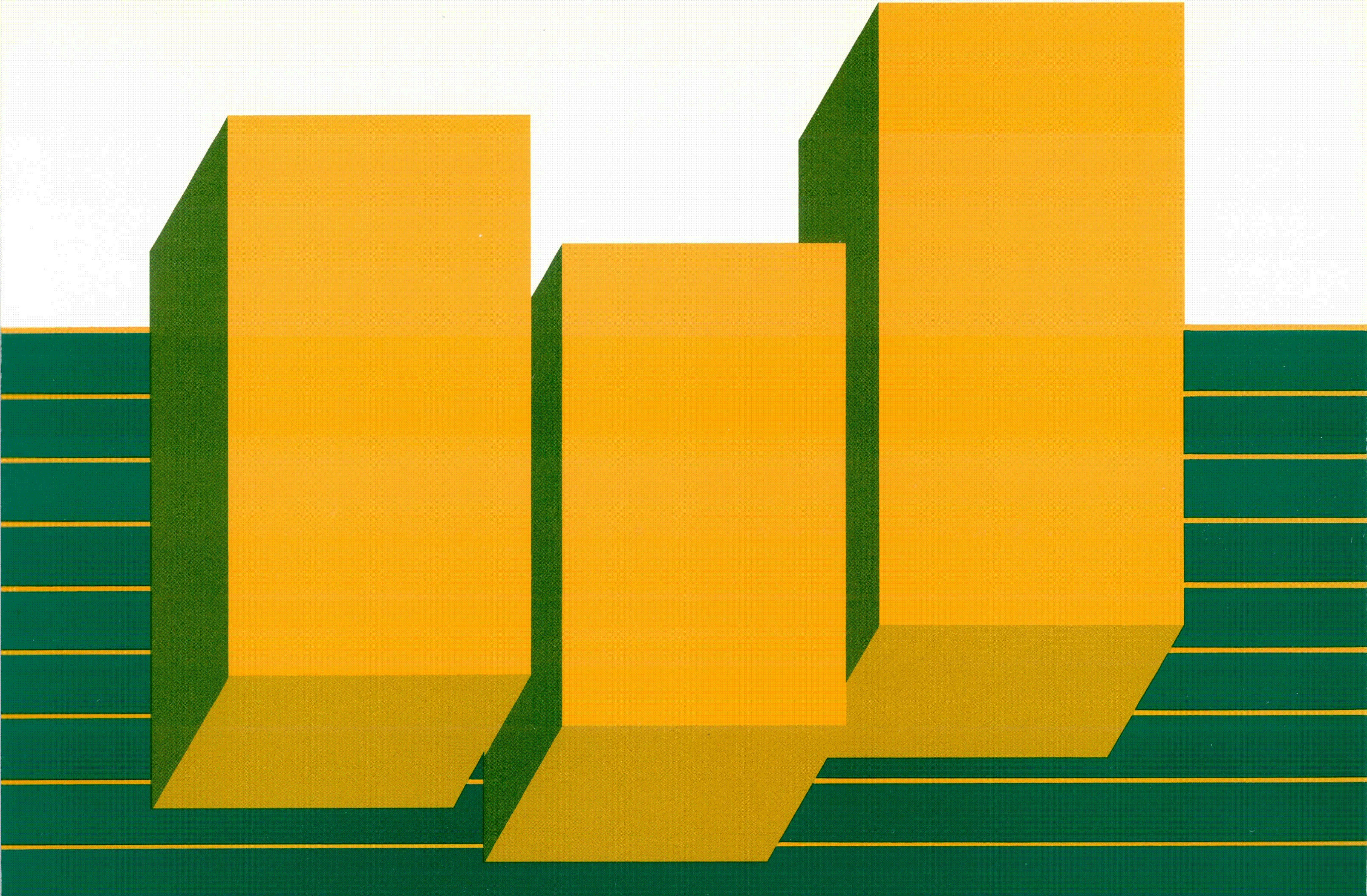
DSG/3000 Manual

HEWLETT
PACKARD



HEWLETT
PACKARD

HP 3000 Computer Systems

DSG/3000 Manual

Part No. 32250-90001

Product No. 32250A

Printed in U.S.A. 8/82

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

PRINTING HISTORY

New editions are complete revisons of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The data on the title page and back cover of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the editions are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

The software product part number printed alongside the date indicates the version and update level of the software product at the time the manual edition or udpate was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

The second edition of the DSG/3000 Manual incorporates several new enhancements which reflect the increased power of the program, whether it is run interactively or programmatically.

A range of new text fonts, text sizes and choices of colors allows for improved chart design. Native language characteristics for several languages are available to users.

Figure files may be created which make DSG/3000 charts available for use with HPDRAW, TDP/3000, HPMAIL and the HP2680A (Laser Printing System).

New peripheral devices have also been added to the list of those supported by DSG/3000. PASCAL has been added to the list of supported languages.

First Edition		Nov 1980
Update		Sept 1981
Second Edition	32250A.01.00	Aug 1982

CONVENTIONS USED IN THIS MANUAL

NOTATION	DESCRIPTION
[ ]	An element inside brackets is *optional*. Several elements stacked inside a pair of brackets means the user may select any one or none of these elements. Example: $\begin{bmatrix}\mathrm{A}\\\mathrm{B}\end{bmatrix}$ user may select A or B or neither
{ }	When several elements are stacked within braces the user **must** select one of these elements. Example: $\left\{\begin{matrix}\mathrm{A}\\\mathrm{B}\\\mathrm{C}\end{matrix}\right\}$ user must select A or B or C.
italics	Lowercase italics denote a parameter which must be replaced by a user-supplied variable. Example: CALL *name* *name* one to 15 alphanumeric characters.
. . .	A horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.
underlining in dialog	When it is necessary to distinguish user input from computer output, the input is underlined.

Contents

SECTION 1 -- INTRODUCTION

SECTION 2 -- GRAPHING TERMS AND CONCEPTS

SECTION 3 -- CREATING A DATA FILE

SECTION 4 -- USING DSG/3000 INTERACTIVELY

SECTION 5 -- MENU SPECIFICATION

SECTION 6 -- USING DSG/3000 PROGRAMMATICALLY

SECTION 7 -- PROCEDURES - REFERENCE SPECIFICATIONS

APPENDICES

List of Figures

SECTION I

SECTION 2

SECTION 3

SECTION 4

SECTION 5

SECTION 6

SECTION 7

List of Tables

SECTION 1

SECTION 3

SECTION 4

SECTION 5

SECTION 7

Preface

Decision Support Graphics/3000 (DSG/3000) is designed to produce graphic displays of data. It operates on the HP 3000 computer system.

DSG/3000 provides these capabilities:

o design of pie charts, bar charts, line charts and scattergrams

o data input from any suitably formatted MPE file

o interactive chart design and production through screen menus requiring no programming

o programmatic chart design and production through procedures callable from COBOL, BASIC, FORTRAN, PASCAL, and SPL

DSG/3000 can be used by anyone who wishes to design or produce graphs from data stored in the HP 3000. Graph designers may be managers, administrators, or other personnel with or without programming experience. The non-programmer can perform the same functions interactively as a programmer writing in one of the above languages.

In order to use DSG/3000, you should know how to operate the terminals and plotters used at your facility. If you use the programmatic interface, you may also need the manual for the language in which you code.

You may wish to consult the following manuals:

Manual	Part Number
DSG/3000 Guide	32250-90002
Using the HP 3000	03000-90121
MPE Commands Reference Manual	30000-90009
MPE Intrinsics Reference Manual	30000-90010
Using Files	30000-90102
VPLUS/3000 Reference Manual	32209-90001
QUERY/3000 Reference Manual	30000-90042
EDIT/3000 Reference Manual	03000-90012
BASIC/3000 Interpreter Reference Manual	30000-90026
BASIC/3000 Compiler Reference Manual	32103-90001
FORTRAN/3000 Reference Manual	30000-90040
SPL/3000 Reference Manual	30000-90024
PASCAL/3000 Reference Manual	32106-90001
Text & Document Processor/3000	36578-90001

And those manuals associated with your terminals and plotters.

A self-paced course, *Using Decision Support Graphics/3000*, product number 22833A, is also available.

SECTION 1

Introduction

HOW TO USE THIS MANUAL

Because this manual must answer the needs of inexperienced users as well as programmers, its contents range from very basic to highly detailed information. Section 1 need only be read by those who are unfamiliar with graphs and with the equipment needed to operate DSG/3000. Section 2 provides explanations of concepts and terminology which are referred to throughout the manual. It is essential to the new user, and will provide review materials for those who use DSG/3000 infrequently.

Section 3 covers the six methods of creating data files and will aid you in making decisions as to how you wish to handle yours. Only those who plan to use DSG/3000 interactively need to read Sections 4 and 5, while programmatic users will rely on Sections 6 and 7 for information. A brief description of the contents of each section follows.

Section 1, *Introduction*, contains general information about DSG/3000 including examples of basic graphs that you may create, a list of features and their uses, and illustrated overviews of the subsystem's components. It concludes with a discussion of a typical graphics terminal keyboard and information about plotters and printers supported by DSG/3000.

Section 2, *Graphing Terms and Concepts*, familiarizes the reader with the purpose of graphs, how to select a graph type for a particular use, and terminology associated with graphing. Numerous examples point out the enhancements available which increase the effectiveness of a graph.

Section 3, *Creating a Data File*, describes the six methods available for declaring the values of your input data. Because a data file must exist before a graph can be drawn, this section precedes those which focus on creating the graph. The methods range from highly interactive to programmatic use.

Section 4, *Using Graph Interactively*, is devoted to running the interactive program, GRAPH. It covers the use of the special function keys, logging on, characteristics of the program and the screen display. It concludes with a general introduction to the screen menus, menu loops, and how menus are used in designing, enhancing, and modifying a chart.

Section 5, *Menu Specifications*, lists alphabetically all the menus and legal entries for all the fields you may fill in, your options, default values; it also includes selected examples.

Section 6, *Using DSG/3000 Programmatically*, serves as a general introduction to the use of procedures when DSG/3000 is run programmatically. Following an overview and summary of the procedures, the way in which they are used to produce and enhance the different graph types is covered.

Section 7, *Procedures--Reference Specifications*, contains formatting information for calling procedures from different program languages, data type and communications area information, and an alphabetical coverage of the procedures, their parameters, and selected examples of their use in programs.

Appendix A contains tables showing the available textures and colors as well as their order of use. Examples of programs written in COBOL, BASIC, SPL, PASCAL, and FORTRAN are included in *Appendix B*. *Appendix C* contains a list of error messages. Maximum limits for interactive and programmatic use appear in *Appendix D*. *Appendix E* contains information regarding the use of figure files. *Appendix F* instructs the user in converting DSG/3000 files created using an earlier version. They may be updated interactively or programmatically. *Appendix G* contains detailed information on the plotters and terminals supported by DSG/3000. (Supported devices are listed at appropriate places throughout the manual.)

Introduction to DSG/3000

HP Decision Support Graphics/3000 is a data display system that allows you to design, produce, and save business graphs drawn from information kept in a data file. The graph can be displayed on a graphics terminal, or a permanent copy can be made on a plotter. The devices supported by DSG/3000 are listed later in this section.

DSG can be used interactively through a series of screen menus, or programmatically through a set of procedures. Both the interactive user and the programmatic user have access to the full capabilities of the product.

HP Family of Graphics

HP has been a major contributor to the field of technical graphics for years, producing top-quality printers, plotters, standalone graphics terminals, and desktop computers. Software has been developed which extends the use of graphics to business. Besides DSG/3000's extensive capabilities, HP offers the following graphing tools to its users:

HPEASYCHART -- a chartmaker that allows a variety of office workers to produce high-quality graphs and charts in in minutes by filling out a few simple menus. It is ideal for one-time chart making when it isn't necessary to access stored data.

HPDRAW -- a visual aid design tool with which you can create simple figures using lines, boxes, arrows, and circles and save them for later use. You can also include charts developed by DSG/3000 or HPEASYCHART on regular or flip-top sized paper or overhead transparencies.

HP2680A GRAPHICS PACKAGE -- allows you to merge text and graphics on the HP2680A Laser Printing System, eliminating tedious cutting and pasting in the production of manuals and reports,which you can print on demand -- at 45 pages per minute.

Types of DSG/3000 Graphs

DSG/3000 enables you to draw the following types of graphs:

-pie charts
-line charts
-scattergrams
-horizontal and vertical bar charts
-slides consisting of simple annotations

Some examples of graph types follow on the next few pages.

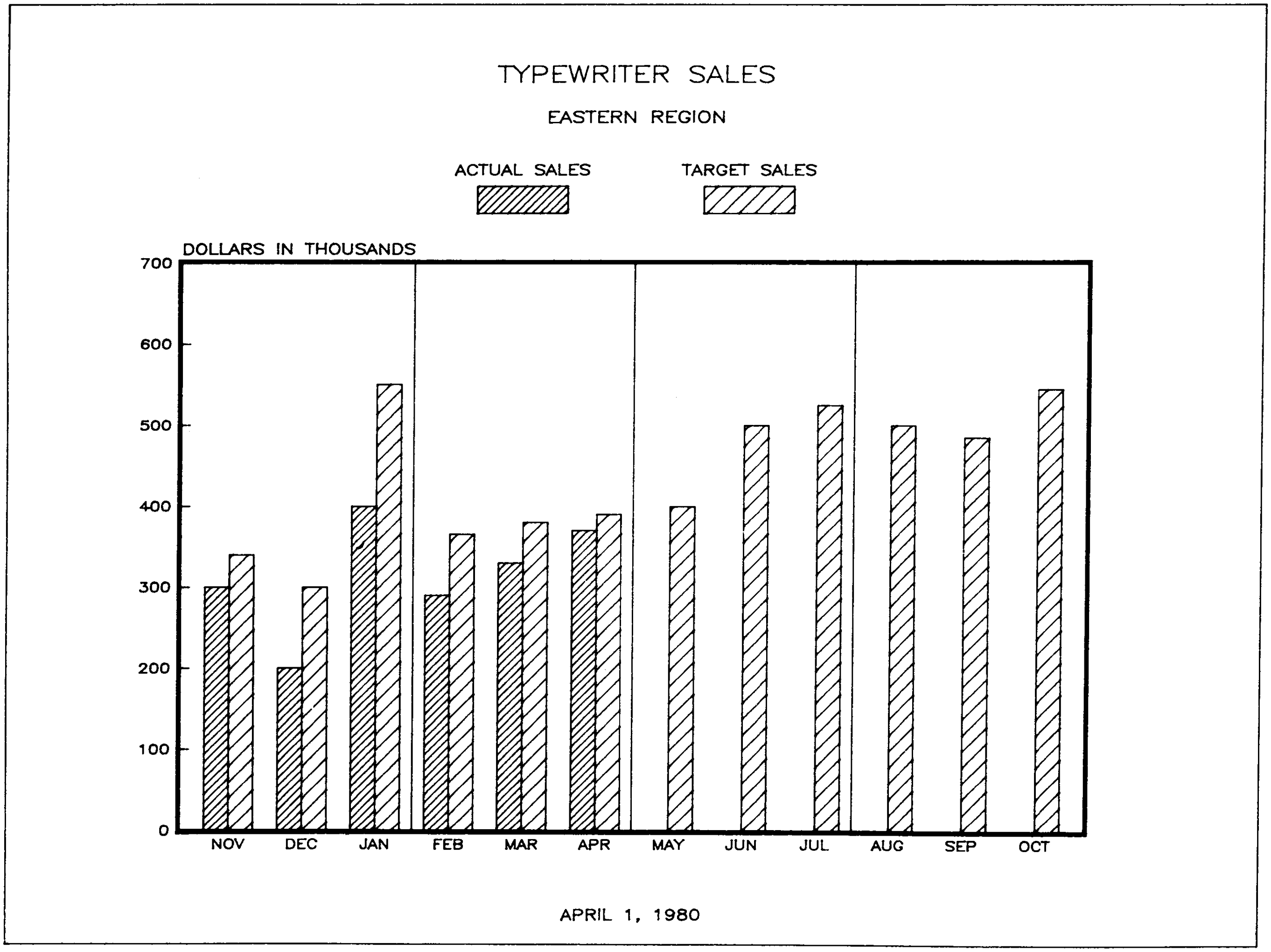


Figure 1-1. Bar Graph

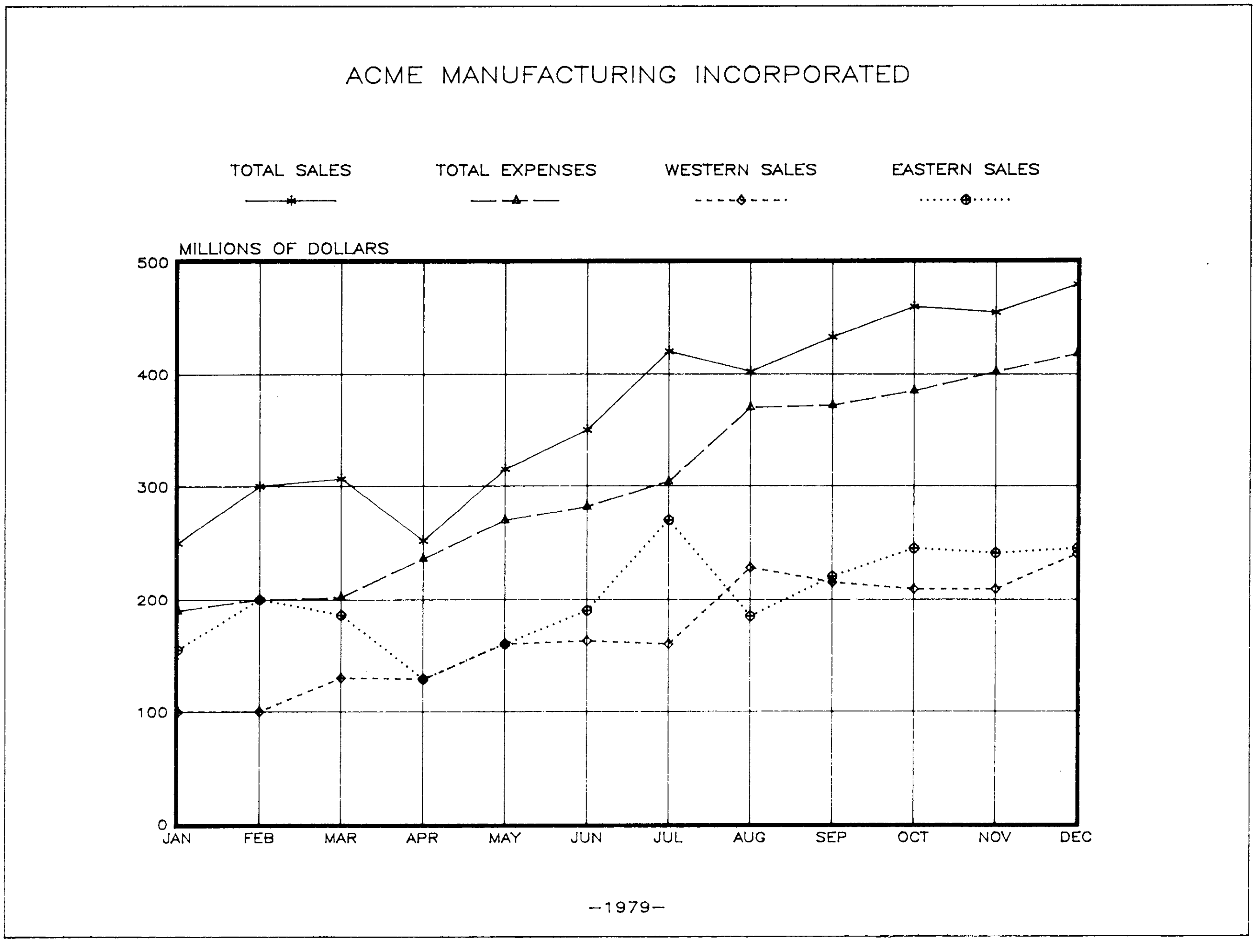


Figure 1-2. Line Graph

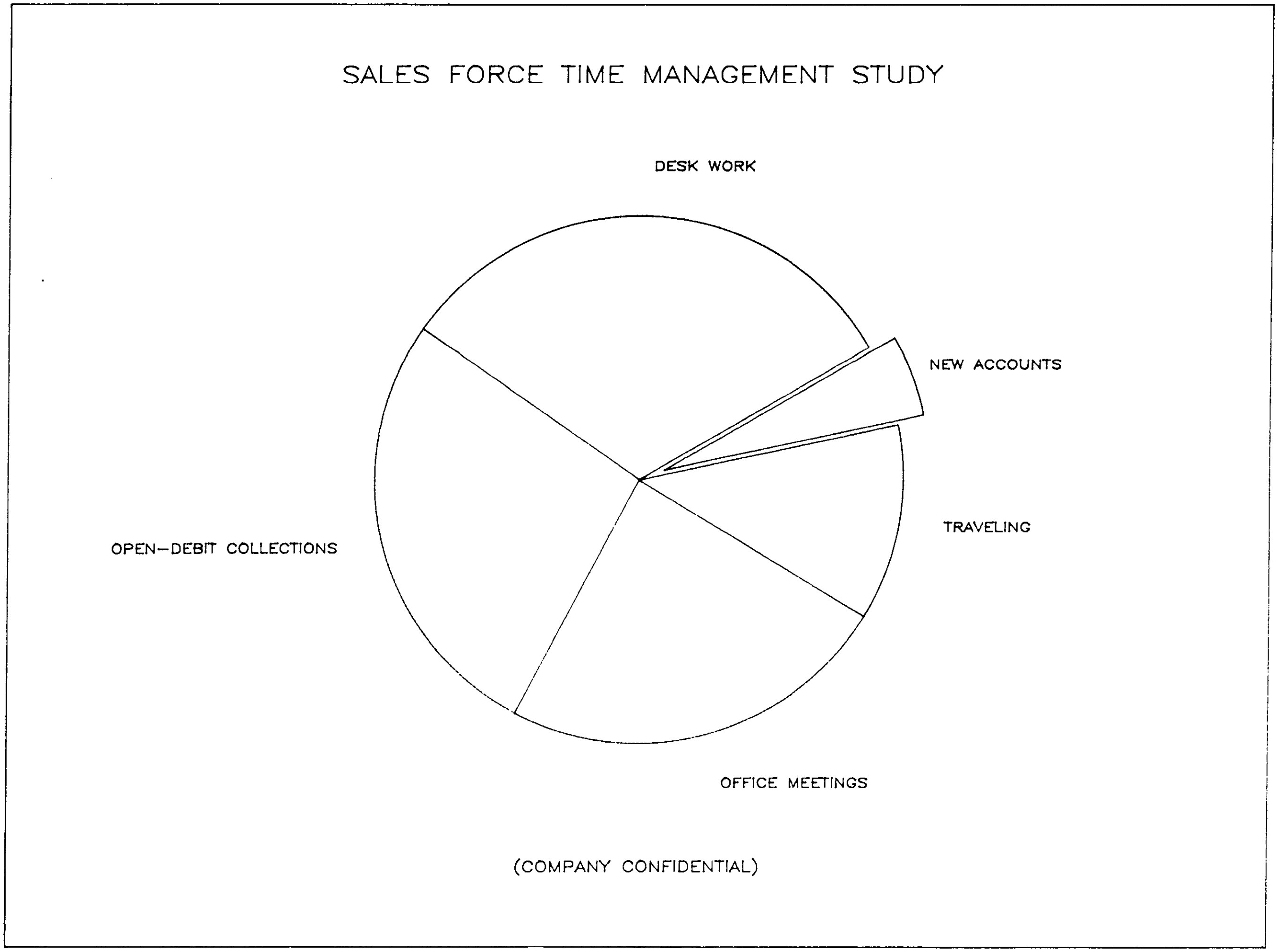


Figure 1-3. Pie Graph

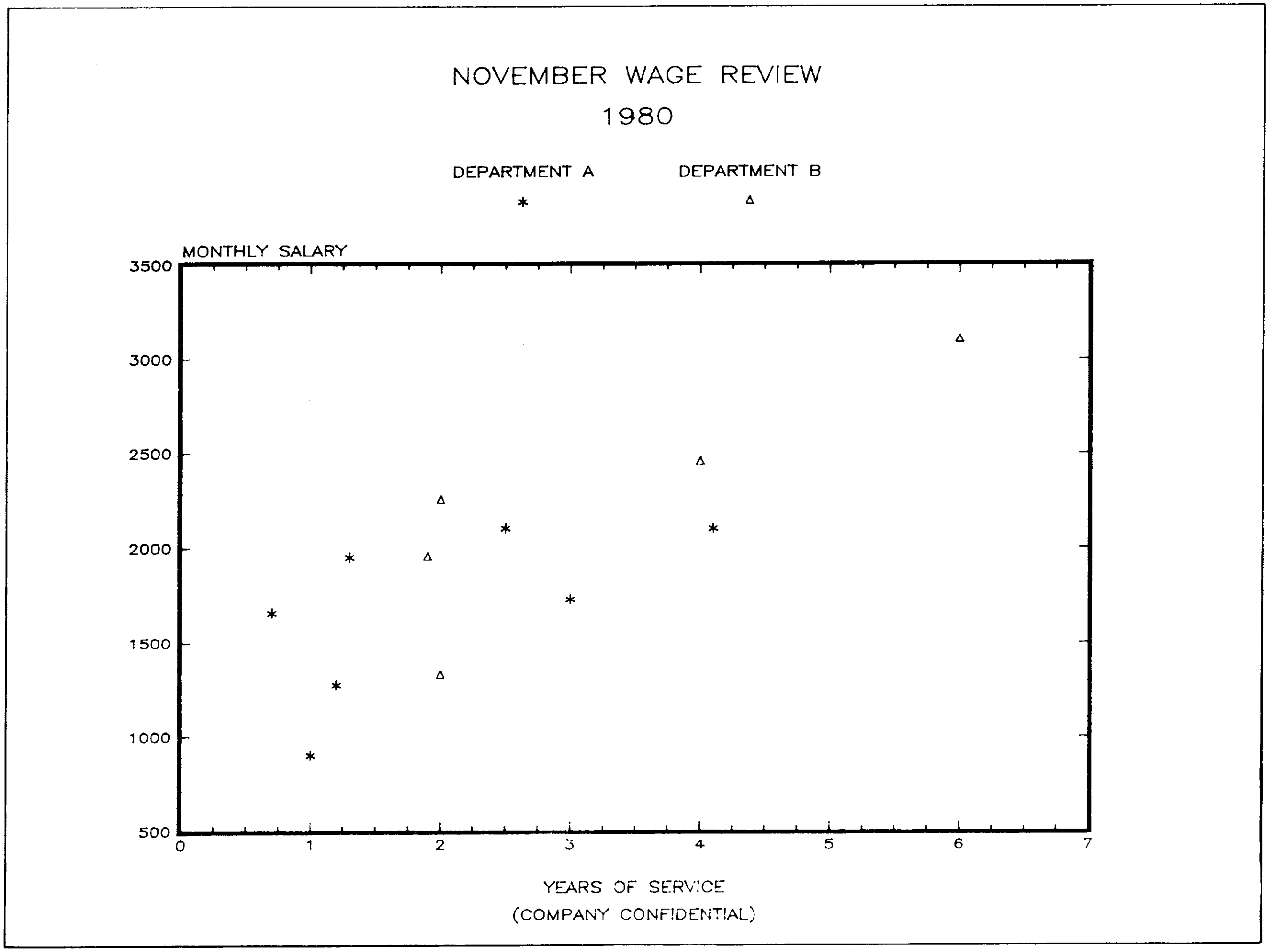
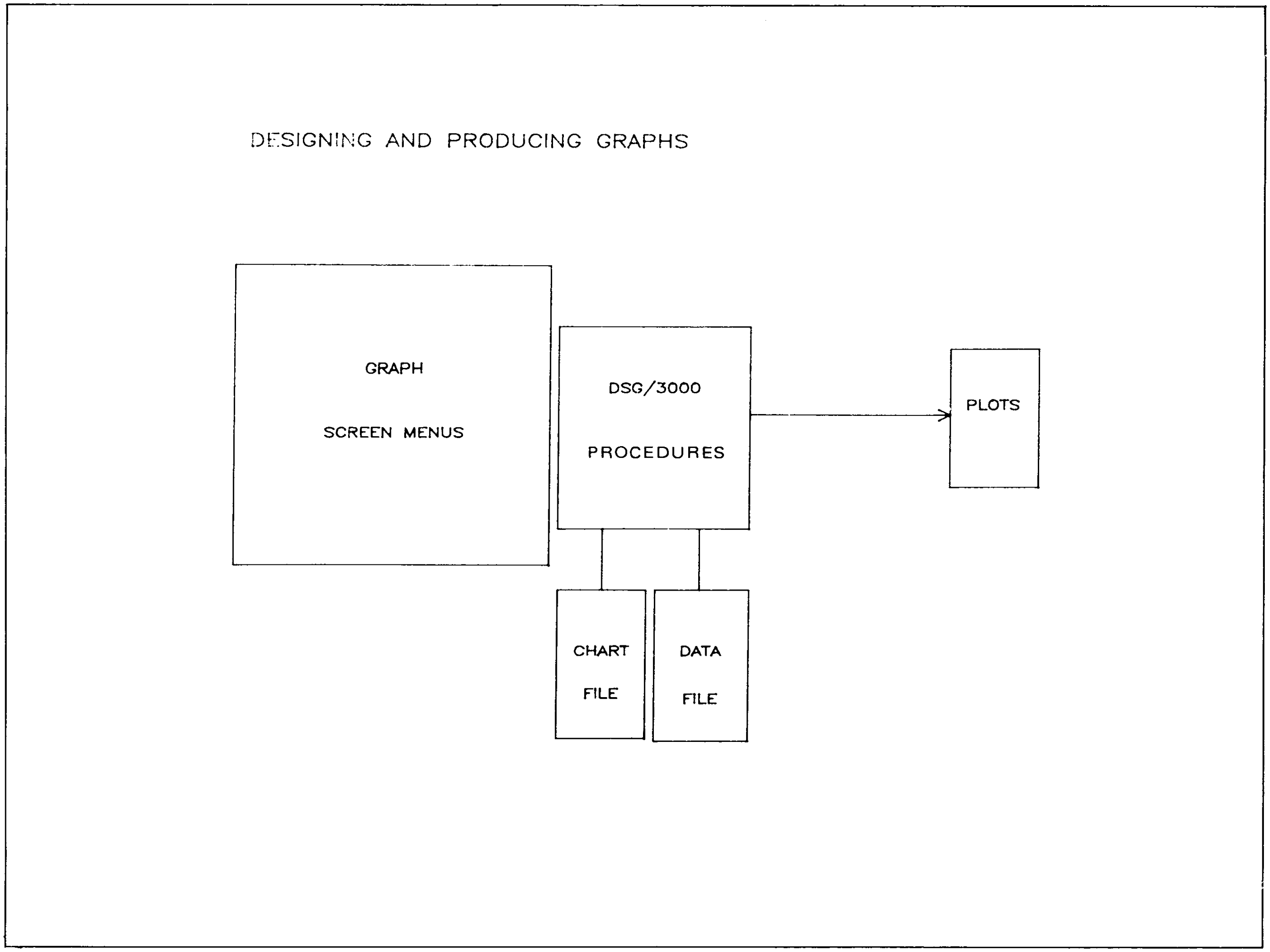


Figure 1-4. Scattergram



DESIGNING AND PRODUCING GRAPHS

GRAPH

SCREEN MENUS

DSG/3000

PROCEDURES

PLOTS

CHART

FILE

DATA

FILE

Figure 1-5. Slide

Capabilities and Features

The following are the main features of DSG/3000:

- o Data can be taken from any MPE file that is suitably formatted.
- o Data can be automatically scaled for you by DSG/3000.
- o Since chart design is independent of the data, data can be updated or modified without changing the chart specifications.
- o Arithmetic and special functions can be applied to your data and plotted.
- o Chart design is device independent.
- o Chart specifications can be stored for periodic use.
- o Graphs can be designed to be plotted in multiple colors when output is routed to a multi-color device.
- o A collection of charts can be drawn periodically in production mode. If a scrollable plotter is used, this process can be completely automated.
- o Graphs can be annotated with symbols and text.
- o Graphs drawn by DSG/3000 are accessible to other subsystems through the creation of figure files.

Access

DSG/3000 can be accessed interactively through the program GRAPH which generates screen menus, or programmatically through procedures.

Interactive Access

The DSG/3000 interactive program, known as GRAPH, is designed for ease of use. Chart design is guided by screen menus presented in logical sequence with fields to be filled by the user. No special command language is needed, and a special HELP facility built into the system can be called upon for assistance when required.

GRAPH also provides flexibility of design. Many options and enhancements are available to the user, but sensible defaults are provided so that you can produce high quality charts quickly and easily. At the user's option, GRAPH will dynamically alter specifications such as axis scaling and labeling to adapt to changes in the data.

Programmatic Access

User application programs can use DSG/3000 procedures tó create, modify, and produce charts. This capability enables programs which periodically generate reports to include graphs as a tool to clarify data, present a summary, or emphasize a point.

Data and Chart Specifications

In order to produce a graph, you must specify both the chart attributes and the data file. A *chart* contains the skeletal characteristics of a graph such as graph-type, title, legends, and annotations. Several different charts can be stored in a single *chart file.* A chart file is a graphics portfolio. The *data file* contains the actual values of the variables to be plotted, and is completely separate from the chart definition.

DSG/3000 keeps the chart specifications separate from the data so that you can modify data values without making changes to the contents of the chart file.

Periodic Use

Once a chart has been designed and saved in a *chart file*, you may use the chart periodically with different or revised sources of data. An interactive option allows you to draw all of the charts in a chart file at the same time.

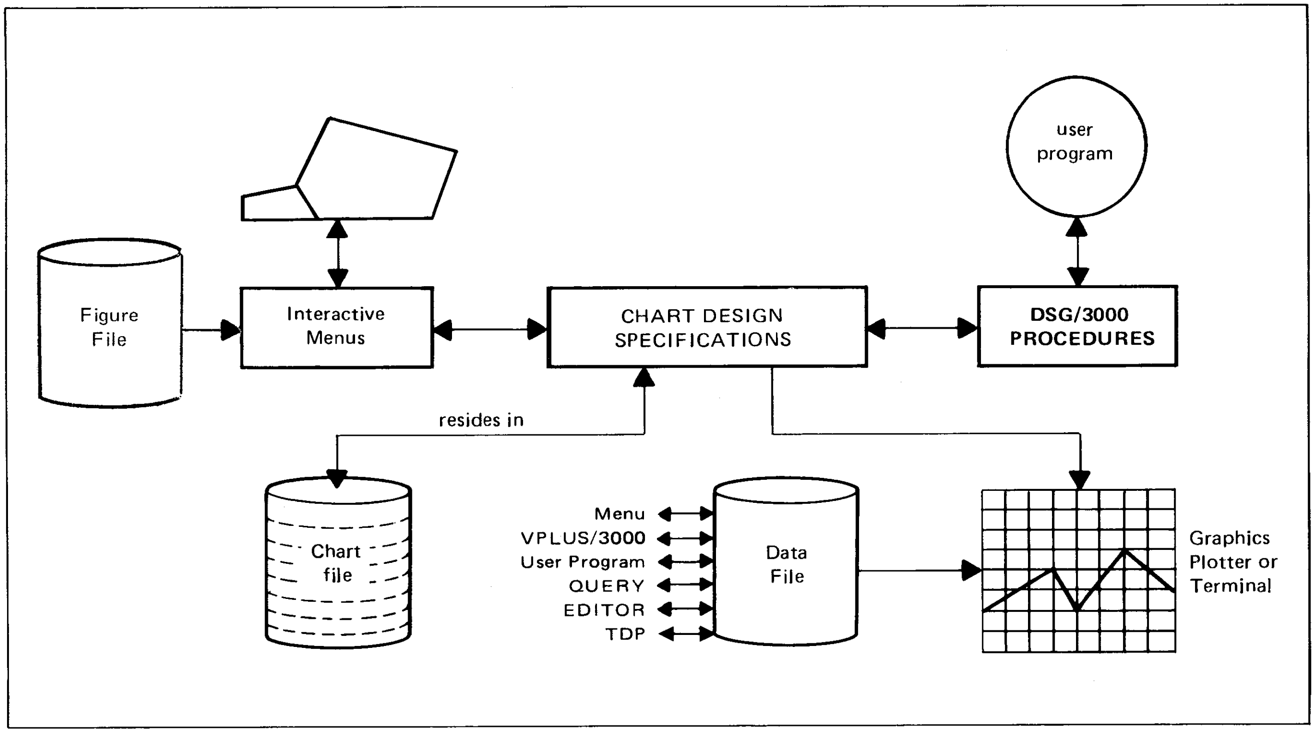


Figure 1-6. DSG/3000 Overview

Charts and Chart Files

A user may create several different chart files, each containing charts that relate to one another or that are often drawn at the same time. You can access one chart file at a time; this is called the *current chart file*. Within the current chart file, you can only add or modify one chart at a time; this is called the *current chart*. Figure 1-2 illustrates the current chart and chart file.

The file code for chart files is 1083.

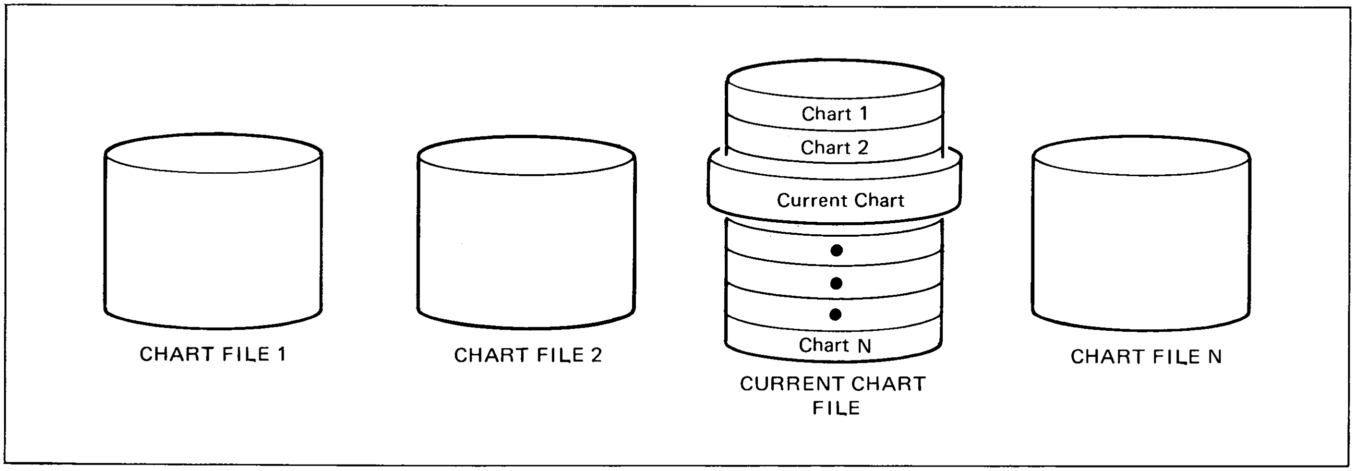


Figure 1-7. Current Chart File and Current Chart

Accessing Other Subsystems

Users of other subsystems such as HPDRAW, TDP/3000, and the 2680A (Laser Printer) can access graphs and slides produced and stored by DSG/3000 through the creation of *figure files*. Figure files provide long term storage for graphical information. They offer the following features:

o a means of transporting device independent graphical information from one office systems application to another.

o a method of storing a combination of chart and data information; if definitions later change, figure will not change.

o an archival medium for storing graphical information.

When you store a chart or slide in a figure file, you are effectively drawing it to the file. The information that will be stored in the file is complete and cannot be changed. Unlike the contents of chart files, which include only chart information, figure file contents include both data and chart information.

DSG/3000 can create figures but not manipulate them. Subsystems such as HPDRAW can place the figures built by DSG/3000 in a drawing and perform operations such as scaling, rotating, and moving. Modification of data or chart must be undertaken before the image is stored in a figure file. Multiple figures, each with a unique name, may be created. The contents of figure files can only be displayed by way of another subsystem.

Data Sources

The data file, containing the values of the data variables, is an MPE file containing ASCII data that can be generated by:

-QUERY/3000
-VPLUS/3000
-EDIT/3000
-TDP/3000
-user report-generating programs
-GRAPH data-entry screen

DSG/3000 also allows you to create new data from original data values and plot the new, transformed values. The following types of transformations are provided:

Arithmetic Operations	Add, subtract, multiply, divide, modulo, and exponentiate
Common Functions	Common log, natural log, square root, and absolute value
Special Functions	Cumulate, moving average

A Graphics Terminal Keyboard

It is recommended that you run DSG/3000 on a supported graphics terminal if possible. These terminals produce the screen menus in standard alphanumeric mode and draw the graphs in graphics mode. If you are not using a graphics terminal see "Selecting a Device" in this section for information about non-graphics terminals and devices to which graphic output can be directed.

The keyboards of the 2647A and 2648A graphics terminals are similar. Refer to Figure 1-8, showing the 2647A terminal keyboard, to locate some of the special keys used in GRAPH. (See the User's Manual that describes your terminal for complete instructions on terminal use.) The keyboard of the 2623A graphics keyboard is shown later in this section.

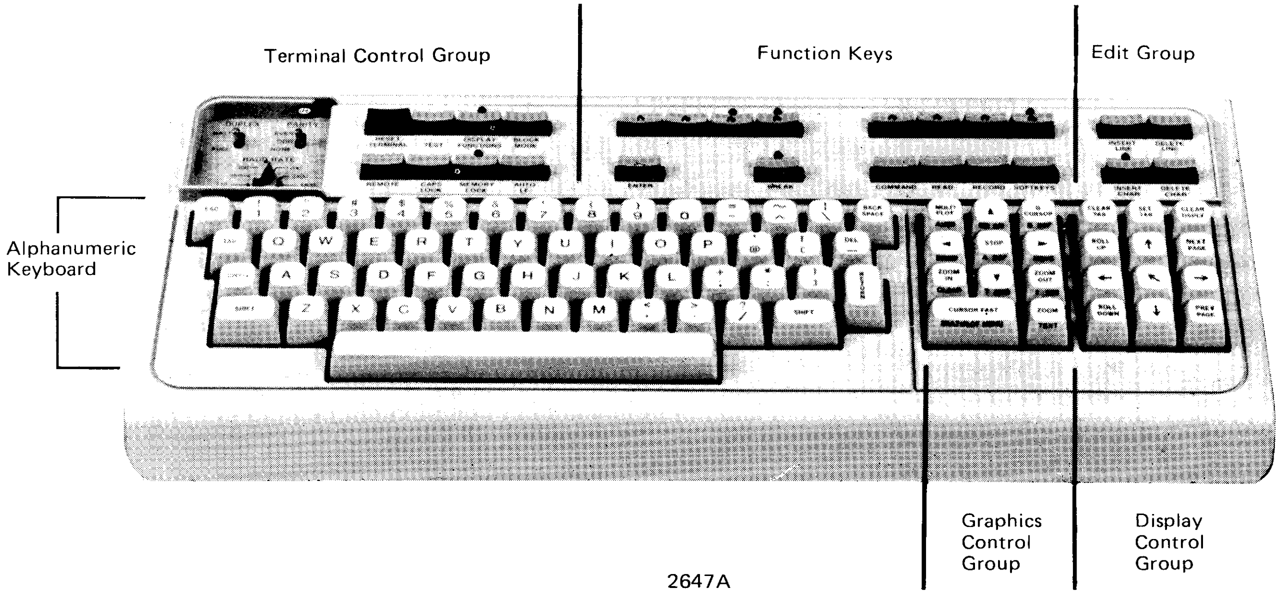


Figure 1-8. 2647A Graphics Terminal Keyboard

Character Set Group

Use the TAB key to go from one item (field) to the next in each screen menu. If you are using a 2647 or 2648, you can move the cursor backwards by holding down the CONTROL key (labeled CNTL) while you press the TAB key. If you are using a 2623A you hold down the SHIFT key and press TAB to move the cursor back.

Graphics Control Group

If you are using a 2647 or 2648 and you wish to turn the alphanumeric display on and off, hold down the SHIFT key and press the key front-labeled "A DSP".

To turn the graphics display on and off, hold down the SHIFT key and press the key front-labeled "G DSP". (You will not be able to test this effect until you enter some graphics into graphics memory.)

The 4 keys labeled with heavy pointers control the movement of the graphics cursor. Two keys can be pressed simultaneously for diagonal motion. Use these keys when positioning annotations on the graphics terminal.

If you are using a 2623A, you will find some inconsistencies with these groups. Check your user's manual for specific information about the terminal's operation.

Edit Group

Because GRAPH operates in block mode, you can key all of the information into a single menu before actually sending it to the computer. You can use the INSERT CHAR and DELETE CHAR keys to insert and delete characters if you wish.

Display Control Group

The Cursor Control Keys, marked with arrows, move the cursor around the screen menu. The NEXT PAGE and PREV PAGE keys can be used to go from one page to another for those HELP Roadmap menus that take up more than one page.

The CLEAR DISPLAY key can be used to clear all of the information from the unprotected fields, starting with the cursor location. To clear all unprotected fields on the menu, press the Home Cursor key (↖), followed by the CLEAR DISPLAY key.

Terminal Control Group

To send information that you have entered in a menu to the computer, you must press the ENTER key. The computer has no knowledge of any changes you may have made on a screen until you actually enter it with this key, because GRAPH runs in block mode.

The BREAK key and the RESET TERMINAL key should only be used under unusual circumstances discussed in Section VI, Using DSG/3000 Interactively.

Special Functions Group

The keys labeled f1 through f8 have special functions assigned to them by GRAPH. When you enter the Help Facility, these functions change. These functions are discussed in detail in Section VI and their purposes are described in association with individual menus in Section V.

The use of special function keys in setting the HP-IB address is discussed under Raster Printers in this section.

Figure 1-9 below is a photograph of the 2623 keyboard with key groups highlighted. Your terminal manuals will explain the functions of the various groups.

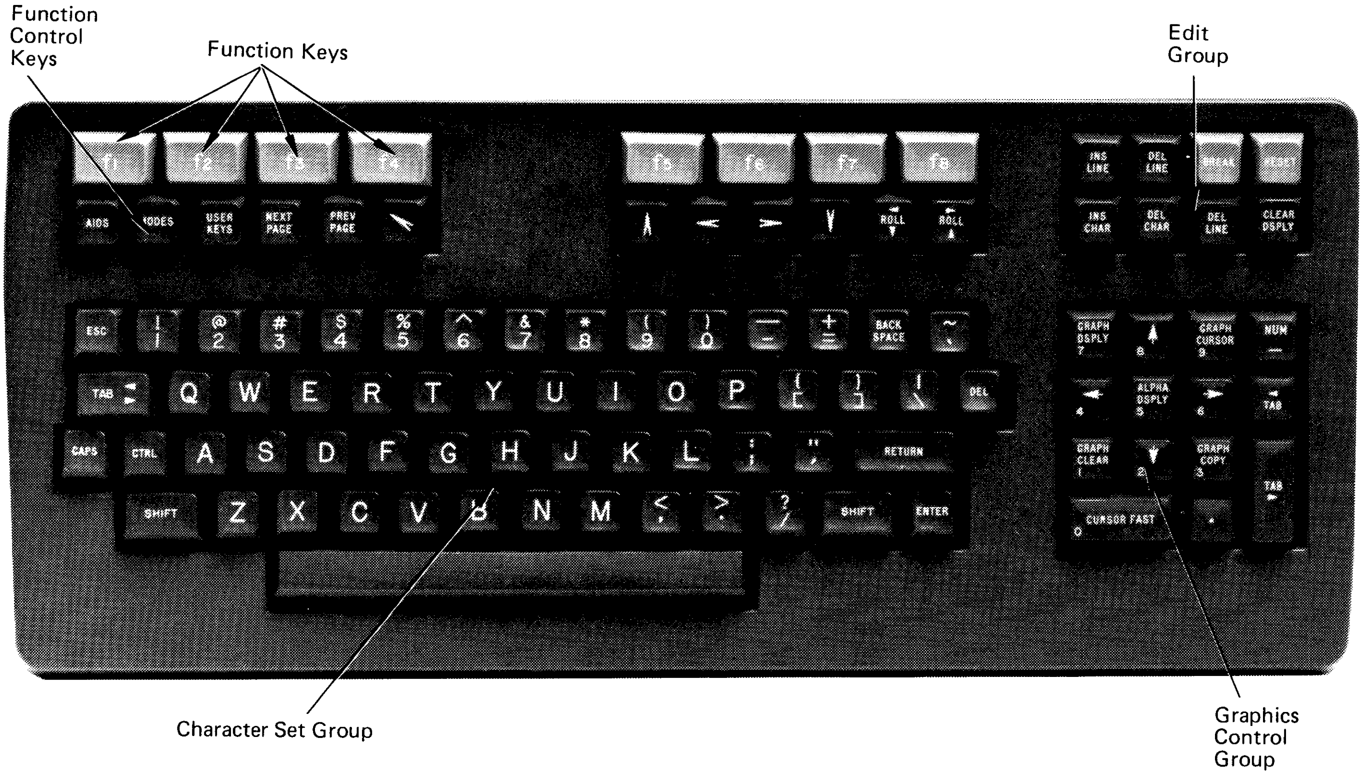


Figure 1-9. 2623A Graphics Terminal Keyboard

Supported Devices

DSG/3000 supports, directly and indirectly, a number of terminals, plotters, and plotter/printers. Table 1-1 lists them, and notes how they are addressed. More information about addressing may be found under *Selecting a Device* below.

Table 1-1. Supported Devices

Terminal Designator	HP-IB Interface	Graphics Capability	
		YES	NO
2382			X
2622			X
2623A		X	
2624A/B			X
2626A			X
2641			X
2642			X
2645			X
2647A	X	X	
2648A	X	X	
2703		X	

Laser Printer (2680A)

The Laser Printing System is treated as a hardcopy graphics output device.

(Table 1-1, continued)

Plotter and Printer Designators	Interface	
	HP-IB	RS-232
2-pen		
7470		
(opt.001)		X
(opt.002)	X	
4-pen		
7220A/S		X
7221A/B/S		X
9872A/B/S	X	
8-pen		
7220C/T		X
7221C/T		X
7580A		
(opt.001)		X
(opt.002)	X	
7858		
(opt.001)		X
(opt.002)	X	
9872C/T	X	
Thermal Plotter/ Printer		
7240A		X
7245A/B	X	
Mini plotter		
7225A/B		
(17601A)	X	
(17603A)		X
(17604A)		X

Printers such as the 7310A, 2631G, and 9876, while not directly supported by DSG/3000, may be attached to a 2648, 2647, or 2623 graphics terminal for hardcopy purposes. The integral printer of the 2623 may also be used for hardcopy production.

Selecting a Device

As stated previously, it is recommended that DSG/3000 be run on a supported graphics terminal. On these terminals, screen menus may be produced in a standard alphanumeric mode and drawn in graphics mode.

Alternatively, you can run DSG/3000 on a non-graphics terminal such as the 2626A, 2382A, or 2645A. These terminals will produce all of the screen menus but cannot draw graphs. However, graphic output can be directed to a 7220, 7221, 7225, or 7240. Also, these terminals do not support plotters requiring HP-IB interface. Table 1-2 lists valid terminal and plotter combinations of all those devices supported, directly or indirectly, by DSG/3000.

Table 1-2. Valid Terminal and Plotter Combinations

INTERFACE	TERMINAL USED	VALID PLOTTERS
RS-232 (Eavesdrop Mode)	2622,2623,2624,2626 2641,2642,2645,2647 2648,2382,2703	7220,7221,7240,7580 Opt.001 7585 Opt.001,7225A (17604A)
RS-232 (Stand-alone mode)		7220A/C/S/T,7240 ** 7221A/B/C/S/T,7225A (17604A) 7225A (17063A),7470 Opt.001 7580 Opt.001,7587 (Opt.001)
HP-IB	2647,2648	7225A (17601A),7245A/B 7580A Opt.002,7585 Opt.002

**S=scrollable paper,4 pens;T=scrollable paper,8 pens;
A=non-scrollable paper,4 pens;C=non-scrollable paper,8 pens

Repositioning and Redirecting Output

If you are running GRAPH from a terminal and want to direct the output to a hard-copy device, you can connect any one of the HP-IB plotters directly to the terminal. HB-IB (Hewlett-Packard Interface Bus) devices must be used with an HP-IB board (13296A Shared Peripheral Interface) and a raster dump ROM to connect them to a terminal. The 2647A graphics terminal is installed with the raster dump ROM. Recent models of the 2648A also include it. Older models of the 2648A with device support firmware designated HP1326A with Option 3 have it. Other older models may have the device support firmware but not the raster dump ROM. In that case, it can be ordered as a separate part (P/N 1818-0746), or with the shared peripheral interface, HP 13296A Option 48, for the retrofit.

RS-232 devices are used between the computer and the terminal in the "eavesdrop" mode and/or directly connected to the computer in a stand-alone configuration. All devices which may be used in eavesdrop mode may also be used standalone. In this case, the plotter is connected to a separate port, and graphic output is directed to it by issuing the following file equation:

```
:FILE GRAPHDEV;DEV=ldev
```

where *ldev* is the logical device number of the plotter. (The logical device number depends on the configuration of the system; your system manager should be able to supply you with this number.)

In order to direct graphics to a plotter, specify the device and HP-IB address, if applicable, on the GRAPHING OPTIONS MENU (See Section V). This menu also allows you to position the graph on a page.

For devices which are strictly HP-IB addressable (9872A/B/C/S/T and 7245A/B), GRAPH will default to using HP-IB address 5. For devices which may be either HP-IB or RS232 addressable, such as the 7225A or 7580, a valid HP-IB address must be entered if desired. Otherwise, non-HP-IB addressing is assumed.

Additional information concerning devices is provided throughout the manual as it pertains to specific operations of DSG/3000.

Formal File Designators

The following formal file designators are associated with DSG/3000:

MENUDEV designates the device at that displays the menus (GRAF program only)

GRAPHDEV designates the device at that draws the graphs

$STDLIST is the default actual file designator for both MENUDEV and GRAPHDEV. In a session, $STDLIST is the log-on terminal.

Raster Printers

If you are running GRAPH on a supported HP graphics terminal, it is possible to transfer graphic output from the terminal to a raster printer such as the HP 2631G or 7310A, using an HP-IB. The plotter HP-IB address should be set to 6.

From the 2647A, use the following special function key commands:

```
COMMAND
f1
f1
f3
f2
f4
f7
6
RETURN
```

This is equivalent to the command:

```
TRANSFER FILE FROM GRAPHICS TO HP-IB 6
```

From the 2648A terminal, press the following function keys before logging on:

```
GOLD
INSERT
CHAR
GREEN
f5
INSERT CHAR
RETURN
```

From the 2623A, you have a choice of using the integral printer or an external device.

To select the desired procedure, press the following in succession before logging on:

```
AIDS
f1
f3
```

This displays the Device Control labels, and takes you to the "to devices" selection.

Next, press one of the following:

```
f3  (to integral printer)
f2  (to external device)
```

When you wish to print, press:

```
ESC  Ø
```

If the graphics/numeric pad isn't set for graphics operation, press the [SHIFT] key and [NUM] key simultaneously.

Press:

```
graph copy
```

If you wish to copy a menu displayed on the screen, press the following:

```
AIDS
f1
```

This takes you to device control. Next:

```
home the cursor
press the [copy all] key
```

When you wish to print, press:

```
ESC  Ø
```

SECTION 2

Graphing Terms and Concepts

Purpose of Graphs

The purpose of a graph is to display data visually. A single graph can distill hundreds of lines of data in a compact image. A graph is also an analytical tool, allowing you readily to perceive and recall trends in the data, and to make quick approximations.

The information on this page, presented in its tabulated form, would require considerable study in order for you to draw conclusions from it.

MONTH	QUOTA-SALES	ACTUAL-SALES
JAN	8680	7405
FEB	8130	9356
MAR	9620	8953
APR	8009	6772
MAY	9361	7440
JUN	10153	9231
JUL	9433	8139
AUG	10231	10524
SEP	9233	7882
OCT	9026	8261
NOV	8657	7193
DEC	10568	10730

Using DSG/3000, you could plot QUOTA-SALES and ACTUAL-SALES against time in a graph, and the results would be readily apparent, as in Figure 2-1.

The graph enables you to grasp the monthly fluctuations in each of the variables. The maximum and minimum occurrences can be detected and approximated at a glance. Furthermore, the graph shows the relationship between the variables, each of which is represented by a line. For example, you can quickly spot those months in which actual sales met or exceeded quota.

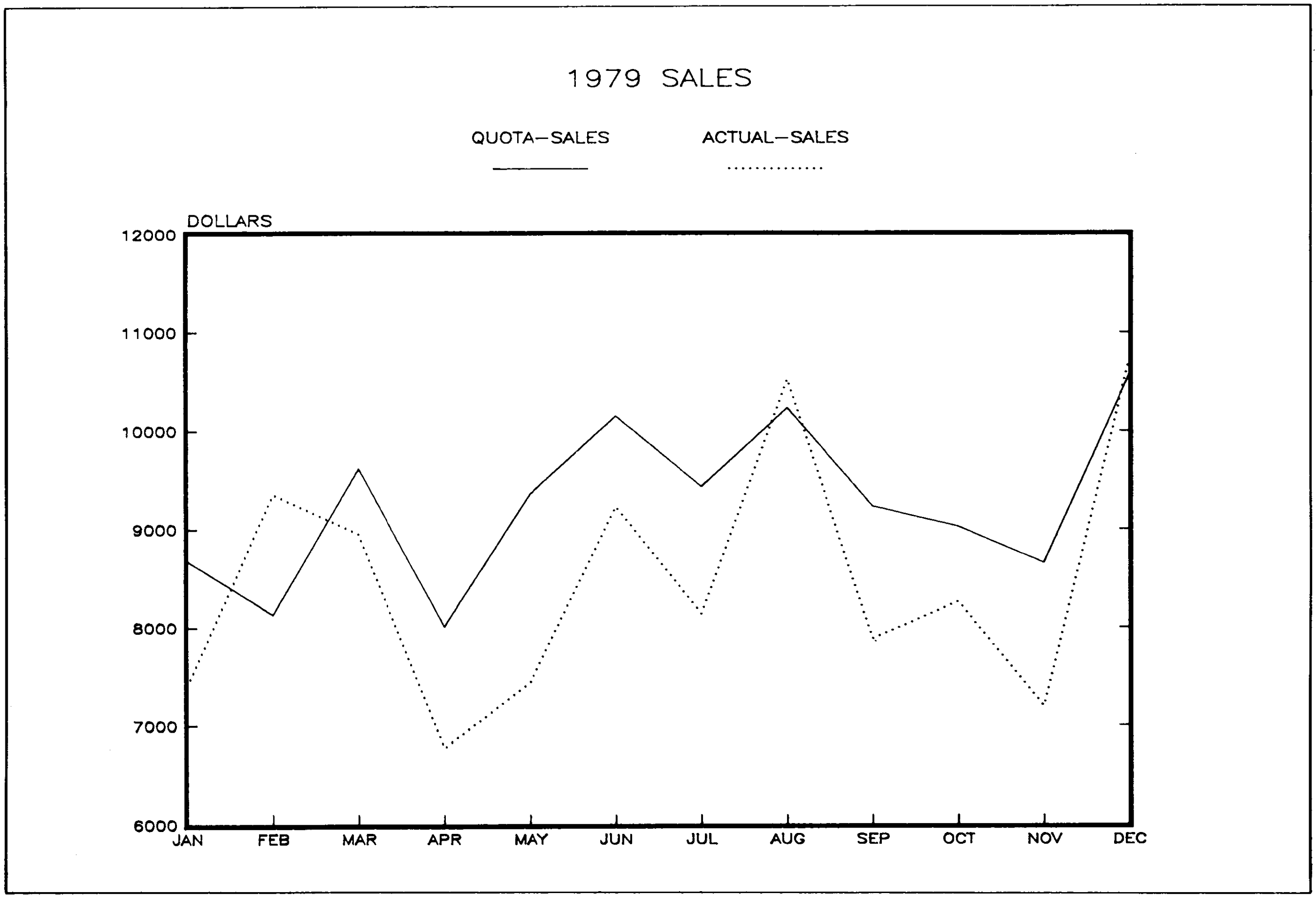


Figure 2-1. Line Graph

Types of Data and Graphs

DSG/3000 allows you to choose between 3 basic types of graphs: lines (including scattergrams), bars, and pies. The graph type you choose depends on the nature of the data and the purpose of the graph. In order to understand the capabilities of different graph types, you need to understand some basic terms and concepts.

Independent and Dependent Variables

For the graph given in Figure 2-1, MONTH is considered to be the *independent variable* because its changing value is governed independently from the rest of the data file. There can be only one independent variable per graph, and often it is a time variable. The independent variable is plotted on the X-axis of a line graph or bar graph.

On the other hand, SALES-QUOTA and SALES-ACTUAL take on values that change with time. They are considered *dependent variables* because they depend on the variable MONTH. Dependent variables are plotted along the Y-axis of a line graph. There is one line for each dependent variable.

Bar graphs have one bar for each dependent or Y-axis variable, stacked or clustered around each data point. For example, the bar graph in Figure 2-2 has two dependent variables which are drawn as clustered bars at each of the 12 data points.

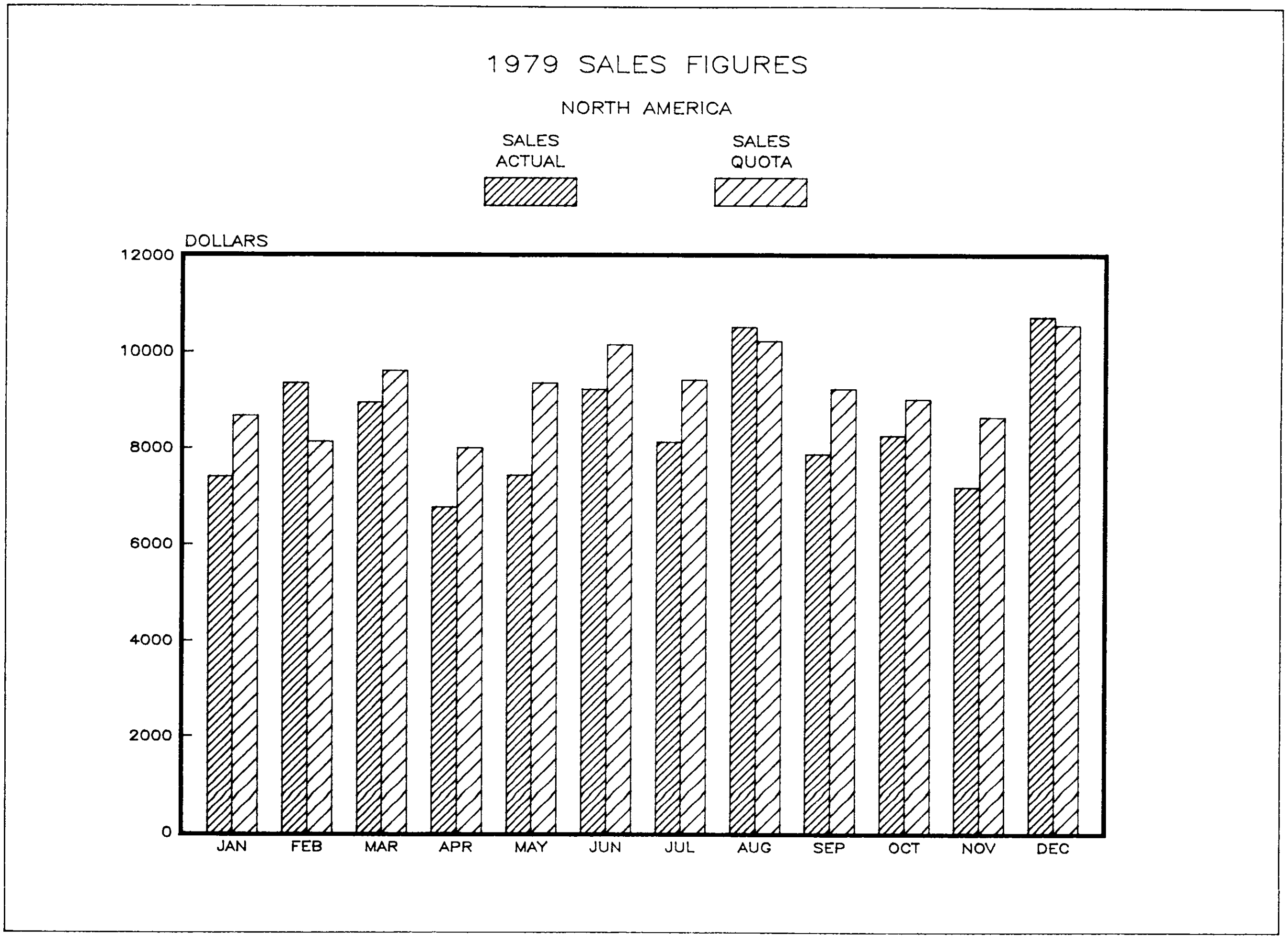


Figure 2-2. Bar Graph

Continuous and Discrete Data

In Figures 2-1 and 2-2, there is exactly one data point for each month. The graph is drawn by plotting individual points. Then, if you are drawing a line chart, connecting lines are drawn. If you are drawing a bar chart, bars are drawn up to the points. Note that, in bar graphs, you cannot easily estimate the value of sales at mid-month, for example. This data is considered to be *discrete* because each data point is a separate entity. Other common cases of discrete data occur when the independent variable uses text that does not represent time, as in Figure 2-3.

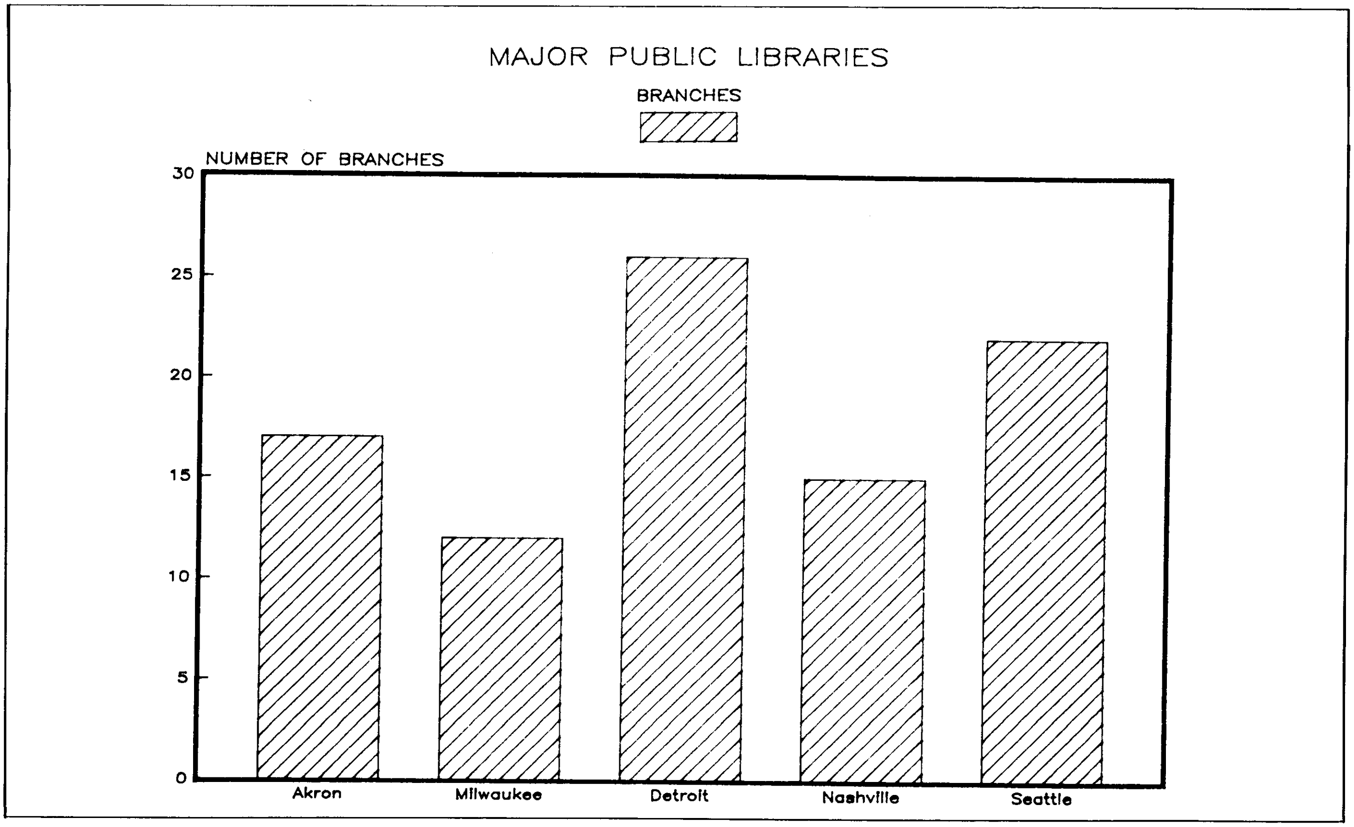


Figure 2-3. Discrete Data

In other cases, the nature of the data does allow you to read between data points. For example, suppose you graphed the total actual sales-to-date for the twelve months given previously. In this case, the data is *continuous* because it forms an uninterrupted series of points. Although there still may be only one data point for each month, the value of SALES-TO-DATE can be estimated or *interpolated* for dates between the data points.

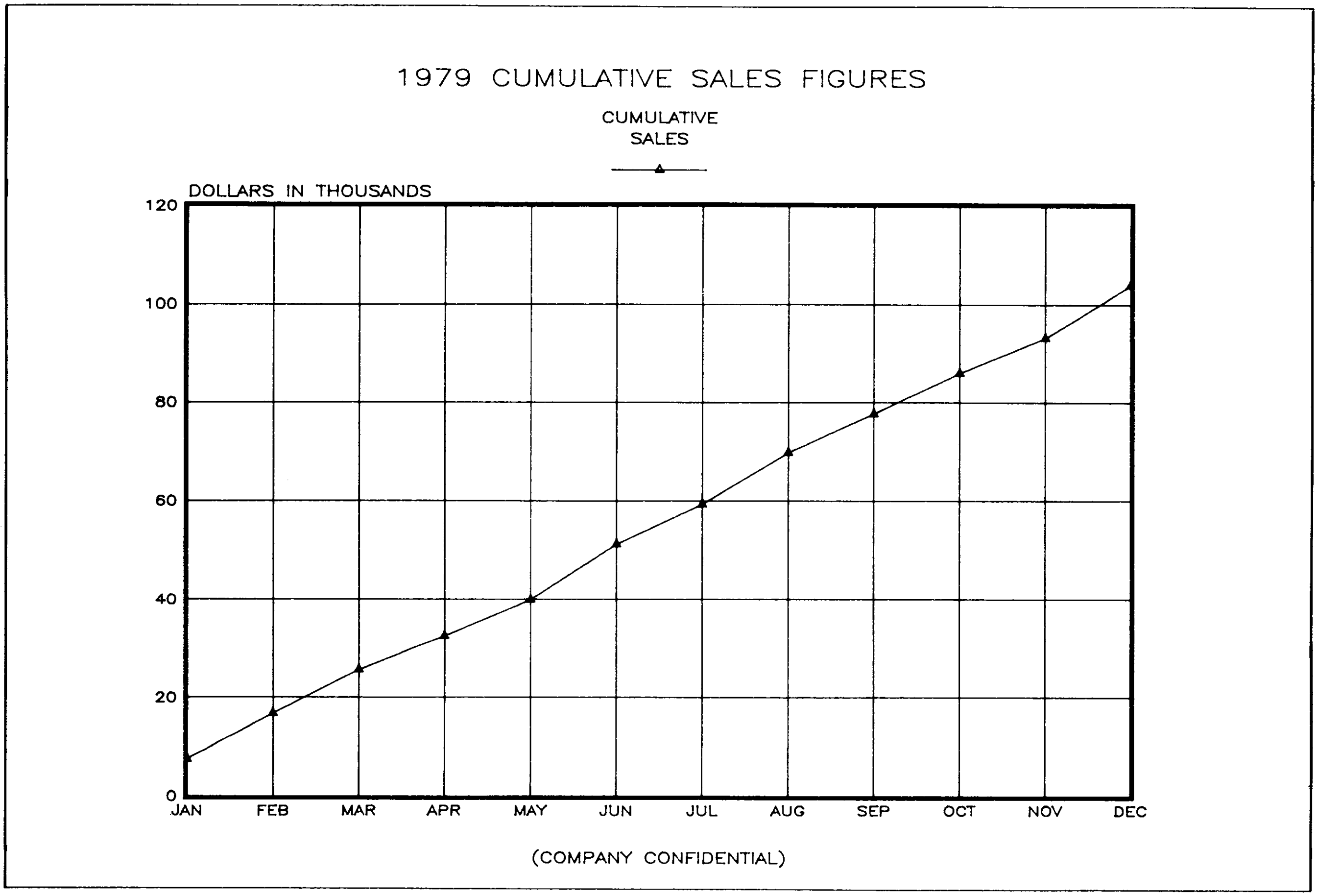


Figure 2-4. Continuous Data

Choosing A Graph Type

If you want to create a graph that shows how a variable changes over time, choose either a line or bar graph. Line graphs are best for showing trends in the data, especially when the values are continuous.

Bar graphs always show data as discrete values. They are useful for showing relative weights of variables because the data values are represented by physical blocks. Bar graphs are often used to show growth.

Showing Proportions

If you want to show the relative distribution of a variable, use a pie chart. While bar and line graphs involve at least two variables, pie charts graph one variable at a time as a percent of the total.

A pie chart shows an entity broken down into its component parts. Each part is represented by a proportionate sector (slice) of the pie. For example, Figure 2-5 shows the distribution of orders by product line:

This pie chart gives a clear determination of the products that had the greatest proportion of total sales.

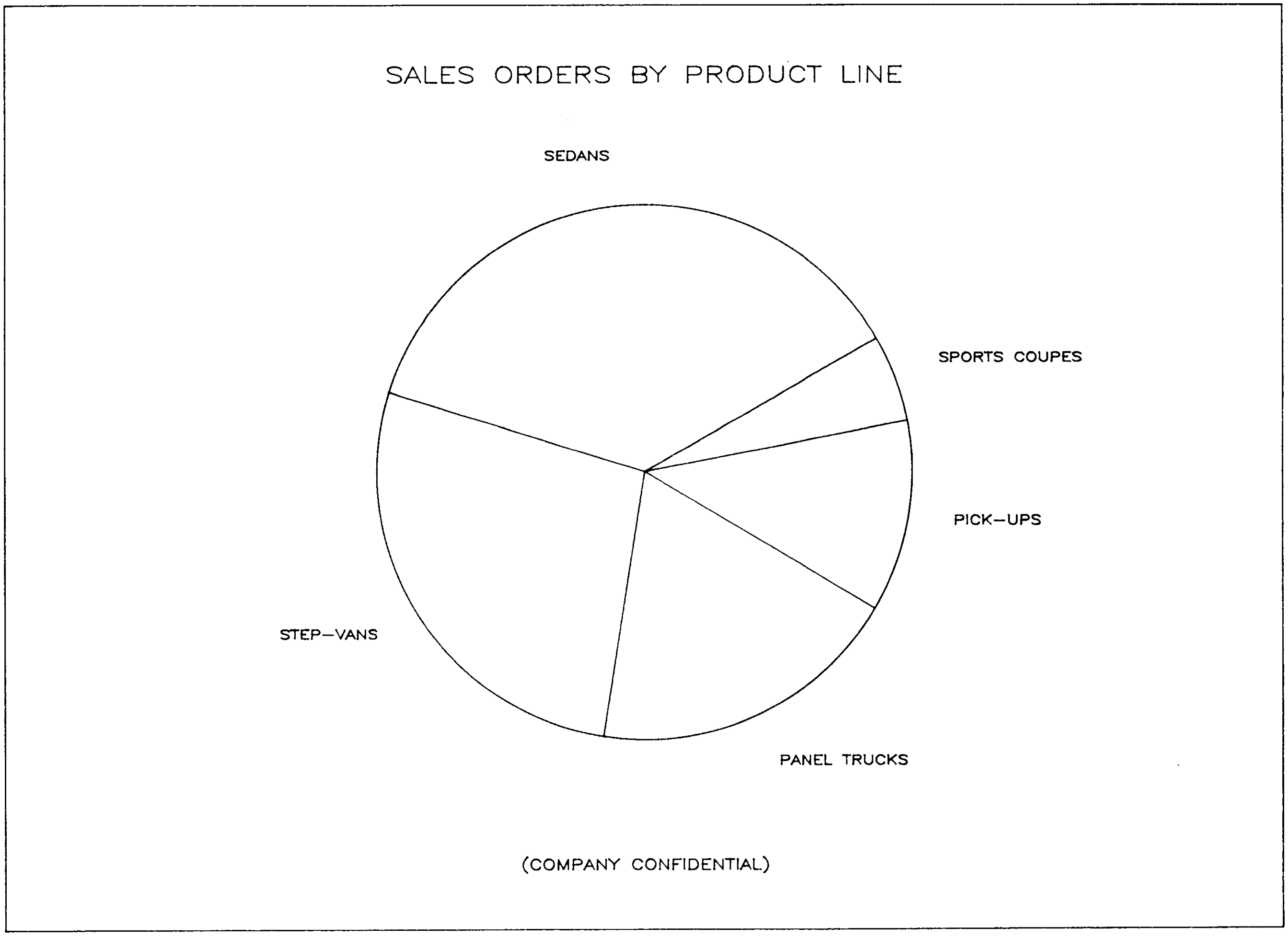


Figure 2-5. Pie Graph Showing Proportion

Showing Correlation

If you want to show how variables in the data file correlate with one another, use a scattergram. You can think of a scattergram as a special type of line graph in which data point markers are plotted, but connecting lines are not drawn. Unlike a line graph, however, what is important in a scattergram is the clustering of data points rather than the direction taken from one point to another.

You can use a scattergram when you want to explore your data for possible relationships between the variables, and when a simple functional relationship is not evident. Suppose that two of the variables in your data file are HEIGHT and WEIGHT. If there is an orderly relationship between these two variables, there will be a pattern to the distribution of points in the scattergram. For example, the graph in Figure 2-6 suggests a roughly linear relationship between the variables.

After looking at the scattergram, you can try to approximate the relationship between the variables and predict its course.

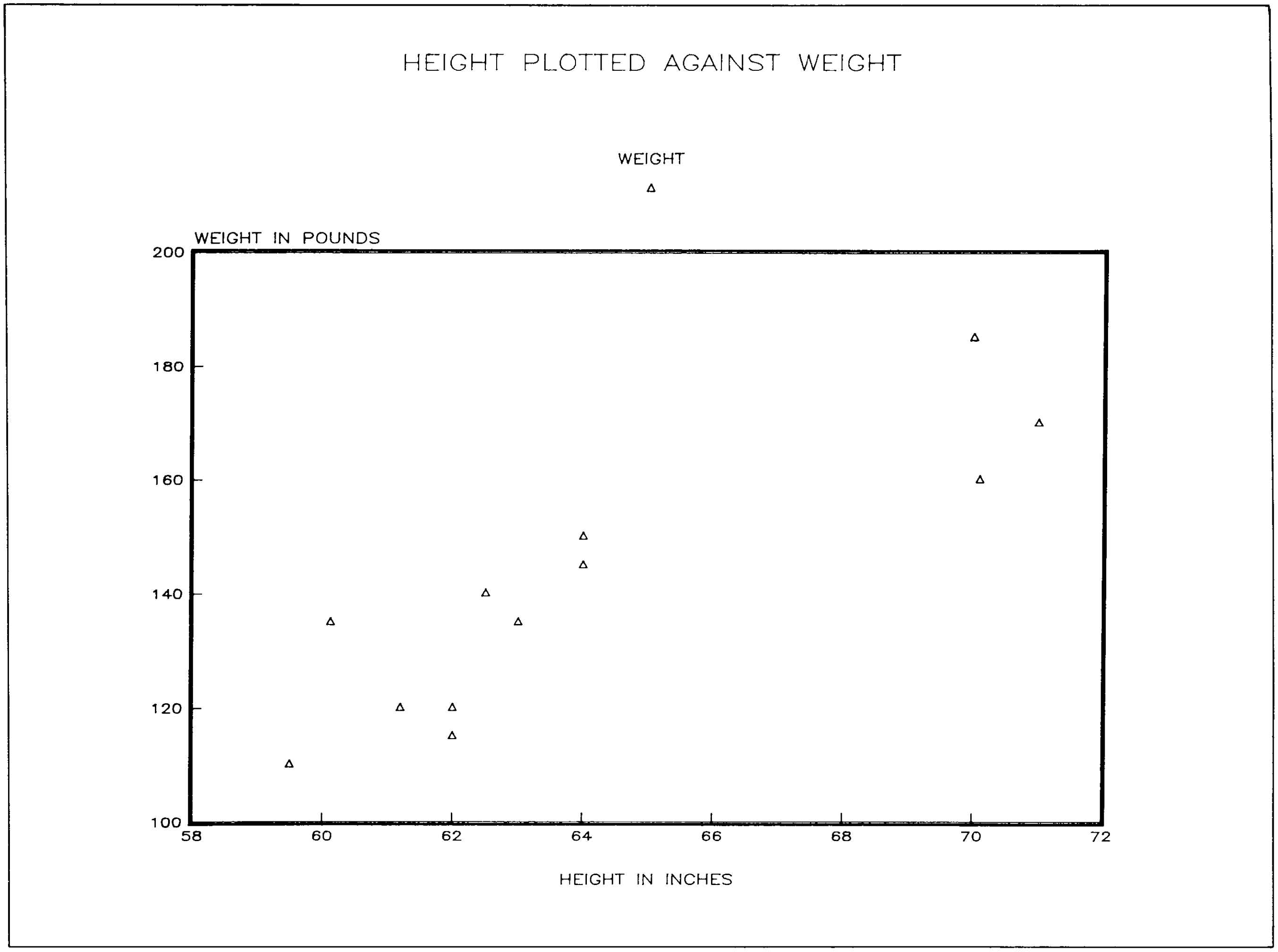


Figure 2-6. Scattergram Showing Correlation

Further Terms and Enhancements

Once you have chosen the type of graph that fits your needs, DSG/3000 allows you to specify many of the graph's characteristics. The examples on the pages following illustrate several of the options and enhancements that DSG/3000 makes available for your use.

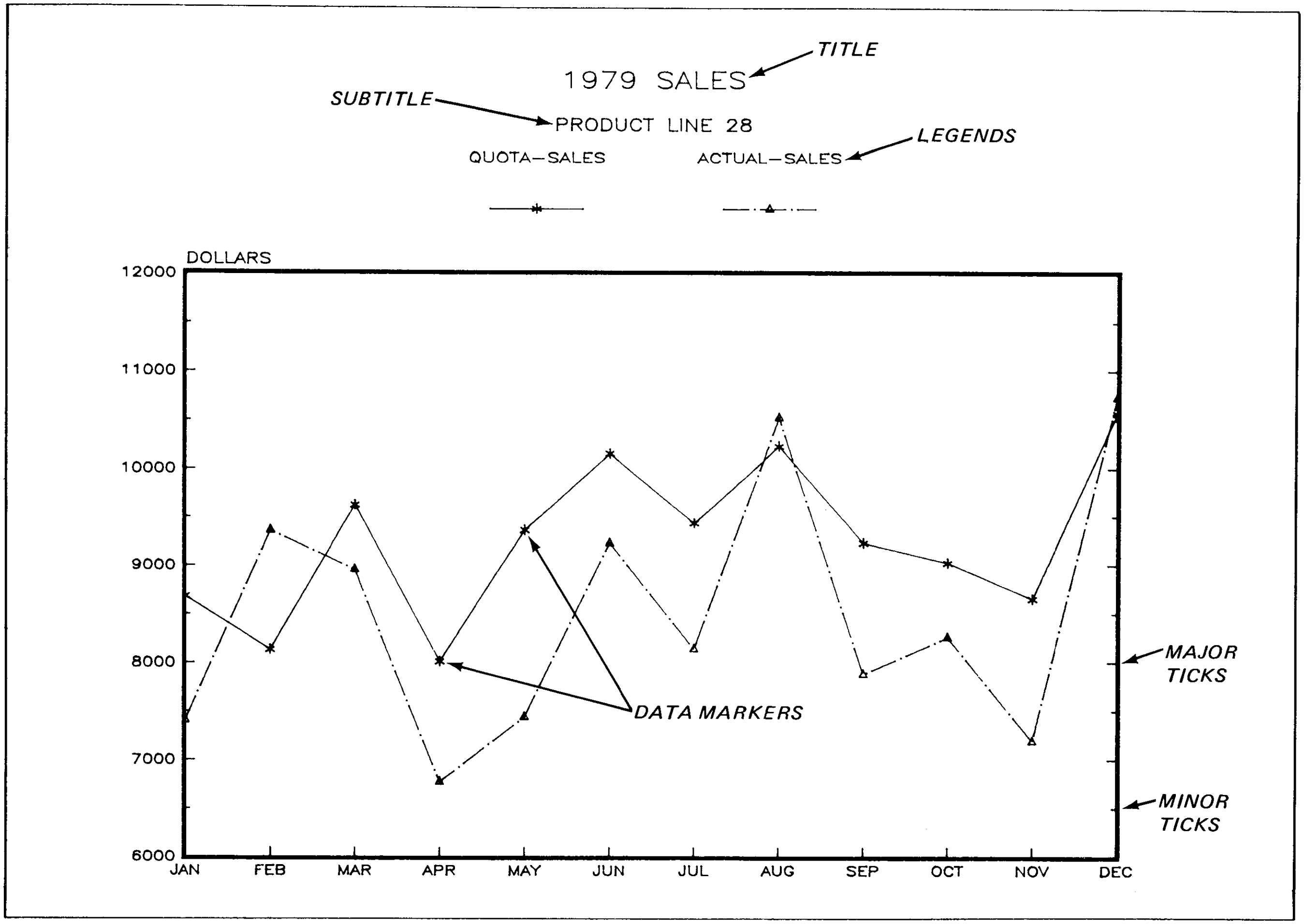


Figure 2-7. Line Graph

DSG/3000 provides eight different symbols for data markers, the points marking the actual data values. There are also eight different line texture types that are identified with legends. The scaling marks along the axes are called ticks. They come in large and small sizes, called major and minor ticks respectively.

If your data ranges over very widespread values, it may be best to use logarithmic scaling rather than linear scaling. The axes are scaled exponentially.

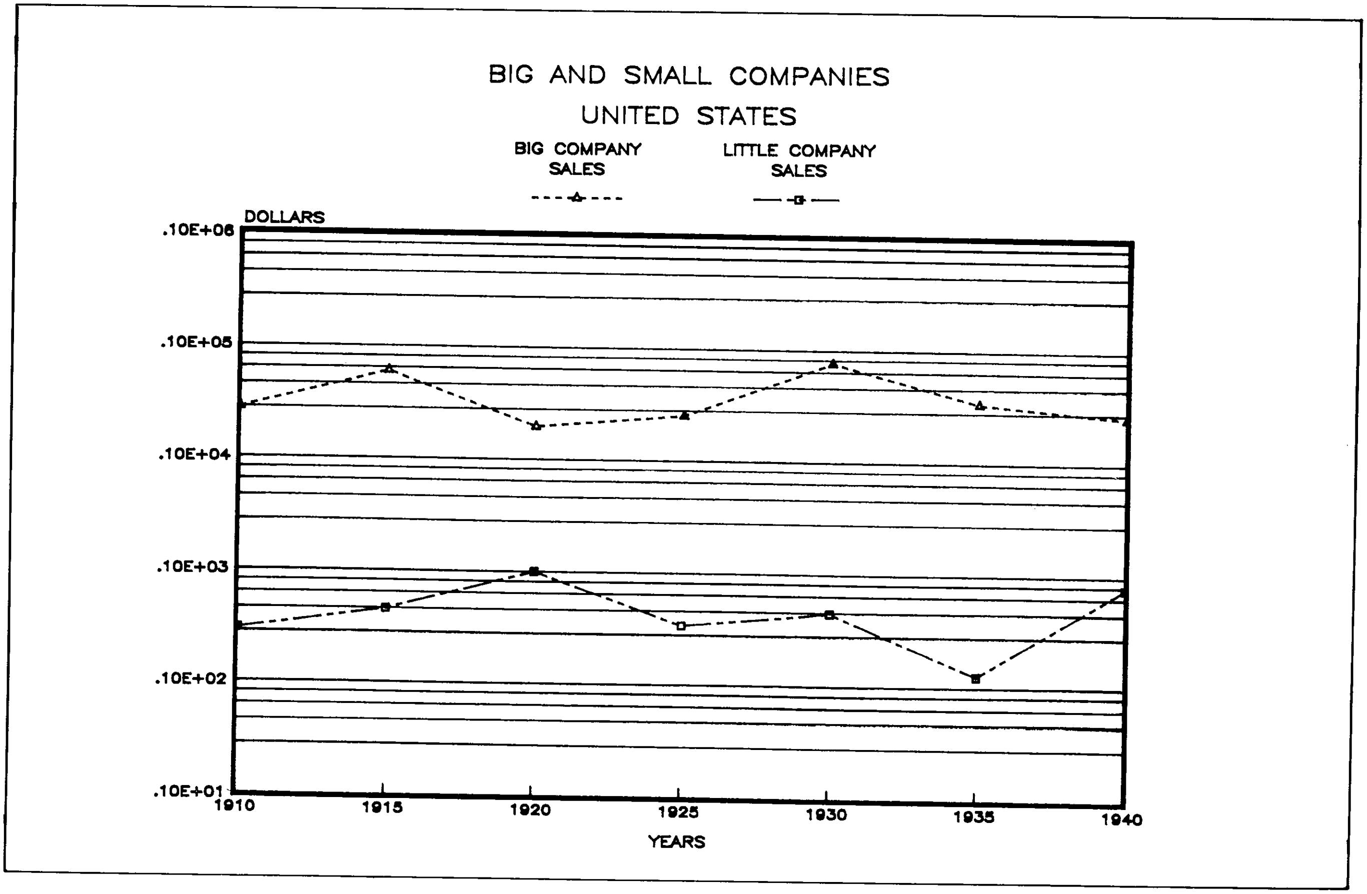


Figure 2-8. Line Graph with Logarithmic Scaling

A scattergram is like a line graph without lines drawn between the data markers.

