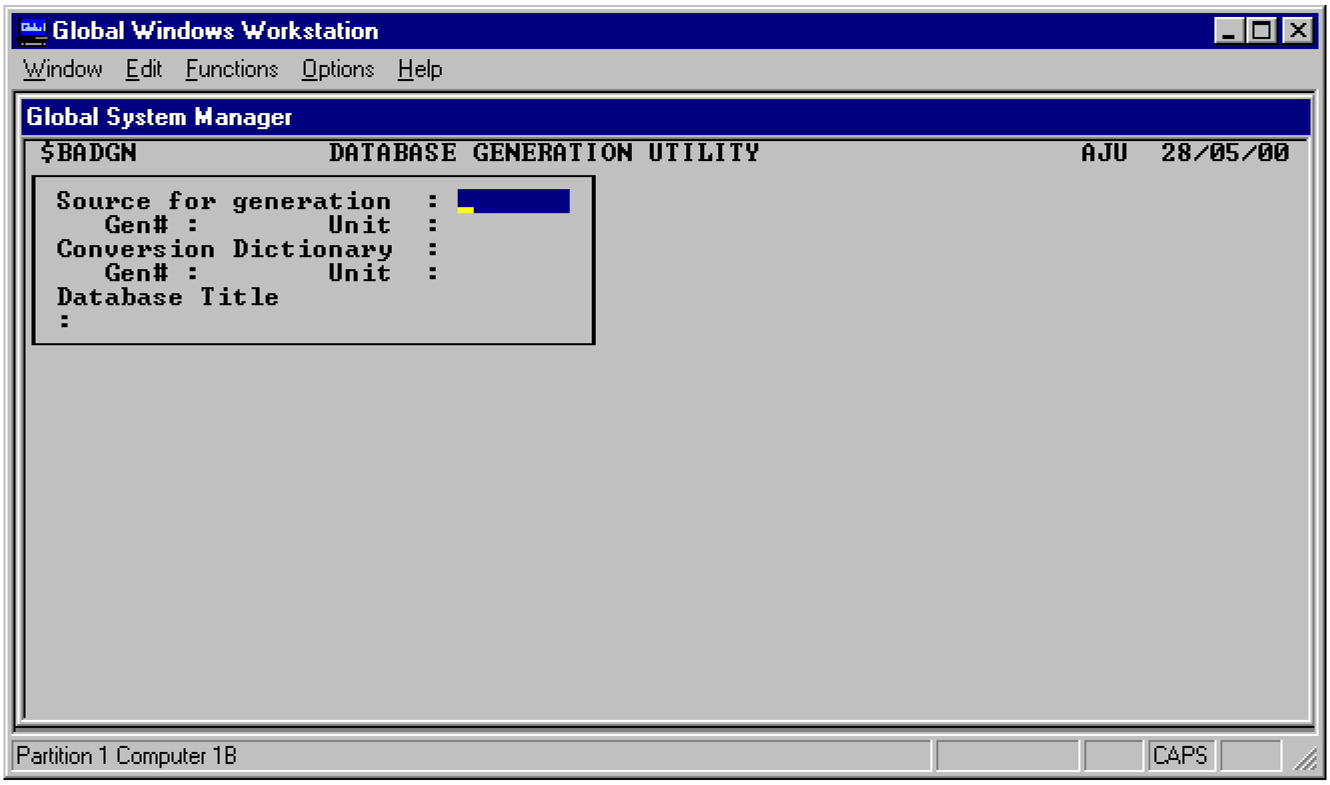
4.2.4 File Allocation During the Restore Process

Once you have replied to the Data Concentration baseline prompt the file allocation windows are displayed. You now have an opportunity to change the maximum number of records allocated to a record type. At the same time you may change the datafile in which the records of that type are stored. This is done in the same way as for the initial database set-up, described in section 4.3, except that the default number of data records is equal to that specified for each record type on the original database. Key <NXT> to complete this allocation process.

You now have an opportunity to change the database name and file allocation details. To choose a new database name simply key the new name. The size of the main index file may be modified, and its disk unit. You may also modify the disk unit of the three datafiles. Having completed the changes, key <RET>.

4.3 Generating a New Database

To run the generation utility and generate a new database, key 4 at the main menu. The utility displays the window:


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BADGN              DATABASE GENERATION UTILITY                     AJU   28/05/00
 Source for generation  :
    Gen# :         Unit  :
 Conversion Dictionary  :
    Gen# :         Unit  :
 Database Title
 :
```

Figure 4.3a - The Generation Input Window

Enter the source for the generation. Since you are creating a new database you should reply:

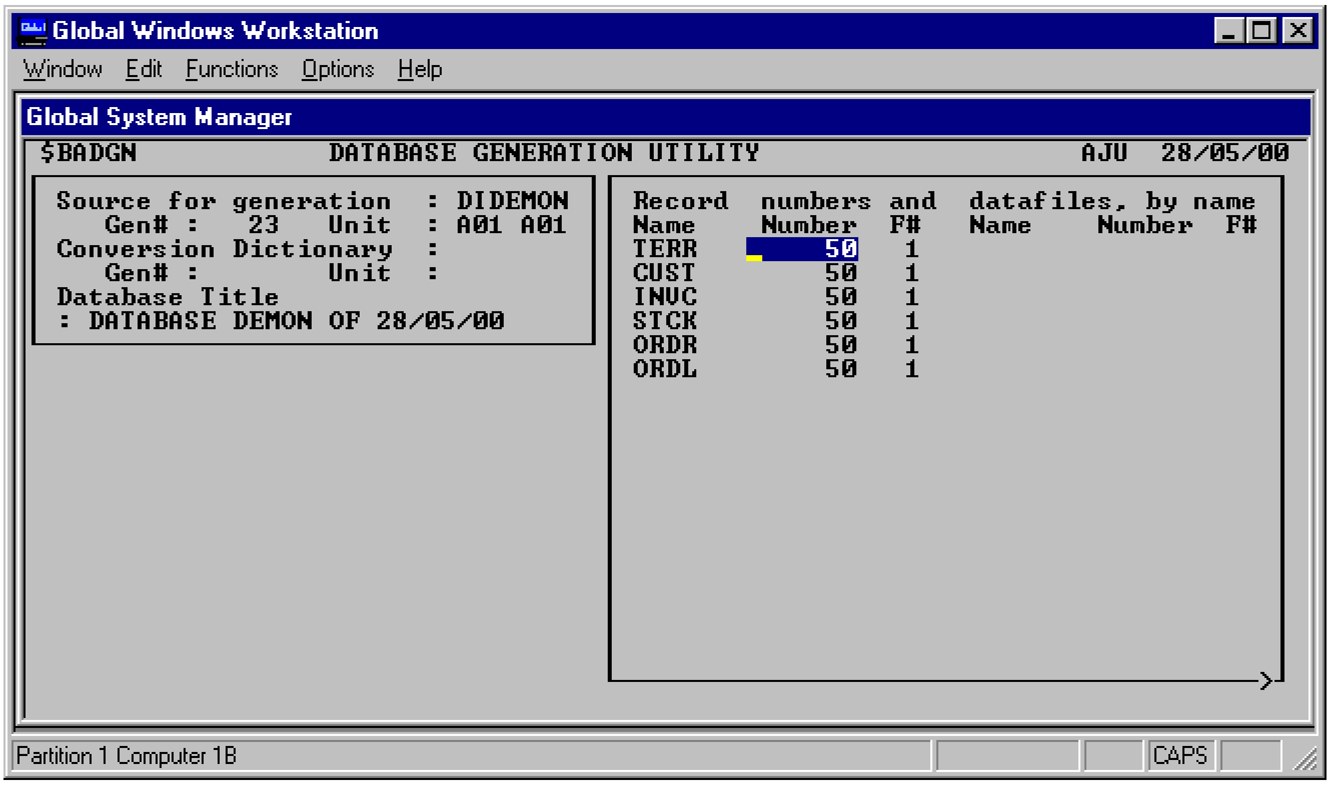
DI*xxxxx*

where xxxxx is the database name. You next enter the disk unit holding the dictionary (e.g. 100). This dictionary is now used to structure the new database, and is later copied to the main database unit and renamed, if necessary, to DIyyyyy where yyyyy is the database name you will specify.

You should now enter the database title, which has a maximum of thirty characters. If you wish to use the default title, simply key <RET>. The default title is of the form:

DATABASE *xxxxx* OF *dd/mm/yy*

where xxxxx is the database name derived from the dictionary and dd/mm/yy is today's date. The generation utility now displays the record allocation window, see Figure 4.3b.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
$BADGN          DATABASE GENERATION UTILITY               AJU  28/05/00

 Source for generation   : DIDEMON     Record  numbers and  datafiles, by name
    Gen# :   23   Unit   : A01 A01     Name    Number  F#   Name    Number  F#
 Conversion Dictionary   :             TERR        50   1
    Gen# :        Unit   :             CUST        50   1
 Database Title                        INVC        50   1
 : DATABASE DEMON OF 28/05/00          STCK        50   1
                                       ORDR        50   1
                                       ORDL        50   1
```

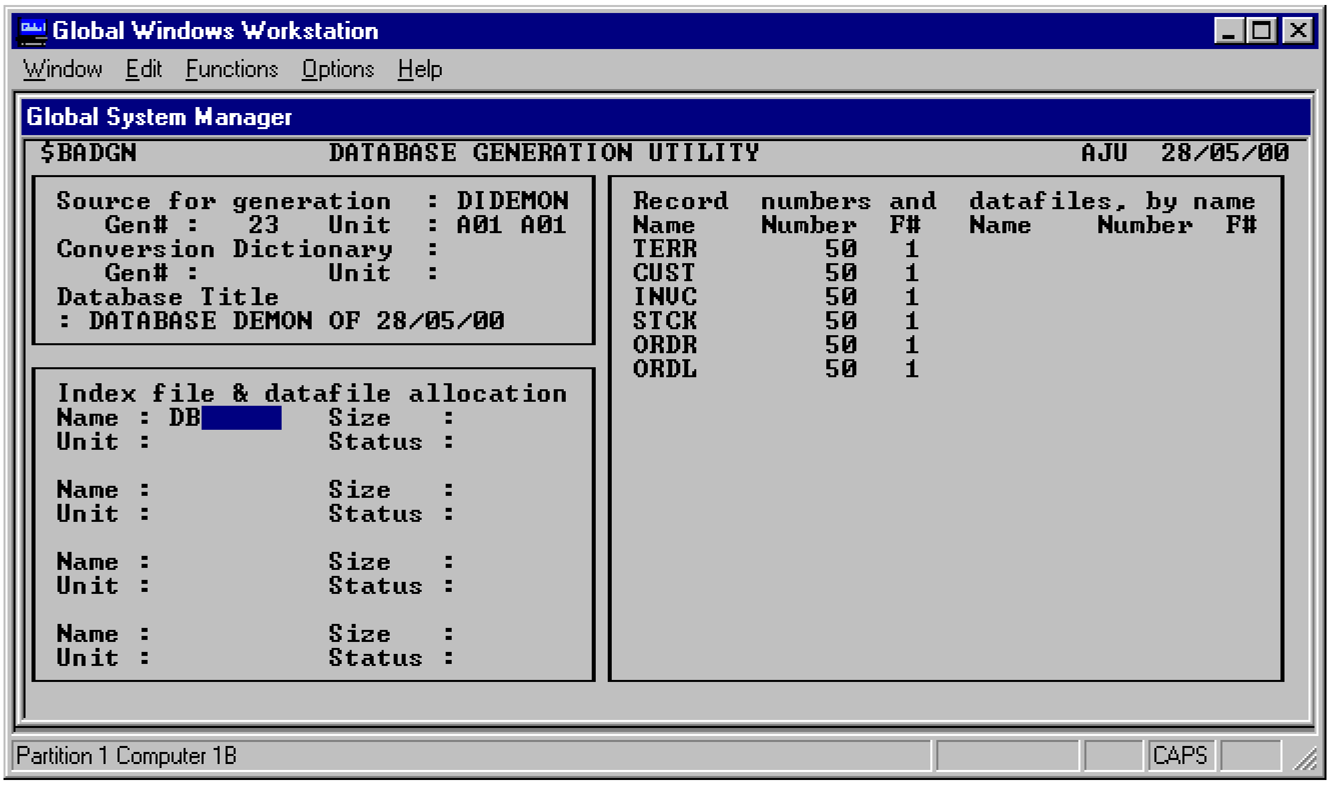
Partition 1 Computer 1B
CAPS

Figure 4.3b - The Record Allocation Window

The cursor is at the first record type in the window used to specify record numbers and datafiles. If you wish to accept the default of fifty records and datafile 1 for each record type, simply key <NXT>. If you wish to change the number of records or the datafile for a particular record type, move to it using the cursor keys, key <RET>, and make the changes you require. When this process is complete, key <NXT>. Note that the maximum number of records of any record type is 8,388,350 and that you may store the records of each record type in one of three datafiles.

The default number of records is set according to the type of generation being performed. If a dictionary was used as input, then a default of 50 will be provided. If a Global format Speedbase database was used as input, then the default record number will be equal to that source database. Where a Unix format database was provided, the number of records will be set to 1.3 times the actual number of records stored in the source database (subject to a minimum of 50 records). Note that for some applications these default values may need to be increased to allow space for future expansion.

The utility displays the file allocation window, see Figure 4.3c.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BADGN              DATABASE GENERATION UTILITY                AJU   28/05/00

 Source for generation   : DIDEMON    Record  numbers and  datafiles, by name
    Gen# :   23   Unit   : A01 A01    Name    Number  F#   Name    Number  F#
 Conversion Dictionary   :            TERR        50   1
    Gen# :        Unit   :            CUST        50   1
 Database Title                       INVC        50   1
 : DATABASE DEMON OF 28/05/00         STCK        50   1
                                      ORDR        50   1
 Index file & datafile allocation     ORDL        50   1
 Name  : DB          Size    :
 Unit  :             Status  :

 Name  :             Size    :
 Unit  :             Status  :

 Name  :             Size    :
 Unit  :             Status  :

 Name  :             Size    :
 Unit  :             Status  :
```

Figure 4.3c - The File Allocation Window

Enter the database name, up to five characters and <RET> to accept the default index file size. This has been calculated to allow an index fill rate of 75%, which is considered appropriate for a database in which indexes are generally modified randomly. An absolute minimum file size is also calculated and you will receive an error message if you enter a file size less than this minimum value.

You next enter the disk unit for the main index file, the default being the unit holding the dictionary or input database. The best approach is to use a logical disk assignment (e.g. FLS). The index file is now opened and you proceed to enter the disk units for up to three datafiles, their file sizes having already been calculated by the utility. The default for the datafile disk units is the index file disk unit. The datafiles are opened in turn and the file allocation process is complete.

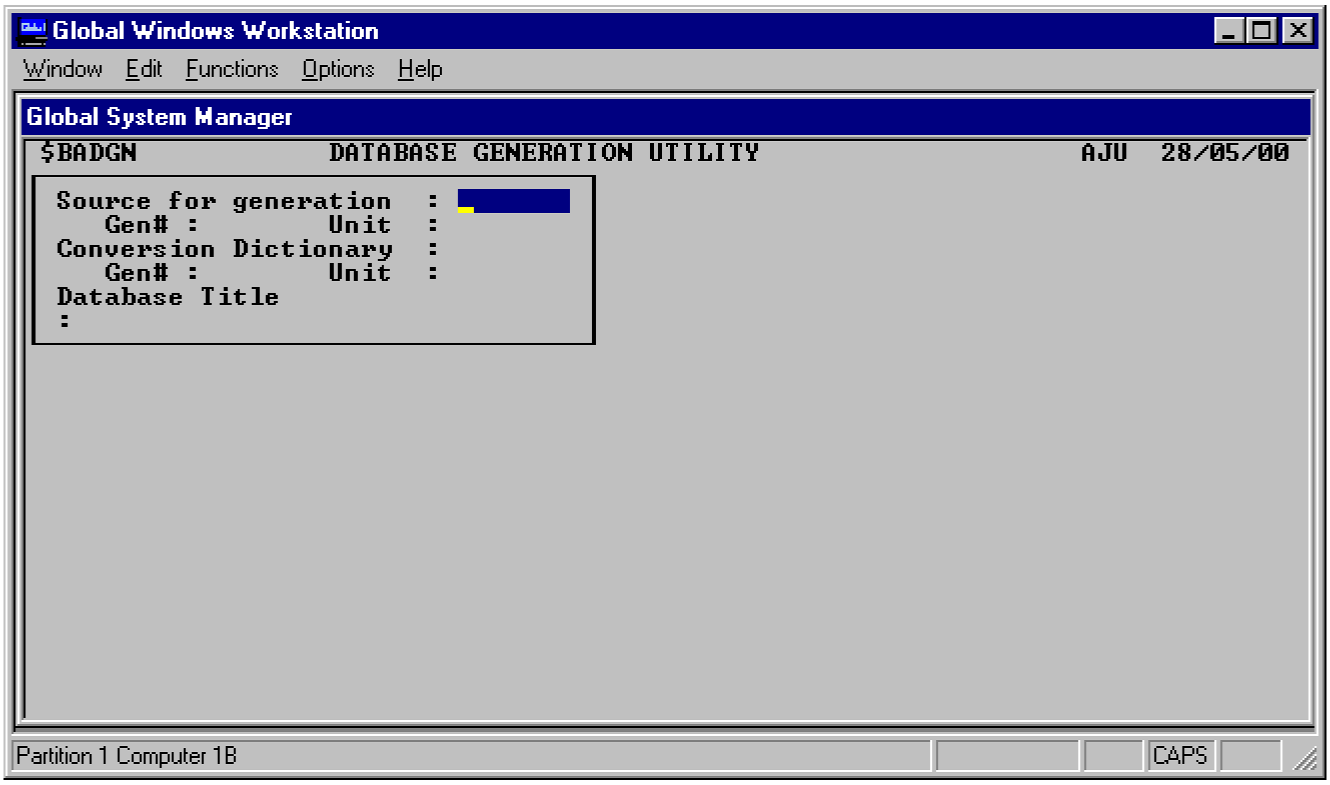
The baseline message:

```
New database created ...
```

is displayed and control passes to the database rebuild utility which automatically completes the database creation process by building the database index file.

4.4 Modifying a Database

The generation utility is also used to copy a Speedbase database, with the option of carrying out data conversion, data concentration and file allocation changes during the process. To run the generation utility and modify an existing database, key 4 at the main menu. The utility displays the window:


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BADGN          DATABASE GENERATION UTILITY                    AJU   28/05/00

 Source for generation   :
    Gen# :          Unit  :
 Conversion Dictionary   :
    Gen# :          Unit  :
 Database Title
 :
```

Figure 4.4a - The Generation Input Window

Enter the name of the source of the generation. Since you are modifying an existing database, you should reply:

```
DBxxxxx
```

where xxxxx is the name of that database. You next specify its disk unit (e.g. FLS). The utility then proceeds to generate the new database as documented in section 4.3. Note that when you reply to the file allocation prompt for the database name you may specify a new name and disk unit. If you do not do so the new database replaces the original. When the utility displays the baseline message:

```
Updating new Database ...
```

it transfers the data from the original to the new database. It then displays the baseline message:

```
New database created ...
```

and passes control to the database rebuild utility which automatically completes the database modification process by building the new database index file.

4.5 Database Conversion

This section documents the database conversion process. Database conversion is the process whereby the structure of a database, as defined by its database dictionary, is changed.

BEFORE PERFORMING A CONVERSION, TAKE A BACKUP OF THE DATABASE.

The essential components required to perform database conversion are a database and a modified dictionary, termed the conversion dictionary. The database may be on-line or in the form of a backup, and the conversion dictionary will have been produced using the RGN option of the Speedbase Development Language compiler. The actual dialogue involved in the conversion process is documented in section 4.2.2 of this manual.

The generation utility reads both the original and the conversion dictionaries to produce the following:

- A mapping of record type to record type to determine whether any have been added or deleted;

- A work file containing details of all data to be transferred or converted between the two databases.

In producing this work file, the utility determines whether a particular record type is in fact being converted and whether it is possible to buffer I/O operations in order to enhance performance. It also determines whether changes to the database necessitate a total rebuild of master/servant record linkages.

During database conversion, record types may be added, deleted or linked to different masters. The correspondence between a record type in different dictionaries is the two character record-id specified in the frame ACCESS statement. Data fields may be added, deleted or modified, in type, size or format. Any data fields added will not be updated with data from the input database.

Four types of data field exist in a Speedbase frame. They are set out as follows, together with the data types to which they may be converted:

Input Field	**Converted Field**
Computational Numeric	Computational Numeric or Date
Alphanumeric	Alphanumeric
Date	Date or Computational Numeric
GVA	GVA or Computational Numeric

Table 4.5a - Conversion Options

If conversion of a data field would result in illegal format conversion, no data is transferred and the output field is initialised according to its picture clause. For example, if you tried to convert a PIC 9(4) field (all numeric fields are computational) to PIC X(2), the latter field would be initialised to spaces. If you tried to convert in the opposite direction, the numeric field would be initialised to binary zeros. This rule also applies to date and GVA fields. If you reduce the capacity of a numeric field and overflow occurs, you will receive an appropriate error message and the field will be initialised to binary zeros.

4.6 Destructive Rebuild Facility

You may, from time to time, need to regenerate large databases. This may be necessary, for example, to upgrade a database to a new generation. While diskettes could be used, this would be prohibitively time-consuming.

An alternative is to back up the database to a hard disk unit using the $BASAV utility. The backing up process condenses the database to a relatively compact format, which typically needs around half the space to store. Thus you would need 1.5 times the total database space in order to perform the regeneration.

However, if free disk space if very limited, you may not be able to do this. You may be in the situation where the only sufficiently large disk unit already holds the source database, and no space is available elsewhere. To help with this situation, the $BADGN generation utility has a Destructive Rebuild facility.

The Destructive Rebuild facility causes the input Database to be progressively deleted during the regeneration process. This option is only useful if the input and output databases both reside on the same unit, since this allows space released from the old (superseded) database to be used to construct its replacement.

You should note that in the event that this process cannot complete (for example due to a power failure) the database will be lost. It is therefore essential that you have a current backup before considering use of this option.

4.6.1 Performing a Destructive Rebuild

In order to perform a destructive rebuild, you enter the characters "%DR" in response to the Generation Input Window (Fig 4.3a) Source For Generation prompt. The utility will then ask you to confirm that you want to perform a destructive rebuild, to which you reply "Y".

The utility then re-prompts the Source For Generation, where you now enter the source database name and unit. Note that you must specify an on-line database in the form DBxxxxx; a back-up format database cannot be processed using this option. Further data entry and displays are then the same as for a normal database generation; see section 4.3 for more details.