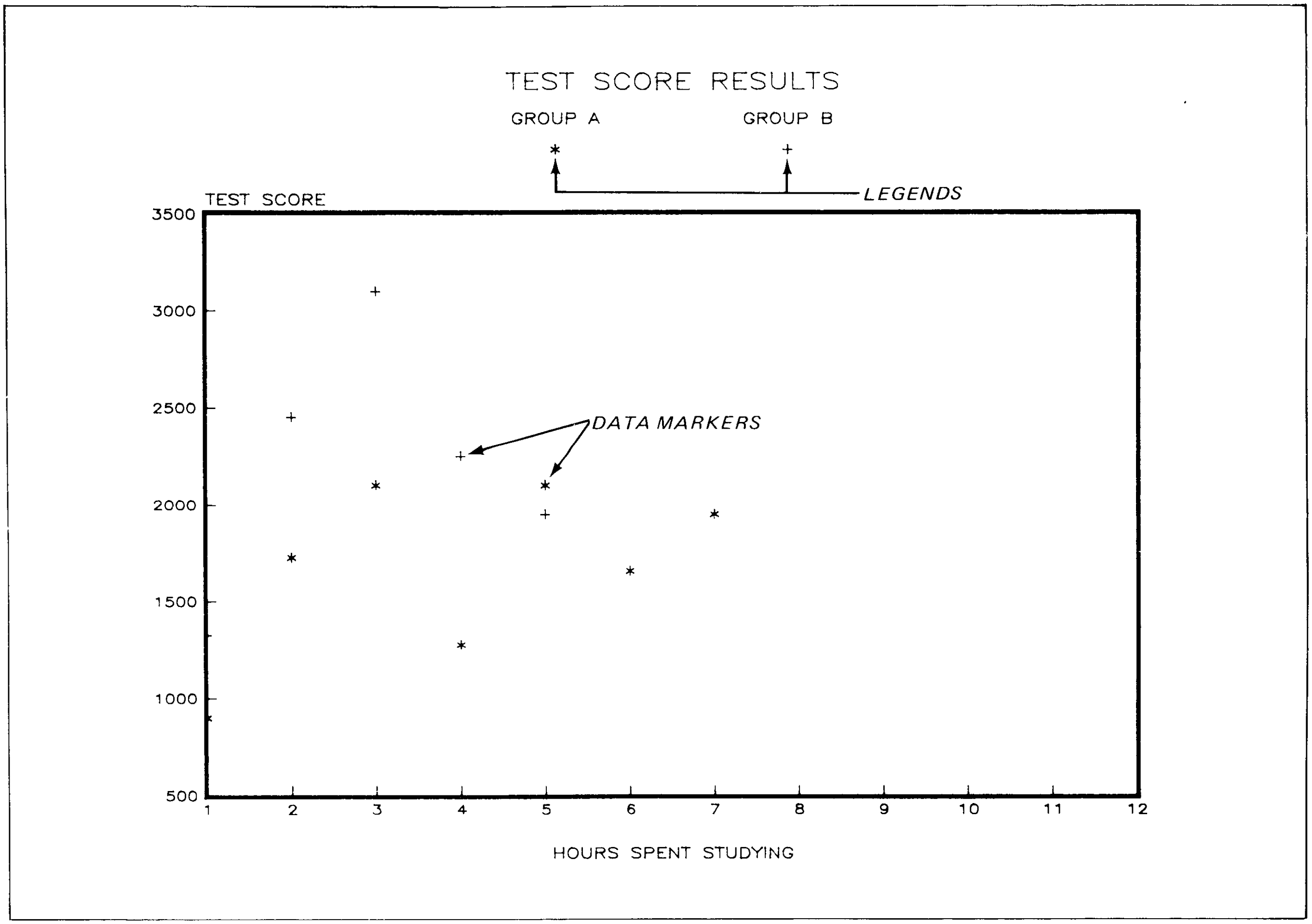


Figure 2-9. Scattergram

Four enhanced bar charts are shown on the next few pages.

There are eight different surface textures that can be used to draw a bar chart. As in line charts, a legend identifies the texture. Grid lines can make it easier to read the data values from the graph, as shown in Figure 2-10.

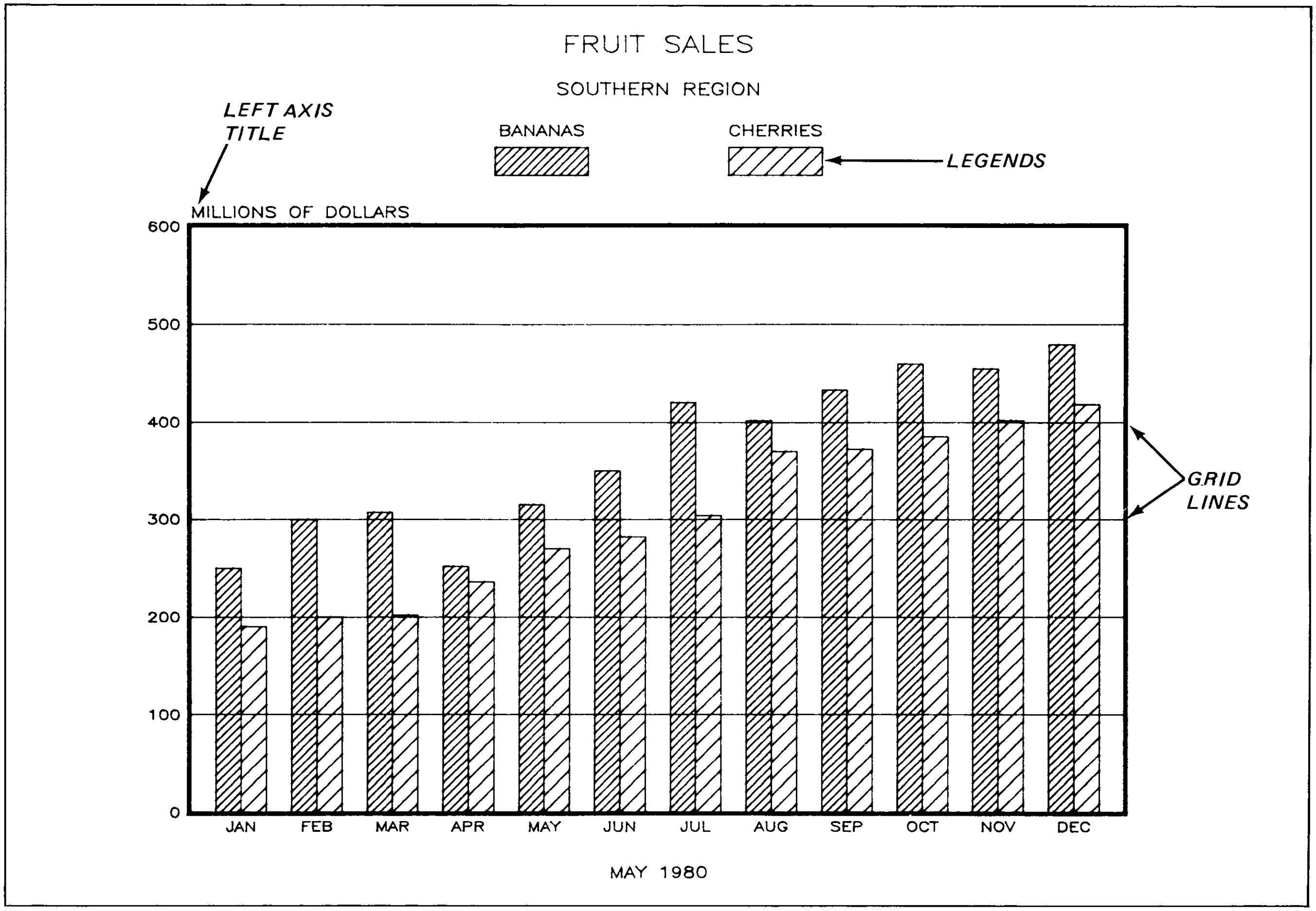


Figure 2-10. Vertical Clustered Bar Graph

Clustered bar means that each data value is represented by a single bar, and that multiple bar variables are clustered around each independent data point.

Clustered bar graphs are useful for making quantitative comparisons, whereas in the following example, stacked bars emphasize the totals at each point.

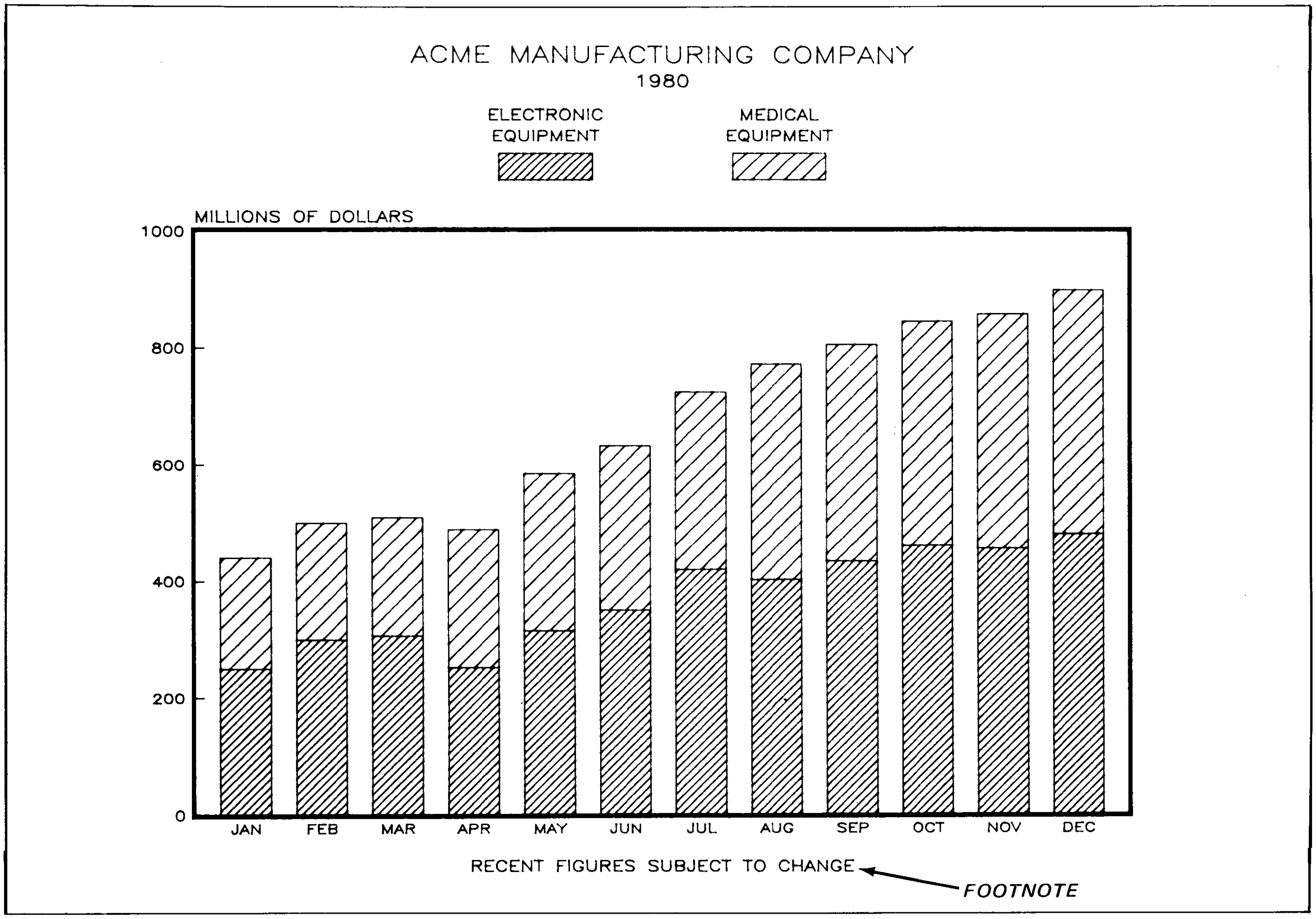


Figure 2-11. Vertical Stacked Bar Graph

When the independent or X-axis variable is not time, it is preferable to draw the bars horizontally, as illustrated in Figure 2-12 and 2-13. In a horizontal chart, the independent or X-axis appears on the left side of the graph and the dependent or Y-axis appears along the bottom of the graph, as though a vertical graph had been rotated 270 degrees.

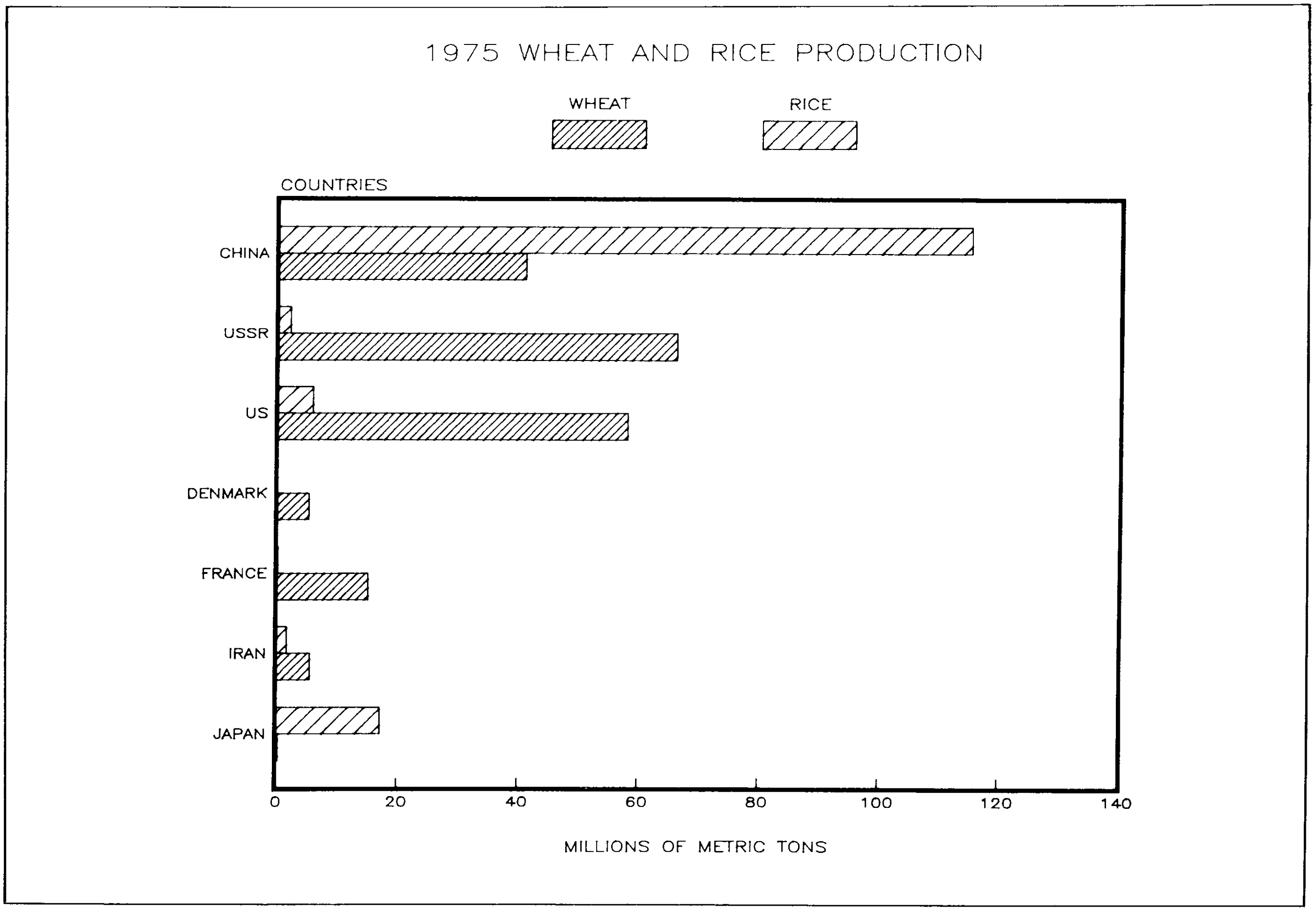


Figure 2-12. Horizontal Clustered Bar Graph

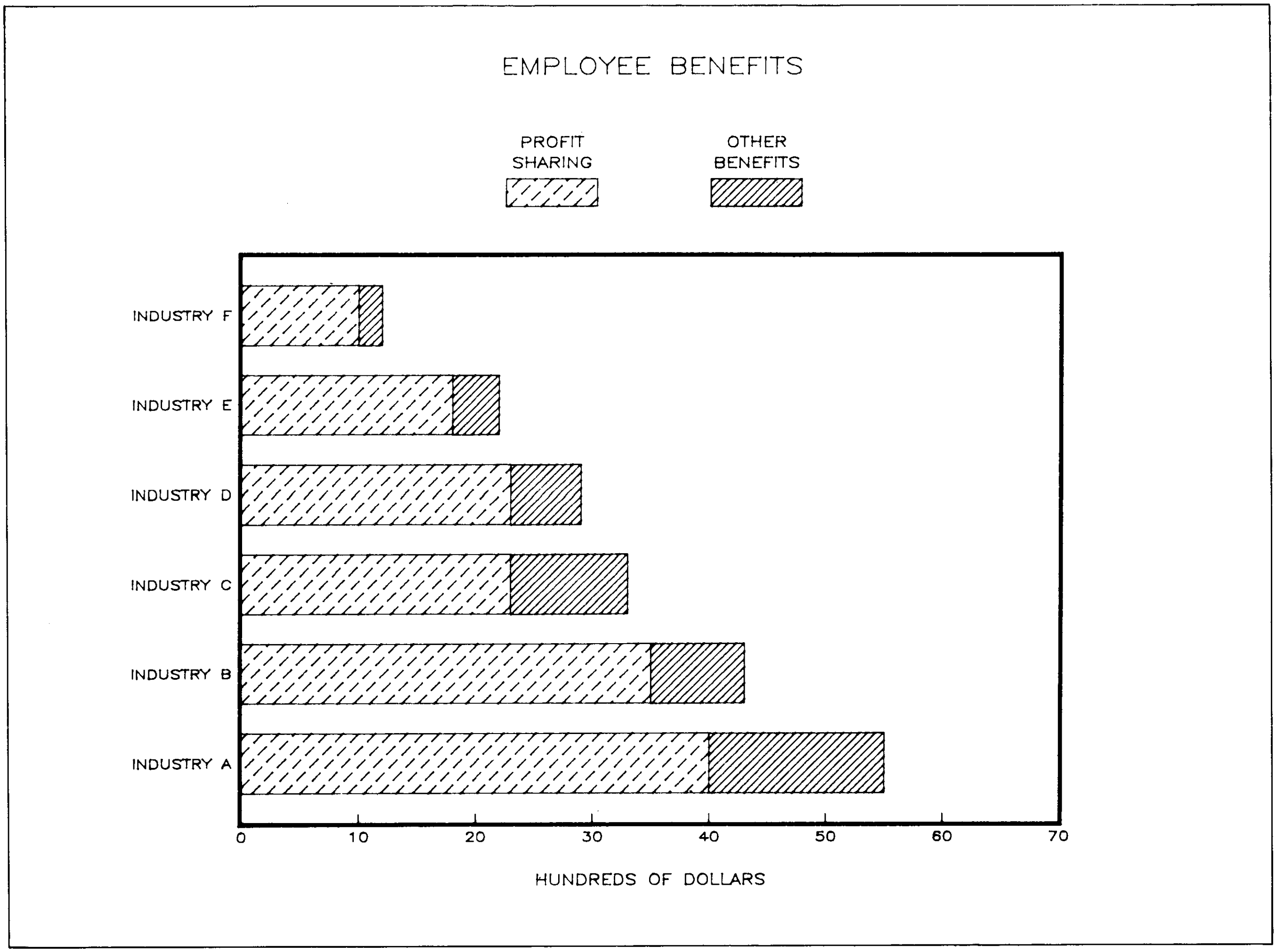


Figure 2-13. Horizontal Stacked Bar Graph

You can draw attention to one or more segments of a pie chart by exploding it--that is, drawing it apart from the rest of the graph.

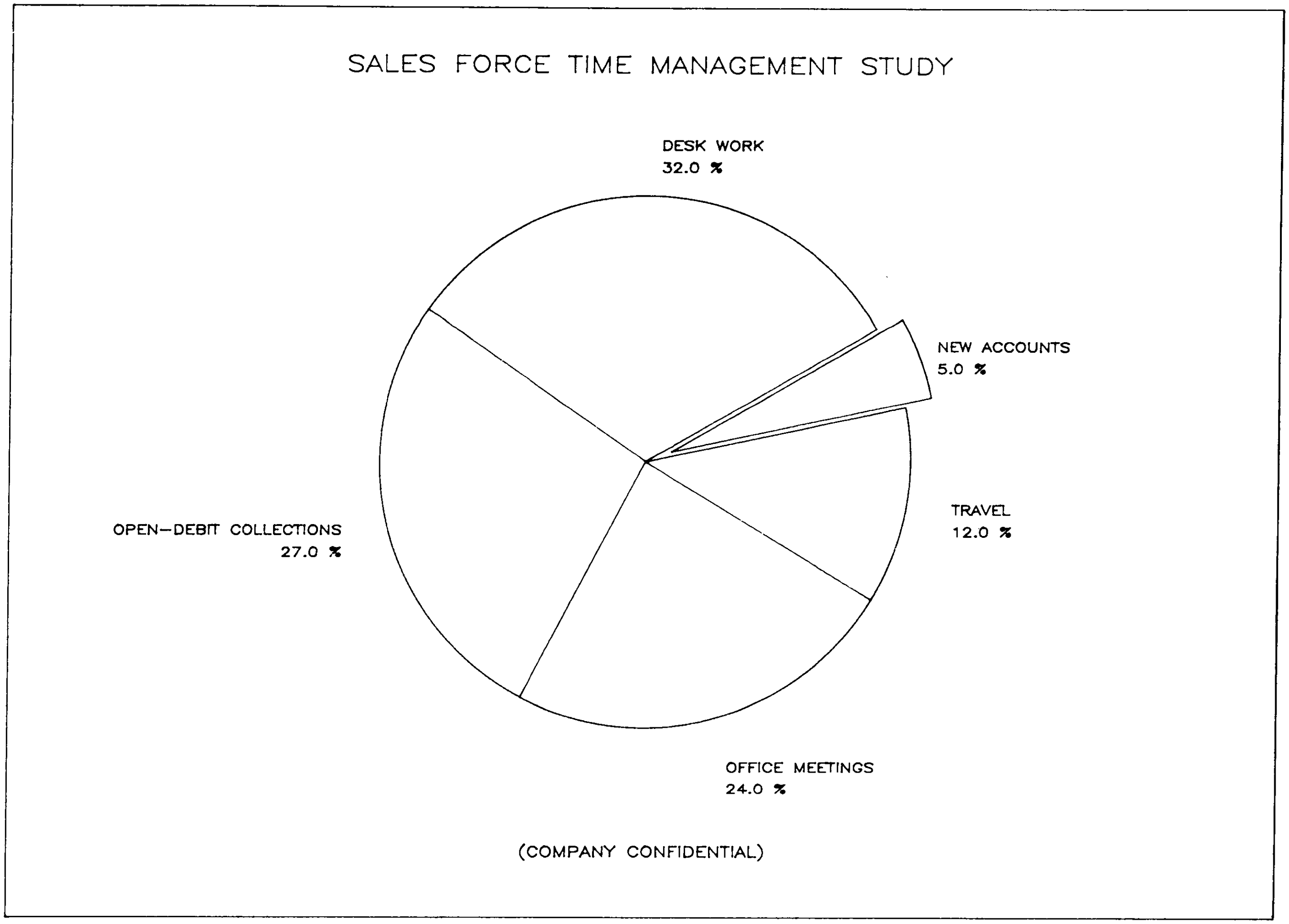


Figure 2-14. Pie Graph with Exploded Segment

SECTION 3

Creating a Data File

Before you specify chart characteristics and draw the graph, you should decide how you want DSG/3000 to access your data. The six methods of creating a data file are summarized in Table 3-1.

Since your data is kept in a file separate from the chart design specifications, you can modify or add values to the variables in the data file without changing the chart design.

METHOD	DATA ENTRY	DATA MODIFICATION	RESTRICTIONS
DATA PROMPT Screen Menu	Interactive data entry and definition. User enters data on screen menu from within the GRAPH program.	Modify data directly on screen menu.	Data file contains up to 5 variables; each variable can take up to 12 values.
EDIT/3000	Data entered into an EDITOR file is used as a DSG/3000 data file.	Modify data using the EDITOR.	Must be formatted according to DSG/3000 specifications.
TDP/3000	Data entered into a TDP file is used as a DSG/3000 data file.	Modify data using TDP.	Must be formatted according to DSG/3000 specifications.
QUERY/3000	Reports data values from an IMAGE data base to a data file or self-describing file.	Modify data base using calls to IMAGE. Run QUERY to output new values.	Data must reside in an IMAGE data base. Output file must be formatted according to DSG/3000 specifications.
VPLUS/3000	VPLUS/3000 batch file is accessed as a fixed-format data file.	Run ENTRY program in BROWSE mode to modify data.	Determine offset and length of each data field, using VPLUS/3000 formsfile.
User Program	Periodic or report generating program creates DSG/3000 data file.	Data is modified by the program.	Output file must be formatted according to DSG/3000 specifications.

Table 3-1. Summary of DSG/3000 Data Entry Methods

Interactive Data Entry

If you are using DSG/3000 interactively and your data consists of a limited number of variables and data values, the easiest method for entering the data is through the DATA PROMPT screen menu. When you use this method, DSG/3000 builds the data file for you in the proper format and keeps a definition of the data internally. An illustration of this method is given below in Figure 3-4.

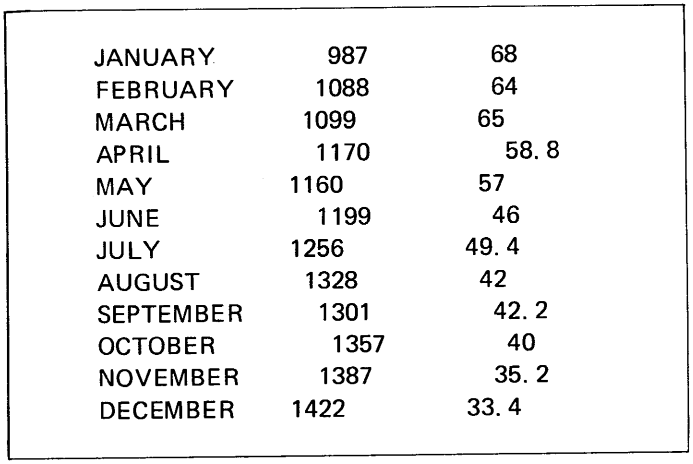
If you choose to use this method, it is not necessary to read the portions of this section on using other programs and subsystems to enter the data. You can refer to Figure 3-4, read the paragraphs on Structuring the Data File, and then proceed to Section IV, *Using DSG/3000 Interactively.*

Data File Format

A DSG/3000 data file is an MPE file containing ASCII representation of data values. The maximum record length allowed is 512 bytes. The data values can be either numeric or textual, and numeric values can be signed (i.e., preceded by a + or -) or unsigned. The data file is restricted to a single record type, which means that the arrangement of fields must be the same in all of the records. Each data variable can be associated with a particular field. A maximum of nine data variables from a data file may be used on any one graph.

Note: Maximum data value = 1.OE77
      Minimum data value = 1.OE-77

For example, Figure 3-1 is an example of a DSG/3000 data file with three fields.



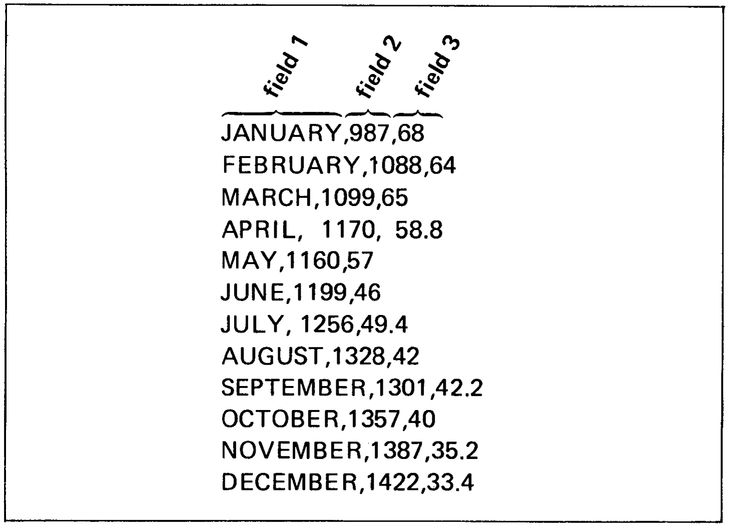
```
JANUARY      987       68
FEBRUARY    1088       64
MARCH       1099      65
APRIL        1170        58.8
MAY        1160       57
JUNE         1199       46
JULY       1256      49.4
AUGUST      1328      42
SEPTEMBER    1301       42.2
OCTOBER       1357        40
NOVEMBER     1387       35.2
DECEMBER   1422      33.4
```

Figure 3-1. DSG/3000 Data File

The first field of this data file might be associated with the variable MONTH, the second field with SALES, and the third field with UNIT-PRICE. Note that the data file does not contain any labels to identify the fields, only data values.

The data file can be in either free or fixed format. A *free format* data file is arranged in fields separated by commas or by any number of blanks. Any data value containing embedded blanks must be enclosed in quotes so that the blank is not mistaken for a delimiter.

Free format is a flexible format in that the fields do not have to be justified, although the variables must appear in the same order in every record. For any record, the first field starts with the first printable character and extends up to the first blank or comma; the second field starts with the next printable character and extends up to next blank or comma, and so forth. The data file in Figure 3-1 is in free format. Figure 3-2 illustrates another way of free-formatting the same data.

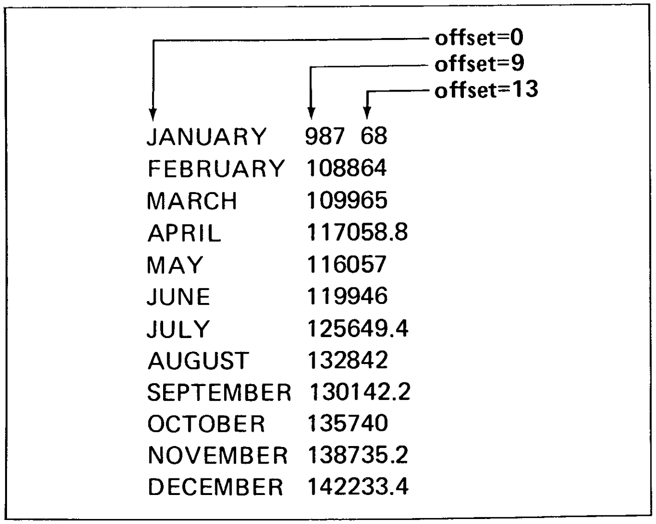

field 1
field 2
field 3

```
JANUARY,987,68
FEBRUARY,1088,64
MARCH,1099,65
APRIL, 1170, 58.8
MAY,1160,57
JUNE,1199,46
JULY, 1256,49.4
AUGUST,1328,42
SEPTEMBER,1301,42.2
OCTOBER,1357,40
NOVEMBER,1387,35.2
DECEMBER,1422,33.4
```

Figure 3-2. Free Format Data File

In a *fixed format* file, a particular field must start in the same column and have the same length in every record. The *offset* of a data field is the number of characters from the beginning of the record to the beginning of the data field. Although each field has a fixed beginning and end column, the placement of the data within each field is unrestricted.

Figure 3-3 shows the data file from Figure 3-1 in fixed format. The first field is 9 characters long with an offset of 0; the second field is 4 characters long with an offset of 9; and the third field is 4 characters long with an offset of 13.



```
JANUARY  987 68
FEBRUARY 108864
MARCH    109965
APRIL    117058.8
MAY      116057
JUNE     119946
JULY     125649.4
AUGUST   132842
SEPTEMBER 130142.2
OCTOBER  135740
NOVEMBER 138735.2
DECEMBER 142233.4
```

Figure 3-3. Fixed Format Data File

Note that VPLUS/3000 generates fixed format MPE files. Therefore, if you have used VPLUS/3000 for data entry, the resulting batch file can be used as input to DSG/3000.

You cannot mix free format and fixed format data in the same DSG/3000 data file.

Filename and Security

Since the data file is an MPE file, standard MPE conventions are in effect for naming the file and establishing security. A filename consists of up to 8 alphanumeric characters beginning with a letter, and an optional group and account name. You can use passwords and lockwords to provide security, as documented in the *MPE Commands and Reference Manual*.

Data Entry Procedures and Examples

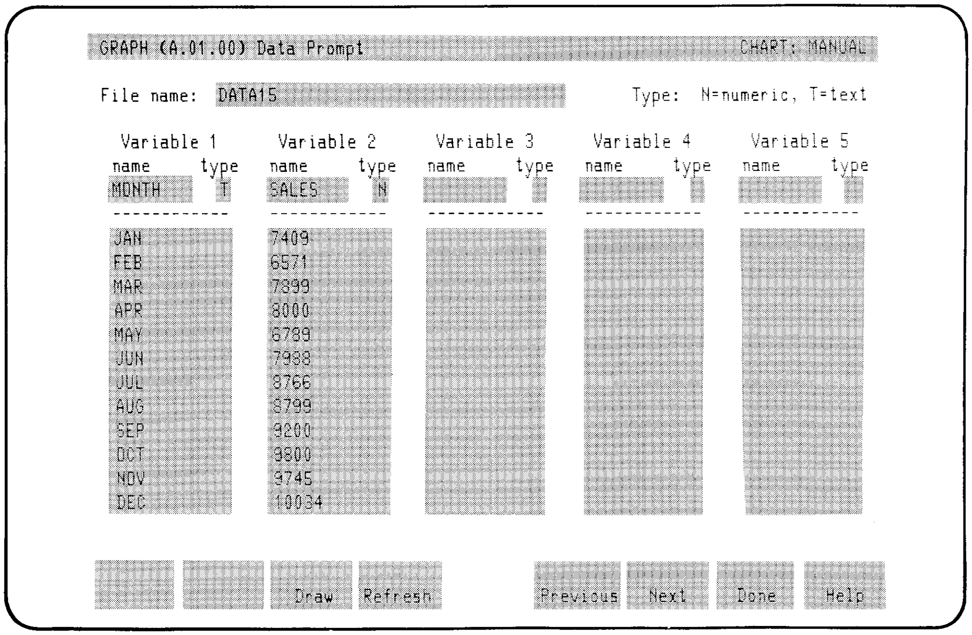
The six methods of entering data in a DSG/3000 data file are illustrated in the following examples:

Method 1 -- DATA PROMPT Menu

To use this method, you must run the GRAPH program interactively. After you enter GRAPH and choose to build a data file, the DATA PROMPT menu appears on the terminal. Note that this method allows for up to 5 variables taking up to 12 data values each. If you need to graph more than 5 variables or more than 12 data values, you must use another method for data entry.

In this example, the user filled in the filename DATA15 and information about the two variables. MONTH is a textual data variable (T); SALES is a numeric data variable (N). The user then entered all of the values of the two variables.

Section IV, *Using DSG/3000 Interactively*, gives the procedures for entering the GRAPH program. Section V gives complete specifications for this menu.

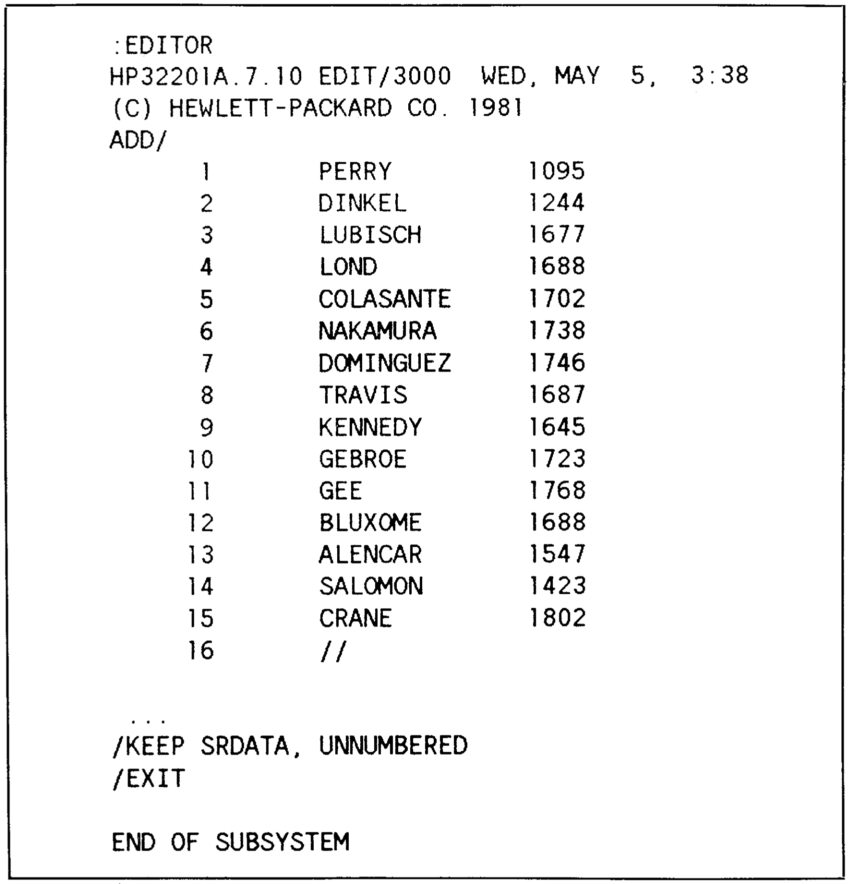


```
GRAPH (A.01.00) Data Prompt                                   CHART: MANUAL

File name: DATA15                                   Type:  N=numeric, T=text

 Variable 1        Variable 2        Variable 3        Variable 4        Variable 5
name     type     name     type     name     type     name     type     name     type
MONTH     T       SALES     N
------------      ------------      ------------      ------------      ------------
JAN               7409
FEB               6571
MAR               7899
APR               8000
MAY               6789
JUN               7988
JUL               8766
AUG               8799
SEP               9200
OCT               9800
NOV               9745
DEC               10034
```

Figure 3-4. DATA PROMPT Example

Method II-- EDIT/3000

Using the EDITOR to create a data file is an easy method. There are no restrictions on the number of variables or data values, other than the restriction that only nine variables may be used on any single graph and record size may not exceed 512 bytes.



```
:EDITOR
HP32201A.7.10 EDIT/3000  WED, MAY  5,  3:38
(C) HEWLETT-PACKARD CO. 1981
ADD/
      1       PERRY         1095
      2       DINKEL        1244
      3       LUBISCH       1677
      4       LOND          1688
      5       COLASANTE     1702
      6       NAKAMURA      1738
      7       DOMINGUEZ     1746
      8       TRAVIS        1687
      9       KENNEDY       1645
     10       GEBROE        1723
     11       GEE           1768
     12       BLUXOME       1688
     13       ALENCAR       1547
     14       SALOMON       1423
     15       CRANE         1802
     16       //

 ...
/KEEP SRDATA, UNNUMBERED
/EXIT

END OF SUBSYSTEM
```

Figure 3-5. EDITOR-Entered Data File

In Figure 3-5, the resulting data file contains two columns which represent two variables: NAME and TOTALSALES. Each of the variables takes 15 data values.

The data file, SRDATA, was kept UNNUMBERED so that the line numbers would not be appended to the data records. (If the file was kept numbered, the line numbers in columns 75 through 80 would count as an additional data field.)

Refer to the *EDIT/3000 REFERENCE MANUAL* for details about how to used the EDITOR.

Method III -- TDP/3000-Entered Data File

The TEXT AND DOCUMENT PROCESSOR, TDP, operates similarly to EDIT/3000 in many ways. The data file in Figure 3-5 could be created by TDP with few changes. In place of the command :EDITOR, you enter :RUN TDP.PUB.SYS to begin. A TDP banner will appear, and you can then ADD, TEXT, and KEEP UNNumbered your data file.

Refer to the *TEXT AND DOCUMENT PROCESSOR/3000 MANUAL MANUAL* for details about the use of TDP/3000.

Method IV -- QUERY/3000

QUERY/3000 allows you to access a data base interactively. In order to create a data file for DSG/3000 using QUERY, follow these steps:

1. Build the data file using the MPE BUILD command. The file must be a fixed length ASCII file with a record size of no more than 512 bytes. (Since QUERY output files have longer record lengths by default, you must first build the file to specification.) For example:

   ```
   :BUILD FINANCE;REC=-80,,,ASCII
   ```

   In this example, the name of the data file is FINANCE, the record length is 80 bytes, and the records are ASCII coded. Disc files are fixed length by default.

   The BUILD command has other optional parameters. For complete specifications, see the *MPE Commands Reference Manual*.

2. Equate QSLIST (QUERY output file) to the disc file built in step 1. For example:

   ```
   :FILE QSLIST=FINANCE,OLD;DEV=DISC
   ```

   You must specify the keyword OLD because the system attempts to build a new file by default.

3. Enter the QUERY subsystem. You can do this through the Command Interpreter by issuing the command:

   ```
   :RUN QUERY.PUB.SYS
   ```

   Alternatively, you can run GRAPH. Then, by entering a Q on the MAIN CONTROL Menu, you can temporarily leave GRAPH, use QUERY to enter data into the data file, and automatically return to GRAPH after exiting QUERY.

4. Define the QUERY environment by issuing the >DEFINE command. In order to route QUERY output to the data file built in step 1, you must set OUTPUT equal to LP. Locate the entries that you want written to the data file by issuing the >FIND command. (These commands might be kept in a QUERY XEQ file for convenience.)

5. If you want all of the data items from the records located by the FIND command to be written to the data file, issue a >REPORT command including the following statements:

   A NOPAGE statement so that the first 2 records of the file are non-empty.

   A Detail statement for each data item to be included in the data file, specifying the column offset for each item.

   Note that if the data base entries are longer than 512 bytes, you must use the REPORT command to select items totalling no more than that limit. The REPORT command statements can be kept in a QUERY/3000 procedure file.

6. Exit from QUERY.

Figure 3-6 gives an example of using QUERY to create a data file. For more information, consult the *QUERY Reference Manual*.

DSG/3000 accesses a data file built by QUERY as a fixed format file. You must indicate to DSG/3000 the length and offset of each variable. This can be accomplished interactively using the DATA DEFINITION Menu (Section IV) or programmatically using the GDEFNSET procedure (Section VII).

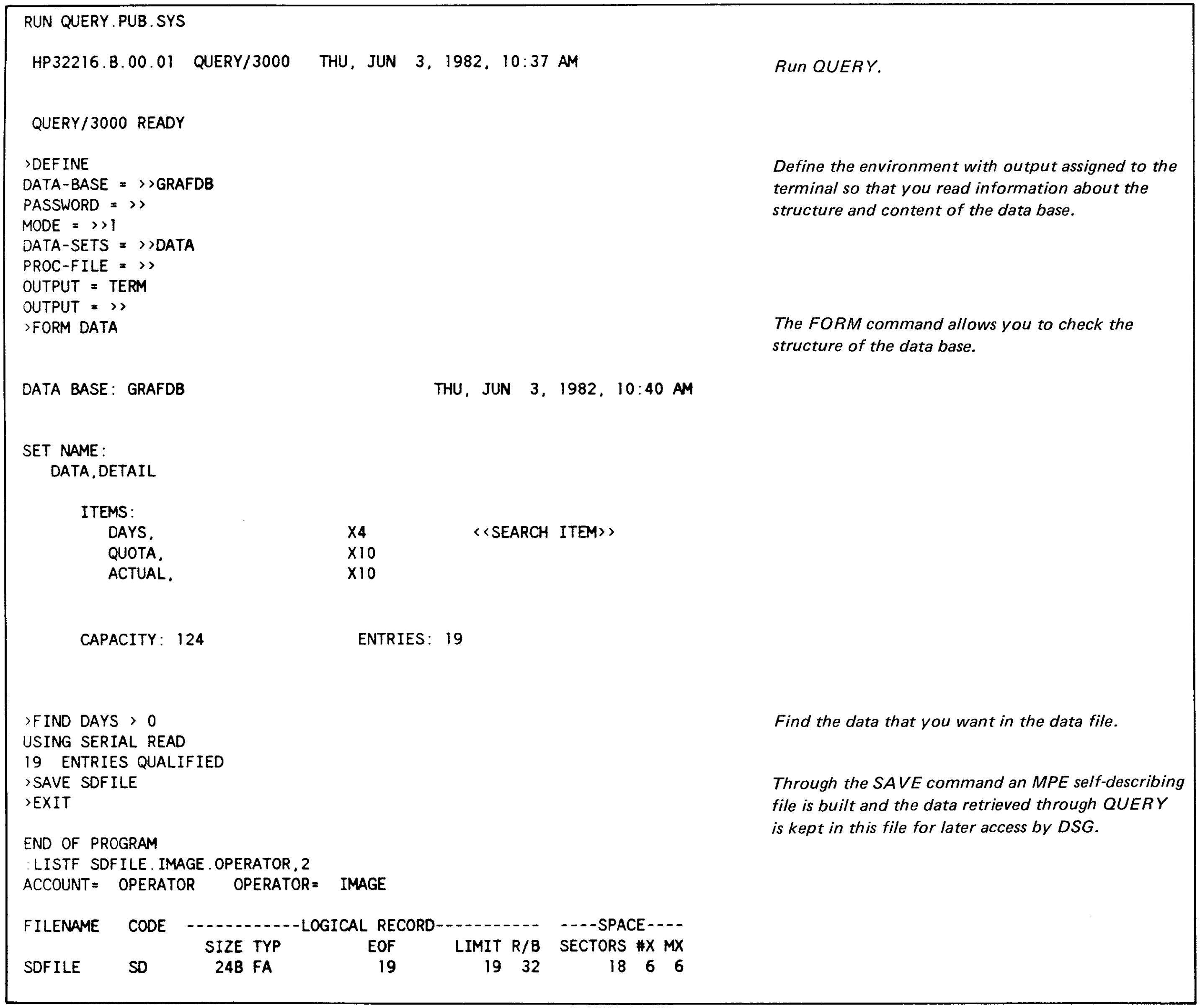
The data file created by Query in Figure 3-6 contains three variables: DAY (beginning in column 1), QUOTA (beginning in column 6), and ACTUAL (beginning in column 17).

QUERY places a carriage control character at the beginning of each line of data in the file. For DSG/3000 to properly access the data, you should add a 1 (one) to the value of each offset.

Therefore, you should specify the offsets and lengths for the three data variables as follows:

Variable	Offset	Length
DAYS	1	4
QUOTA	6	10
ACTUAL	17	10

Note that the offsets are the same as the column numbers.



```
RUN QUERY.PUB.SYS

 HP32216.B.00.01  QUERY/3000   THU, JUN  3, 1982, 10:37 AM
```

Run QUERY.

```
 QUERY/3000 READY

>DEFINE
DATA-BASE = >>GRAFDB
PASSWORD = >>
MODE = >>1
DATA-SETS = >>DATA
PROC-FILE = >>
OUTPUT = TERM
OUTPUT = >>
```

Define the environment with output assigned to the terminal so that you read information about the structure and content of the data base.

```
>FORM DATA
```

The FORM command allows you to check the structure of the data base.

```
DATA BASE: GRAFDB                          THU, JUN  3, 1982, 10:40 AM


SET NAME:
   DATA,DETAIL

      ITEMS:
         DAYS,                    X4            <<SEARCH ITEM>>
         QUOTA,                   X10
         ACTUAL,                  X10


      CAPACITY: 124               ENTRIES: 19



>FIND DAYS > 0
USING SERIAL READ
19  ENTRIES QUALIFIED
```

Find the data that you want in the data file.

```
>SAVE SDFILE
>EXIT
```

Through the SAVE command an MPE self-describing file is built and the data retrieved through QUERY is kept in this file for later access by DSG.

```
END OF PROGRAM
:LISTF SDFILE.IMAGE.OPERATOR,2
ACCOUNT=  OPERATOR    OPERATOR=  IMAGE

FILENAME   CODE  ------------LOGICAL RECORD-----------  ----SPACE----
                   SIZE TYP         EOF      LIMIT R/B  SECTORS #X MX
SDFILE     SD       248 FA           19         19  32       18  6  6
```

Figure 3-6. Using QUERY to Build a Data File

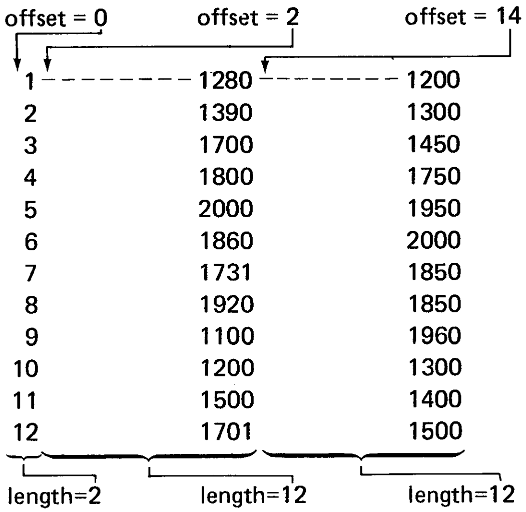
Method V -- VPLUS/3000

If you usually use VPLUS/3000 for data entry, your data is kept as a VPLUS/3000 batch file. DSG/3000 can access this batch file as a fixed format file. You must indicate the length and offset of each variable to DSG/3000. This is done interactively in the DATA DEFINITION screen menu (Section V) of GRAPH or programmatically in the GDEFNSET procedure (Section VII).

You can determine the length and offset of the variables from the VPLUS/3000 formsfile, which lists the variables in the order in which they appear in a data record with their corresponding lengths. By adding up the lengths of all the preceding variable fields in the record, you can determine the offset of any given variable.

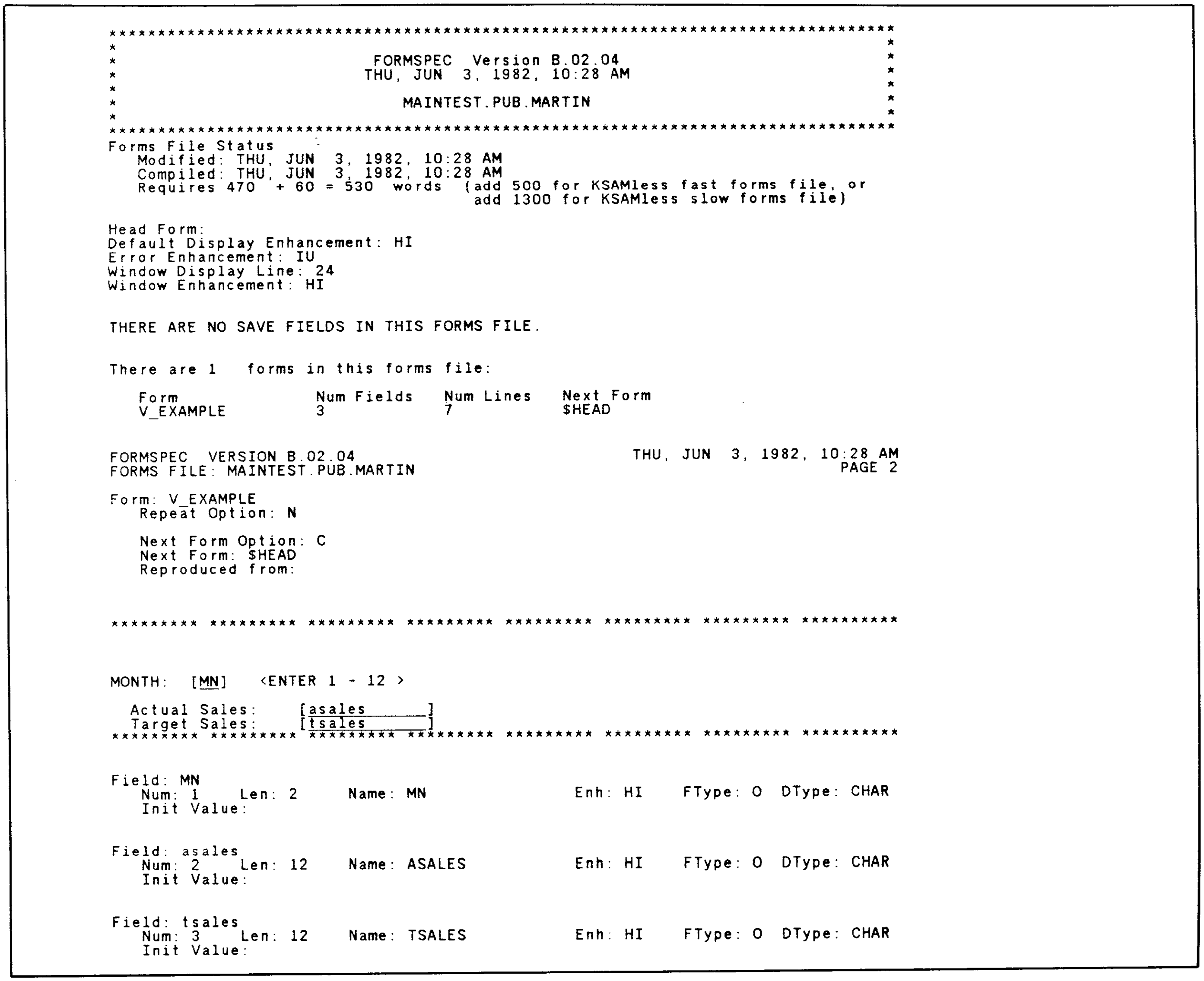
See the *VPLUS/3000 Reference Manual* for further information about VPLUS/3000 and using REFORMAT to subset multi-form batch files.

Figure 3-7 shows an example formsfile which could be used to collect three variables named MONTH, SALESACTUAL, and SALES-TARGET. After running ENTRY.PUB.SYS, a batch file could be produced with the following format:



1	1280	1200
2	1390	1300
3	1700	1450
4	1800	1750
5	2000	1950
6	1860	2000
7	1731	1850
8	1920	1850
9	1100	1960
10	1200	1300
11	1500	1400
12	1701	1500

Note that blank spaces have been added.



```
********************************************************************************
*                                                                              *
*                        FORMSPEC  Version B.02.04                             *
*                       THU, JUN  3, 1982, 10:28 AM                            *
*                                                                              *
*                           MAINTEST.PUB.MARTIN                                *
*                                                                              *
********************************************************************************
Forms File Status
   Modified: THU, JUN  3, 1982, 10:28 AM
   Compiled: THU, JUN  3, 1982, 10:28 AM
   Requires 470  + 60 = 530  words  (add 500 for KSAMless fast forms file, or
                                     add 1300 for KSAMless slow forms file)

Head Form:
Default Display Enhancement: HI
Error Enhancement: IU
Window Display Line: 24
Window Enhancement: HI


THERE ARE NO SAVE FIELDS IN THIS FORMS FILE.


There are 1   forms in this forms file:

   Form              Num Fields   Num Lines   Next Form
   V_EXAMPLE         3            7           $HEAD


FORMSPEC  VERSION B.02.04                           THU, JUN  3, 1982, 10:28 AM
FORMS FILE: MAINTEST.PUB.MARTIN                                          PAGE 2

Form: V_EXAMPLE
   Repeat Option: N

   Next Form Option: C
   Next Form: $HEAD
   Reproduced from:


********* ********* ********* ********* ********* ********* ********* **********


MONTH:  [MN]   <ENTER 1 - 12 >

  Actual Sales:     [asales      ]
  Target Sales:     [tsales      ]
********* ********* ********* ********* ********* ********* ********* **********


Field: MN
   Num: 1    Len: 2     Name: MN                   Enh: HI    FType: O  DType: CHAR
   Init Value:


Field: asales
   Num: 2    Len: 12    Name: ASALES               Enh: HI    FType: O  DType: CHAR
   Init Value:


Field: tsales
   Num: 3    Len: 12    Name: TSALES               Enh: HI    FType: O  DType: CHAR
   Init Value:
```

Figure 3-7. VPLUS/3000 Example

Method VI -- User Program

Any properly formatted file generated by a user program can be used as a DSG/3000 data file. For example, if you have a program that generates reports periodically, you can output data to an MPE file in free or fixed format. You can then designate this file as your DSG/3000 data file. If you wish to include graphic displays with your reports, you can call DSG/3000 procedures from your report generating program to access the data file and draw the graph.

Structuring the Data File

DSG/3000 provides several optional features which allow you control over the variables and data to be plotted. These features include:

- Qualification of variables
- Data subsetting
- Transformation of data values
- Specification of missing data

These features are discussed below so that you will be aware of them as capabilities when you create your data file. The actual specifying procedures are discussed for interactive users in Section IV and for programmatic users in Section V.

Data Manipulation Order

In the data manipulation order below, steps 2 through 6 are optional.

1. Raw data is converted to an internal DSG file.
2. A sort is performed if you have transformations.
3. Transformations are done.
4. Subsetting is done.
5. A sort is done by independent variable (if no transformations are being performed).
6. Qualifications are done.
7. Graphing is done.

Note that only one sort is performed for a line or pie chart, pie charts at the user's option. Bar charts and scattergrams are not sorted.

Data Qualifications

There is often more than one way of constructing a data file to produce a particular graph. Your data may have been stored and retrieved in such a way that a single data field contains information about more than one graphable entity. DSG/3000 allows you to *qualify* the data in a field so that the individual dependent or Y axis variables are uniquely identified.

For example, suppose you want to draw a four-line graph with each line representing the number of employees in a particular company plotted against time. The data file in Figure 3-8 has only three data fields: COMPANY, NUM-EMPLOYEES, and TIME.

Acme	20	1975
Acme	23	1976
Acme	24	1977
Bayside	10	1975
Bayside	15	1976
Bayside	17	1977
Calco	21	1975
Calco	45	1976
Calco	50	1977
Dunwit	19	1975
Dunwit	24	1976
Dunwit	34	1977

Figure 3-8. Data File to be Qualified

The actual graph (shown in Figure 3-9) has four dependent variables (number of employees in Acme, Bayside, Calco, and Dunwit) plotted against time, but their values all appear in the second data field. You must qualify the variable NUM-EMPLOYEES using the variable COMPANY to create the following 4 subsets:

VARIABLE	QUALIFIER
NUM-EMPLOYEES	COMPANY=Acme
NUM-EMPLOYEES	COMPANY=Bayside
NUM-EMPLOYEES	COMPANY=Calco
NUM-EMPLOYEES	COMPANY=Dunwit

Qualifiers are described by expressions using the relational and logical operators listed in Table 3-2. You can qualify data variables interactively in the QUALIFICATIONS Screen Menu (Section V) or programmatically in the GDEPVARSET procedure (Section VII).

Table 3-2. Data Qualifying and Subsetting Operators

Relational Operators	=	Equals
	<>	Not equal
	>	Greater than
	>=	Greater than or equal to
	<	Less than
	<=	Less than or equal to
Logical Operators	not	Condition not true
	NOT	
	and	Both conditions true
	AND	
	or	At least one condition true
	OR	

The relational operators are performed in the order they appear from left to right and have precedence over logical operators. Logical operations are performed in the following order:

1. NOT
2. AND
   OR

This means that NOT operations are performed first, and OR operations are performed last. This order may be overridden by expressions in parentheses, whose operations are performed before all others.

Examples of Qualifying Expressions

```
COMPANY="Acme Corp." AND YEAR<1975
SEX=M OR SALARY<20000
MEMPLOYEES>FEMPLOYEES
```

In these examples, COMPANY, YEAR, SEX, SALARY, FEMPLOYEES, and MEMPLOYEES are variable names associated with fields in the data file. Quotation marks are required when the character string contains embedded blanks. Either single quote marks (apostrophes) or double quotation marks are allowed.

Note: When data is qualified, the data that does not meet the qualifications is treated as missing data. Therefore, you may get unexpected results if the independent X-axis variable is textual. This will not occur when the independent X-axis variable is numeric. (Textual variables are always treated as unique where numeric variables are not unique, and the matching numbers align themselves appropriately.) Missing data is discussed later in this section.

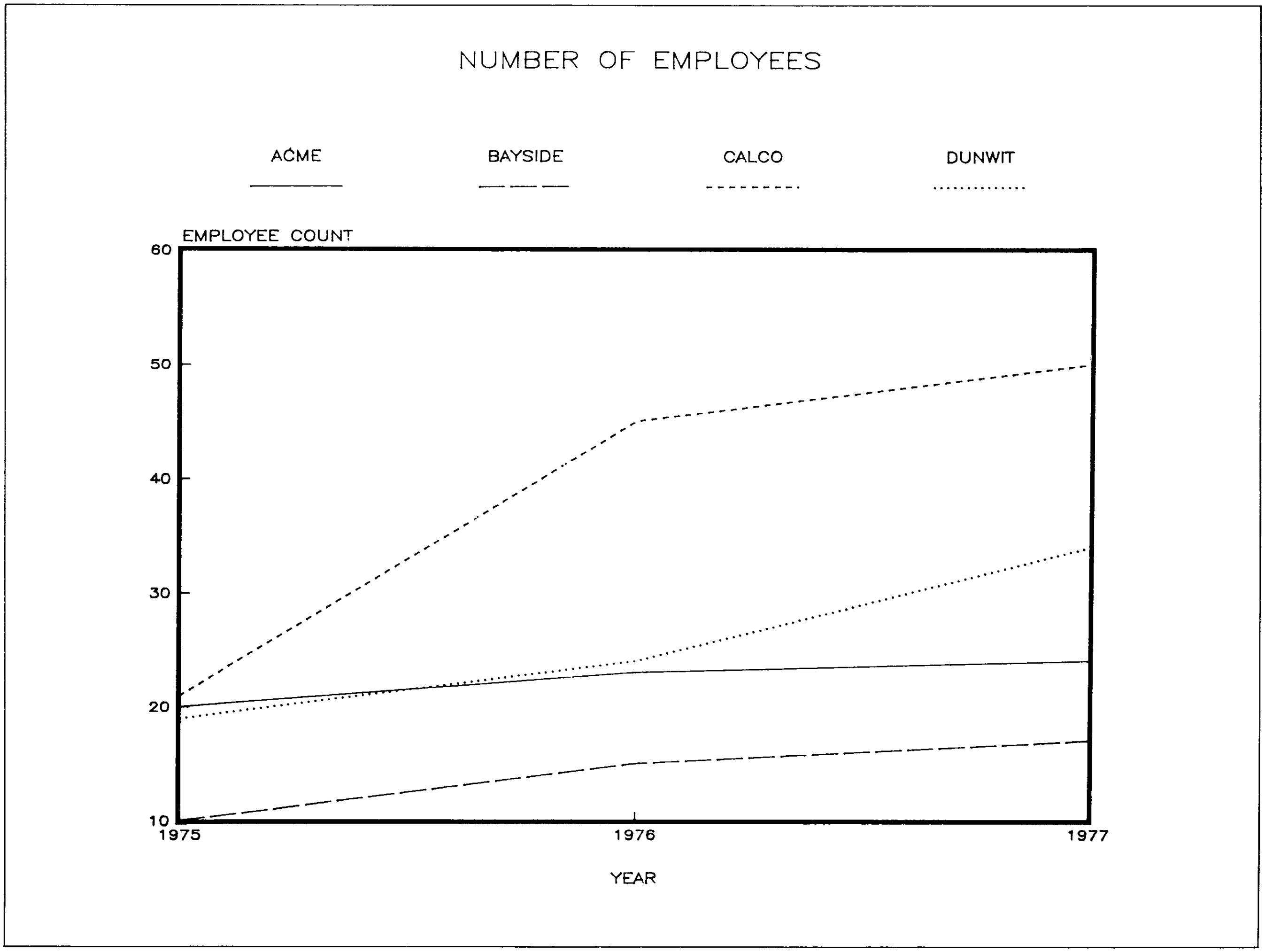


Figure 3-9. Example Graph

Data Subsetting

The data file that you create may contain more information than you want to include in any one graph. DSG/3000 allows you to select a *subset* of the data file for inclusion in the particular chart that you are designing. This means that you can make several graphs from different subsets of the same data file.

Subsets are described by expressions using the same relational and logical operators listed in Table 3-2. While qualifications allow each data variable to have restrictions placed on it, subsetting restricts all variables to be graphed.

You can subset a data file by the value of any data field, not just by the values of the data fields that are being graphed. For example, the data file in Figure 3-10 has three data fields associated with the variables SALARY, EXPERIENCE, and POSITION. You could make a single-line graph of SALARY plotted against EXPERIENCE for all sales representatives by subsetting the data file with the expression:

```
POSITION="Sales Rep"
```

data file

2200	1	Sales Rep
2300	2	Sales Rep
2480	3	Engineer
2630	4	Engineer
2650	2	Sales Rep
3020	6	Sales Rep
2750	5	Sales Rep
2800	5	Manager

→ (subset)

data plotted

2200	1	Sales Rep
2300	2	Sales Rep
2650	2	Sales Rep
3020	6	Sales Rep
2750	5	Sales Rep

Figure 3-10. Subsetting a Data File

Data Transformations

Instead of plotting the raw data as it exists in the data file, you may prefer to create new data by mathematically transforming the existing data and plotting the resulting values. You can name new variables and define them as mathematical functions of the old variables using expressions in Table 3-3. DSG/3000 will perform the necessary calculations without changing the original data values. You can then reference the new variables as if they were fields in your data file even though the data file has not been altered.

You can specify data transformations interactively in the DATA TRANSFORMATIONS Screen Menu (Section IV) of of GRAPH, or programmatically in the GTRANSET procedure (Section V).

Table 3-3. Data Transformations

Arithmetic Operators	+	Add
	-	Subtract
	*	Multiply
	/	Divide
	**	Exponentiate
	MOD	Modulo (divide and save remainder)
Common Functions	LOG (exp)	Common (base-10) logarithm
	LN (exp)	Natural (base-e) logarithm
	ABS (exp)	Absolute value
	SQRT (exp)	Square root
Special Functions	CUMULATE (var)	Cumulation
	MOVEAVG (var,n)	Moving average

where *"var"* is a variable, *"exp"* is an expression, and *"n"* is an integer.

Order of Precedence:
1. **, common and special functions
2. *, /, MOD
3. +, -

Parentheses can be used to override the standard order of precedence.

Note: CUMULATE and MOVEAVG cannot be nested.

The common functions operate on mathematical expressions of data variables and arithmetic operators. For example, in the following transformation:

```
BACKLOG=ORDERS-SHIPMENTS
```

ORDERS and SHIPMENTS are previously defined variables.

Common functions can also be compounded to form a new variable, such as:

```
ROOT=SQRT(ABS(A+B))
```

where the variables A and B are added, the absolute value taken and the square root extracted to define the new variable ROOT. Functions can also be used themselves within expressions. For example:

```
G=20*LOG(P)
```

The transformations are processed in the order they are specified. The names associated with transformations may not match the names of variables in the data file. Therefore, transformed variables can be used within subsequent transformation statements, as this example illustrates:

```
FNUMBER=FNUM/1000 MNUMBER=MNUM/1000
TOTAL=FNUMBER+MNUMBER
```

NOTE: If data cannot be transformed or has illegal results, the missing data value will be substituted as the result of the equation.

The special cumulative function provides a running total of numeric data for a given variable. For example, if your data file contains TIME and SALES, and if you requested the following transformation:

```
SUM=CUMULATE(SALES)
```

the data file could be regarded as containing the following information:

TIME	SALES	SUM
1970	27658	27658
1971	24115	51773
1972	28224	79997
1973	36543	116540
1974	43428	159968

The special moving average function provides a running average for a given variable. The average is taken over the previous n values, as specified by the user. If no previous values are present, they are not averaged. For example, if your data file contains TIME and SALES, and if you requested the following transformation:

```
AVG = MOVEAVG(SALES,3)
```

The data file could be regarded as containing the following information:

TIME	SALES	AVG
1970	10000	10000
1971	15000	12500
1972	17000	14000
1973	21000	17666.7
1974	15500	17833.3

Missing Data

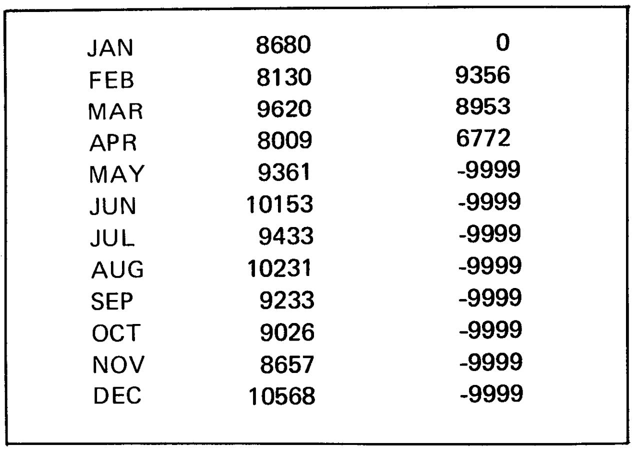
Some of the values of your variables may not be available to you at the time you create your data file. For example, a file that contains MONTH, SALES-QUOTA and SALES-ACTUAL may be created when all of the values for the first two variables are known, but the SALES-ACTUAL values are entered month by month. The values that are not present at the time you generate a graph are called *missing data.* If you designate data as missing, DSG/3000 will ignore it when plotting the graph. Any blank values in the data file are treated as missing data.

When you create the data file, choose a numeric value to enter wherever there are missing data values. You then specify this missing data value to DSG/3000 interactively in the DATA DEFINITION Screen Menu (Section IV), or programmatically through the GDATAFILESET procedure.

For the data file in Figure 3-11 with variables representing TIME, SALES-QUOTA, and SALES-ACTUAL, the entries with value -9999 are designated as missing data. A warning will be returned by DSG/3000 when the chart is drawn to let you know that some fields contain the missing data value. This warning may also be produced by using qualifications or transformations.

Any time a calculation is performed on missing data, its value becomes missing data except in the cases of CUMULATE and MOVEAVG. When the special function CUMULATE is being performed, the current value is substituted for the data point.

Also, when line charts are drawn, the pen continues from the present point to the next present point. It does not lift for a missing value unless the value occurs at the beginning or end of a line.



```
JAN      8680       0
FEB      8130    9356
MAR      9620    8953
APR      8009    6772
MAY      9361   -9999
JUN     10153   -9999
JUL      9433   -9999
AUG     10231   -9999
SEP      9233   -9999
OCT      9026   -9999
NOV      8657   -9999
DEC     10568   -9999
```

Figure 3-11. File with Missing Data

Note that a missing data variable can only refer to numeric variables. A blank column is also treated as a missing data variable for fixed format.

This also applies to the end-of-line for free format files. In free format, commas must be used as delimiters when missing data occurs between columns of other variables.

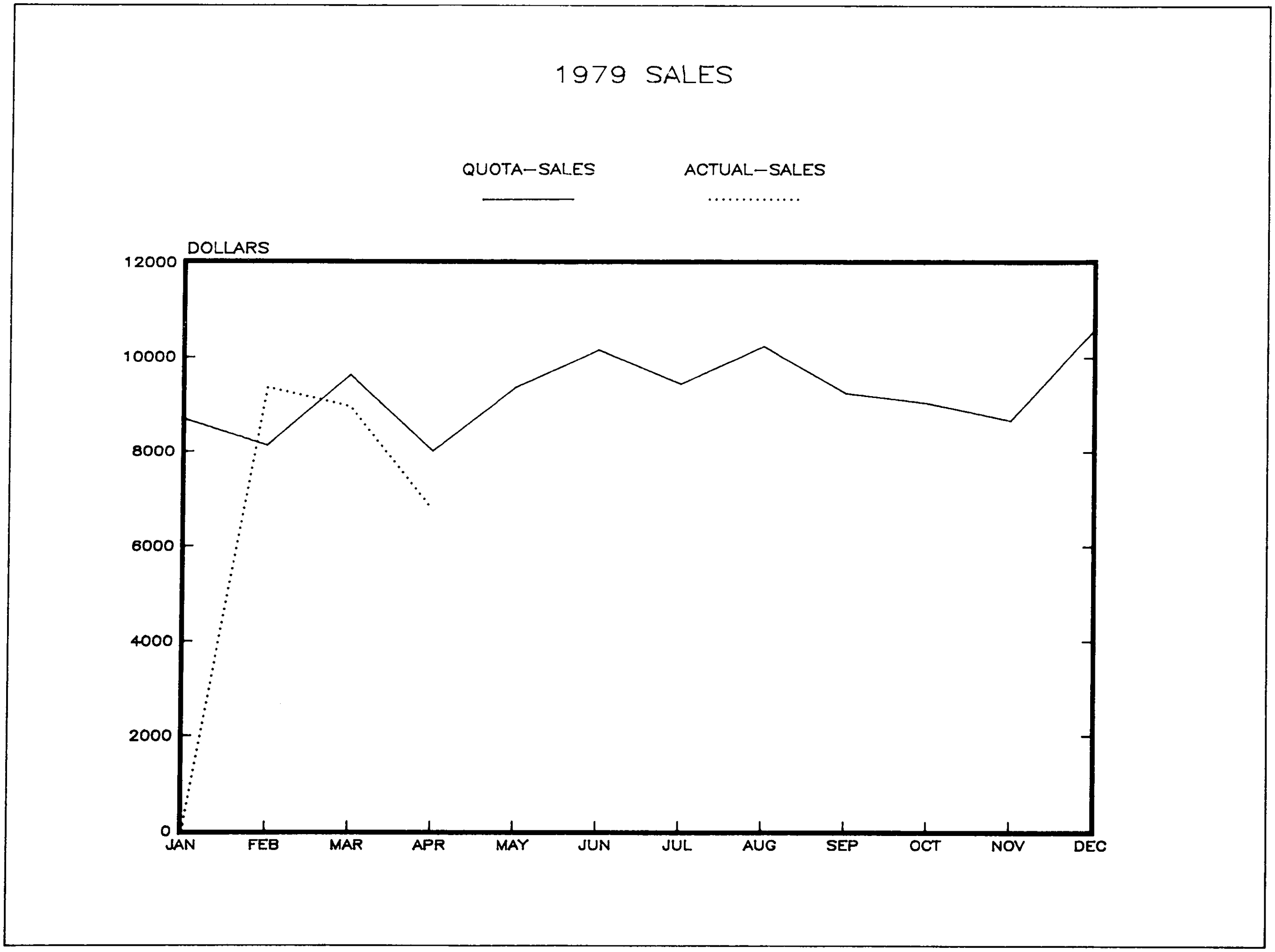


Figure 3-12. Line Graph with Missing Data Values

SECTION 4

Using DSG/3000 Interactively

Running DSG/3000 Interactively

DSG/3000 may be run interactively by using the program called GRAPH or programmatically using the procedures discussed in Section VII. This section presents general information about interactive use. Section V discusses individual menus in detail.

After you log on and run GRAPH.PUB.SYS, a series of formatted screens, called menus, will appear on your terminal. You design charts by simply filling out the menus. Each of the menus is complemented by the explanatory HELP facility, which appears on your terminal at the touch of a function key.

Special Function Keys

The keys labeled f1 through f8 have special functions assigned to them by GRAPH. If you enter the GRAPH Help Facility by pressing the last function key (f8), some of the keys are re-assigned to have different functions.

The function keys are displayed and labeled at the bottom of each screen menu. Any keys that are unlabeled are inactive for that screen menu. It is not necessary to memorize the function keys at any time.

Running GRAPH

To enter GRAPH, simply log on and type:

```
   :RUN GRAPH.PUB.SYS
```

After you press the carriage return, the Main Control Menu will appear on your terminal. In this and each succeeding menu, fill in the fields that appear in inverse video (black letters on a white field). Information can be entered only in these areas of the screen, called *unprotected fields*, since the rest of the screen is "protected" against any modification. The cursor will automatically move to the next unprotected field when you reach the end of the present field or if you press the TAB key.

To use DS to access GRAPH on another system, the DSLINE command must be issued as follows.

```
    :DSLINE dsdevice;LINEBUF=4048
```

where dsdevice is the name assigned to the remote computer.

When using a modem to access a remote computer, you must specify the terminal type to be 10 on the HELLO command as follows:

```
    :HELLO user.acct;TERM=10
```

Block Mode

Whenever you enter or change information on a menu, you must press the ENTER key to send the information to the computer. This is because GRAPH operates in *block mode*, in which the ENTER key performs a function similar to that of the RETURN key in *character mode*. (In block mode, the RETURN key does nothing other than position the cursor at the beginning of each line.)

If you wish to go on to another menu without making changes to the current menu, it is more efficient to press the DONE, PREVIOUS, or NEXT keys where applicable than to press ENTER. This avoids retransmitting information to the computer.

Message Window

The top line of each menu displays the title of the menu and, in some cases, the name of the current chart. GRAPH uses this "window" to return messages to you when it is performing certain functions or when it detects an error.

Correcting Errors

If you notice an error anywhere on a screen menu, you can correct it by positioning the cursor under the character you want to correct, using any of the terminal keys described in Section I. After you correct the mistake, press ENTER to send the corrected information to the computer.

Some errors are detected by GRAPH when you press ENTER. If GRAPH detects any errors, it returns a message in the message window as described above. This message provides the information you need to correct the error. You can then press ENTER. If you need more information about the menu, press HELP (f8), or refer to Appendix C.

Break Key

If you accidentally press the BREAK key while running GRAPH, you can recover by performing the following steps:

1. Press RESET TERMINAL twice.

2. Press RETURN to get the MPE colon prompt.

3. To turn the echo on, press the Escape (ESC) key and then the Colon (:) key (optional to see what you are typing).

4. Type RESUME and press RETURN. The message READ PENDING is displayed on the screen. On 262X terminals, in order to display the user keys, you must first press

   USER
   KEYS

5. Press REFRESH (function key f4) to display the current menu.

A power failure will simply interrupt execution of your program, but a full system failure will log you off the system. After a power failure, you can resume execution using the steps outlined above. If the system had failed when you attempted to resume in this manner, a system message would have prompted you to log on when you typed RESUME.

Terminating Plotting

If you find it necessary to terminate plotting while the plotter is operating, use <Control Y> to do so; turning off the plotter itself halts communication between the system or terminal and the plotter. <Control Y> operates when you press the Y key while holding down the CNTL key. The Y key may have to be pressed several times before control returns because of storage areas within the plotter itself. Recovery is the same as from Break, described on this page.

Terminating GRAPH

You can terminate a GRAPH session by returning to either the Main Control Menu or the Chart Design Control Menu (press the special function key labeled DONE) and entering an "E" for EXIT. The END OF PROGRAM message and colon prompt (:) indicate that you have been returned to the control of the operating system in character mode.

Using Menus

GRAPH menus are self-explanatory to the experienced user; for those less familiar,the HELP screens provide necessary information relative to the menu on your screen. Table 4-2 below summarizes the purpose of each menu. In Section V each menu is discussed in detail, including the possible entries for each field, examples of legal entries, defaults, and the particular use of function keys relative to individual menus.

It is not necessary to memorize the functions of the menus. The information included here is a guide which may prove helpful in following the chart design information that concludes this section.

Table 4-2. Summary of Screen Menus

MENU	FUNCTION
Main Control	Names and creates the current chart and chart file; selects the graphing function or utility to be performed. Also allows you to enter the QUERY subsystem or access figure files.
Chart Design Control	Branches to any one of the design menus used to specify or modify chart attributes.
Data Prompt	Constructs or modifies a GRAPH data file for input to DSG/3000.
Data Definition	Specifies the name of the data file; names and locates the variables within the data file. Optionally specifies a missing data value.
Data Transformations	Creates new variables by mathematically transforming the original data variables.
Pie Chart	Specifies variables, design options, data subsetting and explosions for a pie chart.
Line Chart	Specifies variables and data subsets for line charts and scattergrams.
Bar Chart	Specifies variables, data subsets, and design options for bar charts.
Qualifications	Qualifies data variables from the data file.

Table 4-2 (Continued)

MENU	FUNCTION
Font Definition	Allows users to map fonts and native languages to the four font numbers which are active when a chart is being designed.
Pie Chart Labels	Specifies color, texture, and labels for each segment of a pie chart.
Label Prompt	Specifies a label for each bar in a bar chart or for each tick in a line chart.
Line Chart Legends	Specifies color, texture, markers, and legend for each Y-axis variable in a line graph.
Bar Chart Legends	Specifies color, texture, and legend for each Y-axis variable in a bar chart.
Bar and Line Chart Titles	Specifies titles, subtitles, footnotes, and axes titles for line and bar charts.
Pie Chart Titles	Specifies titles, subtitles, and footnotes for pie charts.
Line Chart Axes	Specifies axis type, scaling, ticks, grids, and tick labels for both axes of a line chart.
Bar Chart Axes Numeric)	Specifies axis type, scaling, ticks, grids, tick and bar labels, and number of bars for both axes of a bar chart with a numeric independent variable.

Table 4-2 (Continued)

MENU	FUNCTION
Bar Chart Axes Textual)	Specifies axis type, scaling, ticks, grids, tick and bar labels, and number of bars for both axes of a bar chart with a textual independent variable.
Bar and Line Text Control	Specifies color, size, and font for text on a bar or line chart, including titles, subtitles, axis labels, legend labels, and footnotes.
Pie Text Control	Specifies color, size, and font for text on a pie chart, including titles, subtitles, labels and footnotes.
Annotations	Adds or deletes annotations: arrows, lines, boxes, or text. Fully specifies type and placement of each annotation.
Figure File	Adds or deletes figures (data and chart design) in a figure file for access by other subsystems.
Graphing Options	Specifies plotting device and address, speed, number of copies, size, and suppression of individual components.
Chart Information	Returns information about the current chart such as creator, date of creation, and chart type. Allows you to enter comments about the chart that are returned by both the Chart Information and Browse Menus.
Chart Browse	Returns names and comments about all charts in the current chart file and chart type.

Control Menus

The MAIN CONTROL MENU gives an overview of the system. The operations that you can perform using GRAPH can be grouped into two areas: chart file utilities and chart design. All of the chart file utilities can be performed from the MAIN CONTROL Menu. Chart design, which includes creating a new chart or modifying an existing chart, is managed through the CHART DESIGN CONTROL Menu. Entering an A or M (for Add or Modify a chart) on the Main Control Menu transfers you to the Chart Design Menu.

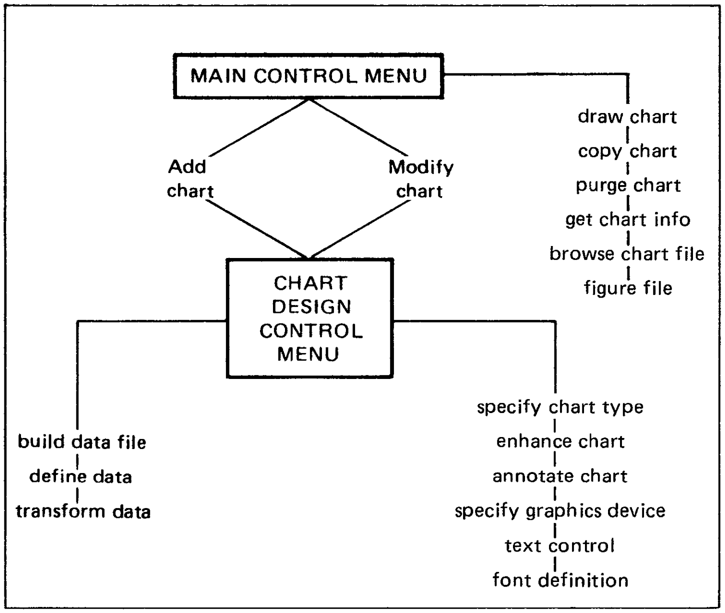


Figure 4-1. Functional Overview of GRAPH Control Menus

Defaults

Some of the items in the above menus have defaults assigned to them by GRAPH. In these cases, the default is already filled in for you when the menu appears on your screen. In order to override the default, you only need to type over the given values and press ENTER. For example, the CHART DESIGN CONTROL Menu comes up on your screen with and *S* for "Slide" as the default chart type. When you are specifying the chart type, you can type the appropriate letter over the *S*.

All of the defaults are documented in the screen menu references in Section V.

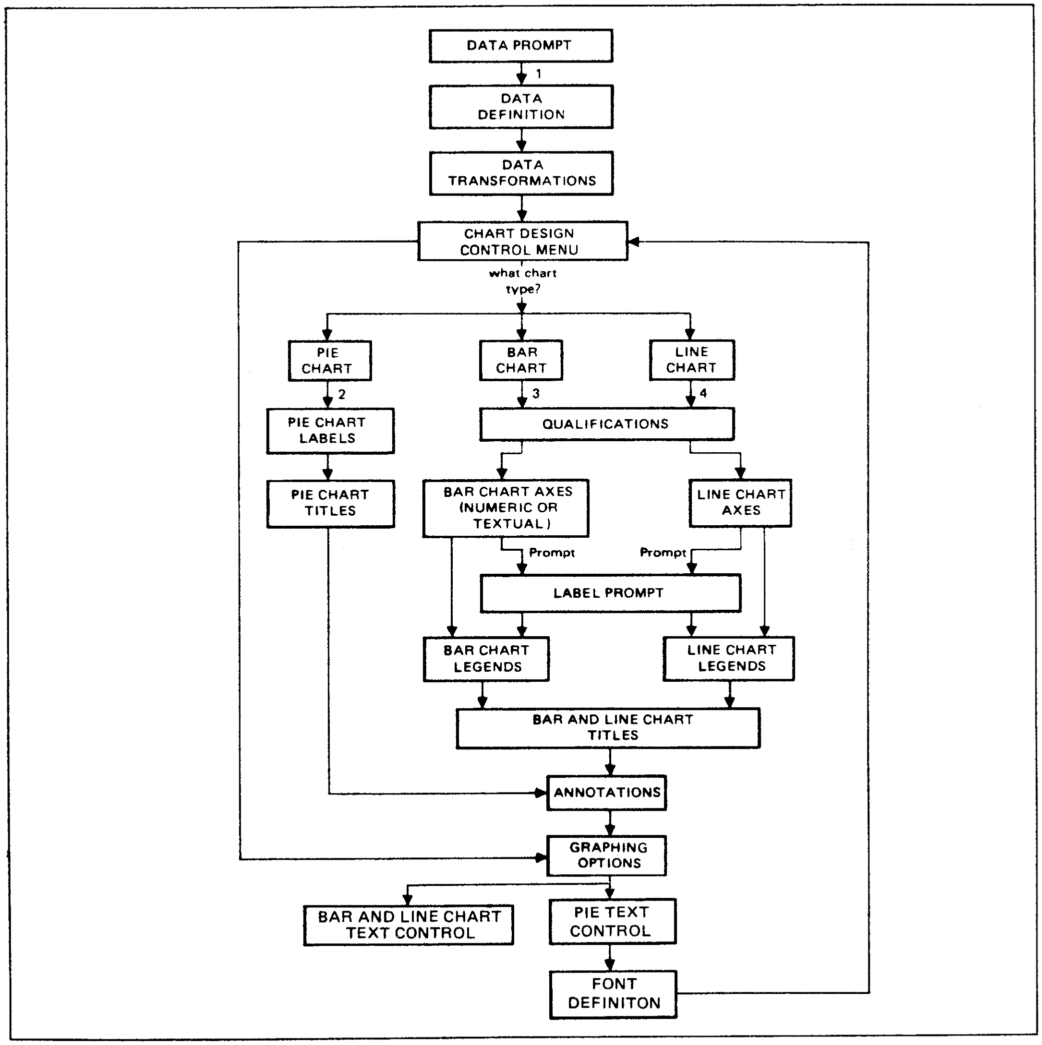


Figure 4-2. Menu Loops Controlled by NEXT Function Key

Moving from Menu to Menu

There is usually more than one path that you can take to get from your current screen menu to any other menu. Pressing DONE (f7) always brings you back to the CHART DESIGN CONTROL Menu or MAIN CONTROL Menu, from which you can reach almost any other menu by entering a letter designation. Alternatively, pressing NEXT (f6) brings you through a loop of menus that are logically linked together. You can go backwards through the loop by pressing PREVIOUS (f5). At the end of the loop you are returned to a Control Menu.

You can use NEXT efficiently as long as you need to enter information on subsequent menus. In general, you would use a combination of DONE and NEXT to quickly step through the menus needed to design a chart.

Figures 4-3, 4-7, and 4-9 illustrate how the menus are linked through the Control Menus for designing Pie Charts, Line Charts, and Bar Charts, respectively. Figure 4-2 above shows how the menus are linked by NEXT and PREVIOUS loops. Pressing NEXT brings you through the previous menus in the direction of the arrows, and pressing PREVIOUS brings you in the opposite direction.

There are four basic loops:

1. Data Loop
2. Pie Chart Loop
3. Line Chart Loop
4. Bar Chart Loop

Note that from the CHART DESIGN CONTROL Menu, you cannot use the NEXT key because the menu sequence depends on the type of chart you are designing. Otherwise, pressing the NEXT key allows you to step through a whole sequence of menus used to design a chart.

Steps in Chart Design

The following pages are intended to relate menus to the steps necessary in designing a typical chart -- in this instance, a pie chart. The basic chart is designed, enhancements are added, and then it can be modified. The chart is drawn at each stage. Less detailed information on line chart and bar chart design and enhancements follow.

In order to design a complete chart, you must perform the following steps:

1. Name the chart and the chart file in which it will reside. (MAIN CONTROL Menu)

2. If a suitable MPE data file does not already exist, build a data file. (DATA PROMPT Menu)

3. Name the data file; if it was not built from the Data Prompt Menu, describe its structure. (DATA DEFINITION Menu)

4. Specify the chart type. (CHART DESIGN CONTROL Menu)

5. Select the variables to be plotted. (PIE/BAR/LINE CHART Menus)

After filling in these menus you can draw the chart. You can also add some or all of the following features:

6. Create mathematical data transformations. (DATA TRANSFORMATIONS Menu)

7. Qualify the data variables. (QUALIFICATIONS Menu)

8. Select a subset of the data to be graphed. (PIE/LINE/BAR CHART Menus)

9. Enhance the chart. (TITLES, LEGENDS, LINE/BAR AXES, PIE LABELS, LABEL PROMPT, TEXT CONTROL, FONT DEFINITION Menus)

10. Annotate the chart. (ANNOTATIONS Menu)

11. Redirect output to plotter. (GRAPHING OPTIONS Menu)

To modify a chart, perform the following steps:

12. Inform GRAPH of the menu you wish to modify. (MAIN CONTROL Menu)

13. Instruct GRAPH concerning the type of information you wish to modify. (CHART DESIGN CONTROL Menu)

14. Modify chart by typing directly over information you desire to change.

Note: All of the data file concepts (format, transformations, subsets, qualifications, etc.) are discussed in Section III. The menu specifications are explained in detail in Section V.

Creating the Basic Pie Chart

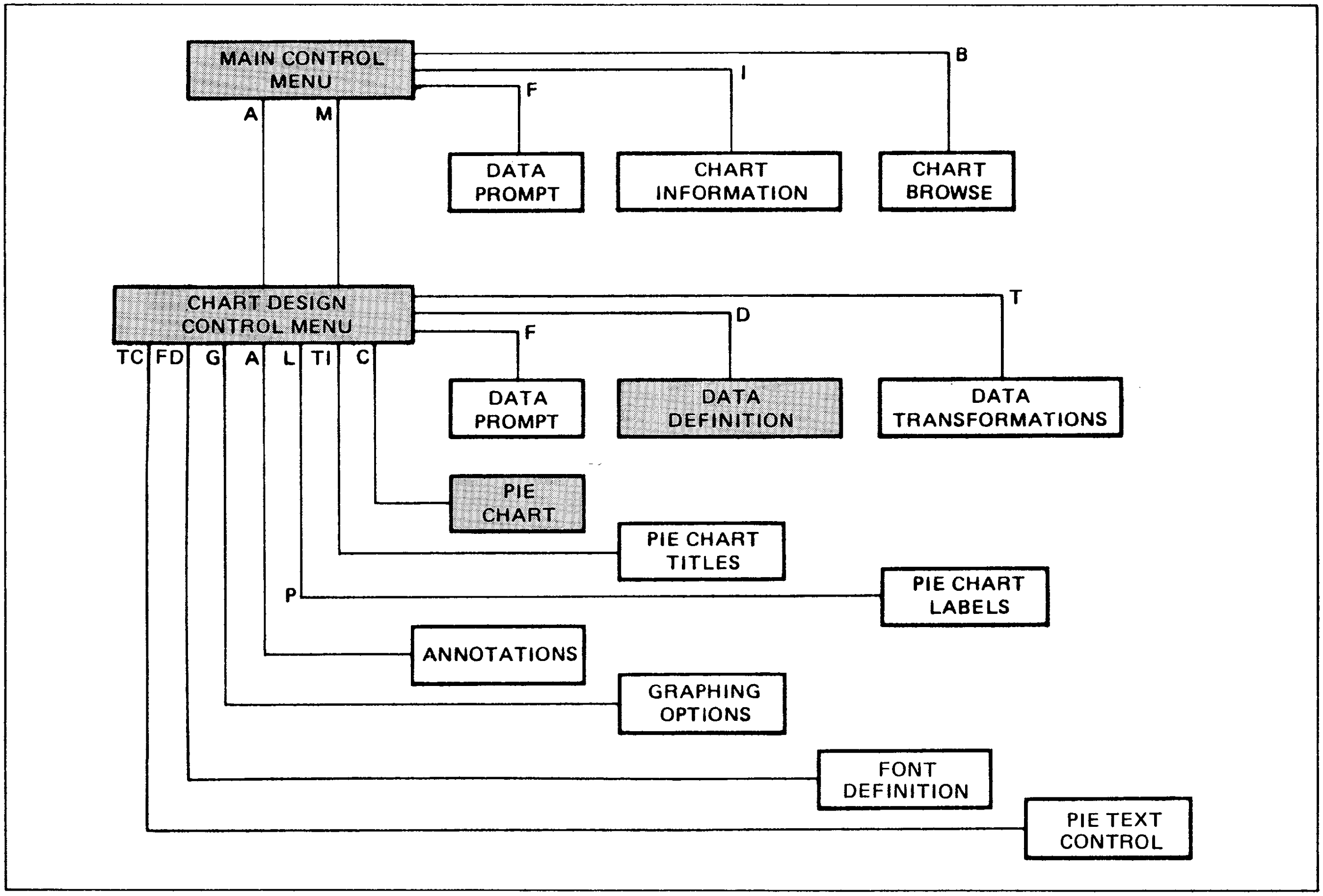


Figure 4-3. Pie Chart Road Map (required menus are shaded)

Figure 4-4 shows the routing of the menus used in designing a Pie Chart. It is followed by an example of menu use in such a design, divided into three phases--creating the pie chart, enhancing the chart, and modifying the chart.