However, processing is markedly different. The utility proceeds through the following steps:

1. The indexes of the input database are discarded, causing the Main Index file of the input database to be reduced to a size of 50K Bytes or less. Note that this process renders the database unusable by application programs.

2. The utility then creates a "stub" index file and the data-files for the new database. The Stub Index file contains header information only, and is thus only 512 bytes long.

3. Data is then transferred from the old to the new database.

4. On completion of the data transfer, all files belonging to the old database are deleted.

5. The utility then attempts to extend the Stub Index file to its full length, thus allowing it to reuse space released by the deletion of the remaining input database files.

If the utility is able to create the new index file in step 5 above, then it enters the rebuild stage of the generation process in the normal way. However it is probable that free disk space on the database disk unit has been fragmented either by the above process or earlier activity. In this event the utility will not be able to create the new index file, and will display the following prompt:

```
No room for New Index File - Retry (Y/N)?
```

If this occurs you should now concentrate the database unit using the System Manager $REORG or $F utility using another screen or partition. Once this has been done you key "Y" in response to the above prompt. This procedure should make sufficient space available to allow the regeneration process to complete successfully. In the event that this does not release sufficient space, your only recourse is to enlarge the specified unit using $REORG, or to install larger disks. Having done this, you may then run the Rebuild utility $BARBL, which will automatically extend the main index file to its proper size.

4.7 Operating Considerations

The generation utility requires exclusive access to all the database files, so none may be in use by another user or background job etc. When database conversion is performed, a temporary work file is used and this will be deleted when the utility terminates, whether successfully or not. The output data dictionary is always renamed and copied to the main output unit and when renamed at the end of processing may, in some circumstances, replace a duplicate dictionary.

Problems which arise during database conversion (e.g. the modification of a primary index resulting in all records of that type having a duplicate key) will not be recognised by the generation utility. Instead, this type of error will cause error messages to be displayed by the rebuild utility when rebuilding the index structure.

Concentration of a database causes data to be transferred on a record by record basis rather than in whole blocks, thereby increasing processing time. No concentration is allowed from a backup cycle containing incremental backups. This is because the records in full incremental and part incremental backup files have appended to them a record number to allow them to be replaced at the appropriate location in the database. This record number is invalidated by database concentration. Note that concentration is automatic when transferring from a Unix C-ISAM format Speedbase database to a Global format Speedbase database

Database conversion processing will attempt to transfer data in whole blocks wherever possible. However, when the layout of a record type has been modified, the generation utility is forced to build up each record on a field by field basis and then transfer each record individually to the output database. Database conversion is described in section 4.5.

The generation utility uses memory in several stages. A program of approximately 30 Kbytes must be loaded in the first stage, in addition to which an I/O buffer is required to transfer the dictionary to the main output unit. Therefore, this initial stage requires a minimum user memory area of approximately 32 Kbytes. In subsequent stages of processing the memory requirements are considerably reduced. Of this minimum user area of 32 Kbytes, an I/O buffer of approximately 10 Kbytes is used while writing the conversion work file.

The final stage, consisting of the actual data transfer, requires 13 Kbytes of buffer. The buffer in this final stage must be capable of containing one complete data record (of any record type). During conversion processing this buffer must contain at least one input record, one output record, and an area to store internally produced conversion logic. This latter area is generally very small and therefore the minimum buffer requirement during conversion is basically that of the input and output record.

5. Database Rebuild Utility ($BARBL)

5.1 Partial and Total Rebuilds

If your Speedbase database is stored in the Global file structure, the database rebuild utility is used to perform two main functions. It may be used to re-organise indexes only, known as a partial rebuild. Alternatively it may be used to re-establish all linkages within the database, known as a total rebuild.

5.1.1 Partial Database Rebuild

A partial database rebuild is essentially a re-organisation function. It causes all indexes stored in the database to be re-created, which guarantees index accuracy and normally reduces the disk space used for index storage. 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.

5.1.2 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 indexes 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 (such an invalid account balance, or an invoice referencing to 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 loss of power) or from extremely serious program errors. The utility guarantees the internal integrity of the database following such errors, but some application programs may require further processing to be performed.

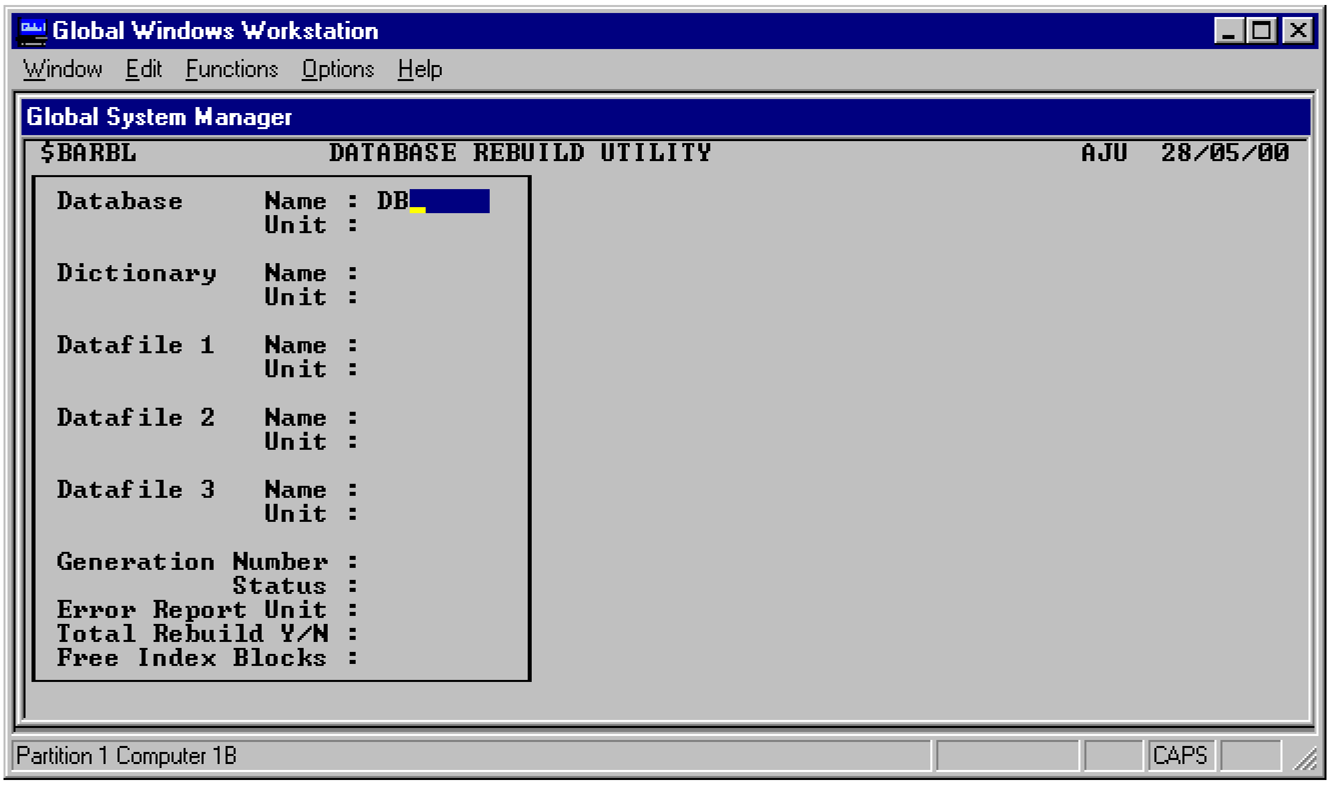
The rebuild utility operates by examining the data records physically stored in the database disk files. This means that all information up to and including the last successfully completed database update can normally be rebuilt. The utility cannot be used, however, if the disk on which the database resides is lost or develops permanent errors. In this event the only recourse is to resort to backup, as described in Chapters 3 and 4 of this manual.

During normal use, Speedbase 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.

When a database is created or re-generated using the database generation utility, the rebuild utility is automatically run. This occurs in order to produce the necessary index structures, and under some circumstances will also re-link elements of the database. When control is passed to it by the generation utility, operation of the rebuild utility is entirely automatic.

5.2 Rebuilding a Database

To run the rebuild utility key 5 at the main menu. The utility displays the window:

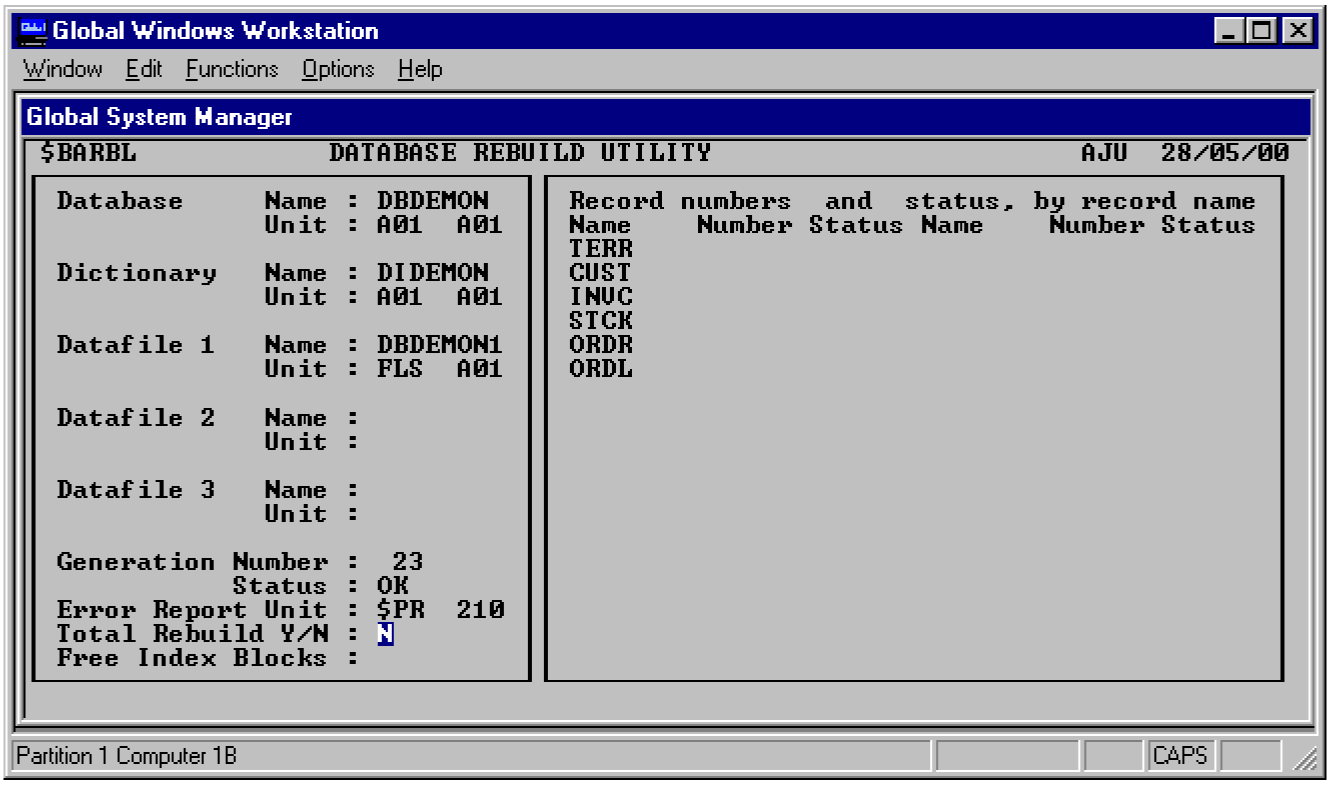

Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BARBL              DATABASE REBUILD UTILITY              AJU   28/05/00

  Database          Name  : DB
                    Unit  :

  Dictionary        Name  :
                    Unit  :

  Datafile 1        Name  :
                    Unit  :

  Datafile 2        Name  :
                    Unit  :

  Datafile 3        Name  :
                    Unit  :

  Generation Number       :
                Status    :
  Error Report Unit       :
  Total Rebuild Y/N       :
  Free Index Blocks       :
```

Figure 5.2a - The Rebuild Utility File Window

Enter the name of the database to be rebuilt (e.g. DBDEMON) and its disk unit (e.g. FLS). Provided the database is found and is not in use by another operator, details relating to the opened database are displayed in the file window and details of the records on the database in the record display window, see Figure 5.2b.

The utility produces an error report during the rebuild process which details any problems found. This report will be printed to the Error Report Unit shown in the window. The file name of the error file is ERxxxxx where xxxxx is the database name. If, for any reason, this file cannot be opened, then errors will be reported on the screen.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BARBL            DATABASE REBUILD UTILITY                  AJU   28/05/00

Database      Name : DBDEMON          Record numbers  and  status, by record name
              Unit : A01   A01        Name     Number Status Name     Number Status
                                      TERR
Dictionary    Name : DIDEMON          CUST
              Unit : A01   A01        INVC
                                      STCK
Datafile 1    Name : DBDEMON1         ORDR
              Unit : FLS   A01        ORDL

Datafile 2    Name :
              Unit :

Datafile 3    Name :
              Unit :

Generation Number :  23
           Status : OK
Error Report Unit : $PR   210
Total Rebuild Y/N : N
Free Index Blocks :
```

Figure 5.2b - The Database Rebuild Window

You should now reply to the Total Rebuild prompt. Valid responses are Y for a total rebuild of the database or N for a rebuild of the index structure only. If the database is corrupt, the Status will be shown as BAD rather than OK and a total rebuild (i.e. a response of Y) is mandatory.

The database rebuild will then proceed, and this is confirmed by one of the baseline messages:

```
Total database rebuild in progress
```

or:

```
Index re-organisation in progress
```

During the rebuild process, various details are displayed in the file window. Free Index Blocks displays the number of unused index blocks remaining while creating the index structures. The number of free index blocks can be seen to decrease as they are progressively used.

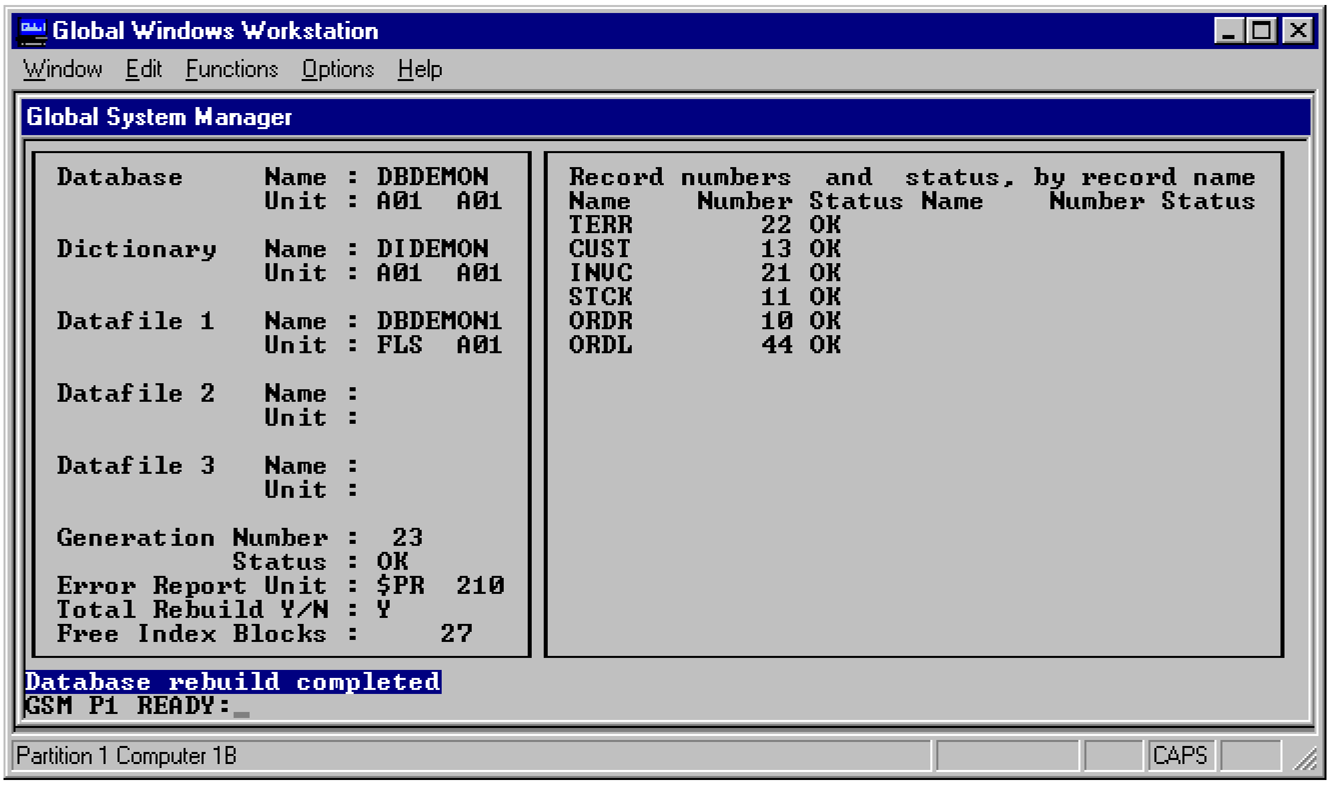
The number of records processed so far for each record type is displayed in the records window. A short status message is also displayed for each record type under the heading Status. The status is one of the following:

RELINK	is displayed whilst the data records are read and optionally relinked to their master records.
ERRORS	is displayed if data in this record type is corrupt and could not be correctly rebuilt.
Xnncyy	shows details of the index rebuilding process where nn is the number of the index currently being created, and yy is the number of index chains currently being merged together to form the index tree.

Completion of the rebuild process is indicated by the baseline message:

```
Database rebuild completed
```

Key <RET> to return to the main menu. See Figure 5.2c.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
Database       Name : DBDEMON        Record numbers  and  status, by record name
               Unit : A01   A01      Name     Number Status Name     Number Status
                                     TERR         22 OK
Dictionary     Name : DIDEMON        CUST         13 OK
               Unit : A01   A01      INVC         21 OK
                                     STCK         11 OK
Datafile 1     Name : DBDEMON1       ORDR         10 OK
               Unit : FLS   A01      ORDL         44 OK

Datafile 2     Name :
               Unit :

Datafile 3     Name :
               Unit :

Generation Number :  23
           Status : OK
Error Report Unit : $PR  210
Total Rebuild Y/N : Y
Free Index Blocks :      27

Database rebuild completed
GSM P1 READY:_
```

Figure 5.2c - Database Rebuild Completed

5.3 Operating Considerations

5.3.1 Processing Time

The processing time required to perform a total rebuild is considerably longer than that required to reorganise the indexes only, especially on large databases. Whereas a partial rebuild only needs to process each data record once, a total rebuild must read not only every record, but also each associated record. A total rebuild will therefore take several times longer than a partial rebuild.

5.3.2 Exclusive Use

The rebuild utility requires exclusive use of the dictionary and database and will not proceed if any of these files are in use by another operator.

5.3.3 Incremental Backups

A total rebuild notionally updates all records in the database. An incremental backup performed following a total rebuild will therefore cause **all** records residing on the database to be copied to the backup file. While this will not cause any errors, it will take considerably longer than performing a full backup, and will certainly extend the time needed to restore the database, should this become necessary.

For these reasons it is therefore recommended that a full, rather than incremental backup is performed following a total database rebuild.

5.3.4 Memory Requirements of the Rebuild Process

This section covers the memory requirement of the rebuild program, and has been provided for the guidance of application product designers.

The rebuild utility requires an absolute minimum partition size of 30 Kbytes, but will make use of larger areas if available. The utility requires approximately 22 Kbytes to load, the remaining space in the partition being used as work space for sorting and linking operations to take place. For example, if a 38 Kbyte partition is available, the work space will be 16 Kbytes. Larger partitions will naturally provide more work space.

Work space is reallocated as each record type stored on the database is processed in turn. For each record type, it must accommodate the following three areas:

Index Buffers	One 512 byte buffer is required for each index associated with the current record type. Thus if a record has 4 indexes, 2 Kbytes would be required.
Master Record Area	During a total rebuild, each master record linked to the current record type is read and processed. This area therefore needs to be large enough to accommodate the largest master record. Thus if a record has 3 masters, with lengths of 0.5 Kbytes, 0.8 Kbytes, and 1.4 Kbytes then 1.4 Kbytes will be required.
Current Record Area	Records of the current record type are read into this area. Normally records will be read in blocks of hundreds, but it must be at least large enough to allow a single record to be read at a time.

These memory requirements are best illustrated using an example. Let us take a large record, say 4 Kbytes, which has 12 indexes, and is linked to 6 master records the largest of which is 3 Kbytes. The work space requirement to process this record is therefore as follow:

12 Index Buffers at 0.5 Kbytes each	6 Kbytes
Largest Master Record Area	3 Kbytes
Current record area	4 Kbytes
Total **minimum** work space required	13 Kbytes

The absolute minimum work space required for this record to be processed is therefore 13 Kbytes. Given that the rebuild utility uses a further 22 Kbytes, the minimum partition size is therefore 35 Kbytes. As can be seen from this example, memory requirements are unlikely to present a practical problem unless ridiculously large record sizes are in use.

Note that, unlike certain sort operations, the memory requirements of the rebuild program do not vary with the number of records being processed. Thus, if there is any doubt that these limitations could be exceeded, we recommend that a database is established containing one record of each type. If the rebuild utility succeeds with this small test, it will operate on any database with the same structure, irrespective of the actual number of records stored.

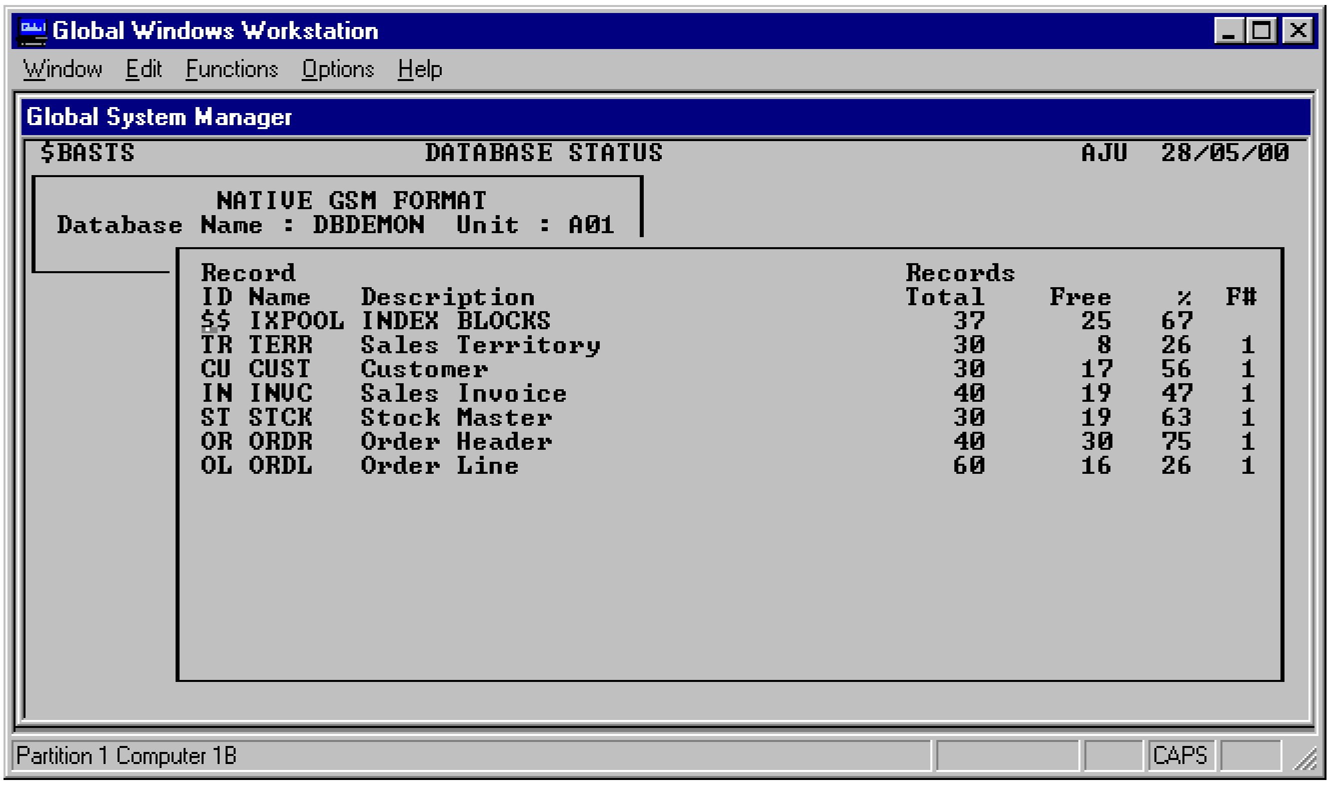
6. Database Status Utilities ($BAST & $BASTS)

Two status utilities are provided. $BASTS provides a single screen report on the database, and is designed for use by end-users of database systems. The second status utility $BAST provides far greater detail and has been principally been designed for use in program development.

These two utilities are described in detail in the following sections.

6.1 Database Status Utility $BASTS

This status utility displays details of the records stored on your database.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BASTS                  DATABASE STATUS                    AJU   28/05/00

        NATIVE GSM FORMAT
 Database Name : DBDEMON   Unit : A01

          Record                                      Records
          ID Name    Description                      Total     Free     %   F#
          $$ IXPOOL  INDEX BLOCKS                       37        25    67
          TR TERR    Sales Territory                    30         8    26    1
          CU CUST    Customer                           30        17    56    1
          IN INVC    Sales Invoice                      40        19    47    1
          ST STCK    Stock Master                       30        19    63    1
          OR ORDR    Order Header                       40        30    75    1
          OL ORDL    Order Line                         60        16    26    1
```

Figure 6.1a - The Database Status Display

The first line of the display shows the total number of index blocks, the number free and the percentage free. If the percentage of free index blocks is less than 10% you should consider doing a partial rebuild, see Chapter 5, or a database generation, see Chapter 4.