Designing a Pie Chart

A pie chart plots one variable, referred to as the data variable, from the data file. A second variable, called the label variable, can serve as a source of labels for your chart or you can supply the labels as enhancements. Effective pie charts have a limited number of segments, often no more than 8. If the data results in a segment of less than 2 degrees, a warning message is returned: you should select a subset of the data file for plotting so that the segments and labels will be more than 2 degrees and thus discernable when drawn.

The following simple example illustrates the steps required to design a basic pie chart such as the one shown in figure 4-5. Suppose you want to draw a pie chart showing the breakdown of a budget by department. To do so, perform the steps described below at your terminal.

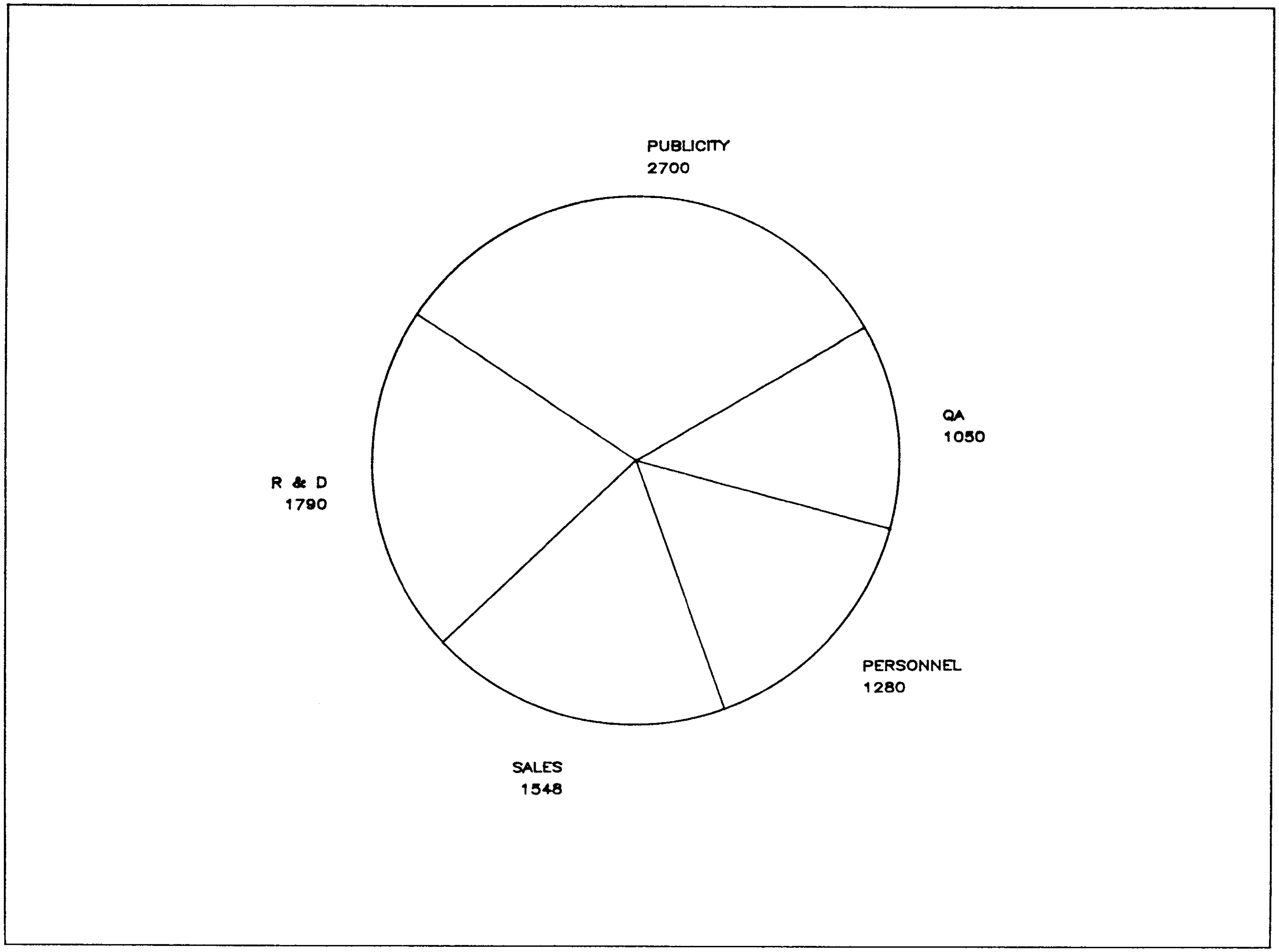


Figure 4-4. Pie Chart without Enhancements

1. After logging on to the system at the graphics terminal, type

   RUN GRAPH.PUB.SYS

   GRAPH displays the MAIN CONTROL menu on your screen.

2. On the MAIN CONTROL Menu, type *A*; then type in *FIRSTCHART* for Chart Name and *CF1* for Chart File Name. Press ENTER.

   You are telling GRAPH that you want to create a chart named FIRSTCHART to reside in a file named CF1.

3. On the CHART DESIGN CONTROL type *F* and press ENTER.

   You are telling GRAPH that you want to build a data file.

4. Fill out the DATA PROMPT Menu as follows, pressing TAB to go to the next field, or Control TAB to go back to the previous field:

```
File name:  FINANCE

VAR 1                    VAR 2
name        type         name        type
DEPT         T           TOTAL        N

R&D                      1790
SALES                    1548
Q A                      1050
PERSONNEL                1280
PUBLICITY                2700
```

When you are done, press ENTER.

You are giving your data file a name (FINANCE) and specifying two variables. DEPT is a textual variable; Total is a numeric variable. You then enter the values for the variables. When you press ENTER, the cursor scans the screen and returns to the top of the screen when finished.

Press NEXT, which is shown on the bottom of the menu to be f6.

GRAPH brings you to the DATA DEFINITION Menu.

5. On the DATA DEFINITION Menu, type FINANCE next to Data File Name and press ENTER.

   Press DONE (f7)

   You are naming your data file FINANCE. GRAPH fills in the necessary fields from the information you specified in the DATA PROMPT Menu. GRAPH returns you to the CHART DESIGN CONTROL Menu.

6. On the CHART DESIGN CONTROL Menu, type *C* next to Selection, type *P* for Chart Type, and Press ENTER.

   You are specifying your chart to be a pie chart.

7. On the PIE CHART Menu, type TOTAL next to Data Variable, type DEPT next to Label Variable, and press ENTER.

   You want the pie segments to represent the amounts given by the variable TOTAL, and to be labeled by DEPT.

8. Press DRAW (f3).

   GRAPH returns a message in the message window telling you that it is preparing the data. The pie chart in Figure 4-4 should appear on your terminal.

Note: This example continues below after a general discussion of pie chart enhancements. Keep GRAPH running on your terminal as you read on. If you do not wish to go further at this time, press DONE (f7), type E on the CHART DESIGN CONTROL Menu, and press ENTER.

Optional Pie Chart Enhancements

There are optional enhancements that you may add if you wish. The enhancements that are listed here can be added in any combination or order. Refer to the specifications at the end of this section for information on filling out each particular chart.

To redirect output to a plotter or the Laser printer, enter a *G* on the CHART DESIGN CONTROL Menu and fill out the GRAPHING OPTIONS Menu.

To arrange the segments in the order of occurrence in the data file rather than ascending order, enter a *C* and *P* in the CHART DESIGN CONTROL Menu and blank out the sort field in the PIE CHART Menu.

To specify colors or textures, enter an *L* in the CHART DESIGN CONTROL Menu or press NEXT from the PIE CHART Menu and fill out the PIE CHART LABELS Menu.

To add labels or later modify the existing labels, type the segment labels in the PIE CHART LABELS Menu. (GRAPH places labels outside the segments; to place labels inside the segments use the ANNOTATIONS Menu.)

To label each segment with percentages calculated from the data file, type any character in the "Print Percentages" box in the PIE CHART Menu.

To explode a segment, enter an explosion specification expression in the PIE CHART Menu.

To specify size, color, and font for all standard text, enter *TC* on the CHART DESIGN CONTROL Menu and fill out the PIE TEXT CONTROL Menu. (To map the four available fonts to the font numbers, enter *FD* on the CHART DESIGN CONTROL Menu and fill out the FONT DEFINITION Menu.)

To add titles or footnotes, press NEXT (f6) from the PIE CHART LABELS Menu or enter *TI* on the CHART DESIGN CONTROL Menu and fill out the PIE CHART TITLES Menu.

To annotate the chart, enter *A* on the CHART DESIGN CONTROL Menu or press NEXT from the PIE CHART TITLES Menu and fill out the ANNOTATIONS Menu.

To select a subset of the data, enter a subset specification expression on the PIE CHART Menu.

To create new data mathematically, enter a *T* on the CHART DESIGN CONTROL Menu and fill in the DATA TRANSFORMATIONS Menu.

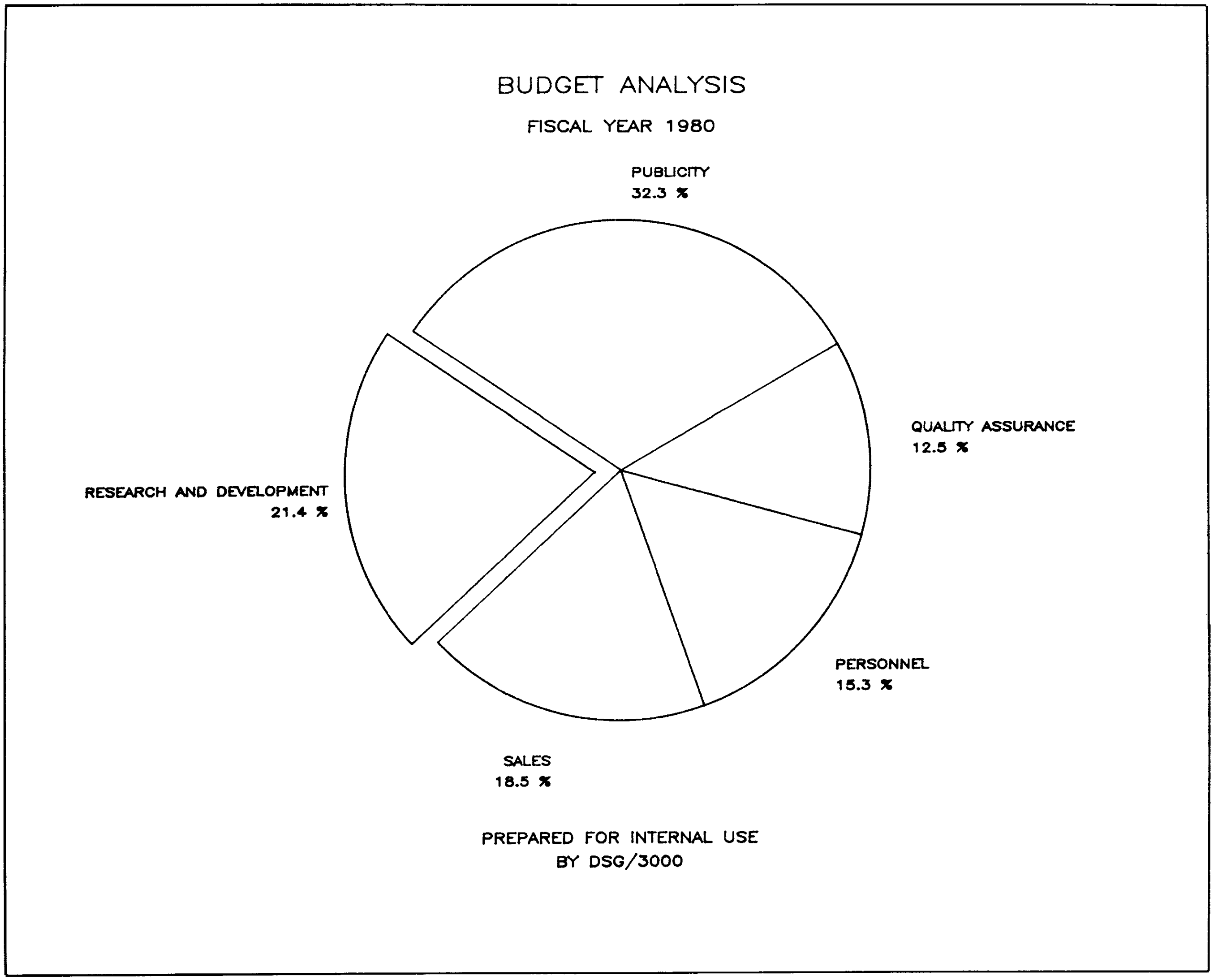


Figure 4-5. Pie Chart with Enhancements

Enhancing A Pie Chart

To add enhancements to the basic pie chart so that your chart looks like figure 4-5, perform the steps below.

If you have exited from GRAPH since creating the basic pie chart FIRSTCHART, described above, you should return it to the current chart status now, to continue the example. To do so, log on, run GRAPH, enter an *M*, *FIRSTCHART*, and *CF1* on the MAIN CONTROL Menu, and enter a *C* and *P* on the CHART DESIGN CONTROL Menu. Press DRAW to see the graph.

9. Press SHIFT-G DSP.

   You are turning off the graphics display.

   Press SHIFT-A DSP.

   You are turning on the alphanumeric display.

   In the PIE CHART Menu, space over the first field under *Options* to blank it out. Then type any character in the second field. Press ENTER.

   You are telling GRAPH to label the segments with percentages.

   Press DRAW (f3).

10. Press SHIFT-G DSP and SHIFT-A DSP.

    You are turning off the graphics display and turning on the alphanumeric display.

    On the PIE CHART Menu, enter this "explosion" specification:

    DEPT="R&D"

    In explosion or subset specifications, quotes are required around expressions containing blanks.

Press ENTER, and then DRAW (f3).

The R & D segment is exploded.

11. Press NEXT. (F6)

    You are indicating that you want to go on to the PIE CHART LABELS Menu.

    On the PIE CHART LABELS Menu, enter the following segment labels in the third column:

    QUALITY ASSURANCE
    PERSONNEL
    SALES
    RESEARCH AND DEVELOPMENT
    PUBLICITY

    Press ENTER

    Rather than take labels from the data file, you name them here in the desired form, replacing R & D with RESEARCH AND DEVELOPMENT and QA with QUALITY ASSURANCE. Note that the segments are labeled clockwise from the two o'clock position. (Turn on the graphics display to review the graph.)

12. Press NEXT (f6) to go on to TITLES Menu.

    Fill in the TITLES Menu as follows:

    Title: BUDGET ANALYSIS

    Subtitle: FISCAL YEAR 1982

    Footnote: PREPARED FOR INTERNAL
    USE BY [YOUR NAME]

    When you are done, press ENTER and then DRAW.

    You are adding a title, subtitle, and footnote. The pie chart in Figure 4-6 should appear on your graphics device.

13. Press SHIFT-G DSP and SHIFT-A DSP.

    Press DONE (f7)

    After you turn off the graphics display and turn on the alphanumeric display, GRAPH returns you to the CHART DESIGN CONTROL Menu.

    Type "E" and press ENTER

    You exit from GRAPH. The colon (:) prompt indicates that you have been returned to the operating system.

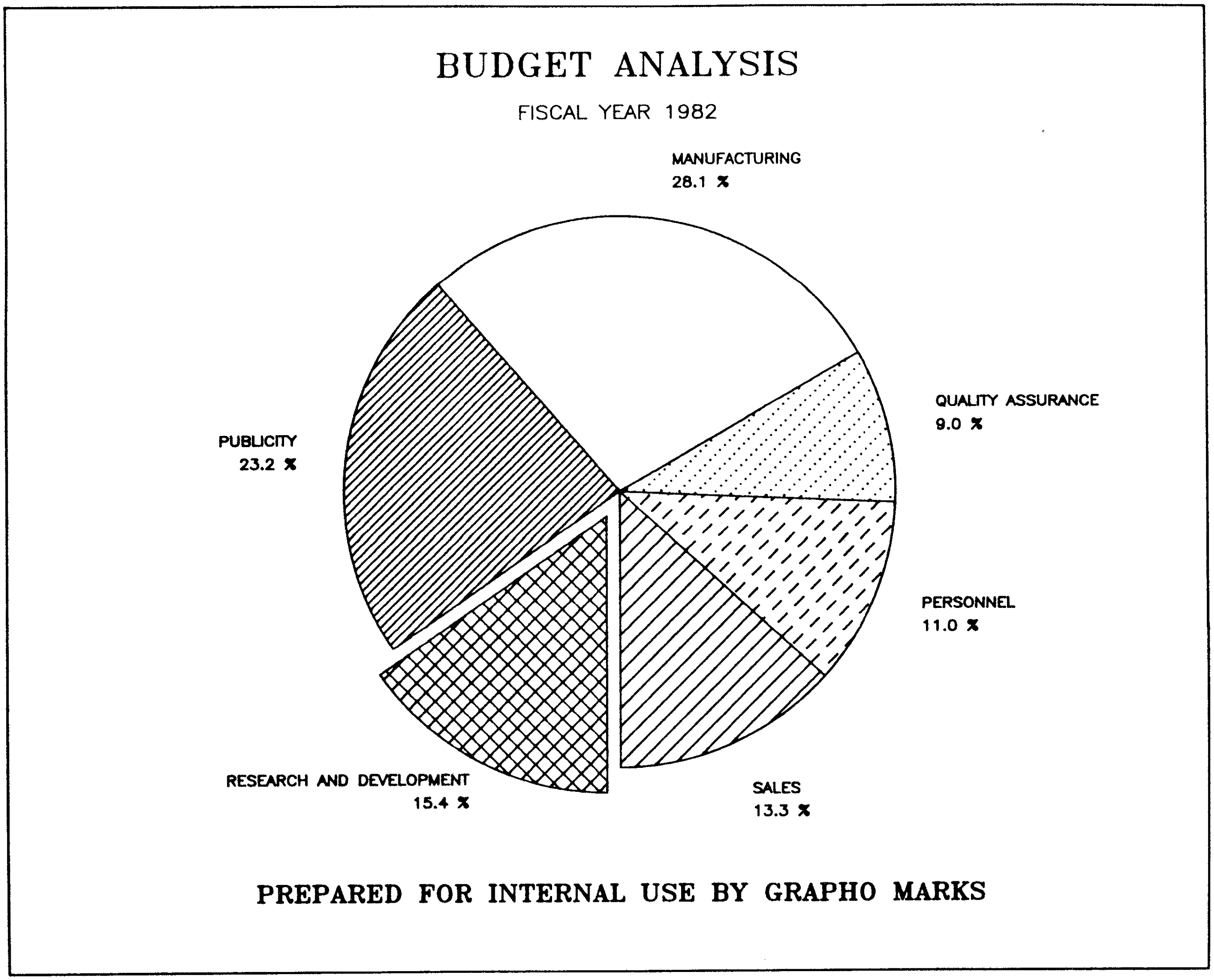


Figure 4-6. Modified Pie Chart

Modifying A Pie Chart

Suppose you want to make the following modifications to the pie chart:

-- add the MFG Department to the data file
-- print the label MANUFACTURING next to the graph
-- add surface textures and vary the type fonts to make the graph more interesting visually

Perform the following steps at your terminal (if you have not logged off, ignore steps 14 and 15):

14. Log on and enter GRAPH:

```
RUN GRAPH.PUB.SYS
```

15. On the MAIN CONTROL MENU, type *M*; then type in *FIRSTCHART* for Chart Name and *CF1* for Chart File. Press ENTER.

16. On the CHART DESIGN CONTROL MENU, type *F* and press ENTER.

You inform GRAPH that you want to modify the chart named FIRSTCHART in file CF1. You then ask to see the data file.

17. On the DATA PROMPT Menu, type in *FINANCE* for File name and press ENTER.

On the first blank line, type *MFG* in the first column and *3264* in the second column.

You are adding a data entry.

Press ENTER

GRAPH displays a warning message: "Will Overwrite File: FINANCE Press ENTER to confirm"

Press ENTER again.

When the file has been modified, the cursor returns to the top of the screen and the warning message is cleared.

Press DONE (f7).

18. On the CHART DESIGN CONTROL Menu, type *L* and press ENTER.

    You are indicating that you want to enhance the chart labels.

19. On the PIE CHART LABELS Menu, add *MANUFACTURING* to the end of the list of segment labels. Press ENTER.

    You are labeling the MFG segment. MFG is the last segment because it has the largest data value in the data file.

    In the column labeled Texture, type 2 opposite QUALITY ASSURANCE, 3 opposite PERSONNEL, 4 opposite SALES, 5 opposite RESEARCH and DEVELOPMENT, and 6 opposite PUBLICITY. Press ENTER.

    You are specifying textures for the segments of the pie.

    Press DONE.

    Returns you to the CHART DESIGN CONTROL Menu.

20. On the CHART DESIGN CONTROL Menu, type *TC*. Press ENTER.

    You are indicating that you wish to modify the graph using the PIE TEXT CONTROL Menu.

21. On the TEXT CONTROL MENU, in the column labeled "Size," type *12* opposite Main Title and *8* opposite Footnote.

    In the column labeled "Font," type *3* opposite Main Title and Footnote. Press ENTER.

    In the Size column you are changing the size of the type used in the Main Title and Footnote. In the Text column, you are changing the type font used for the Main Title and Footnote.

    Press DONE (f7).

22. On the CHART DESIGN CONTROL Menu, type G. Press ENTER.

    You are moving to the GRAPHING OPTIONS Menu.

23. On the GRAPHING OPTIONS Menu, under Device, type S over the default H opposite TEXT.

    Press ENTER.

    Press DRAW.

    You are making the four software fonts available for use. You have specified the use of one of these fonts in step 16. Only one hardware font is available.

NOTE: To plot the chart that is now drawn, you must enter the Device identification and HP-IB address. Once you change the identification of the output device, it becomes the default device and when you press DRAW, DSG/3000 will attempt to plot your chart rather than draw it on the screen. You have to change your Device ID on the GRAPHING OPTIONS Menu back to the number of your terminal if you wish to draw to the screen again.

Designing a Line Chart or Scattergram

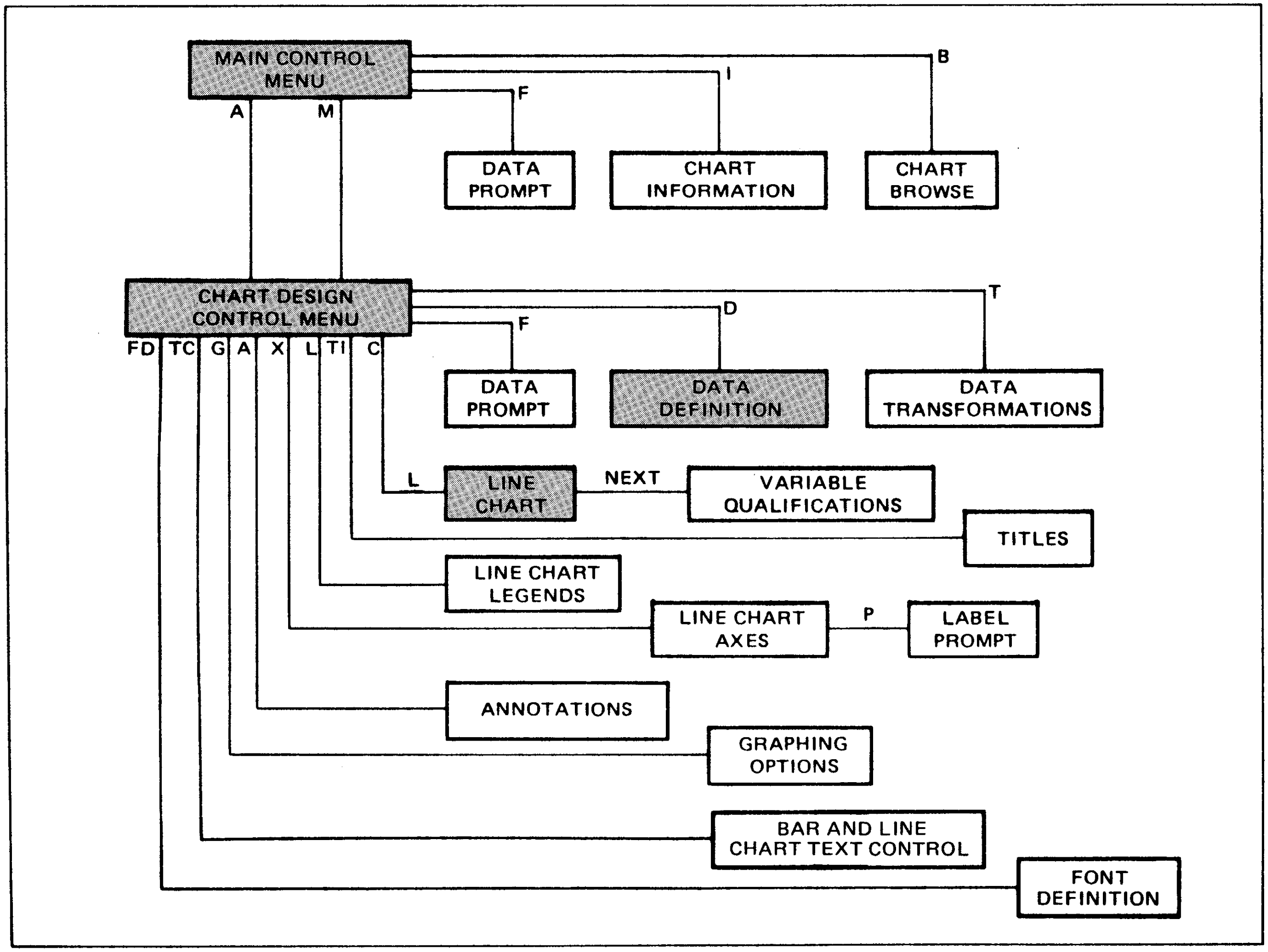


Figure 4-7. Line Chart Road Map (required menus are shaded)

In a line chart or scattergram, all of the variables must be numeric. There can be up to 8 Y-axis variables, each represented by a different line texture (or different markers for scattergrams) scaled along the Y-axis. There is one independent variable, usually scaled along the X-axis.

Figure 4-7 shows the routing of the menus used in designing a line chart or scattergram from the CHART DESIGN CONTROL Menu. The required menus are shaded. Each menu is explained is Section V.

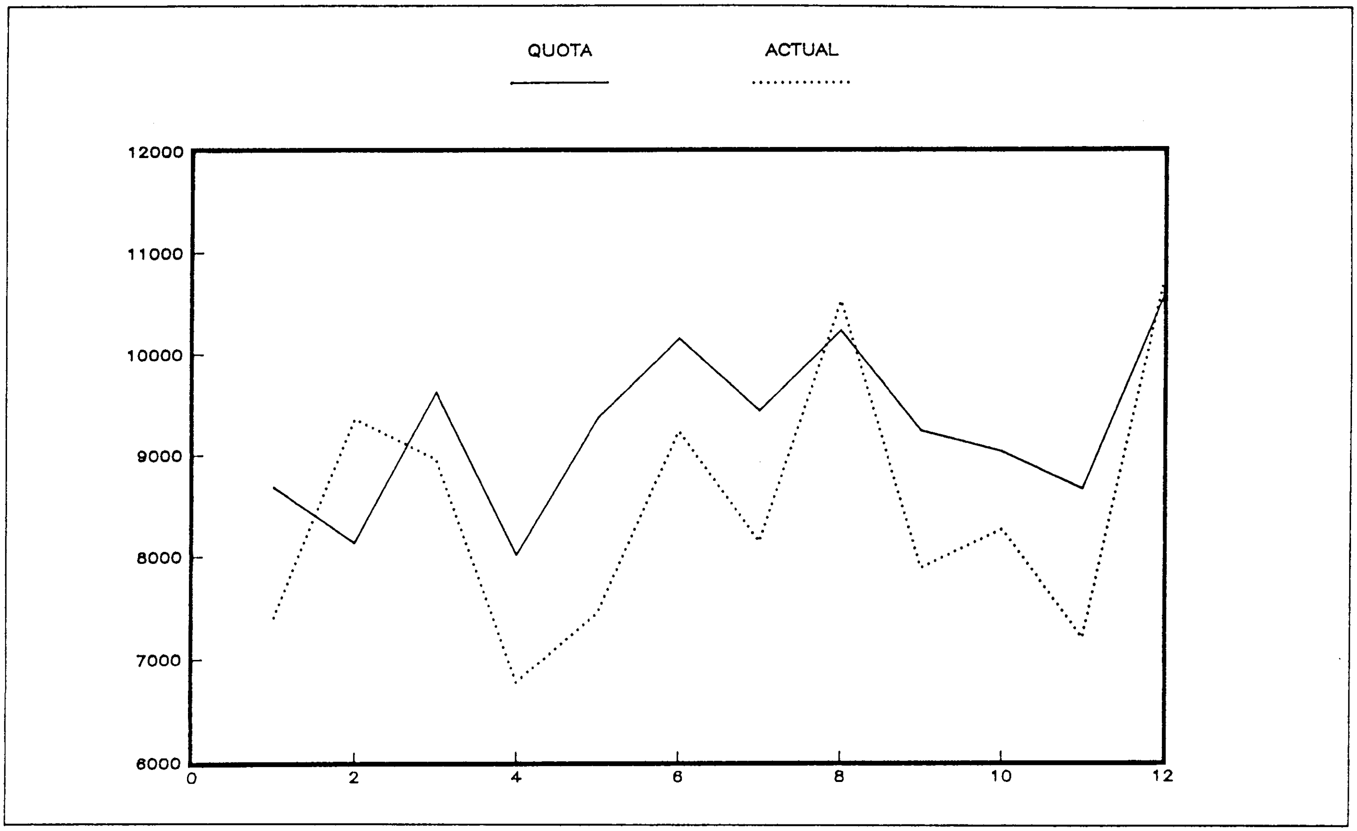


Figure 4-8. Basic Line Graph

A basic line chart like the one in Figure 4-8 can be designed by completing the shaded menus in Figure 4-7 (MAIN CONTROL, CHART DESIGN CONTROL, DATA DEFINITION and LINE CHART Menus, and DATA PROMPT if a data file does not already exist). Pressing DRAW at this point will produce:

-- a line chart on the graphics terminal, with every data point plotted
-- each dependent variable represented by a line of a different texture
-- each line texture has a legend labeled with the data variable name
-- axes are scaled according to the data and marked with labeled ticks
-- no data markers, no grid lines
-- no titles, footnotes, or annotations
-- stick font used for all text on charts

To create a scattergram, you must specify the scattergram option on the LINE CHART Menu. Otherwise, the scattergram is designed in the same way as the line chart.

Line Chart Enhancements

You can add any of the following enhancements to the basic line graph. The specifications in Section V give information about each particular menu.

To redirect output to a plotter or Laser printer, enter a *G* on the CHART DESIGN CONTROL Menu and fill out the GRAPHING OPTIONS Menu.

To select a subset of the data, enter a *C* on the CHART DESIGN CONTROL Menu and enter a subset expression on the LINE CHART Menu.

To qualify the data, press NEXT from the LINE CHART Menu and enter a qualification expression. (See Section III for information on subsets and qualifications.)

To create new data mathematically, enter a *T* on the CHART DESIGN CONTROL Menu and enter a transformation expression on the DATA TRANSFORMATIONS Menu.

To use logarithmic scaling or otherwise modify the axes in terms of minimum and maximum, major and minor ticks, tick labels and grid lines, enter an *X* on the CHART DESIGN CONTROL Menu and fill in the LINE CHART AXES Menu.

To provide different labels for the major tick marks, enter a *P* (Prompt) next to Labels on the LINE CHART AXES Menu, press ENTER and NEXT, and fill in the LABEL PROMPT Menu.

To specify different textures or legends, or to add data markers or colors, enter an *L* on the CHART DESIGN CONTROL Menu or press NEXT on the LINE CHART AXES Menu (or LABEL PROMPT Menu if used) and fill in the LINE CHART LEGENDS Menu.

To specify titles, footnotes, or axes titles, enter *TI* on the CHART DESIGN CONTROL Menu or press NEXT from the LINE CHART LEGENDS Menu and fill in the TITLES Menu.

To specify color, size and font for all standard text, enter *TC* on the CHART DESIGN CONTROL Menu and fill out the BAR/LINE TEXT CONTROL Menu. (To map the four available fonts to the font numbers, enter *FD* on the CHART DESIGN CONTROL Menu and fill out the FONT DEFINITION Menu. You then specify software text on the GRAPHING OPTIONS Menu.)

To annotate the chart, enter *A* on the CHART DESIGN CONTROL Menu and fill in the ANNOTATIONS Menu.

Designing A Bar Chart

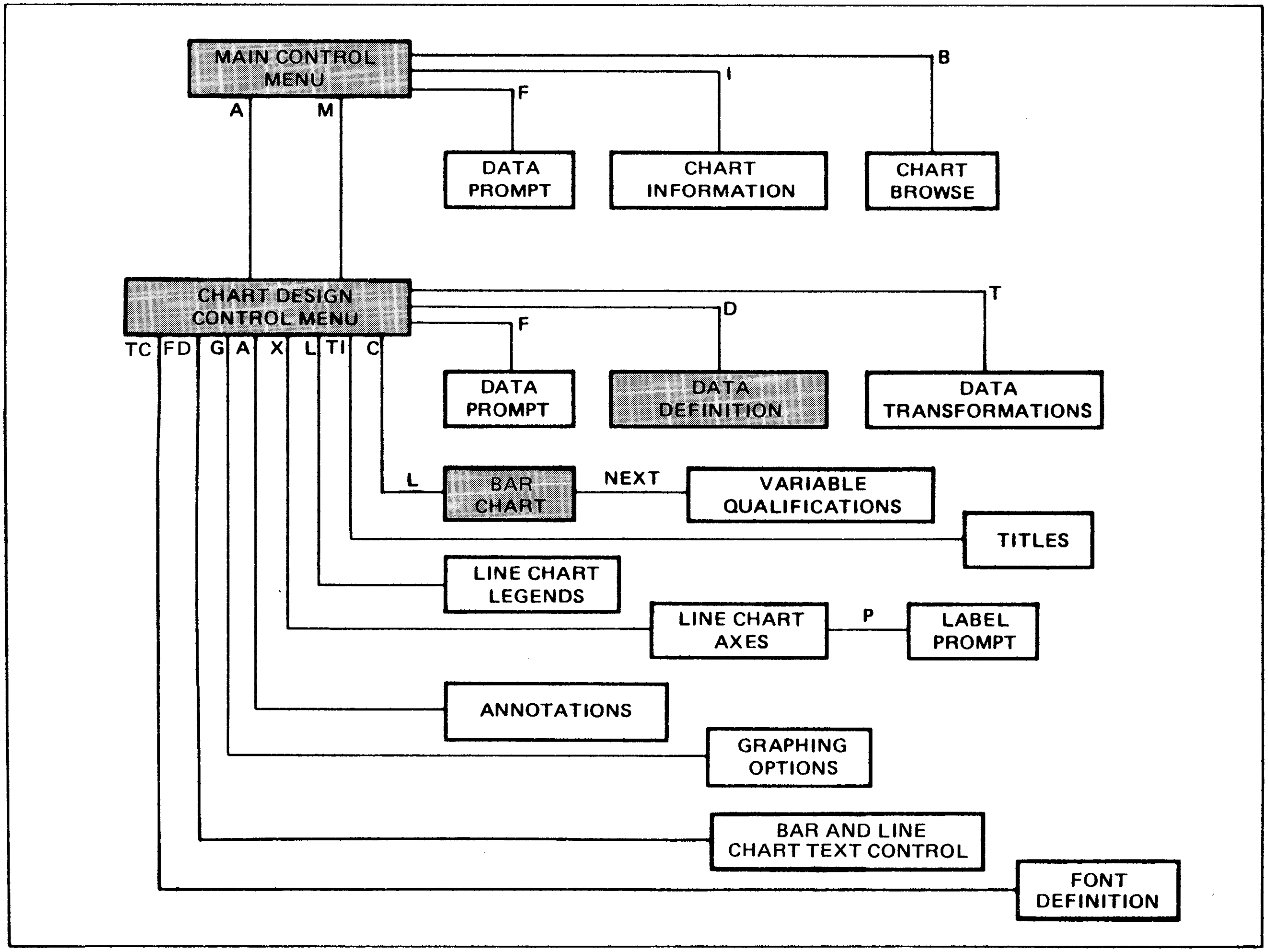


Figure 4-9. Bar Chart Road Map (required menus are shaded)

A bar chart can have up to 8 Y-axis variables, each represented by a different bar texture and containing numeric data. There is only one X-axis variable which can be either numeric or textual.

Figure 4-9 shows the routing of the menus used in designing a bar graph from the CHART DESIGN CONTROL Menu. Required menus are shaded.

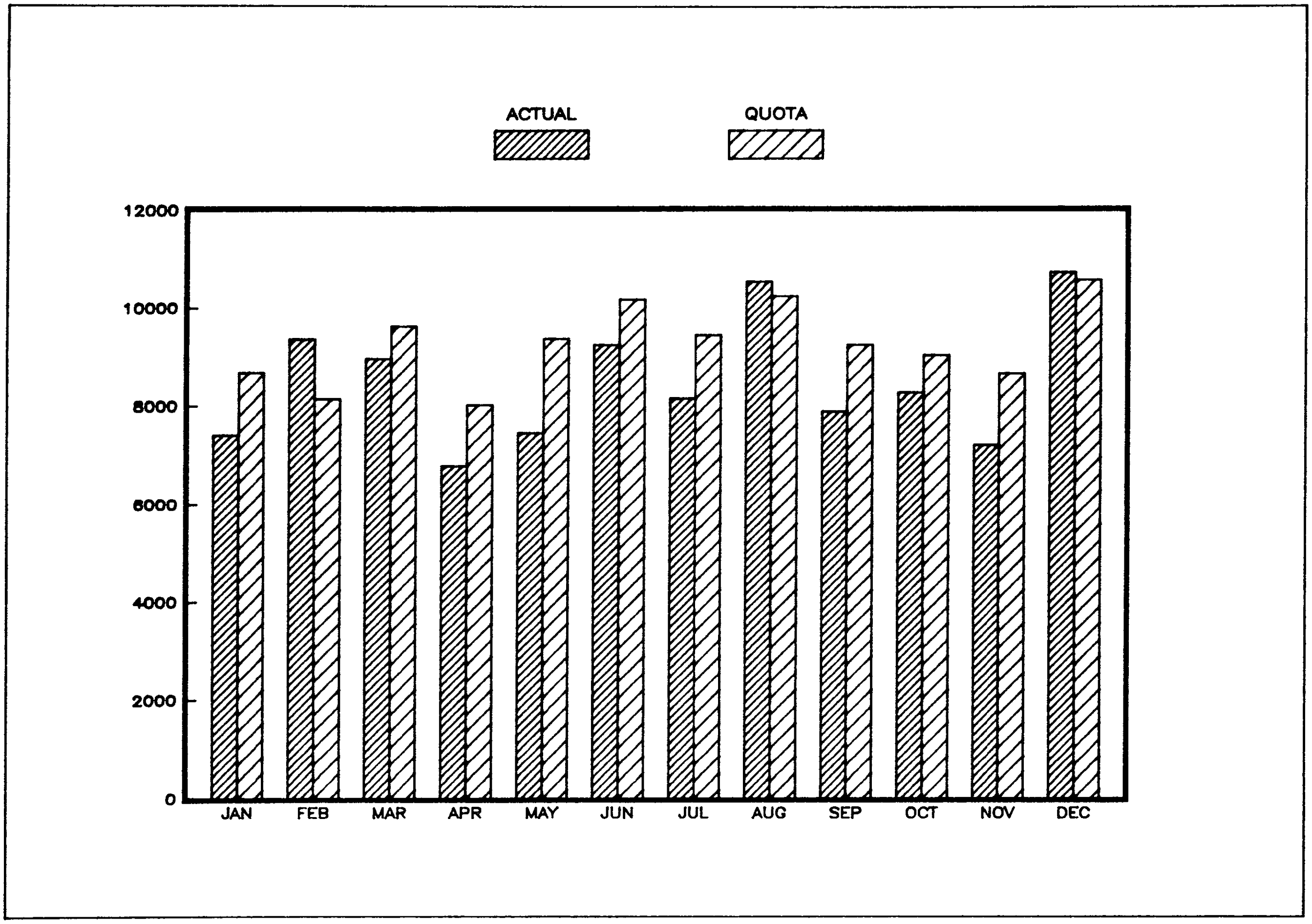


Figure 4-10. Basic Bar Graph

A basic bar chart can be produced by completing the shaded menus in Figure 4-9 (MAIN CONTROL, CHART DESIGN CONTROL, DATA DEFINITION and BAR CHART Menus, and DATA PROMPT if a data file does not already exist). Pressing DRAW at this point will produce:

-- a bar chart on the graphics terminal with stacked, vertical bars
-- y variables represented by bars of different textures
-- each texture has a legend labeled with the variable name
-- Y-axis with labeled ticks
-- the independent variable is labeled
-- axes are scaled according to the data
-- no grid lines
-- no titles, footnotes, or annotations

Bar Chart Enhancements

You can add any of the following enhancements to the basic bar graph. The specifications in Section V give information about each particular menu.

To redirect output to a plotter or Laser printer, enter a *G* on the CHART DESIGN CONTROL Menu and fill out the GRAPHING OPTIONS Menu.

To subset the data, enter a *C* on the CHART DESIGN CONTROL Menu and enter a subset expression on the BAR CHART Menu.

To qualify the data, press NEXT from the BAR CHART Menu and enter a qualification expression. (See Section III for information on subsets and qualifications.)

To create new data mathematically, enter a *T* on the CHART DESIGN CONTROL Menu and enter a transformation expression on the DATA TRANSFORMATIONS Menu.

To use logarithmic scaling or otherwise modify the axes in terms of minimum and maximum, major and minor ticks, tick labels, number of bars, bar labels, and grid lines, enter an *X* on the CHART DESIGN CONTROL Menu and fill in the BAR CHART AXES Menu.

To provide different labels for the major tick marks, enter a *P* (Prompt) next to Labels on the BAR CHART AXES Menu and Press ENTER; then press NEXT and fill in the LABEL PROMPT Menu.

To specify different textures or legends, or to add colors, enter an *L* on the CHART DESIGN CONTROL Menu or press NEXT from the BAR CHART LEGENDS Menu and fill in the TITLES Menu.

To specify main title, subtitle, axis titles or footnote, enter *TI* on the CHART DESIGN CONTROL Menu or press NEXT from the BAR CHART LEGENDS Menu and fill in the TITLES Menu

To specify color, size, and font for all standard text enter *TC* on the CHART DESIGN CONTROL Menu and fill out the BAR/LINE TEXT CONTROL Menu. (To map the four available fonts to the font numbers, enter FD on the CHART DESIGN CONTROL Menu and fill out the FONT DEFINITION Menu and specify software text on the GRAPHING OPTIONS Menu.)

To annotate the chart, enter *A* on the CHART DESIGN CONTROL Menu and fill in the ANNOTATIONS Menu.

Designing A Slide

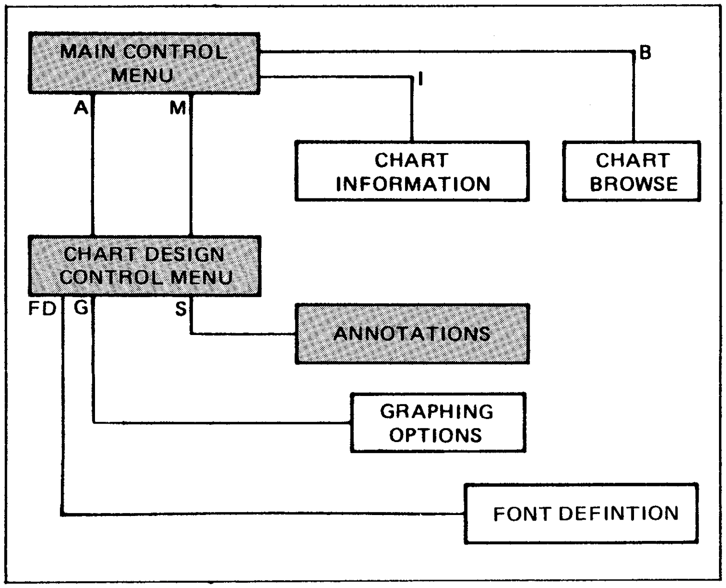


Figure 4-11. Slide Road Map (required menus are shaded)

GRAPH allows you to design a slide consisting of text, boxes, arrows, and lines without plotting variables from a data file. As with other graphs, the output can be directed to a plotting device that draws on paper or transparencies. Figure 4-12 shows the routing of the menus used to design a slide.

A slide is designed as a series of annotations. To reach the ANNOTATIONS Menu, enter a *C* for Chart Type on the CHART DESIGN CONTROL Menu (an *S* for Slide, the default chart type, is already printed on the Menu). Then enter each element of the slide as a separate annotation. (See the ANNOTATIONS Menu in Section V.)

The annotations are drawn on your graphics terminal by default. To draw the slide on a graphics plotter, fill in the GRAPHING OPTIONS Menu appropriately.

Note: the maximum number of annotations per chart is 20.

Utilities

All of the GRAPH Utilities can be executed from the MAIN CONTROL Menu. After filling the appropriate fields, press ENTER.

Drawing a Chart or Chart File

You can draw all of the charts in a chart file by entering a *D* under OPERATION along with the chart file name. GRAPH pauses after drawing each chart. When you press CONTINUE (f2), GRAPH draws the next chart. If you want the charts drawn without pauses, type any character in the first option field under DRAW.

Alternatively, you can draw a single chart by entering a chart name as well as the chart file name on the MAIN CONTROL Menu.

You can override the device that was specified for each chart when it was designed by typing in a device model number and HP-IB address in the option fields under DRAW. Device model numbers and HP-IB addresses are given in Section V under the GRAPHING OPTIONS Menu.

Copying a Chart

Before you start modifying an existing chart, you might want to make a copy of it to save in case you want to refer to the original again. GRAPH allows you to copy a chart into the same chart file or into a different one in your current group by entering a *C* under OPERATIONS along with the name of the new copy of the chart and its chart file. Enter the names of the original chart and chart file in the fields under COPY CHART.

Purging a Chart

To purge a chart, enter a *P* under OPERATIONS along with the name of the chart to be purged and the file in which it resides. When you press ENTER, GRAPH displays a warning message. You must press ENTER a second time to purge the chart.

Browsing a Chart File

If you enter a *B* under OPERATIONS along with a chart file name, GRAPH returns information about all of the charts in the chart file. The information includes chart name, chart type, and comments that may have been entered in the CHART INFORMATION Menu. The CHART BROWSE Menu can be used as an index to the chart file.

Supplying and Displaying Chart Information

If you enter an *I* under OPERATION along with the names of the chart and chart file, a page of information is returned, including chart name, chart type, creation date, creator's logon name and account, and comments. Only the comments field can be modified. (Remember to press ENTER after typing in comments.) These comments will also be displayed when you browse the chart file.

Creating Figure Files

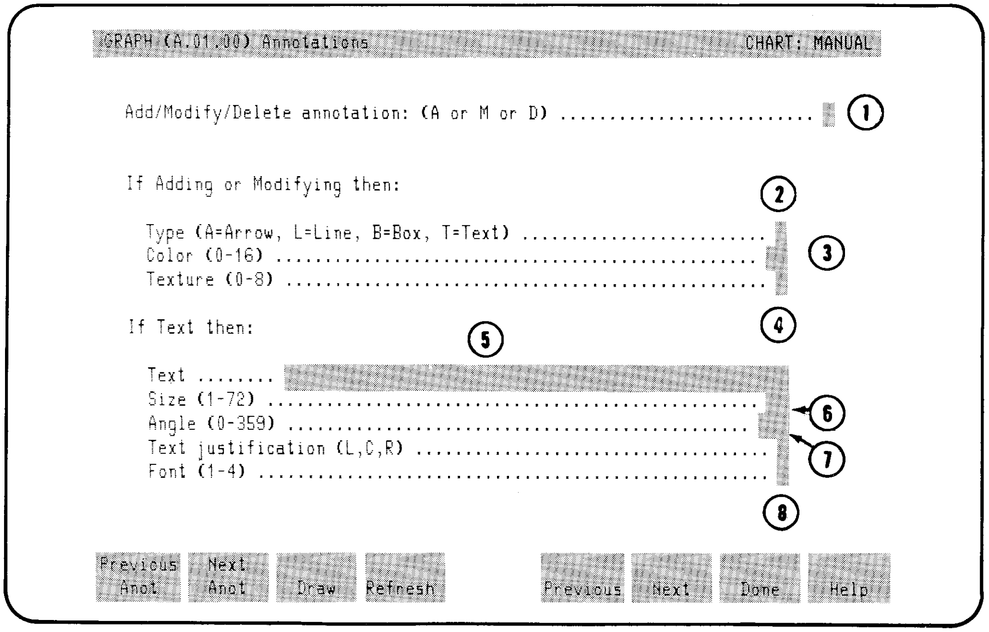
If you enter *CF*, the FIGURE FILE Menu will appear on your screen. Figure files are your means of longterm storage of graphical information. They are used primarily for transporting graphical information from one office systems application to another. See Section 1 for a complete description of additional features offered by this utility.

SECTION 5

Menu Specification

This section contains reference specifications for all of the GRAPH screen menus. For general discussions on using the menus, read Section IV, *Using DSG/3000 Interactively (GRAPH)*.

Annotations Menu



```
GRAPH (A.01.00) Annotations                                    CHART: MANUAL

  Add/Modify/Delete annotation: (A or M or D) .........................

  If Adding or Modifying then:

    Type (A=Arrow, L=Line, B=Box, T=Text) ........................
    Color (0-16) ..............................................
    Texture (0-8) ..............................................

  If Text then:

    Text ........
    Size (1-72) ..............................................
    Angle (0-359) ...........................................
    Text justification (L,C,R) ..................................
    Font (1-4) ..............................................
```

Figure 5-1. ANNOTATIONS Menu

Use this menu to specify any of the following annotations: boxes, lines, arrows, text. The plotting device must be a graphics terminal or plotter in order to position the annotations.

FIELD EXPLANATION

1. Function
   Enter one of the following letters:

   A To add a new annotation
   M To modify an existing annotation
   D To delete an existing annotation

2. Type (for adding only)
   A Arrow
   L Line
   B Box
   T Text containing any combination of printable characters

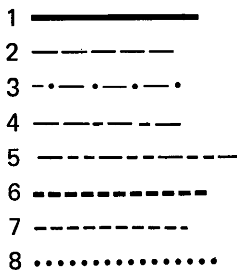
3. Color
   This field has meaning only when the graph is drawn on a color device. Enter one of the following numbers:

   1. Black (default)
   2. Red
   3. Green
   4. Blue
   5...16

   This presumes that the pens 1-4 are inserted in the plotter in this order. (See Appendix A for information about colors.)

4. Texture
   This field is for arrows, lines, and boxes. Enter one of the following numbers:

   For arrows, lines, and boxes:



5. Text (textual annotations only)
   Enter the annotation as you want it printed. It can consist of any combination of up to 51 printable characters.

6. Size (textual annotations only)
   Enter a number from 1 to n where 1 is the smallest text size. The default is 3.

   On plotters, software and hardware characters will be the same size. On terminals, hardware characters are larger than the software characters. Refer to the GRAPHING OPTIONS Menu for a discussion of hardware and software characters.

7. Angle (textual annotations only)
   Enter a number from 0 to 359 that represents the angle counterclockwise from the origin along which the text is to be printed. The default is 0. See Figure 5-2.

   If the graph is drawn using hardware characters, specified in the GRAPHING OPTIONS Menu, the annotations can only be drawn at 0, 90, 180, or 270 degrees on terminals. Plotters will draw at any angle.

8. Justification (textual annotations only)
   Enter one of the letters listed below:

L The text will be drawn with the left-most character located at the digitized font.

C The text will be drawn with the midpoint of the the text located at the digitized point.

R The text will be drawn with the right-most character located at the digitized point.

Note: The default is L.

9. Font (textual annotations only) Enter a number, 1-4, corresponding to a font number specified in the FONT DEFINITION Menu. The default (font 1) is the Stick font.

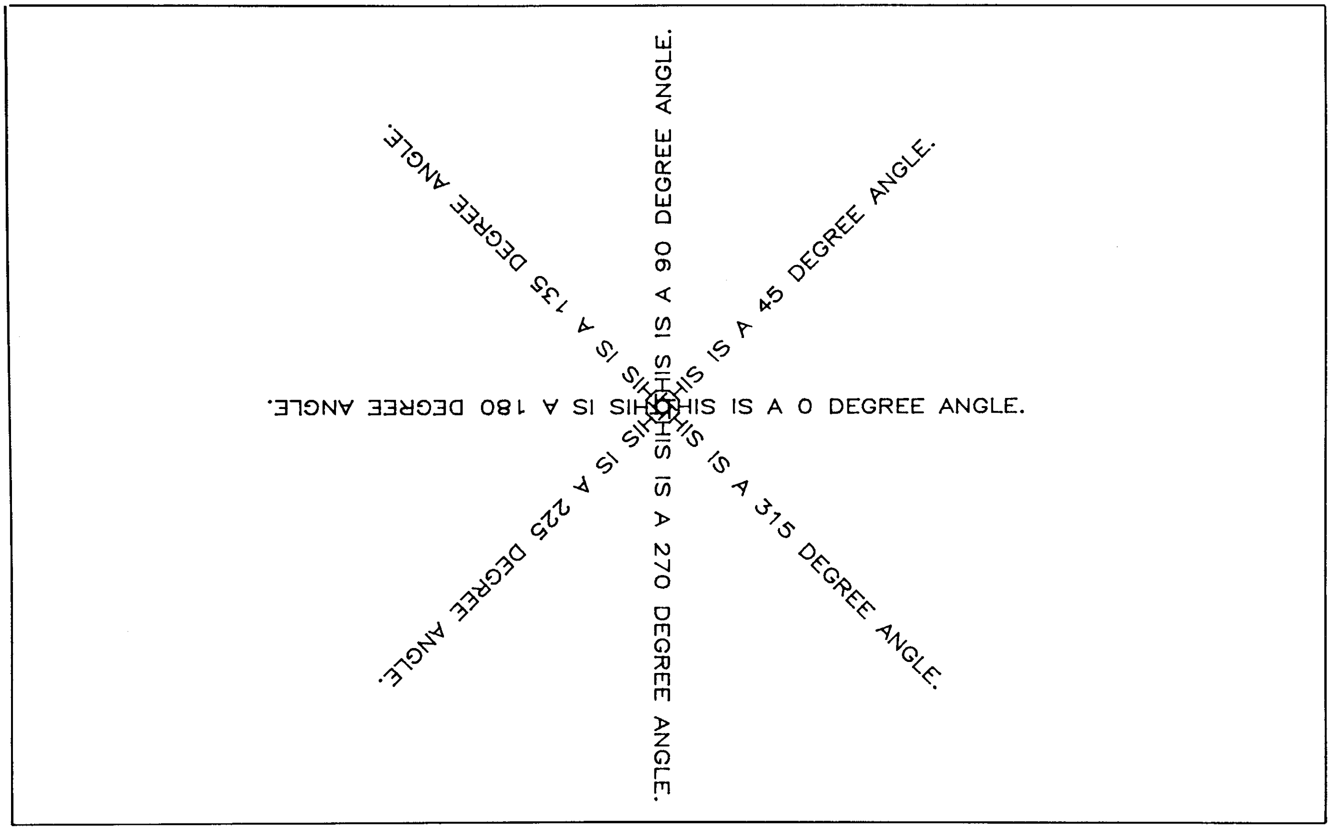

THIS IS A 0 DEGREE ANGLE.
THIS IS A 45 DEGREE ANGLE.
THIS IS A 90 DEGREE ANGLE.
THIS IS A 135 DEGREE ANGLE.
THIS IS A 180 DEGREE ANGLE.
THIS IS A 225 DEGREE ANGLE.
THIS IS A 270 DEGREE ANGLE.
THIS IS A 315 DEGREE ANGLE.

Figure 5-2. Text Angles

DISCUSSION

Each chart or slide can contain up to 20 annotations.

Adding Annotations

To add an annotation on the graphics terminal, follow these steps:

1. Type an *A* in the first field of the menu.
2. Fill in the second field according to the type of annotation being added.
3. If the annotation is textual, fill in field 5. All the other fields are optional.
4. Press ENTER. GRAPH returns a message in the message window telling you to position the graphics cursor to the start location for this annotation (see Table 5-1).
5. Move the graphics cursor, originally positioned at the end position of the most recent annotation, to the screen, to the start position by using the Graphics Cursor Keys (labeled with arrows on your terminal).
6. Press the return key to digitize the point.
7. For arrows, boxes, and lines, you must position the graphics cursor to the end location and press the space bar again.

The annotation is then drawn on your screen. If you wish to add another annotation, you can repeat the entire process.

If a graphics plotter is being used to draw the chart, follow the same procedures with one exception. To position the annotation, move the pen holder on the plotter to the desired point and press the ENTER key on the plotter while the light on the ENTER button is turned on.

Table 5-1. Start and End Locations for Annotations

Annotation	Start	End
Arrow	Tail End	Arrow Head
Box	One corner	Diagonal corner
Line	Either end	The other end
Text:		
Left justified	First character of text	
Right justified	Last character of text	
Centered	Center of text	

Modifying Annotations

To modify an annotation on the graphics terminal, follow these steps:

1. If the annotations are not already displayed, draw the chart by pressing f3 (DRAW).

2. Press PRV ANOT (f1) to position the graphics cursor to the most recent annotation.

3. If the current annotation is not the one you want to modify, press PRV ANOT (f1) again to step backwards through all the annotations until you reach the one you want.

   Notice that pressing NXT ANOT (f2) brings you forward through the same sequence. If you have modified an annotation -- for example 8 -- and you wish to modify 9, pressing f2 will bring to the screen the following error message, "Next annotation not available." To modify 9, it is necessary to step backwards (using F1) through annotations until you reach 9 again. This occurs because modifying an annotation moves it to the end of the list.

   If the menu is not displayed, turn on the alphanumeric display by pressing SHIFT-A DSP.

5. Type an *M* in the first field of the ANNOTATIONS Menu.

6. If you are changing anything other than the position of the annotation, type in your corrections on the menu.

7. Press ENTER.

8. Follow the positioning procedure used in adding annotations to relocate the current annotation if desired.

GRAPH will draw the modified annotation on your screen. Notice that the former version is not erased, but if you draw the chart again only the modified version appears.

If a graphics plotter is being used to draw the chart, follow the same procedures with one exception. To position the annotation, move the pen holder on the plotter to the desired point and press the ENTER key on the plotter when the light on the ENTER button is turned on. Notice that when you press f1 or f2 on the terminal, the pen holder positions itself to the current annotation on the plotter.

Deleting Annotations

To delete an annotation, follow steps 1-4 above for modifying. Then enter a *D* in the first field of the ANNOTATIONS Menu and press ENTER.

FUNCTION KEYS

PRV ANOT (f1) If the plotting device is a graphics terminal, f1 first turns on the graphics cursor, positions it to the most recent annotation and displays its specifications on the menu.

Thereafter, pressing f1 positions the graphics cursor to the previous annotation and displays its specifications. If the plotting device is a plotter, it positions the pen holder instead of the cursor.

NXT ANOT (f2) If the plotting device is a graphics terminal, f2 positions the graphics cursor to the next annotation and displays its specifications. If the plotting device is a plotter, the pen holder is positioned instead of the cursor.

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing the menu with the specifications of the current annotation. Also clears the graphics display.

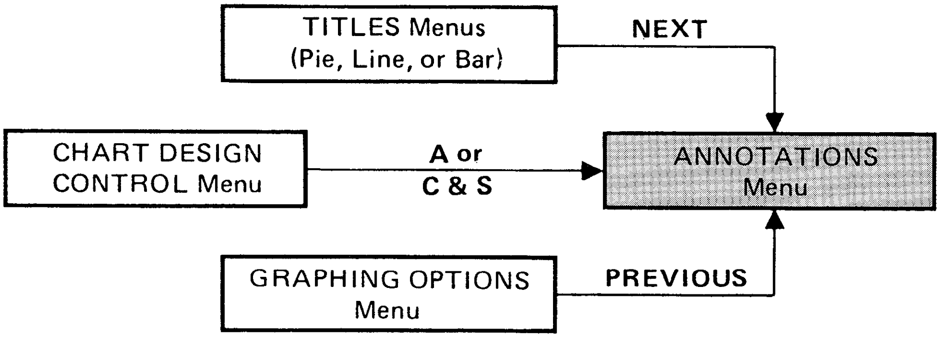
PREVIOUS (f5) Goes to PIE CHART TITLES Menu if the chart type is a pie chart, or goes to BAR AND LINE CHART TITLES Menu if the chart type is a line or bar chart. If you are designing a slide, it goes to the CHART DESIGN CONTROL Menu.

NEXT (f6) Goes to GRAPHING OPTIONS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

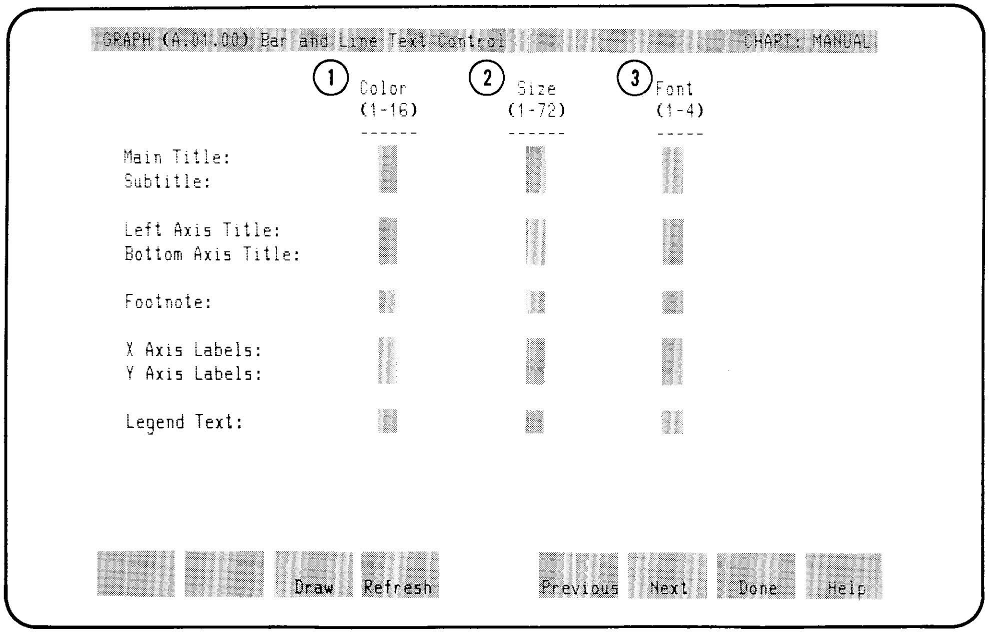
HELP (f8) Initiates the Help Facility, giving an explanation of the ANNOTATIONS Menu.

ROADMAP



When creating a slide and not plotting data, enter *C* (Select Chart Type) and *S* (Slide) on the CHART DESIGN CONTROL Menu to get to this menu. If you want to annotate a graph, enter *A* on the CHART DESIGN CONTROL Menu, or press NEXT from one of the TITLES menus or press PREVIOUS from the GRAPHING OPTIONS Menu.

Bar and Line Chart Text Control



GRAPH (A.01.00) Bar and Line Text Control CHART: MANUAL

	Color (1-16)	Size (1-72)	Font (1-4)
Main Title:	▨	▨	▨
Subtitle:	▨	▨	▨
Left Axis Title:	▨	▨	▨
Bottom Axis Title:	▨	▨	▨
Footnote:	▨	▨	▨
X Axis Labels:	▨	▨	▨
Y Axis Labels:	▨	▨	▨
Legend Text:	▨	▨	▨

Figure 5-3. BAR AND LINE CHART TEXT CONTROL Menu

Use this menu to specify color, size, and font for text on a bar or line chart. This includes titles, subtitles, axis labels, legend labels, and footnotes but not annotations.

FIELD EXPLANATION

1. Color
   This field has meaning only when the graph is drawn on a color device. Enter 0-16 or blank. (Entering 0=1.)

1	Black (default)	5...
2	Red	6...
3	Green	7...
4	Blue	8...16

The above presumes that pens 1-4 are inserted in the plotter in the order shown. (See Appendix A for information about colors.) The maximum number that can be entered is 16. The default (blank) is 1.

2. Size
   Enter a number from 1 to 74 where 1 is the smallest text size and 72 is the largest. The defaults (blanks) use the following sizes:

Main Title	8
Subtitles	6
Left Axis Title	5
Bottom Axis Title	5
X Axis Labels	4
Y Axis Labels	4
Legend Text	5

   On plotters, software and hardware characters will be the same size. On terminals, hardware characters may be larger than the software characters. Refer to GRAPHING OPTIONS Menu for a discussion of hardware and software characters.

   Note: SIZE of text will reflect chart size changes and SIZE is relative rather than absolute. If you have reason to shrink or stretch the size of a chart, you may wish to adjust the SIZE specifications for your text to ensure legibility.

3. Font
   Enter a number from 1 to 72 you have defined for your chart using the FONT DEFINITION Menu. Four software fonts may be defined for a single chart. The default (blank) is font 1.

   Only one hardware font may be used. See GRAPHING OPTIONS for more information about software and hardware fonts.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing the menu with the specifications for color, size, and font as of the last ENTER.

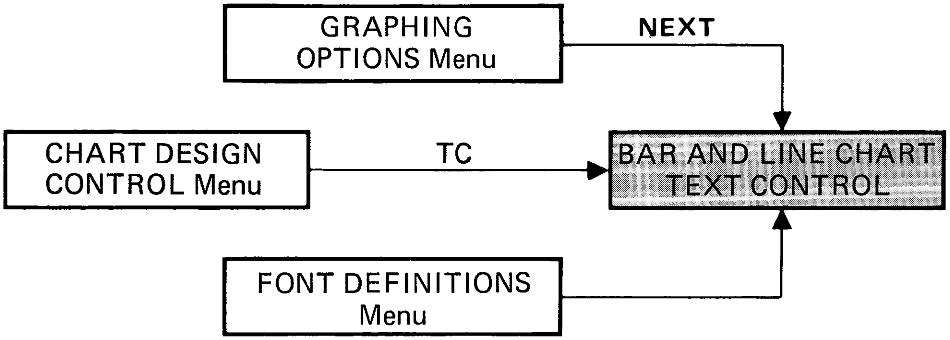
PREVIOUS (f5) Goes to GRAPHING OPTIONS Menu.

NEXT (f6) Goes to FONT DEFINITION Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

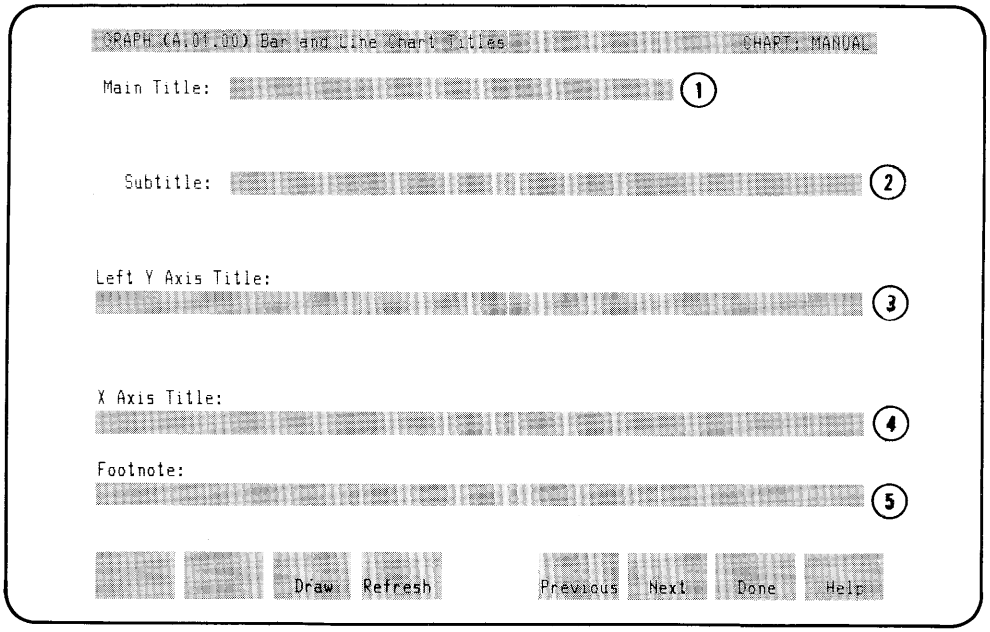
HELP (f8) Initiates Help Facility, giving an explanation of BAR AND LINE CHART TEXT CONTROL Menu.

ROADMAP



If you are designing a Bar and Line Chart and enter *TC* on the CHART DESIGN CONTROL Menu, this menu appears. Pressing PREVIOUS from FONT DEFINITION or NEXT from the GRAPHING OPTIONS menus also brings it up.

Bar and Line Chart Titles Menu



GRAPH (A.01.00) Bar and Line Chart Titles CHART: MANUAL

Main Title: ①

Subtitle: ②

Left Y Axis Title: ③

X Axis Title: ④

Footnote: ⑤

Figure 5-4. BAR AND LINE CHART TITLES Menu

You may specify title, subtitle, axes titles, and footnotes for line charts, bar charts, and scattergrams with this menu.

FIELD EXPLANATION

1. Title
   The title may be any combination of up to 45 printable characters.

2. Subtitle
   The subtitle is printed in a smaller text size than the title and contains up to 64 printable characters.

3. Left Vertical Axis Title
   This title may contain contain up to 78 printable characters, and will be justified with the left vertical axis when drawn.

4. Horizontal Axis Title
   This title may up to 78 printable characters, and will be centered beneath the horizontal axis when drawn.

5. Footnote
   The footnote may contain up to 78 printable characters.

DISCUSSION

The following features can be entered on this menu:

* title
* subtitle
* left vertical axis title
* horizontal axis title
* footnote

All sizes can be specified using the BAR AND LINE TEXT CONTROL Menu. If the sizes specifications are changed, the maximum number of printable characters listed above may be affected.

In every case, GRAPH will left-justify the text that you enter. When the titles are actually drawn on the graph, the left axis title is left-justified, and all other titles are centered.

Any of the titles entered in this menu can include any of the following special commands:

$TODAY Represents the current date in the form mm/dd/yy with leading zeros suppressed. For example, 2/15/82.

$MONTH Represents the current month spelled out in full. For example, FEBRUARY.

$M Represents the current month numerically. For example, 2.

$DAY Represents the current numeric day of the month. For example, 15.

$YEAR Represents the current year in 4 digits. For example, 1982.

$Y Represents the current year by its last two digits. For example, 82.

For example, you might specify this subtitle:

THIS GRAPH WAS PRODUCED ON $TODAY

When the graph is drawn, the subtitle would be:

THIS GRAPH WAS PRODUCED ON 2/15/82

The titles entered on this menu can be suppressed when the graph is actually drawn by filling in the GRAPHING OPTIONS Menu appropriately.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

PREVIOUS (f5) Goes to LINE CHART LEGENDS or BAR CHART LEGENDS Menu.

NEXT (f6) Goes to ANNOTATIONS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the BAR AND LINE CHART TITLES Menu.

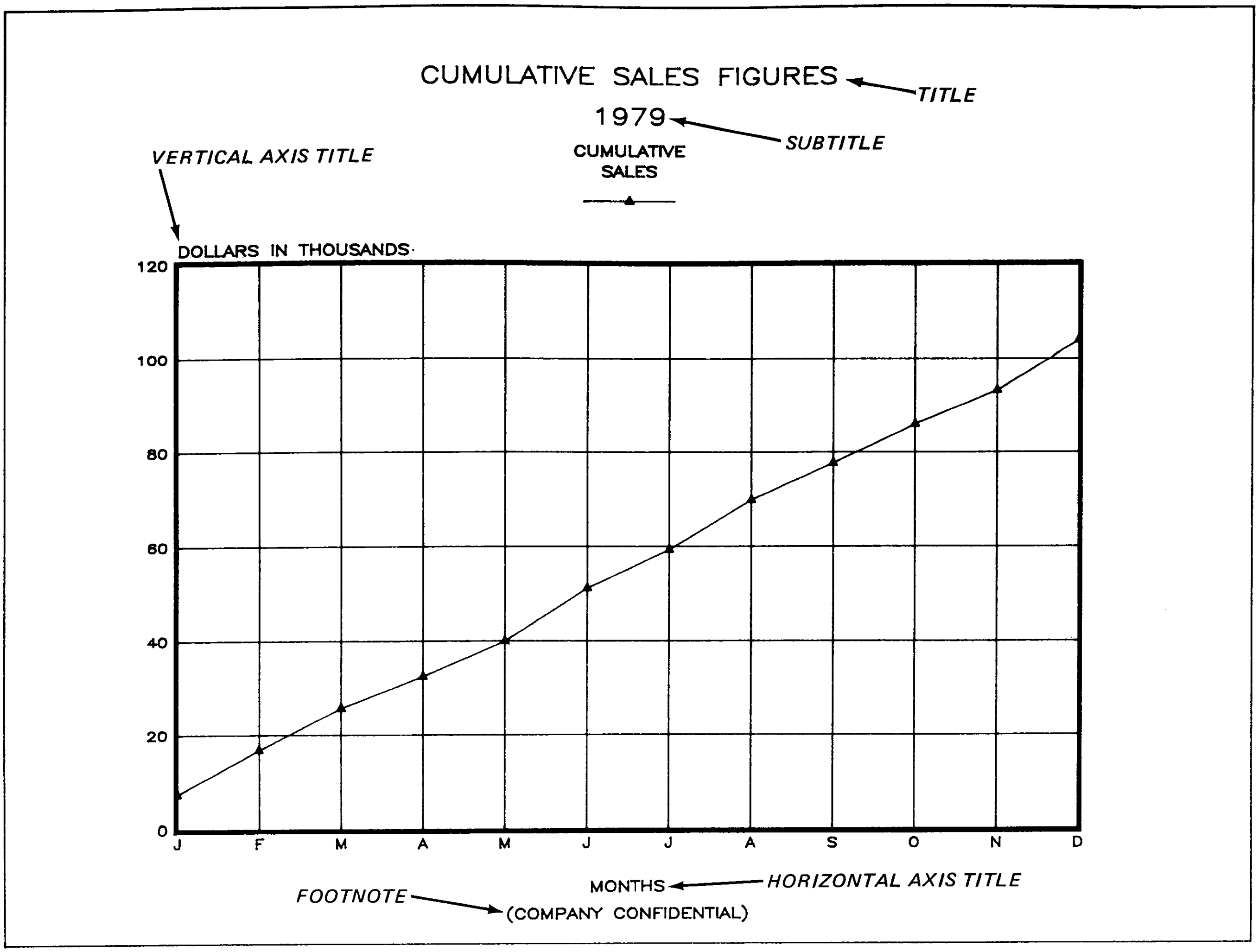
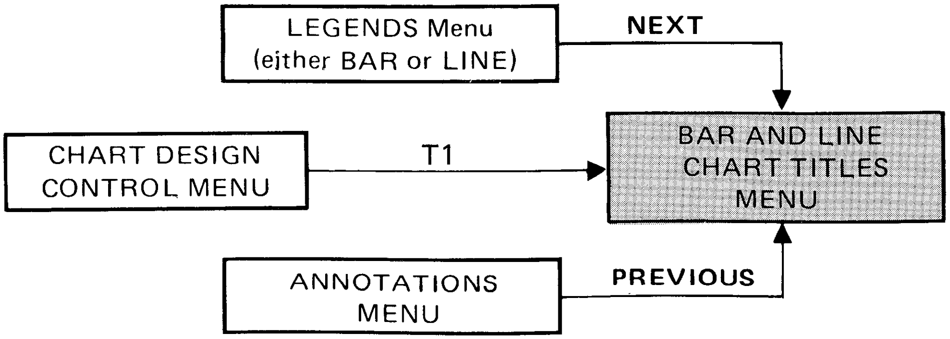


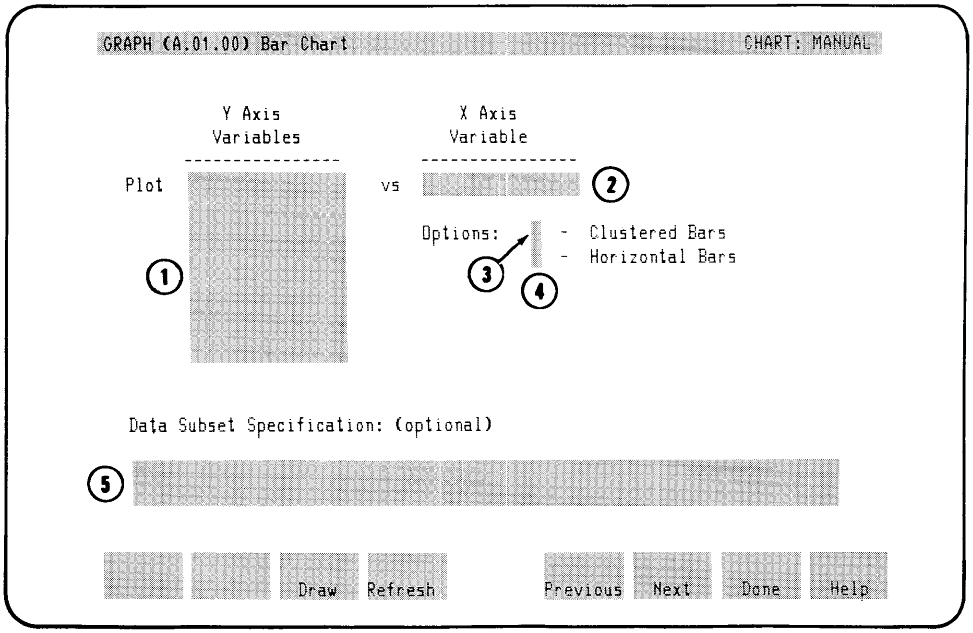
Figure 5-5. Standard Titles for Line and Bar Charts

ROADMAP



Pressing *T1* on the CHART DESIGN CONTROL Menu when you have specified that you are drawing a line or bar chart brings this menu to your screen. Alternatively, you can press NEXT from the BAR CHART LEGENDS or LINE CHART LEGENDS Menu when designing a line or bar chart or PREVIOUS from ANNOTATIONS.

Bar Chart Menu


GRAPH (A.01.00) Bar Chart CHART: MANUAL

Y Axis Variables

X Axis Variable

Plot vs

Options: - Clustered Bars - Horizontal Bars

Data Subset Specification: (optional)

Figure 5-6. BAR CHART Menu

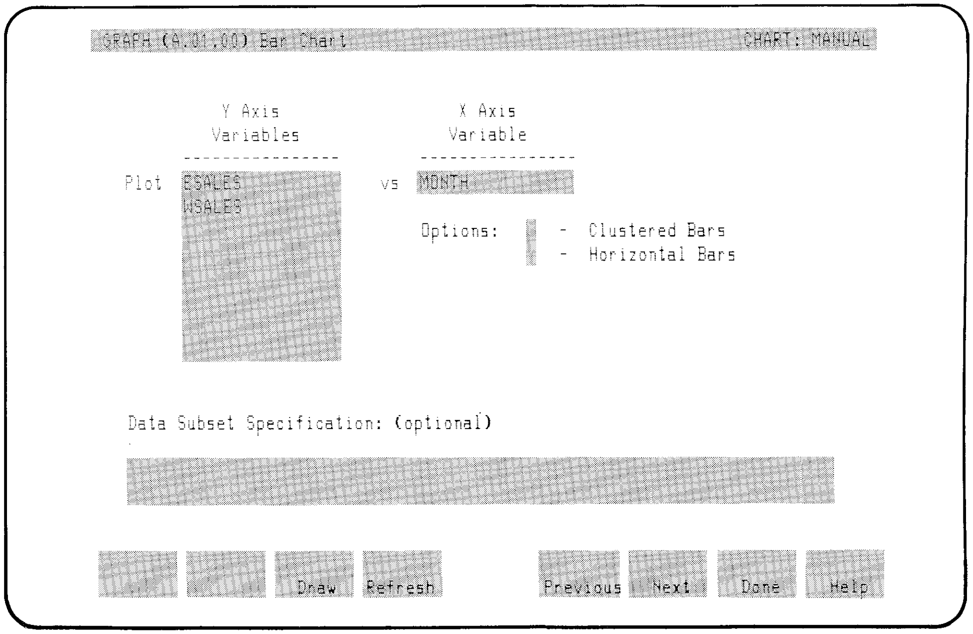
This menu names the variables to be plotted, selects a subset of the data, and specifies options such as horizontal or clustered bars.

FIELD EXPLANATION

1. Y Axis Variables (dependent)
   Fill in the names of the Y-axis variables as they were defined in the DATA DEFINITION Menu, or GRAPH returns a warning message. This is a required variable. You must specify at least one and no more than 8 variables in this field.

2. X Axis Variable (independent)
   Fill in the name of the X-axis variable that will be plotted along the horizontal axis. (If you choose to have horizontal bars, this variable will actually be plotted on the vertical axis.) This variable name must also be defined in the DATA DEFINITION Menu, or GRAPH returns a warning message.

   If you leave this field blank, the bars will be labeled with numbers starting with 1. The X- and Y-axes may also be defined on the DATA TRANSFORMATIONS Menu.



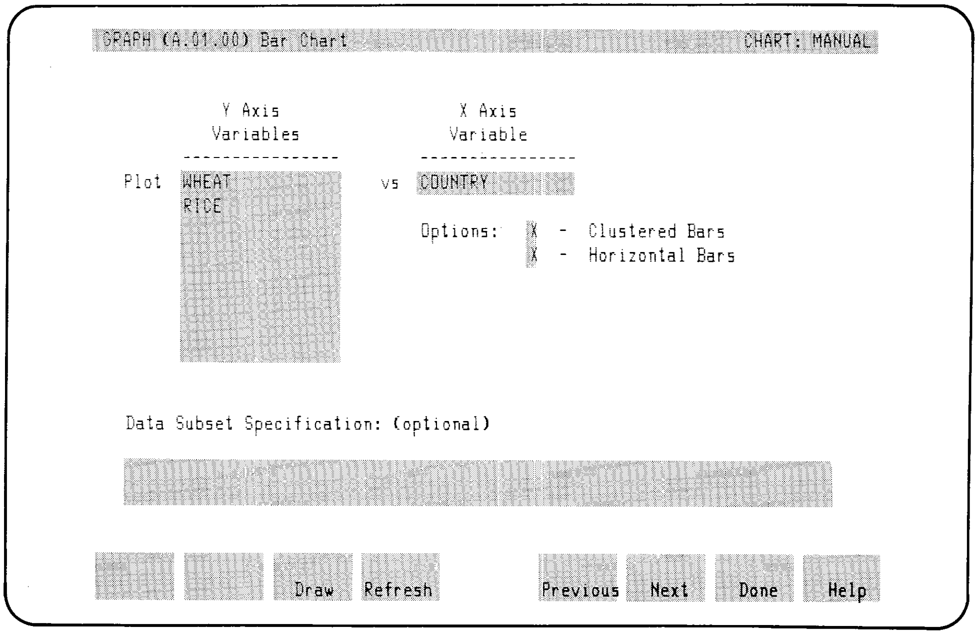
```
GRAPH (A.01.00) Bar Chart                               CHART: MANUAL

                 Y Axis               X Axis
                Variables            Variable
            ----------------     ----------------
    Plot    ESALES            vs MONTH
            WSALES
                                 Options:    -  Clustered Bars
                                             -  Horizontal Bars

    Data Subset Specification: (optional)
```

Figure 5-7. BAR CHART Menu Specifying Vertical Stacked Bars

3. Option: Clustered Bars
If you do not fill in this field, a bar graph plotting more than one Y-axis variable is drawn with stacked bars (see Figure 5-7). If you want the bars to be clustered, enter any character in this field (Figure 5-8).

Note that if the data is scaled logarithmically (see the BAR AXES Menu) and more than one axis variable is used, the bars may not be stacked.

4. Option: Horizontal Bars
If you do not fill in this field, bars are drawn vertically. If you enter any character in this field, the bars will be drawn horizontally; this means that the variables named in field 1 above are plotted on the horizontal axis and the variable named in field 2 above is plotted on the vertical axis.



```
GRAPH (A.01.00) Bar Chart                              CHART: MANUAL

              Y Axis              X Axis
             Variables           Variable
          ----------------    ----------------
   Plot   WHEAT            vs COUNTRY
          RICE
                              Options:  X  -  Clustered Bars
                                        X  -  Horizontal Bars

   Data Subset Specification: (optional)
```

Figure 5-8. BAR CHART Menu Specifying Horizontal Clustered Bars

5. Data Subset Specification
   If you want to graph a subset of the data file, enter a subset specification expression. In general, the expression has the form:

```
variablename relop value
 [ {AND} [NOT] variablename relop value [...]]
   {OR }
```

where *relop* is a relational operator from Table 5-2. For example,

```
DEPT=SALES OR EXPENSES>800
```

is a subset expression where DEPT and EXPENSES are names of variables in the data file. The variables used to specify the subset are not necessarily those being graphed.

Table 5-2. Relational Operators For Subset Expressions

Parentheses can be used to override the standard order of precedence:

1. Relational operators are processed from left to right
2. NOT processed from left to right
3. AND and OR processed from left to right

If you are stating the expression in terms of a textual variable whose *value* contains embedded blanks, use quotation marks or apostrophes around the value.

Use quotation marks around a value with an embedded apostrophe; likewise use apostrophes around a value with embedded quotation marks.

Examples

```
DEPT="R & D"
NAME= ` "X" '
POSITION= "BOSS'S SON"
```

DISCUSSION

Bar charts are drawn with bars running vertically unless you specify horizontal bars. When there is more than one dependent (Y-axis) variable, the bars are stacked unless you specify clustered bars. The Y axis variables all must contain numeric data and must be specified as numeric in the DATA DEFINITION Menu.

Each one of the bar variables is represented by a different bar texture. The single independent (or X-axis) variable (plotted on the horizontal axis if the bars are vertical) can be numeric, textual, or left blank. If it is left blank, the X-axis will be labeled 1,2,3, and so forth at each bar.

If the X-axis variable is time, the bars are usually drawn vertically. If the X-axis variable is not time, drawing the bars horizontally by designating the horizontal option will produce a clearer visual representation.

FUNCTION KEYS

DRAW (f3) Draws the current chart

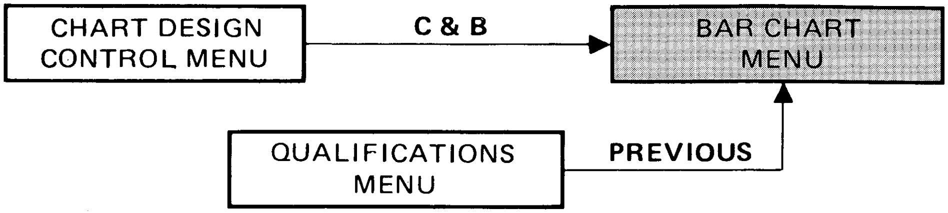
REFRESH (f4) Refreshes the screen by redrawing·it with the information entered as of the last ENTER.

PREVIOUS (f5) Goes to the CHART DESIGN CONTROL Menu
NEXT (f6) Goes to QUALIFICATIONS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

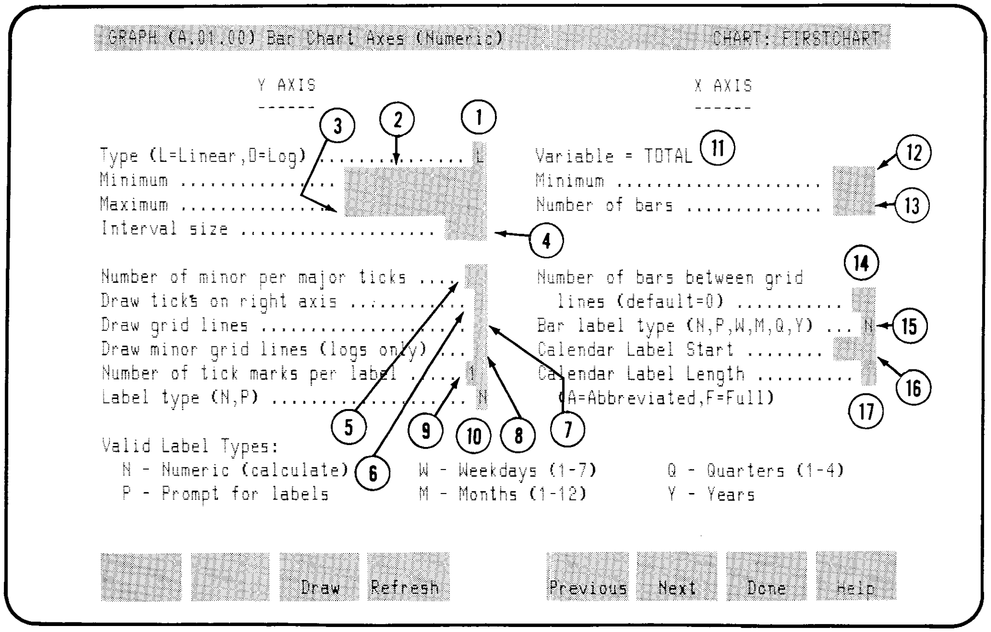
HELP (f8) Initiates Help Facility, giving an explanation of the BAR CHART Menu.

ROADMAP



This menu appears when you specify a Bar Chart from the CHART DESIGN CONTROL Menu. You can also reach it by pressing PREVIOUS from the QUALIFICATIONS Menu when designing a bar chart.

Bar Chart Axes Menu (Numeric)



```
GRAPH (A.01.00) Bar Chart Axes (Numeric)                    CHART: FIRSTCHART

                 Y AXIS                                      X AXIS
                 ------                                      ------

Type (L=Linear,O=Log) ............ L      Variable = TOTAL
Minimum ..............                    Minimum ....................
Maximum ..............                    Number of bars .............
Interval size ...................

Number of minor per major ticks ....      Number of bars between grid
Draw ticks on right axis ...........        lines (default=0) ..........
Draw grid lines ....................      Bar label type (N,P,W,M,Q,Y) ... N
Draw minor grid lines (logs only) ...     Calendar Label Start ........
Number of tick marks per label ..... 1    Calendar Label Length ..........
Label type (N,P) ................... N      (A=Abbreviated,F=Full)

Valid Label Types:
  N - Numeric (calculate)      W - Weekdays (1-7)        Q - Quarters (1-4)
  P - Prompt for labels        M - Months (1-12)         Y - Years
```

Figure 5-9. BAR CHART AXES (NUMERIC) Menu

Specifies the characteristics of both axes for a numeric bar chart, including: scaling, spacing, and labeling of ticks, and grid lines.

In the numeric bar chart, all variables have been specified as numeric on the BAR CHART Menu. If the X-axis variable is textual or is left blank, GRAPH presents the BAR AXES Menu (TEXTUAL) instead.

FIELD EXPLANATION

1. Type
   The *L* that is entered in this field by default represents linear scaling. If you want logarithmic scaling instead, enter an *O*. Logarithmic scaling may be useful if the data takes values over a very large range, or if you want to draw a linear plot of two variables that have an exponential relationship.

2. Minimum
   Enter the minimum value for the axes. If this field is left blank, DSG/3000 calculates the minimum from the data values.

3. Maximum
   Enter the maximum value for the axis. The maximum value must be greater than the minimum value. If you do not enter anything here, DSG/3000 calculates the maximum from the data values.

4. Interval Size
   Enter the number of units you want between major ticks in terms of the data. For example, if you want $100 between ticks, enter 100. If you do not fill this in, DSG/3000 calculates the units between ticks from the data in such a way that there are approximately 5 intervals along the axis. If Type is logarithmic (O), this field represents the number of powers of 10 which will separate major tick marks. For example, a value of one means that every power of 10 will have a tick mark.