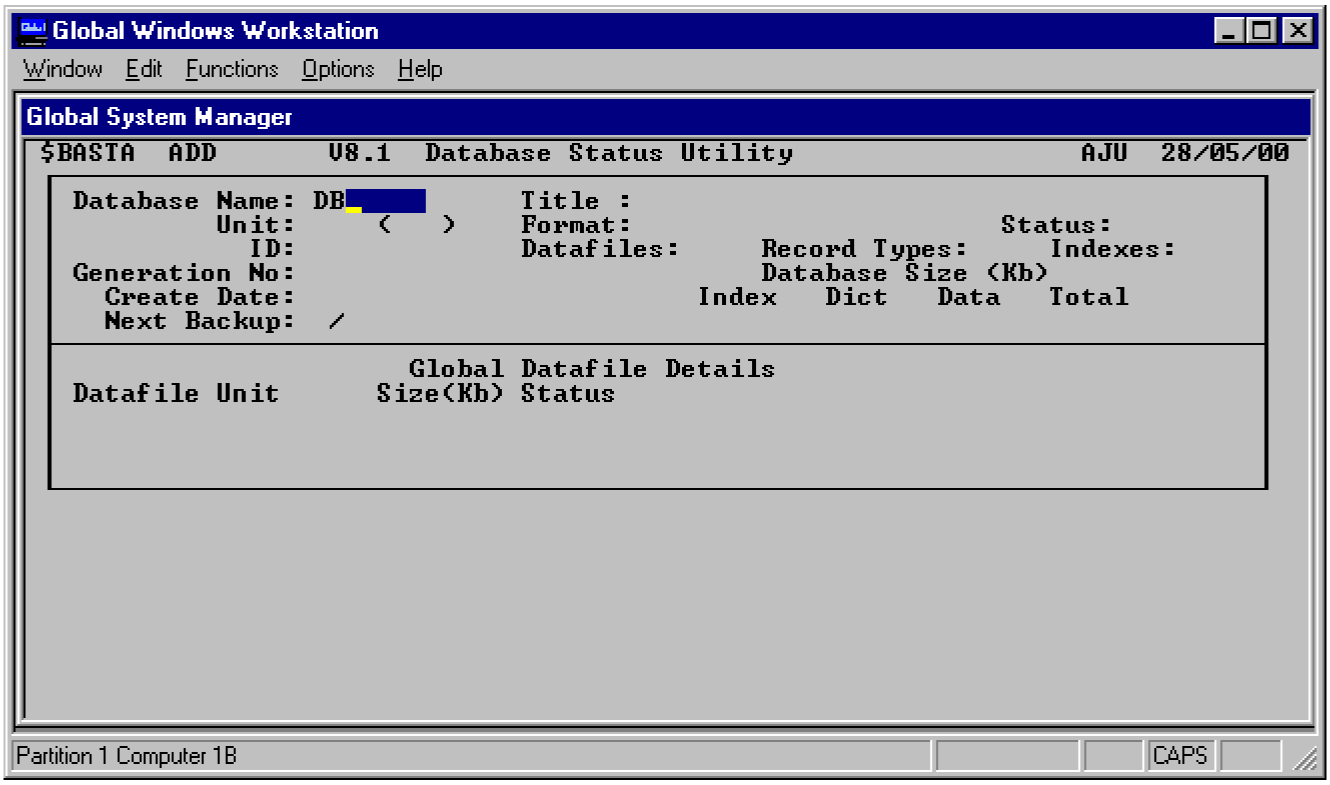
For each record type the window displays the record description, the total number of records that may be stored of that type and the number free. It also shows what percentage the free records are of the total number of records. For example, if the total number of customer records is 2000 and 450 are free, this represents 25% of the customer record space. The window also displays the datafile number for the record type. For example if the record CUST in the sample database DEMON has an F# of 1, its records are to be found in the datafile DBDEMON1.

Note that if your database is stored in a Unix file system, then the columns headed "Total and "%" will contain the literal "N/A".

If you have more than fourteen record types in your database, key <PGE> to see the second page of the display. Key <BPG> to go back one page and <ABO> to return to the database name window.

6.2 Database Status utility $BAST

The following window is displayed after selection from the main menu:


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BASTA  ADD      V8.1  Database Status Utility                AJU  28/05/00

 Database Name: DB           Title :
          Unit:     (   )    Format:                      Status:
            ID:              Datafiles:     Record Types:     Indexes:
 Generation No:                             Database Size (Kb)
   Create Date:                          Index    Dict    Data    Total
   Next Backup:   /

                        Global Datafile Details
 Datafile Unit        Size(Kb) Status
```

Figure 6.2a $BASTA initial Enquiry Screen

You may now enter the Database Name and Unit ID. The utility then displays summary details of the database, with the options menu. These options are:

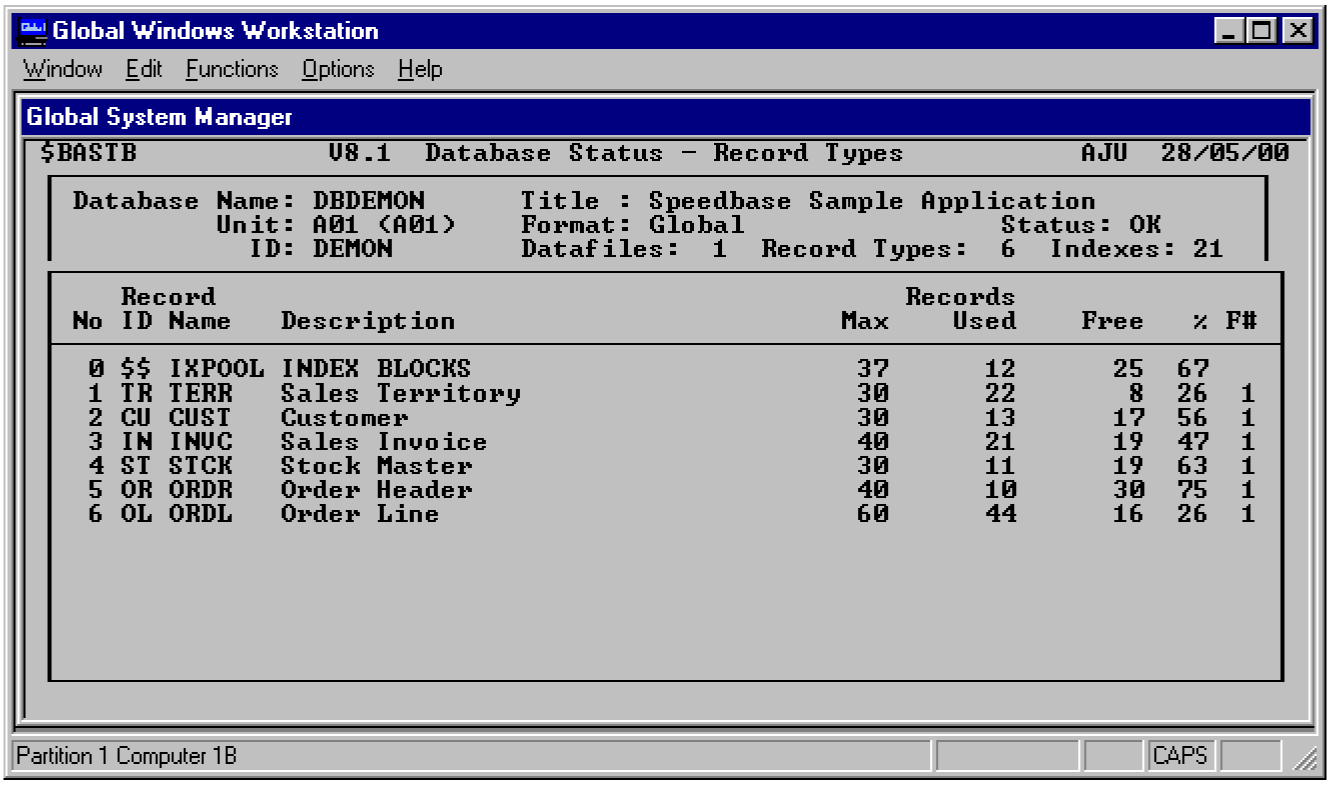
1. Record Type Display: This option displays details of all record types stored on the database.

2. Index Display: This options displays details of all indexes stored on the database.

3. Short Status Report: This option provides a short report of database status, consisting of the database summary and details of each record type stored.

4. Long Status Report: This option provides a more extensive database status report which includes descriptions of all indexes held on the database.

On selection of the Record Type Display (option 1), the following window is displayed:

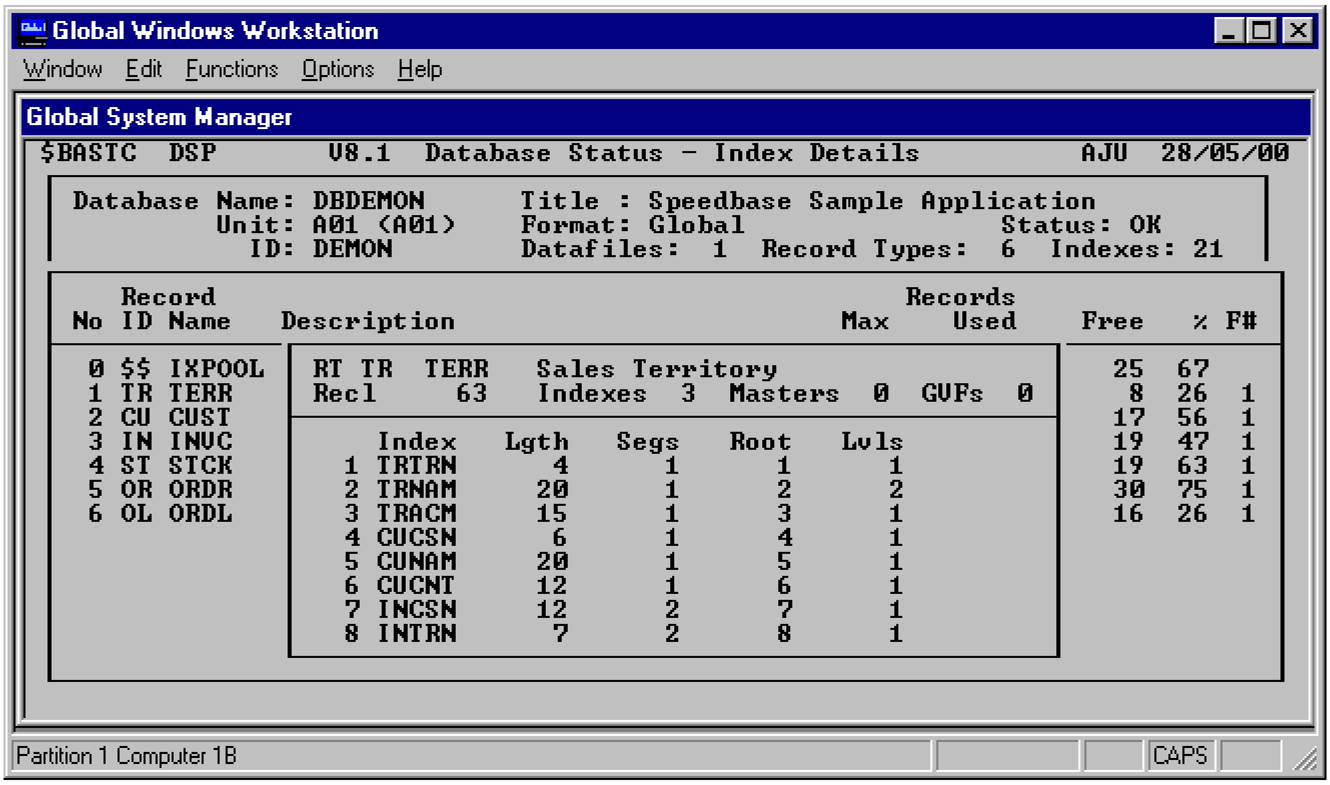

Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
$BASTB             V8.1  Database Status - Record Types              AJU   28/05/00

  Database Name: DBDEMON      Title : Speedbase Sample Application
           Unit: A01 (A01)    Format: Global                Status: OK
             ID: DEMON        Datafiles:  1  Record Types:  6  Indexes: 21

     Record                                               Records
  No ID Name   Description                         Max     Used      Free   %  F#

   0 $$ IXPOOL INDEX BLOCKS                         37       12        25  67
   1 TR TERR   Sales Territory                      30       22         8  26  1
   2 CU CUST   Customer                             30       13        17  56  1
   3 IN INVC   Sales Invoice                        40       21        19  47  1
   4 ST STCK   Stock Master                         30       11        19  63  1
   5 OR ORDR   Order Header                         40       10        30  75  1
   6 OL ORDL   Order Line                           60       44        16  26  1
```

Figure 6.2b Record Type Display Window

This window displays details of all record types stored in the database. This window shows each record ID, Name and Description, as well as the number of records that may be stored, are currently in use, and remain free. Note that no maximum limit exists on databases stored in Unix C-ISAM format, and the Maximum and Free Records fields are therefore only provided for Global format Speedbase databases.

You may select a record type from the display to enquire on the indexes associated with that record type. Selecting a record causes the following window to be displayed:


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BASTC  DSP        V8.1  Database Status - Index Details            AJU   28/05/00

  Database Name: DBDEMON      Title : Speedbase Sample Application
           Unit: A01 (A01)    Format: Global                 Status: OK
             ID: DEMON        Datafiles:  1  Record Types:  6  Indexes: 21

     Record                                              Records
  No ID Name    Description                          Max     Used    Free    %  F#

   0 $$ IXPOOL   RT  TR   TERR    Sales Territory                      25   67
   1 TR TERR     Recl       63    Indexes  3  Masters   0   GVFs   0    8   26  1
   2 CU CUST                                                           17   56  1
   3 IN INVC          Index    Lgth     Segs     Root     Lvls         19   47  1
   4 ST STCK       1  TRTRN       4        1        1        1         19   63  1
   5 OR ORDR       2  TRNAM      20        1        2        2         30   75  1
   6 OL ORDL       3  TRACM      15        1        3        1         16   26  1
                   4  CUCSN       6        1        4        1
                   5  CUNAM      20        1        5        1
                   6  CUCNT      12        1        6        1
                   7  INCSN      12        2        7        1
                   8  INTRN       7        2        8        1
```

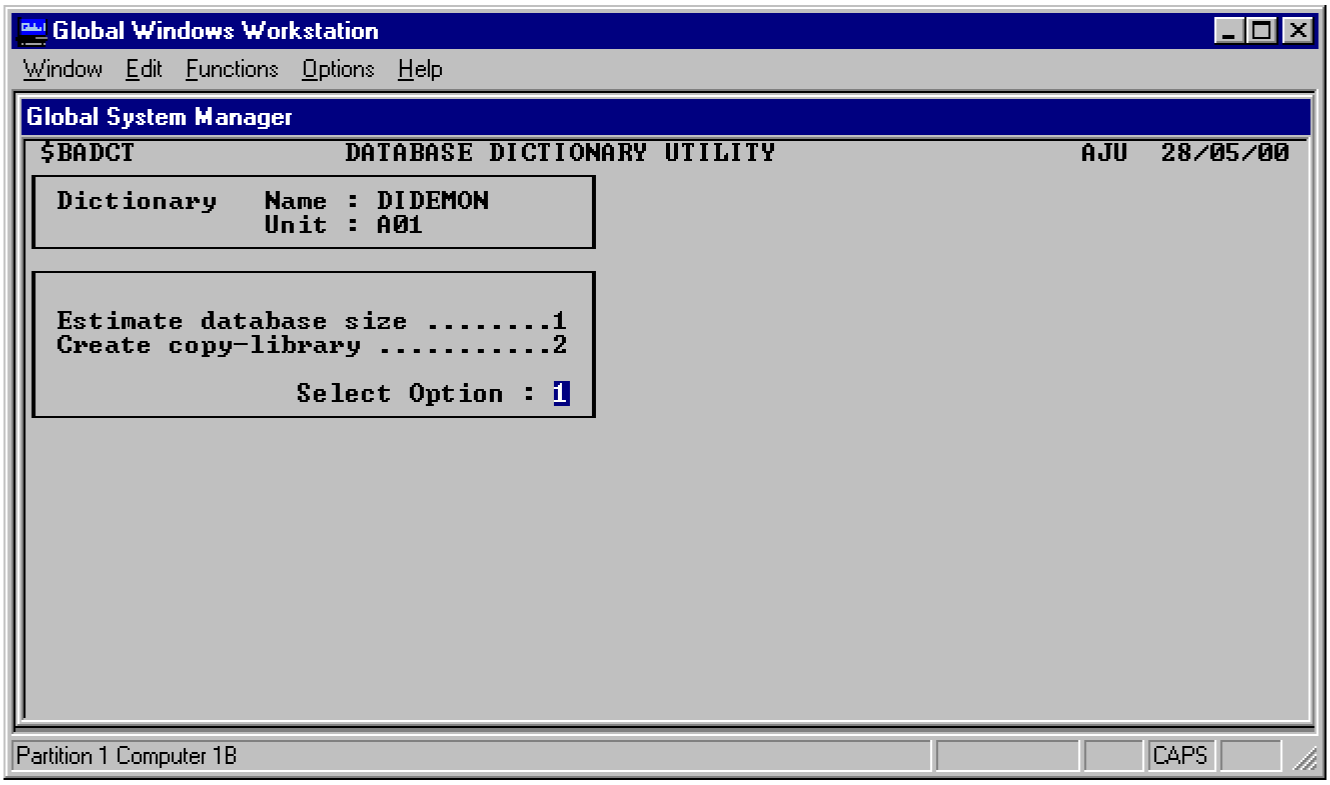
Figure 6.2c The Index Enquiry Window

This window displays details of the indexes applicable to the currently selected record type. (Note that if the Index Pool was selected in the preceding window, then all indexes stored in the database will be displayed.

The Index Enquiry Window displays the Index Name, length and number of segments (fields) making up each index key. The index root block number and number of levels comprising the B-Tree index structure are also displayed.

7. Database Dictionary Utility ($BADCT)

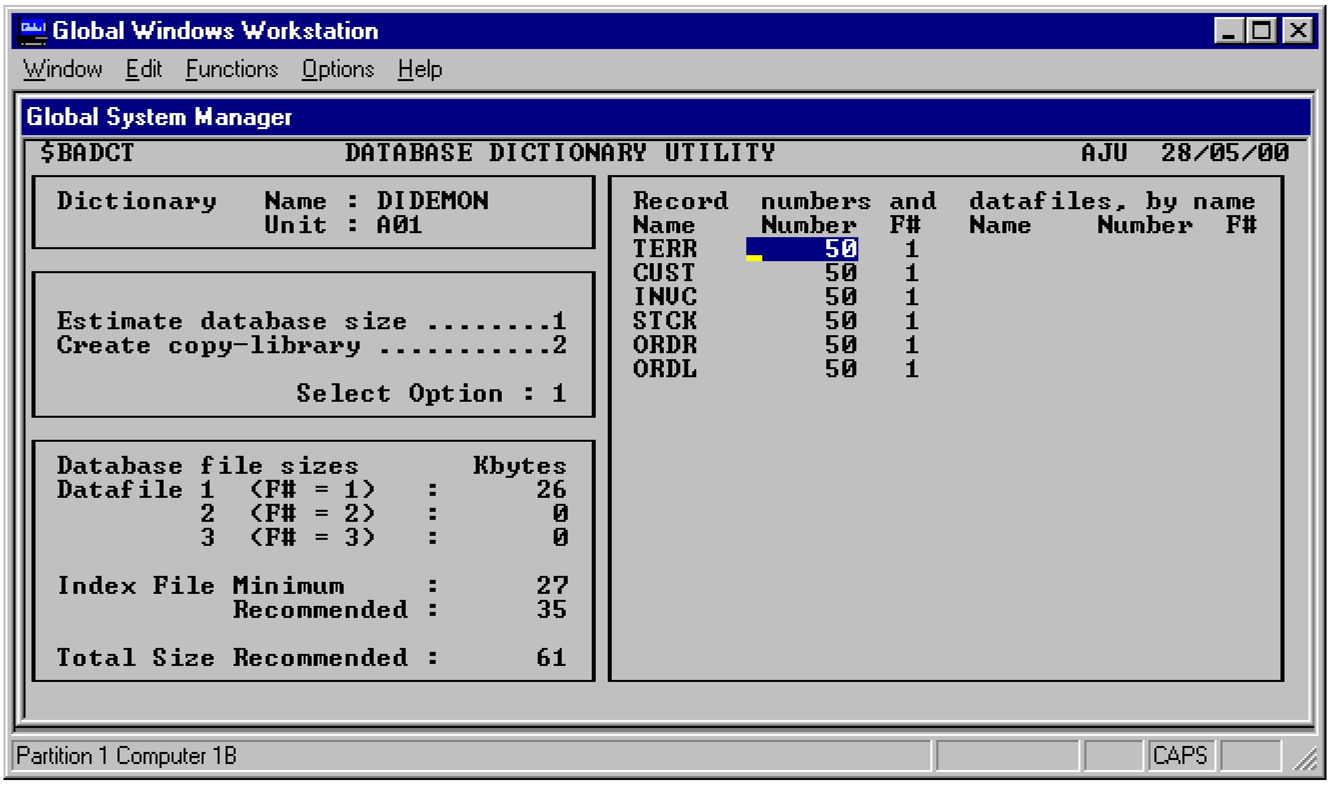
The database dictionary utility performs two separate functions. You may use it to print a listing of your database dictionary. If your database is to be stored in the Global file structure you may use the utility to estimate its size. To run the dictionary utility, key 7 at the main menu. Having replied to the dictionary name and unit prompts, you choose which function to perform by replying to the option menu:


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BADCT               DATABASE DICTIONARY UTILITY              AJU   28/05/00

  Dictionary    Name : DIDEMON
                Unit : A01

  Estimate database size ........1
  Create copy-library ...........2

                Select Option : 1
```

Figure 7a - Choosing a Dictionary Utility Option

If you wish to do a database size estimation, reply 1 to the select option prompt. The utility displays the record and file windows shown in Figure 7b, and places the cursor on the first line of the record window.


Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 Computer 1B
CAPS

```
$BADCT              DATABASE DICTIONARY UTILITY              AJU   28/05/00

 Dictionary    Name : DIDEMON          Record   numbers  and  datafiles, by name
               Unit : A01              Name     Number   F#   Name      Number  F#
                                       TERR         50    1
                                       CUST         50    1
 Estimate database size .........1     INVC         50    1
 Create copy-library ............2     STCK         50    1
                                       ORDR         50    1
                 Select Option : 1     ORDL         50    1

 Database file sizes         Kbytes
 Datafile 1   (F# = 1)    :      26
          2   (F# = 2)    :       0
          3   (F# = 3)    :       0

 Index File Minimum       :      27
            Recommended   :      35

 Total Size Recommended   :      61
```

Figure 7b - Database Size Estimation

To change the number of records or the datafile for a particular record type from the default of fifty records and datafile 1, move to it using the cursor keys, key <RET>, and make the changes you require. Note that the maximum number of records of any record type is 8,388,350 and that you may store record types in one of three datafiles.

The utility calculates the sizes of the three datafiles and the minimum and recommended size of the index file, and displays the calculated file sizes as you make changes to the record specification. Key <ABO> to return to the dictionary name window.

To produce a copy-library listing of your database dictionary, reply 2 to the select option prompt. Once you have replied to the copy library name and disk unit prompts the listing file is written to disk. Key <ABO> to return to the dictionary name window. To print the listing file, use the $PRINT utility.

8. Unix Database Management Utility ($BADB)

If your Speedbase database is stored in a Unix file system the Unix Database Management Utility is used to create your database, to rebuild its indexes, and to transfer data from a database stored in the Global file structure, to dump and regenerate databases, and to delete unwanted databases.

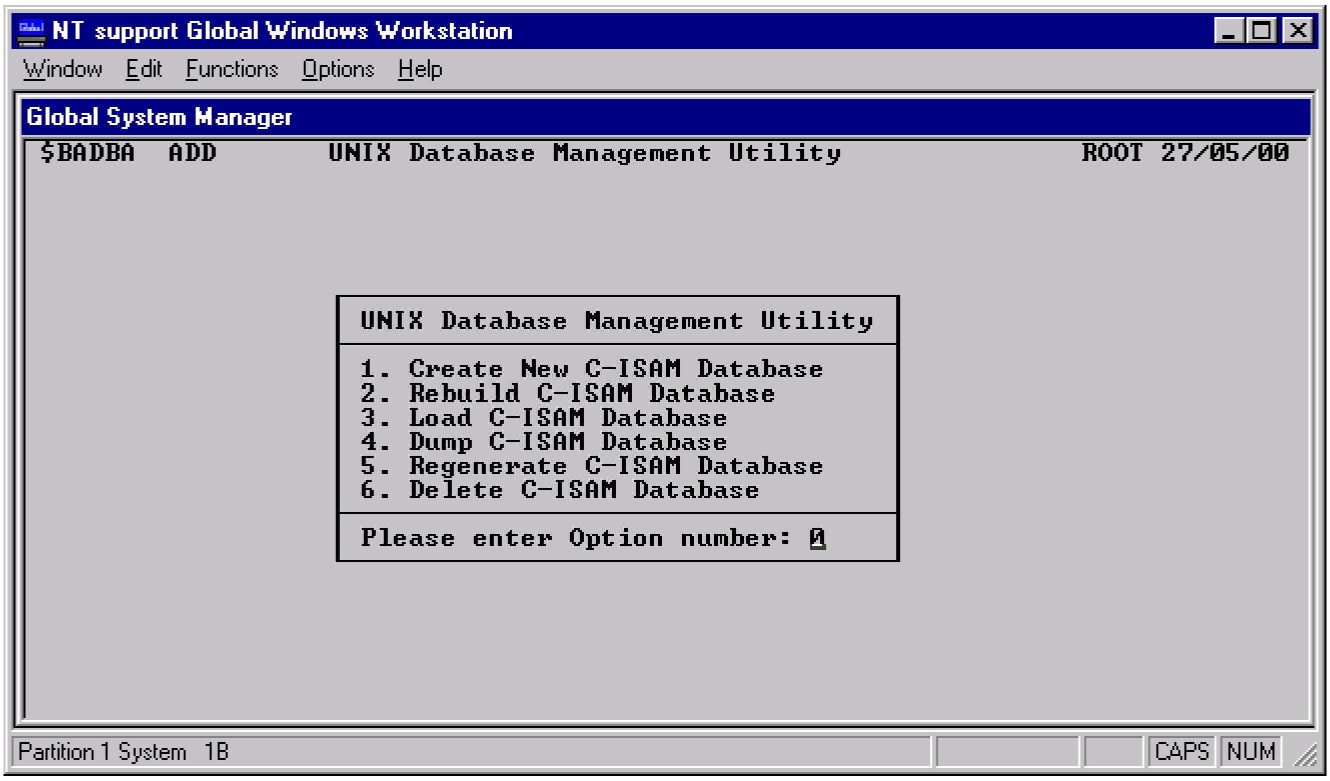
The Speedbase database consists of a data dictionary file and a schema file, both stored in the usual Global file structure, and in addition, a special index file and several datafiles and index files stored in the Unix file structure. The names of the files are as follows, where xxxxx is the database name and rt1 to rtn are up to thirty-six record types:

File-id	**Description**	**Format**
DIxxxxx	Data Dictionary	Global
DBxxxxx	Schema File	Global
DBxxxxx	Special Index File	Unix
DBxxxxxrt1.dat	C-ISAM Data File for record rt1	Unix
DBxxxxxrt1.idx	C-ISAM Index File for record rt1	Unix
DBxxxxxrt2.dat	C-ISAM Data File for record rt2	Unix
DBxxxxxrt2.idx	C-ISAM Index File for record rt2	Unix
DBxxxxxrtn.dat	C-ISAM Data File for record rtn	Unix
DBxxxxxrtn.idx	C-ISAM Index File for record rtn	Unix

Table 8a - Unix Speedbase Database Files

The data dictionary is created by the application developer using the dictionary utility of the Speedbase Development System. The five character database name xxxxx is specified when the database is created using the Unix database management utility, see section 8.1. The C-ISAM datafiles which each store the data of a specific record type are adjusted in size automatically. There is no need to re-allocate the datafiles to provide additional space, as is occasionally required when the database is stored in Global file structure.

To run the Unix database utility key 8 at the main menu. The utility displays the menu:



```
$BADBA   ADD        UNIX Database Management Utility               ROOT 27/05/00

                  UNIX Database Management Utility

                  1.  Create New C-ISAM Database
                  2.  Rebuild C-ISAM Database
                  3.  Load C-ISAM Database
                  4.  Dump C-ISAM Database
                  5.  Regenerate C-ISAM Database
                  6.  Delete C-ISAM Database

                  Please enter Option number: 0
```

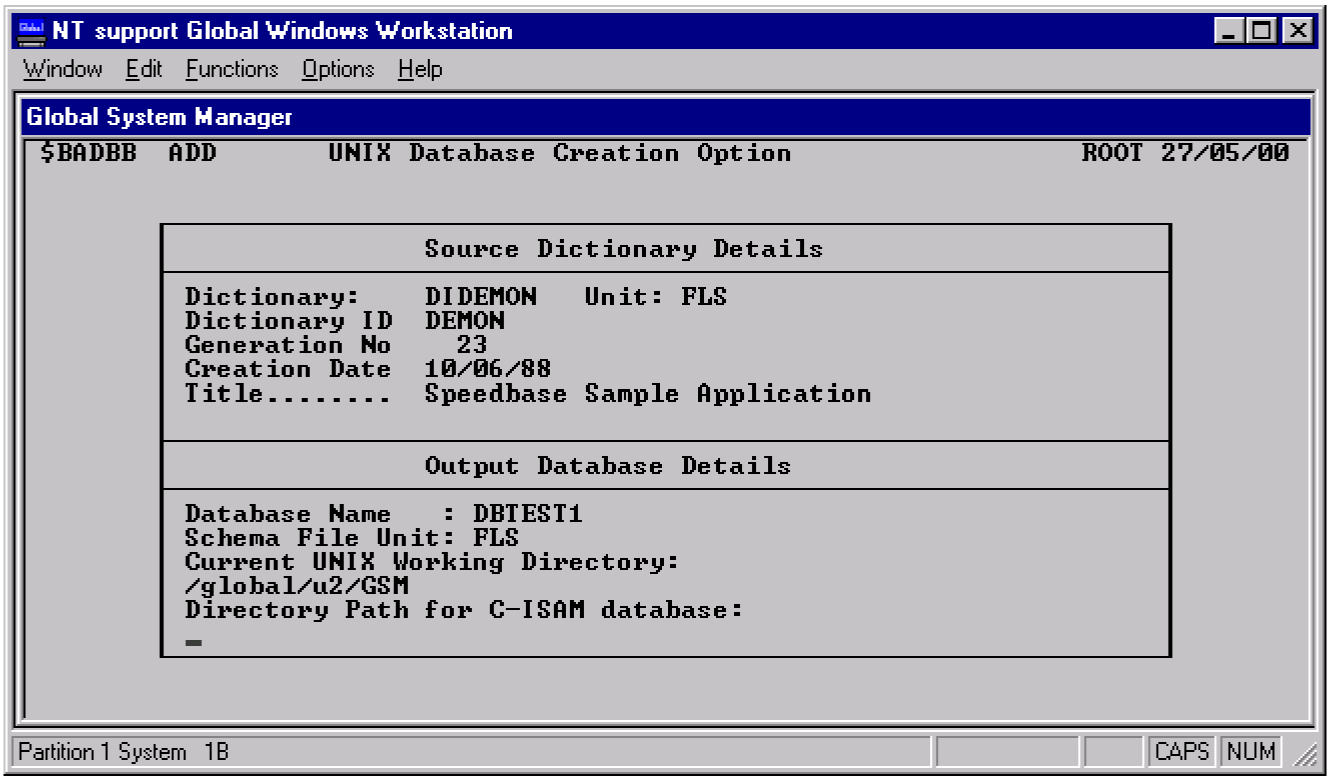
NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS
NUM

Figure 8a - The Unix Database Utility Menu

The six menu options are discussed in the following sections of this chapter.

8.1 Create Unix Database

To create a new Unix database key 1 at the Unix database utility menu. The utility displays the window:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS NUM

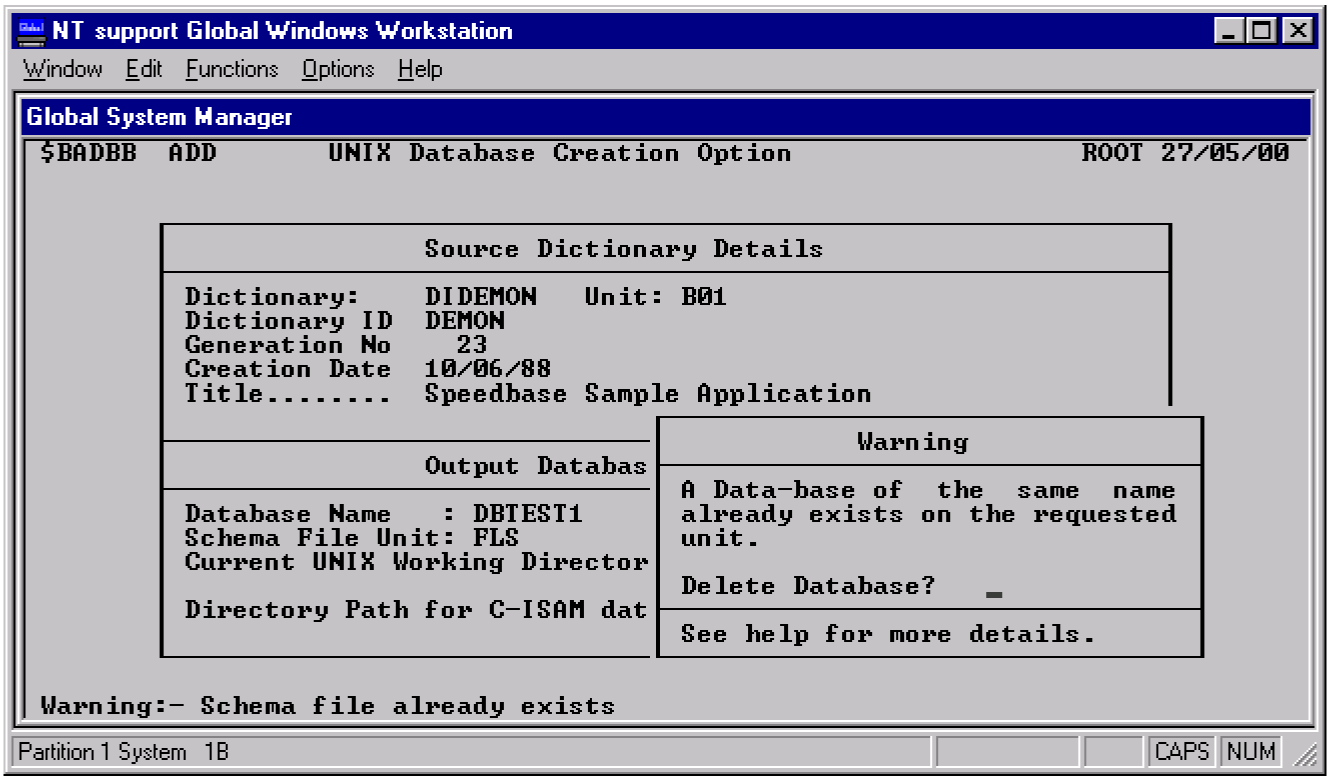
```
$BADBB  ADD       UNIX Database Creation Option                    ROOT 27/05/00

                        Source Dictionary Details

        Dictionary:     DIDEMON   Unit: FLS
        Dictionary ID   DEMON
        Generation No     23
        Creation Date   10/06/88
        Title........   Speedbase Sample Application

                        Output Database Details

        Database Name    : DBTEST1
        Schema File Unit: FLS
        Current UNIX Working Directory:
        /global/u2/GSM
        Directory Path for C-ISAM database:
        _
```

Figure 8.1a - The Unix Database Creation Window

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 schema file and new dictionary file within Global.

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.


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

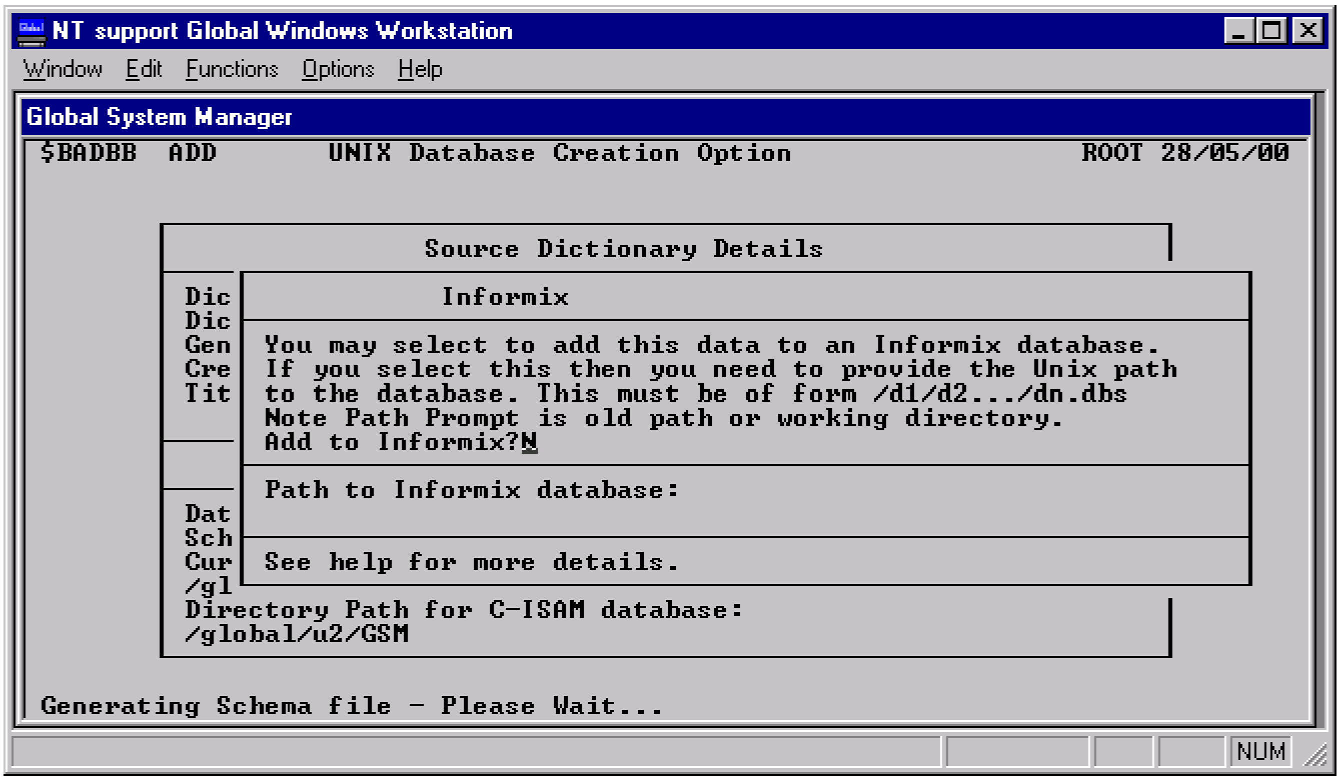
```
$BADBB  ADD      UNIX Database Creation Option                ROOT 27/05/00

                  Source Dictionary Details

  Dictionary:     DIDEMON   Unit: B01
  Dictionary ID   DEMON
  Generation No     23
  Creation Date   10/06/88
  Title........   Speedbase Sample Application

                  Output Databas          Warning

  Database Name   : DBTEST1       A Data-base of the same name
  Schema File Unit: FLS           already exists on the requested
  Current UNIX Working Director   unit.

  Directory Path for C-ISAM dat   Delete Database? _

                                  See help for more details.

Warning:- Schema file already exists
```

Figure 8.1b - The Database Already Exists Pop-Up Window

You then enter the Unix directory path for the database datafiles (e.g. /usr/dir). 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, this 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 following window then appears:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
NUM

```
$BADBB  ADD       UNIX Database Creation Option              ROOT 28/05/00

                    Source Dictionary Details
          Dic          Informix
          Dic
          Gen  You may select to add this data to an Informix database.
          Cre  If you select this then you need to provide the Unix path
          Tit  to the database. This must be of form /d1/d2.../dn.dbs
               Note Path Prompt is old path or working directory.
               Add to Informix?N

               Path to Informix database:
          Dat
          Sch
          Cur  See help for more details.
          /gl
          Directory Path for C-ISAM database:
          /global/u2/GSM

Generating Schema file - Please Wait...
```

Figure 8.1c - The Informix Selection Window

This window allows you to update an optional Informix dictionary, which must earlier have been generated 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 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 to this fact. 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 Utility will then update these, thus completing the database creation process.

8.2 Rebuild Unix Database

The Unix 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.

8.2.1 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.

8.2.2 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 (such an invalid account balance, or an invoice referencing to 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.