5. Number of Minor Per Major Ticks
   Enter the number of minor tick marks between the major ticks. Minor ticks are drawn smaller than major ticks and are not labeled. If you do not fill in this field, there will be no minor ticks. For logarithmic charts, a common number of minor per major ticks is eight. Lowering this number results in a less dense axis or chart.

6. Draw Ticks on Right Axis
   If you want the Y-axis ticks to be drawn on the right axis as well as the left one, enter any character in this field. (If you opted for horizontal bars on the BAR CHART Menu, entering a character here causes the ticks to be drawn on the upper axis as well as the lower axis.)

7. Draw Grid Lines
   If you want grid lines to be drawn wherever there are major ticks, enter any character here.
8. Draw Minor Grid Lines(Log scaling only)
   This field is ignored unless you have specified:

   log scaling (field 1)
   grid lines (field 7)
   number of minor ticks is >0 (field 5)

   If these conditions are satisfied and you type any character in this field, grid lines are drawn at minor ticks as well as major ticks.

9. Number of Tick Marks per Label
   If you leave this field blank or enter a 1, every major tick will be labeled. Enter the number of major ticks per label. For example, if you want to label every other tick, enter a 2, which means that there will be 2 ticks per label.

10. Label Type
   If you want the major ticks to be labeled according to the axis scaling, enter an *N* in this field. This is the default. If you want to supply your own labels which might be textual rather than numeric, enter a *P*; after you have pressed ENTER for this menu, press NEXT and you can enter the labels on the LABEL PROMPT Menu.

11. X-Axis Variable (display only)
   The name of the independent variable is displayed here. It cannot be altered.

12. Minimum
   If you do not fill in this field, this axis is scaled according to the data. If you wish to leave more space before the first bar, enter a value less than the minimum value of the X-axis variable.

13. Number of Bars
If you do not fill in this field, DSG/3000 draws one bar for each data point. If you specify that you want a greater number of bars than there are data points, DSG/3000 leaves blank space for the remaining data values. If you specify fewer bars than there are data points, DSG/3000 draws only as many bars as you request, starting from the beginning of the data file. For clustered bars, 1 bar equals one cluster of bars.

14. Number of Bars Between Grid Lines
If you do not fill in this field, no grid lines are drawn parallel to the bars. If you want grid lines to be drawn, enter the number of bars between grid lines. For clustered bars, each cluster counts as one bar.

15. Bar Label Type
Enter one of the following:

N DSG/3000 calculates numeric labels from the axis scaling. (Default)

P Use this option if you want to enter your own bar labels which might be textual rather than numeric. After you have pressed ENTER and then NEXT, GRAPH issues the LABEL PROMPT Menu. If you entered a *P* for both axes, you will fill in the LABEL PROMPT Menu twice.

For convenience, GRAPH allows you to specify the following time or calendar labels regardless of the values of the X-axis variable. Fields 16 and 17 are used with calendar labels to give the starting point and length of the labels.

W Weekdays. Each bar will be labeled with a day of the week. If there are more than 7 bars, the labels will be repeated.

M Months. Each bar will be labeled with the name of a month. If there are more than 12 bars, the labels will be repeated.

Q Quarters. Each bar will be labeled with a quarter. If there are more than 4 bars, the labels will be repeated.

Y Year. Each bar will be labeled with a year.

16. Calendar Label Start (used only with time labels)
Enter a number corresponding to the weekday, month, quarter, or year that labels the first (leftmost) bar. The default is 1, which corresponds to MON for weekdays, JAN for months, QTR1 for quarters, or 1 for year. For example, if you want the bars to be labeled by month beginning with November, type *11* in this field. If you want the bars to be labeled by years beginning with 1982, type *1982* in this field.

17. Calendar Label Length (used for time labels only)

A Abbreviated label form is printed on the graph. (default)

L Long form is printed on the graph.

	Abbreviated	Long
Weekdays	M	MON
Months	J	JAN
Quarters	Q1	QTR1
Year	N/A	1982

DISCUSSION

The X-axis is labeled with the name of the X-axis variable.

If you do not fill in this menu, the axes characteristics take the following defaults:

* linear axes
* axes scaled according to the data values
* major ticks with numeric labels calculated from the data
* no minor ticks or grid lines

Note that calendar labels are blindly applied to tick marks, irrespective of the actual data. To avoid mislabeling, the X-axis interval size should be set to 1 in most cases.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

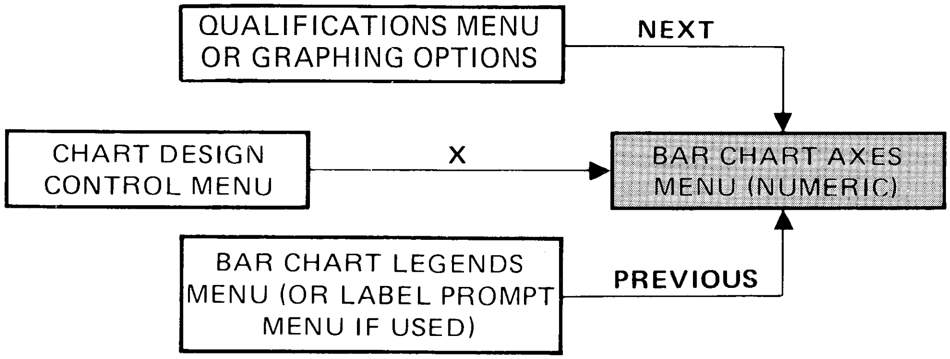
PREVIOUS (f5) Goes to the QUALIFICATIONS Menu.

NEXT (f6) If a *P* (Prompt) is entered in field 10 or 15, goes to LABEL PROMPT Menu. Otherwise, goes to BAR CHART LEGENDS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

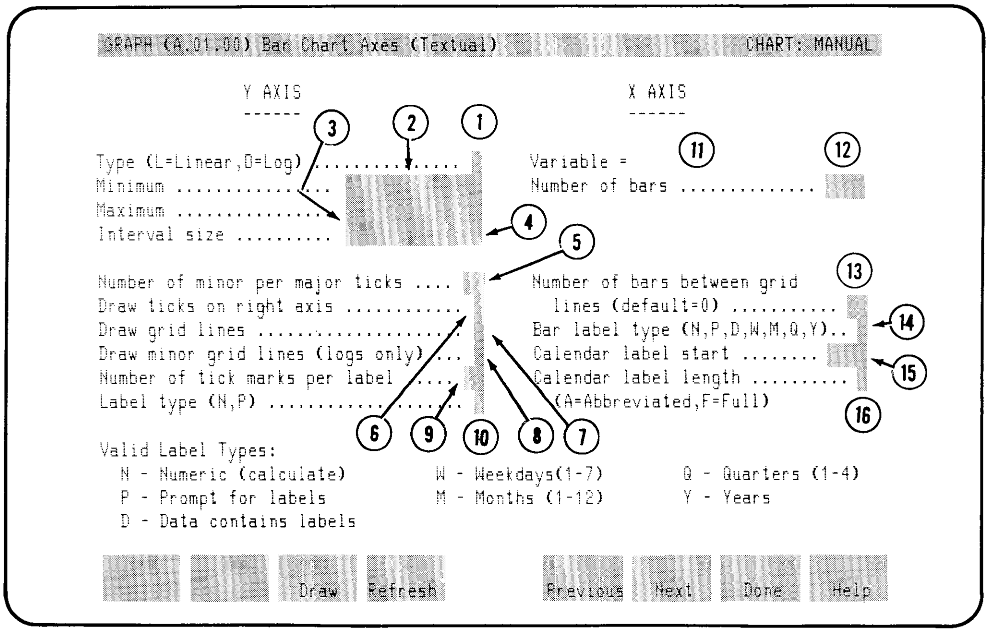
HELP (f8) Initiates Help Facility, giving an explanation of the BAR CHART AXES Menu (Numeric).

ROADMAP


QUALIFICATIONS MENU OR GRAPHING OPTIONS
NEXT
CHART DESIGN CONTROL MENU
X
BAR CHART AXES MENU (NUMERIC)
BAR CHART LEGENDS MENU (OR LABEL PROMPT MENU IF USED)
PREVIOUS

If you specify X (Axes) on the CHART DESIGN CONTROL Menu for a bar chart with a numeric X-axis variable, this menu comes up on your screen. It also appears when you press PREVIOUS from the BAR LEGENDS Menu or LABEL PROMPT Menu if that menu was used, or NEXT from QUALIFICATIONS Menu or GRAPHING OPTIONS Menu if user labels are not used.

Bar Chart Axes Menu (Textual)



```
GRAPH (A.01.00) Bar Chart Axes (Textual)                         CHART: MANUAL

              Y AXIS                                    X AXIS
              ------                                    ------

Type (L=Linear,O=Log) ..............           Variable =
Minimum ...............                        Number of bars .............
Maximum ...............
Interval size ..........

Number of minor per major ticks ....           Number of bars between grid
Draw ticks on right axis ...........             lines (default=0) ..........
Draw grid lines ....................           Bar label type (N,P,D,W,M,Q,Y)..
Draw minor grid lines (logs only)...           Calendar label start ........
Number of tick marks per label .....           Calendar label length .........
Label type (N,P) ...................             (A=Abbreviated,F=Full)

Valid Label Types:
  N - Numeric (calculate)          W - Weekdays(1-7)         Q - Quarters (1-4)
  P - Prompt for labels            M - Months (1-12)         Y - Years
  D - Data contains labels

                  Draw   Refresh              Previous   Next   Done   Help
```

Figure 5-10. BAR CHART AXES (TEXTUAL) Menu

Specifies the characteristics of both axes for a textual bar chart including: scaling, spacing and labeling of ticks, and grid lines.

In a textual bar chart, the X-axis variable (usually plotted on the horizontal axis) has been specified as textual (or left blank) on the BAR CHART Menu. If the X-axis variable is numeric, GRAPH uses the BAR AXES Menu (Numeric) instead. In either case, the Y-axis variables must be numeric.

FIELD EXPLANATION

1. Type
   The *L* that is entered in this field by default represents linear scaling. If you want logarithmic scaling instead, enter an *O*. Logarithmic scaling may be useful if the data takes values over a very long range, or if you want to draw a linear plot of two variables that have an exponential relationship.

2. Minimum
   Enter the minimum value for the axis. If this field is left blank, DSG/3000 calculates the minimum from the data values.

3. Maximum
   Enter the maximum value for the axis. The maximum value must be greater than the minimum value. If you do not enter anything here, DSG/3000 calculates the maximum from the data values. For both numeric and textual Bar Chart axes, minimum and maximum cannot both be on one side of the 0 base line.

4. Interval Size
   Enter the number of units you want between major ticks in terms of the data. For example, if you want $100 between ticks, enter 100. If you do not fill this in, DSG/3000 calculates the units between ticks from the data in such a way that there are approximately 5 intervals along the axis. If Type is logarithmic (*O*), this field represents the number of powers of 10 which will separate major tick marks. For example, a value of 1 means that every power of 10 will have a tick mark.

5. Number of Minor per Major Ticks
   Enter the number of minor tick marks between major ticks. Minor ticks are drawn smaller than major ticks and are not labeled. If you do not fill in this field, there will be no minor ticks. For logarithmic charts, a common number of minor per major ticks is 8. Lowering this number results in a less dense axis or chart.

6. Draw Ticks on Right Axis
   If you want the Y-axis ticks to be drawn on the right axis as well as the left one, enter any character in this field. (If you opted for horizontal bars on the BAR CHART Menu, entering a character here causes the ticks to be drawn on the upper axis as well as the lower axis.)

7. Draw Grid Lines
   If you want grid lines to be drawn wherever there are major ticks, enter any character here.

8. Draw Minor Grid Lines (Log scaling only)
   This field is ignored unless you have specified:

   log scaling (field 1)
   grid lines (field 7)
   number of minor per major ticks is >0 (field 5)

   If these conditions are satisfied and you type any character in this field, grid lines are drawn at minor ticks as well as major ticks.

9. Number of Tick Marks per Label
   If you enter a 1 in this field (or leave it blank), every major tick will be labeled. Enter the number of major ticks per label. For example, if you want to label every other tick, enter a 2, which means that there will be two ticks per label.

10. Label Type
   If you want the major ticks to be labeled according to the axis scaling, enter an N in this field. This is the default. If you want to supply your own labels which might be textual rather than numeric, enter a *P*; after you have pressed the ENTER for this menu, you can enter the labels on the LABEL PROMPT Menu.

11. X-Axis Variable (Display only)
   The name of the independent variable is displayed here. It cannot be altered.

12. Number of Bars
If you do not fill in this field, DSG/3000 draws one bar for each data point. If you specify that you want a greater number of bars than there are data points, DSG/3000 leaves blank space for the remaining data values. If you specify fewer bars than there are data points, DSG/3000 draws only as many bars as you request, starting from the beginning of the data file.

13. Number of Bars Between Grid Lines
If you do not fill in this field, no grid lines are drawn parallel to the bars. If you want grid lines to be drawn, enter the number of bars between grid lines.

14. Bar Label Type
Enter one of the following:

N DSG/3000 calculates numeric labels from the axis scaling.

D The data variable that you specified as the X-axis variable in the BAR CHART Menu contains the bar labels. (Default)

P Use this option if you want to enter your own bar labels. After you have pressed ENTER and then NEXT, GRAPH issues the LABEL PROMPT Menu. (If you entered a *P* for both axes you will fill in the LABEL PROMPT Menu twice.)

For convenience, GRAPH allows you to specify the following time labels regardless of the values of the independent variable. Fields 15 and 16 are used with time labels to give the starting point and length of the labels.

W Weekdays. Each bar will be labeled with a day of the week. If there are more than 7 bars, the labels will be repeated.

M Months. Each bar will be labeled with the name of a month. If there are more than 12 bars, the labels will be repeated.

Q Quarters. Each bar will be labeled with a quarter. If there are more than 4 bars, the labels will be repeated.

Y Year. Each bar will be labeled with a year.

15. Calendar Label Start (used only with time labels)
Enter a number corresponding to the weekday, month, quarter, or year that labels the first (leftmost) bar. The default is 1, which corresponds to MON for weekdays, JAN for months, QTR1 for quarters, or 1 for year. For example, if you want the bars to be labeled by month beginning with November, type *11* in this field. If you want the bars to be labeled by years beginning with 1982, for example, type 1982 in this field.

16. Calendar Label Length (used for time labels only)
A Abbreviated label form is printed on the graph. (default)
L Long form is printed on the graph.

	Abbreviated	Long
Weekdays	MON	MONDAY
Months	JAN	JANUARY
Quarters	Q1	QTR1
Years	N/A	1982

DISCUSSION

The X-axis is labeled with the name of the x variable.

If you do not fill in this menu, the axes characteristics take the following defaults:

* linear axes
* axes scaled according to the data values
* major ticks on Y-axis with numeric labels calculated from the scaling
* major ticks on X-axis with textual labels calculated from the data.
* no minor ticks or grid lines

Note that calendar labels are blindly applied to tick marks, irrespective of the actual data. To avoid mislabeling, the X-axis interval size should be set to 1 in most cases.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

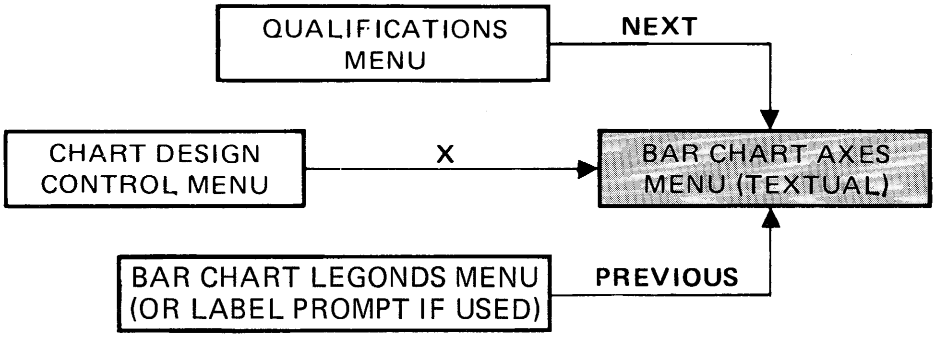
PREVIOUS (f5) Goes to QUALIFICATIONS Menu.

NEXT (f6) If a *P* (Prompt) is entered in field 14, goes to LABEL PROMPT Menu. Otherwise, goes to BAR CHART LEGENDS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an of the BAR CHART AXES Menu (Textual).

ROADMAP


QUALIFICATIONS MENU
NEXT
CHART DESIGN CONTROL MENU
X
BAR CHART AXES MENU (TEXTUAL)
BAR CHART LEGONDS MENU (OR LABEL PROMPT IF USED)
PREVIOUS

If you are drawing a bar chart with a textual X-axis variable and you press *X* (axes) on the CHART DESIGN CONTROL Menu, this menu appears on your screen. You can also display it by pressing NEXT from QUALIFICATIONS or PREVIOUS from BAR CHART LEGENDS or LABEL PROMPT Menus.

Bar Chart Legends Menu

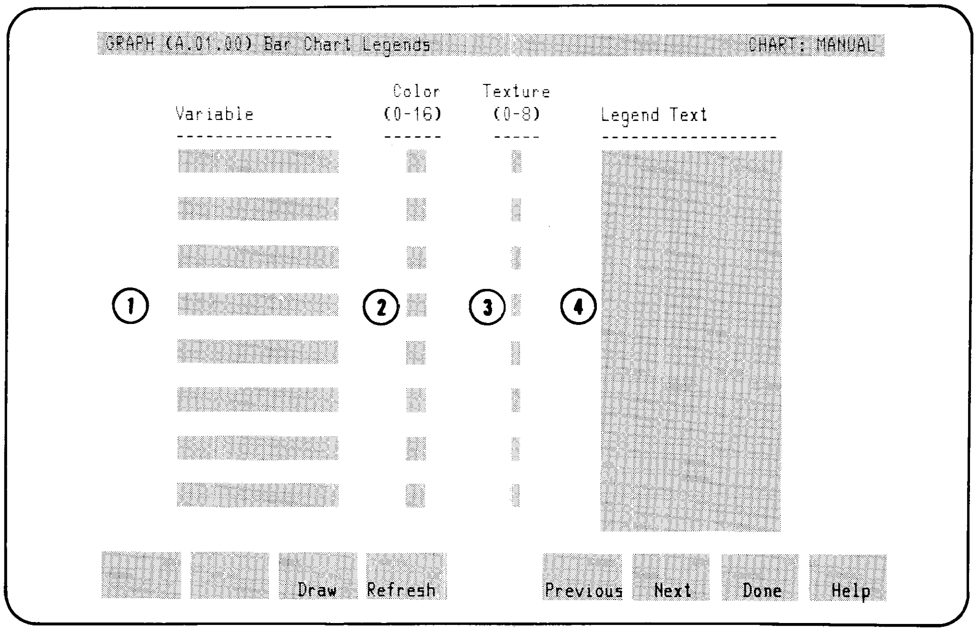


Figure 5-11. BAR CHART LEGENDS Menu

Specifies color, texture, and legend text for each bar variable in a bar chart.

FIELD EXPLANATION

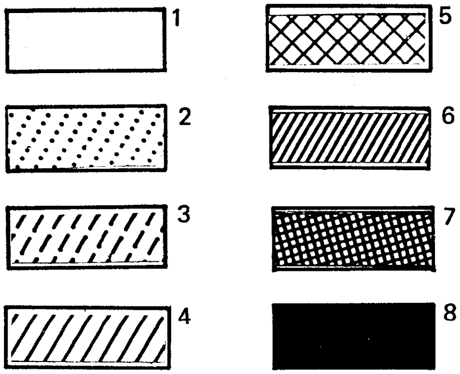
1. Variable (display only)
   Each of the Y-axis variables is displayed here. These names, taken from the BAR CHART Menu, cannot be altered on the BAR LEGENDS Menu.

2. Color
This field only affects the chart if it is drawn on a multi-color plotter with multiple pens or color terminal. If nothing is entered in this field, the entire graph is drawn in black. If you want to assign a color to any of the bar variables, enter one of the following numbers next to the corresponding variable name:

1	Black
2	Red
3	Green
4	Blue
5...16	

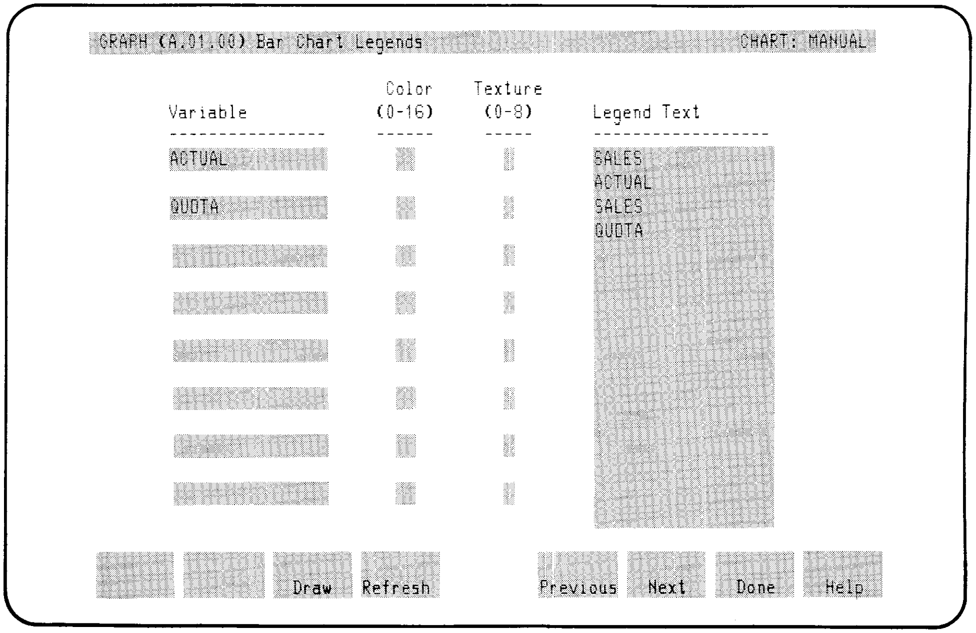
This presumes that the pens 1-4 are inserted in the plotter in this order. (See Appendix A for information about color.)

3. Texture
If more than one Y-axis variable is being plotted, DSG/3000 assigns a different texture to each in the order shown in Appendix A. If you want to change the texture for any of the variables, enter one of the following numbers next to the corresponding variable name:



4. Legend Text
For each Y-axis variable shown, GRAPH prints a legend showing each texture and the legend text shown in the last column. To change the legend text, enter the revised text here. This field can contain up to 18 printable characters on each of 2 lines.

You can draw the graph without the legend by filling in the GRAPHING OPTIONS Menu appropriately. If your legend will only fill one line, use the lower for the best visual effect.



```
GRAPH (A.01.00) Bar Chart Legends                          CHART: MANUAL

                                  Color    Texture
         Variable                 (0-16)    (0-8)      Legend Text
         ----------------         ------    -----      ------------------
         ACTUAL                                        SALES
                                                       ACTUAL
         QUOTA                                         SALES
                                                       QUOTA

                  Draw    Refresh           Previous   Next    Done    Help
```

Figure 5-12. BAR CHART LEGENDS Menu with Legend Text

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

PREVIOUS (f5) Goes to BAR AXES Menu, or LABEL PROMPT Menu if it was used.

NEXT (f6) Goes to BAR AND LINE CHART TITLES Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the BAR CHART Menu.

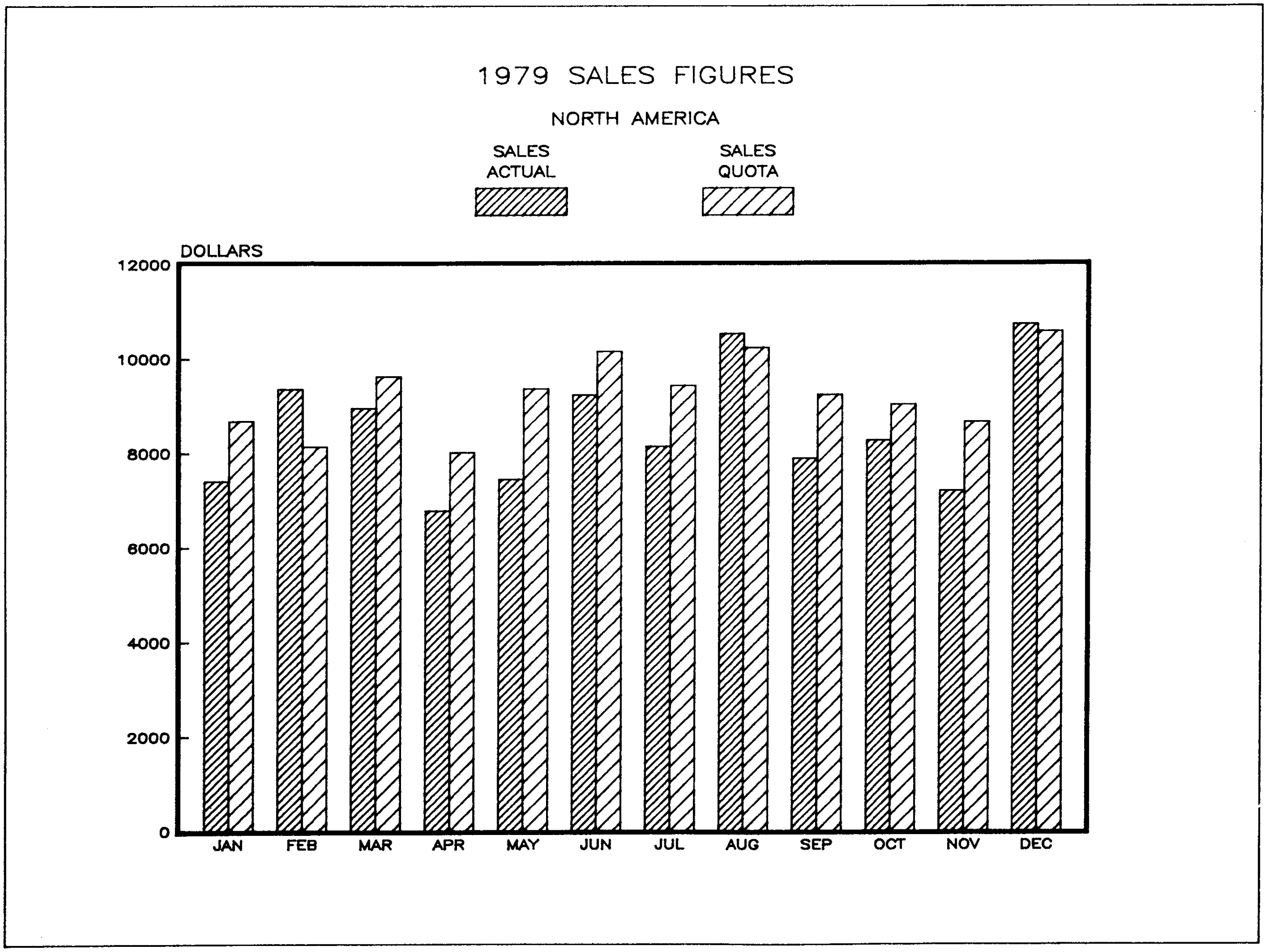
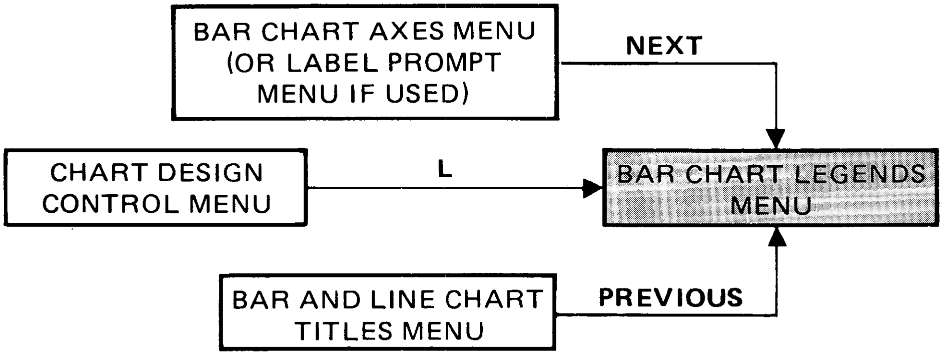


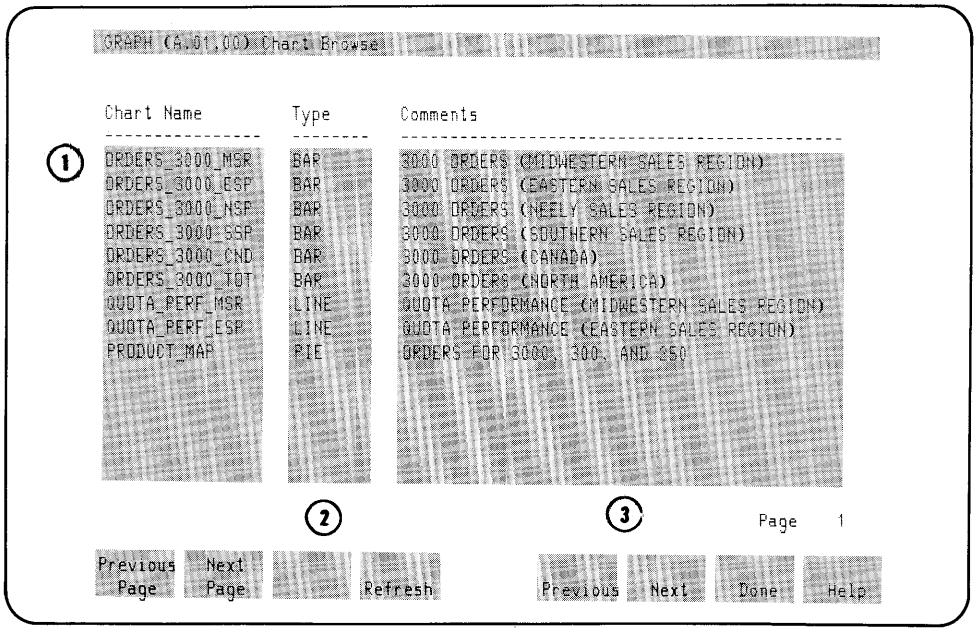
Figure 5-12a. BAR LEGENDS GRAPH

ROADMAP


BAR CHART AXES MENU (OR LABEL PROMPT MENU IF USED)
NEXT
CHART DESIGN CONTROL MENU
L
BAR CHART LEGENDS MENU
BAR AND LINE CHART TITLES MENU
PREVIOUS

If you are drawing a bar chart and enter L (Legends) on the CHART DESIGN CONTROL Menu, this menu appears. You can also press NEXT from BAR AXES or LABEL PROMPT Menus, or PREVIOUS from BAR AND LINE CHART TITLES.

Chart Browse Menu



```
GRAPH (A.01.00) Chart Browse

Chart Name          Type        Comments
----------------    --------    ---------------------------------------------
ORDERS_3000_MSR     BAR         3000 ORDERS (MIDWESTERN SALES REGION)
ORDERS_3000_ESP     BAR         3000 ORDERS (EASTERN SALES REGION)
ORDERS_3000_NSP     BAR         3000 ORDERS (NEELY SALES REGION)
ORDERS_3000_SSP     BAR         3000 ORDERS (SOUTHERN SALES REGION)
ORDERS_3000_CND     BAR         3000 ORDERS (CANADA)
ORDERS_3000_TOT     BAR         3000 ORDERS (NORTH AMERICA)
QUOTA_PERF_MSR      LINE        QUOTA PERFORMANCE (MIDWESTERN SALES REGION)
QUOTA_PERF_ESP      LINE        QUOTA PERFORMANCE (EASTERN SALES REGION)
PRODUCT_MAP         PIE         ORDERS FOR 3000, 300, AND 250

                                                              Page    1

Previous  Next                               Previous  Next  Done  Help
  Page    Page          Refresh
```

Figure 5-13. CHART BROWSE Menu Display Information

Displays information about all charts in the current chart file, including chart name, chart type, and comments. It is for information display only. When you have more than 14 charts in the file, use function key f1 and f2 to page through your list.

FIELD EXPLANATION

1. Chart Name (Display only)
   Lists the names of every chart in the chart file. Charts are named or renamed on the MAIN CONTROL Menu.

2. Type (Display Only)
Gives the chart type as one of the following: Pie, Line, Bar or Slide. Chart type is specified on the CHART DESIGN CONTROL Menu.

3. Comments (Display only)
For each chart, displays optional comments that may have been entered on the CHART INFORMATION Menu.

FUNCTION KEYS

PRV PAGE (f1) Displays information about charts preceding those currently displayed.

NXT PAGE (f2) Displays information about charts following those currently displayed.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

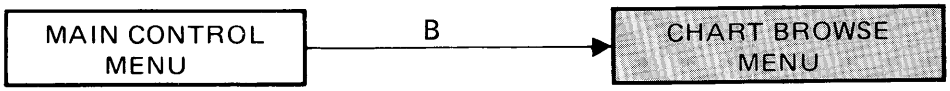
PREVIOUS (f5) Goes to MAIN CONTROL Menu.

NEXT (f6) Goes to MAIN CONTROL Menu.

DONE (f7) Goes to MAIN CONTROL Menu.

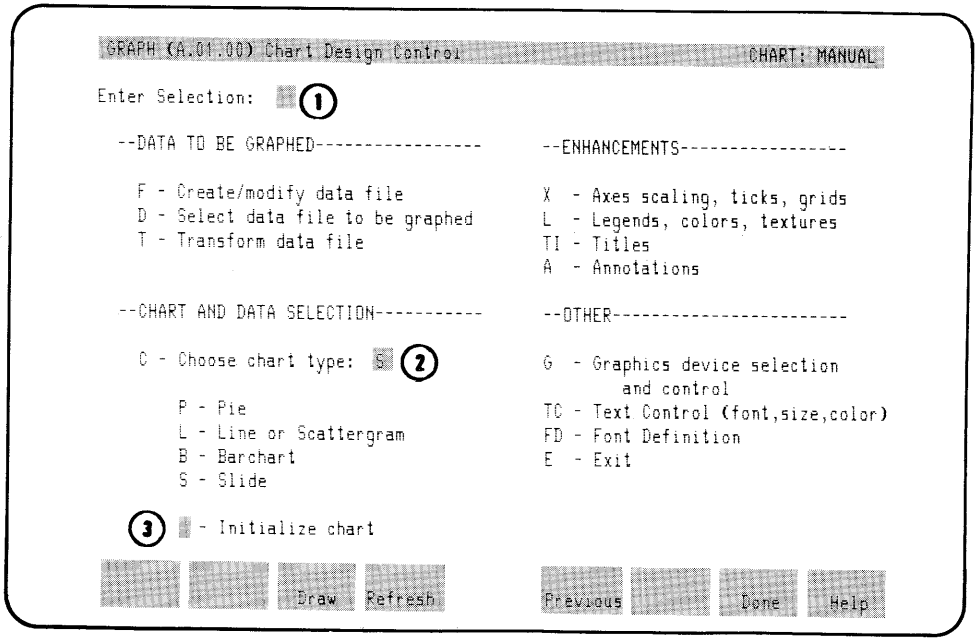
HELP (f8) Initiates Help Facility, giving an explanation of the CHART BROWSE Menu

ROADMAP



Entering a *B* on the MAIN CONTROL Menu brings this to your screen.

Chart Design Control Menu



```
GRAPH (A.01.00) Chart Design Control                                  CHART: MANUAL

Enter Selection:

  --DATA TO BE GRAPHED-----------------        --ENHANCEMENTS-----------------

    F - Create/modify data file                X  - Axes scaling, ticks, grids
    D - Select data file to be graphed         L  - Legends, colors, textures
    T - Transform data file                    TI - Titles
                                               A  - Annotations

  --CHART AND DATA SELECTION-----------        --OTHER-----------------------

    C - Choose chart type: S                   G  - Graphics device selection
                                                     and control
      P - Pie                                  TC - Text Control (font,size,color)
      L - Line or Scattergram                  FD - Font Definition
      B - Barchart                             E  - Exit
      S - Slide

       - Initialize chart

                  Draw    Refresh              Previous            Done     Help
```

Figure 5-14. CHART DESIGN CONTROL Menu

This menu branches to any one of the design menus used to create or modify a chart. Required for all chart design.

FIELD EXPLANATION

1. Selection (Required)
   Must be one of the following letters:

   F Build/modify data file. Goes to DATA PROMPT Menu, which allows you to supply or alter the input data.

D Data Definition. Goes to DATA DEFINITION Menu, which allows you to name the data file and describe the variables.

T Data transformations. Goes to DATA TRANSFORMATIONS Menu, which creates new variables mathematically from the original variables.

C Specify chart type. Goes to either the PIE CHART, LINE CHART, BAR CHART, or ANNOTATIONS Menu. If a pie, line or bar chart is being specified, the data should already have been specified via the DATA DEFINITION Menu or DATA TRANSFORMATIONS Menu.

X Axes. Goes to the LINE CHART AXES or BAR CHART AXES Menu. Allows you to specify scaling, major and minor ticks, grids, and the label source.

L Legends. Goes to the appropriate legends menu. You must have specified the chart type.

TI Titles. Goes to the appropriate titles menu. You must have specified the chart type.

A Annotate chart. Goes to ANNOTATIONS Menu, which allows you to draw boxes, lines, arrows, or text on the chart.

G Graphics device. Goes to the GRAPHING OPTIONS Menu, which allows you to route output to a plotter.

TC Text control. Goes to the either the PIE TEXT CONTROL Menu or the BAR AND LINE TEXT CONTROL Menu, whichever is appropriate. Allows you to specify font, size, and color for titles, subtitles, axis labels, legend labels, and footnotes but not annotations.

FD Font Definition Goes to FONT DEFINITION Menu, which allows you to relate the four fonts available to font files as well as specify native languages for those fonts.

Note: Only one hardware font is available and the font type depends on the device used. The native language characteristics of the font are established by the language specified for font 1 on the FONT DEFINITION Menu.

E Exit. Terminates GRAPH and returns you to MPE.

2. Chart Type
Must be one of the following letters:

P	Pie chart
L	Line chart or scattergram
B	Bar chart
S	Slide (annotations only)

When this menu first comes up on your terminal, this field contains an "S" because the default chart-type is Slide only. To specify a different chart type, type a *C* in the first field, type over the *S* if you are changing the type, and press ENTER.

This field has no effect on the chart type unless you have also entered a "C" (Specify chart type) to establish the current chart type.

3. Initialize
By typing any character in this field, you reset certain chart attributes to their default values. For example, there are no titles, legends, annotations, subsets, and pie explosions. Data file information entered through the DATA PROMPT, DATA DEFINITION, and DATA TRANSFORMATIONS Menus remains unchanged. Initialize can be used to start over on a chart and still maintain the order of the charts in the chart file. This field is ignored unless you have also entered a "C" (Specify chart type) to establish the current chart type.

DISCUSSION

You pass through this menu at least twice when you are creating a chart: first to go to the DATA DEFINITION Menu (D), and then to go to the appropriate chart type menu (C). The menu functions as a transfer point between the different menus used in designing a chart, and several design menus return control to this menu after they have been completed.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

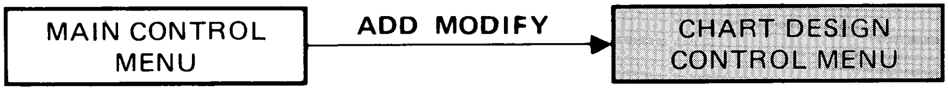
REFRESH (f4) Refreshes the menu by redrawing it with the information entered as of the last ENTER.

PREVIOUS (f5) Goes to the MAIN CONTROL Menu.

DONE (f7) Returns to MAIN CONTROL Menu.

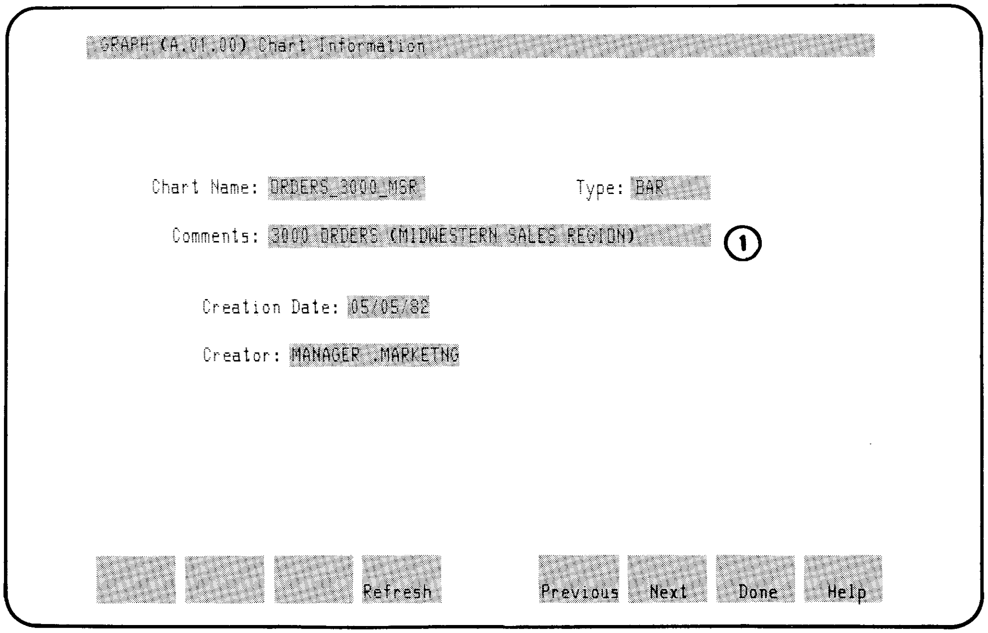
HELP (f8) Initiates the Help Facility which gives an explanation of the CHART DESIGN CONTROL Menu.

ROADMAP



If you enter *A* or *M* on the MAIN CONTROL Menu, this menu appears on the screen.

Chart Information Menu



```
GRAPH (A.01.00) Chart Information

     Chart Name: ORDERS_3000_MSR              Type: BAR

       Comments: 3000 ORDERS (MIDWESTERN SALES REGION)

           Creation Date: 05/05/82

           Creator: MANAGER .MARKETNG
```

Figure 5-15. CHART INFORMATION Menu

This menu displays information and allows you to enter comments about the current chart.

FIELD EXPLANATION

1. Comments
   This field is optional. Enter any information you would like. The comments are also displayed when you browse the chart file by entering a *B* on the MAIN CONTROL Menu.

DISCUSSION

When the CHART INFORMATION Menu comes up on your screen, the chart name, type, the date that it was created, and the name of the user and account in which it was created are displayed. The cursor is positioned in the "Comments" field so that you can enter any comment of up to 45 characters in length which will help you to identify the chart.

There is only one unprotected field in this menu. You cannot change the chart name, date, or creator.

FUNCTION KEYS

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

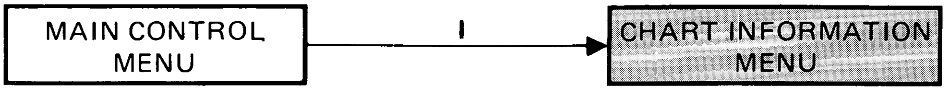
PREVIOUS (f5) Returns to MAIN CONTROL Menu.

NEXT (f6) Returns to MAIN CONTROL Menu.

DONE (f7) Returns to MAIN CONTROL Menu.

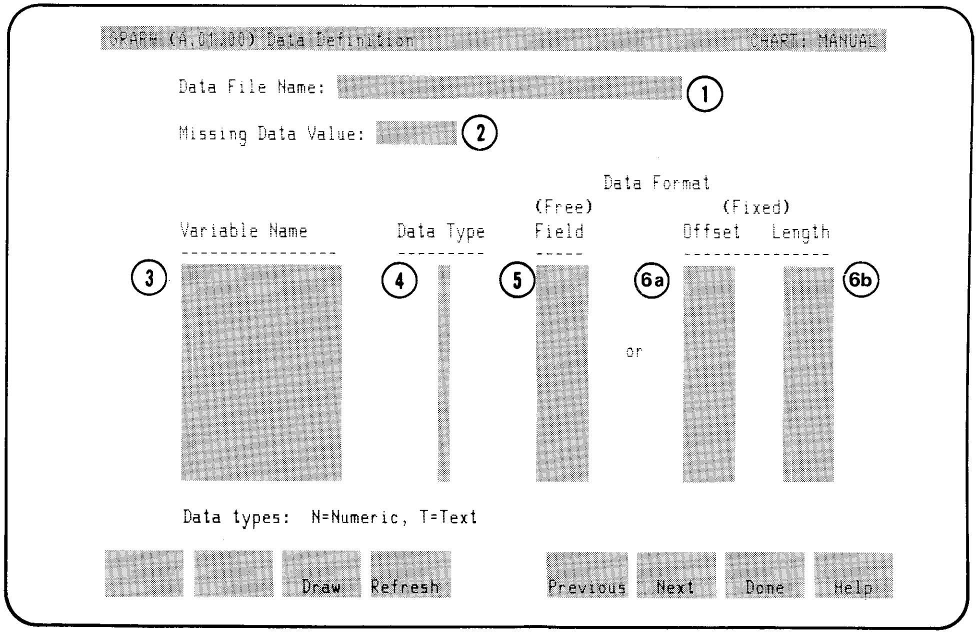
HELP (f8) Initiates the Help Facility, giving an explanation of the CHART INFORMATION Menu.

ROADMAP



To display this menu on your screen, enter an *I* on the MAIN CONTROL Menu.

Data Definition Menu



GRAPH (A.01.00) Data Definition CHART: MANUAL

Data File Name:

Missing Data Value:

```
                                        Data Format
                                 (Free)          (Fixed)
Variable Name        Data Type   Field       Offset   Length
----------------     ---------   -----       ---------------
                                          or
```

Data types: N=Numeric, T=Text

Figure 5-16. DATA DEFINITION Menu

Use this menu to specify the data file and describe the variables within that data file. Required for all charts except slides.

FIELD EXPLANATION

1. Data File Name (Required)
   Enter the name of the MPE file containing your data. The file name can be fully qualified:

   filename[/lockword][.group [.account]]

The file name can also be back-referenced with a formal designator preceded by an asterisk (*) if you have issued a :FILE command for your session. See the *MPE Commands Reference Manual* for information about back-referencing files.

2. Missing Data Value (Optional)
   You may have used a special data value in your data file to represent data that was missing at the time the file was created. If you want DSG/3000 to ignore the missing data points rather than plotting them as value zero, enter the value of the missing data here. The missing data value must be numeric.

   The default value is -9999. Blanks will be treated as 0. See Figures 5-17 and 5-18 for an example of missing data.

3. Variable Name (Required)
   Enter the name of each variable in the data file. Each variable must have a unique name. Variable names can be up to 16 characters long, beginning with a letter. The remaining characters can be letters, numbers, hyphens (-) or underlines (_).

   You cannot use any of the following reserved words as variable names:

   ABS
   AND
   CUMULATE
   LN
   LOG
   MOD
   MOVEAVG
   NOT
   OR
   SQRT

4. Data Type (Required)
   Each variable must be identified as numeric (N) or textual (T). If the variable contains only numbers with optional leading sign (+ or -), decimal point, or exponent (E), specify the data as numeric (N).

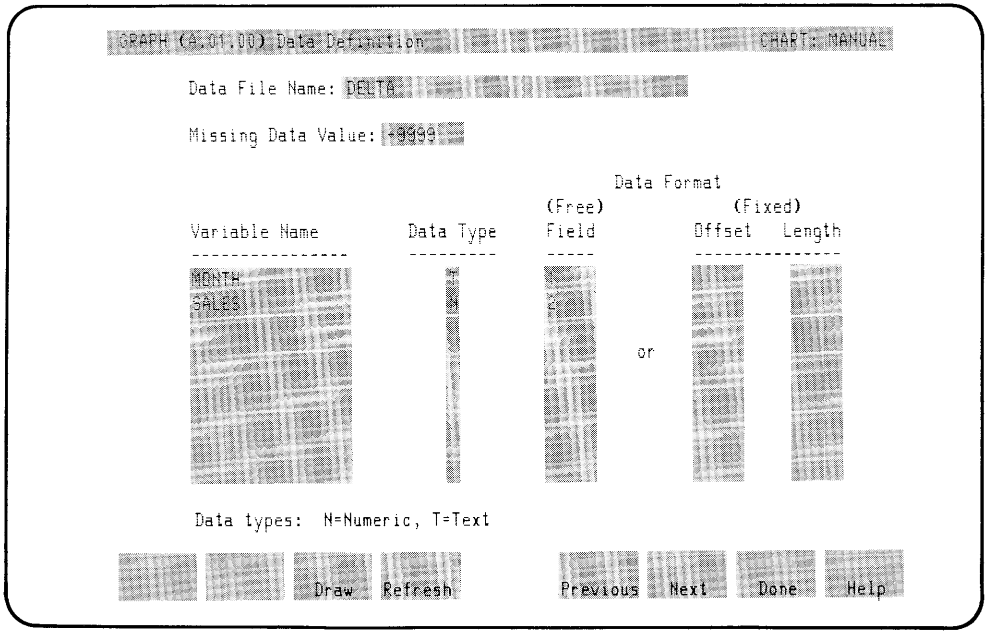
   For example, the following data values might be contained in a numeric variable:

   236.7
   325
   -99
   +2.3E3

   where the last value is equal to 2.3 * (10) **3. Any real data values are acceptable for numeric variables. (Note: negative sign (-) must precede the value.)

If the data values contain alphabetic characters, you must specify specify the data as textual (T).

5. Data Format (Free): Field
If the data file is in free format, you must fill in this field. Free format data is arranged in fields that are separated by commas or blanks. (See Section III for a full discussion of data formats.) For each variable, enter the number of the data field in which it occurs. Figure 5-17 shows a free format data file and the corresponding DATA DEFINITION Menu. The first field is field 1.



GRAPH (A.01.00) Data Definition CHART: MANUAL

Data File Name: DELTA

Missing Data Value: -9999

		Data Format		
		(Free)	(Fixed)	
Variable Name	Data Type	Field	Offset	Length
MONTH	T	1		
SALES	N	2		

or

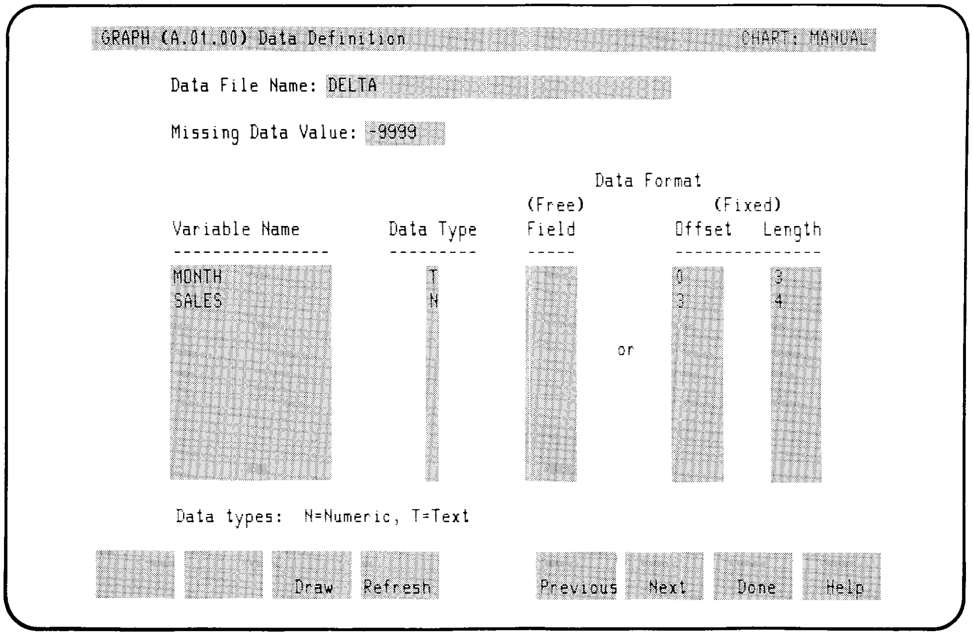
Data types: N=Numeric, T=Text

Figure 5-17. Free Format Data File and DATA DEFINITION Menu

6a. Data Format (Fixed): Offset
If the data file is in fixed format, you must fill in this field. In a fixed format data file, each data field begins in a particular column and has a fixed length. Specify the offset for each variable as the number of columns or characters from the beginning of the data record to the beginning of the data field.

Figure 5-18 shows a fixed format data file and the corresponding DATA DEFINITION Menu.

The variable MONTH has offset 0 because there are no characters before it in the data file; the variable SALES has offset 3 because there are 3 characters before it in the data file.



GRAPH (A.01.00) Data Definition CHART: MANUAL

Data File Name: DELTA

Missing Data Value: -9999

		Data Format		
		(Free)	(Fixed)	
Variable Name	Data Type	Field	Offset	Length
MONTH	T		0	3
SALES	N		3	4

Data types: N=Numeric, T=Text

Figure 5-18. Fixed-Format Data File and DATA DEFINITION Menu

6b. Data Format (Fixed): Length
If the data file is in fixed format, you must specify the length in characters of the data field for each variable. In the above example, the variable MONTH is 3 characters long and the variable SALES is 4 characters long.

DISCUSSION

If your data file was generated using the DATA PROMPT Menu, simply fill in the name of the file and press ENTER. GRAPH will fill in the variable names, data types, and data format for you. If your data file was created by any other method, you must fill in all the fields yourself.

You can only plot values from one data file at a time. If you wish to use the same chart specifications to plot data from a different data file, go to the DATA DEFINITION Menu and type the name and variable description of the new file over the information for the old file. If the new data file has been generated by the DATA PROMPT Menu, the procedure to change data is even more simple. Type the new data file name over the old one, fill in the Missing Data Value field, and press CLEAR DISPLAY to clear the rest of the menu following the cursor. When you press ENTER, GRAPH will fill in the variable descriptions for you.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

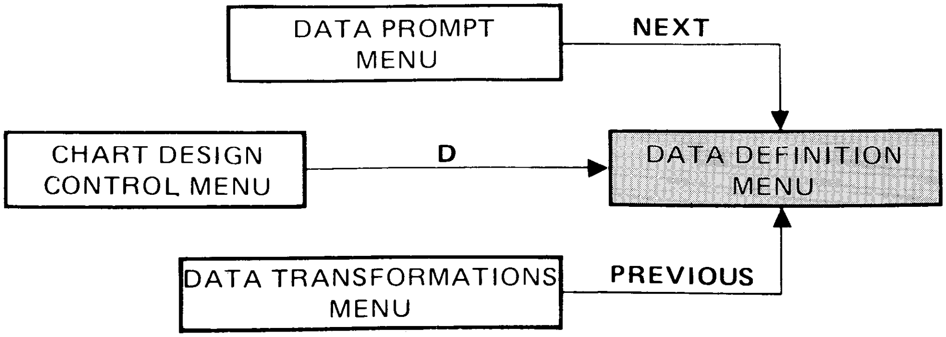
PREVIOUS (f5) Goes to DATA PROMPT Menu.

NEXT (f6) Goes to DATA TRANSFORMATIONS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the DATA DEFINITION Menu.

ROADMAP



If you enter a *D* on the CHART DESIGN CONTROL Menu, you bring this menu to the screen. Pressing NEXT from the DATA PROMPT Menu or PREVIOUS from DATA TRANSFORMATION Menu will also display the menu.

Data Prompt Menu

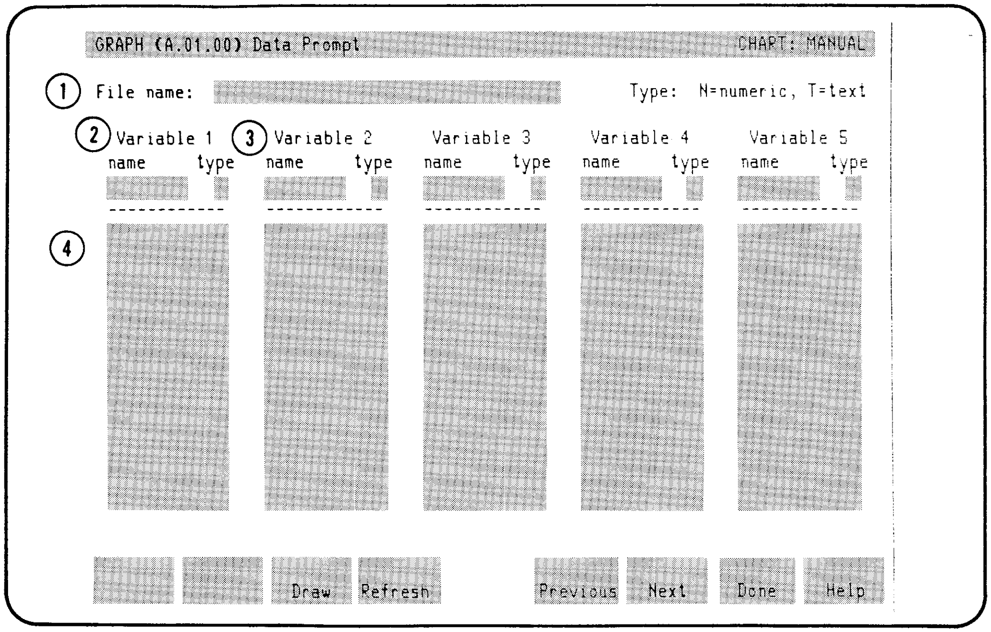


Figure 5-19. DATA PROMPT Menu

This menu allows you to construct a data file to be graphed. The file can contain up to 5 variables with 12 values each.

FIELD EXPLANATION

1. File Name (Required)
   Type in the data file name, which can take the following form:

filename[/lockword][.group[.account]]

Filenames and lockwords may each contain up to 8 alphanumeric characters, beginning with a letter.

Filenames cannot contain any special characters or embedded blanks. If you are accessing a file that resides in a different group or account, you must supply their names also. See the *MPE System Manager Reference Manual* for more information on lockwords and security.

2. Variable Name
You must supply a name for every variable that you include in the data file. The variable name can be up to 8 characters beginning with a letter and consisting of letters, numbers, dashes (-), or underlines (_).

3. Type
For each variable in the data file, you must specify either an *N* for numeric or a *T* for textual (non-numeric). When you specify *N*, you must use only numeric values; specifying *T* allows you to use anything.

For example, Figure 5-19 shows a data file containing two variables. MONTH, containing letters, is a textual variable, and SALES, containing numbers, is a numeric variable.

4. Data Values
Fill in the values for each variable that you are creating in this menu. The values can each be up to 12 characters long. This is the data that will actually be plotted on the graph. Remember that numeric variables can only have numbers as data values, as described in *Type* above. If any values are left blank, they are treated as missing data. Any signs (+,-) must be leading.

DISCUSSION

Before you can produce a chart, you must create a data file by one of the 6 methods outlined in Section III. If you intend to create a data file by entering the data interactively through GRAPH, you must fill out the DATA PROMPT Menu. Note that data files created through the DATA PROMPT Menu are not automatically associated with a particular chart. You must use the DATA DEFINITION Menu to assign a data file to a chart.

If you are creating a new data file, enter the name of the file, the names of the variables, and the data values. If you want to read or modify the values in a file that already exists and was created with the DATA PROMPT Menu, type in the file name and press ENTER; GRAPH will fill in the rest of the menu for you if it is blank.

In general, when you press ENTER, GRAPH will attempt to create and permanently save the data file as it appears on your screen. For example, if you have made changes or additions to an existing data file and press ENTER, GRAPH asks you to confirm that you want to write over the original contents of the file. If you again press ENTER, the new contents are saved and the previous contents are lost. If you press a function key, the original contents of the file remain unaltered.

If you wish to access a data file other than the one currently displayed by the DATA PROMPT Menu, type the name of the new file over the old one, press CLEAR DISPLAY to clear the rest of the display, and press ENTER. GRAPH will return the contents of the new data file to your screen.

Remember that the DATA PROMPT Menu allows you to access only those files that were created using it. When you have finished creating your data file, press ENTER. When the cursor returns to the first field (File Name), you can press DONE (f7) to continue designing the graph. If you have already defined the data and designed the chart, you can draw the graph by pressing DRAW (f3).

FUNCTION KEYS

DRAW (f3) Draws the current chart.

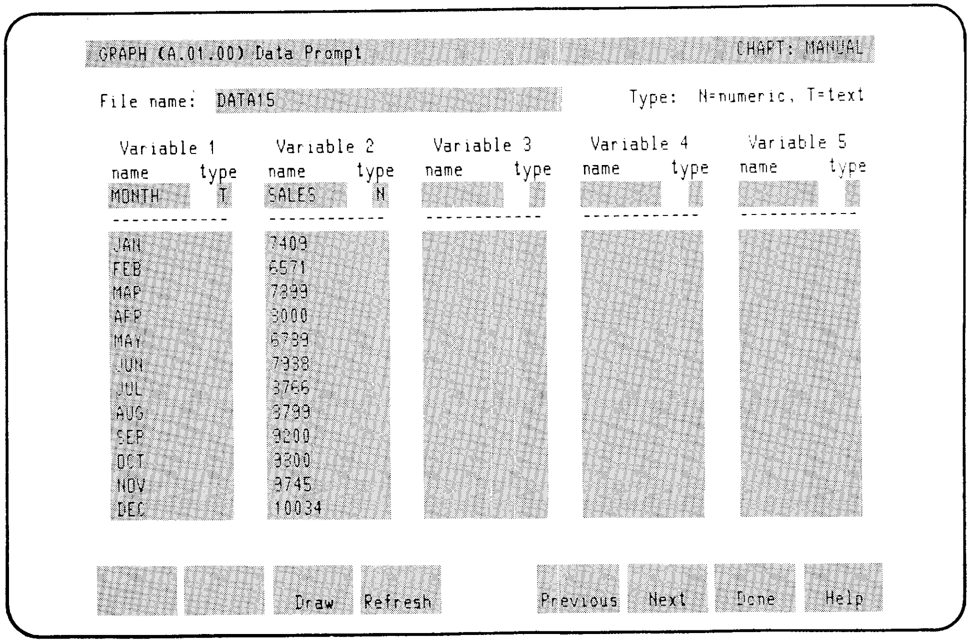
REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

PREVIOUS (f5) Goes to the MAIN CONTROL Menu or CHART DESIGN CONTROL Menu, as applicable.

NEXT (f6) Goes to DATA DEFINITION Menu or MAIN CONTROL MENU, as applicable.

DONE (f7) Goes to MAIN CONTROL Menu or CHART DESIGN CONTROL Menu, as applicable.

HELP (f8) Initiates Help Facility, giving an explanation of the DATA PROMPT Menu.



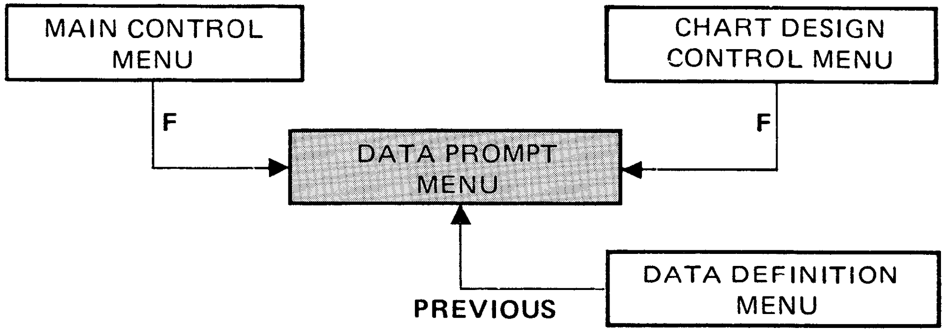
GRAPH (A.01.00) Data Prompt CHART: MANUAL

File name: DATA15 Type: N=numeric, T=text

Variable 1		Variable 2		Variable 3		Variable 4		Variable 5	
name	type	name	type	name	type	name	type	name	type
MONTH	T	SALES	N						
JAN		7409							
FEB		6571							
MAR		7899							
APR		3000							
MAY		6799							
JUN		7938							
JUL		3766							
AUG		3799							
SEP		9200							
OCT		3300							
NOV		3745							
DEC		10034							

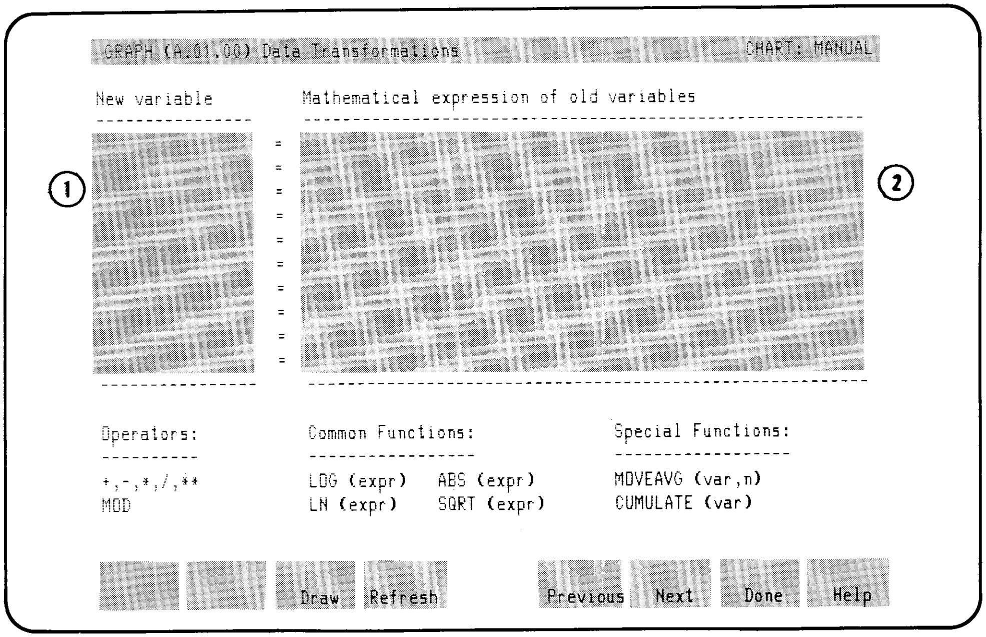
Figure 5-20. Completed DATA PROMPT Menu

ROADMAP



Entering *F* from the MAIN CONTROL or CHART DESIGN CONTROL Menus brings this menu to the screen. It will also appear if you press PREVIOUS from the DATA DEFINITION Menu.

Data Transformations Menu



GRAPH (A.01.00) Data Transformations CHART: MANUAL

New variable

Mathematical expression of old variables

Operators:

+,-,*,/,**
MOD

Common Functions:

LOG (expr) ABS (expr)
LN (expr) SQRT (expr)

Special Functions:

MOVEAVG (var,n)
CUMULATE (var)

Figure 5-21. DATA TRANSFORMATIONS Menu

Using this menu you create new variables for graphing by mathematically transforming the original data variables.

FIELD EXPLANATION

1. New Variable
Enter the names of the variables that you are creating. Each variable name can have up to 16 characters. Naming conventions are the same as for other variable names.

2. Mathematical Expression
Enter a mathematical expression using variable names from the data file, and operators and functions as defined below. Remember that the data variables used here must be previously defined in either the DATA DEFINITION Menu or the DATA TRANSFORMATIONS Menu. (Table 5-3 summarizes the operators that can be used in mathematical operators.)

DISCUSSION

This Menu allows you to name and define up to 10 new variables as mathematical expressions of the variables in your data file. The data file variables must have been defined in the DATA DEFINITION Menu before being used in these mathematical expressions. After you define the new variables, you can specify them as variables to be plotted on the PIE CHART, BAR CHART, or LINE CHART Menus.

OPERATOR OR FUNCTION	NOTATION	EXAMPLE
Add	+	SALES + INVENTORY
Subtract	−	SALES - INVENTORY
Multiply	*	SALES * 12
Divide	/	SALES / 12
Exponentiate	**	SALES**2
Modulo (divide and save remainder)	MOD	YEAR MOD 100
Common Logarithm (base 10)	LOG (expression)	LOG (SALES + INVENTORY)
Natural logarithm (base e)	LN (expression)	LN (SALES + INVENTORY)
Absolute Value	ABS (expression)	ABS (SALES - INVENTORY)
Square Root	SQRT (expression)	SQRT (AREA * 9)
Moving Average	MOVEAVG (variable,n)	MOVEAVG (SALES,3)
Cumulate (running sum)	CUMULATE (variable)	CUMULATE (SALES)

Table 5-3. Operators and Functions Used in Data Transformations

The operators and functions can operate on constants as well as variables. For example,

```
TARGET=4
```

Since transformations are processed in the order they are specified, you can use new variables within subsequent transformation expressions. For example:

```
TOTAL=SALES+INVENTORY
MONTHTOTAL=TOTAL/12
```

Here, SALES and INVENTORY have been defined as data variables, TOTAL is a new variable created as a a data transformation of SALES and INVENTORY, and MONTHTOTAL is a new variable defined in terms of TOTAL.

Note: MOVEAGE and CUMULATE cannot be nested.

If data cannot be transformed or has illegal results, the missing data value will be substituted for the result of the equation.

You can use the operators and functions in any combination. Parentheses can be used to override the standard order of precedence:

* functions and exponentiation performed from left to right

* multiplication and division performed from left to right

* addition and subtraction performed from left to right

Examples

```
ABS(PROFIT)/CUMULATE(SALES)
MOVEAVG(SALES+BACKLOG,2)
```

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

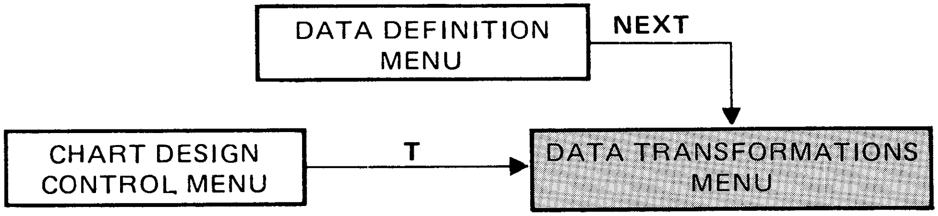
PREVIOUS (f5) Goes to DATA DEFINITION Menu.

NEXT (f6) Goes to CHART DESIGN CONTROL Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates the Help Facility, giving an explanation of the DATA TRANSFORMATION Menu.

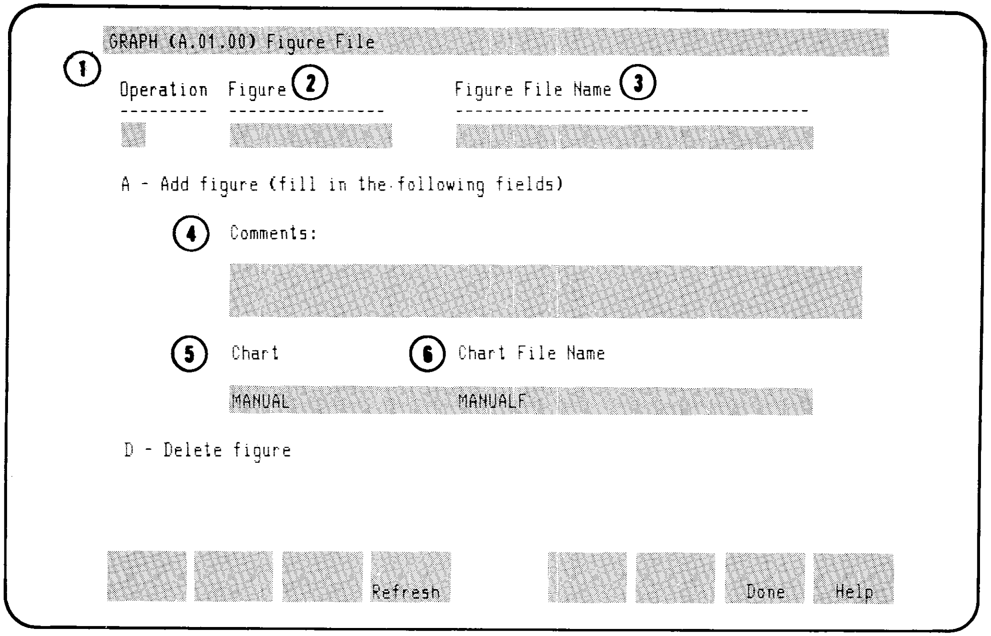
ROADMAP



When you enter a *T* on the CHART DESIGN CONTROL Menu, this menu comes up on your screen.

Alternatively, you can press NEXT from the DATA DEFINITION Menu.

Figure File Menu



```
GRAPH (A.01.00) Figure File

Operation  Figure                Figure File Name
---------  ----------------      ------------------------------------

A - Add figure (fill in the following fields)

      Comments:

      Chart                Chart File Name
      MANUAL               MANUALF

D - Delete figure
```

Figure 5-22. FIGURE FILE Menu

Using this menu, you create a figure file for access by other subsystems.

FIELD EXPLANATION

1. Operation
   A Add a figure to a figure file. If this is the first figure in a new figure file, you must create names for both the figure and the figure file. You need to fill in the chart or chart file names.

Chart and chart file names will already appear in the appropriate fields if you have filled them in on the MAIN CONTROL Menu.

D Delete a figure from the figure file. This operation will not delete the figure file when you have deleted all figures within it. To delete the figure file, use the :PURGE command in MPE. (See the *MPE Commands Reference Manual* for information on the PURGE command.)

2. Figure
   Name of the figure you are storing. The name must be from 1 to 16 characters in length beginning with a letter. Remaining characters can be letters, numbers or underbars (_). This field is required for creating the figure as well as accessing it from a subsystem.

3. Figure File Name
   Unique name of the file in which the figure is to reside. The figure file name is of the following form:

   filename[/lockword][.group[.account]]

   The filename and the lockword can each contain up to 8 alphanumeric characters, beginning with a letter. They cannot contain any special characters or embedded blanks. If you are accessing a file that resides in a different group or account, you must supply their names also. See the *MPE System Manager Reference Manual* for more information on lockwords and security.

4. Comment associated with the figure
   Enter any comment you wish. Maximum length is 128 characters -- 64 characters per line. Optional entry.

5. Chart
   Name of the current chart which you wish to have drawn to the figure file. It is automatically filled in from the MAIN CONTROL Menu.

6. Chart File Name
   Name of the current chart file in which the chart you are drawing resides. It is automatically filled in from the Main Control Menu.

DISCUSSION

This menu allows you to draw the current chart from the current chartfile to a figure file. Figure files provide storage for complete figures. (See Section 1 for a discussion of the use of figure files.)

Figure files may be accessed from TDP/3000, HPDRAW and other Office System applications as well as the HP2680A (Laser Printer). Figure files can contain multiple figures.

See Appendix E for information about accessing figure files.

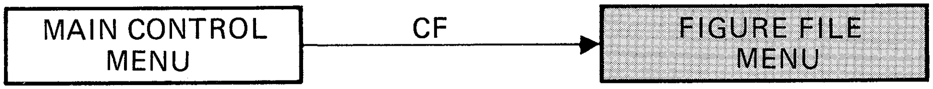
FUNCTION KEYS

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

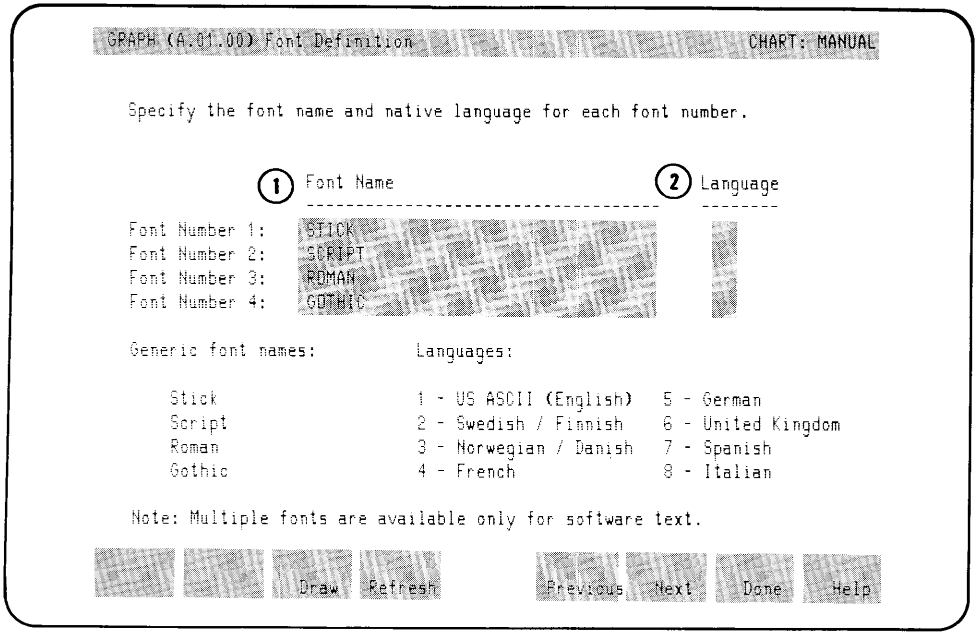
HELP (f8) Initiates Help Facility, giving an explanation of the FIGURE FILE Menu.

ROADMAP



To bring this menu to your screen you must enter *CF* (Create Figure) on the MAIN CONTROL Menu.

Font Definition Menu



```
GRAPH (A.01.00) Font Definition                          CHART: MANUAL

  Specify the font name and native language for each font number.

                   Font Name                          Language
                   -----------------------------     --------
  Font Number 1:   STICK
  Font Number 2:   SCRIPT
  Font Number 3:   ROMAN
  Font Number 4:   GOTHIC

  Generic font names:         Languages:

     Stick                    1 - US ASCII (English)   5 - German
     Script                   2 - Swedish / Finnish    6 - United Kingdom
     Roman                    3 - Norwegian / Danish   7 - Spanish
     Gothic                   4 - French               8 - Italian

  Note: Multiple fonts are available only for software text.
```

Figure 5-23. FONT DEFINITION Menu

With this menu you specify the font file name and native language for each font used in the current chart.

FIELD EXPLANATION

1. Font Name
   Four fonts may be specified under font name.

The four fonts currently available are:

```
   Font 1. Stick (Simplex Roman)
           (STICK.VCHARSET.SYS)
   Font 2. Script (Simplex Script)
           (SCRIPT.VCHARSET.SYS)
   Font 3. Roman (Triplex Roman)
           (ROMAN.VCHARSET.SYS)
   Font 4. Gothic (Gothic English)
           (GOTHIC.VCHARSET.SYS)
```

You need only enter a single font name in your specification-- Stick, Script, Roman, or Gothic-- following each of the Font Numbers.

The fonts are set in the above order by default. Unless you enter different font names, leaving font 1 blank will result in the Stick font appearing where font 1 is specified, Script where font 2 is specified, and so forth.

Stick, Script, Roman and Gothic are generic names for the fonts. If you have a file called, for example, STICK, you would access it using MPE conventions:

```
STICK.mygroup.myaccount
```

2. Languages

   Enter the number of the selected language you wish to associate with the font file you have chosen. You have a choice of 7 languages. They are:

   1 USASCII (English)
   2 Swedish/Finnish
   3 Norwegian/Danish
   4 French
   5 German
   6 United Kingdom
   7 Spanish

   The default language (blank) will usually be USASCII. Currently the foreign languages are only available with the Stick font (software only).

DISCUSSION

This menu allows you to define the software fonts -- maximum of four per chart -- which you wish to use with the chart you are creating. It also allows you to associate the native language with each font you have specified.

Only one hardware font is available and the type of font depends upon the device being used. Usually, it is a Stick font. The native language characteristics of the single hardware font are mapped by the language number you specify for Font Number 1 on this menu. (See the GRAPHING OPTIONS Menu for further information.)

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

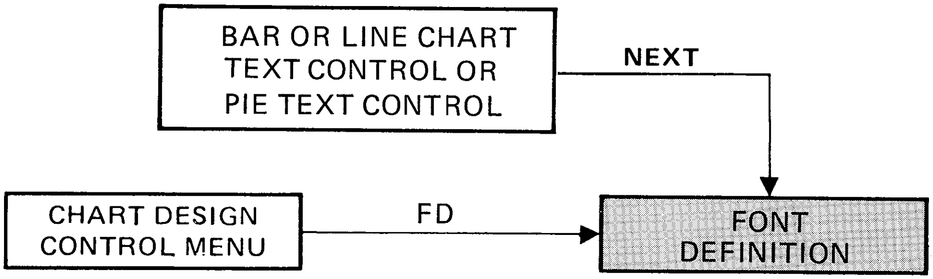
PREVIOUS (f5) Goes to PIE TEXT CONTROL or BAR AND LINE TEXT CONTROL Menu, dep pending on the type of chart you are drawing.

NEXT (f6) Goes to CHART DESIGN CONTROL Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

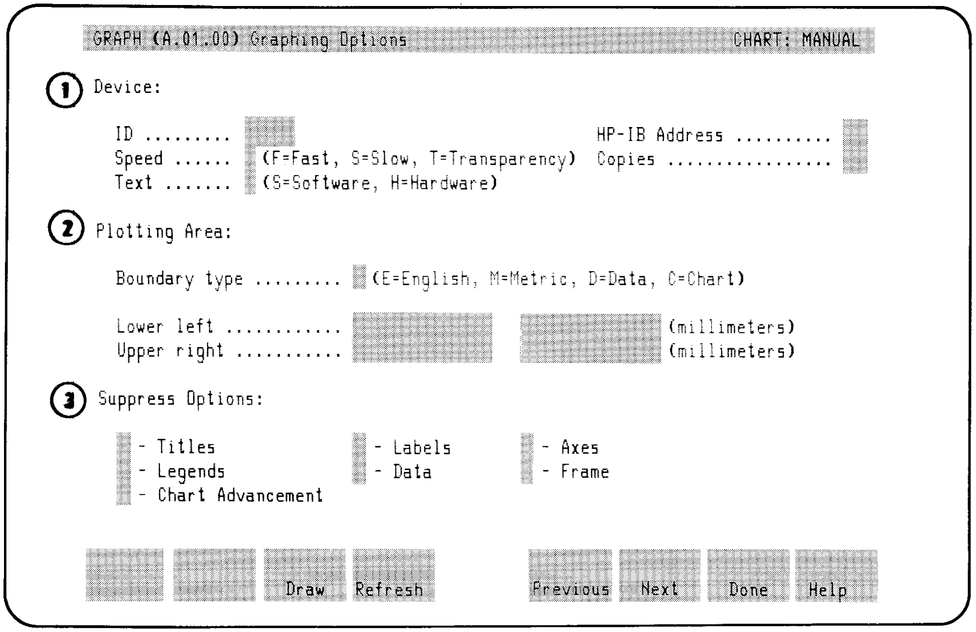
HELP (f8) Initiates Help Facility, giving an explanation of the FONT DEFINITION Menu.

ROADMAP


BAR OR LINE CHART TEXT CONTROL OR PIE TEXT CONTROL
NEXT
CHART DESIGN CONTROL MENU
FD
FONT DEFINITION

If you enter *FD* from the CHART DESIGN CONTROL Menu while creating a chart, this menu comes to the screen. Pressing NEXT from the BAR OR LINE CHART TEXT CONTROL or PIE TEXT CONTROL Menus will also display it.

Graphing Options Menu



```
GRAPH (A.01.00) Graphing Options                                  CHART: MANUAL

1 Device:

   ID .........                                    HP-IB Address ..........
   Speed ......   (F=Fast, S=Slow, T=Transparency) Copies ................
   Text .......   (S=Software, H=Hardware)

2 Plotting Area:

   Boundary type .........   (E=English, M=Metric, D=Data, C=Chart)

   Lower left ............                         (millimeters)
   Upper right ...........                         (millimeters)

3 Suppress Options:

     - Titles                - Labels        - Axes
     - Legends               - Data          - Frame
     - Chart Advancement
```

Figure 5-24. GRAPHING OPTIONS Menu

Use this menu to specify plotting device and the device's characteristics, number of copies, plotting area, and elements to be suppressed when the chart is drawn.

FIELD EXPLANATION

1. Device
   If you are running GRAPH from a graphics terminal this field is filled in with the terminal as the default plotting device.

A) ID
The model of the plotting device. Must be one of the following:

Terminals	2647A 2623A 2647F 2648A 2703
Two-color Plotter	7470A Opt. 001 (RS-232-C) 002 (HP-IB)
Four-Color Plotters	7220A/S 7221A/B/S 9872A/B/S
Eight-Color Plotters	7220C/T 7221C/T 7580A Opt. 001 (RS-232-C) 002 (HP-IB) 7585 9872C/T
Thermal Plotter Printer	7240A 7245A/B
Mini-Plotter	7225A/B (17601A,17603A, 17604A)
Laser Printer	2680A

Note that the 7220S/T, the 7221S/T, and the 9872S/T use scrollable paper. If you are using one of these plotters without scrollable paper, you should designate the ID as 7220A/C, 7221A/C, or 9872A/C.

B) HP-IB Address
If the plotting device is a 9872A/B/C/S/T, 7225A using the 1706A module, 7245A/B, 7470A (Opt.002), or 7580A Opt. 002, the HP-IB address set on the back of the device must be supplied. The default address on these devices is 5. If this field is blank when the device is a 9872A/B/C/S/T or a 7245A/B, GRAPH will supply an address of 5. If this field is blank and the device is a 7225A, 7470A, or 7580A, the field is left blank and non-HP-IB addressing is assumed.

C) Speed
This field refers to plotters only. Enter one of the following:

F Fast speed, appropriate for drawing on paper. (default)

S Slow speed, should be used when drawing on transparencies.

T Transparencies with 15 minute pauses between pen changes. (This assures that colors will have time to dry. The length of the pauses can be changed by the system manager.)

Note: For the 7580 and 7470, fast speed is set by pen carousel as is pen force.

D) Copies

Enter the number of copies of this chart to be drawn, from 1 to 99. This feature is useful for producing multiple copies of charts on scrollable plotters with automatic paper advance. Note that from the MAIN CONTROL Menu you can draw the whole chart file with pauses between charts for reloading non-scrollable plotters; you will get as many copies of each chart as you have specified on the GRAPHING OPTIONS Menu for that chart.

The default is one copy.

E) Text

Enter one of the following:

S Software. Text is scaled precisely by software.

H Hardware. Text is scaled and drawn much faster by hardware.

On plotters, software and hardware characters will be the same size. On terminals, hardware characters are larger and more legible than the software characters, but software characters show the correct proportion for hardcopy devices. Hardware characters on terminals may cause titles and footnotes to be truncated or written beyond the frame boundaries. Also, since hardware characters on terminals do not shrink, labels may overlay each other if the plotting area shrinks. If character size is a problem, use software characters.

Only one hardware font is available. The type is determined by the device used. Usually, it is the Stick font. The native language characteristics of the hardware font will be determined by the native language specified for Font 1 on the FONT DEFINITION Menu.

2. Plotting Area
On plotters, this is usually 8-1/2 inch by 11 inch horizontal format by default. To change the boundaries, give the coordinates for either the data (actual plotting area) or chart (including titles and footnote area). When the 7580 is used, the default is a D-size sheet of paper.

A) Boundary Type
Specify one of the following:

D Data boundary. This coincides with the axes on a line or bar chart. It may not be used for pie charts or slides. Data boundaries include areas required for legend text.

C Chart boundary. Coincides with the frame that can be drawn around the entire chart area, including titles and footnotes.

For pie charts, DSG/3000 may adjust the specified chart boundaries to ensure a pie chart of correct proportions.

E English standard. The entire charts fits on an area approximately 8-1/2 inch by 11 inch in size. This is the default value. The upper right and lower left fields are ignored for all devices except the 7580A.

M Metric standard. The entire chart fits on an area approximately 210mm by 298mm in size. The upper right and lower left fields are ignored.

B) Lower Left
Give coordinates (X,Y) of the lower left corner in millimeters. The x-coordinate (horizontal) is given first.

C) Upper Right
Give coordinates (X,Y) of the upper right corner in millimeters. The x-coordinate (horizontal) is given first.

Example

To draw a chart in the upper right hand quarter of an 11 inch x 17 inch (HP9280-0180) sheet of paper, specify the following Chart Boundary:

Lower Left	200,140
Upper Right	400,270

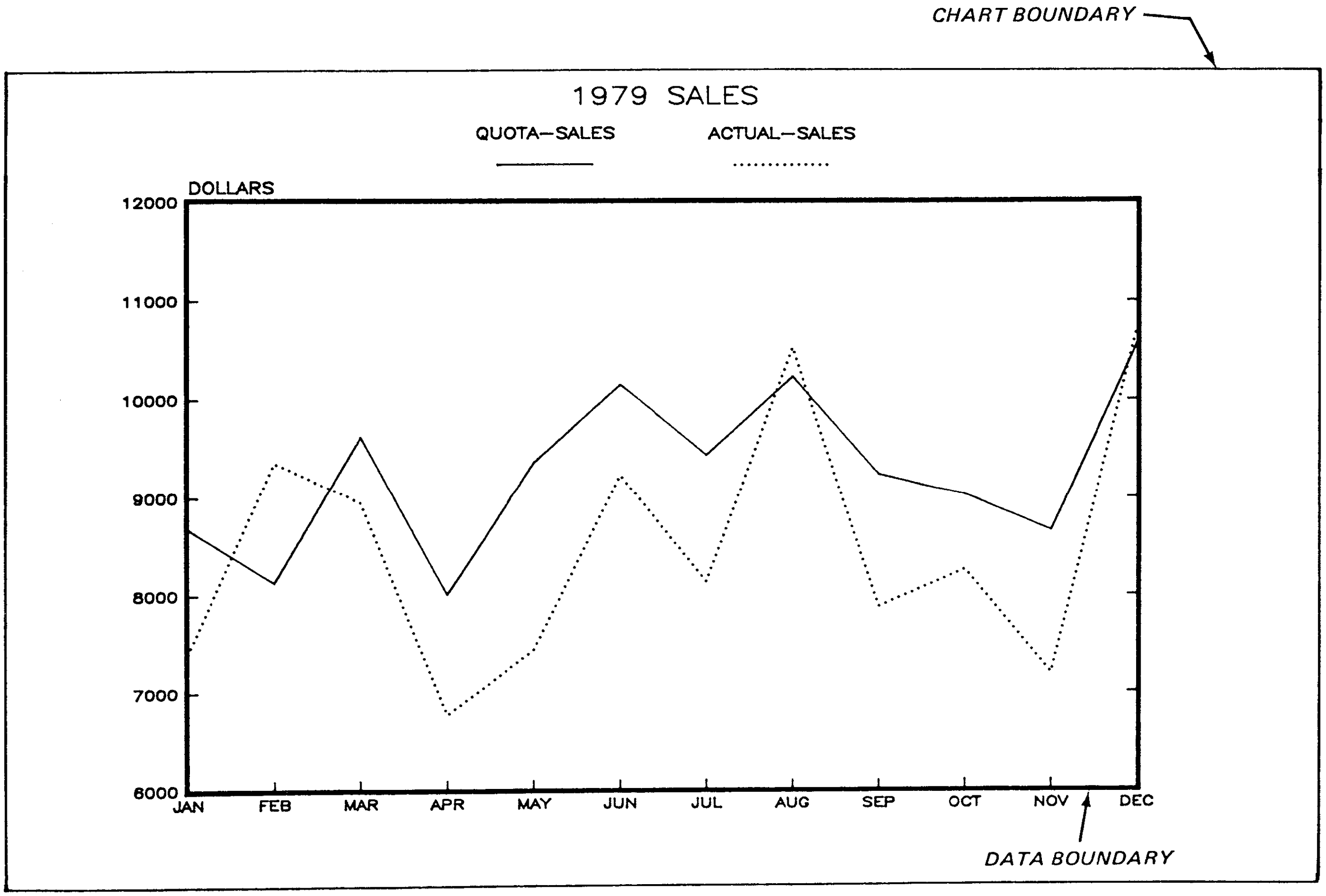


Figure 5-25. Chart and Data Boundaries

3. Suppress Options
If you want to suppress any of the chart attributes so that they will not be drawn with the rest of the chart, type any character in the highlighted field to the left of the appropriate option(s):

Titles

Anything entered on the PIE TITLES Menu or BAR AND LINE CHART TITLES Menu is suppressed. This includes main title, subtitle, footnotes, and axis titles.

Labels The labels on major ticks, bars, or pie segments are suppressed.

Axes The axes, including ticks and grid lines, are not drawn.

Legends The legends that are usually drawn on the top or side of bar and line charts are suppressed.