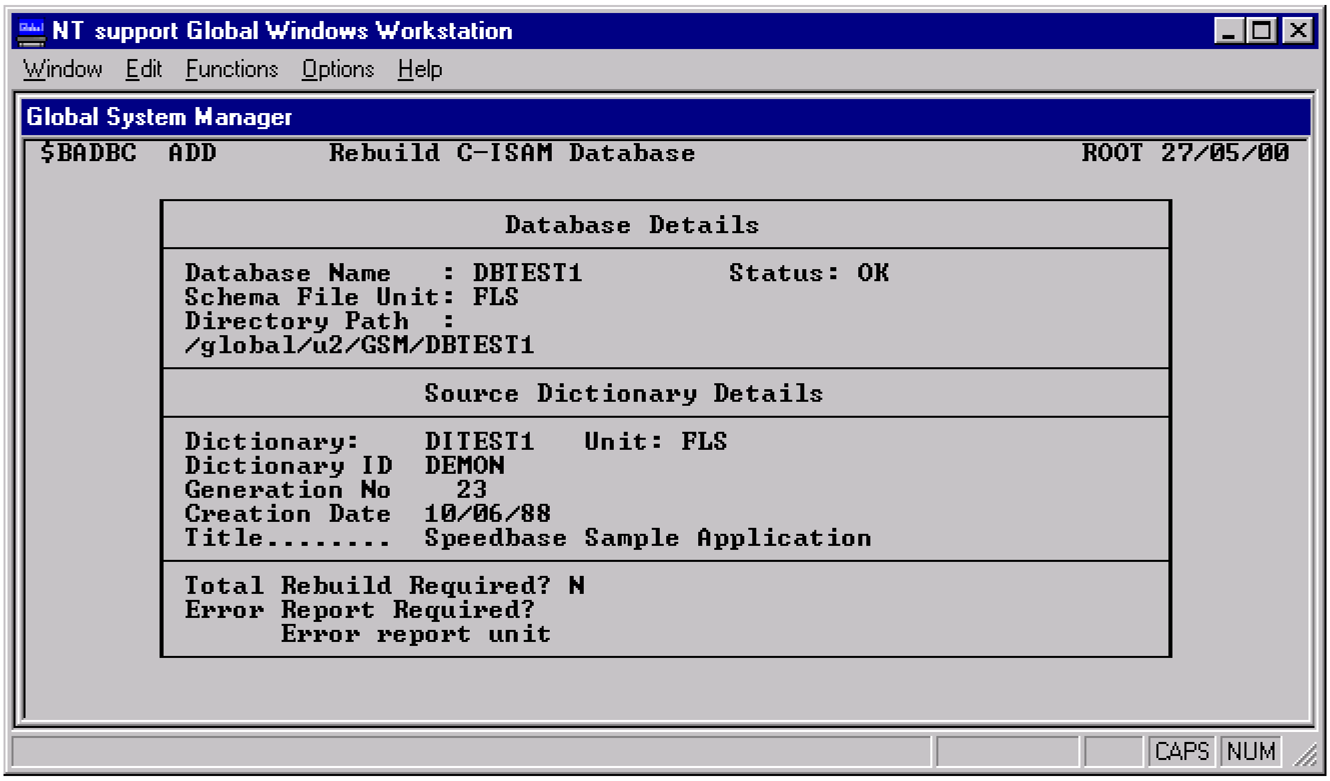
The type of errors discussed above can only arise from certain hardware malfunctions (e.g. a loss of power) or from extremely serious program errors. The utility guarantees the internal integrity of the database following such errors, but some application programs may require further processing to be performed.

The rebuild utility operates by examining the data records physically stored in the database disk files. This means that all information up to and including the last successfully completed database update can normally be rebuilt. The utility cannot be used, however, if the disk on which the database resides is lost or develops permanent errors. In this event the only recourse is to resort to backup.

During normal use, Speedbase 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 Unix 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 datafiles by Unix programs other than Speedbase programs.

8.2.3 Unix Database Rebuild

To rebuild your Unix database key 2 at the Unix database utility menu. Reply to the schema file name and unit prompts and the utility displays the window:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
CAPS NUM

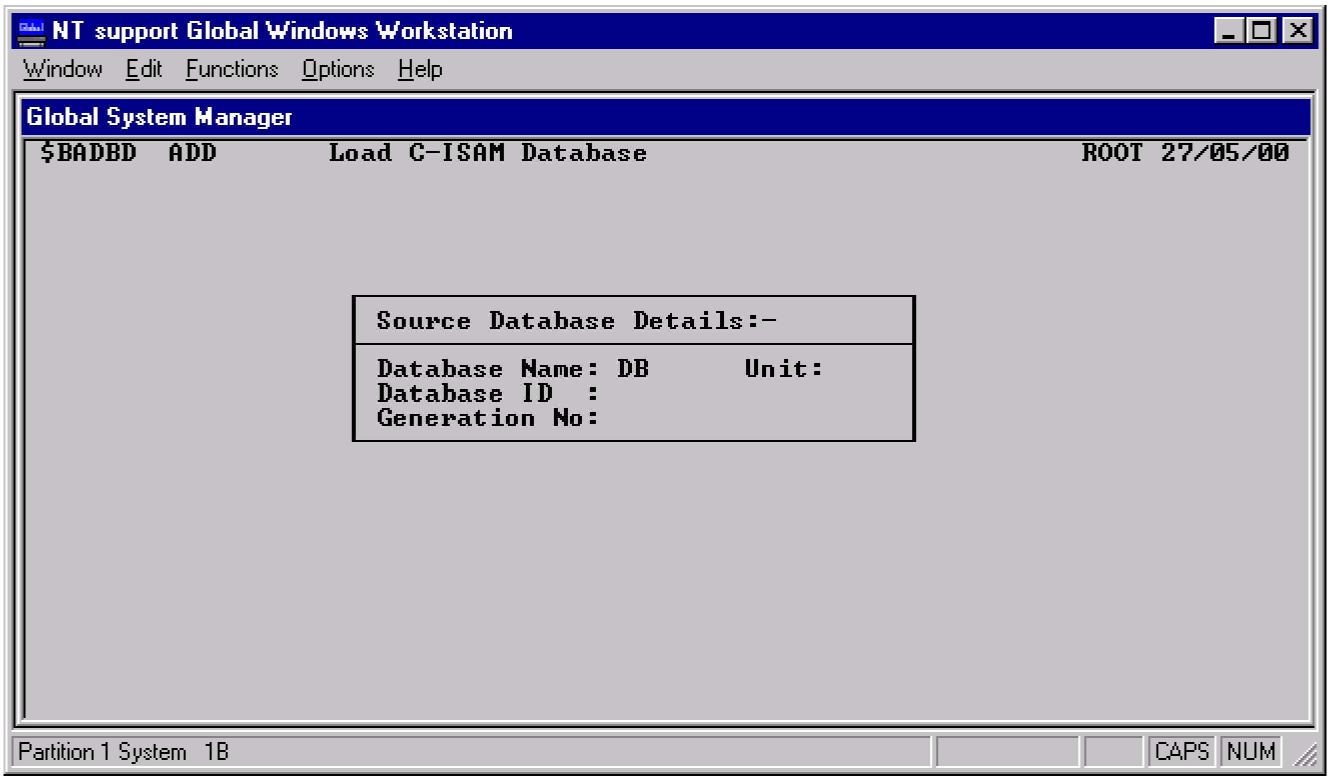
```
$BADBC   ADD       Rebuild C-ISAM Database                      ROOT 27/05/00

                          Database Details
       Database Name    : DBTEST1         Status: OK
       Schema File Unit: FLS
       Directory Path   :
       /global/u2/GSM/DBTEST1
                       Source Dictionary Details
       Dictionary:     DITEST1   Unit: FLS
       Dictionary ID   DEMON
       Generation No     23
       Creation Date   10/06/88
       Title........   Speedbase Sample Application

       Total Rebuild Required? N
       Error Report Required?
             Error report unit
```

Figure 8.2.2a - The Unix Database Rebuild Window

Reply to the total rebuild prompt and the utility displays the record number as the rebuild proceeds.

8.3 Load Unix Database

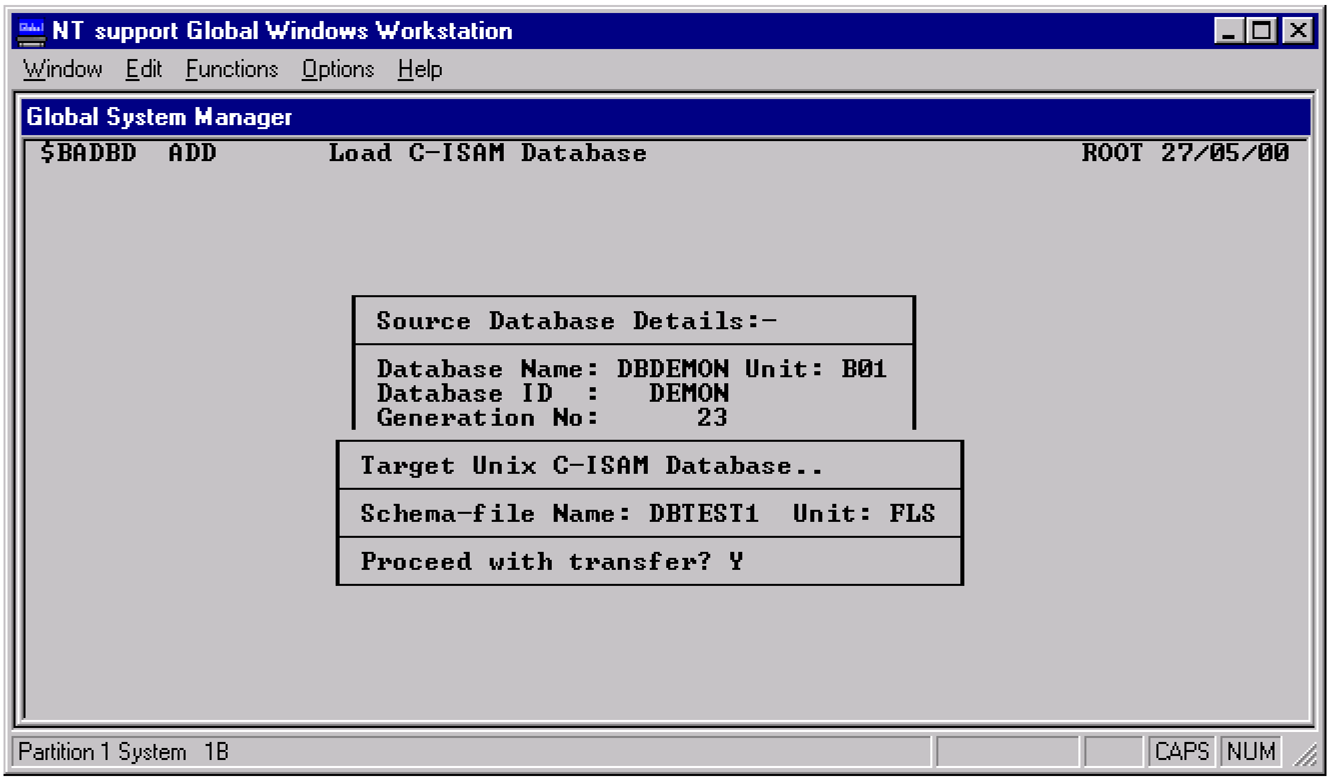
If you have data stored in a Global format Speedbase database you may load it into a Unix C-ISAM format Speedbase database. To load a newly created Unix database with data from a database stored in the Global file structure, key 3 at the Unix database utility menu. The utility displays the window:



```
$BADBD   ADD        Load C-ISAM Database                          ROOT 27/05/00

                    Source Database Details:-

                    Database Name: DB       Unit:
                    Database ID   :
                    Generation No:
```

Figure 8.3a - Source Database Details

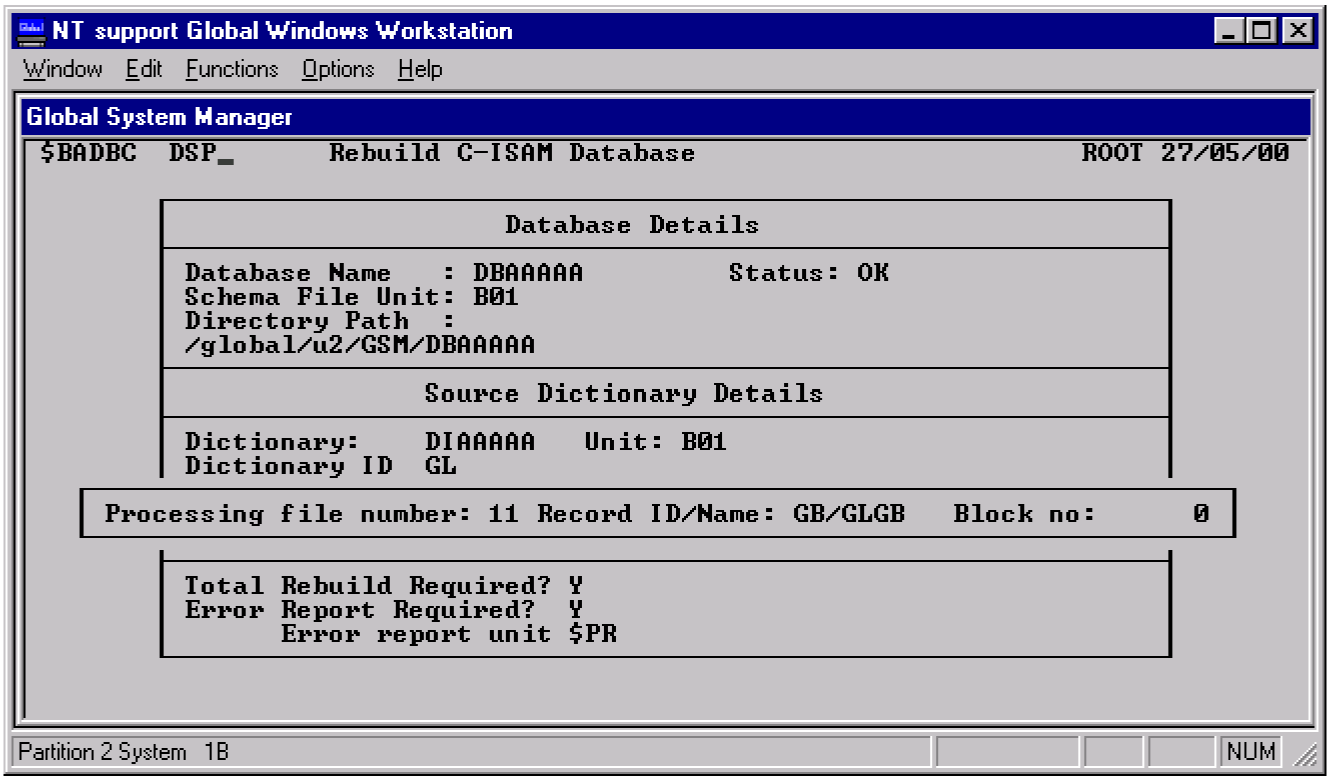
Reply to the database name and unit prompts and the target Unix database window is displayed:



```
$BADBD   ADD        Load C-ISAM Database                          ROOT 27/05/00

                    Source Database Details:-

                    Database Name: DBDEMON Unit: B01
                    Database ID   :   DEMON
                    Generation No:        23
                   Target Unix C-ISAM Database..

                   Schema-file Name: DBTEST1   Unit: FLS

                   Proceed with transfer? Y
```

Figure 8.3b - Target Unix Database Window

Reply with the name and unit of the schema file you specified in the database creation phase and confirm that the transfer is to proceed. The utility displays the transfer details pop-up as records are transferred from the Global format Speedbase database to the Unix C-ISAM format Speedbase database:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager

```
$BADBC  DSP_      Rebuild C-ISAM Database                         ROOT 27/05/00

                         Database Details

        Database Name    : DBAAAAA          Status: OK
        Schema File Unit: B01
        Directory Path   :
        /global/u2/GSM/DBAAAAA

                     Source Dictionary Details

        Dictionary:     DIAAAAA    Unit: B01
        Dictionary ID   GL

  Processing file number: 11 Record ID/Name: GB/GLGB   Block no:       0

        Total Rebuild Required? Y
        Error Report Required?  Y
              Error report unit $PR
```

Partition 2 System 1B
NUM

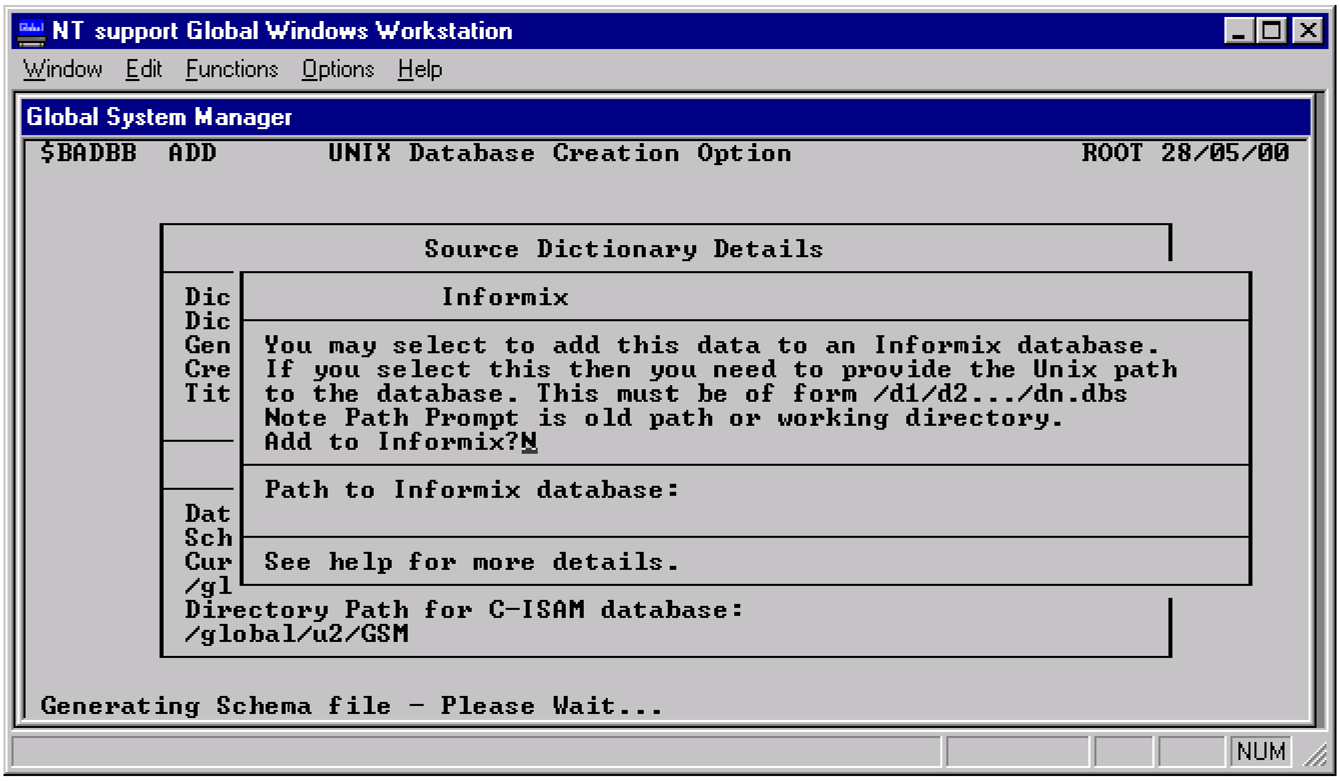
Figure 8.3c - Transfer Details Pop-up

Note that you may only transfer data to a newly-created database in the Unix file structure. You may not use the utility to transfer data to a database containing any data records. If you wish to re-transfer data you should create the Unix C-ISAM format Speedbase database anew, deleting the previous version, and transfer data from the Global format Speedbase database to it. Any data present in the Unix database but not present in the Global database would be lost in this process.

8.4 Dump Unix Database

This facility used to write the data contained in your Unix C-ISAM 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 one Unix machine to another. The Dump facility 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.

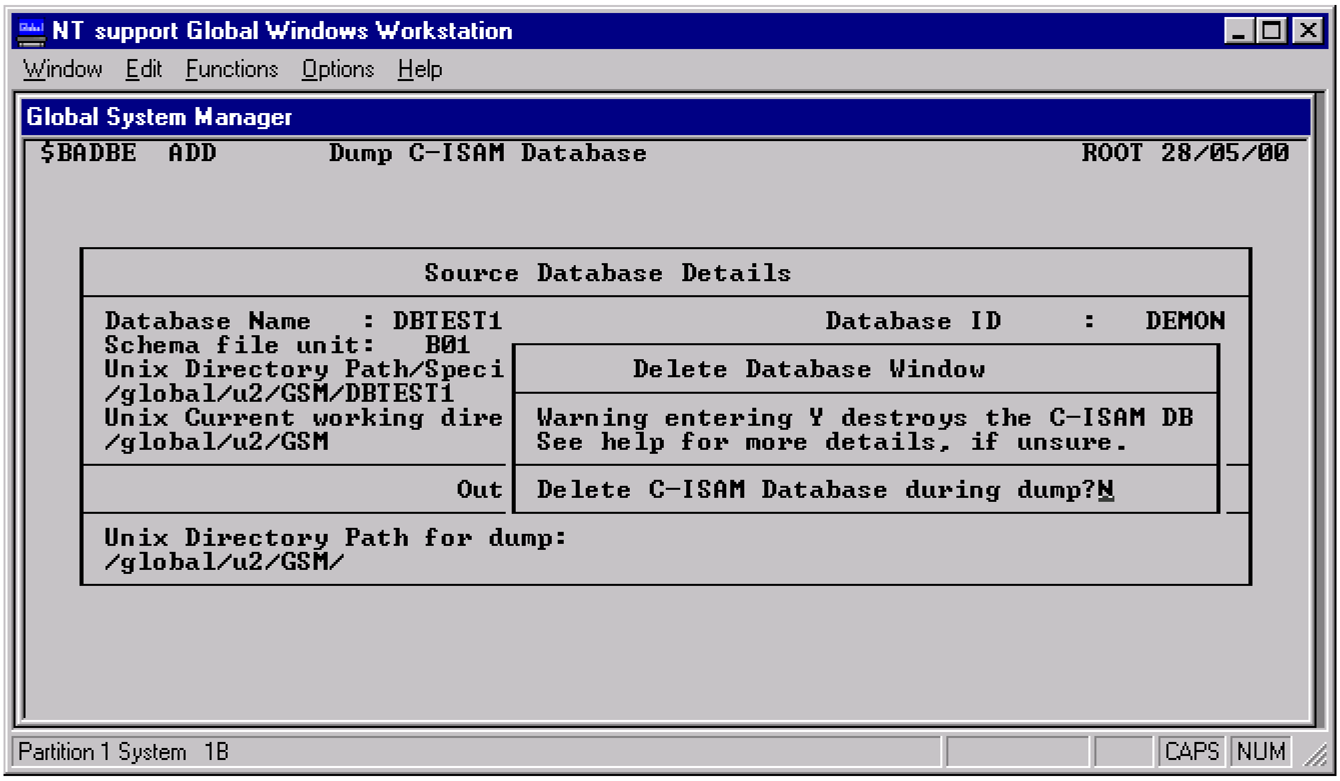
To dump your Unix database, key 4 at the Main menu. You must then enter the Schema file name and unit, after which the following window is displayed:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
NUM

```
$BADBB   ADD        UNIX Database Creation Option              ROOT 28/05/00

                  Source Dictionary Details
   Dic           Informix
   Dic
   Gen   You may select to add this data to an Informix database.
   Cre   If you select this then you need to provide the Unix path
   Tit   to the database. This must be of form /d1/d2.../dn.dbs
         Note Path Prompt is old path or working directory.
         Add to Informix?N

         Path to Informix database:
   Dat
   Sch
   Cur   See help for more details.
   /gl
   Directory Path for C-ISAM database:
   /global/u2/GSM

 Generating Schema file - Please Wait...
```

Figure 8.4a The Unix Database Dump Window

The window displays 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:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS
NUM

```
$BADBE   ADD        Dump C-ISAM Database                       ROOT 28/05/00


              Source Database Details

   Database Name    : DBTEST1                  Database ID    :   DEMON
   Schema file unit:   B01
   Unix Directory Path/Speci    Delete Database Window
   /global/u2/GSM/DBTEST1
   Unix Current working dire  Warning entering Y destroys the C-ISAM DB
   /global/u2/GSM             See help for more details, if unsure.

                          Out  Delete C-ISAM Database during dump?N

   Unix Directory Path for dump:
   /global/u2/GSM/
```

Figure 8.4b The Unix Dump 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.