Data The actual bars or lines are not drawn on the chart. For pie charts, the pie circles and segments are not drawn.

Frame The frame is the box that encloses the whole graph including titles and footnotes. It is suppressed by default. (Blank out this field if you want the frame to be drawn.)

Chart Advance For scrollable plotters, the paper is not advanced; for graphics terminals, the graphics screen is not cleared. In either case, the current graph is drawn over the previous graph.

DISCUSSION

If you are using a supported HP graphics terminal, the terminal is the default plotting device and is listed as such when the GRAPHING OPTIONS Menu first comes up on your screen. If the device specified for the chart is another graphics terminal, the current graphics terminal will be used automatically. To redirect output, or to assign a plotting device from a non-graphics terminal, enter the appropriate device identification number; for the HP-IB devices you must also specify the HP-IB address.

If you do not specify the plotting area, DSG/3000 will scale the graph to fit your terminal screen, or to approximate an 8-1/2 inch x 11 inch page if the output device is a plotter. To override the size and placement, specify the plotting area in millimeters. Note that these menu specifications override any limit points that were set on the plotter itself with plotter keys P1 (or lower left) and P2 (or upper right). The specifications entered on the menu have the advantage of being stored with the chart, unlike the plotter limit points.

Any of the chart attributes listed below may be suppressed (not drawn) when the graph is drawn:

- titles
- legends
- labels
- data (lines or bars)
- axes
- frame (suppressed by default)
- advance paper (or clear video display)

If the output device is a scrollable plotter, suppressing the chart advance results in the current chart overlaying the one previously drawn. If the output device is a graphics terminal, the graphics display is not cleared before drawing the current chart, which also results in overlayed graphs.

If it is necessary to terminate plotting before it is completed, use <Control Y> rather than shutting off the plotter. This is done by pressing the Y key while the CNTL key is held down. It may be necessary to press the Y key more than once before control returns. Note that plotting may continue although control returns because of storage areas within the plotter itself.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

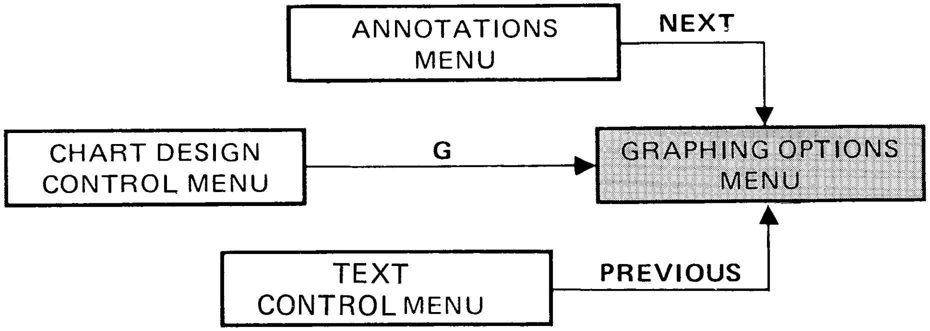
PREVIOUS (f5) Goes to ANNOTATIONS MENU.

NEXT (f6) Goes to TEXT CONTROL MENU for the chart type you are designing.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

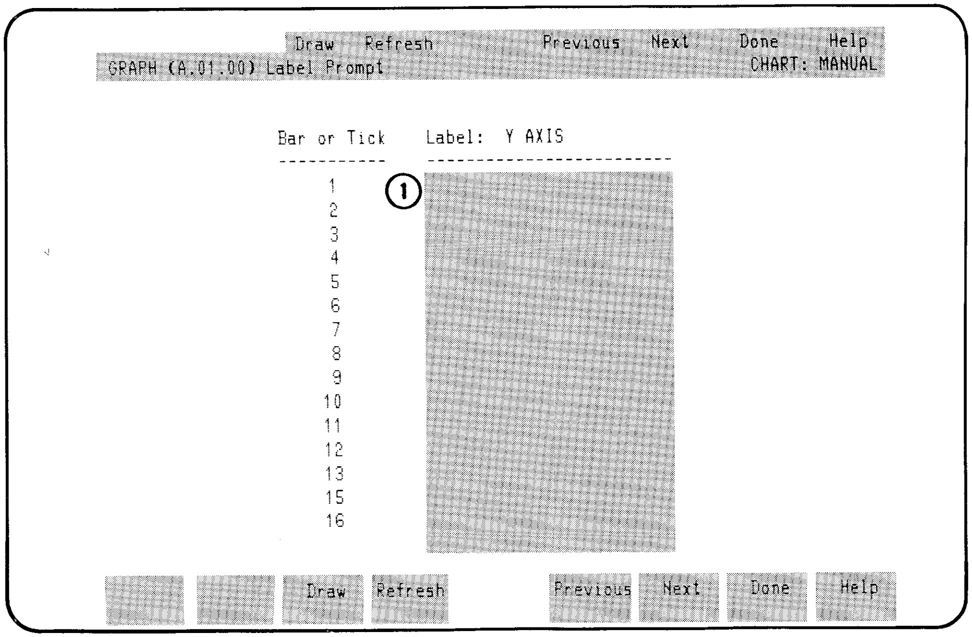
HELP (f8) Initiates Help Facility, giving an explanation of the GRAPHING OPTIONS Menu.

ROADMAP



When you enter a *G* on the CHART DESIGN CONTROL Menu, this menu comes up on your screen. Alternatively, you can press NEXT from the ANNOTATIONS Menu or PREVIOUS from the TEXT CONTROL Menus.

Label Prompt Menu



Label: Y AXIS

Figure 5-26. LABEL PROMPT Menu

Provides labels for bars on bar chart or or major ticks on either a bar or line chart. If this menu is used, the labels entered here override any labels contained in the data.

FIELD EXPLANATION

1. Labels
   Enter the labels as they correspond with the bars or major ticks along the axis. The first bar or tick is on the left (horizontal axis) or on the bottom (vertical axis).

Remember that the labels will repeat themselves if you enter fewer labels than bars or ticks.

DISCUSSION

When the LABEL PROMPT Menu comes up on your screen, the axis being labeled is identified as Y-axis or X-axis. If you want to supply labels for both axes, fill in the labels for the first axis, press ENTER, press NEXT (f6), fill in the labels for the second axis, and press ENTER again. Pressing NEXT one more time will bring you to a legends menu, while pressing PREVIOUS brings you back to the first LABEL PROMPT Menu.

The labels that you enter in this menu are stored in circular buffers. This means that if you enter fewer labels than there are ticks or bars along the axis, the labels are repeated in order until all of the ticks or bars are labeled.

The labels entered on this menu can be suppressed when the chart is actually drawn by filling in the GRAPHING OPTIONS Menu appropriately.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

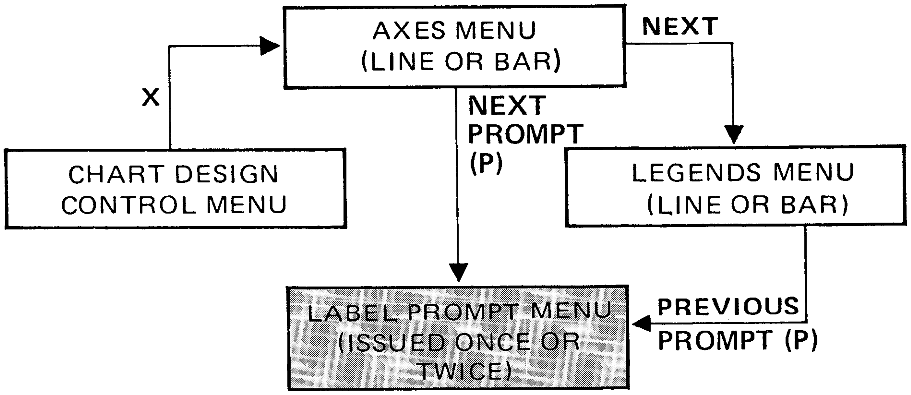
PREVIOUS (f5) Goes to the appropriate axes definition menu: LINE CHART AXES, BAR CHART AXES (NUMERIC), or BAR CHART AXES (TEXTUAL).

NEXT (f6) If you requested to be prompted for labeling both axes, this menu is repeated. Otherwise, goes to LINE LEGENDS or BAR CHART LEGENDS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the LABEL PROMPT Menu.

ROADMAP


AXES MENU (LINE OR BAR)
NEXT
X
NEXT PROMPT (P)
CHART DESIGN CONTROL MENU
LEGENDS MENU (LINE OR BAR)
LABEL PROMPT MENU (ISSUED ONCE OR TWICE)
PREVIOUS PROMPT (P)

To display this menu you must enter *P* for label type from the LINE AXES or BAR AXES (NUMERIC or TEXTUAL) Menus and press NEXT, or press PREVIOUS from a legends menu.

Line Chart Menu

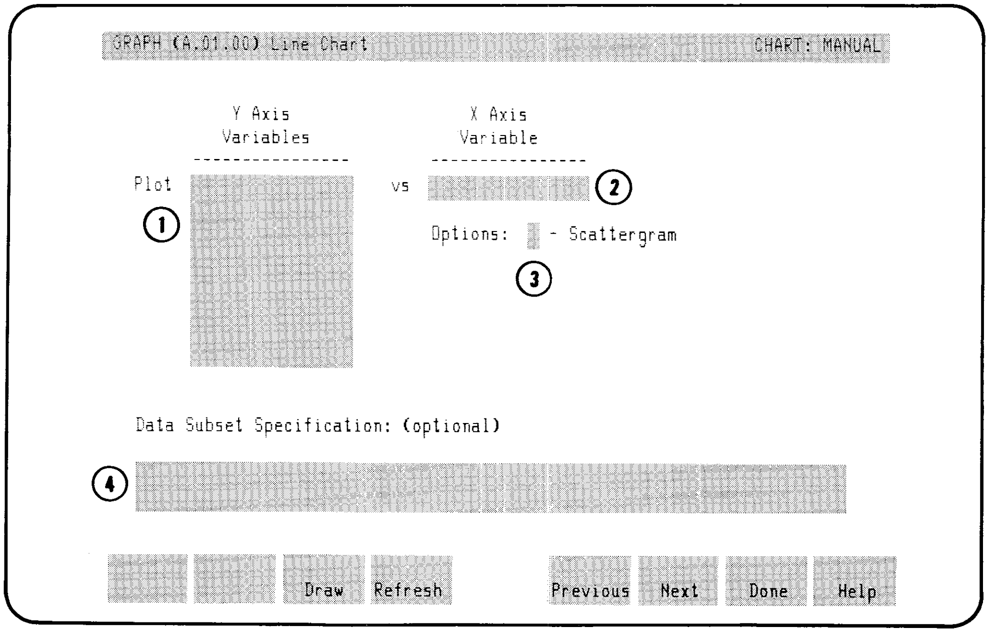


Figure 5-27. LINE CHART Menu

This menu names the variables to be plotted, selects a subset of the data, and specifies the scattergram option. Required for line charts and scattergrams.

FIELD EXPLANATION

1. Y Axis Variables (Required)
   Fill in the names of the Y-axis variables exactly as they appear in the DATA DEFINITION Menu or DATA TRANSFORMATIONS Menu. You must specify at least one and no more than 8 variables in this field.

2. X Axis Variable
   Fill in the name of the independent variable that will be plotted along the X-axis. Like the Y-axis variables, this variable name must also be defined in the DATA DEFINITION or DATA TRANSFORMATIONS Menus. If you do not fill in this field, GRAPH supplies an X-axis variable with values 1,2,3, and so forth, up to the number of data points taken by the Y-axis variables.

3. Options: Scattergrams
   Enter any letter in this field if you wish to produce a scattergram instead of a line chart. A scattergram plots data points without connecting lines and without sorting the X-axis variable.

4. Data Subset Specification
   If you want to graph a subset of the data file, enter a subset specification expression. In general, the expression has the form:

```
variablename relop value
 [{AND}[NOT] variablename relop value [...]]
  {OR }
```

where *relop* is a relational operator from Table 5-4. For example,

```
DEPT=SALES OR EXPENSES>800
```

is a subset expression where DEPT and EXPENSES are names of variables in the data file. The variables used to specify the subset are not necessarily those being graphed.

Parentheses can be used to override the standard order of precedence:

1. Relational operators processed from left to right.
2. NOT processed from left to right.

3 AND and OR processed from left to right.

If you are stating the expression in terms of a textual variable whose *value* contains embedded blanks, use quotation marks or apostrophes around the value. Use quotation marks around a value with an embedded apostrophe; likewise, use apostrophes around a value with embedded quotation marks.

Examples

```
DEPT="R$D" NAME='"X"'
POSITION="BOSS'S SON"
```

Table 5-4. Relational Operators for Subset Expressions

=	Equals
<>	Not equal
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

DISCUSSION

All line chart variables must contain numeric data and must be specified as numeric in the DATA DEFINITION Menu. There is only one X-axis variable plotted along the Y-axis, but there can be up to 8 Y-axis variables plotted along the X-axis. Each Y-axis variable corresponds to a line on the line chart.

Functions Keys

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

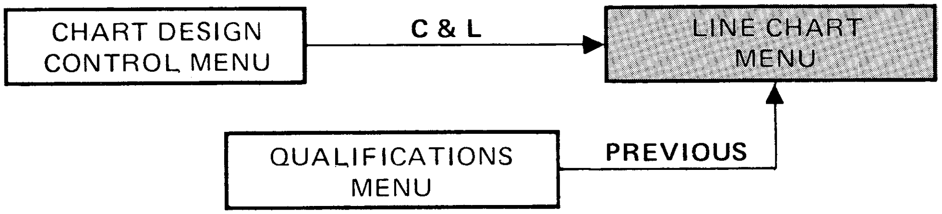
PREVIOUS (f5) Goes to CHART DESIGN CONTROL Menu.

NEXT (f6) Goes to QUALIFICATIONS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

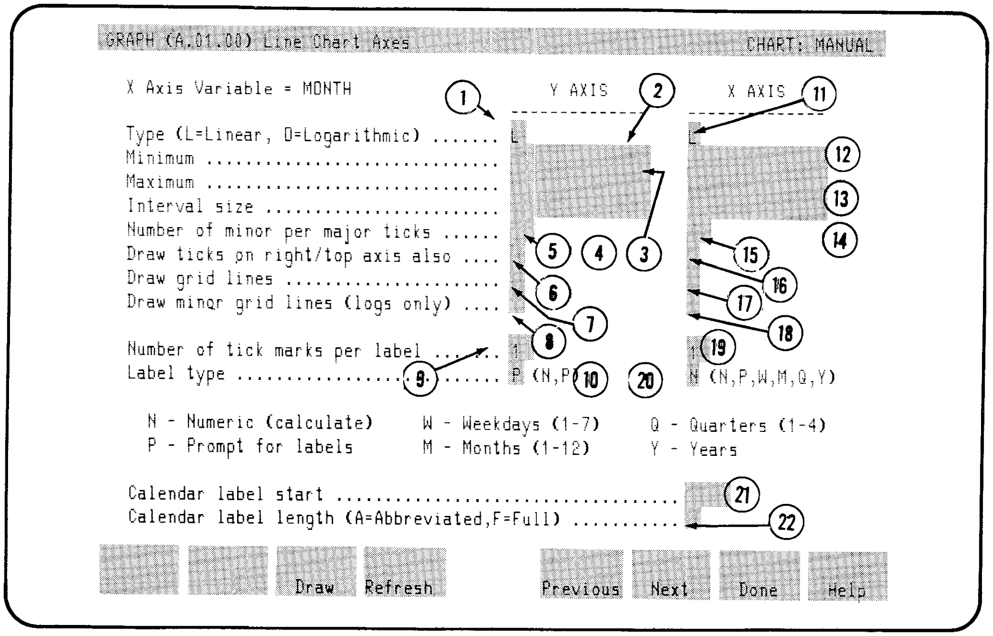
HELP (f8) Initiates Help Facility, giving explanation of the LINE CHART Menu.

ROADMAP



You can bring this menu to your screen by entering *C* and *L* on the CHART DESIGN CONTROL Menu or by pressing PREVIOUS from the QUALIFICATIONS menu if you are designing a line chart.

Line Chart Axes Menu



```
GRAPH (A.01.00) Line Chart Axes                                    CHART: MANUAL

   X Axis Variable = MONTH                     Y AXIS            X AXIS
                                            -------------     -------------
   Type (L=Linear, O=Logarithmic) ....... L                  L
   Minimum ..............................
   Maximum ..............................
   Interval size ........................
   Number of minor per major ticks ......
   Draw ticks on right/top axis also ....
   Draw grid lines ......................
   Draw minor grid lines (logs only) ....

   Number of tick marks per label ....... 1                  1
   Label type ........................... P (N,P            N (N,P,W,M,Q,Y)

     N - Numeric (calculate)     W - Weekdays (1-7)     Q - Quarters (1-4)
     P - Prompt for labels       M - Months (1-12)      Y - Years

   Calendar label start .................................
   Calendar label length (A=Abbreviated,F=Full) ..........

                  Draw  Refresh             Previous  Next   Done   Help
```

Figure 5-28. LINE CHART AXES Menu

Use this menu to specify the characteristics of the X- and Y-axes for a line chart menu, including: axis type, scaling, spacing and labeling of ticks, and grid lines.

FIELD EXPLANATION

1. Type
   The *L* that is entered in this field by default represents linear scaling. If you want logarithmic scaling instead, enter an *O*.

Logarithmic scaling may be useful if the data takes values over a very large range, or if you want to draw a linear plot of two variables that have an exponential relationship.

2. Minimum
   Enter the minimum value for the axis. If this field is left blank, DSG/3000 calculates the minimum from the data values.

3. Maximum
   Enter the maximum value for the axis. The maximum value must be greater than the minimum value. If you do not enter anything here, DSG/3000 calculates the maximum from the data values.

4. Interval Size
   Enter the number of units you want between major ticks in terms of the data. For example, if you want $100 between ticks, enter 100. If you do not fill this in, DSG/3000 calculates the units between ticks from the data in such a way that there are approximately 5 intervals along the axis. If Type is logarithmic, this field represents the number of powers of 10 which will separate major tick marks. For example, a value of 1 means that every power of 10 will have a tick mark.

5. Number of Minor per Major Ticks
   Enter the number of minor tick marks between major ticks. Minor ticks are drawn smaller than major ticks and are not labeled. If you do not fill in this field, there will be no minor ticks. For logarithmic charts, a common number of minor per major ticks is 8. Lowering this number results in a less dense axis or chart.

6. Draw Ticks on Right/Top Axis
   If you want the Y-axis ticks to be drawn on the right axis as well as the left one, enter any character in this field.

7. Draw Grid Lines
   If you want grid lines to be drawn wherever there are major tick marks, enter any character in this field.

8. Draw Minor Grid Lines (Log scaling only)
   This field is ignored unless you have specified:

   log scaling (field 1)
   grid lines (field 7)
   number minor ticks is >0 (field 5)

If these conditions are satisfied and you type any character in this field, grid lines are drawn at minor ticks as well as major ticks.

9. Number of Tick Marks per Label
If you enter a 1 in this field (or leave it blank) every major tick will be labeled. Enter the number of major ticks per label. For example, if you want to label every other tick, enter a 2, which means that there will be 2 ticks per label.

10. Label Type
If you want the major ticks to be labeled according to the axis scaling, enter an *N* in this field. This is also the default. If you want to supply your own labels which might be textual rather than numeric, enter a *P*; after you have pressed ENTER from this menu, you can enter the labels on the LABEL PROMPT Menu.

11. Type
Same as field 1, but refers to the X-axis.

12. Minimum
Same as field 2, but refers to the X-axis.

13. Maximum
Same as field 3, but refers to the X-axis.

14. Interval Size
Same as field 4, but refers to the X-axis.

15. Number of Minor per Major Ticks
Same as field 5, but refers to the X-axis.

16. Draw Ticks on Right/Top Axis also
If you want the X-axis ticks to be drawn on the top axis as well as the bottom one, enter any character in this field.

17. Draw Grid Lines

Same as field 7, but refers to the X-axis.

18. Draw Minor Grid Lines
Same as field 8, but refers to the X-axis.

19. Number of Tick Marks per Label
Same as field 9, but refers to the X-axis.

20. Label Type
Enter one of the following:

- N DSG/3000 calculates numeric labels from the axis scaling. (Default)
- P Use this option if you want to enter your own tick labels, which might be textual rather than numeric. After you have pressed ENTER and then NEXT, GRAPH issues the LABEL PROMPT Menu. (If you entered a *P* for both axes, you will fill in the LABEL PROMPT Menu twice.)

For convenience, GRAPH allows you to specify the following time labels regardless of the values of the X-axis variables. Fields 21 and 22 are used with time labels to give the starting point and length of the labels.

- W Weekdays. Each major tick will be labeled with a day of the week. If there are more than 7 tick marks, the labels will be repeated.
- M Months. Each major tick will be labeled with the name of a month. If there are more than 12 tick marks, the labels will be repeated.
- Q Quarters. Each major tick will be labeled with a quarter. If there are more than 4 tick marks, the labels will be repeated.
- Y Year. Each major tick will be labeled with a year.

21. Calendar Label Start (used only with time labels) Enter a number corresponding to the weekday, month, quarter, or year that labels the first (left -most) tick mark. The default is 1, which corresponds to MON for weekdays, JAN for months, QTR1 for quarters, or 1 for year. For example, if you want the ticks to be labeled by month beginning with November, type *11* in this field. If you want the ticks to be labeled by years beginning with 1982, for example, type *1982* in this field.

22. Calendar Label Length (used for time labels only)

A Abbreviated label form is printed on the graph. (Default)

L Long form is printed on the graph.

	Abbreviated	Long
Weekdays	M	MON
Months	J	JAN
Quarters	Q1	QTR1
Year	82	1982

DISCUSSION

This menu consists of two columns of fields. The first column (fields 1-10) refers to the Y-axis. The second column (fields 11-20) refers to the X-axis. The name of the X-axis variable appears in the upper left hand corner of the screen.

If you do not fill in this menu, the axes characteristics take the following defaults:

* linear axes
* axes scaled according to the data values
* major ticks with numeric labels calculated from the data. Y-axis is labeled from data; X-axis is labeled with 1,2,3, etc.
* no minor ticks or grid lines

Note that calendar labels are blindly applied to tick marks, irrespective of the actual data. To avoid mislabeling, the X-axis interval size should be set to 1 in most cases.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

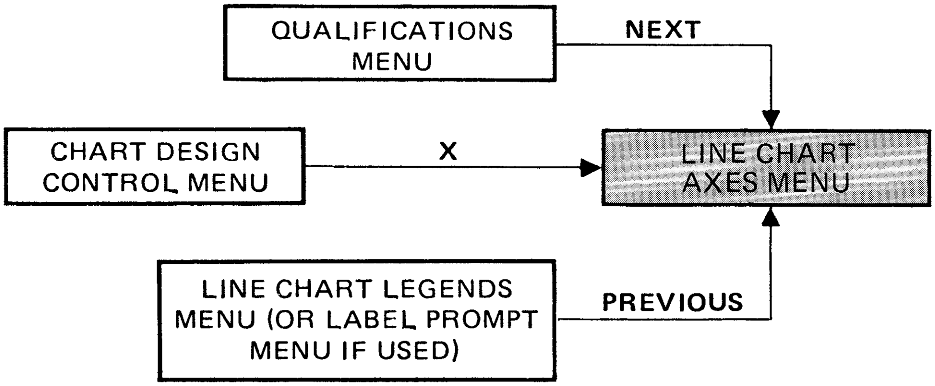
PREVIOUS (f5) Goes to QUALIFICATIONS Menu.

NEXT (f6) If a *P* (Prompt) is entered in field 7 or 10, goes to LABEL PROMPT Menu. Otherwise, goes to LINE CHART LEGENDS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the LINE CHART AXES Menu.

ROADMAP


QUALIFICATIONS MENU
NEXT
CHART DESIGN CONTROL MENU
X
LINE CHART AXES MENU
LINE CHART LEGENDS MENU (OR LABEL PROMPT MENU IF USED)
PREVIOUS

Press *X* from the CHART DESIGN CONTROL Menu while designing a line chart and this menu comes to your screen. You can also reach it by pressing NEXT from the QUALIFICATIONS Menu or PREVIOUS from the LABEL PROMPT Menu, if it was used, or the LINE CHART LEGENDS Menu.

Line Chart Legends Menu

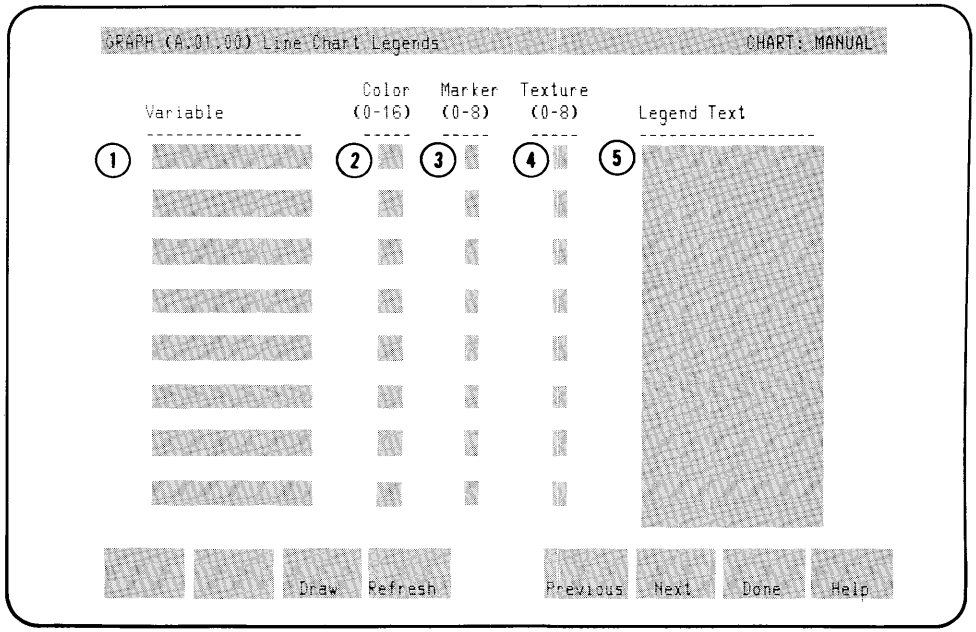


Figure 5-29. LINE CHART LEGENDS Menu

This menu specifies colors, markers, textures, and legend text for each line of a line chart.

FIELD EXPLANATION

1. Variable (display only)
   Each of the Y-axis variables is displayed here. These names, taken from the LINE CHART Menu, cannot be altered on the LINE CHART LEGENDS Menu.

2. Color
This field only affects a graph when it is drawn on a plotter with multiple pens. If you do not enter anything is this field, all of the lines in the line chart will be drawn in black. If you would like to assign colors to any of the lines in the graph, enter one of the following numbers next to the corresponding variable name:

1. Black
2. Red
3. Green
4. Blue
5...16

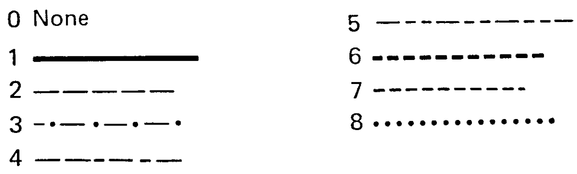
This presumes that the pens 1-4 are inserted in the plotter in this order. (See Appendix A for information about colors.)

3. Marker
If you do not enter anything in this field, no data markers are drawn on the graph unless the scattergram option has been specified. If you want to assign data markers for each of the data values taken by a particular variable, enter one of the following numbers next to the corresponding variable name:

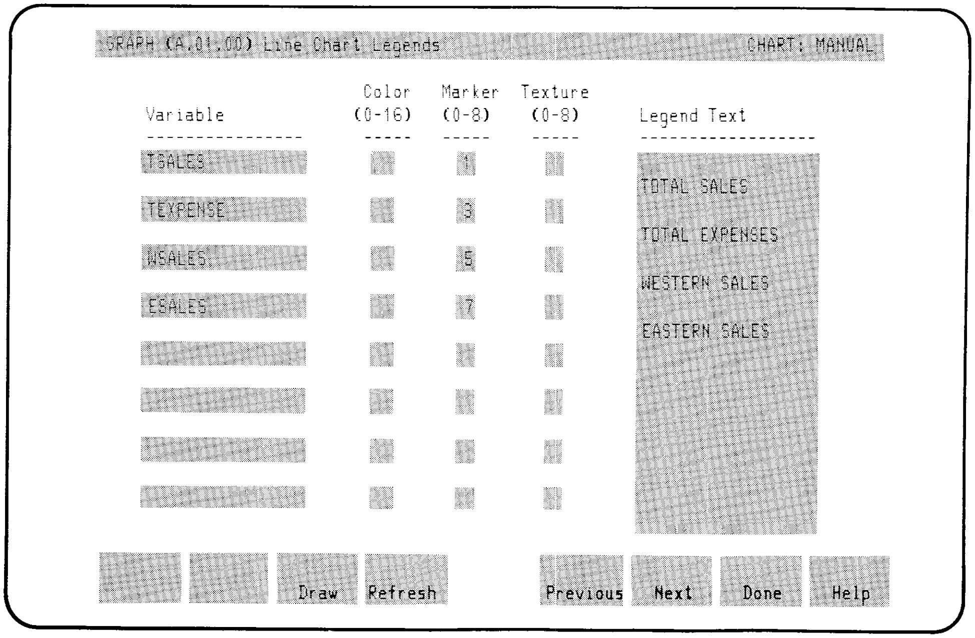
0 = none
1 = *
2 = +
3 = △
4 = □
5 = ◇
6 = ○
7 = ⊕
8 = x

See Figure 5-30B for an illustration of data markers.

4. Texture
If you do not enter anything in this field, each of the plotted lines is drawn with a different texture. The order of assignment of textures is shown in Appendix A. To change the texture of any of the lines, enter one of the following numbers next to the corresponding variable name:



5. Legend Text
   If you do not enter anything in this field, a legend is drawn on the graph identifying each line texture with the corresponding variable name from field 1. If you choose to modify the legend text for any of the variables, enter the new legend text here. The default size of legend text is 5. See Appendix A for examples of sizes.



GRAPH (A.01.00) Line Chart Legends CHART: MANUAL

Variable	Color (0-16)	Marker (0-8)	Texture (0-8)	Legend Text
TSALES		1		TOTAL SALES
TEXPENSE		3		TOTAL EXPENSES
WSALES		5		WESTERN SALES
ESALES		7		EASTERN SALES

Figure 30a. LINE CHART LEGENDS Menu
with Legend Text Entered

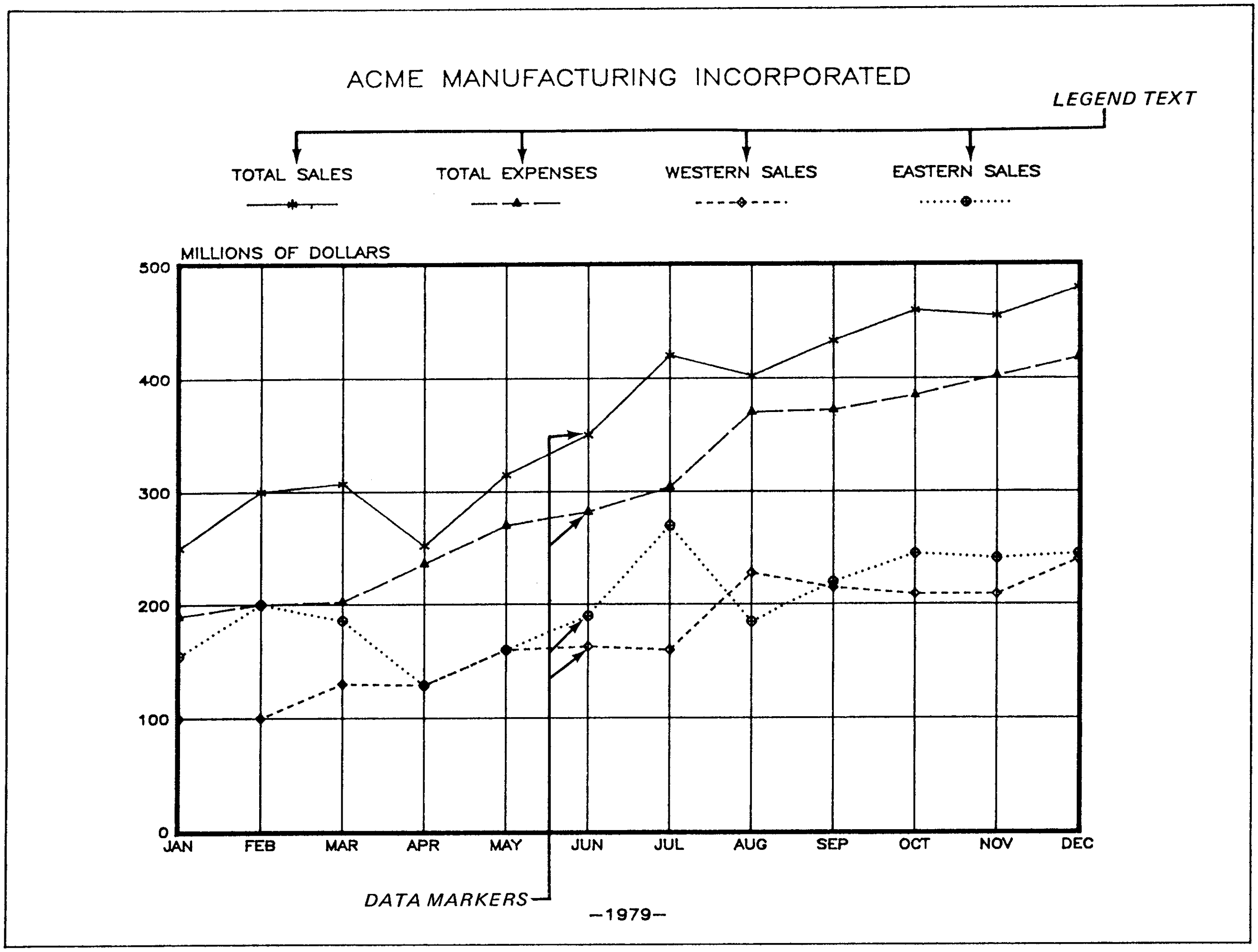


Figure 5-30b. LINE CHARTS Graph

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

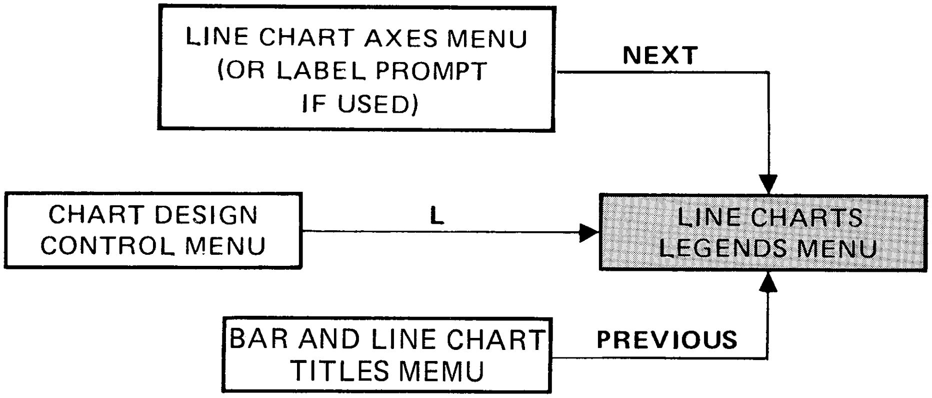
PREVIOUS (f5) Goes to LINE CHART AXES Menu or LABEL PROMPT, if used.

NEXT (f6) Goes to BAR AND LINE CHART TITLES Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the LINE CHART LEGENDS Menu.

ROADMAP


LINE CHART AXES MENU (OR LABEL PROMPT IF USED)
NEXT
CHART DESIGN CONTROL MENU
L
LINE CHARTS LEGENDS MENU
BAR AND LINE CHART TITLES MEMU
PREVIOUS

This menu comes to your screen when you enter an *L* on the CHART DESIGN CONTROL Menu while designing a line chart. You can also press NEXT from the LINE CHART AXES Menu or LABEL PROMPT, if used, or PREVIOUS from BAR AND LINE CHART TITLES Menu.

Main Control Menu

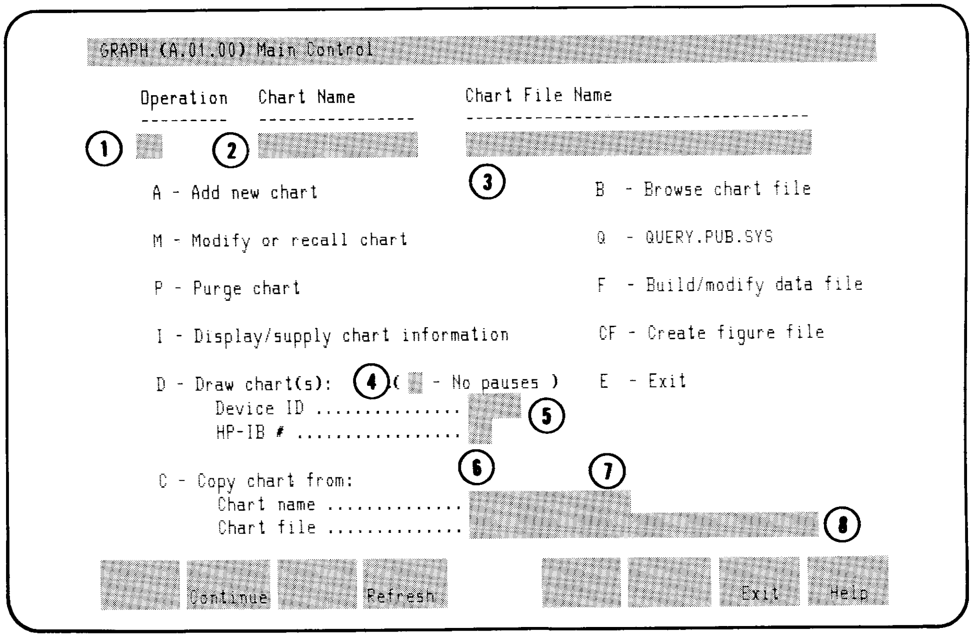


Figure 5-31. MAIN CONTROL Menu

The MAIN Menu provides entry to all GRAPH utilities and functions. You must complete this menu before going on to other screens.

FIELD EXPLANATION

1. Operations (Required)
   Must be one of the letters listed on the next page.

A Add new chart. Used to design a new chart. You must also enter a chart name and chart file name. Control is then transferred to the CHART DESIGN CONTROL Menu.

M Modify chart. Used to review or alter the characteristics of an existing chart. You must also enter a chart name and chart file name. Control is then transferred to the CHART DESIGN CONTROL Menu.

P Purge Chart. Permanently eliminates the chart from the chart file. You must also enter the chart name and chart file name.

I Chart information. Goes to CHART INFORMATION Menu. You must also enter a chart name and chart file name.

D Draw chart or chart file. If you enter the chart file name without a specific chart name, all of the charts in the file will be drawn. If you name a chart as well as the chart file name, only that particular chart is drawn. (See Step 4 for discussion of drawing.)

C Copy chart. Makes a copy of an existing chart. Can be used to change a chart name. Enter the existing chart and chart file name in fields 7 and 8, and enter the new chart and chart file name in fields 2 and 3. If you do not fill in the name of the existing file (field 8), it is assumed to be the same as the new file (field 3). The new chart file is created at this time if it does not exist.

B Browse chart file. Used to display the name and type of each of the charts in the file along with comments, if any. You must also enter a chart file name in field 3.

Q QUERY.PUB.SYS. Used to exit GRAPH and enter the QUERY/3000 subsystem. When you exit Query, you will be returned to the MAIN CONTROL Menu.

F Build or modify data file. Goes to DATA PROMPT Menu.

CF Create Figure File

Goes to FIGURE FILE Menu, which enables you to store figure information for access by other subsystems and certain intelligent devices.

E Exit. Used to terminate GRAPH.

2. Chart Name
Type in the chart name, which must be 1 to 16 characters in length beginning with a letter. Remaining characters can be letters, numbers or underscores(_). The chart name must be unique within the chart file. This field is required for adding, modifying, purging, displaying chart information, copying a chart, or drawing a specific chart in the chart file.

3. Chart File Name
Required unless building or modifying a data file or exiting from GRAPH. The chart file name is of the following form.

filename[/lockword][.group[.account]]

where the filename and the lockword can each contain up to 8 alphanumeric characters, beginning with a letter. They cannot contain any special characters or embedded blanks. If you are accessing a file that resides in a different group or account, you must supply their names also. See the *MPE System Manager Reference Manual* for more information on lockwords and security.

Examples

```
FILE1/SECURITY
PIEFILE
```

4. Draw with no Pauses
If you leave this field blank, the entire chart file will be drawn with pauses between charts. In this case, GRAPH draws each successive chart when you press CONTINUE (f2). If you want the charts to be drawn in succession without pauses, type a *Y* in this field. If you have entered a chart name in field 2, only that chart is drawn and field 4 is ignored.

5. Device Model Number
   If you do not fill in this field, each graph will be drawn on whatever device has been specified for it in the GRAPHING OPTIONS Menu. If you wish to override this device selection, enter the model number and letter of the desired output device. For example:

       7221S

6. HP-IB ADDRESS
   If an HP-IB device is named in field 5 above, enter its address here. If no address is given, it defaults to 5. However, if the device used is a 7225A, 7470A, or 7580A the HP-IB address (usually 5) must be explicitly entered if HP-IB addressing is desired. Otherwise, non-HP-IB addressing is assumed.

7. Chart Name (Required if copying a chart) The name of the chart that you are copying.

8. Chart File (Required if copying a chart from a different chart file than is named in field 3)
   The name of the file, including lockword if one exists, in which the chart to be copied resides. The default is the same chart file specified in field 3.

FUNCTION KEYS

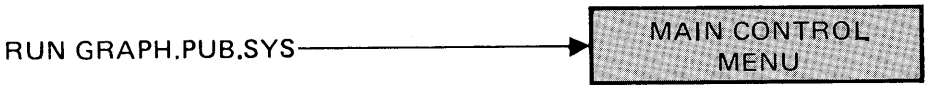
CONTINUE (f2) When drawing all of the chart in a chart file with pauses, press this key to go on to the next chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

EXIT (f7) Exits you from GRAPH.

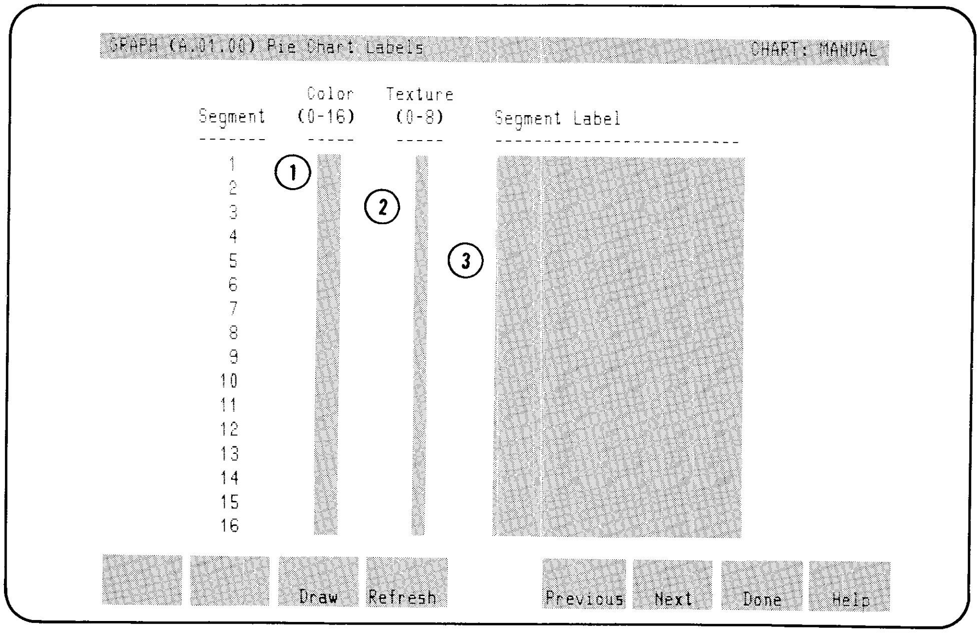
HELP (f8) Initiates Help Facility, giving an explanation of the MAIN CONTROL Menu.

ROADMAP



When you run GRAPH, this menu comes up on your screen. If you are in the process of designing or modifying a chart, control is shifted to the CHART DESIGN CONTROL Menu. If you wish to work on a different menu or exercise one of the GRAPH utilities, press DONE from the CHART DESIGN CONTROL Menu and you will be returned to this menu.

Pie Chart Labels Menu



Segment	Color (0-16)	Texture (0-8)	Segment Label
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Figure 5-32. PIE CHART LABELS Menu

Use this menu to specify colors, textures, and labels for each segment of a pie chart.

FIELD EXPLANATION

1. Color
   The graph is affected by this field only if the output device is a plotter with multiple pens. If you do not enter anything in this field, the graph will be drawn in black.

You can specify any of the following colors:

1. Black
2. Red
3. Green
4. Blue
5...16

This presumes that the pens 1-4 are inserted in the plotter in this order. (See Appendix A for information about colors.)

Remember that the colors are stored in a circular buffer, so that if you want the entire graph to be drawn with a blue pen, for example, you only need to specify a 4 in the first segment.

2. Texture
   If you do not enter anything in this field, the segments will be drawn without textures. Alternatively, you can enter an integer from 1 to 8 corresponding to the textures given in Appendix A. The textures are stored in a circular buffer.

3. Segment Label
   If you fill in this field, labels are printed next to each segment of the pie chart. The segment labels entered on this menu override labels taken from the data file if you specified a label variable on the PIE CHART Menu. You cannot take some labels from the label variable and some from the PIE LABELS Menu. Since the label buffer used by this menu is circular, if you enter fewer labels than there are segments, the labels are repeated.

   The segment labels can be up to 25 characters long, and can contain any combination of printable characters.

   The segment labels entered in this field can be suppressed when the chart is actually drawn by filling in the GRAPHING OPTIONS Menu appropriately.

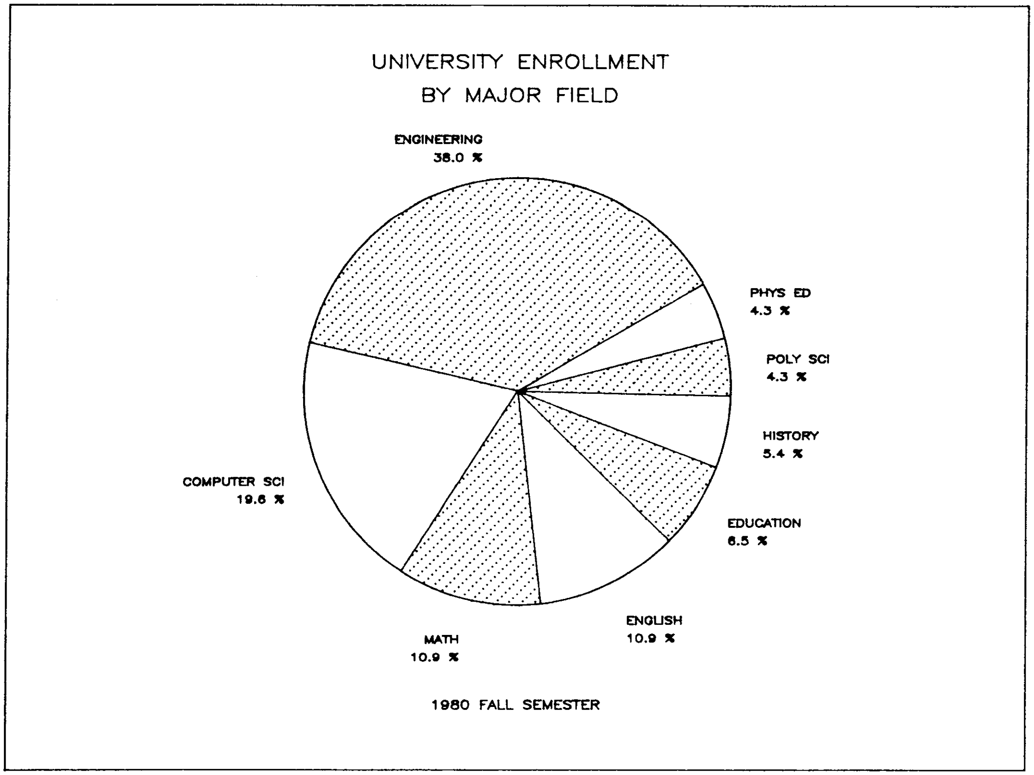


Figure 5-33. Results of Pie Labels Specifications

DISCUSSION

The colors, textures and segment labels are all stored in circular buffers. The length of the buffer is determined by the buffer with the most entries.

For example, if the Texture column contain 2 textures, say 1 and 2, but the Segment Label column contains 3 labels, the circular buffer length is 3 for all buffers. In this case, the textures will alternate 1, 2, default, 1, 2, default, and so forth.

For two textures to alternate, both columns must contain the same number of entries. To achieve this, set the Textures column to 1, 2, 1, 2, 1 and 2 and fill the Segment Label column with two sets of the 3 labels.

If you enter fewer textures than there are segments in the pie, the textures are repeated from the beginning in the same order until all of the segments have textures. Colors and labels are treated the same way. To illustrate, if a pie chart has 8 segments, the following specifications on the PIE LABELS Menu would produce these results.

Segment	Color	Texture	Segment Label
1		1	
2		2	
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

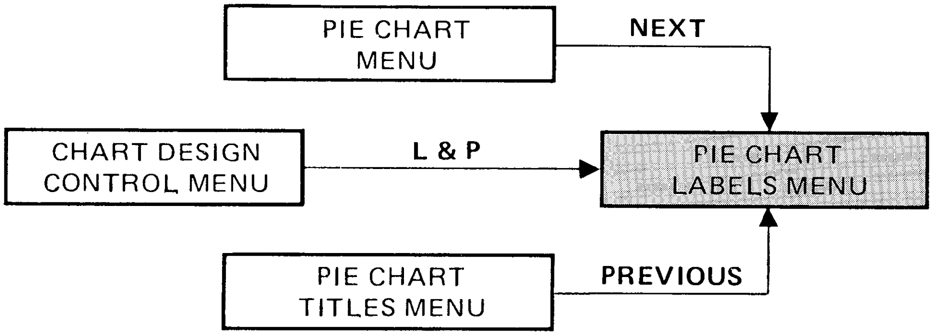
PREVIOUS (f5) Goes to PIE CHART Menu.

NEXT (f6) Goes to PIE CHART TITLES Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

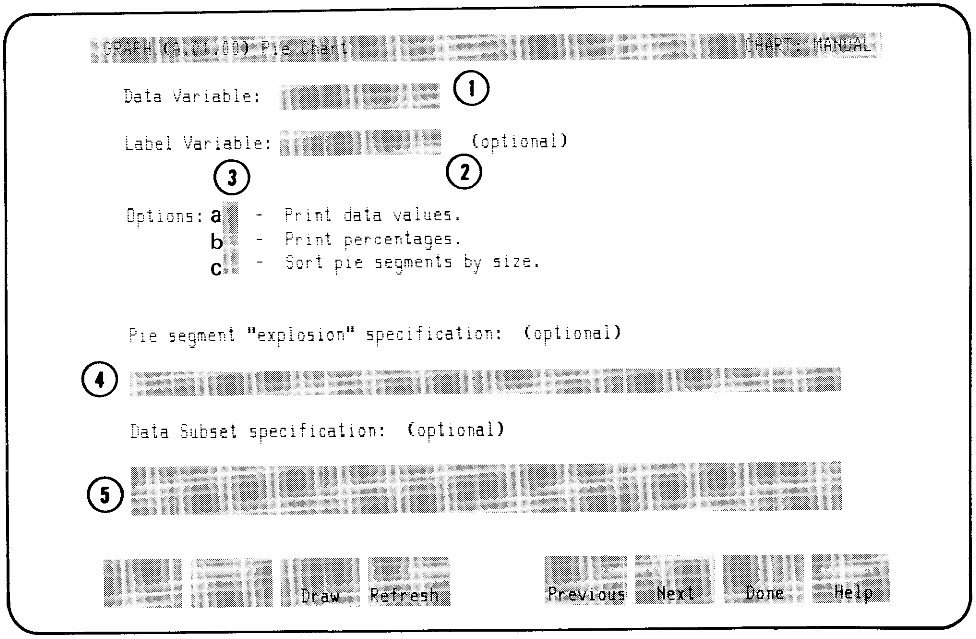
HELP (f8) Initiates Help Facility, giving an explanation of the PIE CHART LABELS Menu.

ROADMAP



To bring this menu to the screen, enter *L* on the CHART DESIGN CONTROL Menu when designing a pie chart. You can also press NEXT from the PIE CHART Menu or PREVIOUS from the PIE CHART TITLES Menu.

Pie Chart Menu



GRAPH (A.01.00) Pie Chart CHART: MANUAL

Data Variable:

Label Variable: (optional)

Options: a - Print data values.
b - Print percentages.
c - Sort pie segments by size.

Pie segment "explosion" specification: (optional)

Data Subset specification: (optional)

Figure 5-34. PIE CHART Menu

Use to specify variables, data subsets, and pie segment explosions. Also allows for sorting segments by size, with options to include data values or percentages with segments. This menu is required for pie charts.

FIELD EXPLANATION

1. Data Variable (Required)
   Type in name of the variable from your data file that contains the numeric values to be plotted. This is the same name that you gave the variable in the DATA DEFINITION or DATA TRANSFORMATIONS Menus.

If you need to return to the DATA DEFINITION Menu to recall the exact name of the variable, press DONE (f7) and enter *D* on the CHART DESIGN CONTROL Menu.)

2. Label Variable
If your data file has a textual variable that contains the labels for each segment, enter the variable name here as it appears in the DATA DEFINITION Menu.

3a. Options: Include Data Values
This box is filled in by default. If you do not want the actual data values to be printed by the segments, space over this field to blank it out.

3b. Options: Include Percentages
If you want to label each segment with its relative percentage, enter any character in this field.

3c. Options: Sort Segments in Ascending Order
Since this box is already filled in when the menu first comes up on your screen, GRAPH automatically sorts segments with the smallest segment starting in the two o'clock position when the graph is drawn, and proceeding in a clockwise direction. If you do not want the segments to be sorted, blank out this box by spacing over it or by pressing the DELETE CHAR key on your terminal; the segments will be drawn in the order of the values in the data file.

4. Explosion Specification
If you wish to draw attention to a particular segment by drawing it slightly removed from the rest of the pie, enter an explosion specification here. The expression specifies the segment in terms of the value of a data variable. In general, the explosion specification has the form:

$$\begin{array}{l}\textit{variablename relop value}\\ \left[\begin{Bmatrix}\text{AND}\\ \text{OR}\end{Bmatrix}[\text{NOT}]\ \textit{variablename relop value}\ [\ldots]\right]\end{array}$$

where *relop* is a relational operator from Table 5-5.

For example, consider this data file:

DEPT	EXPENSES
MARKETING	534
SALES	925
MFG	829
TRAINING	657
QA	432

The following expression would explode the segments corresponding to QA, SALES and MFG:

```
DEPT=QA OR EXPENSES>800
```

Parentheses can be used to override the standard order of precedence:

1. Relational operators processed from left to right
2. NOT processed from left to right
3. AND and OR processed from left to right.

If you are stating the expression in terms of a textual variable whose *value* contains embedded blanks, use quotation marks or apostrophes around the value. Use quotation marks around a value with an embedded apostrophe; likewise use apostrophes around a value with embedded quotation marks.

Examples

```
DEPT="R&D"
NAME=' "X" '
POSITION="BOSS'S SON"
```

5. Subset Specifications
   If you want to graph a subset of the data file, enter a subset specification expression. The form and allowable operators are the same as for Explosion Expressions given above. Note that if your data file contains more than 16 values for the data variable, you may want to select a subset of no more than 16 values for your graphing so your chart will be clear and easy to read.

Table 5-5. Relational Operators For Explosions and Subsets

Relational Operators	=	Equals
	<>	Not equal
	>	Greater than
	>=	Greater than or equal to
	<	Less than
	<=	Less than or equal to

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

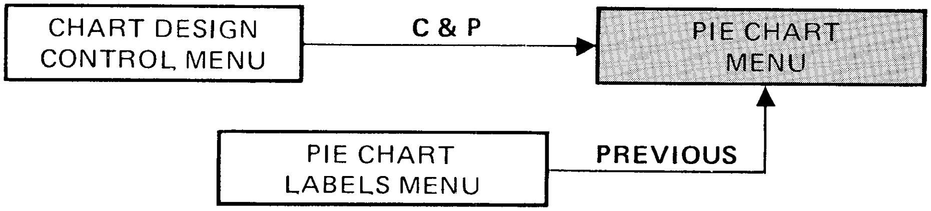
PREVIOUS (f5) Goes to CHART DESIGN CONTROL Menu.

NEXT (f6) Goes to PIE CHART LABELS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the PIE CHART Menu.

ROADMAP



To arrive at the PIE CHART Menu, enter a *C* and *P* (Pie Chart) on the CHART DESIGN CONTROL Menu.

Alternatively, you can press PREVIOUS from the PIE CHART LABELS Menu.

Pie Chart Titles Menu

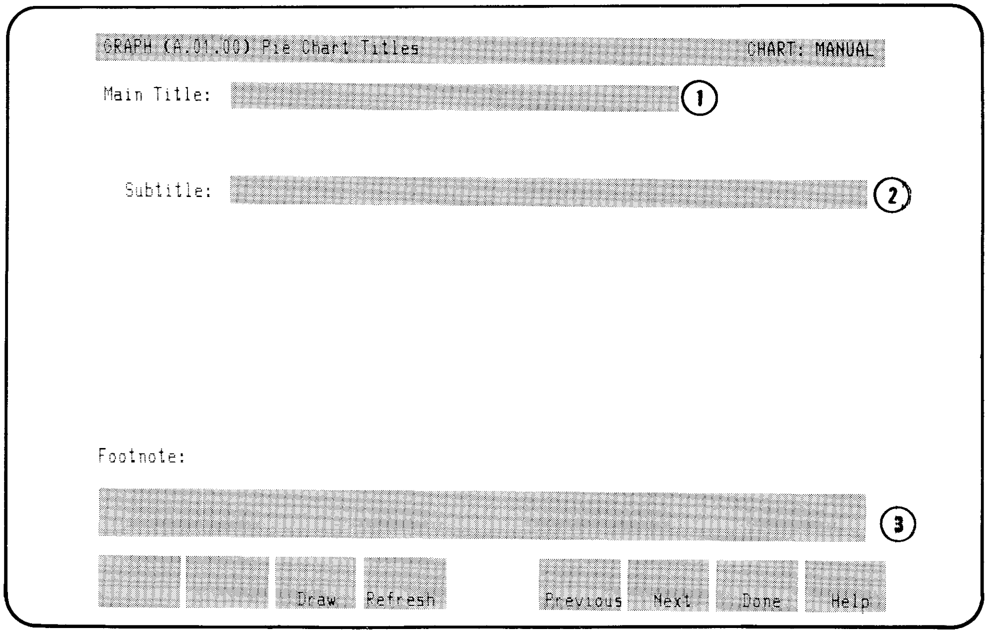


Figure 5-35. PIE CHART TITLES Menu

Use this menu to specify title, subtitle, and footnote for pie charts.

FIELD EXPLANATION

1. Title
   The title may be any combination of up to 45 printable characters.

2. Subtitle
   This subtitle is printed in the same text size as the title, and contains up to 64 printable characters.

3. Footnote
   The footnote may be 1 or 2 lines, each containing up to 78 printable characters.

DISCUSSION

On this menu, you may enter any or all of the following:

- a one-line title
- a one-line subtitle, using a text size that is the same as the title or smaller
- a one or two-line footnote

In every case, GRAPH will left-justify the text that you enter and center it when it is actually printed on the graph.

If the size of the text is too large, characters may be truncated, depending on the exact text positioning.

Any of the titles entered in this menu can include any of the following special commands:

$TODAY Represents the current date in the form mm/dd/yy with leading zeros suppressed.
For example, 9/15/82.

$MONTH Represents the current month spelled out in full. For example, SEPTEMBER.

$M Represents the current month numerically. For example, 9.

$DAY Represents the current numeric day of the month. For example, 15.

$YEAR Represents the current year in 4 digits. For example, 1982.

$Y Represents the current year by its last two digits. For example, 82.

For example, you might specify this subtitle:

THIS GRAPH WAS PRODUCED ON $TODAY

When the graph is drawn, the subtitle would be:

THIS GRAPH WAS PRODUCED ON 9/15/82

The titles entered on this menu can be suppressed when the chart is actually drawn by filling in the GRAPHING OPTIONS Menu appropriately.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

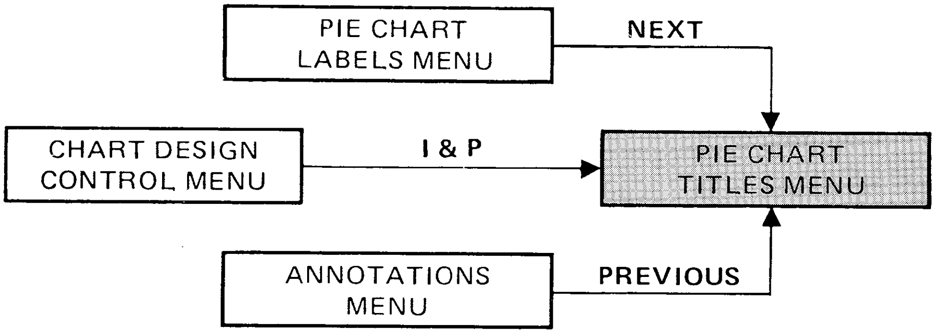
PREVIOUS (f5) Goes to PIE CHART LABELS Menu.

NEXT (f6) Goes to ANNOTATIONS Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the PIE CHART TITLES Menu.

ROADMAP



This menu appears on your screen when you enter *TI* on CHART DESIGN CONTROL Menu, when you press NEXT from the PIE LABELS Menu, or when you press PREVIOUS from the ANNOTATIONS Menu.

Pie Text Control

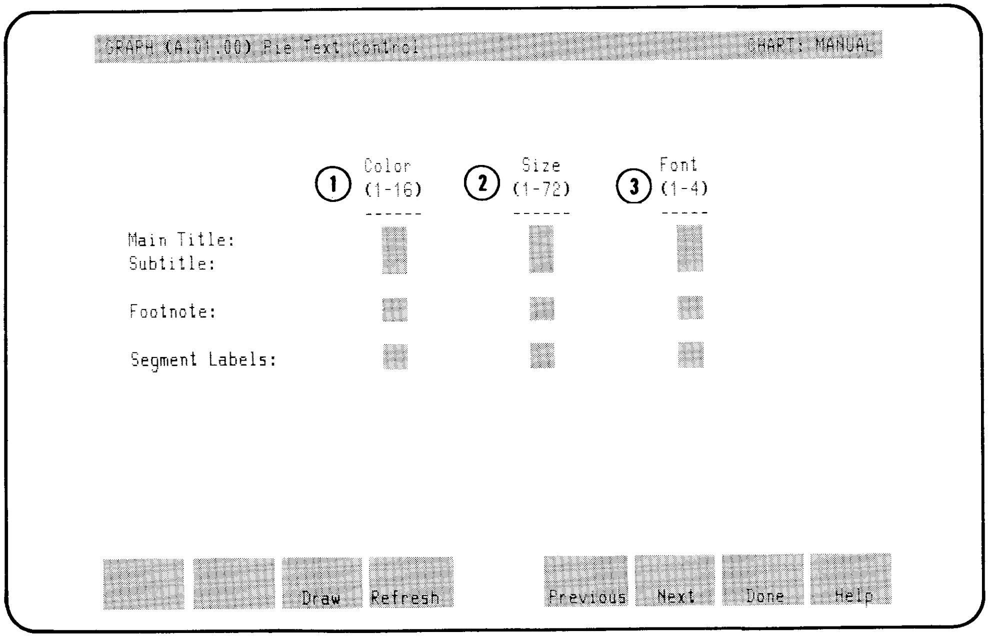


Figure 5-36. PIE TEXT CONTROL Menu

Use to specify color, size and font for text (titles, subtitles, labels and footnotes, but not annotations) on a pie chart.

Field Description

1. Color
   This field has meaning only when the graph is drawn on a color device. Enter 0-16 or blank as shown on the next page. (Entering 0=1.)

1 Black (default)
2 Red
3 Green
4 Blue
5...16

This presumes that the pens 1-4 are inserted in the plotter in this order. (See Appendix A for information about colors.) The maximum number that can be entered is 16. The default (blank) is 1.

2. Size
Enter a number from 1 to 72 where 1 is the smallest text size and 72 is the largest. The defaults are:

Main Title	8
Subtitle	6
Segment labels	4

On plotters, software and hardware characters will be the same size. On terminals, hardware characters can be larger than the software characters. Refer to the GRAPHING OPTIONS Menu for a discussion of hardware and software characters.

Note: Size of text will reflect chart size changes and size is relative rather than absolute. If you have reason to shrink or stretch the size of a chart, you may wish to adjust the size specifications for your text to ensure legibility.

If the text size is changed, it may cause text to overwrite other areas of the chart. It is recommended that you experiment to find the right text size for your charts.

3. Font
Enter the numbers of the fonts you have defined for the chart using the FONT DEFINITION Menu. Only four software fonts may be defined for a single chart. The default (blank) is font 1.

Only one hardware font, usually Stick, is available. The native language characteristics will be that specified for font 1 on the FONT DEFINITION Menu.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

REFRESH (f4) Refreshes the screen by redrawing the menu with the specifications for color, size, and font as of the last ENTER.

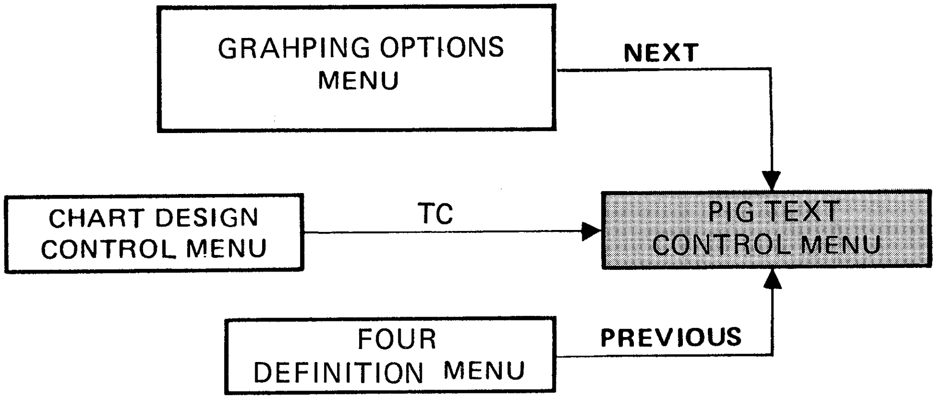
PREVIOUS (f5) Goes to GRAPHING OPTIONS Menu.

NEXT (f6) Goes to FONT DEFINITION Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the PIE TEXT CONTROL Menu.

ROADMAP



This menu appears on your screen if you enter *TC* on the CHART DESIGN CONTROL Menu while designing a pie chart. It will also appear if you press PREVIOUS from the FONT DEFINITION or NEXT from the GRAPHING OPTIONS Menus.

Qualifications Menu

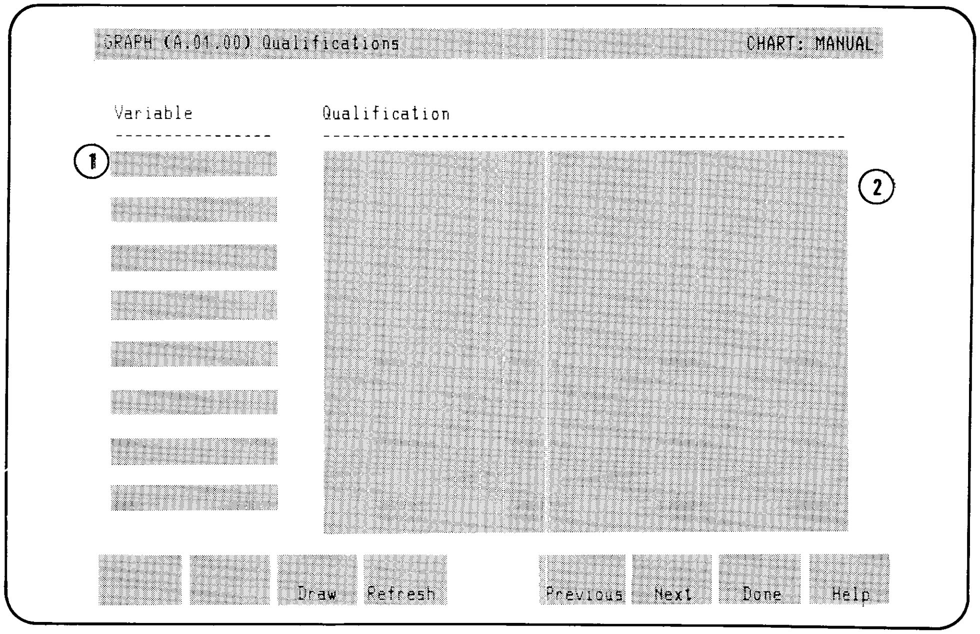


Figure 5-37. QUALIFICATIONS Menu

Use this menu to select a particular subset of the data for each Y-axis variable being graphed. Applies to bar charts, line charts, and scattergrams.

FIELD EXPLANATION

1. Variable (display only)
   Each of the Y-axis variables is displayed here. This information, taken from the LINE CHART or BAR CHART Menu, cannot be altered on the QUALIFICATIONS Menu.

2. Qualifications
   Enter a qualifying expression for any variable that you want to qualify. The general form of a qualifying expression is the same as a subset expression:

```
variablename relop value
  [ {AND} [NOT]  variablename relop value [...]]
    {OR }
```

   where *relop* is a relational operator from Table 5-6. For example,

   COMPANY=ACME AND SALES<=760

   is a qualifying expression where COMPANY and SALES are variables defined in the data file.

Parentheses can be used to override the standard order of precedence:

1. Relational operators processed from left to right
2. NOT processed from left to right
3. AND and OR processed from left to right

If you are stating the expression in terms of a textual variable whose *value* contains embedded blanks, use quotation marks or apostrophes around the value. Use quotation marks around a value with an embedded apostrophe; likewise use apostrophes around a value with embedded quotation marks.

Examples

```
DEPT="R & D"
NAME=' "X" '
POSITION="BOSS'S SON"
```

Table 5-6. Relational Operators for Qualifying Expressions

=	Equals
<>	Not equal
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

DISCUSSION

In the first field, GRAPH displays the names of the Y-axis variables that you have declared on the BAR CHART or LINE CHART Menu. You can enter a data qualifying expression for each of the variables listed.

By using qualifying expressions, you can select a different subset of each variable being graphed if you wish. If instead you want to subset the entire data file according to the value of one variable, specify a data subset on the LINE CHART or BAR CHART Menu.

To use variable qualifications, enter the name of the data variable on the LINE or BAR CHART Menu as many times as you wish to qualify it. For example, if you want to break the data variable NUM-EMP into four separate variables, enter NUM-EMP four times as a horizontal variable. Then press NEXT (f6) to arrive at the QUALIFICATIONS Menu, where GRAPH lists NUM-EMP four times for you. When you specify a different qualifying expression for each occurrence of NUM-EMP, the resulting graph will be drawn with four dependent variables. Figure 5-38 shows a data file created through the DATA PROMPT Menu, a LINE CHART Menu, and a QUALIFICATIONS Menu filled in as an example. Figure 5-39 shows the resulting line graph.

FUNCTION KEYS

DRAW (f3) Draws the current chart.

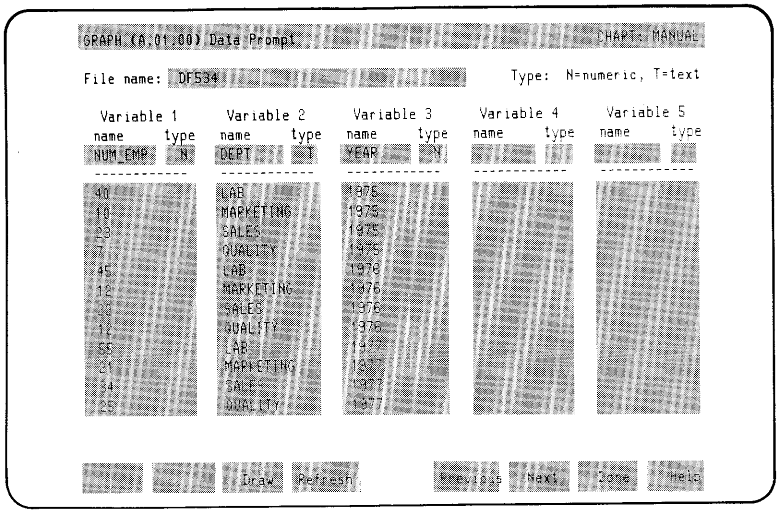
REFRESH (f4) Refreshes the screen by redrawing it with the information entered as of the last ENTER.

PREVIOUS (f5) Goes to LINE CHART or BAR CHART Menu

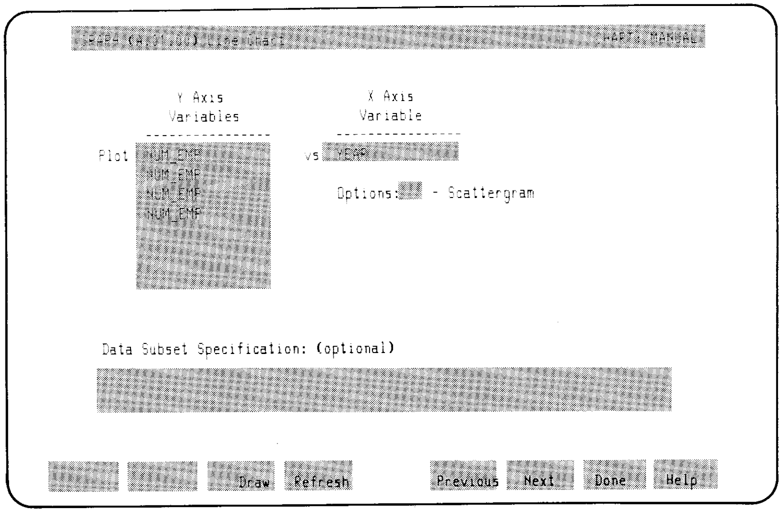
NEXT (f6) Goes to LINE CHART AXES or BAR CHART AXES Menu.

DONE (f7) Goes to CHART DESIGN CONTROL Menu.

HELP (f8) Initiates Help Facility, giving an explanation of the QUALIFICATIONS Menu.

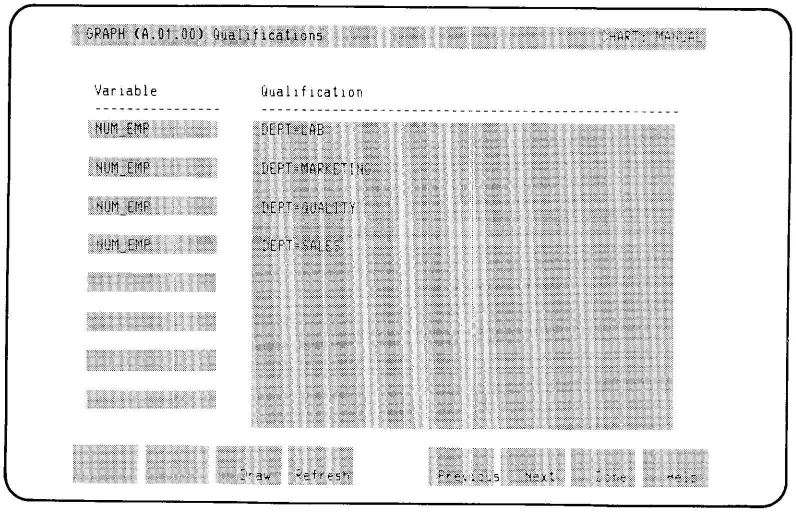


```
GRAPH (A.01.00) Data Prompt                                      CHART: MANUAL

File name: DF534                                  Type:  N=numeric, T=text

 Variable 1        Variable 2       Variable 3       Variable 4       Variable 5
name     type     name     type    name     type    name     type    name     type
NUM_EMP   N       DEPT      T      YEAR      N
------------      ------------     ------------     ------------     ------------
40                LAB              1975
10                MARKETING        1975
23                SALES            1975
7                 QUALITY          1975
45                LAB              1976
12                MARKETING        1976
22                SALES            1976
12                QUALITY          1976
55                LAB              1977
21                MARKETING        1977
34                SALES            1977
25                QUALITY          1977
```



```
GRAPH (A.01.00) Line Chart                                       CHART: MANUAL

            Y Axis                 X Axis
          Variables               Variable
       ----------------       ----------------
Plot   NUM_EMP             vs YEAR
       NUM_EMP
       NUM_EMP                Options:    - Scattergram
       NUM_EMP

Data Subset Specification: (optional)
```

Figure 38a. Qualifications Example on DATA PROMPT and LINE CHART Menus



GRAPH (A.01.00) Qualifications CHART: MANUAL

Variable	Qualification
NUM_EMP	DEPT=LAB
NUM_EMP	DEPT=MARKETING
NUM_EMP	DEPT=QUALITY
NUM_EMP	DEPT=SALES

Figure 5-38b. Qualifications Example

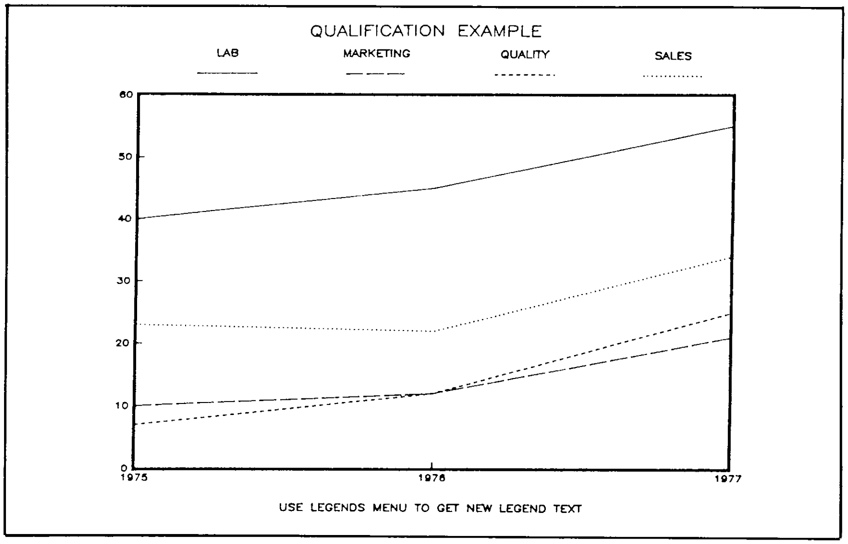
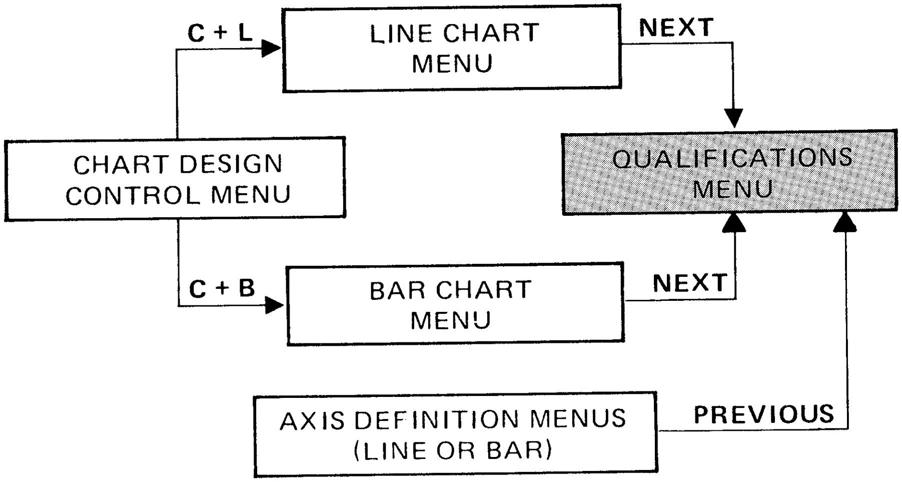


Figure 5-39. Line Chart Drawn Using Variable Qualifications

ROADMAP



To bring up this menu, press NEXT from the LINE CHART or BAR CHART Menus. You can also press PREVIOUS from LINE AXES or BAR AXES Menus.

SECTION 6

Using DSG/3000 Programmatically

Overview

A set of programmatically callable library procedures is provided by DSG/3000. All of the capabilities of DSG/3000 are accessible programmatically, including chart design and execution. The procedures can be called from any user-written program in COBOL, FORTRAN, BASIC, SPL, or PASCAL.

DSG/3000 functions and capabilities are specified by procedures that can be divided into 4 areas:

* Data Procedures
Set and interrogate data file descriptions, data transformations, subsets, and qualifications.

* Chart Procedures
Set and interrogate titles, axes descriptions, labels, legends, annotations, fonts, chart type, and unique characteristics.

* Graphing Procedures
Process data, initialize graphing device, and draw graph.

* Chart File Management Procedures
Create and access chart file, return information about contents of chart file, copy or remove charts from chart file, set and interrogate comments about charts, and store and delete figures in figure files.

Table 6-1 lists the DSG/3000 procedures by function with summary information. Many procedures are given in pairs, where one procedure sets the characteristics, and the other returns information about what has been set. For example, GDATAFILESET sets the data file name, and GDATAFILEINQ returns the data file name. The shorthand notation used in this table is

```
GDATAFILE{SET}
         {INQ}
```

A full description of each procedure appears in Section VII arranged alphabetically for easy reference.

Table 6-1. Summary of DSG/3000 Procedures

PROCEDURE	FUNCTION	PROCEDURE	FUNCTION
DATA PROCEDURES		CHART PROCEDURES	

GDATAFILE{SET}
{INQ}
	Sets or returns data file name and value for missing data.	GCHARTYPE{SET}
		{INQ}
			Sets or returns the chart type.

GDEFN{SET}
{INQ}
	Sets or returns definitions of all variables in the data file.	GBAR{SET}
		{INQ}
			Sets or returns unique bar chart characteristics including bar axis scaling. Not callable from COBOL.

GTRAN{SET}
{INQ}
	Sets or returns variable transformations.	GBARA{SET}
		{INQ}
			Same as above, but callable from COBOL as well.

GSUB{SET}
{INQ}
	Sets or returns data file subset specifications.	GPIE{SET}
		{INQ}
			Sets or returns unique pie chart characteristics.

GINDVAR{SET}
{INQ}
	Sets or returns x axis (or independent) variable.	GAXIS{SET}
		{INQ}
			Sets or returns the scaling for any non-bar axis. Not callable from COBOL.

GDEPVAR{SET}
{INQ}
	Sets or returns y axis (or dependent) variables and data qualifications.	GAXISA{SET}
		{INQ}
			Same as above, but callable from COBOL as well.

GLABVAR{SET}
{INQ}
	Sets or returns the label variable for pie or bar charts.	GDEVICE{SET}
		{INQ}
			Sets or returns output device options.

GEXPLODE{SET}
{INQ}
	Sets or returns explosion expressions for pie chart segments.	GTITLE{SET}
		{INQ}
			Sets or returns title, subtitle, footnotes, and axes titles.

GSORT{SET}
{INQ}
	Sets or returns a flag to sort the x axis variable.	GLABEL{SET}
		{INQ}
			Sets or returns labels for major tick marks.

GDATAINFO	Returns information about the data specification of the current chart.	GLEGEND{SET}
		{INQ}
			Sets or returns color, texture, marker, and legend text for line and bar charts.

Table 6-1. Summary of DSG/3000 Procedures (continued)

PROCEDURE	FUNCTION	PROCEDURE	FUNCTION
CHART PROCEDURES (cont.)		**CHART PROCEDURES (cont.)**	

GSYMBOL{SET}
{INQ}
	Sets or returns annotation specifications (lines, boxes, arrows). Not callable from COBOL.	GLIMIT{SET}
		{INQ}
			Sets or returns the position of the corners of the chart. Not callable from COBOL.

GSYMBOLA{SET}
{INQ}
	Same as above, but callable from COBOL as well.	GLIMITA {SET}
		{INQ}
			Same as above, but callable from COBOL as well.

GFONT{SET}	Sets and returns font mapping of available fonts to font numbers.	GCHARTINFO	Returns name and account of chart creator and time of creation.

GTEXT{SET}
{INQ}
	Sets or returns characteristics of text annotations. Not callable from COBOL.		

GTEXTA{SET}
{INQ}
	Same as above, but callable from COBOL as well.		

GTEXTCONTROL {SET} {INQ}	Sets and returns size, color and font for legends, labels and titles.		
GDELETEANOT	Deletes an annotation.		
GREADANOT	Reads graphics cursor to position annotations. Not callable from COBOL.		
GREADANOTA	Same as above, but callable from COBOL as well.		
GANOTINFO	Returns information about all annotations for current chart.		

GSUPPRESS{SET}
{INQ}
	Sets or returns parts of the chart that are not to be drawn.		

Table 6-1. Summary of DSG/3000 Procedures (continued)

PROCEDURE	FUNCTION	PROCEDURE	FUNCTION
GRAPHING PROCEDURES		**CHART FILE MANAGEMENT PROCEDURES**	
GEXECDATA	Executes data transformations and converts data to useable form.	GINITGRAF	Initializes the global data structure.
		GCREATEFILE	Creates a new chart file.
GEXECHART	Draws graph.	GOPENFILE	Opens an old chart file.
		GCLOSEFILE	Closes an open chart file.
		GCREATECHART	Creates a new chart in the chart file.
		GGETCHART	Gets an old chart from the chart file, making it into the current chart.
		GNUMCHARTS	Returns the number of charts in the chart file.
		GLISTCHARTS	Returns the names of the charts in the chart file.

		GCOMMENT {SET}
		{INQ}
			Sets, modifies, or returns comments about a chart.

		GDELCHART	Removes a chart from the chart file.
		GCOPYCHART	Makes a copy of a single chart.
		GEXECFIGURE	Creates the specified figure by drawing the current chart file to a figure file as if it were being drawn to a device. If the figure file does not already exist, it is created.
		GDELETEFIGURE	Deletes the specified figure from the figure file.
		GERRORMSG	Returns error message if an error number has been returned by a procedure.
		GTERMGRAF	Closes any open files.

Current Chart and Chart File

A chart is a collection of design attributes that define a graph. Several charts can reside together in a chart file. An opened chart file is referred to as the current chart file. The chart file must be open (by a previous call to GOPENFILE or GCREATEFILE) before the charts within it can be accessed. A call to GGETCHART or GCREATECHART designates one of the charts from the current chart file as the current chart. All subsequent data procedures, chart procedures, and graphing procedures refer to the current chart.

Using Procedures to Design a Chart

Whenever you access DSG/3000 programmatically, you must call GINITGRAF to initialize the global data area called GRAF. After completing all DSG/3000 procedures, call GTERMGRAF to close the internal DSG/3000 files.

To design a chart, you must call the following procedures in the order given. All of the other chart design and data procedures provide optional features or override the defaults.

GINITGRAF	To initialize global communications area.
GCREATEFILE or GOPENFILE	To create a new chart file or open an existing chart.
GCREATECHART or GGETCHART	To create a new chart or modify and existing chart.
GDATAFILESET	To specify the data file.
GDEFNSET	To define and format all variables.
GINDVARSET	To specify the X-axis (independent) variable.
GCHARTYPESET	To specify the chart type.
GDEPVARSET	For line and bar charts, to specify the Y-axis (dependent) variables.
GCLOSEFILE	To close the chart file.
GTERMGRAF	To close internal files.

May be called in any order

	PIE	LINE	BAR
initialize GRAF area	GINITGRAF	GINITGRAF	GINITGRAF
what chart file? existing new	GOPENFILE GCREATEFILE	GOPENFILE GCREATEFILE	GOPENFILE GCREATEFILE
what chart? existing new	GGETCHART GCREATECHART	GGETCHART GCREATECHART	GGETCHART GCREATECHART
what data file?	GDATAFILE	GDATAFILE	GDATAFILE
variable names?	GDEFN	GDEFN	GDEFN
what chart type?	GCHARTYPE	GCHARTYPE	GCHARTYPE
independent variable? sorted?	GINDVAR GSORT	GINDVAR GSORT	GINDVAR GSORT
dependent variable?		GDEPVAR	GDEPVAR
labels in data file?	GLABVAR		GLABVAR
chart attributes?	GPIE		GBAR
ticks, grids, scaling, etc.?		GAXIS	GAXIS
set fonts, colors, size for labels, titles legends?	GTEXTCONTROL	GTEXTCONTROL	GTEXTCONTROL
label ticks/segments?	GLABEL	GLABEL	GLABEL
color, texture?	[GPIE]	GLEGEND	GLEGEND
specify four active fonts for each chart?	GFONT	GFONT	GFONT
explode segment?	GEXPLODE		
titles, footnotes, axes titles?	GTITLE	GTITLE	GTITLE
annotations? where?	GSYMBOL, GTEXT GREADANOT	GSYMBOL, GTEXT GREADANOT	GSYMBOL, GTEXT GREADANOT
size and placement?	GLIMIT	GLIMIT	GLIMIT
what plotting device?	GDEVICE	GDEVICE	GDEVICE
suppress parts?	GSUPPRESS	GSUPPRESS	GSUPPRESS
create figure file?	GEXECFIGURE	GEXECFIGURE	GEXECFIGURE
delete figure file?	DELETEFIGURE	DELETEFIGURE	DELETEFIGURE
close chart file	GCLOSEFILE	GCLOSEFILE	GCLOSEFILE
close internal tables	GTERMGRAF	GTERMGRAF	GTERMGRAF

Figure 6-1. Logical Order of Procedures to Design Chart

Figure 6-1 lists the procedures that may be used to design a chart in logical order. As illustrated, you can call the procedures from GSORT through GSUPPRESS in any order. However, it is recommended that the order be followed so that you can be assured of expected results.

Using Procedures to Produce A Chart

You may decide to design a chart interactively as described in Section IV, and then draw the chart from within a report-generating program. To programmatically produce a graph that has already been designed either programmatically or interactively, you must call the following procedures in the order given.

GINITGRAF	To initialize the communications area.
GOPENFILE	To open the chart file.
GGETCHART	To access the chart.
GEXECDATA	To prepare the data.
GEXECHART	To draw the graph.
GEXECFIGURE	To draw to a figure file.
GCLOSEFILE	To close the chart file.
GTERMGRAF	To close internal tables.

Using Procedures to Control Output

If you draw a chart without specifying a particular plotting device, it will be produced on your graphics terminal by default. (If you are not using a graphics terminal, an error is returned.) To route the chart to a different output device such as a plotter, call GDEVICESET. This procedure also allows you to specify the number of copies you want produced.

If you do not specify the size or position of the graph on the output device, DSG/3000 will scale the output page appropriately. If the output device is a plotter, the output is scaled so that it approximates an 8-1/2 inch x 11 inch format (English standard).

You can override this format by calling GLIMIT[A]SET to set either the outer limits (frame boundary) or inner limits (data boundary) of the graph (see Figure 6-2), or to use Metric standard format (210mm x 298.5mm). You cannot specify BOTH the outer and inner limits because DSG/3000 determines the appropriate space for headings and footnotes.

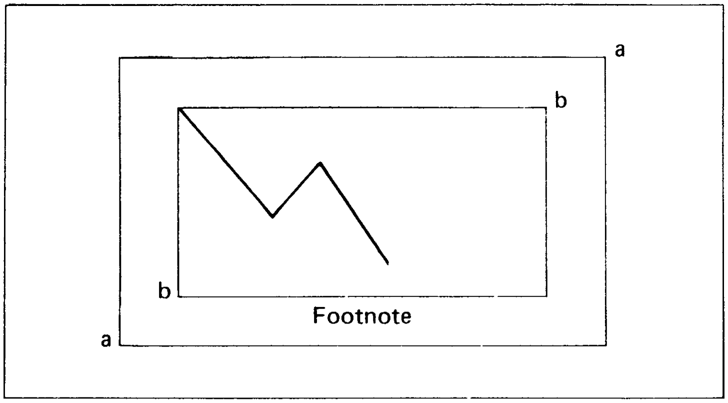


Figure 6-2. Plotting Limits: a) outer; b) inner

The coordinates of the plotting limits are given in millimeters from the bottom left corner.

Any limits set by GLIMIT[A]SET override the plotter limit points. When the plotting device is opened, the plotter limits are reset.

To summarize, you can accept the position defaults or set one of the following:

outer limits (frame boundary)
inner limits (data boundary)
plotter limits (page boundary)
Metric standard page (210 mm x 298.5 mm)
English standard page (8 1/2 by 11 inches)

Note that the default limits are English for most devices, D-size paper for the 7580.

Using Procedures to Create a Figure File

To draw the current chart from the current chart file to a figure file you call GEXECDATA to prepare the data for drawing. You then call GEXECFIGURE to create the file. In effect, it is drawn to a figure file as if it was being drawn to a device. Figures stored in figure files may not be scaled, modified, or rotated except by HPDRAW and TDP/3000. They reside in the files so that they may be accessed by TDP/3000, HPDRAW and other office systems applications.

To programmatically draw a previously prepared chart to a figure file, you must call the following procedures in the order given:

GINITGRAF	To initialize the communications area.
GOPENFILE	To open the chart file.
GGETCHART	To access the chart.
GEXECDATA	To prepare the data.
GEXECFIGURE	To draw the current chart to a figure file.
GCLOSEFILE	To close the chart file.
GTERMGRAF	To close internal tables.

To delete a figure that has been stored in a figure file, call the following procedures in the order given:

GINITGRAF	To initialize communications area.
GDELETEFIGURE	To delete the figure.
GTERMGRAF	To close internal tables.

Using Procedures to Annotate a Graph

Four types of annotations are provided by DSG/3000:

* box
* arrow
* line
* text

You assign a unique identifying number to each annotation when you create it. To specify a box, arrow, or line, call GSYMBOL[A]SET, and to specify a textual annotation, call GTEXT[A]SET.

It is best to position each annotation interactively. The program should call GREADANOT[A] and prompt the user to position the graphics cursor at the appropriate point on the graph. GREADANOT[A] reads the cursor and returns the relative location to the programs. If you are positioning an annotation requiring two points (box, line, or arrow) you must call GREADANOT[A] a second time and repeat the procedure. You can then pass coordinates of the points to GSYMBOL[A]SET or GTEXT[A]SET which draws the annotation on the screen.

If you are setting a text annotation, you must also specify to DSG/3000 the label origin, which is the placement of the text relative to the digitized point. For example, in Figure 6-3, to position the annotation "High", you could digitize the position of "H", specifying the label origin as the left part of the string. To position the annotation "Low", you could digitize the position of the "w" and specify the label origin as the right part of the string.

Annotation positions are passed in units that are device independent so that the relative position of the annotation remains the same relative to the chart regardless of size and scaling.

GANOTINFO returns a list of annotation identifiers and their types. If you call GSYMBOL[A]INQ or GTEXT[A]INQ for a particular annotation (specified by identifier), that annotation will be drawn on your graphics terminal if the correct parameter is set.

An annotation can be deleted by calling GDELETEANOT.

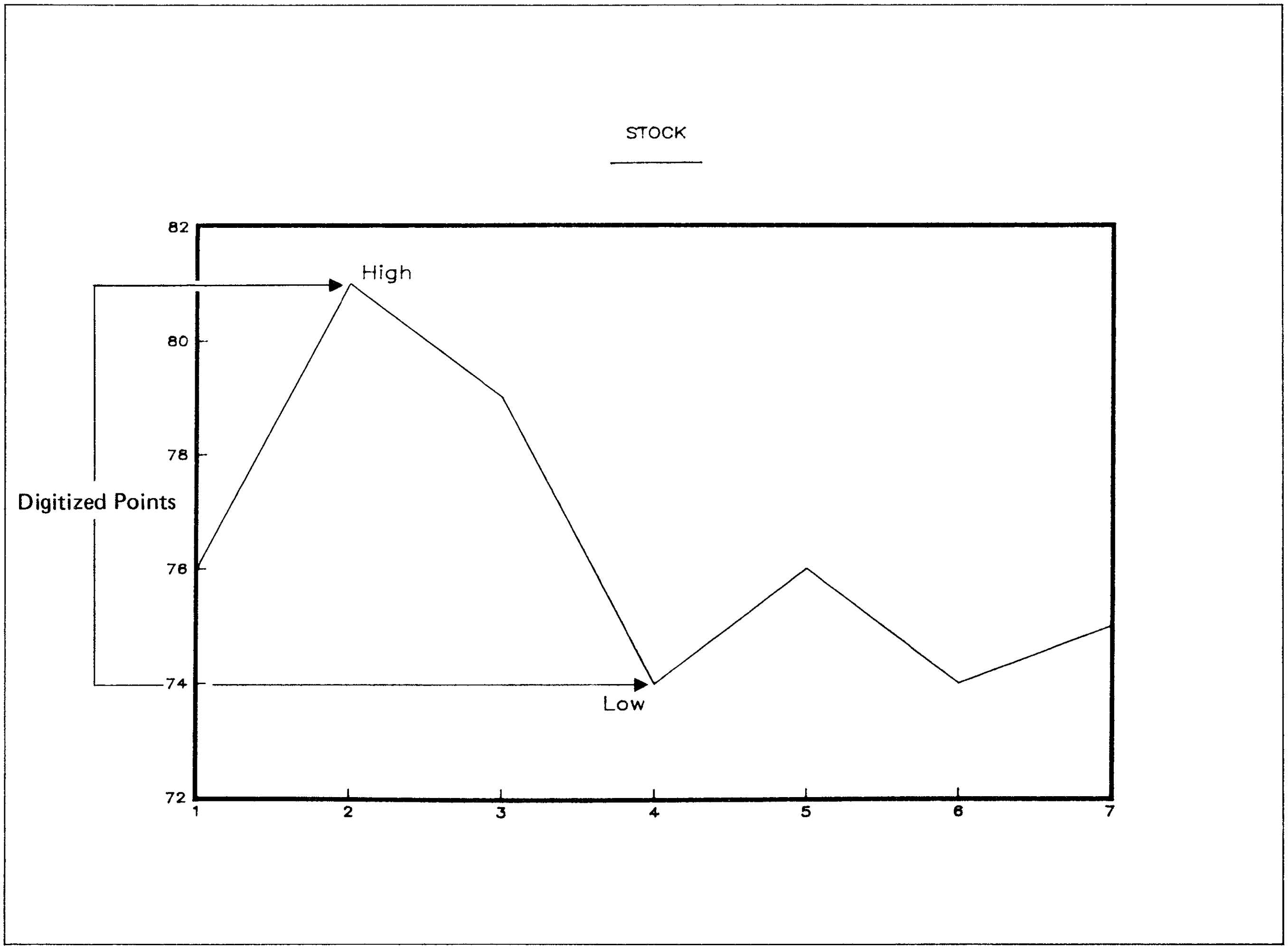


Figure 6-3. Digitized Points and Text Annotation

Using Procedures to Design a Bar Chart

If you specify a bar graph as the chart type in the GCHARTTYPESET procedure, you must specify at least one dependent variable by calling GDEPVARSET. Then, you can either draw the chart using all of the DSG/3000 design defaults, or you can call GBAR[A]SET and other design procedures to set the attributes yourself. If you allow the design to default, you will get:

- vertical, stacked, labeled bars
- axes internally scaled according to the values in the data file
- a bar drawn for each data point
- Y-axis having labeled ticks
- Y variables represented by different textures, but no legends
- no titles, no grids

For definitions and examples of usage of bar chart characteristics, See Section II, Graphing Terms and Concepts.

Clustered or Horizontal Bars

If you want the bars to be horizontal, or if you want them to be clustered rather than stacked, you must call GBAR[A]SET. For examples of the usage of stacked vs. clustered bars, see Section II.

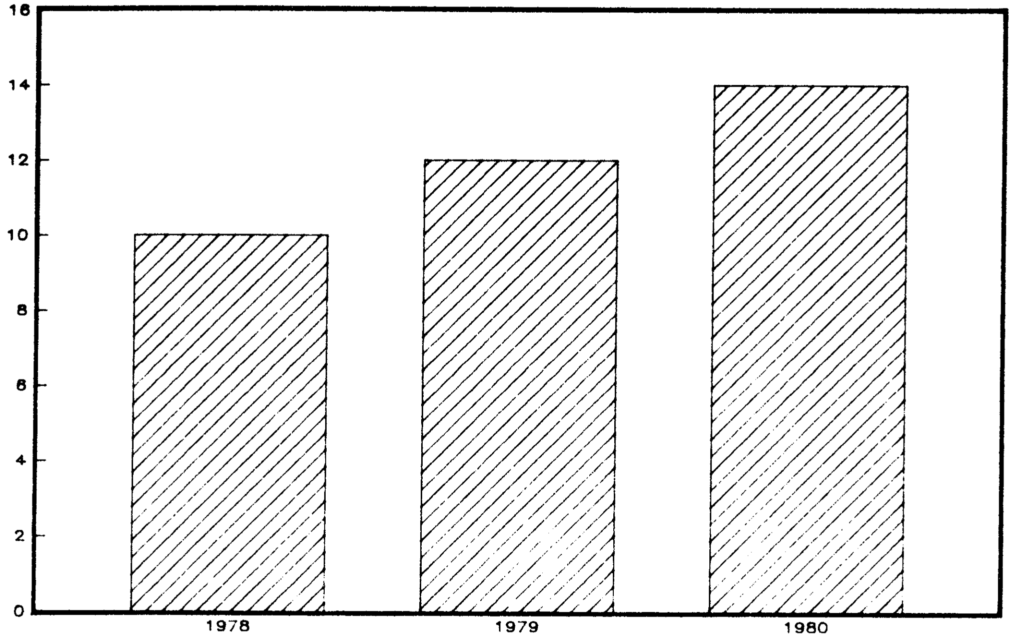


Figure 6-4. Bar Graph with Default Characteristics

Bar Axis Scaling

If you want intervals to be added at the beginning of the bar axis before the bars begin, as in Figure 6-5a, call GBAR[A]SET to give a value less than the minimum data value to the *start* parameter. To add intervals at the end of the bar axis, call GBAR[A]SET to set the *numintervals* parameter to a value greater than the number of data items (number of bars). For example, in Figure 6-5b, *numintervals* is set equal to 12, although there are only three values in the data file.

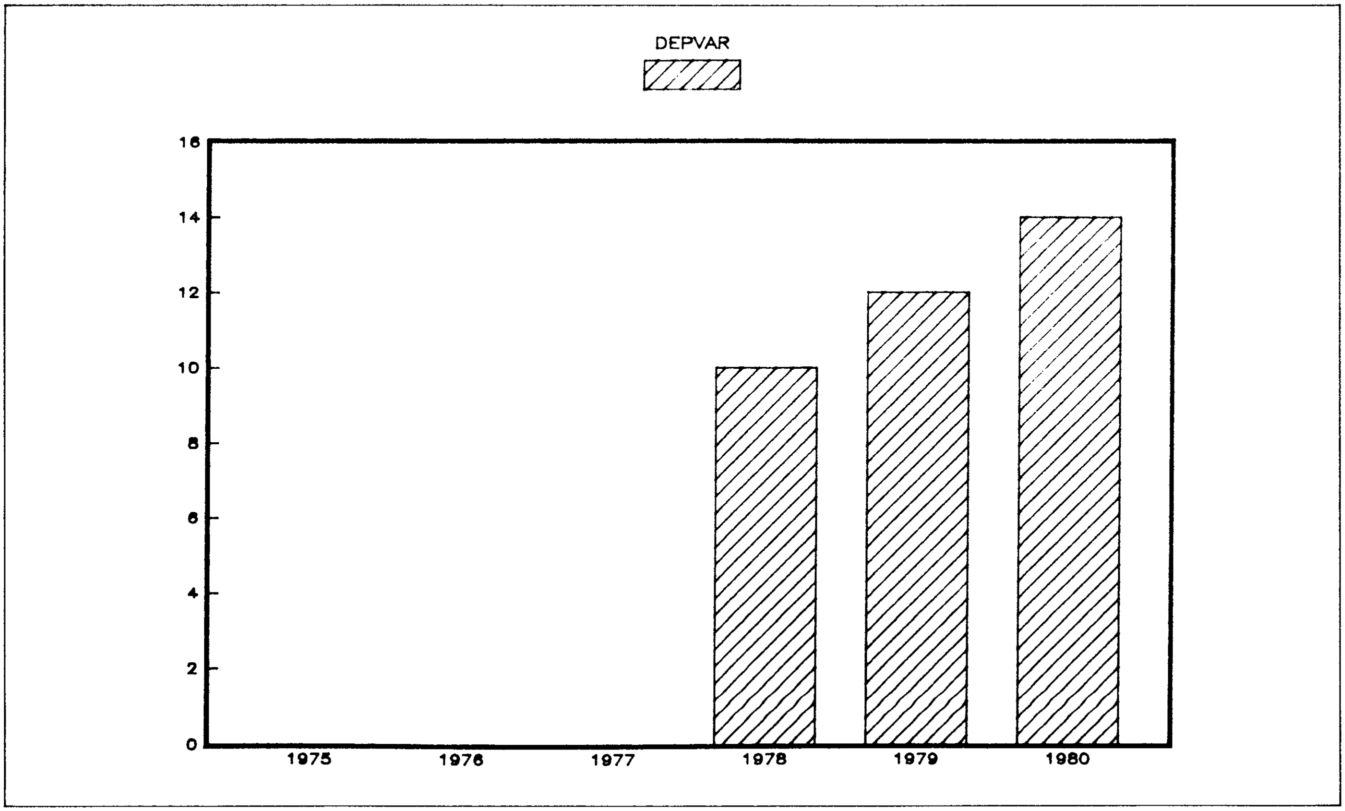


Figure 6-5. Scaling the Bar Axis a) using the *start* parameter

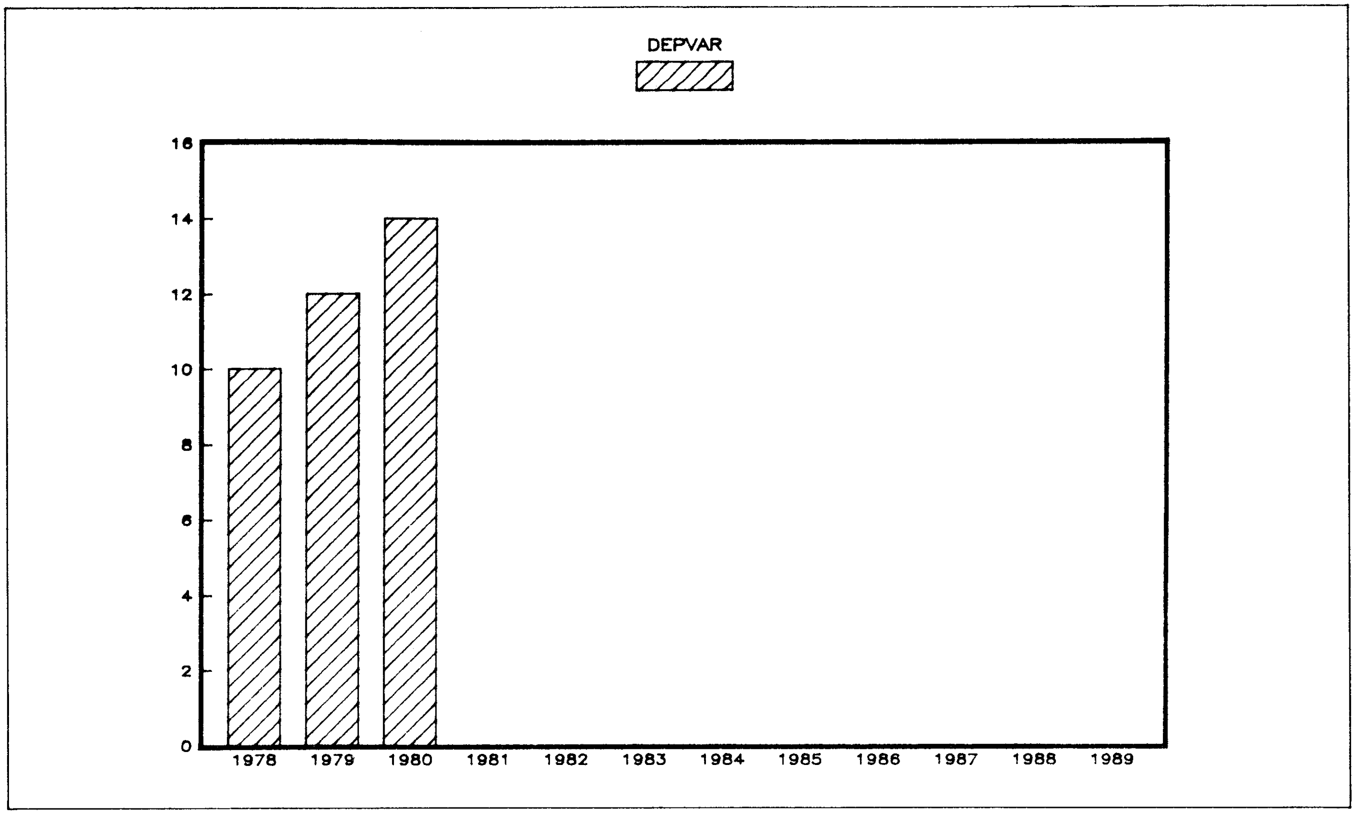


Figure 6-5. Scaling the Bar Axis b)
using *numintervals* parameter

Y Axis Scaling

The default axes scaling, which takes the maximum and minimum values from the data file as the maximum and minimum values of the axis range (as in Figure 6-6a), may be overridden. To do so you call GAXIS[A]SET, set the *min* and *max* parameters to the desired values, and set the *defaults* parameter. You can also set the size of the interval between major ticks by setting the *intervalsize* parameter. For example, in Figure 6-6b, *min* is set equal to -5, *max* is set equal to 20, *defaults* is set equal to 4, and *intervalsize* is set equal to 5.

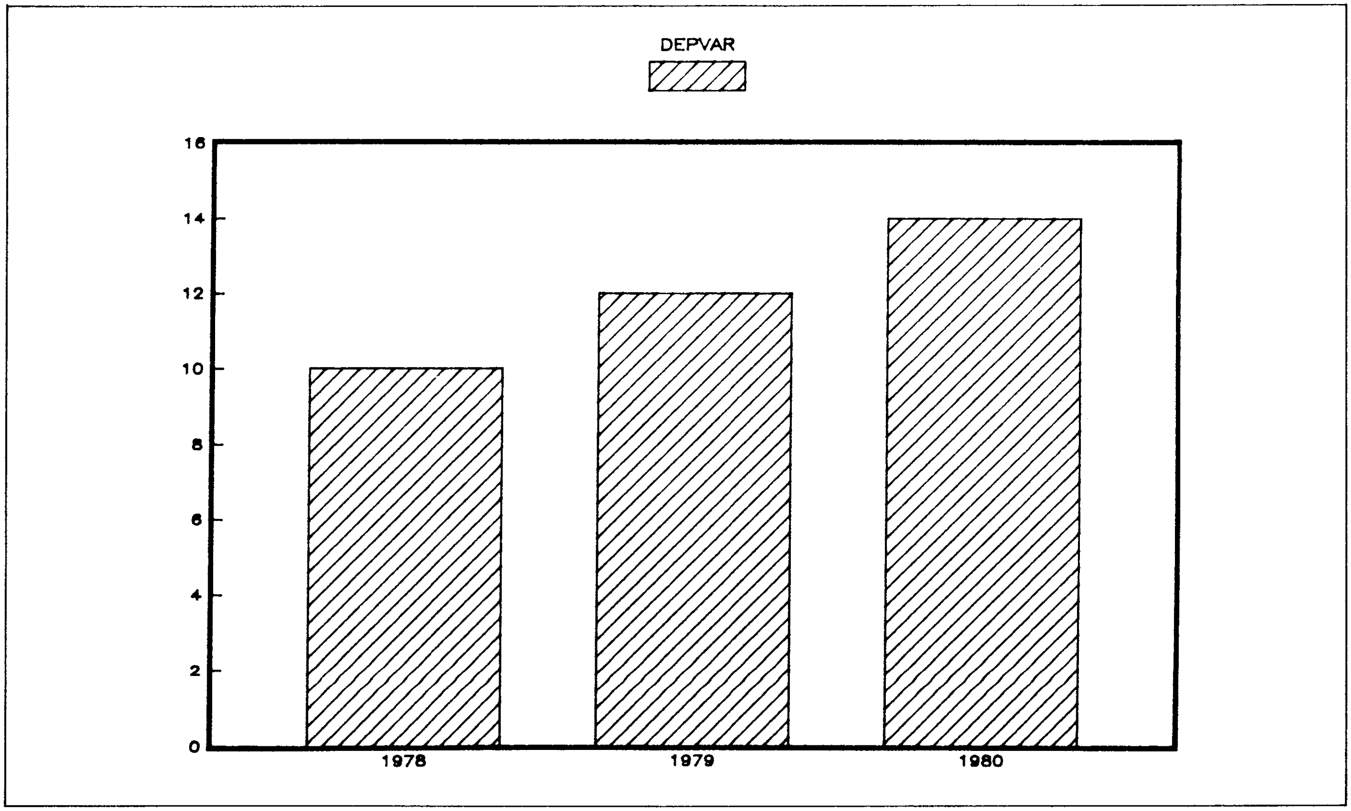


Figure 6-6. Scaling the Y Axis a) defaults

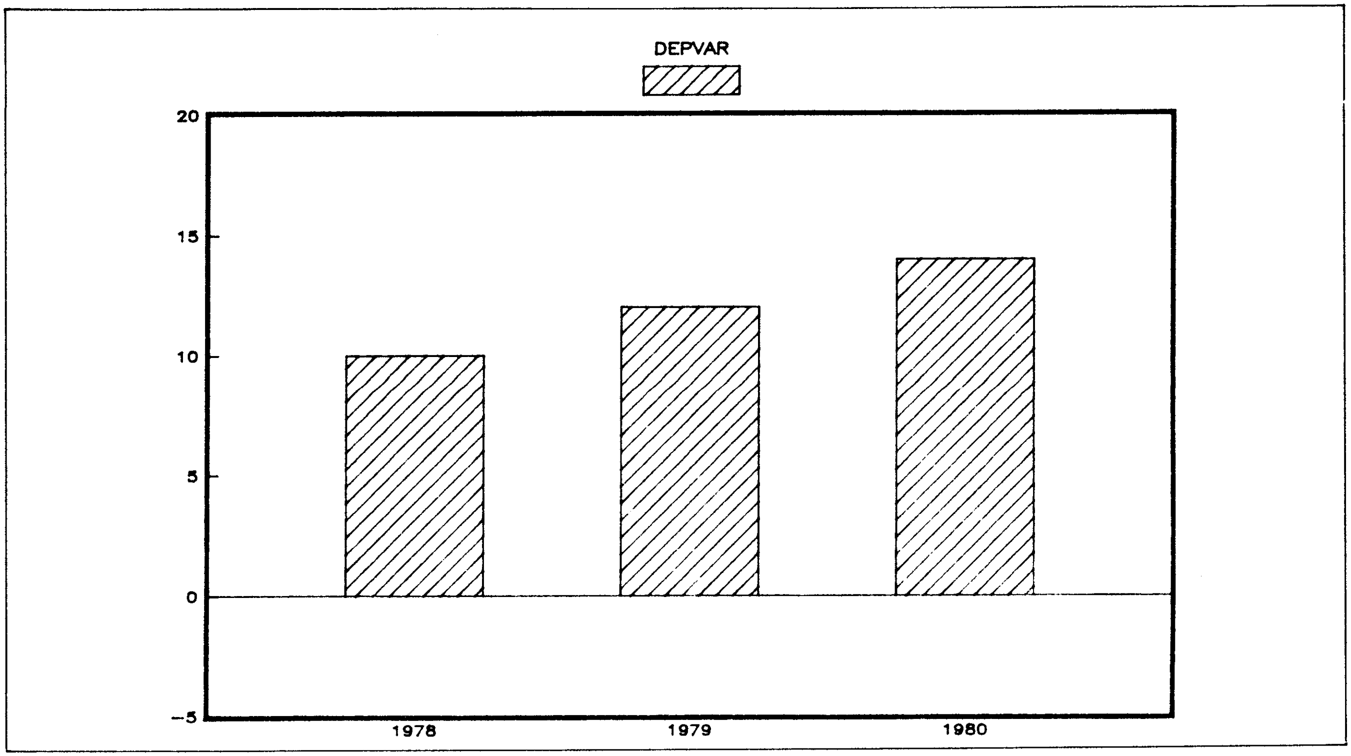


Figure 6-6. Scaling the Y Axis b) using the GAXIS[A]SET Procedure

Ticks

You can delete the ticks that appear along the Y-axis by setting the *noticks* parameter in the GAXIS[A]SET procedure. If you want both major ticks and smaller minor tick marks, specify the number of minor ticks per interval in the *minorticks* parameter of GAXIS[A]SET.

If you want to label the major ticks, call GLABELSET. You can label the ticks according to the way the axis is scaled by setting the *type* parameter equal to 2. GLABELSET also allows you the option of labeling every other tick or every *n*th tick.

Labeling the Bar Axis

There are three ways of labeling the bar axis in a bar graph. In any of these cases you must call GLABELSET for the X-axis.

Method 1 -- The data file contains a label variable. For example, in a bar graph of EXPENSES by DEPARTMENT, the DEPARTMENT variable might contain department names. To implement the labels, set the *type* parameter equal to 1 in GLABELSET, and call GLABVARSET to name the label variable, in this example, DEPARTMENT.

Method 2 -- The bars are labeled numerically. For example, if the minimum is 1965, maximum is 1974, and interval width is one, the bars will be labeled 1965, 1966, 1967,...1974. Set *type* parameter equal to 2 in GLABELSET.

Method 3 -- You supply labels dynamically in a circular buffer when you call GLABELSET. For example, suppose the data variable MONTH contains numeric values 1 through 15, but you want the bars to be labeled JAN through DEC and back to JAN again. Use the parameters *labelbuf*, *labelen*, and *numlabels* in GLABELSET to specify the labels, and set the *type* parameter equal to 3.

Grids

Either or both axes can have grid lines drawn through the plotting area. Specify grids for the bar axis in a call to GBAR[A]SET, and for the Y-axis in a call to GAXIS[A]SET.

Legends, Colors and Textures

The bars for each y variable are shaded differently in a stacked or clustered bar chart. If you want to show a legend indicating which variable is represented by each texture, call GLEGENDSET. This procedure also allows you to change the textures or add colors to differentiate the variables.

Using Procedures to Control Text

You can specify size, font, and color for legends, labels and titles by calling GTEXTCONTROLSET. This procedure allows you to design your chart so that important information receives suitable emphasis. The color, size, and font values will be the same as those available through annotations.

The range of text sizes is 1 to 72, where 1 is the smallest size. The defaults (blanks) use the following sizes:

Bar and Line Charts

Main Title	8
Subtitles	6
Left Axis Title	5
Bottom Axis Title	5
Y-axis Labels	4
X-axis Labels	4
Legend Text	5

Note that the DSG/3000 text sizes correspond to standard point measurements only when you plot to 8 1/2 by 11 inch paper. If you use a different size, the text size will not conform to the standard.

Four software fonts may be specified for each chart. The mapping of the four fonts selected is accomplished by calling GFONTSET. This same procedure allows you to select native language characteristics for the fonts. Only one hardware font is available.

Using Procedures to Design a Line Graph

If you specify a line graph as the chart type in the GCHARTTYPE procedure, you must specify at least one dependent variable by calling GDEPVARSET. Then you can either draw the chart using all of the DSG/3000 design defaults, or you can call other procedures to set the attributes yourself. If you allow the design to default, you will get:

* each dependent variable represented by a line with a different texture (see appendix A) but no legends

* all lines are black

* no data markers

* linear axes scaled according to the data and marked with labeled ticks

* no titles, no grids

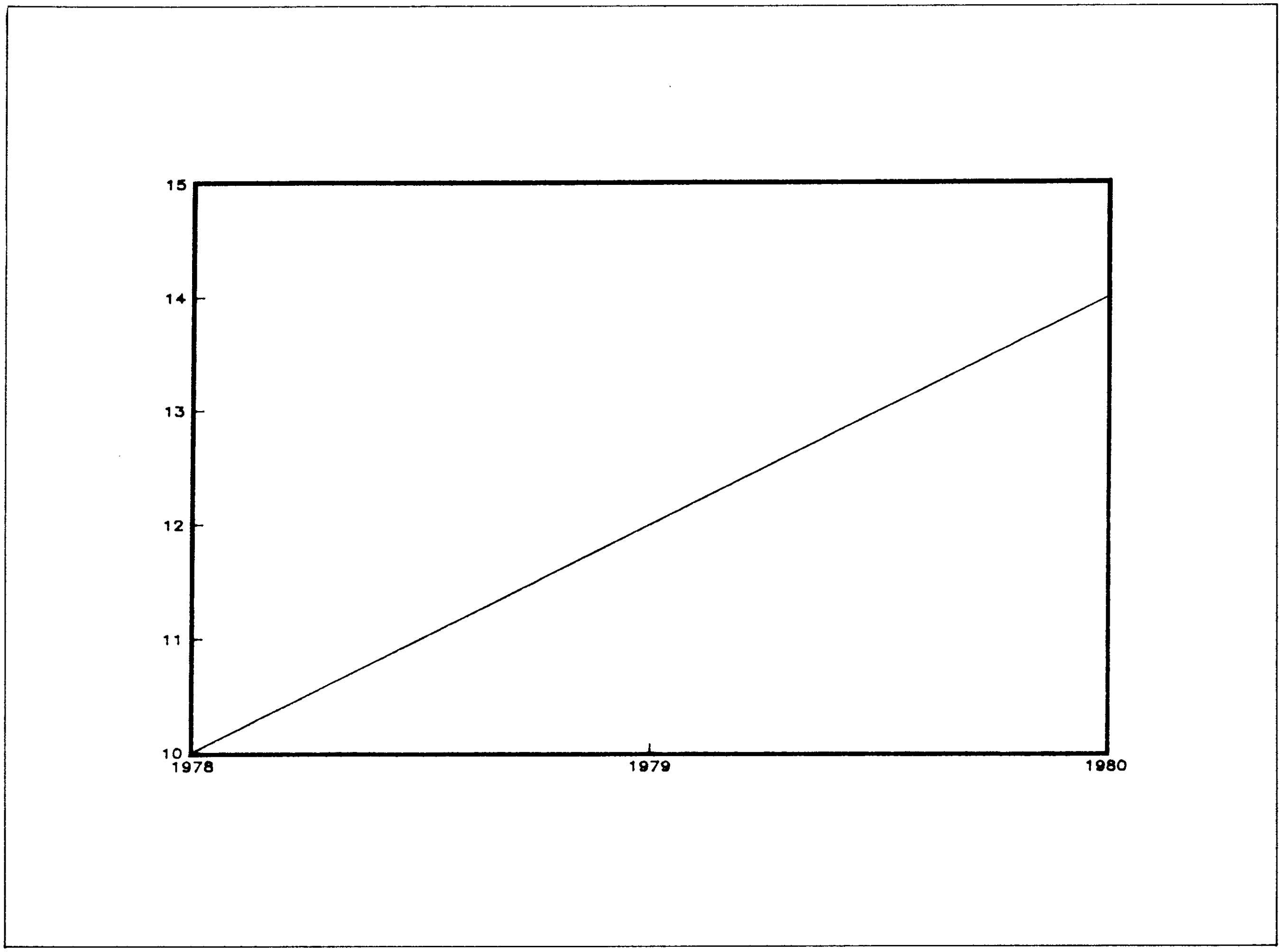


Figure 6-7. Line Graph with Default Design

Legends, Textures, Colors, Markers

If you want the graph to show a legend indicating which line texture represents each variable, call GLEGENDSET. This procedure also allows you to change the assignment of textures or to use colors to differentiate the variables. You can also assign a marker from Appendix A to mark each data point on the line. Note that you must call GLEGENDSET once for each variable.

Axis Scaling

To override the default scaling, which takes the maximum and minimum values from the data file as the maximum and minimum values of the axis range, call GAXIS[A]SET. Set the *min* and *max* parameters to the desired values, and set the *defaults* parameter. You can also set the size of the interval between major ticks by setting the *intervalsize* parameter. You can call GAXIS[A]SET once for each axis.

Labeling the Axis

There are two ways of labeling the X-axis in a line graph. In either case, you must call GLABELSET.

Method 1 -- The lines are labeled numerically. Set *type* parameter equal to 2 in GLABELSET.

Method 2 -- You supply labels dynamically in a circular buffer when you call GLABELSET. Use the parameters *labelbuf*, *labelen*, and *numlabels* in GLABELSET to specify the labels, and set the *type* parameter equal to 3. Make sure that labels match up with data points.

Ticks

You can delete the ticks that appear along either axis by setting the *noticks* parameter in the GAXIS[A]SET procedure. If you want the smaller minor ticks as well major ticks, specify the number of minor ticks per interval in the *minorticks* parameter of GAXIS[A]SET.

If you want to label the major ticks, call GLABELSET. You can label the ticks numerically by setting the *type* parameter equal to 2. Alternatively, you can supply the labels dynamically using the *labelbuf*, *labelen*, and *numlabels* parameters in GLABELSET, and setting the *type* parameter equal to 3. GLABELSET also allows you the option of labeling every other tick, or every *n*th tick.

Logarithmic Scaling

If you want either or both axes to be logarithmic rather than linear, call GAXIS[A]SET to set the *scaletype* parameter.

Grids

Either or both axes can have grid lines drawn through the plotting areas. Call GAXIS[A]SET to set the number of major ticks per grid line.

Controlling Text Size, Color, and Font

If you wish to design your graph so that all standard text receives the desired emphasis, call GTEXTCONTROLSET. Using this procedure you can specify pen number for color, select a text size that will be appropriate for the title, subtitle, axis labels, legend labels, and footnotes, and use up to four fonts per chart.

The range of text sizes is 1 to 72, where 1 is the smallest size. The defaults (blanks) use the following sizes:

Bar and Line Charts

Main Title	8
Subtitles	6
Left Axis Title	5
Bottom Axis Title	5
Y-axis Labels	4
X-axis Labels	4
Legend Text	5

Note that the DSG/3000 text sizes correspond to standard point measurements only when you plot to 8 1/2 by 11 inch paper. If you use a different size, the text size will not conform to the standard.

Use GFONTSET to map fonts to the four font numbers which may be used with a single chart. The same procedure allows you to specify native language characteristics for the four software fonts. Only one hardware font is available.

Using Procedures to Design a Pie Chart

If you specify a pie chart in the GCHARTTYPESET procedure, you can either draw the chart using all of the DSG/3000 design defaults, or you can call GPIESET and other procedures to set the chart attributes yourself. If you allow the design to default, you will get:

* a pie chart with segments drawn in black, no textures

* segments drawn in the order in which the data values appear in the data file, beginning at the two o'clock position

* no labels

* no percentages or actual values printed

* no exploded segments

* no titles or footnotes

Effective pie charts have a limited number of segments, often no more than 16. Maximum number of segments permitted is 30. If the data results in a segment of less than 2 degrees, a warning message is returned; you should subset the data file so that the segments and labels will be discernible.

Colors and Textures

You can assign colors and textures to each segment of the pie chart by calling the GPIESET procedure. See Appendix A for a table of colors and textures.

Text Control

You may specify color, size, and font for all standard text -- title, subtitle, labels, and footnotes. To do so, call GTEXTCONTROLSET. The values are the same as those used for annotations.

The range of text sizes is 1 to 72, where 1 is the smallest size. The defaults (blanks) use the following sizes:

Pie Charts

Main Title	8
Subtitles	6
Segment Labels	4

Note that the DSG/3000 text sizes correspond to standard point measurements only when you plot to 8 1/2 by 11 inch paper. If you use a different size, the text size will not conform to the standard.

Four fonts may be used in a single chart. You specify those fonts by mapping them to font numbers using the GFONTSET procedure. The same procedure allows you to select native language characteristics for the fonts. Only one hardware font is available for a chart.

Sorting the Segments

You may want the pie segments to be sorted and drawn from smallest to biggest for ease of reading. Call GSORTSET to exercise this option. The first segment is drawn starting at the two o'clock position regardless of whether the data is sorted. See Figure 6-8.

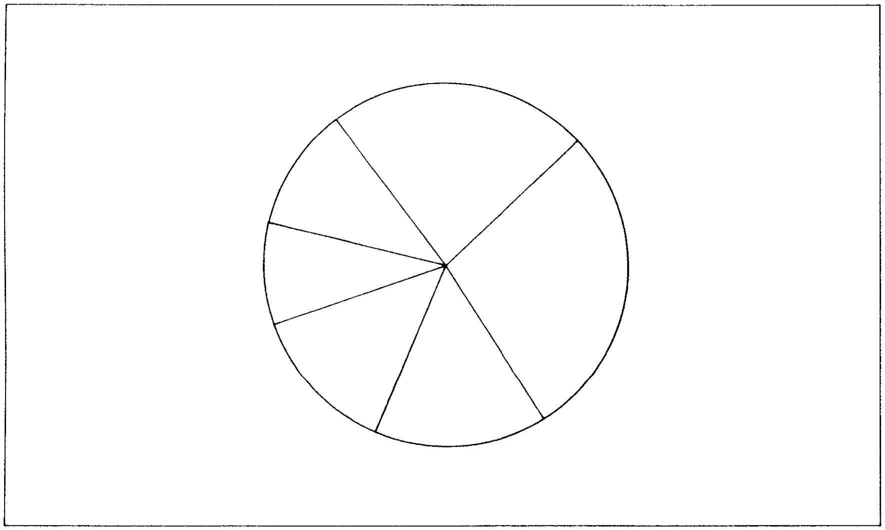


Figure 6-8. Pie Chart with Default Characteristics a)unsorted

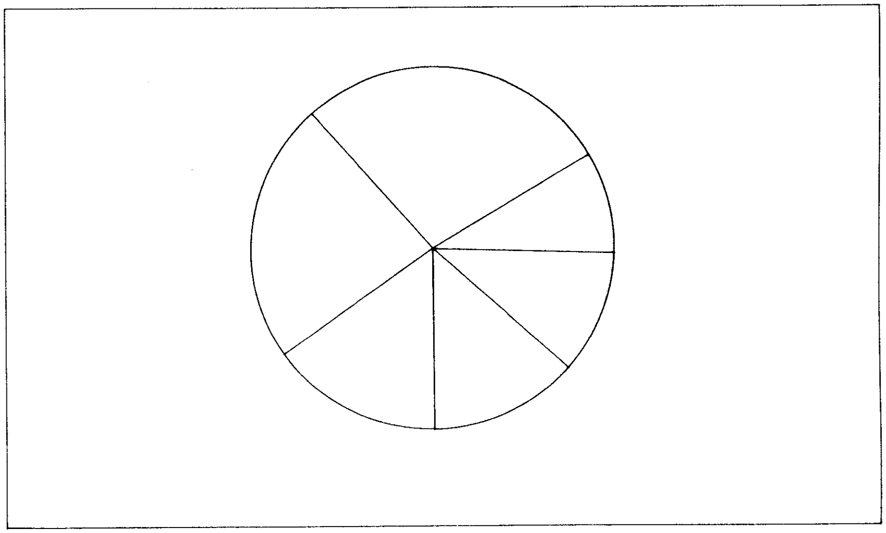


Figure 6-8. Pie Chart with Default Characteristics b)sorted

Labeling the Segments

There are 4 ways of labeling the pie segments:

- actual values taken from the data file
- percentages calculated from the data by DSG/3000
- text taken from a label variable in the data file
- text supplied directly by you

You can use these methods alone or in combination. For example, you might want to have percentages listed together with the text from the label variables.

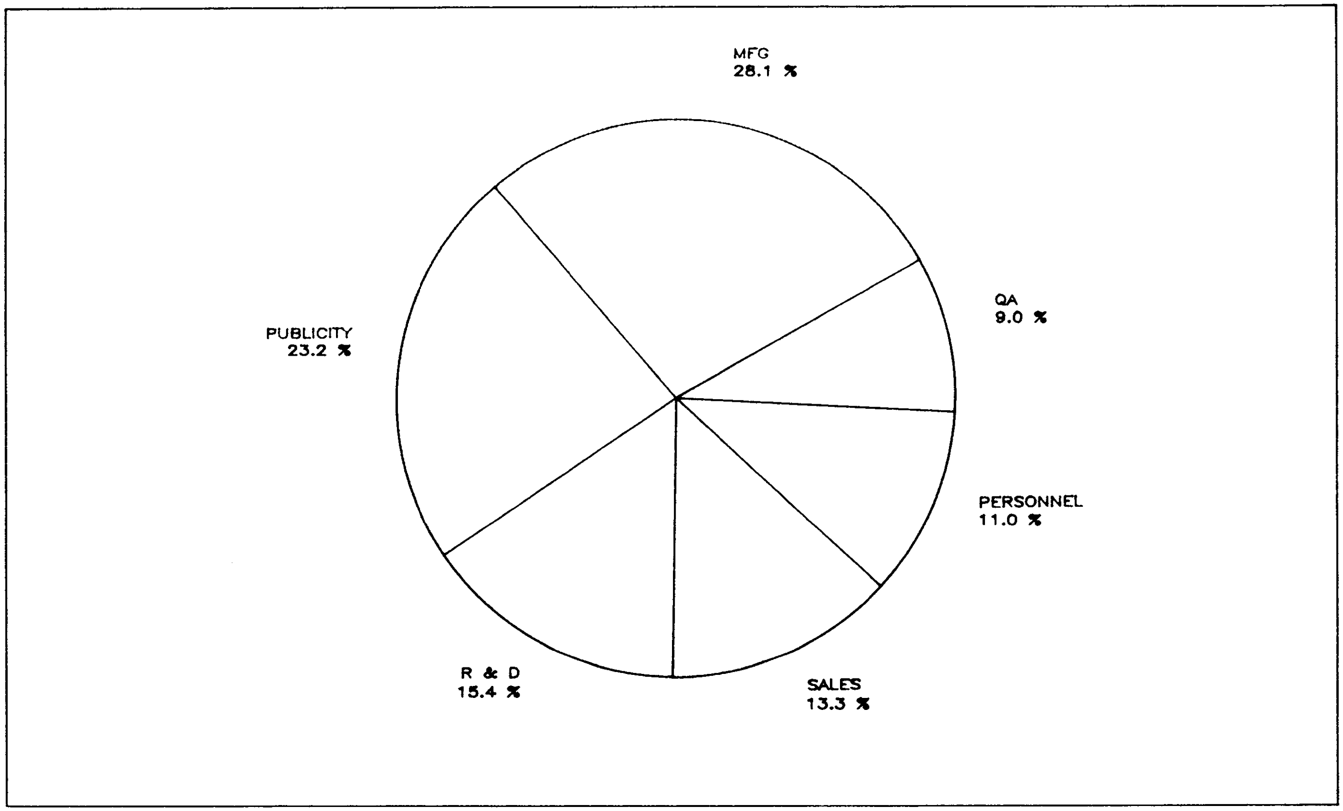


Figure 6-9. Labeled Pie Charts: a) percents

Actual values and percentages are specified by calling GPIESET.

To specify that a variable from the data file is to be used for labels, call GLABVARSET, and also set the *type* parameter equal to 1 in the GLABELSET procedure. For example, if you are making a pie chart of EXPENSES by DEPARTMENT where DEPARTMENT is the variable that contains department names, specify DEPARTMENT as the label variable when calling GLABVARSET.

You can supply your own labels by calling GLABELSET. Set the *type* parameter equal to 3, and set the *labelbuf*, *labelen*, and *numlabels* parameters to describe the labels buffer. The first label name you put in the buffer corresponds to the first segment drawn in the pie (two o'clock position).

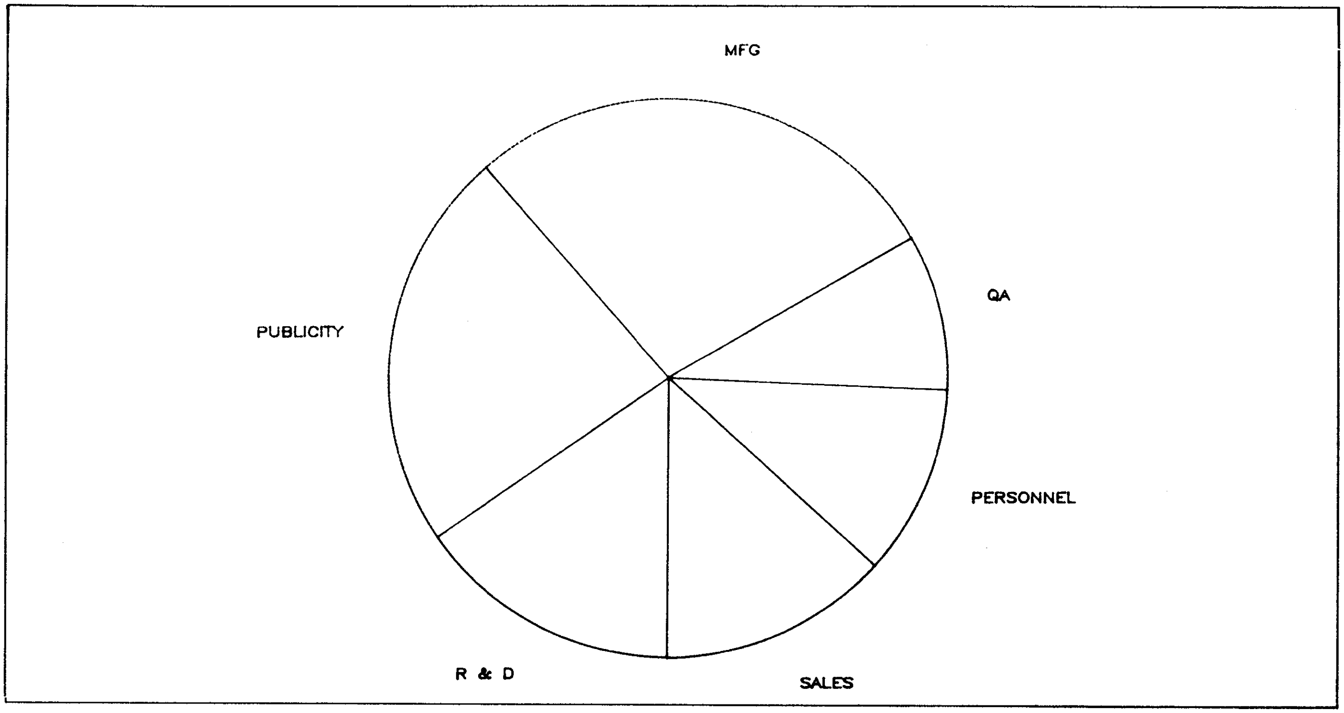


Figure 6-9. Labeled Pie Charts: b) label variable

Exploding Segments

If you want to call attention to a particular segment of a pie chart, you can explode it by calling the GEXPLODESET procedure.

For simplicity, you would usually want to explode no more than one segment of a pie chart.

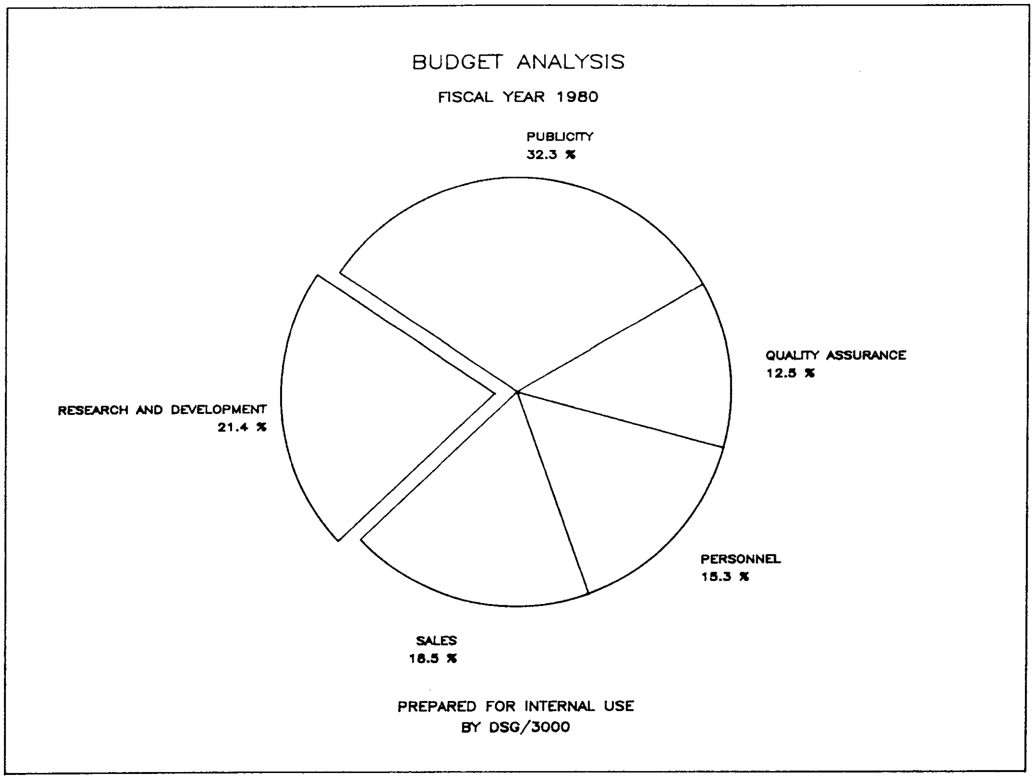


Figure 6-10. Pie Chart with One Exploded Segment

Using Procedures to Design a Scattergram

Design a scattergram the same way you would design a line chart with one variation: you must call GLEGENDSET, setting the *texture* parameter equal to 0. This suppresses connecting lines between data points. Be sure to specify a nonzero value for the *marker* parameter. All of the other chart attributes of line graphs apply to scattergrams as well.

SECTION 7

Procedures -- Reference Specifications

Calling DSG/3000 Procedures

For all programming languages, the following rules apply to all parameters:

- * Parameters are passed by reference; a literal value cannot be used as a parameter.
- * No condition codes are returned; the status of the call is returned in the first four words of the communication area, included in every procedure call as the *graf* parameter.
- * All parameters are required. For those parameters that have defaults, you can usually pass a 0 to allow the default to be taken.

Table 7-1 shows the format of calls to DSG/3000 procedures from each language.

Table 7-1. Format for Calling Procedures

LANGUAGE	PROCEDURE CALL FORMAT
COBOL	CALL "procedurename" USING parameter1, parameter2 . . .
FORTRAN	CALL procedurename (parameter1, parameter2 . . .)
BASIC	linenumber CALL procedurename (parameter1,parameter2 . . .)
SPL	procedurename(parameter1,parameter2 . . .)
PASCAL	procedurename(parameter1,parameter2 . . .)

where:

procedurename identifies the procedure being called.

parameter at least one parameter is required for each procedure; the particular parameters are listed in the formats for the individual procedure descriptions. Note that when more than one parameter is specified, each is separated by a comma; and for COBOL calls, by a comma (optional) and a space.

Parameter Types

The data types that are allowed in DSG/3000 procedures are shown are shown in Table 7-2.

Table 7-2. Data Types Allowed for Various

DATA TYPE	LANGUAGE				
	COBOL	FORTRAN	BASIC	SPL	PASCAL
Byte Array	DISPLAY PIC X(n)	CHARACTER	STRING	BYTE ARRAY	ARRAY OF CHARACTER
Integer	COMP PIC S9(4)	INTEGER	INTEGER	INTEGER	INTEGER SUB-RANGE (in range of -32768..32767)
Integer Array	GROUP ITEM	INTEGER ARRAY	NUMERIC ARRAY	INTEGER ARRAY	INTEGER ARRAY OF INTEGER SUB-RANGE (in rnage of -32768..32767)
Real	*	REAL	REAL	REAL	REAL

* Since COBOL does not handle real data values, DSG/3000 provides alternative procedures which pass real values as ASCII strings.

Each parameter is described according to its SPL generic type (integer, integer array, byte array, or real). This table is provided for those languages that do not call their data types by these particular names.

Note in particular the special handling of real data values when programming in COBOL. DSG/3000 provides an ASCII version for procedures that must pass real data values. These ASCII procedures are denoted with an "A" in the procedure name; for example, GBARSET becomes GBARASET when used to pass real values from COBOL. The string must contain a real number that is 16 characters long and padded with blanks if necessary. (Padding with trailing blanks is the recommended method.)

The real number itself consists of an integer part, a decimal point, and a decimal fraction (mantissa) part. A leading sign can be used; if it is not used, the default is a positive value. You can also use a scale factor which represents a power of ten by which the constant part is multiplied.

The following are allowable forms of real numbers:

FORM	EXAMPLE	EQUIVALENT VALUE
n	20	
.n	.2	
n.n	20.5	
n.E+e	2.E+2	2. * 100
.nE+e	.2E+3	.2 * 1000
n.nE+-e	2.5E+2 or 2.5E-2	2.5 * 100 or 2.5 * 1/100
nE+-e	2E+2 or 2E-2	2 * 100 or 2 * 1/100
nEe	2E2	2 * 100
.nEe	.2E2	.2 * 100
n.Ee	2.E2	2. * 100
n.nEe	2.2E2	2.2 * 100

The range of e is from -77 to +78. The letter n represents a decimal integer with an optional sign.

Communications Area

GRAF is the global communications area. Because procedures have no global storage of their own, you must provide the space necessary to store information that is to be shared between procedures. GRAF holds information about your particular session, the current chart file, and the current chart.

You should allocate a minimum of 1600 words for GRAF. This is done in the GINITGRAF procedure. If you are programming in BASIC, you must allocate at least an additional 2048 words for chart records, so that GRAF will total about 3700 words. You will need even more space if your charts contain very many annotations and labels.

The first four words of GRAF return status information to the program. The rest of the communications area is for internal use only.

Note that on the HP 3000, one word is two bytes or 16 bits.

Table 7-3. GRAF Contents

WORD	REPRESENTING
1	status returned by procedure
2	parameter position
3	element position
4	File System Error
5-1600	reserved for internal use

The GRAF words listed in Table 7-3 are defined as follows:

status — Integer in which the procedure status is returned. Set to zero if the call is successful; to a non-zero value if an error occurs. If the error is an MPE file error, a file error number is also returned in the fourth word. It is up to you to provide error-handling routines and to reset the status. (See Appendix C for error codes that may be returned here with their meaning.The errors in Appendix C are listed in ascending order, loosely grouped by procedures.)

parameter position — Integer representing which parameter in the calling sequence is in error. The first parameter is numbered one.

element — Integer representing which element in the parameter was in error. For example, if the second color in a color parameter was given in error, a two is returned in this word.

File System Error — MPE file error number (FCHECK number) returned by DSG/3000 procedures when an MPE file error occurs.

For example, if an error is encountered while setting the color of the third segment of a pie chart, GRAF will hold the following codes after the call to GPIESET:

WORD 1	132	No such color
WORD 2	4	Fourth parameter in error
WORD 3	3	Third segment in error

You must initialize the GRAF area before calling other DSG/3000 procedures by calling GINITGRAF.

Format

For every procedure, the format is given as:

```
               type         type
procedurename (param,...,param)
               in/out       in/out
```

The data type is given above each parameter in SPL notation. Programmers writing in other languages should refer back to Table 7-2 for data types.

Below each parameter is a code specifying whether it is for input or output:

in — DSG/3000 uses the value passed by this parameter

out — DSG/3000 sets or modifies the value of this parameter

i/o — DSG/3000 uses some values and returns other values, such as status information.

For those procedures that have a special ASCII version, the ASCII format is given in COBOL format.

For the most commonly used procedures, examples of Data Declarations in each language are included at the end of the procedure discussion.

Preparing your Program For the New Version

If you have been using an earlier version of DSG/3000, a larger communications area is required by the version described in this manual in order to accomodate the enhancements.

To run the new version of DSG/3000, you you should prepare the program to include the following:

```
CAP=PH,MR,DS;MAXDATA=32000
```

If you have chart files prepared using the previous versions of DSG/3000, they will need to be converted. You can update these files programmatically or interactively. See Appendix F for methods of conversion.

GANOTINFO

Returns unique annotation identifying numbers for all annotations in the current chart. Also returns the type for each annotation. Callable from all languages.

FORMAT

```
                  IA      IA      IA       I         I
     GANOTINFO  (graf,annotids,annottypes,maxannots,firstannot,
                  i/o     out     out      in        in

                     I
                 numannots)
                    out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

annotids Integer array returning unique annotation identifying numbers.

annottypes Integer array returning the type of annotation according to the following code numbers:

1 Box
2 Arrow
3 Line
4 Text String

Parameters (continued)

maxannots	Integer variable greater than 0 specifying the length in words of the *annottypes* array. The *annotids* array is the same length as the *annottypes* array. This parameter is also the number of annotations to be returned.
firstannot	Integer variable indicating which annotation to start with. To start with the first annotation, set this parameter equal to one. To start with the *n*th annotaton, set it equal to *n*. This number is not necessarily the same as the annotation identifying number.
numannots	Integer variable greater than 0 holding the number of annotation identifying numbers returned in *annotids*.

DISCUSSION

GANOTINFO is called to return general information about all annotations that have been specified for the current chart. Information about a particular annotation is returned by GSYMBOL[A]INQ and GTEXTINQ. A chart can contain up to 20 annotations.

See chart on page 6-11.

GAXIS[A] $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Describes either axis on a line graph, or the non-bar axis on a bar graph. The ASCII version, callable from COBOL as well as other languages, is designated by an "A" in the procedure name.

FORMAT

```
           IA    I    R   R     I         R          I
GAXISSET (graf,axis,min,max,defaults,intervalsize,noticks,
           i/o  in   in  in   in         in         in

                I         I          I         I         I
          minorticks,gridspacing,minorgrids,scaletype,reflect)
                in        in          in        in        in

           IA    I   R   R      I          R         I
GAXISINQ (graf,axis,min,max,defaults,intervalsize,noticks
           i/o  in  out out    out        out        out

                I         I         I            I        I
          minorticks,gridspacing,minorgrids,scaletype,reflect)
               out       out       out          out      out

CALL "GAXISASET" USING graf,axis,min,max,defaults,intervalsize,
          noticks,minorticks,gridspacing,minorgrids,scaletype,
          reflect

CALL "GAXISAINQ" USING graf,axis,min,max,defaults,intervalsize,
          noticks,minorticks,gridspacing,minorgrids,scaletype,
          reflect
```

PARAMETERS:

graf Integer array containing global information about DSG/3000. The first word returns the call status.

axis Integer variable specifying the axis to be described according to the following code:

1 X-axis
2 Y-axis

min ASCII version: 16-byte array.
REAL version: real variable.

Lowest value of axis range. If you want to use the default calculated by DSG/3000 from the data, set the *defaults* parameter as specified below; whatever value you assign to *min* will be ignored.

For bar charts, *min* must <=0.

max ASCII version: 16-byte array.
REAL version: real variable

Highest value of axis range. This default is ignored if you set the *default* parameter to use the maximum value from the data.

For bar charts, *max* must be =>0.

defaults Integer variable specifying to DSG/3000 whether to use defaults or user supplied data to set minimum and maximum values for the axis range.

1. Both minimum and maximum default to values calculated by DSG/3000 from the data file (default).

Parameters (continued)

2. Minimum is set by user in *min* parameter; maxmum defaults to value calculated by DSG/3000 from the data file.

3. Maximum is set by user in *max* parameter; minimum defaults to value calculated by DSG/3000 from the data file.

4. Minimum and maximum are set by user in *min* and *max* parameters, respectively. (Note that *min* must be set to value less than *max*.)

intervalsize ASCII version: 16-byte array.
REAL version: real variable.

Specifies size of intervals in data units between major and minor tick marks along the axis. Major ticks can be labeled with data values. If you want DSG/3000 to calculate a default value, set *intervalsize* to a negative number or zero.

noticks Integer code specifying whether to display tick marks on this axis:

0 Display tick marks (default)
1 Do not display tick marks

minorticks Integer variable specifying number of minor tick marks per major tick, not to exceed 10. Minor ticks are drawn smaller than major ticks and cannot be labeled.

Parameters (continued)

gridspacing Integer variable specifying number of major ticks per grid line:

0 No grid lines (default)
1 Each major tick has a grid line
n Every *n*th major tick has a grid line

minorgrids Integer variable specifying whether or not grid lines through minor ticks are to be drawn. This parameter is applicable only when *minorticks* is greater than 0, *gridspacing* is equal to 1, and *scaletype* is equal to 2 (logarithmic scaling).

0 No minor grid lines (default)
1 Draw minor grid lines

scaletype Integer code specifying linear or logarithmic scaling:

1 linear (default)
2 logarithmic

Specifying 0 results in the default, linear scaling.

reflect Draws the major and minor tick marks that have been specified by the *minorticks* and *noticks* parameters on the opposite axis according to the following code:

0 no reflection (default)
1 reflect

DISCUSSION

Since GAXIS[A]SET sets a single axis, it can be called twice for line graphs to set x and y axes, and once for bar charts to set the Y-axis. The X-axis (bar axis) of the bar chart is specified by GBAR[A]SET.

The *reflect* parameter is used to draw tick marks on the opposite margin of the chart. For example, if you are setting the X-axis of a line chart, tick marks are drawn along the bottom axis of the graph. Setting the reflect parameter to 1 will also draw the tick marks along the upper axis of the graph.

See page 6-19 for line graph discussion and page 6-12 for bar chart discussion.

GBAR[A] $\left\{\begin{matrix}\text{SET}\\\text{INQ}\end{matrix}\right\}$

Sets or returns unique bar chart characteristics, including bar axis scaling. The ASCII version, callable from COBOL as well as other languages, is designated by an "A" in the procedure name.

FORMAT

```
                IA     I           I        R     I          I
      GBARSET (graf,grouping,orientation,start,defaults,numintervals,
               i/o    in          in       in    in        in

                    I
                gridspacing)
                    in


                IA     I           I        R      I        I
      GBARINQ (graf,grouping,orientation,start,defaults,numintervals,
                i/o   out        out      out    out       out

                    I
                gridspacing)
                    out


      CALL "GBARASET" USING graf,grouping,orientation,start,defaults,
               numintervals,gridspacing

      CALL "GBARAINQ" USING graf,grouping,orientation,start,defaults
               numintervals,gridspacing
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

grouping Integer code specifying whether the bars will be stacked or clustered:

1 stacked bars (default)
2 clustered bars

Specifying a 0 results in the default, stacked bars.

orientation Integer code specifying whether the bars will be drawn vertically or horizontally:

1 vertical bars (default)
2 horizontal bars

start ASCII version: 16-byte array
REAL version: real variable

The minimum value of the bar axis range. If you want blank space to precede the first bar in the graph, supply a number less than the lowest value taken by the independent variable. This value is used or ignored depending on the value of the *defaults* parameter.

defaults Integer code specifying whether to take the minimum value from the start parameter or from the data:

1 minimum from data (default)
2 use value given by start

Parameters (continued)

numintervals Integer variable specifying number of intervals to be plotted, not to exceed 60. A negative number or 0 specifies that DSG/3000 should take the default from the data.

gridspacing Integer code specifying number of intervals between grid lines:

0 no grid lines (default)
1 one bar between grid lines
n *n* bars between grid lines

The value of *n* must be less than the number of intervals in the *numintervals* parameter.

DISCUSSION

See page 6-12.

GCHARTINFO

Returns user name and account name of chart creator and the date of creation. Callable from all languages.

FORMAT

```
                 IA      BA         BA
      GCHARTINFO (graf,createdate,creator)
                 i/o     out        out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

createdate 8-byte array returning the date the chart was created. The format is "mm/dd/yy".

creator 17-byte array returning user name and account name of chart creator. Format is "user.account".

GCHARTYPE $\begin{Bmatrix}\text{SET}\\\text{INQ}\end{Bmatrix}$

Sets or returns the chart type. Callable from all languages.

FORMAT

```
                    IA       I
      GCHARTYPESET (graf,charttype)
                    i/o     in

                    IA      I
      GCHARTYPEINQ (graf,charttype)
                    i/o     out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

charttype Integer code specifying chart type:

1 Annotations only (default)
2 Line (includes Scattergrams)
3 Bar
4 Pie

DATA DECLARATIONS - GCHARTYPESET, GCHARTYPEINQ

```
COBOL:        DATA DIVISION.
              01  GRAF.
                  05  GSTATUS      PIC S9(4) COMP.
                  05  PARAMETER    PIC S9(4) COMP.
                  05  ELEMENT      PIC S9(4) COMP.
                  05  FILE-ERR     PIC S9(4) COMP.
                  05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
              01  CHART-TYPE       PIC S9(4) COMP.
                .
                .
                .
              CALL "GCHARTYPESET" USING GRAF, CHART-TYPE.

BASIC:        10  REM   G IS GRAF ARRAY    C IS CHART-TYPE
              15  REM   STATUS IS G[1]
              20  INTEGER G[3700], C
                .
                .
                .
              50  CALL GCHARTYPESET (G[*],C)

FORTRAN:      INTEGER GRAF(1600), CHARTTYPE
              INTEGER STATUS
        C
              EQUIVALENCE (STATUS,GRAF(1))
                .
                .
                .
              CALL GCHARTYPESET (GRAF, CHARTTYPE)

SPL:          INTEGER ARRAY GRAF(0:1599);
              DEFINE
                STATUS  = GRAF (0)#;
              INTEGER CHART'TYPE;
                .
                .
                .
              GCHARTYPESET (GRAF, CHART'TYPE);


PASCAL:     TYPE
                SMALL_INT=-32768...32767
                GRAFTYPE=record
                      status:small_int;
                      parameter :small-int;
                      element   :small_int;
                      file_err  :small_int;
                      graffarea :small_int;
                    end;

             VAR
               GRAF      :GRAFTYPE;
               CHART_TYPE: small_int;
             .
             .
             .
             GCHARTYPESET (GRAF,CHART_TYPE);
```

GCLOSEFILE

Closes the current chart file. Callable from all languages.

FORMAT

```
                 IA
      GCLOSEFILE (graf)
                 i/o
```

PARAMETERS

graf	Integer array containing global information used by DSG/3000. The first word returns the call status.

DISCUSSION

GCLOSEFILE updates any outstanding chart records and closes the chart file. If the chart file was specified as temporary when it was created, it is purged.

DATA DECLARATIONS - GCLOSEFILE

```
COBOL:       DATA DIVISION.
             01  GRAF.
                 05  GSTATUS      PIC S9(4) COMP.
                 05  PARAMETER    PIC S9(4) COMP.
                 05  ELEMENT      PIC S9(4) COMP.
                 05  FILE-ERR     PIC S9(4) COMP.
                 05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
               .
               .
               .
             CALL "GCLOSEFILE" USING GRAF.

BASIC:       10  REM    G IS GRAF ARRAY
             15  REM    STATUS IS G[1]
             20  INTEGER G[3700]
               .
               .
               .
             50  CALL GCLOSEFILE (G[*])

FORTRAN:     INTEGER GRAF(1600)
             INTEGER STATUS
        C
             EQUIVALENCE (STATUS,GRAF(1))
               .
               .
               .
             CALL GCLOSEFILE (GRAF)

SPL:         INTEGER ARRAY GRAF(0:1599);
             DEFINE
               STATUS    = GRAF (0)#;
             INTEGER STATUS=GRAF(0);
               .
               .
               .
             GCLOSEFILE (GRAF);

PASCAL:    TYPE
               SMALL_INT=-32768...32767
               GRAFTYPE=record
                     status:small_int;
                     parameter  :small-int;
                     element    :small_int;
                     file_err   :small_int;
                     graffarea  :small_int;
                   end;

            VAR
              GRAF      :GRAFTYPE;
             .
             .
             .
             GCLOSEFILE (GRAF);
```

GCOMMENT $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Sets or returns comments about a chart in the chart file. Callable from all languages.

FORMAT

```
                  IA     BA          BA          I
     GCOMMENTSET  (graf,chartname,commentsbuf,commentslen)
                   i/o    in          in          in

                  IA      BA        BA                I
     GCOMMENTINQ  (graf,chartname,commentsbuf,commentsbuflen,
                   i/o     in       out               in

                     I             I
                   commentslen,charttype)
                     out           out
```

PARAMETERS

graf — Integer array containing global information used by DSG/3000. The first word returns the call status.

chartname — Byte array holding name of chart whose comments are being set or returned.

Parameters (continued)

commentsbuf Byte array holding comments about the chart.

commentslen Integer variable giving length of comments in bytes. Maximum is 80 characters.

commentsbuflen Integer variable giving length in bytes of the buffer to receive the comments.

charttype Integer variable returning chart type according to the following code:

1 Annotations only
2 Line (includes Scattergrams)
3 Bar
4 Pie

DISCUSSION

GCOMMENT is a helpful documentation tool, especially when you are using your chartfile as a portfolio containing several charts.

GCOPYCHART

Makes a copy of one chart, including all specifications. Callable from all languages. .

FORMAT

```
                   IA       BA         BA        BA       BA
      GCOPYCHART (graf,fromchart,fromfile,tochart,tofile)
                   i/o      in         in        in       in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

fromchart 16-byte array holding name of the chart being copied.

fromfile 36-byte array holding name of file containing chart being copied.

tochart 16-byte array holding name of destination chart.

tofile 36-byte array holding name of file containing destination chart.

DISCUSSION

Before calling GCOPYCHART, be sure that the file specified in the *tofile* parameter has been created with a previous call to GCREATEFILE. The *tofile* can be any DSG/3000 chartfile.

If you want to rename a chart, call GCOPYCHART and pass the same file name to the *fromfile* and *tofile* parameters. You can then delete the old chart by calling GDELCHART.

GCOPYCHART compacts the amount of space needed for the new chart.

GCREATECHART

Initializes a new chart and adds the name to the current chart file directory. Callable from all languages.

FORMAT

```
                   A       BA
      GCREATECHART (graf,chartname)
                   i/o      in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

chartname Byte array holding unique name of new chart. Must be 16 characters starting with a letter.

DISCUSSION

GCREATECHART initializes records containing chart attributes, labels, annotations, and data definition to their defaults. The chart identified in GCREATECHART becomes the current chart, meaning that subsequent chart, data and graphing procedures refer to this chart. The chart name must be unique to the chart file in which it resides.

DATA DECLARATIONS - GCREATECHART

```
COBOL:        DATA DIVISION.
              01  GRAF.
                  05  GSTATUS       PIC S9(4) COMP.
                  05  PARAMETER     PIC S9(4) COMP.
                  05  ELEMENT       PIC S9(4) COMP.
                  05  FILE-ERR      PIC S9(4) COMP.
                  05  FILLER        PIC S9(4) COMP OCCURS 1596 TIMES.
              01  CHART-NAME        PIC X(16).
                 .
                 .
                 .
               CALL "GCREATECHART" USING GRAF, CHART-NAME.

 BASIC:        10  REM   G IS GRAF ARRAY   C$ IS CHART NAME
               15  REM   STATUS IS G[1]
               20  INTEGER G[3700]
               30  DIM C$[16]
                  .
                  .
                  .
               50  CALL GCREATECHART (G[*],C$)

 FORTRAN:      INTEGER GRAF(1600)
               INTEGER STATUS
               CHARACTER*16 CHARTNAME
        C
               EQUIVALENCE (STATUS,GRAF(1))
                  .
                  .
                  .
               CALL GCREATECHART (GRAF, CHARTNAME)

 SPL:          INTEGER ARRAY GRAF(0:1599);
               DEFINE
                 STATUS   = GRAF (0)#;
               BYTE ARRAY CHART'NAME(0:15);
                  .
                  .
                  .
               GCREATECHART (GRAF, CHART'NAME);
```

```
PASCAL:        TYPE
                 SMALL_INT=-32768..32767
                 GRAFTYPE=record
                        status    :small_int;
                        parameter:small_int;
                        file_err :small_int;
                        graffarea:small_int;
                      end;
               VAR
                 GRAF      :GRAFTYPE;
                 CHART_NAME:PACKED ARRAY [1..16] OF CHAR;
                  .
                  .
                  .
                 GCREATECHART (GRAF,CHART_NAME);
```

GCREATEFILE

Builds and initializes a new chart file. Callable from all languages.

FORMAT

```
                 IA    BA         I         I
     GCREATEFILE (graf,filename,numcharts,domain)
                 i/o   in        in        in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

filename Byte array holding name of chart file. Must be 36 alphanumeric characters (including lockword, group, and account) beginning with an alphabetic character.

numcharts Integer variable giving the maximum number of charts to be included in this chart file.

domain Integer variable indicating whether the file is permanent:

1 Temporary (purged by GCLOSEFILE)
2 Permanent

If you supply a 0, *domain* will default to a permanent file.

DISCUSSION

A call to GCREATEFILE builds and opens the chart file. The file is opened with exclusive access, and file code 1083 is assigned to the chart file.

Since the chart file directory can accommodate up to 50 entries, the maximum number of charts in each chart file is 50.

DATA DECLARATIONS - GCREATEFILE

```
COBOL:        DATA DIVISION.
              01  GRAF.
                  05  GSTATUS      PIC S9(4) COMP.
                  05  PARAMETER    PIC S9(4) COMP.
                  05  ELEMENT      PIC S9(4) COMP.
                  05  FILE-ERR     PIC S9(4) COMP.
                  05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
              01  CHART-FILE-INFO.
                  05  FILE-NAME    PIC X(36) VALUE "NEWCHART ".
                  05  NUM-CHARTS   PIC S9(4) COMP VALUE 25.
                  05  DOMAIN       PIC S9(4) COMP VALUE 2.
                .
                .
                .
              CALL "GCREATEFILE" USING GRAF, FILE-NAME,
                                       NUM-CHARTS, DOMAIN.

BASIC:        10  REM   G IS GRAF ARRAY    N IS NUM-CHARTS
              20  REM   D IS FILE DOMAIN   F$ IS FILE NAME
              25  REM   STATUS IS G[1]
              30  INTEGER G[3700], N, D
              40  DIM F$[36]
              50  F$="NEWCHART "
              60  N=25
              70  D=2
                .
                .
                .
             100  CALL GCREATEFILE (G[*],F$,N,D)

FORTRAN:      INTEGER GRAF(1600), NUMCHARTS, DOMAIN
              INTEGER STATUS
              CHARACTER*36 FILENAME
```

```
        C
              EQUIVALENCE (STATUS,GRAF(1))
        C
              FILENAME="NEWCHART "
              NUMCHARTS=25
              DOMAIN=2
                 .
                 .
                 .
              CALL GCREATEFILE (GRAF, FILENAME, NUMCHARTS, DOMAIN)

SPL:          INTEGER ARRAY GRAF(0:1599);
              DEFINE
                STATUS  = GRAF (0)#;
              BYTE ARRAY FILE'NAME(0:35):="NEWCHART ";
              INTEGER NUMCHARTS:=25,
                      DOMAIN:=2;
                 .
                 .
                 .
              GCREATEFILE (GRAF, FILE'NAME, NUMCHARTS, DOMAIN);



PASCAL:       TYPE
                small_int=-32768..32767;
                graftype =record
                       status     :small_int;
                       parameter  :small_int;
                       element    :small_int;
                       file_err   :small_int;
                       graffarea  :array[1..1596] of small_int;
                     end;
              VAR
                GRAF      :graftype;
                FILENAME  :packed array [1...36] of char;
                NUM_CHART:small_int;
                DOMAIN    :small_int;
               .
               .
               .
              GCREATEFILE(GRAF,FILENAME,NUM_CHART,DOMAIN);
```

GDATAFILE $\left\{\begin{matrix}\text{SET}\\\text{INQ}\end{matrix}\right\}$

Sets or returns the data file name and the value indicating missing data. Callable from all languages.

FORMAT

```
                    IA     BA       BA       I
      GDATAFILESET (graf,filename,missbuf,misslen)
                    i/o    in       in       in

                    IA     BA        BA       I         I
      GDATAFILEINQ (graf,filename,missbuf,missbuflen,misslen)
                    i/o    out       out      in        out
```

PARAMETERS

graf — Integer array containing global information used by DSG/3000. The first word returns the call status.

filename — 36-byte array holding the fully qualified name of the data file. Must be terminated by a space or semi-colon.

missbuf — 12-byte array holding the numeric value representing missing data in the data file. A buffer of all blanks is treated as 0.

misslen — Integer variable giving length in bytes of the missing data value.

Parameters (continued)

missbuflen Integer variable giving maximum number of bytes to be returned in *missbuf*.

DISCUSSION

See page 3-21.

DATA DECLARATIONS - GDATAFILESET, GDATAFILEINQ

COBOL:

```
DATA DIVISION.
01  GRAF.
    05  GSTATUS        PIC S9(4) COMP.
    05  PARAMETER      PIC S9(4) COMP.
    05  ELEMENT        PIC S9(4) COMP.
    05  FILE-ERR       PIC S9(4) COMP.
    05  FILLER         PIC S9(4) COMP OCCURS 1596 TIMES.
01  DATA-FILE-INFO.
    05  FILE-NAME      PIC X(36).
    05  MISS-BUF       PIC X(16).
    05  MISS-LEN       PIC S9(4) COMP.
    05  MISS-BUF-LEN   PIC S9(4) COMP.
     .
     .
     .
CALL "GDATAFILESET" USING GRAF, FILE-NAME,
                          MISS-BUF, MISS-LEN.
```

BASIC:

```
10  REM   G IS GRAF ARRAY     F$ IS DATA FILE NAME
20  REM   M$ IS MISS-BUF      M1 IS MISS-LEN
25  REM   M2 IS MISS-BUF-LEN
30  REM   STATUS IS G[1]
40  INTEGER G[3700], M1, M2
50  DIM F$[36], M$[16]
     .
     .
     .
100 CALL GDATAFILESET (G[*],F$,M$,M1)
```

```
FORTRAN:       INTEGER GRAF(1600), MISSLEN, MISSBUFLEN
               CHARACTER*36 FILENAME, MISSBUF*16
               INTEGER STATUS
        C
               EQUIVALENCE (STATUS,GRAF(1))
                 .
                 .
                 .
               CALL GDATAFILESET (GRAF, FILENAME, MISSBUF, MISSLEN)

SPL:           INTEGER ARRAY GRAF(0:1599);
               DEFINE
                 STATUS   = GRAF (0)#;
               BYTE ARRAY FILE'NAME(0:35);
               BYTE ARRAY MISS'BUF(0:15);
               INTEGER MISS'LEN, MISS'BUF'LEN;
                 .
                 .
                 .
               GDATAFILESET (GRAF, FILE'NAME, MISS'BUF, MISS'LEN);


PASCAL:        TYPE
                 SMALL_INT=-32768..32767;
                 GRAFTYPE =record
                        status      :small_int;
                        parameter   :small_int;
                        element     :small_int;
                        file_err    :small_int;
                        graffarea   :array[1..1596] of small_int;
                      end;
               VAR
                 GRAF       :GRAFTYPE;
                 FILENAME   :PACKED ARRAY [1..36] OF CHAR;
                 MISSBUFLEN:SMALL_INT;
                 MISSLEN    :SMALL_INT;
                 MISSBUF    :PACKED ARRAY [1..36] OF CHAR;
               .
               .
               .
               GDATAFILESET(GRAF,FILENAME,MISSBUF,MISSLEN);
```

GDATAINFO

Returns information about the data specification for the current chart. Callable from all languages.

FORMAT

```
                IA    I        I        I          I         I
     GDATAINFO (graf,numdefns,numtrans,numdvars,explodelen,qualen,
               i/o   out      out      out        out       out

                  I       I
                translen,subsetlen)
                   out     out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

numdefns Integer variable returning number of variable definitions.

numtrans Integer variable returning number of transformed variables.

numdvars Integer variable returning number of Y-axis variables.

explodelen Integer variable returning the length in bytes of the explosion expression.

qualen Integer variable returning the length in bytes of the Y-axis variable qualifications.

Parameters (continued)

translen Integer variable returning the length in bytes of a single transformation expression.

subsetlen Integer variable returning the length in bytes of the data file subset specification.

GDEFN $\left\{\begin{matrix}\textbf{SET}\\ \textbf{INQ}\end{matrix}\right\}$

Sets or returns variable definitions for a data file. Definitions for all of the variables are set or returned together. Required for all chart types except slides. Callable from all languages.

FORMAT

```
                IA     I      BA     IA       I      IA     IA
     GDEFNSET (graf,numdefn,varbuf,typebuf,format,posbuf,lenbuf)
                i/o    in     in     in       in     in     in

                IA      I        I       I       BA      I
     GDEFNINQ (graf,firstdefn,maxdefn,numdefn,varbuf,varbuflen,
                i/o     in       in      out     out     in

                 IA      I      IA     IA
               typebuf,format,posbuf,lenbuf)
                 out     out    out    out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status. If an error is detected, the second word contains the number of the parameter in error.

Parameters (continued)

numdefn Integer variable specifying number of variable definitions. Maximum is 9.

varbuf Byte array containing a collection of 16-byte names, one for each data variable.

You cannot use any of the following reserved words as variable names:

ABS
AND
CUMULATE
LN
LOG
MOD
MOVEAVG
NOT
OR
SQRT

typebuf Integer array specifying variable types according to the following code:

1 text (ASCII)
2 numeric (ASCII)

format Integer code specifying free or fixed format data file:

1 free format
2 fixed format

posbuf Integer array specifying the position of a variable within a data record. In free format, *posbuf* is a field number (where the first field is field 1). In fixed format, it is a byte offset, where the first byte in the record has offset 0.

Parameters (continued)

lenbuf Integer array specifying lengths of data values. Only used for fixed format.

firstdefn Integer variable specifying which variable definition is the 1st one to be returned. The definitions are numbered beginning with one.

maxdefn Integer variable specifying the maximum number of definitions to be returned.

varbuflen Integer variable specifying length in bytes of the buffer to receive the variable names.

DISCUSSION

Independent and Dependent variables: page 2-3; Data File Format: page 3-2.

GDEFNINQ can be used to return a subset of the defined variables. For example, if 9 variables have been defined and you want the 6th through 9th to be returned, set *firstdefn* equal to 6 and *maxdefn* equal to 4.

DATA DECLARATIONS - GDEFNSET, GDEFNINQ

```
COBOL:        DATA DIVISION.
              01  GRAF.
                  05  GSTATUS      PIC S9(4) COMP.
                  05  PARAMETER    PIC S9(4) COMP.
                  05  ELEMENT      PIC S9(4) COMP.
                  05  FILE-ERR     PIC S9(4) COMP.
                  05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
              01  VARIABLE-DEFINITIONS.
                  05  NUM-DEFN          PIC S9(4) COMP VALUE 2.
                  05  VAR-BUF.
                      10  VAR-VALUES    PIC X(16) OCCURS 2 TIMES.
                  05  TYPE-BUF.
                      10  TYPE-VALUES   PIC S9(4) COMP OCCURS 2 TIMES.
                  05  FORMAT            PIC S9(4) COMP.
                  05  POS-BUF.
                      10  POS-VALUES    PIC S9(4) COMP OCCURS 2 TIMES.
                  05  LEN-BUF.
                      10  LEN-VALUES    PIC S9(4) COMP OCCURS 2 TIMES.
                  05  FIRST-DEFN        PIC S9(4) COMP.
                  05  MAX-DEFN          PIC S9(4) COMP.
                  05  VAR-BUF-LEN       PIC S9(4) COMP.
                .
                .
                .
              CALL "GDEFNSET" USING GRAF, NUM-DEFN, VAR-BUF,
                                    TYPE-BUF, FORMAT,
                                    POS-BUF, LEN-BUF.

BASIC:         10  REM   G IS GRAF ARRAY      N IS NUM-DEFN
               20  REM   V$ IS VAR-BUF        T IS TYPE-BUF
               30  REM   F1 IS FORMAT         P IS POS-BUF
               40  REM   L IS LEN-BUF         F2 IS FIRST-DEFN
               50  REM   M IS MAX-DEFN        V IS VAR-BUF-LEN
               60  REM   STATUS IS G[1]
               70  INTEGER G[3700], N, F1, F2, M, V
               80  INTEGER T(2), P(2), L(2)
               90  DIM V$[32]
              100  N=2
                .
                .
                .
              150  CALL GDEFNSET (G[*],N,V$,T[*],F1,P[*],L[*])

FORTRAN:       INTEGER GRAF(1600), NUMDEFN, TYPEBUF(2), FORMAT
               INTEGER POSBUF(2), LENBUF(2), FIRSTDEFN, MAXDEFN
               CHARACTER*16 VARBUF(2)
               INTEGER STATUS
        C
               EQUIVALENCE (STATUS,GRAF(1))
        C
               NUMDEFN=2
                 .
                 .
                 .
               CALL GDEFNSET (GRAF, NUMDEFN, VARBUF, TYPEBUF,
              *               FORMAT, POSBUF, LENBUF)
```

SPL:

```
INTEGER ARRAY GRAF(0:1599),
              TYPE'BUF(0:1),
              POS'BUF(0:1),
              LEN'BUF(0:1);
DEFINE
  STATUS  = GRAF (0)#;
INTEGER NUM'DEFN:=2,
        FORMAT,
        FIRST'DEFN,
        MAX'DEFN,
        VAR'BUF'LEN:=32;
BYTE ARRAY VAR'BUF(0:31);
   .
   .
   .
GDEFNSET (GRAF, NUM'DEFN, VAR'BUF, TYPEBUF, FORMAT,
          POS'BUF, LEN'BUF);
```

PASCAL:

```
TYPE
  SMALL_INT=-32768..32767;
  GRAFTYPE =record
          status      :small_int;
          parameter   :small_int;
          element     :small_int;

          file_err    :small_int;
          graffarea   :array[1..1596] of small_int;
        end;
VAR
  GRAF       :GRAFTYPE;
  VARBUF     :ARRAY [1...2] OF CHAR16;
  NUMDEFN,FORMAT,FIRSTDEFN,MAXDEFN:SMALL_INT;
  LENBUF,TYPEBUF,POSBUF:ARRAY [1...2] OF SMALL_INT;

GDEFNSET(GRAF,NUMDEFN,VARBUF,TYPEBUF,FORMAT,POSBUF,LENBUF);
```

GDELCHART

Removes a chart from the chart file. Callable from all languages.

FORMAT

```
                 A      BA
      GDELCHART (graf,chartname)
                i/o      in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

chartname 16-byte array holding name of chart to be deleted.

DISCUSSION

The chart file containing the chart you want deleted must be opened before you call GDELCHART.

GDELCHART removes the chart name from the directory and corresponding chart records are removed from the file.

GDELETEANOT

Deletes an annotation from the chart description. Callable from all languages.

FORMAT

```
                   IA      I
      GDELETEANOT (graf,annotid)
                   i/o     in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

annotid Integer variable uniquely identifying the annotation.

DISCUSSION

GSYMBOLINQ and GTEXTINQ can be called before GDELETEANOT to verify the identifying number of the annotation you wish to delete.

Deleting an annotation compacts the remaining ones so that any new annotations are added to the end of the list.

GDELETEFIGURE

Deletes the specified figure from the specified figure file. Callable from all languages.

FORMAT

```
                     IA      BA          BA
     GDELETEFIGURE (graf,figurename,figurefile)
                     in      in          in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

figurename 16-byte array containing the name of the figure to be deleted.

figurefile 36-byte array containing the name of the figure file in which the figure to be deleted resides.

GDEPVAR {SET / INQ}

Sets or returns all Y-axis (dependent) variables as a group. Required for line graphs and bar charts. Callable from all languages.

FORMAT

```
                   IA    I       BA     BA      I
      GDEPVARSET (graf,numvars,varbuf,qualbuf,qualen)
                  i/o    in      in     in      in
```

```
                   IA    I       BA        I      BA
      GDEPVARINQ (graf,numvars,varbuf,varbuflen,qualbuf,
                  i/o    out     out      in       out

                       I        I
                  qualbuflen,qualen)
                      in       out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status. If an error is detected, the second word contains the number of the parameters in error.

numvars Integer variable specifying number of Y-axis (dependent) variables. Maximum number allowed is 8.

Parameters (continued)

varbuf Byte array holding 16-byte names of variables.

qualbuf Byte array holding qualifications. All qualifications are the same length, as specified by *qualen*.

qualen Integer variable specifying length in bytes of a single qualification. A length of 0 implies no qualifications.

varbuflen Integer variable specifying size in bytes of buffer to receive the variable names.

qualbuflen Integer variable specifying length in bytes of buffer to receive the qualifications.

DISCUSSION

These procedures are not used for pie charts or slides.

GDEPVARINQ will return a warning message if you set *varbuflen* too small to receive all of the variable names of if you set *qualbuflen* too small to receive all of the qualifications.

For information about Dependent Variables: page 2-3.
For information about qualifying dependent variables: page 3-14.

DATA DECLARATIONS - GDEPVARSET, GDEPVARINQ

```
COBOL:          DATA DIVISION.
                01  GRAF.
                    05  GSTATUS         PIC S9(4) COMP.
                    05  PARAMETER       PIC S9(4) COMP.
                    05  ELEMENT         PIC S9(4) COMP.

                    05  FILE-ERR        PIC S9(4) COMP.
                    05  FILLER          PIC S9(4) COMP OCCURS 1596 TIMES.
                01  DEPENDENT-VAR.
                    05  NUM-VARS        PIC S9(4) COMP VALUE 3.
                    05  VAR-BUF.
                        10  VAR-VALUES     PIC X(16) OCCURS 3 TIMES.
                    05  QUAL-BUF        PIC X(50).
                    05  QUALEN          PIC S9(4) COMP VALUE 10.
                    05  VAR-BUF-LEN     PIC S9(4) COMP VALUE 48.
                    05  QUAL-BUF-LEN    PIC S9(4) COMP VALUE 50.
                  .
                  .
                  .
                CALL "GDEPVARSET" USING GRAF, NUM-VARS, VAR-BUF,
                                        QUAL-BUF, QUALEN.

BASIC:          10  REM   G IS GRAF ARRAY     N IS NUM-VARS
                20  REM   V$ IS VAR-BUF       Q$ IS QUAL-BUF
                30  REM   Q1 IS QUALEN        V IS VAR-BUF-LEN
                40  REM   Q2 IS QUAL-BUF-LEN
                50  REM   STATUS IS G[1]
                60  INTEGER G[3700], N, Q1, V, Q2
                70  DIM V$[48], Q$[50]
                80  REM
                90  N=3
               100  Q1=10
               110  V=48
               120  Q2=50
                  .
                  .
                  .
               150  CALL GDEPVARSET (G[*],N,V$,Q$,Q1)

FORTRAN:        INTEGER GRAF(1600), NUMVARS, QUALEN, VARBUFLEN
                INTEGER QUALBUFLEN
                CHARACTER*16 VARBUF(3), QUALBUF*50
                INTEGER STATUS
          C
                EQUIVALENCE (STATUS,GRAF(1))
          C
                DATA NUMVARS/3/, QUALEN/10/, VARBUFLEN/48/
                DATA QUALBUFLEN/50/
                  .
                  .
                  .
                CALL GDEPVARSET (GRAF, NUMVARS, VARBUF,
               *                 QUALBUF, QUALEN)
```

SPL:

```
INTEGER ARRAY GRAF(0:1599);
DEFINE
  STATUS  = GRAF (0)#;
INTEGER NUM'VARS:=3,
        QUALEN:=10,
        VAR'BUF'LEN:=48,
        QUAL'BUF'LEN:=50;
BYTE ARRAY VAR'BUF(0:47),
           QUAL'BUF(0:49);

   .
   .
   .
GDEPVARSET (GRAF, NUM'VARS, VAR'BUF, QUAL'BUF,
            QUALEN);
```

PASCAL:

```
TYPE
  SMALL_INT=-32768..32767;
  GRAFTYPE =record
          status     :small_int;
          parameter  :small_int;
          element    :small_int;
          file_err   :small_int;
          graffarea  :array[1..1596] of small_int;
        end;
VAR
  GRAF       :GRAFTYPE;
  VARBUF     :ARRAY [1...3] OF CHAR16;
  NUMVARS,QUALEN,VARBUFLEN,QUALBUFLEN:SMALL_INT;
  QUALBUF    :PACKED ARRAY [1...50] OF CHAR;
.
.
.
GDEPVARSET(GRAF,NUMVARS,VARBUF,QUALBUF,QUALEN);
```

GDEVICE $\left\{\begin{array}{l}\text{SET}\\\text{INQ}\end{array}\right\}$

Sets or returns plotting device and device-dependent specifications. Callable from all languages.

FORMAT

```
                  IA    BA     I      I       I        I
      GDEVICESET (graf,device,hpib,copies,penspeed,chartype)
                  i/o   in     in     in      in       in


                  IA    BA     I      I       I        I
      GDEVICEINQ (graf,device,hpib,copies,penspeed,chartype)
                  i/o   out    out    out     out      out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

device 6-byte array holding model number and letter for graphing device. Must have trailing blank. For example:

7221A

See Section I for a list of supported devices.

hpib Integer variable holding the HP-IB address of an HP-IB device. The HP-IB address is set on the back of the plotting device. (0-32)

Parameters (continued)

If the device is not an HP-IB device, set the parameter to -1.

copies Integer variable specifying the number of copies to be produced. Must be greater than or equal to 0.

penspeed Integer code specifying pen movement speed for plotters:

0 fast (default)
1 slow
2 transparency speed (slow, with pauses of 15 minutes between pen changes to allow ink to dry completely.)

chartype Integer code specifying character type:

1 Hardware characters (faster)
3 Software characters (slower)

GERRORMSG

Returns an error message for the previous error. Callable from all languages.