Having responded to the above prompt, the utility then dumps the designated database. This causes a number of files to be created in the specified directory. These are:

DIxxxxx	Dump of Database Dictionary File
DBxxxxxRTn	Dump of Unix C-ISAM file record type n

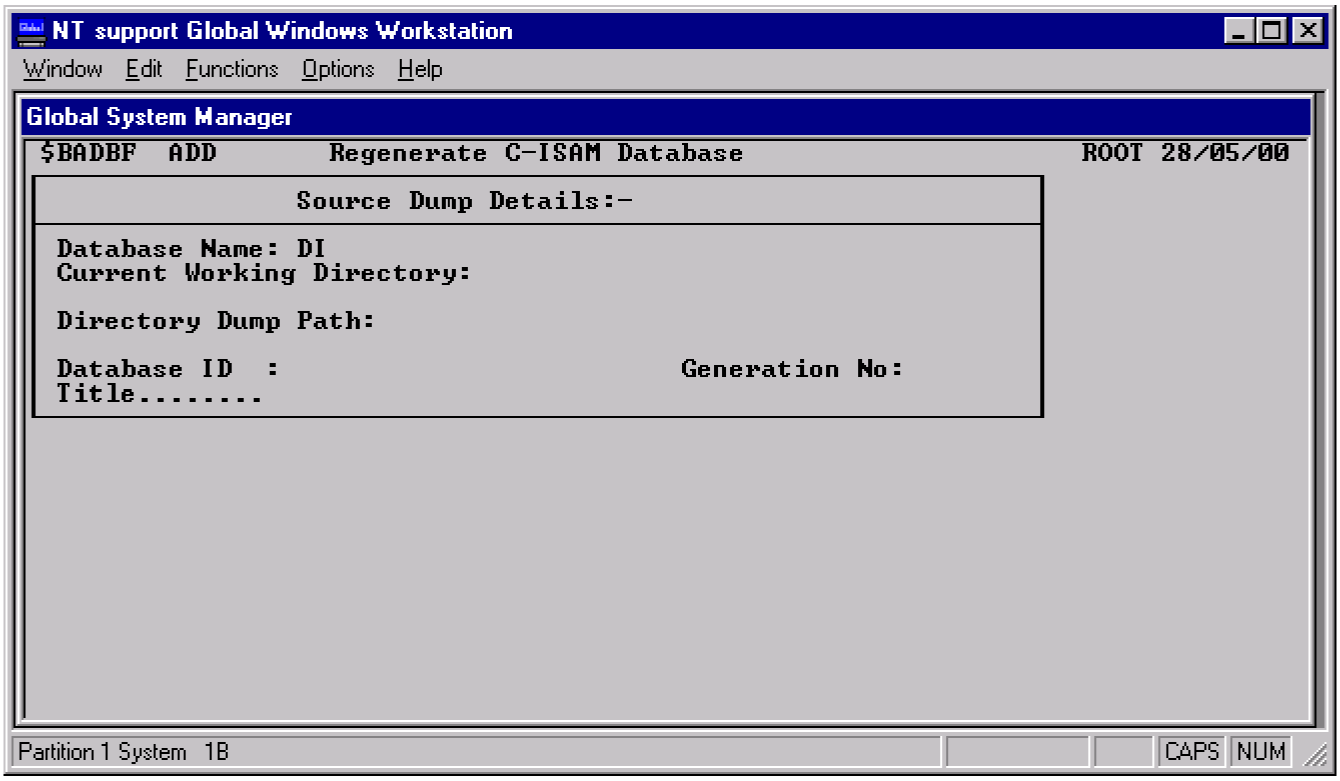
Note that the dump files are all inter-related, and must therefore be restored as a set. Under no circumstances can the files from one dump be interchanged with files from another later or earlier dump.

Once the dump process has completed the utility returns to the main menu.

8.5 Regenerate Unix Database

This option is used to reload and/or convert a database that was previously dumped using the facility described in section 8.4.

To regenerate your database key 5 at the main menu. The following window is then displayed:

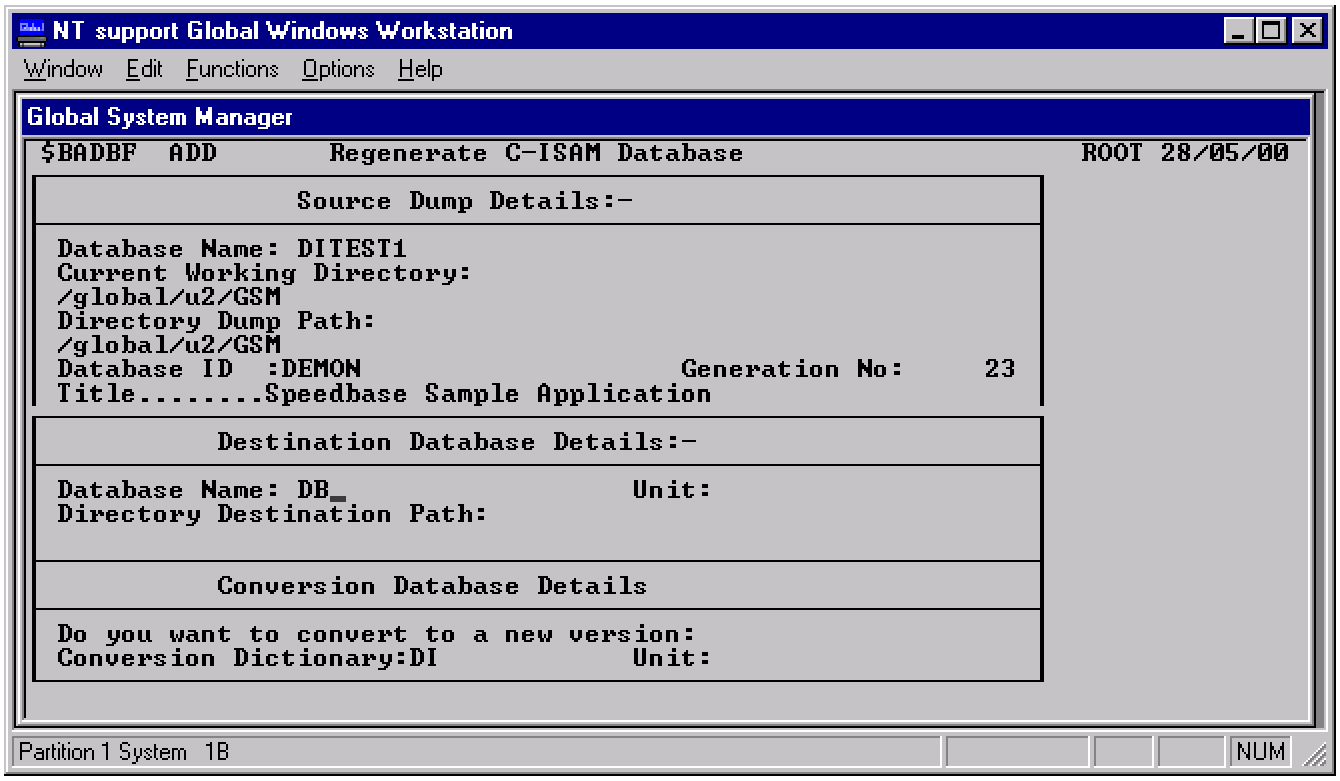

NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS
NUM

```
$BADBF  ADD        Regenerate C-ISAM Database                ROOT 28/05/00

                    Source Dump Details:-

  Database Name: DI
  Current Working Directory:

  Directory Dump Path:

  Database ID  :                          Generation No:
  Title........
```

Figure 8.5a - The Unix Database Regeneration Source Window

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 following window is then displayed:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
NUM

```
$BADBF   ADD        Regenerate C-ISAM Database                     ROOT 28/05/00
              Source Dump Details:-

 Database Name: DITEST1
 Current Working Directory:
 /global/u2/GSM
 Directory Dump Path:
 /global/u2/GSM
 Database ID  :DEMON                        Generation No:     23
 Title........Speedbase Sample Application

         Destination Database Details:-

 Database Name: DB_                   Unit:
 Directory Destination Path:

         Conversion Database Details

 Do you want to convert to a new version:
 Conversion Dictionary:DI             Unit:
```

Figure 8.5b - The Unix Database Regeneration Destination Window

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/dir). 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 apear:

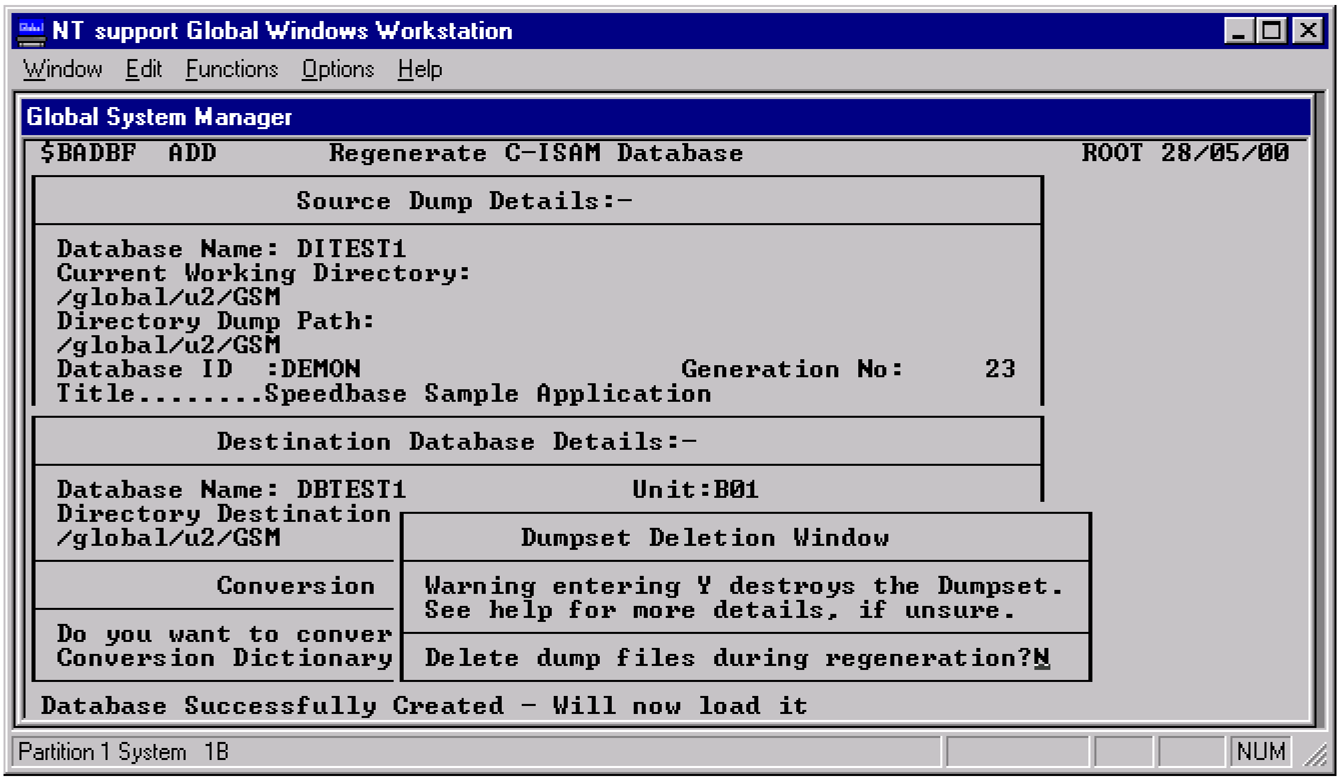
```
Database already exists on this directory - Delete?
```

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 8.1 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 Regeneration Delete option Pop-up:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
NUM

```
$BADBF  ADD       Regenerate C-ISAM Database                    ROOT 28/05/00
             Source Dump Details:-
 Database Name: DITEST1
 Current Working Directory:
 /global/u2/GSM
 Directory Dump Path:
 /global/u2/GSM
 Database ID  :DEMON                     Generation No:    23
 Title........Speedbase Sample Application
          Destination Database Details:-
 Database Name: DBTEST1                Unit:B01
 Directory Destination
 /global/u2/GSM          Dumpset Deletion Window
          Conversion     Warning entering Y destroys the Dumpset.
                         See help for more details, if unsure.
 Do you want to conver
 Conversion Dictionary   Delete dump files during regeneration?N
 Database Successfully Created - Will now load it
```

Figure 8.5c - The Unix Regeneration Delete Pop-up

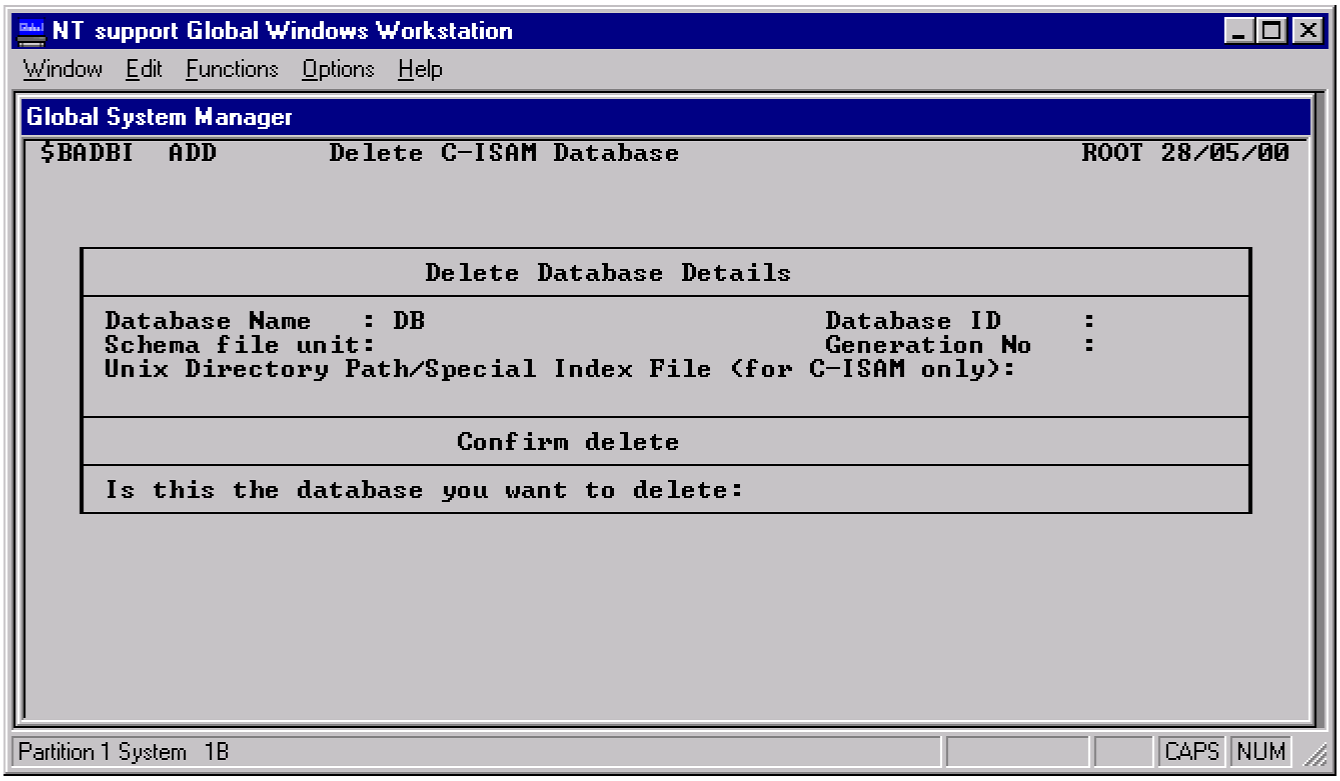
This option allows you to delete the dump files **during the course of regeneration**. If you confirm this prompt, the dump files will be progressively deleted as each file is reloaded. Note that this option causes considerably less disk to be used during the dumping process. However, should the utility 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.

8.6 Delete Unix Database

This option allows you to delete a Unix C-ISAM database. To delete a database, enter 6 at the main menu. The following window is then displayed:


NT support Global Windows Workstation
Window Edit Functions Options Help
Global System Manager
Partition 1 System 1B
CAPS
NUM

```
$BADBI  ADD       Delete C-ISAM Database                         ROOT 28/05/00

            Delete Database Details

 Database Name    : DB                          Database ID     :
 Schema file unit:                              Generation No   :
 Unix Directory Path/Special Index File (for C-ISAM only):

                Confirm delete

 Is this the database you want to delete:
```

Figure 8.6a - The Database Deletion Window

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 and Unix files associated with the database.

On completion the utility returns to the main menu.

Appendix A - Error Messages

This appendix describes all the error messages that may be displayed by Speedbase Presentation Manager.

A.1 Database Loader and Window Manager Error Messages

The error messages described in this section may arise during the use of any application using the Speedbase Presentation Manager. Error messages that are specific to a particular Speedbase utility are documented in later sections of this appendix.

Cannot load $BASYS - Not found or I/O Error

In attempting to run your application program the Speedbase Presentation Manager has been unable to load the system area $BASYS - suspect system corruption.

Cannot Write Record - File is Full

The database datafile has no free record space for the record type to be written.

End of List

You have attempted to display beyond the end (or start) of a page of records in a window.

LOCK: name

This warning message is flashed on the screen approximately every two seconds when the indicated record is locked by another operator.

New Terminal Type - Using default facilities

The terminal information file, set up by the customisation utility $BACUS, is not present. You may proceed using the default facilities or use $BACUS to set up a new terminal information file.

No Records Found

No records match the enquiry key entered.

Record is active - Cannot delete

A master record cannot be deleted if it is linked to servant records. These must be deleted before you attempt to delete this master record.

Record Key Already Exists

A record with the key you have entered already exists. You must choose a different key for the record you are entering.

Record Key Already Exists - Enquire?

A record with the key you have entered already exists. If you wish to display this record reply Y, otherwise reply N.

Record Locked - Try again

Another user of the system has exclusive access to the record at present.

Record may Not be Selected

You may not select this particular record for maintenance. For example, in a series of invoice records you might be allowed to maintain (change) the details of invoices until they are posted but not afterwards.

System I/O Error nn

An I/O error has occurred while processing a Unix C-ISAM format database, where nn specifies the Unix or C-ISAM error more particularly. For more information relating to this error condition, please refer to your Unix operating Manual, and/or Informix C-ISAM documentation.

Terminal file is corrupt or incompatible version

You must set up a new terminal information file using the Speedbase customisation utility $BACUS.

A.2 Customisation Utility Error Messages

The following error messages are displayed by the $BACUS utility.

Attribute Code must be 1 - 8 or U, D, B, R, or F

You may only use one of the codes quoted as an attribute code.

More than two colour attributes (1 - 8) keyed - Invalid

You must specify two different colours for the "ink" and "paper" attributes.

Terminal file invalid or not found

The terminal information file set up by the Speedbase customisation utility $BACUS is corrupt or missing.

Too many different attributes specified - Please simplify

You must reduce the number of different attribute combinations you have specified.

Two Colour Attributes (1 - 8) Required for Ink & Paper

You must specify two different colours for the "ink" and "paper" attributes.

Unsupported Code keyed - Please see Attribute Table above

You may only use those attribute codes listed as available in the attribute table.

A.3 Backup Utility Error Messages

The following error messages are displayed by the $BASAV utility.

Backup unsuccessful - last file deleted

This message follows a prior error message and informs you that the current backup was unsuccessful. The backup file created by the aborted backup has been deleted. A more specific error message should have been displayed prior to this message. Action to take depends upon this prior message as described in this section.

Bad status on dictionary - abort

The data dictionary has been found to be flagged as corrupt. Reproduce the dictionary from backup or recompilation. If recompiling, DO NOT use the RGN option.

Can't use spooler unit for backup

The unit currently assigned as the spooler should not be used to receive backup files. Reassign the backup unit.

Cannot open backup directory

The directory of the volume mounted on the backup unit cannot be read by the backup utility. Check that the correct volume has been mounted.

Date mismatch on last backup

While checking the last backup file produced during incremental backup, the creation date of the last backup file has not matched that shown in the history table. Check that the correct volume has been mounted.

DI/DB generation mismatch - abort

The data dictionary and database have different stored generation numbers which indicates that the database was not originally created with the structure of the current dictionary. This is a potentially dangerous situation and $the backup utility will not proceed.

Database not found or in use
Datafile xxxxx not found or in use
Dictionary not found or in use

The indicated file was not found, or was found to be in use by another user or screen. If in use, the file must be released before backup can proceed.

Main Index is of invalid type
Dictionary is of invalid type
Datafile xxxxx is of invalid type

The indicated file has been found to be of an incorrect file type and cannot be considered a valid Speedbase file.

Disk not correctly formatted

The disk mounted on the backup unit cannot be read by the backup utility. Check that the correct disk has been loaded and that it is of the correct format.

Error - Possible database corruption detected - perform rebuild

The backup utility has detected that an end-of-file marker has been placed incorrectly due to a system crash during initial processing. Rebuild database, see chapter 5.

Error on backup of dictionary - abort
Error opening backup file
Error processing backup header
Error reading dictionary header
Error reading dictionary record header
Error reading database header block
Error reading database control block
Error reading database
Error re-writing database control block
Error re-writing database header
Error updating database header
Error updating backup header
Error updating database during full backup
Error writing initial backup header
Error writing to backup file
Error writing history to backup
Irrecoverable I/O error - abort

All the above error messages indicate a serious file error from which the backup utility cannot recover. If the error occurred whilst performing an incremental backup, a full backup should be attempted. Otherwise, check the relevant file(s) and associated hardware unit for corruption.

File(s) exist on backup volume - delete?

The backup volume mounted has a valid volume-id but there are existing files on the volume. The backup utility wants to delete these files and will do so if you respond Y or <RET>. If you respond "N" the backup will normally be aborted, UNLESS the volume is labeled BACKUP. In this event the backup utility will simply use such space as is available on the unit and leaves existing files intact.