FORMAT

```
               IA   BA        I         I
     GERRORMSG (graf,msgbuf,msgbuflen,msglen)
               i/o  out      in        out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word contains the error number for which a message will be returned and returns the call status.

msgbuf 72-byte array returning error message.

msgbuflen Integer variable giving length in bytes of message buffer.

msglen Integer variable giving length in bytes of returned message.

DISCUSSION

GERRORMSG clears the error flags in the communications area (GRAF) after returning the error message.

GEXECDATA

Executes all data specifications set by previous calls to data procedures. Required before a chart is drawn (not required for slides). Callable from all languages.

FORMAT

```
                 IA
     GEXECDATA (graf)
                i/o
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

DISCUSSION

GEXECDATA prepares and manipulates the data for the current chart. If a chart is drawn, modified, and drawn again, GEXECDATA prepares the data as necessary due to changes. Thus, data preparation is optimized on subsequent calls to GEXECDATA.

COBOL:

```
DATA DIVISION.
01  GRAF.
    05  GSTATUS      PIC S9(4) COMP.
    05  PARAMETER    PIC S9(4) COMP.
    05  ELEMENT      PIC S9(4) COMP.
    05  FILE-ERR     PIC S9(4) COMP.
    05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
  .
  .
  .
CALL "GEXECDATA" USING GRAF.
```

BASIC:

```
10  REM   G IS GRAF ARRAY
15  REM   STATUS IS G[1]
20  INTEGER G[3700]
  .
  .
  .
50  CALL GEXECDATA (G[*])
```

FORTRAN:

```
      INTEGER GRAF(1600)
      INTEGER STATUS
C
      EQUIVALENCE (STATUS,GRAF(1))
        .
        .
        .
      CALL GEXECDATA (GRAF)
```

SPL:

```
INTEGER ARRAY GRAF(0:1599);
DEFINE
  STATUS  = GRAF (0)#;
  .
  .
  .
GEXECDATA (GRAF);
```

PASCAL:

```
TYPE
   SMALL_INT=-32768..32767;
   GRAFTYPE =record
          status      :small_int;
          parameter   :small_int;
          element     :small_int;
          file_err    :small_int;
          graffarea   :array[1..1596] of small_int;
        end;
VAR
  GRAF       :GRAFTYPE;
.
.
.
GEXECDATA(GRAF);
```

GEXECHART

Draws the current chart. Callable from all languages.

FORMAT

```
                 IA      I
      GEXECHART (graf,optimize)
                i/o     in
```

PARAMETERS

graf	Integer array containing global information used by DSG/3000. The first word returns the call status.
optimize	Integer flag indicating whether or not to optimize chart production: 0 Do not optimize 1 Optimize where possible (Not implemented at this time.)

DATA DECLARATIONS - GEXECHART

```
COBOL:        DATA DIVISION.
              77  OPTIMIZE        PIC S9(4) COMP.
              01  GRAF.
                  05  GSTATUS     PIC S9(4) COMP.
                  05  PARAMETER   PIC S9(4) COMP.
                  05  ELEMENT     PIC S9(4) COMP.
                  05  FILE-ERR    PIC S9(4) COMP.
                  05  FILLER      PIC S9(4) COMP OCCURS 1596 TIMES.
                .
                .
                .
              CALL "GEXECHART" USING GRAF, OPTIMIZE.

BASIC:        10  REM   G IS GRAF ARRAY    O IS OPTIMIZE
              15  REM   STATUS IS G[1]
              20  INTEGER G[3700], O
                .
                .
                .
              50  CALL GEXECHART (G[*],O)

FORTRAN:      INTEGER GRAF(1600), OPTIMIZE
              INTEGER STATUS
        C
              EQUIVALENCE (STATUS,GRAF(1))
                .
                .
                .
              CALL GEXECHART (GRAF, OPTIMIZE)

SPL:          INTEGER ARRAY GRAF(0:1599);
              DEFINE
                STATUS  = GRAF (0)#;
              INTEGER OPTIMIZE;
                .
                .
                .
              GEXECHART (GRAF, OPTIMIZE);

PASCAL:       TYPE
                 SMALL_INT=-32768..32767
                 GRAFTYPE=record
                        status   :small_int;
                        parameter:small_int;
                        file_err :small_int;
                        graffarea:small_int;
                      end;
              VAR
                 GRAF      :GRAFTYPE;
                 OPTIMIZE:small_int;
              .
              .
              .
              GEXECHART (GRAF,OPTIMIZE);
```

GEXECFIGURE

Creates a figure in the figure file from the current chart in the current chart file as though it were actually being drawn on a device. If the figure file does not already exist, it is created at this time. Callable from all languages.

FORMAT

```
                    IA     BA           BA          BA
      GEXECFIGURE (graf,figurename,figurefile,commentbuf)
                   i/o     in           in          in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

figurename 16-byte array containing the name of the figure to be created.

figurefile 36-byte array containing the name of the figure file in which the figure will reside. Must be 36 alphanumeric characters (including lockword, group, and account) beginning with an alphabetic character.

commentbuf 128-byte array containing any comment to be associated with this figure.

DISCUSSION

Figures created by this procedure are stored in a figure file for use by other subsystems, such as HPDRAW. All figure information, chart design and data, stored in the file.

GEXPLODE $\left\{\begin{matrix}\textbf{SET}\\ \textbf{INQ}\end{matrix}\right\}$

Sets or returns pie chart explosion specifications. Used only with pie charts. Callable from all languages.

FORMAT

```
                      IA     BA          I
        GEXPLODESET (graf,explodebuf,explodelen)
                      i/o   in           in


                      IA     BA           I            I
        GEXPLODEINQ (graf,explodebuf,explodebuflen,explodelen)
                      i/o    out         in           out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

explodebuf Byte array holding the explosion specification.

explodebuflen Integer variable giving length in bytes of explosion buffer.

explodelen Integer variable giving length in bytes of explosion specification.

DISCUSSION

Exploding a segment of a pie chart means that it will be drawn slightly separated from the rest of the pie. See 2-18 for an example.

You can specify the segment to be exploded with an expression using relational and logical operators listed in Table 7-4.

Table 7-4. Explosion Expression Operators

Relational Operators	=	Equals
	<>	Not equal
	>	Greater than
	>=	Greater than or equal to
	<	Less than
	<=	Less than or equal to
Logical Operators	not NOT	Condition not true
	and AND	Both conditions true
	or OR	At least one condition true

The relational operations are performed in the order in which they appear from left to right, before logical operators. Logical operations are performed in the following order:

1. NOT
2. AND
   OR

BRACKET FOR AND/OR

This means that NOT operations are performed first: and AND and OR operations are performed from left to right. This order may be overridden by expressions in parentheses whose operations are performed before all others.

Examples of Explosion Expressions:

```
DIVISION="GSD" AND YEAR<1975
SEX="M" OR SALARY<2000
FEMPLOYEES>MEMPLOYEES
```

Explosion expressions may include variables from the data file other than the independent variable and label variable.

GFONT $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Allows mapping of available fonts to the four fonts which are active at any given time. Must be called once for each font selected.

FORMAT

```
                 IA      I          BA       I       I      I
      GFONTSET (graf,fontnumber,fontfile,language,slant,charform)
                i/o      in         in       in      in     in


                IA       I          BA       I       I      I
      GFONTINQ (graf,fontnumber,fontfile,language,slant,charform)
                i/o      in         out      out     out    out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

fontnumber (1-4) Integer variable representing the font selected.

Parameters (continued)

fontfile 36-byte array holding the name of the font file to be used. The default fonts presently available in the font files are:

font	font description	file name
1	Stick	STICK.VCHARSET.SYS
2	Script	SCRIPT.VCHARSET.SYS
3	Triplex Roman	ROMAN.VCHARSET.SYS
4	Gothic English	GOTHIC.VCHARSET.SYS

language Integer variable giving the number of the language represented by this font. The default language will be one of the available languages and will be determined by the message catalog, usually US ASCII.

0	Default Language
1	US ASCII
2	Swedish/Finnish
3	Norwegian/Danish
4	French
5	German
6	United Kingdom
7	Spanish

In Europe, the message catalog may be changed to make the default one of the other available languages.

Parameters (continued)

slant	Integer variable for slant of characters (italics) in this font. (Slant values = -90 to +90 degrees.) 0 No slant
charform	Integer variable indicating how characters were stored--7 bits or 8 bits. 0 7-bit characters (ASCII) 1 8-bit characters Only 7-bit character value (0) is presently supported.

DISCUSSION

If GFONTSET is not called, the default mapping is as follows: the default language, a 0 slant, and 7-bit characters are used.

Font 1	Stick
Font 2	Script
Font 3	Triplex Roman
Font 4	Gothic English

Additional fonts will be added to the font files at a later time.

If a font is not available, DSG/3000 finds a font that best matches the one specified. Usually it is the Stick font. If a language is not available, the default language in the file is used.

GFONTSET sets the fonts for the current chart only. It will need to be called once for each of the font numbers you wish to have activated, if the default fonts are not used.

GGETCHART

Accesses a particular chart, making it the current chart. Callable from all languages.

FORMAT

```
                 IA      BA
     GGETCHART (graf,chartname)
                 i/o     in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

chartname 16-byte array holding name of chart.

DISCUSSION

In order to access a chart that has been previously created, call GGETCHART after opening the chart file. The chart you name in this procedure becomes the current chart, meaning that subsequent calls to chart, data and graphing procedures refer to this chart.

If you want to create a new chart, call GCREATECHART instead of GGETCHART.

DATA DECLARATIONS - GGETCHART

```
COBOL:        DATA DIVISION.
              01  GRAF.
                  05  GSTATUS      PIC S9(4) COMP.
                  05  PARAMETER    PIC S9(4) COMP.
                  05  ELEMENT      PIC S9(4) COMP.
                  05  FILE-ERR     PIC S9(4) COMP.
                  05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
              01  CHART-NAME       PIC X(16).
                .
                .
                .
              CALL "GGETCHART" USING GRAF, CHART-NAME.

BASIC:        10  REM   G IS GRAF ARRAY   C$ IS CHART NAME
              15  REM   STATUS IS G[1]
              20  INTEGER G[3700]
              30  DIM C$[16]
                .
                .
                .
              50  CALL GGETCHART (G[*],C$)

FORTRAN:      INTEGER GRAF(1600)
              INTEGER STATUS
              CHARACTER*16 CHARTNAME
       C
              EQUIVALENCE (STATUS,GRAF(1))
                .
                .
                .
              CALL GGETCHART (GRAF, CHARTNAME)

SPL:          INTEGER ARRAY GRAF(0:1599);
              DEFINE
                STATUS  = GRAF (0)#;
              BYTE ARRAY CHART'NAME(0:15);
                .
                .
                .
              GGETCHART (GRAF, CHART'NAME);
```

PASCAL:

```
TYPE
  SMALL_INT=-32768..32767;
  GRAFTYPE =record
         status      :small_int;
         parameter   :small_int;
         element     :small_int;
         file_err    :small_int;
         graffarea   :array[1..1596] of small_int;
       end;
VAR
  GRAF       :GRAFTYPE;
  CHARTNAME  :PACKED ARRAY [1..16] OF CHAR;
.
.
.
GGETCHART(GRAF,CHARTNAME);
```

GINDVAR {SET | INQ}

Sets or returns the X-axis (independent) variable for the current chart. Required for all charts but not for slides. Callable from all languages.

FORMAT

```
                IA     BA
      GINDVARSET (graf,varname)
                i/o    in

                IA     BA
      GINDVARINQ (graf,varname)
                i/o    out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

varname Byte array holding the 16-byte variable name identifying the X-axis (independent) variable. This may not be a textual variable if a pie chart or line chart is being defined. (The data is then ignored.)

DATA DECLARATIONS - GINDVARSET, GINDVARINQ

```
COBOL:        DATA DIVISION.
              01  GRAF.
                  05  GSTATUS     PIC S9(4) COMP.
                  05  PARAMETER   PIC S9(4) COMP.
                  05  ELEMENT     PIC S9(4) COMP.
                  05  FILE-ERR    PIC S9(4) COMP.
                  05  FILLER      PIC S9(4) COMP OCCURS 1596 TIMES.
              01  VAR-NAME        PIC X(16).
                .
                .
                .
              CALL "GINDVARSET" USING GRAF, VAR-NAME.

BASIC:        10  REM   G IS GRAF ARRAY    V$ IS VAR-NAME
              15  REM   STATUS IS G[1]
              20  INTEGER G[3700]
              30  DIM V$[16]
                .
                .
                .
              50  CALL GINDVARSET (G[*],V$)

FORTRAN:      INTEGER GRAF(1600)

              INTEGER STATUS
              CHARACTER*16 VARNAME
       C
              EQUIVALENCE (STATUS,GRAF(1))
                .
                .
                .
              CALL GINDVARSET (GRAF, VARNAME)

SPL:          INTEGER ARRAY GRAF(0:1599);
              DEFINE
                STATUS  = GRAF (0)#;
              BYTE ARRAY VAR'NAME(0:15);
                .
                .
                .
              GINDVARSET (GRAF, VAR'NAME);
```

```
PASCAL:       PROGRAM DOC;
              TYPE
                SMALL_INT=-32768..32767;
                GRAFTYPE =record
                      status      :small_int;
                      parameter   :small_int;
                      element     :small_int;
                      file_err    :small_int;
                      graf_area   :array[1..1596] of small_int;
                    end;
              VAR
                GRAF      :GRAFTYPE;
                VARNAME :PACKED ARRAY [1...16] OF CHAR;
              .
              .
              .
              GINDVARSET(GRAF,VARNAME);
```

GINITGRAF

Initializes communications area. Callable from all languages.

FORMAT

```
                  IA     I        I
      GINITGRAF (graf,grafsize,language)
                  i/o    in       in

      CALL "GINITGRAF" USING graf, grafsize, language.
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

grafsize Integer variable giving length of *graf* in words. Must be at least 1600 words for non-Basic programs or 3700 words for Basic programs.

language Integer code indicating program language:

0 COBOL
1 BASIC
2 FORTRAN
3 SPL
4 (Reserved)
5 PASCAL

DISCUSSION

GINITGRAF should be called whenever you enter DSG/3000. Error status is cleared and MPE operating system file numbers are set to 0.

See page 7-3, Communications Area.

DATA DECLARATIONS - GINITGRAF

```
COBOL:        DATA DIVISION.
              77  GRAFSIZE      PIC S9(4) COMP VALUE 1600.
              77  LANGUAGE      PIC S9(4) COMP VALUE 0.
              01  GRAF.
                  05  GSTATUS   PIC S9(4) COMP.
                  05  PARAMETER PIC S9(4) COMP.
                  05  ELEMENT   PIC S9(4) COMP.
                  05  FILE-ERR  PIC S9(4) COMP.
                  05  FILLER    PIC S9(4) COMP OCCURS 1596 TIMES.
                .
                .
                .
              CALL "GINITGRAF" USING GRAF, GRAFSIZE, LANGUAGE.

BASIC:        10  REM   G IS GRAF ARRAY    S IS GRAFSIZE
              20  REM   L IS LANGUAGE
              30  INTEGER G[3700], S, L
              40  S=3500
              50  L=1
                .
                .
                .
             100  CALL GINITGRAF (G[*],S,L)

FORTRAN:      INTEGER GRAF(1600),GRAFSIZE
              INTEGER STATUS, PARAMETER, ELEMENT, FILERR
        C
        C  NOTE THAT "LANGUAGE" IS AN IMPLICIT INTEGER
        C
              EQUIVALENCE (STATUS,GRAF(1)), (PARAMETER,GRAF(2))
              EQUIVALENCE (ELEMENT,GRAF(3)), (FILERR,GRAF(4))
        C
              DATA GRAFSIZE/1600/, LANGUAGE/2/

                .
                .
                .
              CALL GINITGRAF (GRAF, GRAFSIZE, LANGUAGE)

SPL:          INTEGER ARRAY GRAF(0:1599);
              DEFINE
                STATUS    = GRAF (0)#
                PARAM     = GRAF (1)#
                ELEMENT   = GRAF (2)#
                FILE'ERR  = GRAF (3)#;
              INTEGER GRAFSIZE:=1600,
                      LANGUAGE:=3;
                .
                .
                .
              GINITGRAF (GRAF, GRAFSIZE, LANGUAGE);
```

PASCAL:

```
TYPE
  SMALL_INT=-32768..32767;
  GRAFTYPE =record
          status      :small_int;
          parameter   :small_int;
          element     :small_int;
          file_err    :small_int;
          graffarea   :array[1..1596] of small_int;
      END;(* RECORD *)
VAR
  GRAF        :GRAFTYPE;
  GRAFSIZE,
  LANGUAGE    :SMALL_INT;
.
.
.
GINITGRAF(GRAF,GRAFSIZE,LANGUAGE);
```

GLABEL {SET / INQ}

Sets or returns labels for major tick marks in bar and line graphs or segments in pie graphs. Provides for dynamic labeling. Callable from all languages.

FORMAT

```
                  IA   I    I     BA       I        I
      GLABELSET (graf,axis,type,labelbuf,labelen,numlabels,
                 i/o   in   in     in       in      in

                      I           IA
                 labelspacing,useroptions)
                     in           in


                  IA    I    I     BA        I           I
      GLABELINQ (graf,axis,type,labelbuf,labelbuflen,labelen,
                  i/o  in   out    out      in         out

                     I           I           IA
                 numlabels,labelspacing,useroptions)
                   out         out          out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

axis Integer code indicating what is to be labeled:

1 X axis
2 Y axis
3 Pie segments

type Integer code indicating where labels are to be found:

0 no labels

1 data variable contains labels (pies and bars only); not valid for dependent axis

2 calculate labels from axis scaling (bars and lines only)

3 dynamic labeling (described by the three following parameters)

labelbuf Byte array holding list of dynamically supplied labels. A single label can be at most 25 characters. Maximum buffer size is 650 characters.

labelbuflen Integer variable giving length in bytes of the buffer to receive the labels.

labelen Integer variable specifying length in bytes of one label. Must be a positive number, no greater than 25. (Ignored if labels are not dynamically supplied.)

numlabels Integer variable specifying number of labels in the label buffer. (Ignored if labels are not dynamically supplied.) Must be positive.

labelspacing Integer variable specifying number of major ticks per label:

1 label every major tick
2 label every other major tick
n label every *n*th major tick

Parameters (continued)

useroptions Integer array 2 words in length used as comments field for interactive interface. Ignored by the intrinsics, but used by the interactive program GRAPH as follows:

Byte 1	N	numeric (calculate labels)
	D	data has labels
	W	weekdays
	M	months
	Q	quarters
	Y	years
Byte 2	A	abbreviated
	F	full
Bytes 3 & 4	Any integer start value	

DISCUSSION

Bar chart: page 6-12
Line chart: page 6-19
Pie chart: page 6-23

DATA DECLARATIONS - GLABELSET, GLABELINQ

```
COBOL:          DATA DIVISION.
                01  GRAF.
                    05  GSTATUS        PIC S9(4) COMP.
                    05  PARAMETER      PIC S9(4) COMP.
                    05  ELEMENT        PIC S9(4) COMP.
                    05  FILE-ERR       PIC S9(4) COMP.
                    05  FILLER         PIC S9(4) COMP OCCURS 1596 TIMES.
                01  X-AXIS-LABELS.
                    05  AXIS           PIC S9(4) COMP VALUE 1.
                    05  LTYPE          PIC S9(4) COMP VALUE 3.
                    05  LABEL-BUF.
                        10  LABEL1     PIC X(8) VALUE "WEEK 1".
                        10  LABEL2     PIC X(8) VALUE "WEEK 2".
                        10  LABEL3     PIC X(8) VALUE "WEEK 3".
                        10  LABEL4     PIC X(8) VALUE "WEEK 4".
                    05  LABELEN        PIC S9(4) COMP VALUE 8.
                    05  NUM-LABELS     PIC S9(4) COMP VALUE 4.
                    05  LABEL-SPACING PIC S9(4) COMP VALUE 1.
                    05  USER-OPTIONS   PIC S9(4) COMP OCCURS 2 TIMES.
                     .
                     .
                     .
                CALL "GLABELSET" USING GRAF, AXIS, LTYPE, LABEL-BUF,
                                       LABELEN, NUM-LABELS,
                                       LABEL-SPACING, USER-OPTIONS.

BASIC:           10  REM  G IS GRAF ARRAY    A IS AXIS
                 20  REM  T IS TYPE          L$ IS LABEL-BUF
                 30  REM  L1 IS LABELEN      N IS NUM-LABELS
                 40  REM  L2 IS LABEL-SPACING  U IS USER-OPTIONS
                 50  REM
                 60  INTEGER G[3700],A,T,L1,N,L2,U[2]
                 70  DIM L$[32]
                 80  A=1
                 90  T=3
                100  L$(1)="WEEK 1  "
                110  L$(9)="WEEK 2  "
                120  L$(17)="WEEK 3  "
                130  L$(25)="WEEK 4  "
                140  L1=8
                150  N=4
                160  L2=1
                170  MAT U=ZER
                      .
                      .
                      .
                200  CALL GLABELSET(G[*],A,T,L$,L1,N,L2,U[*])
```

FORTRAN:

```
      INTEGER GRAF(1600), AXIS, TYPE
      CHARACTER*8 LABELBUF(4)
      INTEGER LABELEN, NUMLABELS, LABELSPACING
      INTEGER USEROPTIONS(2)
      DATA USEROPTIONS/2*0/
         .
         .
         .
      AXIS=1
      TYPE=3
      LABELBUF(1)="WEEK 1"
      LABELBUF(2)="WEEK 2"
      LABELBUF(3)="WEEK 3"
      LABELBUF(4)="WEEK 4"
      LABELEN=8
      NUMLABELS=4
      LABELSPACING=1
      CALL GLABELSET (GRAF, AXIS, TYPE, LABELBUF,
     *                LABELEN, NUMLABELS, LABELSPACING,
     *                USEROPTIONS)
```

SPL:

```
INTEGER ARRAY GRAF (0:1599);

INTEGER AXIS:=1,
        TYPE:=3,
        LABELEN:=8,
        NUM'LABELS:=4,
        LABEL'SPACING:=1;
BYTE ARRAY LABEL'BUF(0:31):=32(" ");
INTEGER ARRAY USER'OPTIONS(0:1):=2(0);
   .
   .
   .
MOVE LABEL'BUF(0):="WEEK 1";
MOVE LABEL'BUF(8):="WEEK 2";
MOVE LABEL'BUF(16):="WEEK 3";
MOVE LABEL'BUF(24):="WEEK 4";
GLABELSET (GRAF, AXIS, TYPE, LABEL'BUF,
           LABELEN, NUM'LABELS, LABEL'SPACING,
           USER'OPTIONS);
```

```
PASCAL        CONST
              X_AXIS =1;
              Y_AXIS =2;
              PIE_SEG=3;

              NO_LABEL         =0;
              DATA_VAR_CONTAINS=1;
              CALCULATE        =2;
              DYNAMIC          =3;

              TYPE
                SMALL_INT=-32768..32767;
                USER_OPTION_REC=RECORD
                       optn_array       :packed array[1..2] of char;
                       start_value      :small_int;
                     END;(* USER_OPTION_REC*)
                GRAFTYPE =record
                       status,
                       parameter,
                       element,
                       file_err,   :small_int
                       graffarea   :array[1..1596] of small_int;
                     END;(* GRAFTYPE *)
                LABEL_TYPE     =PACKED ARRAY[1..8] OF CHAR;

              VAR
                AXIS,
                TIPE,
                LABEL_LEN,
                NUM_LABELS,
                LABEL_SPACING     :SMALL_INT ;
                GRAF              :GRAFTYPE;
                LABEL_BUF         :PACKED ARRAY OF [1..4] OF LABEL_TYPE
                USER_OPTIONS      :USER_OPTION_REC;
              .
              .
              .
                GLABELSET(GRAF,AXIS,TIPE,LABEL_BUF,LABEL_LEN,
                         NUM_LABELS, LABEL_SPACING,USER_OPTIONS);
```

GLABVAR $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Sets or returns the label variable. Used by pie and bar charts. Callable from all languages.

FORMAT

```
                  IA    BA
      GLABVARSET (graf,varname)
                 i/o    in


                  IA    BA
      GLABVARINQ (graf,varname)
                 i/o    out
```

PARAMETERS

graf	Integer array containing global information used by DSG/3000. The first word returns the call status.
varname	Byte array holding 16-byte data variable name identifying the label variable.

DISCUSSION

Pie charts: page 6-23.
Bar charts: page 6-12.

GLEGEND $\left\{\begin{matrix}\text{SET}\\\text{INQ}\end{matrix}\right\}$

Sets or returns color, texture, and legend labels for a dependent variable in a line or bar chart. Callable from all languages.

FORMAT

```
                   IA     I        I     I     I       BA
      GLEGENDSET (graf,depvarid,color,texture,marker,legend1
                   i/o    in      in    in     in      in

                     BA       I
                  legend2,useroptions)
                    in        in

                   IA     I        I      I      I      BA
      GLEGENDINQ (graf,depvarid,color,texture,marker,legend1
                   i/o    in      out     out   out     out

                    BA         I
                  legend2,useroptions)
                    out       out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

depvarid Integer variable specifying the relative number of the Y-axis variable being described. The first y axis (dependent) variable named in GDEPVARSET is numbered one.

Parameters (continued)

color Integer code corresponding to the color assigned to the variable (1-16):

1 color of pen 1, usually black
2 color of pen 2, usually red
3 color of pen 3, usually green
4 color of pen 4, usually blue
5...16 if additional pens are used.

This parameter is ignored if the plotting device is a non-color device. The default, specified as 0 or -1, is black. Appendix A contains information about colors.

texture Integer code corresponding to the texture assigned to the variable. For scattergrams, equate *texture* to 0. The default, specified as -1, assigns a different texture to each variable. See Appendix A.

marker Integer code corresponding to the marker type assigned to the variable. Each data point on the line will be marked with this symbol. Ignored for bar graphs.

Scattergrams must be given a non-zero value. The default, specified by 0 or -1, is no markers. See Appendix A.

legend1 Byte array holding the first line of the legend text; 18 characters in length.

Parameters (continued)

legend2 Byte array holding the second line of the legend text; 18 characters in length.

useroptions Integer variable used as comments field. Ignored by procedure.

DISCUSSION

Each Y-axis (dependent) variable is associated with a line on a line chart or a set of bars on a bar chart. Each of these lines or bar sets is numbered (*depvarid*) in the order in which they were specified in GDEPVARSET.

When you plot more than one dependent variable, DSG/3000 gives each one a different texture by default, as shown in Appendix A. Call GLEGENDSET to change the texture or add colors and data markers. Note that GLEGENDSET is called once for each variable.

Legends are drawn under the title of the graph and are used to associate a variable with a particular color and texture. The text of each legend can take up to two lines (*legend1* and *legend2*), each containing 18 characters. If only one line of legend is used, leave line 1 blank and enter the text on line 2 for the best visual effect.

Scattergrams are a special type of line graph having markers but no connecting lines (*texture*=0).

DATA DECLARATIONS - GLEGENDSET

```
COBOL:         DATA DIVISION.
               01  GRAF.
                   05  GSTATUS        PIC S9(4) COMP.
                   05  PARAMETER      PIC S9(4) COMP.
                   05  ELEMENT        PIC S9(4) COMP.
                   05  FILE-ERR       PIC S9(4) COMP.
                   05  FILLER         PIC S9(4) COMP OCCURS 1596 TIMES.
               01  VARIABLE-ONE.
                   05  DEP-VAR-ID     PIC S9(4) COMP VALUE 1.
                   05  COLOR          PIC S9(4) COMP VALUE 2.
                   05  TEXTURE        PIC S9(4) COMP VALUE 6.
                   05  MARKER         PIC S9(4) COMP VALUE 0.
                   05  LEGEND1        PIC X(18) VALUE "LINE 1 TEXT".
                   05  LEGEND2        PIC X(18) VALUE "LINE 2 TEXT".
                   05  USER-OPTIONS   PIC S9(4) COMP VALUE 0.
                  .
                  .
                  .
               CALL "GLEGENDSET" USING GRAF, DEP-VAR-ID, COLOR,
                                       TEXTURE, MARKER, LEGEND1,
                                       LEGEND2, USER-OPTIONS.

BASIC:          10  REM   G IS GRAF ARRAY       D IS DEP-VAR-ID
                20  REM   C IS COLOR            T IS TEXTURE
                30  REM   M IS MARKER           L1$ IS LEGEND1
                40  REM   L2$ IS LEGEND2        U IS USER-OPTIONS
                50  REM
                60  INTEGER G[3700],D,C,T,M,U
                70  DIM L1$[18],L2$[18]
                80  D=1
                90  C=2
               100  T=6
               110  M=0
               120  L1$="LINE 1 TEXT"
               130  L2$="LINE 2 TEXT"
               140  U=0
                    .
                    .
                    .
               200  CALL GLEGENDSET (G[*],D,C,T,M,L1$,L2$,U)
```

```
FORTRAN:       INTEGER GRAF(1600), DEPVARID, COLOR
               INTEGER TEXTURE, MARKER, USEROPTIONS
               CHARACTER*18 LEGEND1, LEGEND2
                   .
                   .
               DEPVARID=1
               COLOR=2
               TEXTURE=6
               MARKER=0
               LEGEND1="LINE 1 TEXT"
               LEGEND2="LINE 2 TEXT"
               USEROPTIONS=0
                   .
                   .
               CALL GLEGENDSET (GRAF, DEPVARID, COLOR, TEXTURE,
              *                 MARKER, LEGEND1, LEGEND2,
              *                 USEROPTIONS)

SPL:           INTEGER ARRAY GRAF(0:1599);
               INTEGER  DEP'VAR'ID:=1,
                        COLOR:=2,
                        TEXTURE:=6,
                        MARKER:=0,
                        USER'OPTIONS:=0;
               BYTE ARRAY LEGEND1(0:17):=18(" ");
               BYTE ARRAY LEGEND2(0:17):=18(" ");
                   .
                   .
               MOVE LEGEND1:="LINE 1 TEXT";
               MOVE LEGEND2:="LINE 2 TEXT";
                   .
                   .
               GLEGENDSET (GRAF, DEP'VAR'ID, COLOR, TEXTURE,
                           MARKER, LEGEND1, LEGEND2, USER'OPTIONS);
```

PASCAL

```
CONST
  BLANK             =1;
  DOTTED            =2;
  DASHED            =3;
  LINED             =4;
  CROSSHATCHED      =5;
  TIGHT_LINED       =6;
  TIGHT_CROSSHATCHED=7;
  SOLID             =8;

  NONE           =0;
  ASTRIC         =1;
  CROSS          =2;
  TRIANGLE       =3;
  SQUARE         =4;
  DIAMOND        =5;
  CIRCLE         =6;
  CROSS_HAIR     =7;
  X              =8;


TYPE
  SMALL_INT=-32768..32767;
  GRAFTYPE =record
         status,
         parameter,
         element,
         file_err,
         graffarea,
  END;(* GRAFTYPE *)

  PEN_NUM_TYPE  =1..8;

VAR
  LEGEND1,
  LEGEND2          :PACKED ARRAY [1...18] OF CHAR
  DEPVARID,
  USER_OPT,
  MARKER,
  TEXTURE          :SMALL_INT ;
  GRAF             :GRAFTYPE;
  PEN_NUM          :PEN_NUM_TYPE;

.
.
.

  GLEGENDSET(GRAF,DEPVARID,PEN_NUM,TEXTURE,MARKER,
     LEGEND1,LEGEND2,USER_OPT);
```

GLIMIT [A] $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Sets or returns the position of a chart on the plotting device. The ASCII version, callable from COBOL as well as other languages, is designated by an "A" in the procedure name.

FORMAT

```
            IA    I    R  R  R  R
   GLIMITSET (graf,limits,x1,y1,x2,y2)
            i/o   in   in in in in

            IA    I    R  R  R  R
   GLIMITINQ (graf,limits,x1,y1,x2,y2)
            i/o   out outoutoutout

   CALL "GLIMITASET" USING graf,limits,x1,y1,x2,y2

   CALL "GLIMITAINQ" USING graf,limits,x1,y1,x2,y2
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

limits Integer flag indicating whether the following coordinates refer to inner or outer limits:

0 default for the device. Usually approximates 8 1/2 inch by 11 inch format (D-size paper on 7580 and 7585)

1 chart boundary (outer limits)

Parameters (continued)

2 data boundary (inner limits)

3 English standard (8 1/2 by 11 inch format)

4 Metric standard (210 mm by 298 mm)

Option 2 not applicable for pie charts or slides.

x1 ASCII version: 16-byte array
REAL version: real variable

X-coordinate (horizontal position) of lower left corner.

y1 ASCII version: 16-byte array
REAL version: real variable

Y-coordinate (vertical position) of lower left corner.

x2 ASCII version: 16-byte array
REAL version: real variable

X-coordinate (horizontal position) of upper right corner.

y2 ASCII version: 16-byte array
REAL version: real variable

Y-coordinate (vertical position) of upper right corner.

DISCUSSION

See page 6-7/8.

Note: If you accept the default or the English or Metric standard for this procedure, the graph is always drawn to fit the format of the plotting device. If you specify limits and later change plotting devices, you may also need to change the limit specifications so that you will not lose a part of the graph.

To use non-standard paper with a plotter, position the paper a few millimeters to the right of the lower left plotter bed. This applies to the use of HP transparencies and paper shorter than 11 inches.

GLISTCHARTS

Returns a list of chart names from the chart file directory. Callable from all languages.

FORMAT

```
                 IA      BA        I         I          I
     GLISCHARTS (graf,chartnames,maxcharts,firstchart,numcharts)
                 i/o     out       in        in         out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

chartnames Byte array returning 16-byte chart names from the current chart file.

maxcharts Integer variable specifying the maximum number of charts to be returned.

firstchart Integer variable specifying the first chart name to be returned. Charts are numbered in the order created, starting with 1.

numcharts Integer variable specifying the number of charts returned.

DISCUSSSION

GLISTCHARTS can return the names of some or all of the charts in the chart file, functioning like a table of contents.

GNUMCHARTS

Returns the total number of charts in chart file. Callable from all languages.

FORMAT

```
                 IA       I
      GNUMCHARTS (graf,numcharts)
                 i/o     out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

numcharts Integer variable specifying number of charts in current chart file.

GOPENFILE

Opens an existing chart file. Callable from all languages.

FORMAT

```
               IA      BA       I
     GOPENFILE (graf,filename,access)
               i/o     in      in
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

filename 36-byte array containing name of file being opened. Can be fully qualified. Must be terminated by a space (blank character).

access Integer flag indicating whether to open file with exclusive or shared access:

1 exclusive access
2 shared access

If you supply a 0, the default results in exclusive access.

DISCUSSION

A call to GOPENFILE establishes the named file as the current chart file. Only charts contained in this file can be subsequently accessed. This procedure places the MPE file number in GRAF, the communications buffer, and closes any previously opened chart file.

Any writing or creating activity must access the file exclusively. Inquiring, executing, opening, and getting charts are sharable activities.

DATA DECLARATIONS - GOPENFILE

```
COBOL:        DATA DIVISION.
              01  GRAF
                  05  GSTATUS      PIC S9(4) COMP.
                  05  PARAMETER    PIC S9(4) COMP.
                  05  ELEMENT      PIC S9(4) COMP.
                  05  FILE-ERR     PIC S9(4) COMP.
                  05  FILLER       PIC S9(4) COMP OCCURS 1596 TIMES.
              01  CHART-FILE-INFO.
                  05  FILE-NAME    PIC X(36) VALUE "OLDCHART ".
                  05  GACCESS      PIC S9(4) COMP VALUE 1.
                .
                .
                .
              CALL "GOPENFILE" USING GRAF, FILE-NAME, GACCESS.

BASIC:        10  REM   G IS GRAF ARRAY   F$ IS CHART FILE NAME
              20  REM   A IS FILE ACCESS
              25  REM   STATUS IS G[1]
              30  INTEGER G[3700], A
              40  DIM F$[36]
              50  F$="OLDCHART "
                .
                .
                .
              80  CALL GOPENFILE (G[*],F$,A)
```

```
FORTRAN:        INTEGER GRAF(1600), STATUS, ACCESS
                CHARACTER*36 FILENAME
        C
                EQUIVALENCE (STATUS,GRAF(1))
                FILENAME="OLDCHART "
                   .
                   .
                   .
                CALL GOPENFILE (GRAF, FILENAME, ACCESS)

SPL:            INTEGER ARRAY GRAF(0:1599);
                DEFINE
                  STATUS   = GRAF (0)#;
                BYTE ARRAY FILE'NAME(0:35):="OLDCHART ";
                INTEGER ACCESS:=1;
                   .
                   .
                   .
                GOPENFILE (GRAF, FILE'NAME, ACCESS);


PASCAL:         TYPE
                  SMALL_INT=-32768..32767;
                  GRAFTYPE =record
                         status     :small_int;
                         parameter  :small_int;
                         element    :small_int;
                         file_err   :small_int;
                         graffarea  :array[1..1596] of small_int;
                      end;
                VAR
                  GRAF     :GRAFTYPE;
                  FILENAME :PACKED ARRAY [1...36] OF CHAR;
                  ACCESS   :SMALL_INT;
                .
                .
                .
                GOPENFILE(GRAF,FILENAME,ACCESS);
```

GPIE $\begin{Bmatrix}\text{SET}\\\text{INQ}\end{Bmatrix}$

Sets or returns unique pie chart characteristics. Callable from all languages.

FORMAT

```
                IA     I      I      IA       I         IA
      GPIESET (graf,percent,actual,colorbuf,colorlen,texturebuf,
                i/o    in     in     in       in        in

                   I
               texturelen)
                   in

                IA    I      I       IA      I             I
      GPIEINQ (graf,percent,actual,colorbuf,colorbuflen,colorlen,
                i/o   out    out     out     in          out

                  IA          I             I
               texturebuf,texturebuflen,texturelen)
                  out         in           out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

percent Integer flag indicating whether to label each segment with percentages:

0 do not label percentages (default)
1 label percentages

Parameters (continued)

actual Integer flag indicating whether to label each segment with its actual data value:

0 do not label (default)
1 label with actual data value

colorbuf Integer array holding colors of segments by relative position. The first segment begins in the two o'clock position, and the buffer is circular. Maximum number of colors is 30. Valid colors are:

1 color of pen 1, usually black
2 color of pen 2, usually red
3 color of pen 3, usually green
4 color of pen 4, usually blue
5...16 if additional pens are used.

This parameter is ignored if the plotting device is a non-color device. The default color, specified as 0 or -1, is usually black. See Appendix A for additional color information.

colorbuflen Integer variable specifying the length in words of the colors buffer.

texturebuf Integer array holding textures of segments by relative position. The first segment is drawn beginning in the two o'clock position, and the buffer is circular. Maximum number of textures is 30. See appendix A for textures.

texturebuflen Integer variable specifying length in words of textures buffer.

Parameters (continued)

colorlen Integer variable specifying number of colors in *colorbuf*.

texturelen Integer variable specifying number of textures in *texturebuf*.

DISCUSSION

If you want the segments to be drawn clockwise from smallest to largest, you must call GSORTSET. Otherwise, the segments are drawn in the order they appear in the data file.

Since the colors and textures buffers are circular, if you specify fewer colors or textures than there are segments, the assignment of colors and textures will be repeated. For example, if for 8 segments you specify the colors red, green, and blue, the segments will be drawn in the following colors:

Segment	Color
1	red
2	green
3	blue
4	red
5	green
6	blue
7	red
8	green

See page 6-26 for a discussion of segment labeling.

GREADANOT[A]

Used for positioning annotations. Reads graphics cursor and returns its coordinates in relative units. The ASCII version, callable from COBOL as well as other languages, is designated by an "A" in the procedure name.

FORMAT

```
                 IA    R  R
     GREADANOT (graf, x, y)
                i/o out out

     CALL "GREADANOTA" USING graf, x, y
```

PARAMETERS

graf	Integer array containing global information used by DSG/3000. The first word returns the call status.
x	ASCII version: 16-byte array REAL version: real variable X-coordinate (horizontal position) of graphics cursor.
y	ASCII version: 16-byte array REAL version: real variable Y-coordinate (vertical position) of graphics cursor.

DISCUSSION

A call to GREADANOT[A] digitizes a point after the user positions the graphics cursor and presses any character key.

See pages 6-9 for a discussion of annotations.

GSORT $\left\{\begin{matrix}\text{SET}\\ \text{INQ}\end{matrix}\right\}$

Sets or returns a flag to sort the independent variable. Callable from all languages.

FORMAT

```
                IA      I
      GSORTSET (graf,sortflag)
               i/o     in

                IA      I
      GSORTINQ (graf,sortflag)
               i/o    out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

sortflag Integer flag indicating whether to sort the X-axis (independent) variable:

0 no sort (default)
1 sort

DISCUSSION

This procedure is applicable to all graph types. The default is that the data is plotted in the order it appears in the data file. If you set the sort flag, the data is drawn in increasing numeric order or, if the X-axis variable is textual, in alphabetic order. In the case of bar graphs, this affects the order of the bars in the completed graph.

When you are designing a pie chart, the segments are drawn in the order of the data in the data file. Call GSORTSET if you want the segments to be drawn clockwise from smallest to biggest starting at the two o'clock position.

GSUB $\left\{ \begin{matrix} \text{SET} \\ \text{INQ} \end{matrix} \right\}$

Sets or returns data file subset specifications. Callable from all languages.

FORMAT

```
                IA      BA         I
      GSUBSET (graf,subsetbuf,subsetlen)
               i/o      in         in

               IA       BA            I            I
      GSUBINQ (graf,subsetbuf,subsetbuflen,subsetlen)
               i/o      out          in          out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

subsetbuf Byte array holding subset specifications. (See Table 3-2.)

subsetlen Integer variable specifying length in bytes of subset specifications.

subsetbuflen Integer variable specifying maximum numbers of bytes to be returned in the subset buffer. A warning is returned if the buffer is too small to receive the subset.

DISCUSSION

Subsets specify which items from the data file will be plotted. Subsets are described using relational and logical operators. See page 3-15.

The parameter *subsetlen* may be retrieved before calling GSUBINQ by calling GDATAINFO.

GSUPPRESS $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Sets or returns the graph components that will not be drawn. Callable from all languages.

FORMAT

```
                     IA    I      I       I    I     I      I
      GSUPPRESSSET (graf,chart,legends,labels,axes,title,frame,
                    i/o    in    in      in    in    in    in

                     I     I
                    data,advance)
                     in     in

                     IA    I      I       I     I    I      I
      GSUPPRESSINQ (graf,chart,legends,labels,axes,title,frame,
                    i/o   out    out    out    out   out  out

                     I     I
                   data,advance)
                     out   out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

Parameters (continued)

chart Integer flag indicating whether the chart attributes should be suppressed:

0 draw chart (default)
1 suppress chart

Chart attributes include legends, labels, axes, titles, frame, data, and advance.

Not included are data, represented by bars, lines or pie segments.

legends Integer flag indicating whether the legends should be suppressed:

0 draw legends (default)
1 suppress legends

labels Integer flag indicating whether the labels should be suppressed:

0 draw labels (default)
1 suppress labels

axes Integer flag indicating whether the axes should be suppressed:

0 draw axes (default)
1 suppress axes

title Integer flag indicating whether the titles should be suppressed:

0 draw titles (default)
1 suppress titles

Parameters (continued)

frame Integer flag indicating whether the frame should be suppressed:

0 draw frame
1 suppress frame (default)

data Integer flag indicating whether the data should be suppressed:

0 draw data (default)
1 suppress data

Data includes bars, lines and pie segments.

advance Integer flag indicating whether the paper advance should be suppressed:

0 advance paper (default) before drawing chart
1 do not advance

DISCUSSION

Call GSUPPRESSSET if you want to overlay charts or parts of charts. The advance parameter affects the graphics terminals (2623A,2647A,2647F, 2648A and 2703) as w as the scrolling plotters (7221S/T,9872S/T and 7220S/T). If you suppress the advance, the video display will not clear, and the current graph will overlay the previous chart.

GSYMBOL[A] $\left\{\begin{array}{l}\textbf{SET}\\\textbf{INQ}\end{array}\right\}$

Sets or returns box, line, or arrow annotations. The ASCII version, callable from COBOL as well as other languages, is designated by an "A" in the procedure name.

FORMAT

```
            IA     I       I    R    R  R   R
GSYMBOLSET (graf,annotid,symbol,x1, y1, x2, y2
            i/o    in      in   in  in  in  in

              I       I
            color,linetype)
              in      in

             IA     I      I    R   R   R   R
GSYMBOLINQ (graf,annotid,symbol,x1, y1, x2, y2
            i/o    in     out  out out out out

              I       I
            color,linetype)
             out     out

CALL "GSYMBOLASET" USING graf anotnum symbol x1 y1 x2 y2
            color linetype

CALL "GSYMBOLAINQ" USING graf anotnum symbol x1 y1 x2 y2
            color linetype
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

annotid Integer variable specifying the unique annotation identifying number. The first annotation is numbered one.

symbol Integer code specifying the type of annotation symbol:

1 box
2 arrow
3 line

x1 ASCII version: 16-byte array
REAL version: real variable

x-coordinate for one end of the line, the head of the arrow, or one corner of the box. The value of the x-coordinate can range from 0 to 100.

y1 ASCII version: 16-byte array
REAL version: real variable

y-coordinate for the same point described by *x1*. The value of the y-coordinate can range from 0 to 100.

x2 ASCII version: 16-byte array
REAL version: real variable

x-coordinate for the other end of the line, the tail of the arrow, or the diagonally opposite corner of the box. The value of the x-coordinate can range from 0 to 100.

Parameters (continued)

y2

ASCII version: 16-byte array
REAL version: real variable

y-coordinate for the same point described by *x2*. The value of the y-coordinate can range from 0 to 100.

color

Integer code specifying the color of the annotative symbol:

1 color of pen 1, usually black
2 color of pen 2, usually red
3 color of pen 3, usually green
4 color of pen 4, usually blue
5...16 if additional pens are used.

This parameter is ignored if the plotting device is a non-color terminal or plotter. The default color, specified as 0 or -1, is black.

linetype

Integer code specifying the texture of the lines making up the annotative symbol. To use the default, which is a solid line, supply a -1. A 0 represents no texture (no annotation). To specify textures enter 1-8. See Appendix A for table of line textures.

DISCUSSION

Call GSYMBOL[A]SET once for each annotation. For text annotations, call GTEXT[A]SET. The coordinates of each annotation should be determined by calling GREAD[A]ANOT.

For dynamic digitizing call GREADANOT twice and pass the returned coordinates on to GSYMBOLSET.

See page 6-9.

GSYMBOL[A]SET draws the annotations on the current graphing device; GSYMBOL[A]INQ moves the graphics cursor to the *x1,y1* coordinate.

GTERMGRAF

Terminates access to DSG/3000 and closes any open chart files. Callable from all languages.

FORMAT

```
                IA
     GTERMGRAF (graf)
                i/o
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

DISCUSSION

GTERMGRAF will call GCLOSEFILE if any chart files are still open. It also closes internal DSG/3000 files.

GTEXT[A] $\left\{\begin{array}{l}\textbf{SET}\\\textbf{INQ}\end{array}\right\}$

Sets or returns a text annotation. The ASCII version, callable from COBOL as well as other languages, is designated by an "A" in the procedure name.

FORMAT

```
                  IA      I     R    R     I      I      I       I
GTEXTSET   (graf,annotid,x1, y1, lorg,angle,color,charsize,
                 i/o      in    in   in    in     in     in      in

                   BA       I       I
                textbuf,textlen,font)
                   in      in      in

                 IA      I      R     R     I     I      I     I
GTEXTINQ   (graf,annotid,x1, y1, lorg,angle,color,charsize,
                 i/o     in    out out  out    out    out    out

                  BA       I             I     I
                textbuf,textbuflen,textlen,font)
                  out      in          out    out

CALL "GTEXTASET" USING graf annotid x1 y1 lorg angle color
   charsize textbuf textlen font

CALL "GTEXTAINQ" USING graf annotid x1 y1 lorg angle color
   charsize textbuf textbuflen textlen font
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

annotid Integer variable specifying the unique annotation identifying number. Annotation numbers must be non-negative.

x1 ASCII version: 16-byte array
REAL version; real variable

x-coordinate of the digitized point.

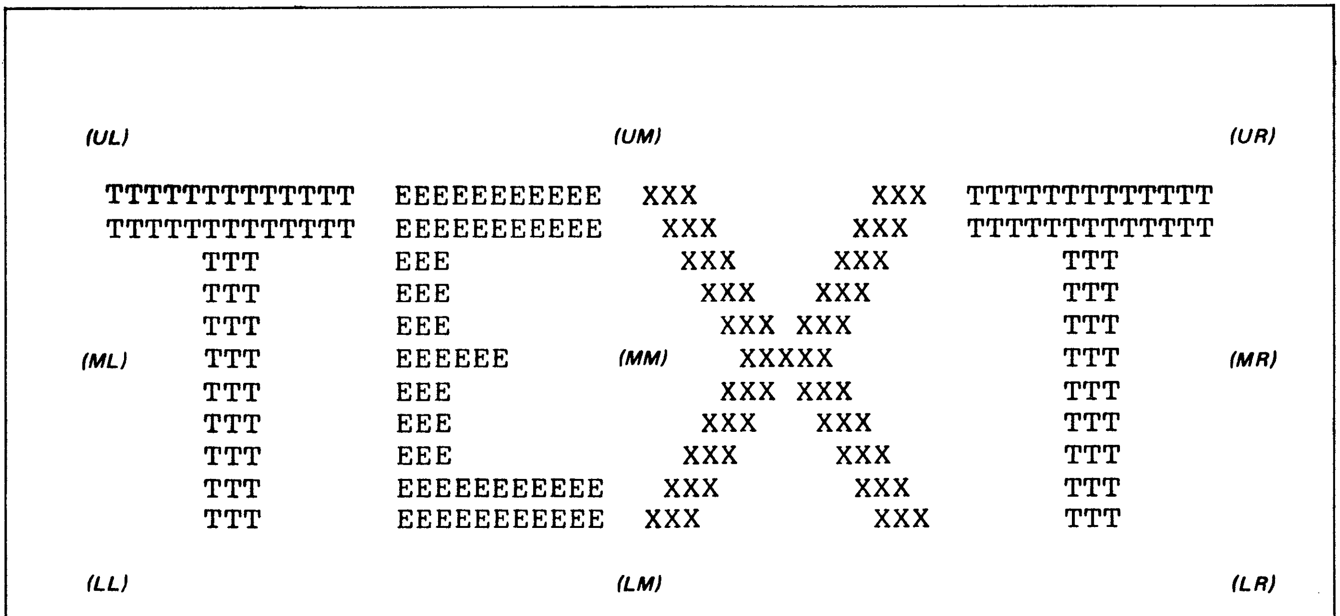
y1 ASCII version: 16-byte array
REAL version: real variable

y-coordinate of the digitized point.

lorg Integer code specifying the label origin:

1 lower left
2 middle left
3 upper left
4 lower middle
5 middle middle
6 upper middle
7 lower right
8 middle right
9 upper right

The label origin tells DSG/3000 which part of the text string should be positioned at the digitized point (x1,y1). Figure 7-1 shows an example of a text string and the locations of the cursor that can be specified.



Label origin codes:

UL	=	Upper Left	UM	=	Upper Middle	UR	=	Upper Right
ML	=	Middle Left	MM	=	Middle Middle	MR	=	Middle Right
LL	=	Lower Left	LM	=	Lower Middle	LR	=	Lower Right

Figure 7-1. Position of Label Origin with Respect to Text Annotation

Parameters (continued)

angle Integer variable ranging from 0 through 359 specifying the angle at which the text is to be placed.

color Integer code specifying the color of the annotation:

1 color of pen 1, usually black
2 color of pen 2, usually red
3 color of pen 3, usually green
4 color of pen 4, usually blue
5...16 if additional pens are used.

This parameter is ignored if the plotting device is a non-color plotter or terminal. The default color, specified as 0 or -1, is black. See Appendix A for additional color information.

charsize Integer variable specifying the relative character size, from 1 (smallest) to 72 (biggest). See Appendix A for examples of sizes.

textbuf Byte array holding the text string. Maximum length is 100 bytes. May contain special text string (see below).

textlen Integer variable specifying length of text string.

font Integer code specifying pre-established font types. Maximum of 4 fonts accepted for a single chart. Enter 1-4. Default=1 (usually Stick font). See GFONTSET.

textbuflen Integer variable specifying the length of the buffer holding the text string.

DISCUSSION

Call GTEXT[A]SET once for each text annotation. Other annotations are set by calling GSYMBOL[A]SET. The coordinates should be determined by calling GREADANOT[A]. (See page 6-9.)

GTEXT[A]SET draws the text annotation on the screen: GTEXT[A]INQ moves the graphics cursor to the digitized point (*x1,y1*).

Four type fonts may be active for a single chart. They are established through GFONTSET, which allows you to map the fonts you have selected to a font number.

Special Character Strings

A text annotation can contain any of the following special strings:

$TODAY	Represents the current date in the form mm/dd/yy with leading zeros suppressed. For example, 9/16/82.
$MONTH	Represents the current month spelled out in full. For example, SEPTEMBER.
$M	Represents the current month numerically. For example, 9.

$DAY Represents the current numeric day of the month. For example, 15.

$YEAR Represents the current year in 4 digits. For example, 1982.

$Y Represents the current year by its last two digits. For example, 82.

Here is an example of a text annotation with a special string embedded:

HIGHEST SALES VOLUME AS OF $TODAY.

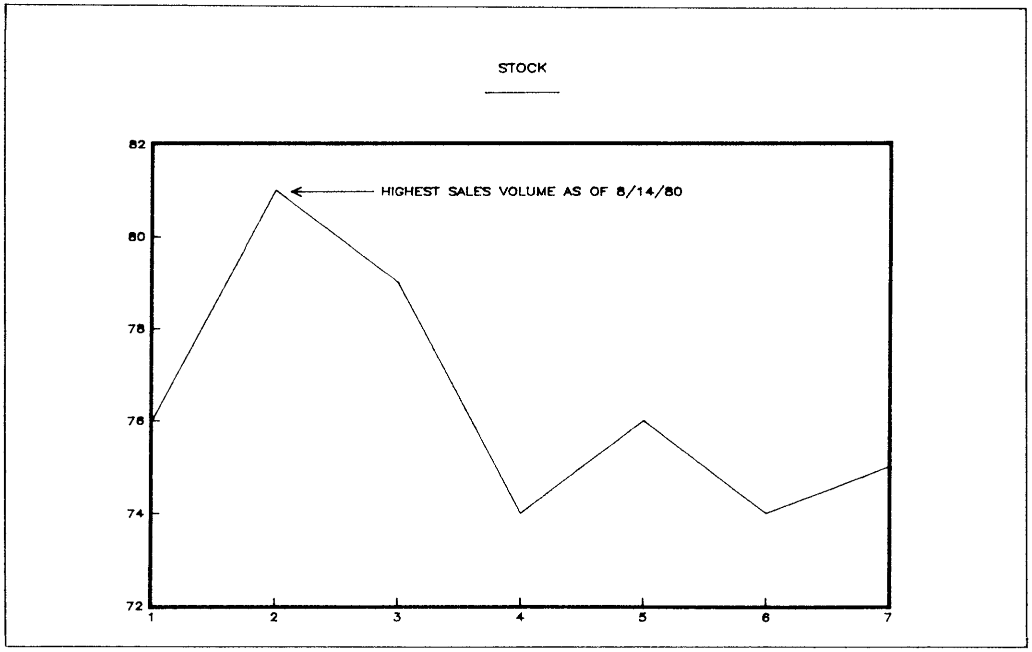


Figure 7-2. Annotation with Special Character String

GTEXTCONTROL $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Allows for specification of size, color, and font for legends, labels, and titles. Callable from all languages.

FORMAT

```
                   IA    I      I    I    I
GTEXTCONTROLSET   (graf,option,color,size,font)
                   i/o   in     in   in   in

                   IA    I      I    I    I
GTEXTCONTROLINQ   (graf,option,color,size,font)
                   i/o   in     out  out  out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

option Integer code indicating which text entity will be affected. Code as follows:

1 Title
2 Subtitle
3 Footnote
4 Horizontal axis title (bar and line charts only)
5 Vertical axis title (bar and line charts only)
6 Right axis title (bar and line charts only)
7 Second footnote (pie charts only)
8 X-axis labels (bar and line charts only)
9 Y-axis labels (bar and line charts only)
10 Pie segment labels (pie charts only)
11 Legend text (bar and line charts only)

Parameters (continued)

color Integer code corresponding to the color assigned to the text entity:

1	color of pen 1
2	color of pen 2
3	color of pen 3
4	color of pen 4
.	.
.	.
.	.
16	color of pen 16

0 = default; maximum value = 16

If the device has less than 16 colors, the numbers will wrap around through the colors available. On a 4-pen plotter, for example, colors 1,5,9, and 13 will all use pen 1.

This parameter is ignored if the plotting device is a non-color device.

size (0-72) Integer code corresponding to the size of the characters. The smallest is 1; 72 is the maximum value. The default (0) size depends on the text option.

Parameters (continued)

The default sizes are:

Bar and Line Charts		Pie Charts	
Main Title	8	Main Title	8
Subtitle	6	Subtitle	6
Left Axis Title	5	Segment Labels	4
Bottom Axis Title	5		
X-Axis Labels	4		
Y-axis Labels	4		
Legend Text	5		

font (0-4) Integer code corresponding to the font previously defined through GFONTSET. The default font (0) is font 1 (Stick font).

See GFONTSET.

DISCUSSION

The size of the text is relative to the entire chart size. For small charts, you may want the text size to be relatively large to insure that it is readable. Note that increasing the size of any text may cause character truncation, depending on the length and position of the string on the chart, or it may result in overlaying other portions of the chart.

GTITLE $\left\{\begin{matrix}\textbf{SET}\\\textbf{INQ}\end{matrix}\right\}$

Sets or returns title, subtitle, footnote, and axes titles. Callable from all languages.

FORMAT

```
                IA    I      BA        I      I   I
      GTITLESET (graf,option,titlebuf,titlelen,size,font,
                i/o   in     in        in     in  in

                 I
                useroptions)
                 in

                IA    I      BA          I          I     I
      GTITLEINQ (graf,option,titlebuf,titlebuflen,titlelen,size,
                i/o   in     out         in         out   out

                 I       I
                font,useroptions)
                 out     out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

option Integer code indicating which title is being set:

1 title
2 subtitle
3 footnote

Parameters (continued)

4 X-axis (horizontal) title (bar and line graphs only)
5 Y-axis (vertical) title (bar and line graphs only)
6 right axis title (bar and line graphs only)
7 second footnote (pie charts only)

titlebuf Byte array holding the title. The maximum lengths allowed are as follows:

title	45 characters
subtitle, same size as title	45 characters
subtitle, smaller than title	64 characters
all others	90 characters

titlelen Integer variable specifying length in bytes of title.

size Integer code specifying the relative character size of the subtitle:

0 small = text size 6 (default)
1 big = text size 8

This parameter is ignored unless a subtitle is being set.

font Integer variable specifying pre-established type font. A maximum of four fonts may be active for any chart. Enter 1-4. Default=1 (usually STICK font).

useroptions Integer variable ignored by procedure. User comment area.

titlebuflen Integer variable specifying length in bytes of the title buffer.

DISCUSSION

Any graph can be given a title, subtitle (smaller type), and footnote. In addition, bar and line graphs can have their axes titled. Pie charts are allowed a second line for footnotes.

The main title, subtitle, footnote, and X-axis title are centered when printed on the graph. The Y-axis title is left-justified. Leading and trailing blanks are stripped when the graph is printed although they are retained when GTITLEINQ returns them.

Call GTITLESET once for each title.

Four type fonts may be active for a single chart. They are established through GFONTSET, which allows you to map the fonts you have selected to a font number.

Note that if the size of titles is changed using GTEXTCONTROLSET, character truncation or title overlay may occur.

Special Character Strings

A title can contain any of the following special strings:

$TODAY	Represents the current date in the form mm/dd/yy with the leading zeros suppressed. For example, 9/16/82.
$MONTH	Represents the current month spelled out in full. For example, SEPTEMBER.

$M   Represents the current month numerically. For example, 9.

$DAY   Represents the current numeric day of the month. For example, 16.

$YEAR   Represents the current year in 4 digits. For example, 1982.

$Y   Represents the current year by its last two digits. For example, 82.

GTRAN {SET | INQ}

Sets or returns variable transformations as a group for the current chart. Callable from all languages.

FORMAT

```
                IA     I         BA    BA      I
      GTRANSET (graf,numtrans,varbuf,exprbuf,exprlen)
                i/o    in        in    in      in

                IA      I         I       I        BA      I
      GTRANINQ (graf,firstrans,maxtrans,numtrans,varbuf,varbuflen,
                i/o     in        in      out      out     in

                   BA        I       I
                exprbuf,exprbuflen,exprlen)
                   out       in      out
```

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

numtrans Integer variable specifying the number of transformations. This is also the number of variable names in *varbuf* and the number of expressions in *exprbuf*. The maximum is 10.

varbuf 16-byte array holding the names of the variables being created.

Parameters (continued)

exprbuf	Byte array holding the transformation expressions. All transformation expressions are of the same length, padded with blanks if necessary, and can use any operators given in Table 3-3.
exprlen	Integer variable specifying the length in bytes of a single transformation expression.
firstrans	Integer variable specifying the relative number of the first transformation you want returned. The first transformation specified in the expression buffer is numbered one.
maxtrans	Integer variable specifying the maximum number of transformations you want returned.
varbuflen	Integer variable specifying the length in bytes of the variable buffer. A warning is returned if the buffer is too small to receive the variable names.
exprbuflen	Integer variable specifying the length in bytes of the expression buffer. A warning is returned if the buffer is too small to receive the transformations.

DISCUSSION

See page 3-18.

Note: The special functions, CUMULATE (Cumulation) and MOVEAVG (Moving Average) may not be nested. Also, total *numtrans* is available from GDATAINFO.

COLORS, TEXTURES, MARKERS AND FONTS

APPENDIX A

Colors

HP makes a variety of pen colors and pen widths available to its users as well as transparency colors in two widths and spectrum colors for paper. This presents you with a variety of choices and a chance to experiment with the effects best suited to your charting needs.

There are 1- ,2- ,4- ,8- and 16-pen holders. The maximum number of colors for your graph equals the number of pens in your holder. The colors are stored in a circular buffer.

If your holder has fewer than 8 pens, the colors will be repeated in the order they have been loaded into the holder. For example, if you have loaded black, green, red, and blue in holders one through four respectively, the fifth color will be black, the sixth red, the seventh green, and the eighth blue. This repetition will continue for all colors called for in your chart.

Generally, black is the color chosen for pen holder 1, because the pen in this position is used to frame your charts, draw axes, and write labels. Red is often used for elements in the chart you wish to emphasize. Below is the standard location for the first four pen positions.

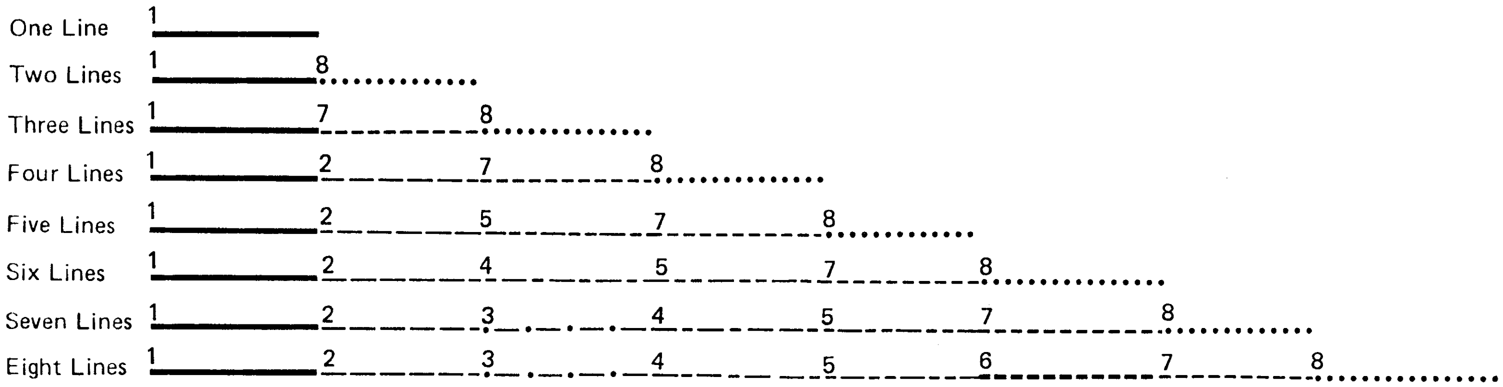
Location	Standard Color
Pen 1	Black
Pen 2	Red
Pen 3	Green
Pen 4	Blue

Line Textures

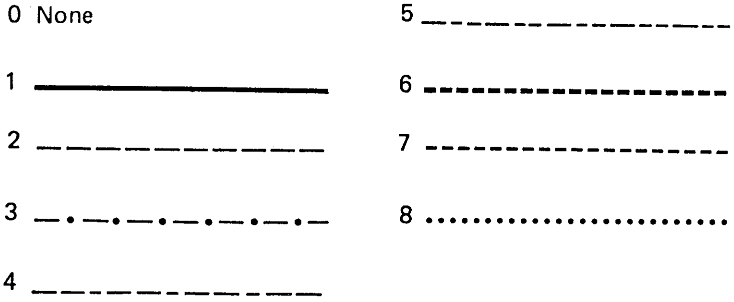
The line textures were developed to visually create a logical transition between two basic line types, continuous and dotted. These line textures provide enough contrast between them so that there is an obvious difference, but not so much that they compete with each other as primary focal points for the graph.

The following describes the line texture types, their dimensions and order of use.

Order of Use



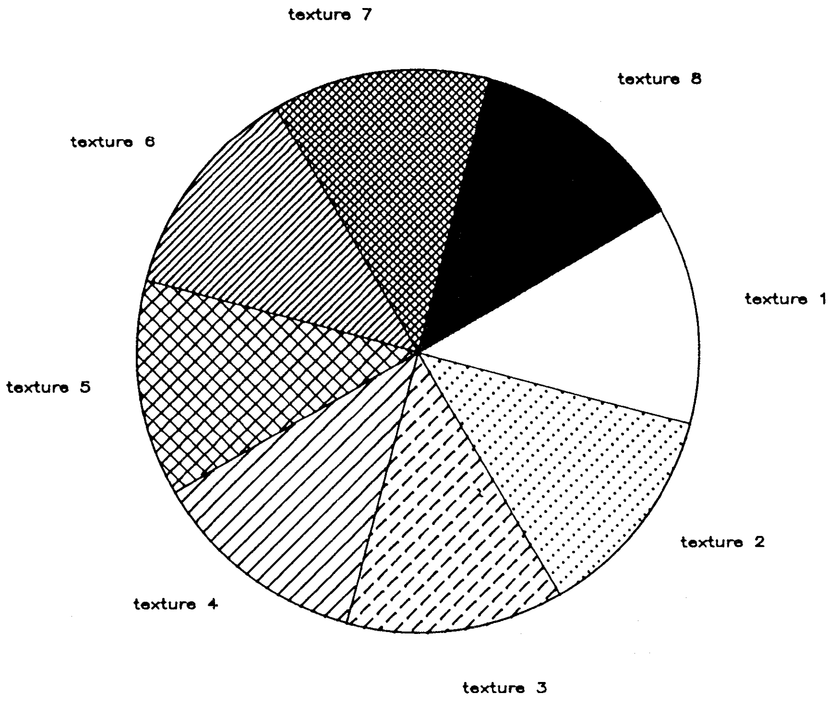
Line Textures



Data Point Markers

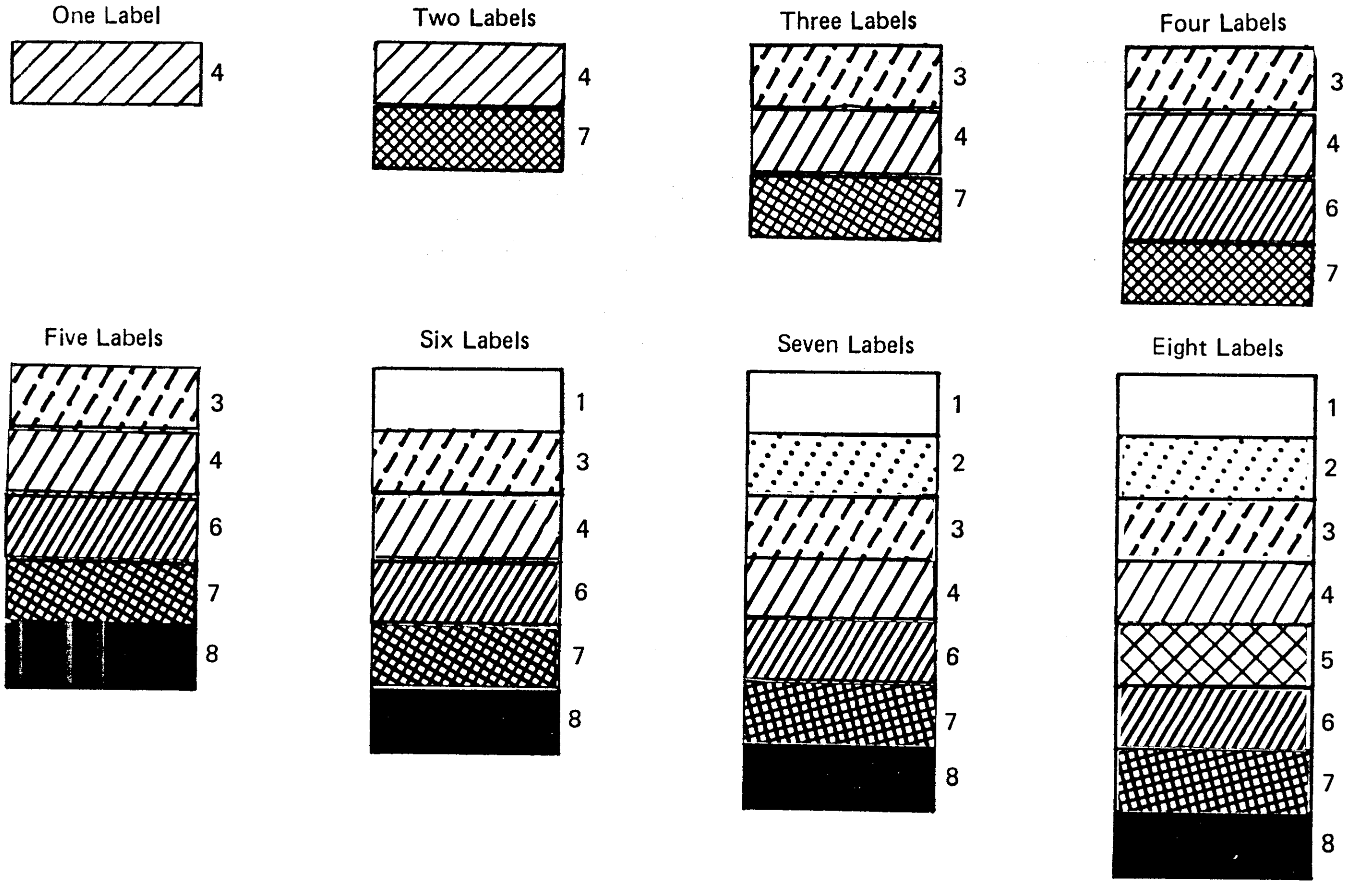
0 = none	5 = ◇
1 = *	6 = ○
2 = +	7 = ⊕
3 = △	8 = x
4 = □	

Available Textures



Surface Textures--Order of Use

SURFACE TEXTURE TYPES
AND ORDER OF USE



Text Sizes

The text size examples shown below are software sizes. These sizes conform to point size measurements when you are using a standard (8 1/2 by 11 inches) sheet of paper. When the paper is a different size, these sizes will expand or reduce accordingly.

Hardware sizes will adjust in the "best fit" mode. When hardware sizes do not match the software size exactly, the hardward size closest to a perfect match will be selected automatically. Quick-drawing hardware sizes may be used effectively when you are designing the chart and wish to draw your chart on the screen to make sure the elements are balanced and command the visual attention they should. You can then use the slow-drawing but more refined software text for plotting the finished chart.

POINT SIZE EXAMPLE (6–72)

6 POINT

8 POINT

10 POINT

12 POINT

15 POINT

20 POINT

25 POINT

30 POINT

35 POINT

40 POINT

50 POINT

60 POINT

72 POINT

Text Fonts

Below are examples of the text fonts presently available for use. These are software fonts. Only one hardware font, Stick, is now available.

THIS IS STICK

this is stick

THIS IS SCRIPT

this is script

THIS IS ROMAN

this is roman

THIS IS GOTHIC

this is gothic

Native Language Characters

DSG/3000 allows users to specify native language characteristics for seven languages for both hardware and software fonts. Examples of the characters for each language are shown below.

1. USASCII (Language=1) — Software

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

#@[\]^`{|}~ !$%&*()_—+=:;"'?/<>,.

USASCII (Language=1) - Hardware

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

#@ [\] ^ ` {| } ~ ! $%&* () _-+=: ; " '?/<>, .

2. Swedish (Language=2) — Software

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

#ÉÄÖÅÜéäöåü !$%&*()_—+=:;"´?/<>,.

Swedish (Language=2) - Hardware

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

ÉÄÖÅÜéäöåü ! $%&* () _-+=: ; "´?/<>, .

3. Danish (Language=3) — Software

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

#@ÆØÅ^`æøå~  !$%&*()_—+=:;"´?/<>,.
```

Danish (Language=3) - Hardware

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

@ÆØÅ `æoå   ! $%&* () _-+=: ; " '?/<>, .
```

4. French (Language=4) — Software

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

£à° ç§^`éùè¨   !$%&*()_—+=:;"´?/<>,.
```

French (Language=4) - Hardware

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

£à˚ç ^`éùè¨  ! $%&* () _-+=: ; "´?/<>, .
```

5. German (Language=5) — Software

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

£§ÄÖÜ^`äöüß  !$%&*()_—+=:;"´?/<>,.
```

German (Language=5) - Hardware

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

£@ÄÖÜ^`äöüB  ! $%&* () _-+=: ; "´?/<>, .
```

6. United Kingdom (Language=6) — Software

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

£@[\]^`{|}~   !$%&*()_—+=:;"´?/<>,.

United Kingdom (Language=6) - Hardware

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

£@[/]^` [!]   ! $%&* () _-+=: ; "´?/<>, .

7. Spanish (Language=7) — Software

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

#@¡Ñ¿°`{ñ}~   !$%&*()_—+=:;"´?/<>,.

Spanish (Language=7) - Hardware

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

@ ¡Ñ¿ ` [ñ] ~   ! $%&* () _-+=: ; "´?/<>, .

PROGRAMMATIC EXAMPLES

APPENDIX B

The following sample programs in COBOL, FORTRAN, SPL, PASCAL, and BASIC assume that you have created the chart file CF1 and the pie chart FIRSTCHART as described in Section IV of this manual. This chart describes budget allocations among six departments of a company. There are six segments, and each is labeled with the department name and the percent of budget which that department receives. The segment describing the Research and Development department is exploded. The chart also has a two-line title and a two-line footnote.

The sample programs will add six different textures to the segments, four different colors, and will produce the chart on a 9872A 4-pen plotter. If you have a different plotter, you will have to substitute appropriate values for that device in the call to GDEVICESET.

The SPL, FORTRAN, PASCAL and COBOL programs must be :PREPed with a MAXDATA value of 32000. Required capabilities are:

```
CAP=PH,MR,DS
```

```
 10 REM****************************************************************
 20 REM*                                                              *
 30 REM*               BASIC SAMPLE PROGRAM                           *
 40 REM*                                                              *
 50 REM****************************************************************
 60 REM                                                               *
 70 REM                   DECLARATIONS                                *
 80 REM                                                               *
 90 REM****************************************************************
100 REM*** For GINITGRAF: G is GRAF, S is GRAFSIZE, L is LANGUAGE
110 REM
120 INTEGER G[3700],S,L
130 REM
140 REM*** For GOPENFILE: F$ is FILENAME, A1 is ACCESS
150 REM
160 DIM F$[36]
170 INTEGER A1
180 REM
190 REM*** For GGETCHART: C$ is CHARTNAME
200 REM
210 DIM C$[16]
220 REM
230 REM*** For GPIESET: P1 is PERCENT, A2 is ACTUAL, C[4] is COLORBUF,
240 REM*** C1 is COLORBUFLEN, T[6] is TEXTUREBUF, T1 is TEXTUREBUFLEN
250 REM
260 INTEGER P1,A2,C[4],C1,T[6],T1
270 REM
280 REM*** For GDEVICESET: D$ is DEVICE, H is HPIB, C2 is COPIES,
290 REM*** P2 is PENSPEED, C3 is CHARQUAL
300 REM
310 DIM D$[6]
320 INTEGER H,C2,P2,C3
330 REM
340 REM*** For GEXECHART: O is OPTIMIZE
350 REM
360 INTEGER O
370 REM
380 REM*** For GERRORMSG: M$ is MSGBUF, M1 is MSGBUFLEN, M2 is MSGLEN
390 REM
400 DIM M$[72]
410 INTEGER M1,M2
420 REM
430 REM*** P$ is for error routine
440 REM
450 DIM P$[16]
460 REM
470 REM****************************************************************
480 REM                                                               *
490 REM                     MAIN PROGRAM                              *
500 REM                                                               *
510 REM****************************************************************
520 REM
```

```
1050 P2=0
1060 C3=1
1070 CALL GDEVICESET(G[*],D$,H,C2,P2,C3)
1080 IF G[1]<>0 THEN DO
1090   P$="GDEVICESET"
1100   GOSUB 1380
1110 DOEND
1120 REM
1130 REM*****     Prepare data for graphing
1140 REM
1150 CALL GEXECDATA(G[*])
1160 IF G[1]<>0 THEN DO
1170   P$="GEXECDATA"
1180   GOSUB 1380
1190 DOEND
1200 REM
1210 REM*****     Draw chart, optimizing when possible
1220 REM
1230 O=1
1240 CALL GEXECHART(G[*],O)
1250 IF G[1]<>0 THEN DO
1260   P$="GEXECHART"
1270   GOSUB 1380
1280 DOEND
1290 REM
1300 REM*****     Terminate access to DSG/3000; closes chart file
1310 REM
1320 CALL GTERMGRAF(G[*])
1330 IF G[1]<>0 THEN DO
1340   P$="GTERMGRAF"
1350   GOSUB 1380
1360 DOEND
1370 END
1380 REM*****************************************************************
1390 REM*                                                               *
1400 REM*              ERROR ROUTINE                                    *
1410 REM*                                                               *
1420 REM*****************************************************************
1430 REM
1440 PRINT "Error returned by call to ",P$
1450 M1=72
1460 CALL GERRORMSG(G[*],M$,M1,M2)
1470 PRINT M$
1480 CALL GTERMGRAF(G[*])
1490 END
>
```

```
 530 REM*****     Initialize GRAF array; MUST be first call
 540 REM
 550 S=3700
 560 L=1
 570 CALL GINITGRAF(G[*],S,L)
 580 IF G[1]<>0 THEN DO
 590   P$="GINITGRAF"
 600   GOSUB 1380
 610   REM
 620 DOEND
 630 REM*****     Open existing chart file CF1 in exclusive mode
 640 REM
 650 F$="CF1 "
 660 A1=1
 670 CALL GOPENFILE(G[*],F$,A1)
 680 IF G[1]<>0 THEN DO
 690   P$="GOPENFILE"
 700   GOSUB 1380
 710 DOEND
 720 REM
 730 REM*****     Retrieve chart FIRSTCHART
 740 REM
 750 C$="FIRSTCHART      "
 760 CALL GGETCHART(G[*],C$)
 770 IF G[1]<>0 THEN DO
 780   P$="GGETCHART"
 790   GOSUB 1380
 800 DOEND
 810 REM
 820 REM*****     Specify 4 colors and 6 textures for pie segments
 830 REM*****     Label percentages, but not actual values
 840 REM
 850 P1=1
 860 A2=0
 870 DATA 1,2,3,4
 880 MAT READ C
 890 C1=4
 900 DATA 1,2,3,4,5,6
 910 MAT READ T
 920 T1=6
 930 CALL GPIESET(G[*],P1,A2,C[*],C1,T[*],T1)
 940 IF G[1]<>0 THEN DO
 950   P$="GPIESET"
 960   GOSUB 1380
 970 DOEND
 980 REM
 990 REM*****     Draw 1 copy of the chart on a 9872A plotter
1000 REM*****     at fast pen speed using hardware characters
1010 REM
1020 D$="9872A "
1030 H=5
1040 C2=1
```

```
$CONTROL USLINIT
<<************************************************************>>
<<                                                            >>
<<                                                            >>
<<           SPL SAMPLE PROGRAM                               >>
<<                                                            >>
<<                                                            >>
<<               Global Declarations                          >>
<<                                                            >>
<<************************************************************>>

BEGIN

INTEGER ARRAY  GRAF(0:1599);
DEFINE
  STATUS    = GRAF (0) #,     << Status word >>
  PARAM     = GRAF (1) #,     << Parameter in error, if any >>
  ELEMENT   = GRAF (2) #,     << Element of the parameter in error >>
  FSERROR   = GRAF (3) #;     << File system error number, if any >>

<<***  GINITGRAF parameters  ***>>
     INTEGER           GRAF'SIZE,
                       LANGUAGE;

<<***  GOPENFILE parameters  ***>>
     BYTE ARRAY        FILE'NAME(0:35):=36(" ");
     INTEGER           ACCESS;

<<***  GGETCHART parameters  ***>>
     BYTE ARRAY        CHART'NAME(0:15):=16(" ");

<<***  GPIESET parameters  ***>>
     INTEGER           PERCENT,
                       ACTUAL,
                       COLOR'BUF'LEN,
                       TEXTURE'BUF'LEN;
     INTEGER ARRAY     COLOR'BUF(0:3),
                       TEXTURE'BUF(0:3);

<<***  GDEVICESET parameters  ***>>
     BYTE ARRAY        DEVICE(0:5);
     INTEGER           HPIB,
                       COPIES,
                       PEN'SPEED,
                       CHAR'QUAL;

<<***  GEXECHART parameters  ***>>
     INTEGER           OPTIMIZE;

<<***  GERRORMSG parameters  ***>>
     BYTE ARRAY        MSG'BUF(0:71);
     INTEGER           LENGTH,
                       RETURN'LEN;

BYTE ARRAY     PROCEDURE'NAME(0:15):=16(" ");
LOGICAL ARRAY  L'MSG'BUF(*)=MSG'BUF(0);

```

```
<<************************************************************>>
<<                                                            >>
<<            DSG/3000 PROCEDURES                             >>
<<                                                            >>
<<************************************************************>>
<<
INTRINSIC
  GINITGRAF,
  GOPENFILE,
  GGETCHART,
  GEXECDATA,
  GEXECHART,
  GTERMGRAF,
  GERRORMSG,
  GDEVICESET,
  GPIESET;
>>
INTRINSIC PRINT;


PROCEDURE GINITGRAF (GRAF,GRAF'SIZE,LANGUAGE);
  INTEGER ARRAY GRAF;
  INTEGER GRAF'SIZE, LANGUAGE;
  OPTION EXTERNAL;
PROCEDURE GOPENFILE(GRAF,FILE'NAME,ACCESS);
  INTEGER ARRAY GRAF;
  BYTE ARRAY FILE'NAME;
  INTEGER ACCESS;
  OPTION EXTERNAL;
PROCEDURE GGETCHART(GRAF,CHART'NAME);
  INTEGER ARRAY GRAF;
  BYTE ARRAY CHART'NAME;
  OPTION EXTERNAL;
PROCEDURE GEXECDATA(GRAF);
  INTEGER ARRAY GRAF;
  OPTION EXTERNAL;
PROCEDURE GEXECHART (GRAF,OPTIMIZE);
  INTEGER ARRAY GRAF;
  INTEGER OPTIMIZE;
  OPTION EXTERNAL;
PROCEDURE GTERMGRAF(GRAF);
  INTEGER ARRAY GRAF;
  OPTION EXTERNAL;
PROCEDURE GERRORMSG(GRAF,MSG'BUF,LENGTH,RETURN'LEN);
  INTEGER ARRAY GRAF;
  BYTE ARRAY MSG'BUF;
  INTEGER LENGTH, RETURN'LEN;
  OPTION EXTERNAL;
PROCEDURE GDEVICESET(GRAF,DEVICE,HPIB,COPIES,PEN'SPEED,CHAR'QUAL);
  INTEGER ARRAY GRAF;
  BYTE ARRAY DEVICE;
  INTEGER HPIB, COPIES, PEN'SPEED, CHAR'QUAL;
  OPTION EXTERNAL;
PROCEDURE GPIESET(GRAF,PERCENT,ACTUAL,COLOR'BUF,COLOR'BUF'LEN,
                  TEXTURE'BUF,TEXTURE'BUF'LEN);
  INTEGER ARRAY GRAF, COLOR'BUF, TEXTURE'BUF;
  INTEGER PERCENT, ACTUAL, COLOR'BUF'LEN, TEXTURE'BUF'LEN;
  OPTION EXTERNAL;

<<************************************************************>>
<<                                                            >>
<<                 ERROR ROUTINE                              >>
<<                                                            >>
<<************************************************************>>

PROCEDURE ERROR (PROCEDURE'NAME);
BYTE ARRAY PROCEDURE'NAME;
BEGIN
  MOVE MSG'BUF:="Error returned by call to ";
  MOVE MSG'BUF(26):=PROCEDURE'NAME, (16);
  PRINT (L'MSG'BUF,-42,%40);
  LENGTH:=72;
  GERRORMSG (GRAF, MSG'BUF, LENGTH, RETURN'LEN);
  PRINT (L'MSG'BUF,-RETURN'LEN,0);
END;


<<************************************************************>>
<<                                                            >>
<<                 MAIN PROGRAM                               >>
<<                                                            >>
<<************************************************************>>

<<  Initialize GRAF array; MUST be first call  >>

GRAF'SIZE:=1600;
LANGUAGE:=3;
GINITGRAF (GRAF, GRAF'SIZE, LANGUAGE);
IF STATUS  <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GINITGRAF";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Open existing chart file CF1 in exclusive mode  >>

MOVE FILE'NAME:="CF1 ";
ACCESS:=1;
GOPENFILE (GRAF, FILE'NAME, ACCESS);
IF STATUS <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GOPENFILE";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Retrieve chart FIRSTCHART  >>

MOVE CHART'NAME:="FIRSTCHART";
GGETCHART (GRAF, CHART'NAME);
IF STATUS <> 0 THEN
```

```
  BEGIN
    MOVE PROCEDURE'NAME:="GGETCHART";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Specify 4 colors and 6 textures for pie segments  >>
<<  Label percentages, but not actual values          >>

PERCENT:=1;
ACTUAL:=0;
MOVE COLOR'BUF:=(1,2,3,4);
COLOR'BUF'LEN:=4;
MOVE TEXTURE'BUF:=(1,2,3,4,5,6);
TEXTURE'BUF'LEN:=6;
GPIESET (GRAF, PERCENT, ACTUAL, COLOR'BUF, COLOR'BUF'LEN,
         TEXTURE'BUF, TEXTURE'BUF'LEN);
IF STATUS <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GPIESET";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Draw 1 copy of the chart on a 9872A plotter at fast pen  >>
<<  speed using hardware characters                          >>

MOVE DEVICE:="9872A ";
HPIB:=5;
COPIES:=1;
PEN'SPEED:=0;
CHAR'QUAL:=1;
GDEVICESET (GRAF, DEVICE, HPIB, COPIES, PEN'SPEED, CHAR'QUAL);
IF STATUS <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GDEVICESET";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Prepare data for graphing  >>

GEXECDATA (GRAF);
IF STATUS <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GEXECDATA";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Draw chart, optimizing when possible  >>

OPTIMIZE:=1;
```

```
GEXECHART (GRAF, OPTIMIZE);
IF STATUS <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GEXECHART";
    ERROR (PROCEDURE'NAME);
    GO TERMINATE;
  END;


<<  Terminate access to DSG/3000; closes chart file  >>

TERMINATE:

GTERMGRAF (GRAF);
IF STATUS <> 0 THEN
  BEGIN
    MOVE PROCEDURE'NAME:="GTERMGRAF";
    ERROR (PROCEDURE'NAME);
  END;

END.
```

```
$CONTROL USLINIT
C****************************************************************
C                                                               *
C            FORTRAN SAMPLE PROGRAM                             *
C                                                               *
C   Global Declarations                                         *
C                                                               *
C****************************************************************
C
      INTEGER GRAF(1600)
C
C....GINITGRAF parameters
      INTEGER GRAFSIZE, LANGUAGE
C
C....GOPENFILE parameters
      CHARACTER*36 FILENAME
      INTEGER ACCESS
C
C....GGETCHART parameters
      CHARACTER*16 CHARTNAME
C
C....GPIESET parameters
      INTEGER PERCENT, ACTUAL, COLORBUF(4), COLORBUFLEN
      INTEGER TEXTUREBUF(6), TEXTUREBUFLEN
C
C....GDEVICESET parameters
      CHARACTER*6 DEVICE
      INTEGER HPIB, COPIES, PENSPEED, CHARQUAL
C
C....GEXECHART parameters
      INTEGER OPTIMIZE
C
C....GERRORMSG parameters
      CHARACTER*72 MSGBUF
      INTEGER LENGTH, RETURNLEN
C
      CHARACTER*16 PROCNAME
C
C...Equivalence status word to GRAF array
C
      EQUIVALENCE(STATUS,GRAF(1))
C
c     SYSTEM INTRINSIC GINITGRAF,GOPENFILE,GGETCHART,GEXECHART
c     SYSTEM INTRINSIC GEXECDATA,GERRORMSG,GPIESET,GTERMGRAF
c     SYSTEM INTRINSIC GDEVICESET
C
C***************************************************************
C                                                              *
C                 MAIN PROGRAM                                 *
C                                                              *
C***************************************************************
C
C...Initialize GRAF array; MUST be first call
C
      GRAFSIZE=1600
      LANGUAGE=2
      CALL GINITGRAF (GRAF, GRAFSIZE, LANGUAGE)
      PROCNAME = "GINITGRAF"
      IF (STATUS .NE. 0) GOTO 900
C
C...Open existing chart file CF1 in exclusive mode
C
      FILENAME="CF1 "
      ACCESS=1
      CALL GOPENFILE (GRAF, FILENAME, ACCESS)
      PROCNAME = "GOPENFILE"
      IF (STATUS .NE. 0) GOTO 900
C
C...Retrieve chart FIRSTCHART
C
      CHARTNAME="FIRSTCHART "
      CALL GGETCHART (GRAF, CHARTNAME)
      PROCNAME = "GGETCHART"
      IF (STATUS .NE. 0) GOTO 900
C
C...Specify 4 colors and 6 textures for pie segments
C...Label percentages, but not actual values
C
      PERCENT=1
      ACTUAL=0
C.....Set up COLORBUF
      DO 10 I=1,4
         COLORBUF(I)=I
   10    CONTINUE
      COLORBUFLEN=4
C.....Set up TEXTUREBUF
      DO 20 I=1,6
         TEXTUREBUF(I)=I
   20    CONTINUE
      TEXTUREBUFLEN=6
      CALL GPIESET (GRAF, PERCENT, ACTUAL, COLORBUF,
     &              COLORBUFLEN, TEXTUREBUF, TEXTUREBUFLEN)
      PROCNAME = "GPIESET"
      IF (STATUS .NE. 0) GOTO 900
C
C...Draw 1 copy of the chart on a 9872A plotter at
C...fast pen speed using hardware characters
C
      DEVICE="9872A "
      HPIB=5
      COPIES=1
      PENSPEED=0
      CHARQUAL=1
      CALL GDEVICESET (GRAF, DEVICE, HPIB, COPIES, PENSPEED,
     &                  CHARQUAL)
      PROCNAME = "GDEVICESET"
      IF (STATUS .NE. 0) GOTO 900
C
C...Prepare data for graphing
C
      CALL GEXECDATA (GRAF)
      PROCNAME = "GEXECDATA"
      IF (STATUS .NE. 0) GOTO 900
C
```

```
C...Draw chart, optimizing when possible
C
      OPTIMIZE=1
      CALL GEXECHART (GRAF, OPTIMIZE)
      PROCNAME = "GEXECHART"
      IF (STATUS .NE. 0) GOTO 900
C
C...Terminate access to DSG/3000; closes chart file
C
      CALL GTERMGRAF (GRAF)
      PROCNAME = "GTERMGRAF"
      IF (STATUS .NE. 0) GOTO 900
C
C
      STOP
C
C****************************************************************
C*                                                              *
C*            ERROR ROUTINE                                     *
C*                                                              *
C****************************************************************
C
900   DISPLAY "Error returned by call to ", PROCNAME
      LENGTH=72
      MSGBUF=" "
      CALL GERRORMSG (GRAF, MSGBUF, LENGTH, RETURNLEN)
      DISPLAY MSGBUF
      CALL GTERMGRAF (GRAF)
      STOP
      END
```

```
$CONTROL USLINIT
 IDENTIFICATION DIVISION.
***********************************************************
*                                                         *
*                                                         *
*                  COBOL SAMPLE PROGRAM                   *
*                                                         *
*                                                         *
***********************************************************

 PROGRAM-ID.   COBOL-SAMPLE.
 DATE-COMPILED.