Files open & in use on backup volume - abort

The new backup volume contains files which are currently in use by another user or screen. The backup utility is therefore unable to delete them and cannot proceed. Use $F to check the files on the backup volume.

Insufficient memory to backup Rec Type <rt>

The backup utility is unable to load a single data record of the record type indicated. See section 3.5 on memory requirements.

Insufficient file space for backup

There has not been enough room on the backup volume to store the complete dictionary. Refer to section 3.5.

Invalid unit-id

The entered unit-id is not a recognised unit in your system. The Global System Manager $U command shows all valid unit-addresses.

Invalid response - must be F,I or P

Only the above responses are valid.

Invalid volume assigned

During incremental backup, you have assigned a unit for backup which will not support a standard Global System Manager directory. Check that you have assigned the correct unit.

Incorrect volume mounted

During incremental backup, you have loaded the wrong volume (disk). You must load the last volume used for the current backup cycle. Check the volume-id of the backup disk with that requested by the mount prompt.

Invalid volume-id for backup

The volume (disk) mounted cannot be used for backup as it does not have a volume-id of BACKUP or the same id as that about to be written. Refer to section 3.3.

Last backup file not found or in use

While checking the last backup file produced during incremental backup, the file was either not found or was found to be in use by another user or screen. The latter case would be unusual and you should check that you have loaded the correct volume.

Last backup file from different database

The prior backup file has been checked during incremental backup and it is found that although all aspects of date, file-names, file types and volume-id match, the backup file was actually from a different database. This is a potentially dangerous situation and could only happen if you have two databases being backed up under the same generic name resulting in duplicate named backup cycles. Refer to section 3.3 regarding generic names and it is suggested that you recreate one of the databases using the generation utility and back it up using a different generic name.

Must start with a full backup

When backing up a database which has never been backed up before, you must initially do a full backup. You cannot perform an incremental backup until the backup cycle has commenced with a full backup.

Prior full backup aborted - must do again

The backup utility is aware that the previous full backup was unsuccessful and that the current backup cycle is invalid. Further incremental backups would be useless and therefore a full backup is needed to commence a valid backup cycle.

Same previous cycle exists in history - Delete? (Y/N)

Entries exist in the history table for the same backup cycle-id that you have requested for this full backup. This is a common occurrence when you recycle backup disks (e.g. cycle A, B, C, then A again etc). The superseded history table entries must be deleted to eliminate confusion between similarly named backup cycles. A response of Y will cause these entries to be deleted from the table only. Any other response will result in the backup process being aborted and the history table remaining unchanged.

Type mismatch on last backup file

When checking the prior backup file during incremental backup, the file type of the last backup file has not matched the type of backup as per the history table. Check that the correct volume has been mounted and not a previous backup volume with the same backup cycle-id.

Unable to scratch volume - Abort

The backup utility has tried to scratch (delete) the files on the newly mounted backup volume but has been unable to do so. Check the volume and if required, mount a different one.

A.4 Generation Utility Error Messages

The following error messages are displayed by the $BADGN utility.

At least nnn K filesize is required

The size of the main index database file must be more than that shown (nnn K) as this is the absolute minimum allowable. You are advised to allocate more file space than this minimum. The calculated default value (obtained by entering <RET>) should be sufficient.

Cannot delete file

You have given your approval to delete an earlier version of a file but the utility has been unable to do so. Use a different unit or abort to investigate the problem.

Confirm - delete old dictionary :

The utility is unable to rename the new output dictionary with the same name and unit as the output database. An earlier version of the dictionary already exists on that unit so a response of Y will cause the utility to delete it.

Confirm - overwrite prior version of database Y/N :

An on-line input database exists with the same name as that about to be produced. A response of Y allows the input database to be deleted when all processing has completed successfully.

Conversion work file space exhausted

This file is opened with maximum available file space and truncated when all conversion records have been written. This temporary file utilises the main output database unit and is generally quite small. Make available more contiguous file space on the main output database unit, or use a different unit.

Cycle x not on unit yyy

The first character x of the BID entered is the major cycle-id and is not the same as the backup cycle currently loaded on the unit indicated. Either the entered BID is wrong or you have loaded the wrong volume.

DI/DB generation mismatch error

The data dictionary and database have different stored generation numbers which indicates that the database was not originally created with the structure of the current dictionary. Restore the dictionary from a full backup of the current database.

Error - possible database corruption detected - perform rebuild

The utility has detected that an end-of-file marker has been placed incorrectly due to a system crash during initial processing. The database must be rebuilt.

File exists, delete ?

The utility is unable to open a new file because one with the same name already exists on that unit. If you wish to delete the old file, enter Y, otherwise respond N and you will be asked to re-enter the unit-id.

File number must be 1, 2 or 3

Up to three data files may be specified in a Speedbase data-base and all record types must be associated with one of these files. Re-enter or type <RET> for the default.

File <f#> record <r#> - start too high

The first data record of the <r#>th record type in data file number <f#> would begin beyond the 33,554,432th (32MB) byte of the file. This is illegal in Speedbase and you should review record type/data file distribution to avoid this occurring.

Datafile xxxxx not found or in use

The indicated file was not found, or was found to be in use by another user or background job etc. If in use, the file must be released before the utility can proceed.

Datafile xxxxx is of invalid type

The indicated file has been found to be of an incorrect file type and cannot be considered a valid Speedbase file.

Insufficient memory available

All dictionary details relating to a particular record type being converted have not been able to be loaded into memory due to inadequate buffer space available. Refer to section 4.7 for details of generation utility memory requirements.

Insufficient memory to process Rec Type <rt>

In order to perform the data transfer stage, the utility must be capable of loading at least one complete data record into memory. Data records of record type <rt> cannot be loaded due to excessive length. Refer to section 4.7 for details of the generation utility's memory requirements.

Insufficient room on device or directory

The utility has been unable to open a file on the unit entered as there is no file space or no directory entries remaining on that unit.

Invalid BID entered

The last two characters of the backup-id entered are not in the range 1 to 99. Re-enter the complete BID.

Invalid number of records

The integer value for the number of record which you have entered is not in the range 1 to 8288350 inclusive. Re-enter or key <RET> for the default value.

Invalid unit

The unit-id entered is not recognised.

Invalid unit for restore

The unit-id entered does not support a standard Global System Manager file directory. Enter a different unit-id.

Memory allocation error

The utility has been unable to acquire the required work space which had previously been calculated as available. This message indicates an internal error contact your software supplier.

Numeric field overflow on Record Type <rt>

Conversion processing has attempted to transfer a numeric value to an output field which has been reduced in capacity. The numeric value has exceeded the capacity of the output field and overflow has resulted. The output field will be initialised to binary zeros. You may respond <RET> to continue, however, you will be prompted for every overflow error which occurs. If you do not wish to be prompted for further overflow errors occurring in this record type, reply <CTRL A>. You will be prompted again if overflow errors occur in subsequent record types, in which case you may again reply <CTRL A>.

Old database file in use - abort

After you have approved the deletion of a superseded database, the utility has been unable to delete it.

Warning - file <file1> could not be renamed as <file2>

Input files have been renamed and were to have been deleted at the end of processing. However, due to a serious error, they should now be renamed with their original names and left as they were. The utility has been unable to rename the file <file2> so it remains on its original unit under the temporary file name <file1>.

Can't delete old temp database file <file description>
Can't open <file description>
Can't rename input database file <file description>
Error processing database file <file description>
Error reading <volume-id>
Error reading <file description>
Error reading root index block
Error truncating <file description>
Error writing to <file description>
Error writing block to database
Error writing record to database
Error writing 1st virgin for RT <rt#>
I/O error on <file description>
Unable to read <file description>

All of the above errors indicate a serious file error from which the utility is unable to recover. Check the indicated file(s) and hardware unit for corruption. If database files are damaged, restore the database from backup.

A.5 Rebuild Utility Error Messages

The following error messages are displayed by the $BARBL utility.

<Rt> <Name1> <Rec#> Record not found for <Name2> <Pri-key>

A master record was not found for this servant record during a total rebuild. The name of the missing master record is shown as <Name2>. The record type, name, and record number of the current record are shown above as <Rt>, <Name1> and <Rec#> respectively. The primary index key of the missing master record is shown as <Pri-key> above. The servant record remains unlinked, meaning that the database remains corrupt. The relevant master record should be added an another total rebuild performed.

<Rt> <Name1> <Rec#> GVA under/overflow <Name2> <Pri-key>

If this error occurs, you will require the assistance of development personnel who have access to the Speedbase Development System. The error indicates that an accumulator residing on a master record has had its capacity exceeded (i.e. a field is simply not large enough to store a required value). The database may be used following this error, but it should be noted that one of the accumulators will be invalid. The affected record can be identified by the record's primary key shown as <Pri-key> above.

In resolving this problem, development personnel should note the following. This error means that the combined total value of GVF fields for servant record type <Rt> has exceeded the capacity of the GVA field in the master record whose name is <Name2> and primary index is <Pri-key>. The GVA field in error should be increased in capacity. This requires re-specification of the database using the Speedbase Development System compiler. Conversion of the database can then be performed using the generation utility (see Chapter 4).

<Rt> <Name1> <Rec#> Dupl primary IDX key <Pri-key>

Two records have the same primary index key value (e.g. two customers numbered 123) have been found. The record concerned is identified by <Name1> and the primary key value <Pri-key>. One of the records should be deleted.

Theoretically, this error can only be caused by suspect database conversion operations, such as reducing the size of an existing primary index key. If this error occurs during a database conversion, it is suggested that you contact your program supplier.

<Rt> <Name1> <Rec#> Illegal index key in <Name2> <Pri-key>

A database record with an index key starting with high-values (#FF) has been detected for the record type indicated. An index key beginning with this value is illegal. The database remains corrupted, and cannot be corrected using functions available to the Speedbase Presentation Manager user.

Again, in theory, this error can only occur as a result of a suspect database conversion. If this is the case, it is suggested you contact your program supplier. Under other circumstances this error indicates that random corruption has occurred to the database, and you should restore the database from backup.

Cannot delete superseded DB file <filename>

The rebuild utility has processed an output database which was generated by the generation utility from an input database of the same name on the same main unit. The input database files were renamed to the above temporary filenames but the rebuild utility has been unable to delete them at the end of processing. Use the $F utility to delete the file named <filename> on the original database unit(s).

Cannot Open Error Report File xxxxxx

The error report file xxxxx could not be opened. Error messages will instead be directed to the screen.

Database is too corrupt for rebuild - rebuild aborted

The rebuild utility has been unable to continue due to irrecoverable errors. This may be caused by permanent errors on the disk unit(s) on which the database resides, or a corruption of the control information stored in the main index file. In all of these cases, the database should be restored from a backup, and the rebuild utility should then be re-run.

Since the utility does **NOT** check for matching version numbers, this error may also be caused by attempting to rebuild mismatching versions of database files. This should be checked by using the $F utility. If the creation dates of the database files are not identical, then this has almost certainly occurred. If the correct files cannot be found, then the only recourse is to restore from backup.

Database requires total rebuild

The database has been recognised as invalid and a total rebuild of the database is necessary to rectify corruption. Respond Y to the Total Rebuild? prompt.

Database not found, in use, or invalid type
Dictionary not found, in use, or invalid type
<filename> not found, in use, or invalid type

The indicated database, dictionary or file was not found on the unit specified, or was found but is in use by another user, or was found to be of an incorrect file type and is therefore not considered to be a valid Speedbase file.

Dictionary generation# does not match database

The dictionary found is not the same one originally used to generate the database. A matching dictionary and database structure must be present to perform a rebuild.

Dictionary is corrupt or invalid

The rebuild has failed because of corruption within the data dictionary. The dictionary should be restored from backup.

Error Report File Full
Error Count overflow
Error Report Page # overflow

So many errors have been detected during the rebuild procedure that the error report file is now full. Further errors will not be written to the report which will therefore be incomplete.

Errors in rebuild - invalid database generated

Irrecoverable serious errors have prevented the rebuild utility from constructing a valid database.

IDB pool exhausted

The size of the database's main index file is insufficient to allow the database to be rebuilt - in other words the utility has run out of index blocks to complete the creation of indexes.

In theory, this error cannot occur if the database was initially created using the generation utility. Please contact your software supplier if this is the case. The problem may be overcome by regenerating the database using the generation utility and specifying a larger index file size The default file size should be sufficient.

Input database has been named to <filename>

This message will appear if serious errors have caused the rebuild to fail and the database being processed was generated by the generation utility from an existing database of the same name on the same main unit. The superseded input database would normally have been deleted but due to failure of the rebuild, it has been left renamed to the above filename.

Insufficient memory available

The available user memory area was not large enough to process a particular record type. Refer to section 5.3.4 for more details.

Invalid free record detected / now corrected

A system crash during processing has caused an end-of-file marker to be placed incorrectly. The rebuild utility has detected the problem and automatically corrected it.

I/O error on error report file

An error has occurred while writing an error message to the report file. No further messages will be written to the report, which is therefore incomplete.

A.6 Status Utility Error Messages

The following error messages are displayed by the $BASTS and $BAST utilities.

Dictionary generation # does not match database

The generation number stored in the database does not match that stored in the dictionary. This indicates that this database was not created using the structure of this dictionary. The dictionary generation number is shown as n and the database generation number is shown on the screen header.

Dictionary not found, in use or invalid type

The dictionary was not found, was in use or was found to be of an incorrect file type and therefore not considered to be a valid dictionary.

Database not found, in use or invalid file type

A database of the name entered was not found, or is in use or was found to have an incorrect file type and is therefore not considered a valid database. Check that you have entered the correct database name.

A.7 Dictionary Utility Error Messages

The following error messages are displayed by the $BADCT utility.

Copy library file xxxxxxxx already exists - Delete (Y/N) ?

The copy library name you have specified already exists. Reply Y to delete the existing file and replace it with your new copy library.

Dictionary not found or in use

The dictionary you have specified is either not on the unit you specified, or is in use by another user.

Error reading dictionary

The utility is unable to read the dictionary file.

File number must be either 1, 2 or 3

Enter a file number of 1, 2 or 3. All other numbers are invalid.

Invalid file type for dictionary

A file exists with the dictionary name you have specified but does not have the correct file type and is therefore not a valid dictionary. Check the dictionary file name and unit.

Please enter either 1 or 2

The only options in the utility menu are numbered 1 and 2.

Unable to delete copy library file

The utility cannot delete the old copy of the file with the name you have specified for the new copy library file. Check the old file.

Unable to open copy library file - Possibly insufficient room

The utility opens a file for the copy library of size 200 Kbytes. The file is truncated when closed. The unit you specify for the copy library file must have at least 200 Kbytes of contiguous spare space.

Write error to copy library xxxxxxxx - Aborted

Check the unit you have specified for the new copy library file.

A.8 Unix Database Management Utility Error Messages

The following error messages are displayed by the $BADB utility.

Cannot Open C-ISAM file channel - Try again later

One of the C-ISAM files comprising the Unix database could not be opened. This may be because the file is missing from the appropriate directory, or may be a temporary condition due to insufficient memory. Try again later and if the condition persists, ensure that the appropriate C-ISAM files are present.

Cannot open Main Index File in Unix Directory

The special index file could not be opened within the Unix directory.

D-ISAM not supported by this utility

The version of System Manager you are running has been packaged with a version of D-ISAM rather than C-ISAM. C-ISAM must be present in order for $BADB to run correctly.

Database Datafiles not found or invalid type

The C-ISAM datafiles could not be located in the appropriate directory, or are not of C-ISAM format.

Database is already open in this partition

The database has already been opened using the $BA command, and is in use. Simply return to the menu and re-run the $BADB command.

Database is in use by another user or partition

The database could not be opened for exclusive use because it is currently open by another user or partition.

Database must be C-ISAM format

The selected database is a Global format Speedbase database rather than a Unix format Speedbase database. In order to rebuild a Global format Speedbase database you should use the $BARBL utility (see Chapter 5).

Insufficient memory available to open - try again later

The database cannot be opened because insufficient memory is currently available.

I/O error - Database cannot be opened

The database could not be opened because of I/O errors. Hardware corruption is indicated.

ISAM Not available on this machine

C-ISAM is not supported by the version of System Manager you are running. The utility cannot operate without this software.

Unable to initialise C-ISAM channels - try again later

A C-ISAM channel could not be initialised, probably as a result of memory non-availability. This error should be transient - try again later.

Unspecified error - Cannot open Database

This error message is included for completeness only and should not arise during normal processing.

A.8.1 Create Database

Cannot add information to Informix File - Path too long

The path specified for the Unix C-ISAM files exceeds the string length permitted by Informix databases. The database could therefore not be added to the requested Informix tables. You should therefore regenerate the database using a shorter path name.

Cannot create Data files - Error nn

The Unix C-ISAM files could not be created on the specified Unix directory, probably due to inappropriate access privileges. nn specifies the C-ISAM error number that occurred; see C-ISAM User Manual for further information.

Cannot create Index file

The Special Index file could not be created on the specified Unix directory, probably due to inappropriate access privileges.

Cannot create new Dictionary on this unit; $$RES = n

The new dictionary could not be opened. $$RES provides further information as to why this was the case.

Cannot create new Schema file on this unit

The schema file cannot be created, probably because there is insufficient space or directory entries remaining on the specified unit.

Cannot delete old Special Index file or file missing

The Special Index file you elected to delete cannot be deleted, possible because of inappropriate file protection.

Cannot Delete old C-ISAM file or file missing

C-ISAM files relating to the database you elected to delete cannot be deleted. They are either missing or inappropriate file protection has been assigned to these files.

Cannot Delete existing schema file

The schema file which you have chosen to supercede cannot be deleted, possibly because it is in use by another user or partition.

Cannot Open files on this directory

A database cannot be created on the specified directory, probably because of inappropriate access privileges.

Database already exists on this directory - Delete?

The Speedbase Special index file already exists on the directory you specified. If you confirm this prompt, then it will be overwritten.

Database contains invalid field or conversion types

The specified database dictionary contains invalid field types or specifications, and is therefore corrupt. Regenerate the dictionary.

Database name must be 5 characters

The Unix format database name must be 5 characters with no intervening spaces.

Database schema contains invalid field sequence - Database Generation Aborted

The database dictionary contains an invalid sequence of User and Systems area fields, and is corrupt. Restore from backup.

Dictionary is corrupt

The database Dictionary must be restored from backup or re-generated.

Each record in the database must have at least one index

It is a requirement that every record type within a Unix C-ISAM database have at least one specified index. The database dictionary must be amended and regenerated.

Error writing index file

The Special Index file could not be written on the specified Unix directory, probably due to inappropriate access privileges or I/O errors.

Generation Aborted

Generation of the Unix C-ISAM database has been aborted due to one of the above error conditions.

Insufficient room on Schema file unit to create Schema File - Reorganise or extend size of this unit and try again

Self-explanatory.

Invalid Field type detected in input dictionary

The specified database dictionary contains invalid field types or specifications, and is therefore corrupt. Regenerate the dictionary.

Invalid or corrupt Dictionary File

The specified dictionary file is corrupt and cannot be read. Restore dictionary file from backup.

Invalid Path entered - contains non-directory file

The Unix path entered did not specify a valid directory.

Logic Error from Delete - Please Report

A logic error has been detected during delete processing. The most probable reason is an internal program error. Please report this error with the circumstances leading up to it.

Path must be in Absolute Format

The Unix path name must start with a "/".

Path to Informix database must end with ".dbs"

The path you specify for the Informix Database must end with the characters ".dbs".

Record type too complex - translation table exceeds 2K

While creating the Schema file, one of the record types defined by the database dictionary was found to be too complex to mapped by a single translation table. The record type in question must be inordinately complex, containing an extremely large number of fields of differing types. The database must be redesigned to produce a simplified record layout, and a new dictionary produced.

Table to Delete from Informix not in Database

While over-writing or deleting a Unix database, the Utility was unable to remove details of the database from the Informix tables. The table entries could not be found. This is a warning, and no further action should be necessary.

Trouble Copying Dictionaries

An unspecified error has arisen while copying a directory, or dictionaries, to the new unit. This could be associated with inappropriate file protection, insufficient space or other I/O errors.

Trouble Re-opening Schema file

An unspecified error has arisen while re-opening the Schema file. This could be associated with inappropriate file protection, use by another user or other I/O errors.

Unable to re-open Dictionary

An unspecified error has arisen while re-opening the Dictionary. This could be associated with inappropriate file protection, use by another user or other I/O errors.

Unexpected entry on Informix xxx

A duplicate record or field definition was detected while adding a new database to the Informix tables. If the entry related to the database being added, then this may be regarded as a warning, and no further action is necessary. Otherwise a warning Pop-up will be displayed giving further details of the error.

A.8.2 Rebuild Database

<Rt> <Name1> <Rec#> Record not found for <Name2> <Pri-key>

A master record was not found for this servant record during a total rebuild. The name of the missing master record is shown as <Name2>. The record type, name, and record number of the current record are shown above as <Rt>, <Name1> and <Rec#> respectively. The primary index key of the missing master record is shown as <Pri-key> above. The servant record remains unlinked, meaning that the database remains corrupt. The relevant master record should be added an another total rebuild performed.

<Rt> <Name1> <Rec#> GVA under/overflow <Name2> <Pri-key>

If this error occurs, you will require the assistance of development personnel who have access to the Speedbase Development System. The error indicates that an accumulator residing on a master record has had its capacity exceeded (i.e. a field is simply not large enough to store a required value). The database may be used following this error, but it should be noted that one of the accumulators will be invalid. The affected record can be identified by the record's primary key shown as <Pri-key> above.

In resolving this problem, development personnel should note the following. This error means that the combined total value of GVF fields for servant record type <Rt> has exceeded the capacity of the GVA field in the master record whose name is <Name2> and primary index is <Pri-key>. The GVA field in error should be increased in capacity. This requires re-specification of the database using the Speedbase Development System compiler. Conversion of the database can then be performed using the generation utility (see Chapter 4).

<Rt> <Name1> <Rec#> Dupl primary IDX key <Pri-key>

Two records have the same primary index key value (e.g. two customers numbered 123) have been found. The record concerned is identified by <Name1> and the primary key value <Pri-key>. One of the records should be deleted.

Theoretically, this error can only be caused by suspect database conversion operations, such as reducing the size of an existing primary index key. If this error occurs during a database conversion, it is suggested that you contact your program supplier.

<Rt> <Name1> <Rec#> Illegal index key in <Name2> <Pri-key>

A database record with an index key starting with high-values (#FF) has been detected for the record type indicated. An index key beginning with this value is illegal. The database remains corrupted, and cannot be corrected using functions available to the Speedbase Presentation Manager user.

Again, in theory, this error can only occur as a result of a suspect database conversion. If this is the case, it is suggested you contact your program supplier. Under other circumstances this error indicates that random corruption has occurred to the database, and you should restore the database from backup.

Database rebuild unsuccessful

The database could not be successfully rebuilt owing to one of the above errors. It must therefore be restored from backup.

Database requires total rebuild

The database has been recognised as invalid and a total rebuild of the database is necessary to rectify corruption. Respond Y to the Total Rebuild? prompt.

Dictionary generation# does not match database

The dictionary found is not the same one originally used to generate the database. A matching dictionary and database structure must be present to perform a rebuild.

Dictionary is corrupt or invalid

The rebuild has failed because of corruption within the data dictionary. The dictionary should be restored from backup.

Dictionary not found, in use, or invalid type

The indicated database, dictionary or file was not found on the unit specified, or was found but is in use by another user, or was found to be of an incorrect file type and is therefore not considered to be a valid Speedbase file.

Insufficient memory available

The available user memory area was not large enough to process a particular record type. Refer to section 5.3.4 for more details.

Invalid Unit - errors will be displayed

$PR has not been correctly assigned to a print or a spool disk unit. The error report will not be produced, and errors will therefore be directed to the screen instead.

Rebuild Aborted

An extremely serious error has been detected during the rebuild process, which has caused the rebuild to be aborted.

Schema file is invalid or corrupt

The schema file for the database that you have elected to rebuild is corrupt, and must be restored from backup.

System I/O error nnn

A System I/O error has occurred while accessing a Unix directory. The error number detected is displayed with the message. For more information on error numbers please see the File Converter Manual.

A.8.3 Load Database

Cannot Open C-ISAM file

One of the C-ISAM files comprising the Unix database could not be opened. This may be because the file is missing from the appropriate directory, or may be a temporary condition due to insufficient memory. Try again later and if the condition persists, ensure that the appropriate C-ISAM files are present.

Cannot open Main Index File in Unix Directory

The special index file could not be opened within the Unix directory.

Cannot open source database Datafile(s)

One of the datafiles comprising the source database cannot be opened.

Database Datafiles not found or invalid type

The C-ISAM datafiles could not be found in the appropriate directory, or are not of C-ISAM format.

Database is already open in this partition

The database has already been opened using the $BA command, and is in use. Simply return to the menu and re-run the $BADB command.

Database is in use by another user or partition

The target database could not be opened for exclusive use because it is currently open by another user or partition.

Database is invalid - please rebuild it first

The source database is invalid or corrupt. It should be rebuilt first using the $BARBL utility.

Database must be C-ISAM format

The selected database is a Global format Speedbase database, rather than a Unix format Speedbase database. You may only transfer data to a Unix format Speedbase database

Database must be empty - See help screen for more info

You may only transfer data into an empty Unix C-ISAM database.

Database not found

The specified database schema file or special index files could not be found.

Database not found or in use

The source database from which data is to be transferred was not found on the specified unit, or is in use by another user or partition.

Databases must be same ID and Generation number

Data may only be transferred between databases of the same structure, and must therefore have been created using a dictionary of the same database ID and Generation number.

Dictionary Generation no does not match database

The dictionary describing the format of the source database is does not match the source database. It should be restored from backup, and the source database rebuilt using the $BARBL rebuild utility before proceeding.

Dictionary is corrupt

The dictionary describing the format of the source database is corrupt, and must be restored from backup

Dictionary not found or in use

The dictionary describing the format of the source database could not be found on the specified unit, or is in use by another user or partition.

Dictionary read error

The dictionary describing the format of the source database cannot be read owing to probable hardware error.

Insufficient memory available to complete transfer

The user partition size is too small to allow the transfer program to read one of the record types stored within the database. If the partition size can be increased, then this should be done. Otherwise the database must be redesigned to consist of smaller records.

Insufficient memory available to open - try again later

The target database could not be opened because insufficient memeory is currently available. Try again later.

I/O error - Database cannot be opened

The database could not be opened because of I/O errors. Hardware corruption is indicated.

Source database must be a Global format Speedbase database

The source database from which data is to be transferred must be a Global format Speedbase database.

Source database read error

A hardware read error has occurred while reading the source database. The source database must be restored after the hardware condition has been corrected.

Unable to initialise C-ISAM channels - try again later

A C-ISAM channel could not be initialised, probably as a result of memory non-availability. This error should be transient - try again later.

Unspecified error - Cannot open Database

This error message is included for completeness only and should not arise during normal processing.

A.8.4 Dump Database

This function can result in most of the error messages described in sections A.8.2 and A.8.3. The following additional errors may also occur:

Cannot Delete Old Dump File or File missing

The utility is unable to delete an existing Dump file. This may be because the file has inappropriate file permissions, or is in use by another user.

Cannot Delete database Dictionary from directory

The utility is unable to delete an existing Dictionary file. This may be because the file has inappropriate file permissions, or is in use by another user.

Cannot Delete database dump file from directory

The utility is unable to delete an existing dump file. This may be because the file has inappropriate file permissions, or is in use by another user.

Database Dump already exists on this directory - Delete?

A database dump is already present on the specified dump directory. You should respond "Y" if you want this existing dump to be deleted.

Trouble writing Unix Dictionary

The utility was unable to write to the Unix dictionary file. This may be due to incorrect file permissions.

Trouble reading Global dictionary - Corrupted

The Global dictionary has been corrupted. You should replace the dictionary with a backup copy, and repeat the Dump process.

Trouble Writing Dump File

A write error has occurred while writing to the Dump File. This may be due to incorrect file permissions, or due to an I/O error.

Trouble Deleting C-ISAM files

The utility is unable to delete existing C-ISAM files. This may be because the files have inappropriate file permissions, or are in use by another user.

Unexpected problem opening Unix Dictionary file

The utility cannot open the Unix directory based database dictionary file. This may be because the file has inappropriate file permissions, or is in use by another user.

A.8.5 Regenerate Database

This function can generate many of the errors listed earlier in section A.8. Additional error conditions are as follows:

Cannot Open Dump Set Dictionary
Cannot Open Database Dump file xxxxxx

The Database Dump set Dictionary or Dump file could not be opened on the Unix directory. This may be due to incorrect permissions, or because the file(s) were in use.

Cannot Open Conversion Work File

The conversion Work File could not be opened in the Global work directory. This was probably because there was insufficient space available, or the directory was full.

Cannot Delete Dump set Datafile xxxxxx
Cannot Delete Dump set dictionary
Cannot Delete Dump Set file xxxxxx

The utility was unable to delete one of the above files. This may be because the files have inappropriate file permissions, or are in use by another user.

Conversion file not found or in use

The utility could not re-open the conversion work file. The most likely reason for this is an I/O error on the hard disk.

Conversion file is corrupt

The conversion work file has been corrupted, possibly as the result of an I/O error.

Conversion work space Exhausted

The conversion work file is opened with maximum available file space on the main output database unit. There was, however, insufficient space on this unit. You should make more contiguous space available on this unit, or use another unit.

Dump file Header is inconsistent with Dictionary

The Dump file header found on the Unix directory does not match with the database dictionary. You should re-dump the database and repeat this operation.

Error reading Conversion Work File or Dictionary
Error writing to Conversion work File

An error has occurred reading or writing to the above files. The most likely cause of this problem is a hardware I/O error.

Insufficient memory available for conversion

The dictionary details relating to a particular record type being converted could not be loaded into memory. Refer to section 4.6 for details of memory requirements.

Numeric Field overflow on record Type xx

Conversion processing has attempted to transfer a numeric value to an output field which has been reduced in capacity. The numeric value has exceeded the capacity of the output field and overflow has resulted. The output field will be initialised to binary zeros, and further errors on this RT will be ignored.

Trouble reading Dump Set Dictionary
Trouble reading Dump File xxxxxx

An error has occurred reading the above files. The most likely cause of this problem is a hardware I/O error.

A.8.6 Delete Database

All errors generated by this function are listed earlier in section A.8.

Appendix B - EXIT and STOP Codes

This appendix documents the EXIT and STOP errors that may occur during the running of your Speedbase application. When the Speedbase Presentation Manager detects an error of this type, the EXIT or STOP code is displayed at the baseline. If you see one of these error messages you should report it to your software supplier.

B.1 EXIT Codes

This sections describes EXIT codes generated by the Speedbase Presentation Manager. EXIT codes occur when an exception condition arises within your Speedbase application program and the exception has not been trapped by an ON EXCEPTION or ON OVERFLOW statement. This causes the following error message to be displayed:

```
     $91 TERMINATED - EXIT nnnnn
```

The number nnnnn is the exit code, the meanings of which are described later in this section. It should be noted that only exit conditions that are produced by the Speedbase Presentation Manager are documented here. Those produced by Global System Manager and various Global system routines are described in detail in the Global Utilities Manual.

EXIT 25301	A window was terminated by the <BCK> function.
EXIT 25302	A window (or series of windows) was terminated by the use of the <ABO> function.
EXIT 253nn	An exception condition was signaled by a window Process Routine in order to terminate the window.
EXIT 254nn	An accept operation was terminated using a function other than <RET>.
EXIT 25501	A record locked exception was signaled following a GET, READ, or FETCH statement coded without the NOLOCK clause.
EXIT 25502	Requested record key not found. The index key specified in a READ or FETCH statement was not found.
EXIT 25503	Exception conditions 25501 and 25502 have occurred simultaneously using the READ or FETCH FIRST, NEXT, LAST or PRIOR statements.
EXIT 25504	An end-of-file or start-of-file condition has occurred using a READ or FETCH FIRST, NEXT, LAST or PRIOR statement. This condition can also occur using the GET statement, when the requested RRN key is beyond the logical end-of-file of the target record type.
EXIT 25505	An attempt has been made to DELETE a record with an active (i.e. non-zero) servant record group.
EXIT 25506	The target of a WRITE statement contained a primary index key which already existed on the database.

EXIT 25507	A WRITE statement could not be completed because no free data records remain in the database for the target record type.
EXIT 25508	The RRN key coded for a GET statement specified a deleted record.
EXIT 25511	A date conversion from display to computational format using a MOVE statement failed because the display format date was invalid.
EXIT 25513	A MOUNT statement could not be fulfilled by the operator.
EXIT 25520	The system area $BASYS was signaled as not loaded using the test routine B$CHK.
EXIT 25521	An attempt was made to open the same database twice using database open routine B$OPN.
EXIT 25522	The database open routine B$OPN was unable to open the specified database either because it was not present on the specified unit, or the file-type was incorrect.
EXIT 25523	The data-file(s) belonging to a database could not be opened by the database open routine B$OPN. This problem is usually caused by incorrect unit assignment.
EXIT 25524	A database could not be opened because it was already exclusively opened by another partition.
EXIT 25525	An I/O error occurred during database open.
EXIT 25526	Insufficient memory on user stack to open database.
EXIT 25527	A call on B$OPN could not complete because the DB index file could not be opened within the Unix directory, and the resulting error was not trapped.
EXIT 25528	A call on B$OPN could not complete because a C-ISAM channel could not be initialised, and the resulting error was not trapped. This may occur when insufficient memory is available.
EXIT 25529	A duplicate key condition has been detected within a C-ISAM file and the resulting exception was not trapped. The database should be rebuilt.
EXIT 25530	The system area $BASYS could not be loaded prior to executing a frame. This occurs if the program $BASYS is not found, an I/O error occurs, or if there is insufficient room on the user stack to load this module.
EXIT 25531	Invalid display formatting codes passed to qualifier definition routine B$QLN.

EXIT 25532	The display format codes passed to B$QLN specify positive or negative value highlighting (<>DC+-) although the target operand's format is unsigned.
EXIT 25542	An attempt has been made to run a frame compiled using the V4.0 development system within the V3.0 Speedbase Presentation Manager.
EXIT 25543	An attempt has been made to execute a frame which is not at the anticipated depth in an overlay structure. For example, this will happen when using the SEQUENCE statement to transfer control between frames at different levels in an overlay structure.
EXIT 25544	An attempt has been made to execute a dependent frame which was not compiled with the currently loaded controlling frame.
EXIT 25546	The system area $BASYS could not be loaded prior to executing a frame. This occurs if the program $BASYS is not found, an I/O error occurs, or if there is insufficient room on the user stack to load this module.
EXIT 25547	A frame load has failed because the required program was not found, or was too large to fit into the available memory. This exit condition will also result if an I/O error occurs during loading.
EXIT 25548	The field name coded in a call on the B$WRJ right justification routine was not found in the coded window-id.
EXIT 25549	A call on B$XCL failed because the routine could not provide exclusive access, and the resulting exception was not trapped.
EXIT 25550	A call on B$STA or B$ST2 failed because the required database was not open, and the resulting exception was not trapped.
EXIT 25551	XA$FAM exception condition caused by unusual Unix file processing condition. $$RES holds corresponding result code. This error will not occur in application programs.

B.2 STOP Codes

This section describes STOP codes generated by the Speedbase Presentation Manager. STOP codes occur when a serious processing condition arises within your application program, from which recovery is not possible. This causes the following error message to be displayed:

```
$91 TERMINATED - STOP nnnnn
```

The number nnnnn is the stop code as listed and described in detail later in this section. Note that only stop codes produced by the Speedbase Presentation Manager are documented here. Global System

Manager and various Global system routines also produce stop codes. These are described in detail in the Global Utilities Manual.

Note that certain STOP codes can arise during error checking performed by the Speedbase Presentation Manager, and indicate that the database is corrupt. In this event the database should be restored or rebuilt before any further processing is performed. These STOP codes are marked with a "*".

STOP 25301	An I/O operation has been performed on the I/O channel of a window's target record type during execution of the window.
STOP 25302	An unsupported I/O operation has been attempted by the window processor. This error indicates a system software malfunction.
STOP 25303	A clear operation has been attempted on a window that has not yet been displayed.
STOP 25304	A DISPLAY WINDOW . . TEXT statement has been executed when the text of the window was already displayed.
STOP 25305	An attempt has been made to display a window using the DISPLAY WINDOW statement while the I/O channel of the window's target record type did not contain a record.
STOP 25306	A recursive call has been made on a window that is executing. For example, this error will occur if a window is cleared from the routines section while still executing as a result of a prior entry to the window.
STOP 25307	The index key length coded within a window does not match the key-length of the target record type. This error should not occur when processing records stored on the database.
STOP 25308	A CLEAR statement has been performed while windows are executing. This error can also occur if a dependent frame without the NOCLEAR option has been EXECed while windows are still active in the controlling frame.
STOP 25501	Database not open or generation number mismatch. This condition arises when a database required by the loaded program is not open, or the open data-base has a generation number different from that expected by the program. This error occurs if old, superseded versions of programs are run.
STOP 25502	Rewrite or delete without current record. An application program has attempted to REWRITE or DELETE a record which was not locked at the time the instruction was executed.
STOP 25503	I/O area outside current partition. A database access verb has specified a data record area which is not within the current program partition. Suspect program file corruption or a compiler malfunction.

STOP 25504	Primary index key modification. A REWRITE Instruction has attempted to modify the target record's primary index key. This is illegal.
STOP 25505	Data record not locked. This stop code indicates an internal error within the Speedbase DBMS. The M$DW I/O routine has detected a data re-write without the presence of a record level lock.
STOP 25506	Master record not locked. A WRITE or REWRITE Instruction has taken place in which master records to be linked to the target record were not locked. The DBMS has recovered from this error and completed the I/O request correctly before terminating the program with this code.
STOP 25507	GET key is negative. The RRN specified in the GET verb's KEY clause was negative. This is invalid. This error can also arise if a GET statement is coded without KEY clause. This occurs when no I/O has so far taken place, or the last retrieval returned an end-of-file or start-of-file condition.
STOP 25513	Illegal line or col. The line or column number coded as a variable in a DISPLAY or ACCEPT... LINE statement is invalid. The number was either not positive or else exceeded the boundaries of the screen.
STOP 25514	PF structure exceeds 16 levels. An attempt has been made to execute a PF construct with more than 16 levels of headers or trailers using the PRINT statement. The PF statement may specify a PF as a header or trailer, and this PF may itself specify a further PF as a header or trailer. However, this is limited to a maximum of 16 levels of PF's.
STOP 25515	An internal error has been detected within the database rebuild utility, and it is likely that this is caused by an programming problem within this program. This problem could alternatively be caused by extremely serious corruption within the database or database dictionary files.
STOP 25530	An attempt has been made to run a frame directly, rather than using the $BA command.
STOP 25540	An unsupported ACCEPT or DISPLAY statement call has been made by a Global Cobol module linked into a Speedbase frame.
STOP 25541	An indexing error has occurred within a copy-library specified during a compilation using $SDL. The copy-library is probably corrupt.
STOP 25550	An internal system error has occurred. You should write down the details from the screen and note in detail what you were doing just before the stop code occurred. This information should be forwarded to your software supplier for analysis.

STOP 25551	Speedquery was unable to find the database dictionary (or it was in use) on the same volume as the database itself. Make sure that the dictionary is available to Speedquery.
STOP 25552	An irrecoverable I/O error occurred while reading the database dictionary. This probably means that the dictionary has become corrupted and should be restored from backup. Care should be taken to ensure that the correct version is recovered.
STOP 25553	Speedquery has exhausted the available memory during the execution of a critical part of the query and was unable to recover. You should simplify the query or increase the user partition size to obtain more memory.
STOP 25554	An irrecoverable I/O error has occurred while trying to read or write the query work file. This could be caused by a bad track on the $P sub-volume. You should verify the volume before continuing.
STOP 25555	Speedquery was unable to create a work file large enough for the current query. You should allocate more file space on the current $P volume before restarting the query.
STOP 25556	Speedquery has detected a corrupt database index. You should perform a rebuild or a restore to recover the index file.
STOP 25557	Speedquery has detected that the database data files are corrupt. You should perform a rebuild or a restore to recover the database.
STOP 25558	Speedquery has detected that the saved query file requested is corrupt. You should perform a restore to recover the query from backup.
STOP 25559	Speedquery was unable to find the saved query requested.
STOP 25560	Incompatible saved query. Speedquery has detected that the database dictionary has been modified since the query was saved. You must re-develop the query.
STOP 25561	Speedquery was unable to open the print file. This could be caused by a system error, no room on the spool unit, or no room available on the $PR volume. Please check before continuing.
STOP 25562	An irrecoverable I/O error occurred while trying to write to the print file. This could be caused by a system corruption, lack of space on the spool unit, lack of space on the allocated $PR sub-volume, or a hardware error.
STOP 25563	The DX file created to hold the database structure is too small. This is caused by an excessively complex

database. Contact your software supplier with the details.

STOP 25564	A bad header was found in the dictionary. This may mean that the dictionary has somehow become corrupted and will have to be restored from backup. Care should be taken to ensure that the correct version is recovered.
STOP 25565	Speedquery was unable to read the dictionary. This may mean that the dictionary has somehow become corrupted and will have to be restored from backup. Care should be taken to ensure that the correct version is recovered. This error may also be caused by a bad disk track, in which case you should verify the dictionary.
STOP 25566	An irrecoverable I/O error occurred while opening or closing the DX file. This could be caused by a system corruption, a corrupt DX file, or a bad disk track. Verify the sub-volume, then delete the existing DX file. When re-started, Speedquery automatically recreates the DX file.
STOP 25567	An irrecoverable I/O error occurred while reading or writing the DX file. This could be caused by a system corruption, a corrupt structure file, or a bad disk track. You should verify the sub-volume then delete the existing DX file. When restarted Speedquery will recreate the DX file.
STOP 25568	Insufficient room for the DX file on the database sub-volume. Speedquery needs 4.5 Kbytes of spare space to create the DX file. You should allocate the space then restart Speedquery.
STOP 25569	An I/O error occurred while trying to read the list of available files on the $P sub-volume. This is usually caused by a corrupt directory. Please check and rectify before continuing.
STOP 25570	An I/O error occurred while reading or writing the sort work file on the SQW sub-volume. Verify the sub-volume before continuing.
STOP 25571	An internal system error has occurred while trying to perform a sort. You should write down the details from the screen and note in detail what you were doing just before the STOP code occurred. This information should be forwarded to your software supplier for analysis.
STOP 25580	Speedbase was unable to open the pop-up image swap file required by the last loaded frame. This will occur if there is insufficient free space on logical unit "BAW", or if an I/O error occurs during opening of the swap-file.
STOP 25581	An I/O error has occurred during swap-file processing. This may be due to a disk error.

STOP 25582	Using the BCDB, BXCL or B$XSH routines, an attempt was made to access a database which was not open at the time of the call.
STOP 25583	I/O error in B$XFAM. The Unix error result code will have been displayed immediately before this stop code. This error will not occur in application programs.
STOP 25584	Unsupported I/O Request in XA$FAM. Internal Systems Software error.
STOP 25586 *	The next free data-record slot was permanently locked during a WRITE operation.
STOP 25587 *	Find exceeds 16 IDX Levels. A random index search has exceeded 16 levels of index.
STOP 25588 *	Dummy high not found in IDB scan. The HIGH-VALUES index terminator was not found during an index scan.
STOP 25589 *	GVA over/underflow. A WRITE, REWRITE or DELETE statement has caused a GVA field to overflow on one of the associated Master records. GVA overflow will occur if the computational capacity of the field is exceeded.
STOP 25590 *	Link fails by permanent lock. A WRITE or REWRITE Instruction failed because one of the master records to be linked was not locked. The DBMS recovery procedure was then unable to recover from this error because the master record to be linked was permanently locked by another partition.
STOP 25591 *	Link fails by non-existent master. A WRITE or REWRITE instruction failed because one of the master records to be linked was not locked. The DBMS recovery procedure was then unable to recover from this error because the master record specified by the master access key did not exist.
STOP 25592 *	Index key not found in delete. During the DELETE verb processing, all index key entries referencing the record to be deleted are normally removed. The DELETE verb was unable to find one of these index keys during processing. See Note 1.
STOP 25593 *	Illegal index key. A WRITE or REWRITE verb has attempted to create an index entry starting with a high-values byte (#FF). This is illegal.
STOP 25594 *	Free IDB pool exhausted. A WRITE or REWRITE instruction required a free index block to complete an index key addition. The free index pool was however empty.
STOP 25595 *	Link to unused or deleted IDB. An error has been detected where an index block is incorrectly linked to a deleted or unused IDB.

STOP 25596 * Irrecoverable I/O error has taken place during an IDB or data transfer.

STOP 25597 * Negative IDB or data record. The IDB/data transfer routine within the DBMS has detected a request for a negative record number. Internal error.

STOP 25598 * File condition. A file condition has arisen during a data transfer to or from disk, such as an attempt to transfer data to or from an area out-side the database file extents.

STOP 25599 * IDB and record key mismatch. An index entry has been read which does not match the index key held on the record as stored on the database.