 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.  HP3000.
 OBJECT-COMPUTER.  HP3000.

 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77  GPROCEDURE          PIC X(16) VALUE SPACES.

 01  GRAF.
     05  GSTATUS         PIC S9(4) COMP.
     05  PARAM           PIC S9(4) COMP.
     05  ELEMENT         PIC S9(4) COMP.
     05  FSERROR         PIC S9(4) COMP.
     05  FILLER          PIC S9(4) COMP OCCURS 1596 TIMES.

 01  GINITGRAF-PARAMETERS.
     05  GRAF-SIZE       PIC S9(4) COMP.
     05  LANGUAGE        PIC S9(4) COMP.

 01  GOPENFILE-PARAMETERS.
     05  FILE-NAME       PIC X(36).
     05  ACCESS-MODE     PIC S9(4) COMP.

 01  GGETCHART-PARAMETERS.
     05  CHART-NAME      PIC X(16).

 01  GPIESET-PARAMETERS.
     05  PERCENT         PIC S9(4) COMP.
     05  GACTUAL         PIC S9(4) COMP.
     05  COLOR-BUF.
         10  COLOR-1     PIC S9(4) COMP.
         10  COLOR-2     PIC S9(4) COMP.
         10  COLOR-3     PIC S9(4) COMP.
         10  COLOR-4     PIC S9(4) COMP.
     05  COLOR-BUF-LEN   PIC S9(4) COMP.
     05  TEXTURE-BUF.
         10  TEXTURE-1   PIC S9(4) COMP.
         10  TEXTURE-2   PIC S9(4) COMP.
         10  TEXTURE-3   PIC S9(4) COMP.
         10  TEXTURE-4   PIC S9(4) COMP.
         10  TEXTURE-5   PIC S9(4) COMP.
         10  TEXTURE-6   PIC S9(4) COMP.
     05  TEXTURE-BUF-LEN PIC S9(4) COMP.
 01  GDEVICESET-PARAMETERS.
     05  DEVICE              PIC X(6)
     05  HPIB                PIC S9(4) COMP.
     05  COPIES              PIC S9(4) COMP.
     05  PEN-SPEED           PIC S9(4) COMP.
     05  CHAR-QUAL           PIC S9(4) COMP.

 01  GEXECHART-PARAMETERS.
     05  OPTIMIZE            PIC S9(4) COMP.

 01  GERRORMSG-PARAMETERS.
     05  MSG-BUF             PIC X(72)
     05  GLENGTH             PIC S9(4) COMP.
     05  RETURN-LEN          PIC S9(4) COMP.


 PROCEDURE DIVISION.
**********************************************************
*                                                        *
*                MAIN PROGRAM                            *
*                                                        *
**********************************************************

 MAINLINE.

******  Initialize GRAF array; MUST be first call  *****

     MOVE 1600 TO GRAF-SIZE.
     MOVE 0 TO LANGUAGE.
     CALL "GINITGRAF" USING GRAF, GRAF-SIZE, LANGUAGE.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GINITGRAF" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Open existing chart file CF1 in exclusive mode  *****

     MOVE "CF1 " TO FILE-NAME.
     MOVE 1 TO ACCESS-MODE.
     CALL "GOPENFILE" USING GRAF, FILE-NAME, ACCESS-MODE.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GOPENFILE" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Retrieve chart FIRSTCHART  *****

     MOVE "FIRSTCHART" TO CHART-NAME.
     CALL "GGETCHART" USING GRAF, CHART-NAME.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GGETCHART" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Specify 4 colors and 6 textures for pie segments  *****
******  Label percentages, but not actual values          *****

     MOVE 1 TO PERCENT.
     MOVE 0 TO GACTUAL.
```

```
**  Set up COLOR-BUF  **
     MOVE 1 TO COLOR-1.
     MOVE 2 TO COLOR-2.
     MOVE 3 TO COLOR-3.
     MOVE 4 TO COLOR-4.
     MOVE 4 TO COLOR-BUF-LEN.
**  Set up TEXTURE-BUF  **
     MOVE 1 TO TEXTURE-1.
     MOVE 2 TO TEXTURE-2.
     MOVE 3 TO TEXTURE-3.
     MOVE 4 TO TEXTURE-4.
     MOVE 5 TO TEXTURE-5.
     MOVE 6 TO TEXTURE-6.
     MOVE 6 TO TEXTURE-BUF-LEN.
     CALL "GPIESET" USING GRAF, PERCENT, GACTUAL,
                          COLOR-BUF, COLOR-BUF-LEN,
                          TEXTURE-BUF, TEXTURE-BUF-LEN.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GPIESET" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Draw 1 copy of the chart on a      *****
******  9872A plotter at fast pen speed    *****
******  using hardware characters          *****

     MOVE "9872A" TO DEVICE.
     MOVE 5 TO HPIB.
     MOVE 1 TO COPIES.
     MOVE 0 TO PEN-SPEED.
     MOVE 1 TO CHAR-QUAL.
     CALL "GDEVICESET" USING GRAF, DEVICE, HPIB, COPIES,
                             PEN-SPEED, CHAR-QUAL.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GDEVICESET" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Prepare data for graphing  *****

     CALL "GEXECDATA" USING GRAF
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GEXECDATA" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Draw chart, optimizing when possible  *****

     MOVE 1 TO OPTIMIZE.
     CALL "GEXECHART" USING GRAF, OPTIMIZE.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GEXECHART" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.

******  Terminate access to DSG/3000; closes chart file  *****

     CALL "GTERMGRAF" USING GRAF.
     IF GSTATUS IS NOT EQUAL TO 0
        MOVE "GTERMGRAF" TO GPROCEDURE
        PERFORM ERROR-ROUTINE.
     STOP RUN.

**********************************************************
*                                                        *
*                  ERROR ROUTINE                         *
*                                                        *
**********************************************************

 ERROR-ROUTINE.
     DISPLAY "Error returned by call to ", GPROCEDURE.
     MOVE 72 TO GLENGTH.
     MOVE SPACES TO MSG-BUF.
     CALL "GERRORMSG" USING GRAF, MSG-BUF, GLENGTH,
                            RETURN-LEN.
     DISPLAY MSG-BUF.
     CALL "GTERMGRAF" USING GRAF.
     STOP RUN.
```

```
****************** PASCAL SAMPLE PROGRAM *****************
$STANDARD_LEVEL 'HP3000'$  { use HP3000 Pascal Standards }
program testdsg (input, output);

type
   int = -32768..32767; { 16 bit integer }

   str = packed array [1..16] of char;

   graf_type = RECORD
               status   : int;
               parm     : int;
               element  : int;
               fserror  : int;
               graf     : array [1..1596] of int;
               END;

var
   grafarea  : graf_type;

{ GINITGRAF parameters }

   grafsize  : int;
   language  : int;

{ GOPENFILE parameters }

   filename  : packed array [1..36] of char;
   access    : int;

{ GGETCHART parameters }

   chartname : packed array [1..16] of char;

{ GPIESET parameters }

   percent   : int;
   actual    : int;
   colorbuflen   : int;
   texturebuflen : int;
   colorbuf      : array [1..4] of int;
   texturebuf    : array [1..6] of int;

{ GDEVICESET parameters }

   device    : packed array [1..6] of char;
   hpib      : int;
   copies    : int;
   penspeed  : int;
   charqual  : int;

{ GEXECHART parameters }

   optimize  : int;

{ GERRORMSG parameters }

   msgbuf    : packed array [1..72] of char;
   length    : int;
   returnlen : int;

{ global vars }

   i         : int;
   proc_name : str;

{ graphics intrinsics }

procedure GINITGRAF;
   intrinsic;
procedure GOPENFILE;
   intrinsic;
procedure GGETCHART;
   intrinsic;
procedure GEXECHART;
   intrinsic;
procedure GEXECDATA;
   intrinsic;
procedure GERRORMSG;
   intrinsic;
procedure GPIESET;
   intrinsic;
procedure GTERMGRAF;
   intrinsic;
procedure GDEVICESET;
   intrinsic;
procedure GEXECFIGURE;
   intrinsic;
procedure GDELETEFIGURE;
   intrinsic;
procedure GTEXTCONTROLSET;
   intrinsic;
procedure GTEXTCONTROLINQ;
   intrinsic;
procedure GFONTSET;
   intrinsic;
procedure GFONTINQ;
   intrinsic;
procedure TERMINATE;
   intrinsic;

procedure paserror (name : str);

begin
   length := 72;

   GERRORMSG (grafarea, msgbuf, length, returnlen);

   writeln ('ERROR IN ', name);
   writeln ( msgbuf);

   GTERMGRAF (grafarea);
   TERMINATE;

end;
```

```
    begin { Main Program }

       {
       { initialize GRAFAREA record
       }

       grafsize := 1600;
       language := 5;

       proc_name := 'GINITGRAF';

       GINITGRAF (grafarea, grafsize, language);
       if grafarea.status <> 0 then
          paserror (proc_name);

       {
       { Open chart file T06T250A in exclusive mode
       }

       filename := 'T06T250A';
       access := 1;

       proc_name := 'GOPENFILE';

       GOPENFILE (grafarea, filename, access);
       if grafarea.status <> 0 then
          paserror (proc_name);

       {
       { Retrieve chart FIRSTCHART
       }

       chartname := 'FIRSTCHART';
       proc_name := 'GGETCHART';

       GGETCHART (grafarea, chartname);
       if grafarea.status <> 0 then
          paserror (proc_name);

       {
       { Specify 4 colors and 6 textures for pie segments
       { Label percentages, but not actual values
       }

       percent := 1;
       actual  := 0;

       { Set up colorbuf }

       for i := 1 to 4 do
          colorbuf [i] := i;
       colorbuflen := 4;

       {
       { Set up texturebuf
       }
```

```

for i := 1 to 6 do
   texturebuf [i] := i;
texturebuflen := 6;

proc_name := 'GPIESET';

GPIESET (grafarea, percent, actual, colorbuf,
         colorbuflen, texturebuf, texturebuflen);
if grafarea.status <> 0 then
   paserror (proc_name);

{
{ Draw 1 copy of the chart on a 2647a terminal at
{ fast pen speed using hardware characters.
}

device := '2647A';
hpib := 5;
copies := 1;
penspeed := 0;
charqual := 1;

proc_name := 'GDEVICESET';

GDEVICESET (grafarea, device, hpib, copies, penspeed,
            charqual);
if grafarea.status <> 0 then
   paserror (proc_name);

{
{ Prepare data for graphing.
}

proc_name := 'GEXECDATA';

GEXECDATA (grafarea);
if grafarea.status <> 0 then
   paserror (proc_name);

{
{ Draw chart, optimizing when possible.
}

optimize := 1;
proc_name := 'GEXECHART';

GEXECHART (grafarea, optimize);
if grafarea.status <> 0 then
   paserror (proc_name);

{
{ Terminate access to DSG/3000 and close chart file.
}

proc_name := 'GTERMGRAF';

```

```
   GTERMGRAF (grafarea);
   if grafarea.status <> 0 then
      paserror (proc_name);

end. { Main Program }
```

ERROR MESSAGES — APPENDIX C

NOTE: THE ! IN THE FOLLOWING ERROR MESSAGES WILL APPEAR AS A NUMBER OR TEXTUAL VARIABLE IN THE ACTUAL MESSAGES.

(Warning messages follow the error messages and are indicated by negative numbers.)

Error Message	Meaning	Action
GINITGRAF		
1. Logon device could not be opened. (FSERR!)	The terminal could not be opened.	Refer to the file system error number for more information and possible action.
2. Could not find out information about logon device. (FSERR !)	The terminal did not respond.	Contact system manager and refer to MPE file system error number.
3. Request for primary terminal status failed (FSERR !)	Could not write the primary status request to terminal.	Refer to MPE file system error number and contact system manager.
4. Could not read primary status (FSERR !)	FREAD failed. Could possibly be a data overrun or timeout.	Try again. If unsuccessful, sta refer to MPE'S error number and contact system manager.
5. Request for secondary terminal status failed. (FSERR !)	Could not write the secondary status request to terminal.	Refer to MPE file system error number and contact system manager.
6. Could not read secondary terminal status. (FSERR !)	FREAD failed. Could possibly be a data overrun or timeout.	Try again. If unsuccessful, refer to MPE file system error number and contact system manager.
7. Terminal read failed. (FSERR !)	FREAD failed. Could possibly be a data overrun or timeout.	Try again. If unsuccessful, refer to MPE file system error number and contact system manager.
9. Failed to determine what kind of terminal user has.	The terminal status did not identify a HP DSG/3000 supported terminal.	Refer to reference manual for list of supported terminals or contact system manager.
10 Request for graphics terminal identifer failed. (FSERR !)	Could not write the graphics ID request to the terminal.	Refer to MPE file system error number and contact system manager.
11 Could not read graphics terminal identifier. (FSERR !)	FREAD failed.	Refer to MPE file system errr number and contact system manager.
12 GRAF AREA must be at least ! words long when called from BASIC.	The communications area is too small.	Expand the grafsize parameter.

Error Message	Meaning	Action
13 GRAF AREA too short. Must be at least ! words long.	Same as number 12.	Same as number 12.
14 LANGUAGE must be an integer in range 0 to 3.	The language parameter must be an integer from 0 to 3.	Change the language parameter.
15 Additional stack space not available.	Required memory resources are not available.	Exit the program and rerun when necessary resources are available.
16 INTERNAL ERROR: illegal value passed to DLSIZE.	Internal Error.	Contact your system manager.
17 Requested DLSIZE exceeded maximum limit allowed.	Not enough room in stack for the requested DLSIZE.	PREP your program with larger MAXDATA.
18 INTERNAL ERROR: AGL GINIT failed.	Internal error accessing graphics device.	See your system manager.
19 Could not close graphics device. (FSERR !)	Graphics device couldn't be closed.	Refer to MPE file system error. Re-run program. If problems, contact system manager.
GCREATEFILE		
20 Chart file must be large enough to contain at least one chart.	Numcharts parameter must be greater than or equal to 1.	Change the numcharts parameter
21 DOMAIN parameter must be either 0, 1, or 2.	Domain parameter must be 0, 1, or 2.	Change the domain parameter.
23 Could not create a new chart file (FSERR !)	The file system could not create the chart file.	Check the file system error number for more information.
24 Could not get information about the new chart file. (FSERR !)	The file system could not access the file.	Same as number 23.
25 Could not create a new chart file. (FSERR !)	The file system could not create the chart file.	Check the file system error number for more information.

Error Message	Meaning	Action
26 Could not write to the graphics device. (FSERR !)	Graphics device may not be connected or turned on.	Check to see if graphic device is connected correctly and is turned on.
27 Convert chart file to new format using GUPDATE utility.	Chart file format has been changed in the newest version of DSG/3000.	Run GUPDATE program to convert old chart file to new chart file format so it can be run on the new version.
GCLOSEFILE		
30 No chart file is open.	There is no current chart file.	No action is necessary.
31 Record update failed. (FSERR !)	Record update failed due to a file system error.	Refer to the file system error number for more information and possible action.
32 Add record failed. (FSERR !)	Add record failed due to a file system error.	Same as number 31.
33 Error trying to find a given record by key. (FSERR !)	Find record failed due to a file system error.	Same as number 31.
34 Failed to read a record from the chart file. (FSERR !)	Read record failed due to a file system error.	Same as number 31.
37 Temporary chart file could not be closed and purged. (FSERR !)	File close failed due to a file system error.	Same as number 31.
GOPENFILE		
50 ACCESS parameter must have a value of 0, 1, or 2.	There is an error in the access parameter.	Change the access parameter to a 0, 1, or 2.
51 Chart file not specified or could not be opened. (FSERR !)	File open failed due to a file system error.	Refer to the file system error number for more informaton and possible action.
52 File is not a chart file.	Specified file was not created by GCREATEFILE.	Check the file name and file code.

Error Message	Meaning	Action
53 Chart file needs stack space of ! words. Could not expand stack.	Required memory resources are not available.	Exit the program and rerun when necessary resources are available.
54 Could not get the name of the previous chart file. (FSERR !)	Product internal error - graf area may have been overridden.	Check your code to see if it is over-writing the graf area.
55 Could not inquire information from the chart file. (FSERR !)	Could not read the chart file due to a file system error.	Same as number 51.
56 Could not expand stack to size of ! words.	Required memory resources are not available.	Exit the program and rerun when necessary resources are available.
57 INTERNAL ERROR: Could not expand stack to size ! words.	System memory resources are not available at this time.	Rerun when required resources are available
58 Could not expand stack because it was frozen.	System memory resources are not available at this time.	Rerun when required resources are available.
GLISTCHARTS		
60 User cannot request less than 0 chart names.	You requested fewer than 0 chart names.	Check numcharts parameter.
61 Chart name numbering starts with 1.	You are passing a non positive in the firstchart parameter.	Check firstchart parameter.
62 Index of first chart name exceeds current number of charts.	You requested chart names that do not exist in the chart file.	Check firstchart parameter.
GCREATECHART		
70 Chart ! already exists in chart file.	The chart you are trying to create already exists.	Delete the chart before you create
71 Chart name cannot be all blanks.	You tried to specify a chart name that contains all blanks.	Check the chartname parameter.
72 Chart name must begin with a letter.	A non alphabetic character begins the chart name.	Check the chartname parameter.

Error Message	Meaning	Action
73 No embedded blanks are allowed in chart names.	Your chart name contains embedded blanks.	Check the chartname parameter.
74 Chart names can contain only letters, numbers and underbars.	Your chart name contains illegal characters.	Check the chartname parameter.
75 Chart file is full. Cannot add any more charts.	No more space to add new charts.	Delete old charts before adding new ones.
76 AGL GPLOTR failed when terminating a device.	Internal error accessing graphics device.	See your system manager.
77 INTERNAL ERROR: Bad parameter to EXPAND'BLOCK.	Internal error.	Same as number 76.
78 INTERNAL ERROR: Block table is wrong for this record.	Internal error.	Same as number 76.
GGETCHART		
80 Chart ! does not exist in this chart file.	The chartname parameter specifies a chart that is not in the current chart file.	Check the name in the chartname parameter.
GCOMMENTSET AND GCOMMENTINQ		
90 Directory comments area too small. No expansion can be done.	No more room in user chart file to add comments.	You can make room by deleting some of the other charts in the chart file.
91 Chart file must be opened exclusively before it can be changed.	You cannot write comments to the chartfile when the chart file is opened with shared access.	Close the chart file and open it again with exclusive access (call GCLOSEFILE and GOPENFILE.)
92 Length of comments must be a positive value.	You supplied a negative value as the length of your comment string.	Check your code and pass a positive value.

Error Message	Meaning	Action
GCHARTYPESET and GCHARTYPEINQ		
100 INTERNAL ERROR: Could not find directory entry for current chart.	Current chart name does not have an entry in the chart fil directory (bad chart file).	Contact your system manager.
101 Chart type ranges from 1 to 4.	The legal chart types range from 1 to 4.	Check to make sure you are passing a valid chart type.
102 There is no currently active chart.	The sequence of intrinsic calls is in error.	Call GGETCHART or GCREATECHART to establish a currently active chart.
GENERAL MESSAGE		
110 This intrinsic not callable from COBOL. (real values passed.)	The intrinsic that was just called from a COBOL program is not callable from COBOL because the intrinsic requires real number(s) to be passed to it.	Call the equivalent intrinsic which is callable from COBOL because an ASC11 character string (or strings) is used instead of real numbe(s). For example, call GBARASET instead of GBARSET.
GPIESET AND GPIEINQ		
120 PERCENT parameter requires a value of 0 or 1.	The percent parameter does not equal 0 or 1.	Change the percent parameter.
121 ACTUAL parameter requires a value of 0 or 1.	The actual parameter does not equal 0 or 1.	Change the actual parameter.
130 Number of pie segment colors ranges from 0 to 30.	The colorlen parameter is not between 0 and 30. Colorlen indicates the number of items in colorbuf.	Change the colorlen parameter.
131 Number of pie segment textures ranges from 0 to 30.	The texturelen parameter is not between 0 and 30. Texturelen indicates the number of items in texturelen.	Change the texturelen parameter.

Error Message	Meaning	Action
132 Colors range from 0 to 16. Item ! illegal.	The specified item in the colorbuf parameter is not between 0 and 16.	Change the specified color item.
133 Textures range from 0 to 8. Item ! illegal.	The specified item in the texturebuf parameter is not between 0 and 8.	Change the specified texture item.
GLEGENDSET AND GLEGENDINQ		
140 Bars or lines range between 1 and 8.	You tried to specify more than 8 lines or bars.	Change the depvarid parameter.
141 Colors range from 0 to 16.	The color parameter is not between 0 to 16.	Change the color parameter.
142 Textures range from 0 to 8.	The texture parameter is not between 0 and 8.	Change the texture parameter to 0-4, or -1 for the default.
143 Markers range from 0 to 8.	The marker parameter is not between 0 and 8.	Change the marker parameter to 0-8, or -1 for the default.
144 Non-printable character was found in the legend's first line.	The legend1 parameter contains an unprintable character, such as escape or tab.	Check the contents of the legend1 parameter.
145 Non-printable character was found in the legend's second line.	The legend2 parameter contains an unprintable character, such as an escape or tab.	Check the contents of the legend2 parameter.
GTITLESET AND GTITLEINQ		
150 Length of string is less than 0.	The titlelen parameter is less than 0.	Change the titelen parameter.
151 Only a stick font (1) is available now.	The font parameter must equal 1.	Change the font parameter.
152 Length of title cannot exceed 45 characters.	The titlelen parameter is greater that 45.	Change the titlelen parameter.
153 Size of subtitle can be only big (1) or small (0).	The size parameter is not 0 or 1.	Change the size parameter.

Error Message	Meaning	Action
154 Length of subtitle cannot exceed 64 characters.	The titlelen parameter is greater than 64.	Change the titlelen parameter.
155 Length of second line of title cannot exceed 45 characters.	The titlelen parameter is greater than 45.	Same as number 154.
156 Length of footnote cannot exceed 90 characters.	The titlelen parameter is greater than 90.	Same as number 154.
157 Length of second footnote cannot exceed 90 characters.	The titlelen parameter is greater than 90.	Same as number 154.
158 Length of right axis title cannot exceed 90 characters.	The titlelen parameter is greater than 90.	Same as number 154.
159 Length of X axis title cannot exceed 90 characters.	The titlelen parameter is greater than 90.	Same as number 154.
160 Length of Y axis title cannot exceed 90 characters.	The titlelen parameter is greater than 90.	Same as number 154.
161 Title, subtitle etc. referred to by numbers from 1 through 7.	The option parameter must have a value between 1 and 7 inclusive.	Change the option parameter.
162 Length of user's buffer must be greater than 0.	The titlelen parameter is less than or equal to zero.	Change the titlelen parameter.
GTEXTCONTROLSET and GTEXTCONTROLINQ		
175 Title, subtitle, etc. referred to by numbers 1 through 11.	The option parameter indicating which text entity will be affected must be in the range 1-11 action.	Set option parameter to the code number of the entity you want to change.
GTERMGRAF		
180 AGL GPLOTR failed with terminate function.	Internal error on a graphing device during termination.	Abort the program.

Error Message	Meaning	Action
181 AGL GEND failed.	Internal error on a graphing device during termination.	Abort the program.
GBARSET AND GBARINQ		
190 GROUPING parameter must have a value of 0 to 2.	The grouping parameter is not between 0 and 2.	Change the grouping parameter.
191 DEFAULTS parameter must have a value of 0 to 2.	The defaults parameter is not between 0 and 2.	Change the defaults parameter.
192 ORIENTATION parameter must have a value of 0 to 2.	The orientation parameter is not between 0 and 2.	Change the orientation parameter.
193 A maximum of 60 bars are allowed.	More than 60 bars were specified in the numintevals parameter.	Change the numintevals parameter.
194 Interval between grid lines is too large.	The gridspacing parameter is larger than the numintervals parameter.	Change either the gridspacing parameter or the numintervals parameter.
GAXISSET AND GAXISINQ . . . and GSUPPRESSSET		
195 Logarithmic interval must be less than 32767.	The intervalsize parameter is greater than or equal to 32767 for a logarithmic chart.	Change the intervalzise parameter.
197 Logarithmic interval size must be an integer.	The intervalsize parameter is not an integer for a logarithmic chart.	Change the intervalsize parameter.
198 Bar chart minimum must be less than or equal to 0.	The min parameter is greater than 0.	Change the min parameter.
199 Bar chart maximum must be greater than or equal 0.	The max parameter is negative.	Change the max parameter.
200 Use 1 for independent variable axis, 2 for dependent axis.	The axis parameter is not 1 or 2.	Change the axis parameter.

Error Message	Meaning	Action
201 DEFAULTS parameter must have some value from 0 to 4.	The defaults parameter is not between 0 and 4, inclusive.	Change the defaults parameter.
202 NOTICKS parameter must be either 1 or 0.	The noticks parameter does not contain a 0 or 1.	Change the noticks parameter.
203 Maximum number of minor tick marks per major tick marks =10.	The minorticks parameter is greater than 10.	Change the minorticks parameter.
204 SCALETYPE parameter must have a value from 0 to 2.	The scaletype parameter has a value other than 0,1, or 2.	Change the scaletype parameter.
205 REFLECT parameter must have a value of 0 or 1.	The reflect parameter must be either 0 or 1.	Change the reflect parameter.
206 Minimum must be less than maximum.	The value of the min parameter must be less than the value max parameter.	Change either the min or max parameter.
207 MINORGRIDS parameter must be either 0 or 1.	The minorgrids parameter does not contain a 0 or 1.	Change the minorgrids parameter.
208 Logarithmic minimum must be greater than 0.	The min parameter is negative or 0 on a logarithmic chart.	Change the min parameter. Logarithms of negative numbers or 0 are meaningless. The value 1 is a a good minimum;the logarithm of 1 is zero.
209 Logarithmic maximum must be greater than 0.	The max parameter is negative or 0 on a logarithmic chart.	Change the max parameter. Logarithms of negative numbers or 0 are meaningless.
GSUPPRESSET		
210 CHART suppress parameter must be either 0 or 1.	The specified chart parameter has a value other than 0 or 1.	Change the chart parameter.
211 LABELS suppress parameter must be either 0 or 1.	The specified labels parameter has a value other than 0 or 1.	Change the labels parameter.

Error Message	Meaning	Action
212 AXES suppress parameter must be either 0 or 1.	The specified axes parameter has a value other than 0 or 1.	Change the axes parameter.
213 LEGENDS suppress parameter must be either 0 or 1.	The specified legends parameter has a value other than 0 or 1.	Change the legends parameter.
214 TITLES suppress parameter must be either 0 or 1.	The secified titles parameter has a value other than 0 or 1.	Change the titles parameter.
215 FRAME suppress parameter must be either 0 or 1.	The specified frame parameter has a value other than 0 or 1.	Change the frame parameter.
216 DATA suppress parameter must be either 0 or 1.	The specified data parameter has a value other than 0 or 1.	Change the data parameter.
217 ADVANCE suppress parameter must be either 0 or 1.	The specified advance parameter has a value other than 0 or 1.	Change the advance parameter.
GAXISSET		
218 The minimum for stacked bar charts must equal 0.	The min parameter is not 0 for a stacked bar chart.	Change the min parameter to 0. Stacked bar charts add data values to a bar which starts at a certain base value; this base value must be zero.
GDEVICESET		
220 COPIES must be >=0.	The specified copies parameter is negative.	Change the copies parameter to 0 or a positive number.
221 PENSPEED must be either fast (0) or slow (1).	The penseed parameter has a value other than 0 or 1.	Change the penspeed parameter.
222 Character type may have a value of 1 or 3.	The chartype parameter has a value other than 1 or 3.	Change the chartype parameter.
223 HPIB address must fall between 0 and 31.	Valid HPIB addresses are from 0 to 31.	Change the HPIB parameter.

Error Message	Meaning	Action
224 DSG/3000 does not support such a device.	The <u>device</u> parameter specifies a device not supported by DSG/3000.	Change the <u>device</u> parameter.
225 "FIGURE" cannot be specified VIA GDEVICESET.	"FIGURE" is not a valid graphics device as a parameter to GDEVICESET.	To draw to a figure file, use the GEXECFIGURE intrinsic.
GLIMITSET		
230 LIMITS parameter may have a value of 0 to 4.	The <u>limits</u> parameter has a value other than 0-4.	Change the <u>limits</u> parameter.
GLABELSET AND GLABELINQ		
235 Type of labels may range from 0 to 3.	The <u>type</u> parameter has a value other than 0 to 3.	Change the <u>type</u> parameter.
236 Length of label must be > 0.	The <u>labelen</u> parameter has a negative value.	Change the <u>labelen</u> parameter.
237 Number of labels in buffer must be > 0.	The <u>numlabels</u> parameter has a negative value.	Change the <u>numlabels</u> parameter.
239 Unprintable character in labels buffer.	The <u>labelbuf</u> parameter contains an unprintable character such as escape or tab.	Check the contents of the labels buffer.
240 No room for labels. Stack could not be expanded.	Required memory resources are not available.	Exit and rerun program when necessary resources are available.
241 AXIS parameter ranges from 1 to 3.	<u>Axis</u> parameter has a value other than 1 2, or 3.	Change the axis parameter.
243 Length of labels buffer must be > o.	The <u>labelen</u> parameter of GLABELINQ must have a positive value.	Change the <u>labelen</u> parameter.
244 Only the X axis variable can be labeled from the data.	The <u>axis</u> parameter has a value 2 and the type parameter has a value 1.	Change the typeor <u>axis</u> parameter.

Error Message	Meaning	Action
245 One label cannot exceed 25 characters in length.	The labelen parameter is greater than 25.	Change the labelen parameter.
ANNOTATIONS ERRORS		
250 Annotation identifiers must be >=0.	No negative numbers are allowed for annotation identifiers.	Change the annotid parameter to a positive number.
251 Legal symbol types range from 1 to 3.	Symbol parameter must be either 1 2, or 3.	Change the symbol parameter.
252 First X coordinate for annotation must be >=0.	You cannot use negative numbers as coodinates.	Change the X1 paraeter.
253 First Y coordinate for annotation must be >=0.	Same as number 252.	Change the Y1 parameter.
254 Second X coordinate for annotation must be > =0.	Same as number 252.	Change the X2 parameter
255 Second Y coordinate for annotation must be >=0.	Same as number 252.	Change the Y2 parameter.
256 Linetypes range from 0 to 8.	The linetype parameter is wrong.	Change the linetype parameter to be an integer from 0 to 8, or -1 to accept the default.
257 No room in primary annotation table.	No more annotations can be added.	Delete one of the existing annotations.
258 Annotation already exists.	A duplicate annotation identifier has been specified.	Change the annotid parameter or delete the annotation with the matching identifier.
259 No such annotation exists.	The annotation identifier does not exist in the current chart.	Change the annotid parameter. Call GANOTINFO for more information.
260 Use GTEXTINQ to look at annotations of type TEXT.	The annotation you are inquiring about is textual.	Call GTEXTINQ.

Error Message	Meaning	Action
261 Cursor position out of graphing area.	The cursor was out of the currently defined graphics area when the cursor was read, i.e., when the key was pressed to enter the cursor location.	Try again, but make sure the cursor falls within the previously defined graphing area (see GLIMITSET and GLIMITINQ).
262 Error occurred in setting up the plotting environment.	The graphics output device failed to return the plotting environment for the annotation.	Make sure that the device is configured properly and set up for plotting.
263 Error occurred in drawing the current annotation.	The current annotation could not be completed.	Check the parameters of the current annotation.
GTEXTSET AND GTEXTINQ		
265 LORG parameter must hold value ranging from 1 to 9.	The label origin is specified by an integer from 1 to 9.	Change the lorg parameter.
266 ANGLE ranges from 0 to 359 degrees.	The angle parameter must be an integer from 0 to 359.	Change the angle parameter.
267 Character size ranges from 1 to 50.	The charsize parameter must be an integer from 1 to 72.	Change the charsize parameter.
268 Length of buffer holding annotation text must be >0.	You passed a value for textbuflen that was 0 or negative.	Change the textbuflen parameter to a positive integer.
269 Unprintable character occurred in the annotation text.	The textbuf parameter contains an unprintable character, such as an escape or tab.	Check the contents of the text buffer.
270 Only the stick character font (1) is allowed.	The font parameter must be equal to 1.	Change the font parameter.
272 The length of textual annotations is limited to 100.	The textlen parameter is greater than 100.	Change the textlen parameter. A long(> 100 character) annotation may need to be divided into 2 or more annotations.

Error Message	Meaning	Action
GANOTINFO		
275 Index into the annotations must be > =1.	The firstannot parameter must be an integer greater than or equal to 1.	Change the firstannot parameter.
276 Length of user's buffers must be >0.	The maxannots parameter must be greater than 0.	Change the maxannots parameter.
277 Secondary annotation table full. Stack could not be specified.	No more textual annotations can be entered.	Delete one or more textual annotations.
278 The index into the annotations exceeds the number of annotations.	The firstannot parameter is greater than the total number of annotations for this chart.	Change the firstannot parameter.
CONVERSION ERRORS		
280 ASCII real number was all blanks.	A value of an ASCII string was all blanks.	Change the string parameter to contain at least one digit.
281 ASCII real number must contain digits as well as a sign.	A value of an ASCII string contained no digits.	Change the string parameter to contain at least one digit.
282 ASCII real number cannot start with a comma.	A specified value of an ASCII string starts with a comma.	Change the string parameter so that it does not start with a comma.
283 Commas must be spaced 3 digits apart.	A specified ASCII string has misplaced commas.	Change the spacing of commas to be 3 digits apart.
284 The exponent is missing although the "E" appears in the number.	A specified ASC11 string contains an E but is not followed by an exponent.	Put the value of the exponent after the E, or delete the E.
285 The exponent cannot have more than one sign.	A specified ASCII string has more than one sign in the exponent.	Delete the extra sign(s).
286 An exponent sign appears although no "E" is given.	The sign of the exponent is not preceded by an E in an ASCII string.	Put an E before the sign of the exponent.

Error Message	Meaning	Action
287 The exponent sign cannot trail the exponent.	The sign must precede the value of the exponent in an ASCII string.	Put the sign before the value of the exponent.
288 Only one decimal point is allowed.	An ASCII string has more than one decimal point.	Delete the extra decimal point(s).
289 Only one "E" is allowed.	An ASCII string has more than one E.	Delete the extra E.
290 No mantissa or exponent was given.	An ASCII string is missing either a mantissa or exponent; i.e., no digits appear either before the "E" or after the "E".	Add digits or delete the "E", as desired.
291 A blank was embedded in the real number.	The real number contained a blank.	Remove the blank.
292 Only 0-9, +,-,., "E",and/or commas are allowed in real numbers.	A specified ASC11 string contains illegal values.	Change the value of the string parameter.
293 Mantissa or exponent is missing.	An ASC11 string is missing either a mantissa or an exponent. No digits appear either before the "E" or after the "E".	Add digits or delete the "E", as desired.
294 EXTIN' failed.	Internal error on conversion.	Contact your system manager.
295 INTERNAL ERROR: Buffer too short to return E-type number.	Internal error on conversion.	Contact your system manager.
296 INTERNAL ERROR: Buffer too short to return free format.	Same as number 295.	Same as number 295.
297 Number too big to convert and contains illegal character.	The ASCII string contains a number larger than the largest possible real number and at least 1 illegal character.	Change the ASCII string to represent a legal value.

Error Message	Meaning	Action
298 Number too small to convert and contains illegal character.	The ASCII string contains a number smaller than the smallest possible real number and at least 1 illegal character.	Change the ASCII string to represent a legal value.
299 Number too big to convert to real.	The number in the ASCII string is larger than the largest possible real number.	Change the ASCII number to represent a legal value.
300 Number too small to convert to real.	The number in the ASCII string is smaller than the smallest possible real number.	Change the ASCII number to represent a legal value.
GCHARTINFO		
310 INTERNAL ERROR: Converting CHRONOS to JULIAN.	Internal date conversion error.	Contact your system manager.
311 INTERNAL ERROR: Converting JULIAN to TEXT.	Internal date conversion error.	Contact your system manager.
GCOPYCHART		
320 FROM FILE name is not a legal file name.	The fromfile parameter does not meet the MPE restrictions for a legal file name.	Change the fromfile parameter to a legal MPE file name. It should also be the name of an existing chart file.
321 TO FILE name is not a legal file name.	The tofile parameter does not meet the MPE restrictions for a legal file name.	Change the tofile parameter to a legal MPE file name.
322 No such chart exists in the FROM FILE.	The fromchart parameter names a chart which does not exist in the file designated by the fromfile parameter.	Check the chart file using the browse option of the GRAPH program. Usually, either the chart name is mispelled or the wrong chart file file was specified. Change the fromfile or fromchart parameter as appropriate.

Error Message	Meaning	Action
323 INTERNAL ERROR: Could not find char ! after it was copied.	Chart file directory error after the copy completed.	Try again using a different destination chart file.
INTERNAL FILE ERRORS		
340 A file already exists with this name.	A file of this name already exists.	Purge file and rerun.
341 INTERNAL ERROR: File access failed. (KKA ERROR !)	Internal error.	Contact your system manger.
342 INTERNAL ERROR: Failed to close chart file properly. (FSERR !)	Internal error.	Contact your system manager.
343 INTERNAL ERROR: Chart file was not closed properly.	Internal error.	Contact your system manager.
344 INTERNAL ERROR: Update failed due to file access problem.	Same as number 343.	Same as number 343.
345 Chart file is full. Cannot add any more charts (KKA error.)	Same as number 343.	Same as number 343.
346 INTERNAL ERROR: Retrieve failed due to file access problem.	Same as number 343.	Same as number 343.
GEXECHART		
400 OPTIMIZE parameter must hold 0 or 1 for chart !	The optimize parameter is not set to 0 or 1.	Set optimize to 1 for optimize or to 0 for no optimization.
401 No data file is open for chart (!). Call GEXECDATA.	Gexecdata was not called to execute all data specifications.	Call gexecdata before calling gexechart.

Error Message	Meaning	Action
402 Could not read from data file for chart (!). (FSERR !)	Internal software error, not under user control.	See file system error number for error information. Contact your system manager.
405 INTERNAL ERROR FOR CHART !: No such chart type.	Internal software error, not under user control.	Contact your system manager.
409 Failed to open figure file!. (KKA ERROR !)	Figure file is busy or is not present.	List your files to see if the particular file is present; if it is, check to see if it is being used elsewhere.
410 Failed to open graphics terminal for chart!.	Graphics terminal cannot be opened or is not pesent.	Check the specified device to ensure it matches the current terminal If not, rerun using the current configuration.
411 Chart ! requires the use of a graphics terminal.	Hit draw on non-graphics terminal.	Use a 2647 or 2648 terminal.
412 Failed to open graphics device for chart!.	The graphics device associated with the indicated chart either cannot be opened or is not connected correctly.	Check the specified device to ensure it matches the current device. Reconnect the device and rerun if necessary. Otherwise contact your system managr.
413 AGL GCLEVEL failed for chart !.	Internal error with graphics device, not under user control.	Contact your system manager.
415 Could not open graphics device for chart!. (FSERR!)	Graphics terminal cannot be opened or is not present.	If the specified device matches the current configuration, contact your system manager.
416 Could not read from graphics device for !. (FSERR !)	Internal error with graphics device.	Contact your system manager.
420 AGL GCLR failed for chart !.	Internal error with graphics device, not under user control.	Contact your system manager.
425 INTERNAL ERROR FOR CHART ! : No such label option.	Internal software error, not under user control.	Contact your system manager.

Error Message	Meaning	Action
430 Lower left corner must be lower than the upper right corner.	The lower left coordinates of the plotting area are higher than upper right coordinates.	Respecify coordinates to place lower left below and to left of upper right.
432 AGL GPAPERSIZE failed for chart !.	Internal error with graphics device,	Contact your system manager.
435 AGL GDMAP stretch failed	Internal error with graphics device, not unde user control.	Contact your system manager.
436 AGL GDMAP square failed for chart !.	Same as number 435.	Same as number 435.
437 INTERNAL ERROR FOR CHART ! : No such chart format.	Same as number 435.	Same as number 435.
440 Pie chart ! cannot be positioned by its data boundary.	The data plotting area for a pie chart cannot be changed.	Change the chart plotting area for the pie chart.
441 The number of pie segments exceeds the maximum allowed (!).	Data exists for more pie segments than can be drawn.	Reduce the number of segments by modifying the data file or using data subsetting.
442 Pie segments less than 2 degrees are not allowed.	One or more data values would cause a segment of less than 2 degrees to be drawn, i.e., a data value is less than .6% of the total for all the data values.	Remove the small data values by modifying the data file or using data subsetting.
450 AGL GVIEWP failed for chart!.	Internal error with graphics device, not under user control.	Contact your system manager.
451 AGL GWINDOW failed for chart !.	Same as number 450.	Same as number 450.
452 AGL convert to NDC's failed for chart !.	Same as number 450.	Same as number 450.
454 AGL GLORG failed for chart !.	Same as number 450.	Same as number 450.

Error Message	Meaning	Action
455 AGL GLDIR failed for chart !.	Same as number 450.	Same as number 450.
456 AGL GCELLSIZE failed for chart!.	Same as number 450.	Same as number 450.
457 AGL GMOVE failed for chart!.	Same as number 450.	Same as number 450.
458 AGL GTEXT failed for chart !.	Same as number 450.	Same as number 450.
459 AGL GPEN failed for chart !.	Internal error with graphics device, not under user control.	Contact your system manager,
460 AGL GLSTYLE failed for chart!.	Same as number 459.	Same as number 459.
462 AGL GXMIT failed for chart!.	Same as number 459.	Same as number 459.
467 AGL GCFONT failed for chart!.	Incorrect font number specified.	Font number must be between 1-4.
468 AGL GFILLPOCY failed for chart!.	Graphics package has not been initialized.	Contact your system manager.
469 AGL GDEFFONT failed for chart !.	One or more font attributes is is invalid.	Check parameters for GFONTSET, and make sure font file exists.
470 Floating point error occurred. Data cannot be graphed.	A floating point error occurred while drawing the chart.	Check the data file, transformations qualifications, and subsetting for possible errors.
471 Overflow on bar scaling due to interval size based on data.	Internal error caused by scaling.	Change the interval size, minimum, and maximum scaling parameters; insure all data falls between the bounds specified.
472 Bars cannot be drawn because too many intervals are present.	Too many intervals present probably due to the intervalsize parameter on GAXISSET. The intervals are so small, bars cannot be drawn within them.	Check interval size, minimum and maximum values; insure that the number of evenly-spaced intervals is a reasonable number, i.e., 1 to 40.

Error Message	Meaning	Action
473 AGL GPLOTR failed to to initialize the 2680.	Same as number 450.	Same as number 450.
480 Error occurred when plotting to the 2680 (FSERR!	A file system error occurred while plotting to the 2680. No raster image is produced.	Look up the file system error and take the appropriate action.
481 IFS/3000 error occurred when plotting to the 2680.(IFSERR!)	Same as number 450	Same as 450.
482 File system error occurred on OUT2680A.	Same as number 480	Same as number 480.
483 CREATEPROCESS error occurred when plotting to the 2680 (ERRNUM!)	Same as number 450	Same as number 450.
490 Font file format for font ! is bad for chart !	Specified file is not a valid DSG/3000 font file.	Use valid DSG/3000 font files.
492 Font file cannot be opened for chart!.	Font file is unavailabe.	Check the file name, make sure it exists and is the proper file type.
404 No font file for font! available for chart!. No best fit.	The font file could not be found or opened for font number N, and the best fit font file used for finding best fit was also unavailable.	The best fit font file should be be provided during DSG/3000 installation. Contact your system manager.
500 Could not draw frame for chart !.	Same as number 459.	Same as number 459.
501 Could not draw axes for chart !.	Same as number 459.	Same as number 459
502 Could not fill an area for chart !.	Same as number 45	Same as number 459
505 Could not draw a box for the legend for chart !.	Same as number 459.	Same as number 459

Error Message	Meaning	Action
506 INTERNAL ERROR FOR CHART! : Too many legends for this format.	Same as number 459.	Same as number 459.
509 Maximum value may not equal zero for stacked bar charts.	The maximum scale on a stacked bar chart is set to 0 or less, probably due to the max parameter with GAXISSET.	Change the maximum for the axis to a positive value greater than the data to be plotted. Remember that stacked bars add data to a base value of 0.
510 INTERNAL ERROR FOR CHART ! : No such scaling option possible.	Same as number 459.	Same as number 459.
511 INTERNAL ERROR FOR CHART !: No such scaling option possible.	Same as number 459.	Same as number 459.
512 INTERNAL ERROR FOR CHART !: No such axis exists.	Same as 459.	Same as number 459.
513 Minimum must be less than maximum for scaling chart!.	The minimum scaling value is greater than the maximum scaling value.	Reset the minimum or maximum to have minimum less than maximum.
514 At least one scaling interval must be requested for chart!.	The default number of intervals is less than or equal zero.	Change the axis scaling by either calling GAXIS[A]SET using the GRAPHICS OPTIONS Menu interactively.
515 Minimum must be less than maximum for scaling chart!:	Min is greater than max.	Make max greater than min.
517 Stacked bar chart ! Cannot have data points below 0.	Negative data points cannot be accurately displayed by a stacked bar chart.	Use a clustered bar chart instead.
518 Negative or 0 data not acceptable for log values on chart!.	The log of negative numbers cannot be calculated.	Use ABS function in transformations to compute absolute values.

Error Message	Meaning	Action
519 Stacked bar charts not allowed with logarithmic scaling.	Stacked bar charts cannot be accurately displayed using log scaling.	Use clustered bar chart.
530 Failed to pick up a pen for chart !.	Internal graphics device error, not under user control.	Contact your system manager.
531 Could not draw bar between minimum and maximum for chart!.	Axis scaling did not permit entire bar segment to be drawn.	Set min lower, and max higher, or let them both default to data.
532 Negative data is not acceptable for stack bar chart!.	Negative data points cannot be accurately displayed by a stacked bar chart.	Use clustered bar chart.
533 Could not draw bar between minimum and maximum for chart!.	Axis scaling did not permit entire bar segment to be drawn.	Set min lower, or max higher, or let them both default to data.
534 Could not draw the zero line for cluster bar chart!.	Zero line for bar chart which falls above and below axis could not be drawn due to an internal device error.	Contact your system manager.
535 Could not draw grid lines between horizontal bar for chart!	Grid line(s) failed the specified chart due to an internal device error.	Contact your system manager.
536 Could not draw bar between minimum and maximum for chart!.	Axis scaling did not permit entire bar segment to be drawn.	Set min lower, or max higher, or let them both default to data.
537 Could not draw bar between minimum and maximum for chart!.	Axis scaling does not permit entire bar to be drawn.	Set min lower, max higher, or let them both default to data.
540 Could not draw vertical grid lines for chart!.	Grid lines for the specified chart failed due to an internal device error.	Contact your system manager.

Error Message	Meaning	Action
541 Could not draw minor tick marks for chart!.	The minor tick marks or grid lines failed due to an internal device error.	Same as number 540.
542 Could not draw major tick marks or grid lines for chart!.	The major tick marks or grid lines failed due to an internal device error.	Same as number 540.
LABELING		
550 Only the bar axis can be labeled from the data for chart!.	Line charts may not use data in data file for tick labels.	Make a bar chart instead.
551 Real number free format conversion failed for chart!.	Internal error in conversion routines.	Use different axis scaling.
552 INTERNAL ERROR FOR CHART!: Non-fraction too big to convert.	Internal error on conversion.	Contact your system manager.
553 Real number exponential conversion failed for chart!.	Internal conversion error of data to a label.	Contact your system manager.
554 Real number too big to fit into 6 character label for for chart!.	Real data value is too large for the 6-character label.	Change label definition to not use data values.
560 Could not move to first point for line chart!.	AGL internal error.	Contact your system manager.
561 Could not draw a line segment for line chart!.	AGL internal error.	Contact your system manager.
562 Call to AGL routine GMARKER for chart!.	AGL internal error.	Contact your system manager.
563 Could not draw line for legend for chart!.	AGL internal error.	Contact your system manager.

Error Message	Meaning	Action
570 Could not position or draw a segment label for chart!.	AGL internal error.	Contact your system manager.
571 Could not convert a percentage into ASC11 for labeling chart!.	Internal error in conversion routines.	Contact your system manager.
572 Data point could not be converted to label for chart!.	Internal error in conversion routines.	Contact your system manager.
573 Numeric labels are not used for pie chart!.	Labels for pie charts must be type text.	Change the variable to be used as a label.
574 Data points too small and/or numerous for pie chart!.	The data contained values all of which converted to segments of less than 1 degree in size.	Change the data or use a different chart type.
575 No data points exist for chart!.	No data is in the data file.	Enter the desired data into the data file.
576 Could not draw a pie segment for chart!.	Internal error accessing graphics device.	Contact your system manager.
578 Negative values present on pie chart!; cannot be drawn.	Pie charts cannot be used to display negative values.	Use ABS function in transformations to calculate absolute value.
579 There is no data>0 for pie chart!.	The data file is empty.	Enter data into the data file.
580 Call to GDIGIT did not return an annotation point for chart!.	AGL internal error.	Contact your system manager.
581 Could not draw a box annotation for chart!.	Same as number 580.	Same as number 580
582 Could not draw a line annotation for chart!.	Same as number 580.	Same as number 580.

Error Message	Meaning	Action
583 Could not draw a text annotation for chart!.	Same as number 580.	Same as number 580.
584 Could not draw an arrow annotation for chart!.	Same as number 580.	Same as number 580.
GFONTSET		
620 Language may range from 0 to 7.	The language parameter has a value other than 0-7.	Change the language parameter.
621 Slant must equal 0.	Slant option is not supported at this time.	Always use 0 for slant parameter.
622 Character format must equal 0.	Extended character format is not supported at this time.	Always use 0 for character format.
750 Switch parameter must be 0 or 1.	Internal error.	Call system manager.
760 AGL GALPHACURSON failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
761 AGL GALPHACURSOFF failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
762 AGL GALPHADISPON failed	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
763 AGL GALPHADISPOFF failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
764 AGL GGRAPHCURSON failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
765 AGL GGRAPHCURSOFF failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
766 AGL GGRAPHDISPON failed.	Problem with the graphics terminal.	Verify that graphics terminalis working properly.

Error Message	Meaning	Action
767AGL GGRAPHDISPOFF failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
768 AGL GKBLOCK failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
769 AGL GKBUNLOCK failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
770 AGL GMCURS failed.	Problem with the graphics terminal.	Verify that graphics terminal is working properly.
GTRANSINQ, GDEFNINQ		
806 Invalid start index. Must be larger than zero.	The firstdefn or firstrans parameter is negative or zero.	Change firstrans or firstdefn to be a positive number greater than zero.
807 Invalid retrieval maximum. Must be positive.	Numdefn or numtrans parameter is negative or zero.	Change numdefn or numtrans parameter to be greater than zero.
GSORTSET		
820 Sort value must be 0 for OFF or 1 for ON.	Sortflag parameter is not 0 or 1.	Set sortflag to 0 or 1.
GDATAFILESET		
830 Missing data value length must be in the range of ! to !	Misslen parameter is negative.	Set misslen to be in range 0 to 16.
831 Missing data value must be numeric.	The missbuf parameter contains an ASC11 value that cannot be converted to a real number.	Change missbuf to be a numeric value.
832 Data file name incorrectly specified.	FILENAME does not follow MPE data file naming conventions or a blank doesn't follow the name.	See MPE naming conventions and put a blank following the name.

Error Message	Meaning	Action
GDEFNSET		
840 Invalid number of definitions.	Numdefn parameter is negative.	Reset numdefn to a positive value between 0 and 9.
841 Invalid data format specified.	Format parameter is not 1 or 2.	Set format to 1 or 2.
842 Invalid data type specified.	Typebuf parameter is not 1 or 2.	Set typebuf to 1 or 2.
843 Column numbers must be in the range of ! to !.	Column numbers are less than 1 or greater than 100.	Respecify columns to be in range of 1 to 100.
844 Offsets must be in the range of ! to !.	Fixed format offsets are less than 0 or greater than 2000.	Set offsets to be in range of 1 to 2000.
845 Lengths must be in the range of ! to !.	Fixed format lengths are less than 1 or greater than 45.	Set lengths to be in range of 1 to 45.
846 Too many variable definitions.	Numdefn parameter is greater than 9.	Set numdefn to a value between 0 and 9.
GTRANSET		
860 Invalid number of transformations.	The numtrans parameter is negative.	Set numtrans to be in the range of 0 to 10.
861 Invalid transformation expression length.	Exprlen parameter is negative.	Set exprlen to be greater than or equal to zero.
862 Too many variable transformations.	The numtrans parameter is greater than 10.	Set numtrans to be in range of 0 to 10.
GDEPVARSET		
870 Invalid number of Y axis variables.	Numvars parameter is negative or greater than 8.	Set numvars to a value in the range of 0 to 8.
871 Invalid qualification length.	The qualen parameter is negative.	Set qualen to a positive number or zero.

Error Message	Meaning	Action
872 Qualifications are too large.	Insufficient room in chart file to hold number of specified qualifications.	Reduce size of qualifications: Shorten variable names, remove embedded blanks, shorten qualifications so that qualen can be shorter.
GSUBSET		
880 Subset expression has invalid length.	Subsetlen parameter is negative.	Set subsetlen to a number in the range of 0 to 100.
881 Subset expression is too large.	Subsetlen parameter is greater than 100 bytes.	Set subsetlen to a number in the range of 0 to 100.
VARIABLE SEMANTIC ERRORS		
1000 Duplicate variable name.	More than 1 variable with the same name.	Change variable names until each is unique.
1001 Duplicate found in transformed variable list.	Defined a variable with the same name as a transformed variable.	Change variable name to be unique from transformed variable names.
1002 Variables may not be redefined.	Defining a transformed variable with the same name as a variable.	Change transformed variable name to be unique from variable names.
VARIABLE SYNTAX ERRORS		
1010 A variable name is required here.	Variable name is all blanks.	Specify a variable name.
1011 Variable names can contain only alphas, digits, "-" 's, and "_" 's.	Variable name contains characters other than ALPHAS, digits, "-" 's, or "_" 's.	Remove illegal character(s) from variable name.
1012 Variable names cannot have embedded blanks.	Variable name contains embedded blanks.	Remove embedded blanks from variable name.
1013 A variable name must start with an alphabetic character.	Variable name doesn't start with an alphabetic character.	Change variable name to begin with an alphabetic character.
1014 Incorrectly specified variable name.	Variable name doesn't follow naming conventions.	Specify variable name to conform to syntax specified in Section 3.

Error Message	Meaning	Action
1015 Variable names cannot be reserved words.	Variable name is one of the following: AND, OR, NOT, LOG, ABS, CUMULATE, LN, MOD, MOVEAVG or SQRT	Change variable name to be unique from reserved words.
EXPRESSION ERRORS		
1101 Expression not completed.	SUBSET, QUALIFICATIONS, or TRANSFORMATIONS expression requires more information to be complete.	Change expression to conform to syntax rules specified in section 3.
1102 Expression can't be blank.	TRANSFORMATIONS exprssion is unspecified.	Specify a transformation expression or remove transformation variable names.
1103 A left parenthesis "(" cannot follow "!".	Left parenthesis is out of context.	See section 3 of reference manual for correct syntax.
1104 A right parentheses ")" cannot follow "!".	Right parenthesis is out of syntax context.	See section 3 of reference manual for correct syntax.
1105 A minus sign "-" cannot follow "!".	Minus sign is out of syntax context.	See section 3 of reference manual for correct syntax.
1106 Exponent operator "**" cannot follow "!".	Exponent operator is out of syntax context.	See section 3 of reference manual for correct syntax.
1107 Multiply operator "**" cannot follow "!".	Multiply operator is out of syntax context.	See section 3 of reference manual for correct syntax.
1108 Divide operator "/" cannot follow "!.	Divide operator is out of syntax context.	See section 3 of reference manual. for syntax rules.
1109 Plus operator "+" cannot follow "!".	Plus operator is out of syntax context.	See section 3 of reference manual for syntax rules.
1110 "!" is not a usable number.	The number is less than -1.15792E-77 or greater than 1.15792E+77.	Change number to be in range of -1.15792E-77 to +1.15792E+77.
1111 "!" cannot be used here.	An illegal character was specified.	Remove illegal character and see section 3 for legal specifications.

Error Message	Meaning	Action
1112 A comma "," cannot follow "!".	Comma is out of syntax context.	See section 3 for syntax rules.
1113 "!" cannot be a function parameter.	The parameter doesn't follow conventions for ABS, SQRT, LOG, LN, CUMULATE, MOVEAVG.	Change parameter to comply with function format.
1114 Expression too complex.	The expression has too many operators or parentheses.	Reduce number of parentheses or define additional variables to simplify arithmetic expression.
1115 Invalid (non-ASC11) character found.	Specified non-printable character.	Specify printable character.
1116 Internal error - parser stack underflow.	Software internal error-not under user control.	See your system manager.
1117 "!" is an undefined variable.	Variable not defined by GDEFNSET OR GTRANSET or Data Definition or Transformations Menus.	Define variable.
1118 A name cannot follow "!".	Variable or function name out of context.	Correct expression. See section 3 of reference manual.
1119 Invalid logical expression.	Software internal error, not under user control.	See your system manager.
1120 Logical operator cannot follow "!".	Logical operator out of context.	Change expression to conform to syntax rules in section 3.
1121 Relational operator cannot follow "!".	Relational operator out of syntax context.	See syntax rules in section 3.
1122 String cannot follow "!".	String is out of syntax context.	See syntax rules in section 3.
1123 Number of sign (+,-) cannot follow "!".	Number or sign is out of syntax context.	See syntax rules in section 3.
1124 Unclosed string cannot be used.	Close " or' not found.	Insert " or ' to terminate string.

Error Message	Meaning	Action
1125 MOD operator not used correctly.	Mod operator out of context.	Specify Mod operator just like/ operator.
1130 Missing left parentheses.	Number of right parentheses exceeds number of left parentheses.	Balance parentheses.
1131 Missing right parentheses.	Number of left parentheses exceeds number of right parentheses.	Balance parentheses.
1132 Missing comma.	Parameters of function call are not separated by a comma.	Separate parameters by a comma, check calling sequence.
1133 Logical operator can only be applied to logical	Logical operator applied to operand which cannot be reduced to valid relation. NAME=JOE OR SUE.	Change operand to a relation, i.e., NAME=RICH OR NAME=SUE.
1134 Numerical operator can only be applied to numerical operands,	Numerical operator applied to a string, textual variable, logical operand.	Change operand to be numeric.
1135 Text operator can only be applied to text operands.	Text operator applied to a non-text operand. DIVISION=4.	Change operand to be textual.
1140 Could not figure out your expression the way its written.	Expression syntax not decipherable.	Conform expression to syntax rules in section 3 of reference manual.
1141 Internal error-bad token received from scanner.	Internal software error, not under user control.	Contact your system manager.
1142 Internal error-GENERATE found bad operator in opstack.	Internal software error, not under user control	Contact your system manager.
1143 Internal error-PARSER.	Internal software error, not under user control.	Contact your system manager.
1171 Internal error-IO buffer too small.	Internal software error, not under user control.	Contact your system manager.

Error Message	Meaning	Action
1172 Data file not specified or could not be opened.(FSERR!)	Data file doesn't exist or for some other reason could not be opened.	Refer to the file system error number for more information and possible action.
1173 Record size or fields inconsistent with definition (FSERR!)	Internal software error, not under user control.	Data file definition specified in GDEFNSET does not match actual data file.
1174 Internal error-SDBUILD failed (FSERR!)	File space not available for data manipulation.	Refer to the file system error number for more information and possible action.
1175 Internal error-SDMODIFY failed (FSERR!)	File space not available for data manipulation.	Refer to the file system number for more information and possible action
1176 Internal error-FAST'READ failed (FSERR!)	Input/output failure.	Refer to the file system number for more information and possible action.
1177 Internal error-FAST'WRITE failed (FSERR!)	Input/output failure.	Refer to the file system number for more information and possible action.
1178 Data file contains unusable numeric value.	Data file contains a non-numeric value.	Correct value to be: digits, with optional signs, exponents and "." in the correct order. Remove any non-displayable control characters; these may be found by looking at the file with the display functions option of the terminal active.
1179 Data file contains unclosed string.	Data contains a string with only 1 quotation mark around it.	Close the quotes in the data file.
1180 Internal error-Could not prepare SD file for I/O.	Internal error accessing the data file; not under user control.	Contact your system manager.
1181 Internal error-SDFILEINFO failed.	Internal error accessing the data file; not under user control.	Contact your system manager.

Error Message	Meaning	Action
1182 Internal error-SDGETALLOCDESC failed.	Internal error accessing the data file; not under user control.	Contact your system manager.
1183 Internal error-SLOW'WRITE failed.	Internal error accessing the data file; not under user control.	Contact your system manager.
1184 Internal error-FCLOSE failed.	Internal error accessing the data file; not under user control.	Contact your system manager.
1185 Internal error-FOPEN failed.	Internal error accessing the data file; not under user control.	Contact your system manager.
1186 Internal error-FGETINFO failed.	Internal error accessing the data file; not under user control.	Contact your system manager.
1187 Internal error-FPOINT failed.	Internal error accessing the data file; not under user control.	Contact your system manager.
1195 Internal error-parser failed(!).	Internal software error, not under user control.	Contact your system manager.
1197 Internal error-MPESORT failed.	Internal software error, not under user control.	Contact your system manager.
1200 <,>,=,>=,<=, or<> has operands of inconsistent types.	The variables on either side of the operand are not of the same type.	Change the variables used so that either both sides are numerical or both sides are textual.
1201 Logical operator (AND,OR, or NOT) used incorrectly.	A logical operator was used incorrectly.	Make sure the logical operators AND and OR have 2 operands and the logical operator NOT has 1 operand.
1202 Internal error-LOG'EXEC failed an internal check.	Internal error not under user control.	Contact your system manager.
1203 Transformation expression has an undefined variable.	Transformation expression contains a variable which was not previously defined.	Change variable to a name previously defined or define variable.

Error Message	Meaning	Action
1204 File subset expression or or qualification has an undefined variable.	Subset expression contains a variable name which hasn't been previously defined.	Define variable name.
1205 Pie explosion expression has an undefined variable.	Pie explosion expression contains a variable name not previously defined.	Define variable name.
1206 The independent (or X axis) variable is not defined.	The independent variable for the chart hasn't been previously defined.	Define varible name.
1207 Label variable is not defined.	The label variable for the pie chart hasn't been previously defined.	Define variable name.
1208 Line or bar Y axis variable is undefined.	The dependent variable hasn't been previously defined.	Define variable name.
ARITHMETIC EXPRESSION RUN TIME ERRORS		
1220 Invalid type used in transformation expression.	A textual variable was specified in transformation expression.	Redefine variable type to be numeric or remove variable from expression.
1221 Special functions not implemented yet.	Certain function may be implemented in future release.	Don't use non-implemented special functions.
1222 Internal error-arithmetic result not at top of stack.	Internal software error, not under user control.	Contact your system manager.
1223 Moving average over too many values. Make n smaller.	For MOVEAVG (var,n); n is too large.	Make n smaller.
1224 Moving average specifies undefined variable.	For MOVEAVG (var,n); var is not a defined variable.	Define var.
1225 First parameter of MOVE-AVG must be a variable	The first parameter for the moveavg function is not a variable name.	Specify a variable name as first parameter.
1226 Second parameter of MOVE-AVG must be a positive	The second parameter for the moveavg function is not a positive integer.	Specify a positive integer as second parameter.

Error Message	Meaning	Action
1227 The parameter for CUMULATE must be a variable name.	The parameter for the cumulate function is not a variable name.	Specify a variable name as cumulate parameter.
PASCAL SUPPORT ERRORS		
1300 The PASCAL stack allocation routine (GETHEAP) is missing.	The system library does not contain the necessary PASCAL routine GETHEAP.	Update system library to include the routine GETHEAP.
1301 The PASCAL stack deallocation routine (RTNHEAP) is missing.	The system library does not contain the necessary PASCAL routing RTNHEAP.	Update the system library to include the routine RTNHEAP.
1302 The stack space could not be expanded for the GRAF area.	DSG/3000 can't allocate required stack space.	Reduce user program stack space.
FILE NAME LEGALITY ERRORS		
1401 The filename length is incorrect.	Illegal file name.	Refer to MPE naming conventions.
1402 Filename can't be all blanks.	Illegal file name.	Refer to MPE naming conventions.
1403 Backword references can't contain lockwords.	Illegal file name.	Refer to MPE naming conventions.
1404 Backword references can't contain group names.	Illegal file name.	Refer to MPE naming conventions.
1405 Backword references can't contain account names.	Illegal file name.	Refer to MPE naming conventions.
1406 Filename must begin with a letter.	Illegal file name.	Refer to MPE naming conventions.
1407 Lockword must begin with a letter.	Illegal filename.	Refer to MPE naming conventions.
1408 Group name must begin with a letter.	Illegal filename.	Refer to MPE namining conventions.

Error Message	Meaning	Action
1409 Account name must begin with a letter.	Illegal filename.	Refer to MPE naming conventions.
1410 Filename must contain 8 characters or less.	Illegal filename.	Refer to MPE naming conventions.
1411 Lockword must contain 8 characters or less.	Illegal filename.	Refer to MPE naming conventions.
1412 Group name must contain 8 characters or less.	Illegal filename.	Refer to MPE naming conventions.
1413 Account name must contain 8 characters or less.	Illegal filename.	Refer to MPE naming conventions.
1414 Filename can't contain embedded blanks.	Illegal filename.	Refer to MPE naming conventions.
1415 Lockword can't contain embedded blanks.	Illegal filename.	Refer to MPE naming conventions.
1416 Group name can't contain embedded blanks.	Illegal filename.	Refer to MPE naming conventions.
1417 Account name can't contain embedded blanks.	Illegal filename	Refer to MPE naming conventions.
1418 Filename must contain only letters and numbers.	Illegal filename.	Refer to MPE naming conventions.
1419 Lockword must contain only letters and numbers.	Illegal filename.	Refer to MPE naming conventions.
1420 Group name must contain only letters and numbers.	Illegal filename.	Refer to MPE naming conventions.
1421 Account name must contain only letters and numbers.	Illegal filename.	Refer to MPE naming conventions.
1422 Expected a backword reference but didn't find one.	Illegal filename.	Refer to MPE naming conventions.

Error Message	Meaning	Action
1423 Expected a lockword but didn't find one.	Illegal filename.	Refer to MPE naming conventions.
1424 Expected a group name but didn't find one.	Illegal filename.	Refer to MPE naming conventions.
1425 Expected an account name but didn't find one.	Illegal filename.	Refer to MPE naming conventions.
1426 One lockword can be included following the filename.	Illegal filename.	Refer to MPE naming conventions.
1427 Only two periods are allowed in a filename.	Illegal filename.	Refer to MPE naming conventions.
1428 Filename can't be longer than 35 characters.	Illegal filename.	Refer to MPE naming conventions.
1429 '$' may not be used here.	Illegal filename.	Refer to MPE naming conventions.
FIGURE FILE ERRORS		
1440 Figure name is illegal.	Specified figure name does not follow MPE naming conventions.	Refer to MPE naming conventions.
1441 Figure ! already exists in the figure file.	Figure already exists with the same name.	Use a unique figure name.
1442 Figure file! does not exist.	User cannot access a non-existent figure file.	Change the name to an existing figure file.
1443 Figure ! does not exist in the figure file.	User cannot access a non-existent figure.	Do not try to access a non-existent figure.
1444 Figure ! is currently in use and cannot be deleted.	Figure is currently being accessed by another user.	Must wait until figure is released by other user.
1450 AGL GBEGINFIG failed for figure !.	Software internal error-not under user control.	See your system manager.

Error Message	Meaning	Action
1451 AGL GENDFIG failed for figure !.	Software internal error-not under user control	See your system manager.
1452 AGL GDELETEFIG failed for figure !.	Software internal error-not under user control.	See your system manager.
1453 Figure file ! is currently in use and cannot be opened.	Figure file is being accessed by another user.	Must wait until other user releases the figure file.
1454 File ! exists but is not a figure file.	A figure file can not be created with the same name as an already	Change the name of the figure file.
LOCALIZATION MESSAGES USED BY INTRINSICS ONLY		
4000 The message file (CO1C250A.PUB.SYS.) cannot be accessed.	The message file is not available in PUB.SYS.	Restore error message file in PUB. SYS as shown in installation instructions. Contact system mgr.
4001 The requested message is not in the message file.	Software internal error-not under user control.	Verify that correct DSG/3000 version was installed.
4020 Internal device tables don't match; new version required.	Software internal error-not under user control.	Verify that correct DSG/3000 version was installed.
4021 Internal device table is bad; need new message catalog.	Software internal error-not under user control.	Verify that correct DSG/3000 version was installed.
GRAPH PROGRAM ERRORS		
5000 Color must be a positive integer (1-16)	User specified a non-positive integer for color.	Specify color as a positive integer (1-16).
5001 Size must be a positive integer (1-50)	User specified a non-positive for size.	Specify size as a positive integer (1-50)
5002 Font must be a positive integer (1-4).	User specified a non-positive integer for font.	Specify font as a positive integer (1-4).

Error Message	Meaning	Action
5003 GENMESSAGE ERROR in GET'COMMAND.	Software internal error-not under user control.	Verify that correct DSG/3000 version was installed.
5004 GENMESSAGE ERROR in IS'GENERIC.	Software internal error-not under user control.	Verify that correct DSG/3000 version was installed.
5005 GENMESSAGE ERROR in GET'GENERIC.	Software internal error-not under user control.	Verify that correct DSG/3000 version was installed.
5006 No figure name specified.	Figure name must not be all blanks.	Supply a valid figure name.
5007 No figure file specified.	Figure file name must not be all blanks.	Supply a valid figure file name.
5008 No chart name specified.	Chart name must not be all blanks.	Supply a valid chart name.
5009 No chart file specified.	Chart file name must not be all blanks.	Supply valid chart file name.
5010 Please enter "A" or "D".	Add and delete are the only valid commands.	Specify "A" for add or "D" for delete.
CONVERSION ERRORS		
15015 Error occurred converting chart!.	Chart was not converted to new format.	See your system managert.
15016 The chart file is newer than the code; update DSG/3000.	A chart file from a system with more recent version of DSG/3000 is being accessed by an older version of DSG/3000.	Install the new version on your system.
15020 Error occurred when purging file ! (CIERR!)	File was not purged.	Verify status of file to be purged.
15021 INTERNAL ERROR: Chart file record longer than buffer.	Software internal error - not under user control.	See your system manager.
15022 Failed to create convert file!. (FSERR!)	File system error occurred when trying to convert file.	Refer to FSERR!

Error Message	Meaning	Action
15023 READ failed on file! (FSERR!)	File system error occurred when trying to convert file	Refer to FSERR!
15024 WRITE failed on file!. (FSERR!)	File system error occurred when trying to convert file.	Refer to FSERR!
15025 Failed to close file!. (FSERR!)	File system error occurred when trying to convert file.	Refer to FSERR!
15026 A name for the convert file name cannot be all blanks.	Convert file name cannot be all blanks.	Refer to MPE naming conventions.
15027 Failed to open file! (FSERR!)	File system error occurred when when tring to convert file.	Refer to FSERR!
15029 Chart file name and convert chart name cannot be the same.		
WARNING MESSAGES		
GCOMMENTINQ		
- 95 WARNING: User's buffer too small to receive comments for chart.	Commentsbuf parameter is not large enough to receive comments for the chart.	Make commentsbuf larger and set commentsbuflen accordingly.
GPIEINQ		
-134 WARNING: User's buffer is too small to receive all colors.	Colorbuf is not large enough to receive all color specifications.	Make colorbuf larger, and set colorbuflen accordingly.
-135 WARNING: User's buffer is too small to receive all textures.	Textbuf is not large enough to receive all textures.	Make texturebuf larger, and set texturebuflen accordingly.

Error Message	Meaning	Action
GTITLEINQ		
-163 WARNING: User's buffer too small to receive complete title.	Titlebuf is not large enough to receive entire title.	Make titlebuf larger, and set titlebuflen accordingly.
-164 WARNING: User's buffer too small to receive complete subtitle.	Same as number 163.	Same as number 163.
-165 WARNING: User's buffer too small to receive complete footnote.	Same as number 163.	Same a number 163.
-166 WARNING: User's buffer too small to receive second footnote.	Same as number 163.	Same as number 163.
-167 WARNING: User's buffer too small to receive right axis title.	Same as number 163.	Same as number 163.
-168 WARNING: User's buffer too small to receive X axis title.	Same as number 163.	Same as number 163.
-169 WARNING: User's buffer too small to receive Y axis title.	Same as number 163.	Same as nmber 163.
GAXISSET		
-196 WARNING: Incompatible minimum, maximum, or interval size.	The specified (or default if none is specified) values for for the minimum, maximum, and interval size do not fit evenly together. For example, the specified minimum and interval size may extend beyond the maximum.	Change one or more of the scaling variables so that (minimum-maximum) /interval size) is an integer value.

Error Message	Meaning	Action
GLABELINQ		
-242 WARNING: User's buffer too short to receive all labels.	Labelbuf is not large enough to receive all labels.	Make labelbuf larger and set labelbuflen accordingly.
GTEXTINQ		
-271 WARNING: User's buffer too short to receive annotation text.	Textbuf is not large enough to receive all annotation text.	Make textbuf larger and set textbuflen accordingly.
GEXECHART		
-414 WARNING: AGL GPENSPEED failed for chart!.	An internal error message.	Contact your system manager.
-463 WARNING: Too many intervals (!) for a readable chart.	Tick marks are too close together.	Increase interval size.
-464 WARNING:No chart has been defined.		
-465 WARNING: Bar interval size too small for a readable chart.	Interval size is so small that bars will be drawn too close together to be very clear.	Set the interval size to a larger value.
-466 WARNING: Bars drawn only for the first 31 intervals.	Data for more than 31 bars is present, but only the first 31 bars, or clusters of bars, can be drawn.	Split the data over multiple charts, or reduce the number of bars to be drawn using data subsetting.
-491 WARNING: Chosen language is not available for font! in chart!	Specified language is not accessible for user font.	Change the language or use a font file which includes the speciied language.
-493 WARNING: Best fit font used for font! in Chart!	Font file is not accessible for user device. The best fit (stick) font was used instead.	Verify status of the named file; use the name of a valid font file.

Error Message	Meaning	Action
-495 WARNING: File! in chart! is not a font file. Best fit used.	Font file is not accessible for user device. The best fit (stick) font was used instead.	Verify status of the named file; use the name of a valid font file.
MISSING DATA		
-577 WARNING: Missing data values were not plotted for chart!.	Missing data values were not used when plotting data.	None.
LOG SCALING		
-600 WARNING: Negative or 0 data on chart! treated as missing.	The data associated with the indicated chart contained non-positive data which does not have a legal logarithm. This data was treated as missing data is treated.	If you are aware the data is negative or 0, ignore this message. Otherwise, change the data.
GDATAFILEINQ, GDEFNINQ, GTRANINQ, GSUBINQ, GDEPVARINQ		
-800 WARNING: User's buffer too small to receive missing data value.	Missbuf is too small to receive the missing data value.	Make missbuf larger and set missbuflen accordingly.
-801 WARNING: User's buffer is too small to receive variable names.	Varbuf is too small to receive the variable names.	Make varbuf larger and set varbuflen accordingly.
-802 WARNING: User's buffer is too small to receive variable definitions.	Varbuf is too small to receive the variable definitions.	Make varbuf larger and set varbuflen accordingly.
-803 WARNING: User's buffer is too small to receive transformations.	Exprbuf is too small to receive transformation expressions.	Make exprbuf larger and set exprbuflen accordingly.
-804 WARNING: User's buffer is too small to receive subset or explosion.	Subsetbuf is too small to receive subset expressions.	Make subsetbuf larger and set subsetbuflen accordingly.

Error Message	Meaning	Action
-805 WARNING: User's buffer is too small to receive qualifications.	Qualbuf is too small to receive the qualification expressions.	Make qualbuf larger and set qualbuflen accordingly.
SD'TRANS		
-865 WARNING: Divide by zero occurred during transformations.	Divide by zero occurred during transformations. The result and subsequent manipulations will be given the missing data value.	Either accept the missing data value result, or change the transformation or data so the divide by zero does not occur.
-866 WARNING: Floating point overflow during transformations.	Floating point overflow occurred during transformations. The result and all subsequent manipulations will be given the missing data value.	Either accept the missing data value results, or change the transformation or data so the floating point overflow does not occur.
GLABVARSET		
-885 WARNING: "!" is not a textual variable.	The label variable specified is a numeric variable.	Change the label variable to a textual variable.
VARIABLE SEMANTIC ERRORS		
-1003 WARNING: "!" is not defined.	The specified variable has not been defined.	Define the specified variable or use a variable which has been previously defined.
-1004 WARNING: "!" is a textual variable.	The specified variable is a textual variable.	Change the variable type to numeric or specify an existing numeric variable.
PASCAL STACK/SUPPORT WARNINGS		
-1310 WARNING: Extended stack space was not deallocated.	User is trying to terminate without without deallocating stack space.	Deallocate stack space. Otherwise the space will be deallocated when the program terminates.

Error Message	Meaning	Action
FIGURE FILE WARNINGS		
-1445 WARNING: Figure ! could not be found.	Figure does not exit in figure file.	Check the figure name and figure file name.

MAXIMUM LIMITS

APPENDIX D

	Interactively	Programmatically
Chart file name	35 characters (fully qualified file name including lockword)	35 characters (fully qualified file name including lockword)
Data file name	35 characters (fully qualified file name including lockword)	35 characters (fully qualified file name including lockword)
Chart name	16 characters	16 characters
Variable name	16 characters	16 characters
Number of charts/ chart file	50	50
Number of defined variables/chart	9	9
Length of chart comments	45 characters	No fixed limit
Length of qualification expression	106 bytes per qualification	860 bytes for the entire chart
Number of qualifications/chart	8	8
Length of subset expression	72 characters	200 characters for the entire chart
Number of subsets/ chart	1	1
Length of transformation expression	57 characters	57 characters

	Interactively	Programmatically
Number of transformed variables/ data file	10	10
Length of explosion expression	72 characters	No fixed limit
Maximum data record length	512 bytes	512 bytes
Maximum data value	1.0E77	
Minimum data value	1.0E-77	
Number of line types	8	8
Number of textures	8	8
Number of colors	Device dependent (1-4)	
BAR AND LINE CHARTS		
Number of vertical axis variables	8	8
Length of main title	45 characters	45 characters
Length of subtitle	64 characters	64 characters

BAR AND LINE CHARTS (Continued)

	Interactively	Programmatically
Left vertical axis title	78 characters	90 characters
Horizontal axis title	78 characters	90 characters
Footnote	78 characters	90 characters
Length of labels	25 characters	25 characters
Number of labels	16	650 characters max.
Legend text	2 lines of 18 characters each	2 lines of 18 characters each
Maximum annotations per chart	20	20
Number of bars	60	60
Number of legends	8	8

PAPER SIZES

A4	210 x 297mm
A3	280 x 420mm
A	8.5 x 11 inches
B	11 x 17 inches
C	17 x 22 inches
D	22 x 34 inches
E	34 x 44 inches

PIE CHARTS

	Interactively	Programmatically
Number of pie segments	30	30
Main title	45 characters	45 characters
Subtitle	64 characters	64 characters
Footnote	2 lines of 78 characters each	2 lines of 90 characters each
Length of labels	25 characters	25 characters

USING FIGURE FILES

APPENDIX E

The purpose of figure files is discussed in Section I of this manual. The precise manner in which figures and figure files are created interactively is described in Sections 4 and 5, where the appropriate menus are covered. Programmatic creation is discussed in Sections 6 and 7.

Figure files have several functions, the foremost among them being that they provide storage for a complete chart -- data file and chart design -which is then accessible by other subsystems, such as HPDRAW and HP2680A (the Laser printing system). Below is a general description of the way in which the figure files are accessed by the HP2680A.

Using Figure Files With the HP2680A

The HP2680A (Laser printer) may be used as a direct output device for plotting a chart. You enter the identifying number on the Graphing Options Menu if you are working interactively, or the GDEVICESET procedure if you are working programmatically.

You will also have occasion to use charts previously prepared and stored in a figure file. These you may wish to merge with text, to be printed by the HP2680A. To accomplish this you would use an element of the Interactive Formatting System, IFS/3000, specifically developed for the Laser printing system.

IFS2680 allows you to call the figure file and define the format of documents to be printed by the Laser system. The procedures required for programmatic use are described in Section 5 of the IFS2680 Reference Guide. If you intend to perform these tasks non-programmatically, you would use the LPS Interpreter.

The LPS Interpreter sends environment files and text files to the HP2680A and executes formatting commands by calling the IFS/3000 programmatic intrinsics. Using the interpreter does not require knowledge of a programming language. Its use is discussed in Section 4 of the IFS2680 Reference Guide.

Use With Other Subsystems

Figures and figure files are also available to users of HPDRAW, HPEASYCHART and TDP/3000. Charts made using HPEASYCHART cannot be enhanced except through the use of DSG/3000.

See the appropriate manuals for information concerning the way figure files may be accessed by the various subsystems.

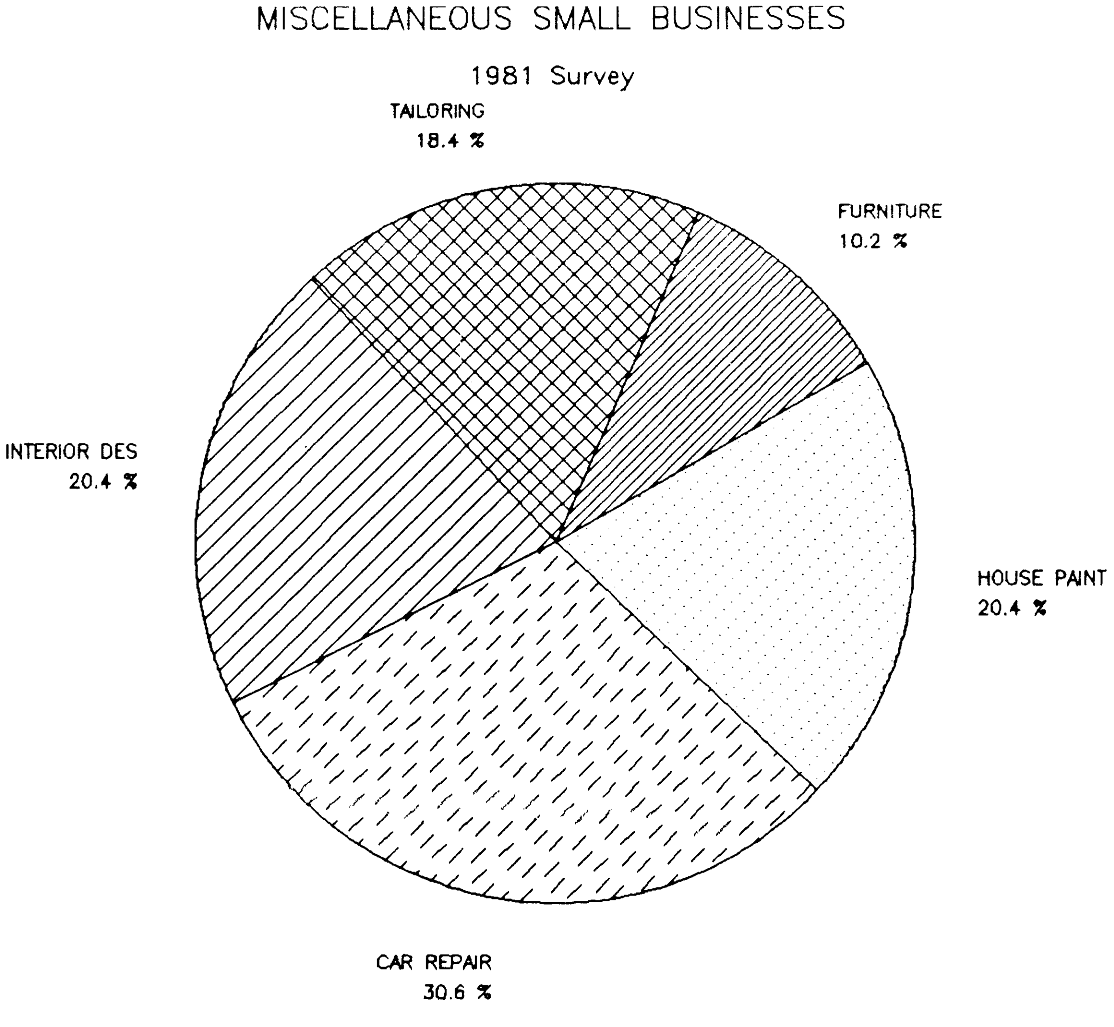
A brief example of the way figure files are used with TDP/3000 follows on the next page.

An excerpt from a report written in TDP/3000 is reproduced below. Explanations the commands appear in italics on the right. The results of this portion of the program is shown on the next page.

```
  1     \envir bzenv                     name of environment file
  2     \lft 5;rht 70                    sets right and left margins
  3     \fontid 1 1                      identifies font to be used
  .                                      from environment
  .
  .
 33     \new                             begins new page
 34
 35
 36
 37     \font 1 1                        font 1 specified for use for line 1
 35     \center 1                        item (the title) to be centered
 36     MISCELLANEOUS                    title
 37
 38
 39
 40
 41
 42     As usual, during the course of our study, we found that
 43     there were some businesses which did not fit in any of
 44     the same gneral categories as the others.  Not knowing
 45     what else to call these, we created a miscellaneous class
 46     of businesses.  These are shown in the following chart.
 47     \name raster rast11             indicates illustration command follows
 48     \illustration figfile:fig 24,C  convert figfile to a raster
                                        and save it under the name rast11.
                                        Then center the figure on the page.
                                        It will take up 24 spaces.
```

MISCELLANEOUS

As usual, during the course of our study, we found that there were some business which did not fit in any of the same general categories as the others. Not knowing what else to call these, we created a miscellaneous class of businesses. These are shown in the following chart.



UPDATING DSG/3000 FILES

APPENDIX F

Files created using an earlier version of DSG/3000 will need to be updated for use with the present version. These would be the versions designated A.00.00, A.00.01, and A.00.02. Files need be converted only once. Two means of updating files are provided. One is an interactive method. To use it, type the following:

```
:RUN GUPDATE.PUB.SYS
```

A banner will appear on the screen, follow by the following instructions:

```
Enter name of chart file to be
converted (return to EXIT):
```

You will enter the old file name after the colon and the following instruction will appear:

```
Enter an unique name for the converted
file to be created:
```

When you enter the two names following the colons, you will receive the following messages:

> Chart file *oldfilename* was converted successfully. The converted file *newfilename* should now be used with DSG/3000.

You can continue to enter names of chart files to be converted if you wish. The first instruction will be repeated:

```
Enter the name of the chart file to be
converted:
```

Press carriage return to exit.

Note that *oldfilename* and *newfilename* cannot be the same. This protects the old file in case of system failure.

newfilename must not already exist as another file.

oldfilename and *newfilename* must follow MPE file naming conventions (may have lockword).

A procedure was developed to convert files programmatically. It is descibed on the next page.

GCONVERTFILE

Converts the contents of a chart file for use with the current version of DSG/3000.

PARAMETERS

graf Integer array containing global information used by DSG/3000. The first word returns the call status.

chartfile Byte array holding name of the old copy of the chart file to be converted. Must be 36 alphanumeric characters (including lockword, group, and account) beginning with an alphanumeric character.

convertfile Byte array holding the name of the converted version of the chart file. This name must 36 alphanumeric characters (including lockword, group, and account) beginning with an alphanumeric character. The file must not exist prior to this call.

DISCUSSION

GCONVERTFILE converts the format of the chart file for use by the new version of DSG/3000. The old chart file is unchanged. The new version is stored in the specified file.

If an error occurs, the converted file may be usable for some charts and other charts may need to be deleted or rebuilt. If the error is such that the entire file is unusable, the chart file will need to be completely rebuilt using the new version of DSG/3000.

Plotters and Terminals

Appendix G

Plotters and Graphics Terminals

On the following pages are some of the plotters and graphics terminals supported by HPDRAW.

Included for each piece of equipment is how to turn it off and on, and how to load pens and paper for plotters. Also included is how to change address codes for HP-IB devices.

HP-IB and Address Codes

Some plotters are ronnected to the HP2647 and HP2648 graphics terminals with an HP-IB cable. Each piece of equipment connected to theseterminals has a unique address code that allows the terminal to recognize responses from that piece of equipment.

HPDRAW recognizes a default address code of 5. If your plotter is not set to 5 you can change the plotter address code, or if you know the address code for the plotter, you can enter the number in the HP-IB field on the Plotting menu. The HP-IB field only appears for Regular and Expert users when the Details key is pressed.

You can have more than one kind of plotter attached to the terminal. In this instance, each plotter should have a unique address code. When you want to use a certain plotter, you can enter the plotter's address code in the HP-IB field.

RS-232-C Connectors

Two types of connectors are available for different terminals: HP-IB and RS-232-C. RS-232-C Connectors are connected to your terminal and do not require HP-IB addresses to be entered. If you have any problems with connectors, see your System Manager.

Plotter Pen Types

Three types of plotter pens are available: fiber tip (2 kinds), roller ball, and drafting. The roller ball and drafting pens are used only for the HP7580A and HP7585A Drafting plotters.

Fiber tip pens are used for all plotters. They are marked with a color matching the pen's ink and a two-character code.

The first character tells the kind of pen: P is for paper, T is for transparency. The second character gives the width of the pen tip in millimeters. For example, P3 is a pen for paper with a 0.3 millimeter tip.

7220 A, C, S, T Plotter

HP7220 Front and Rear Views

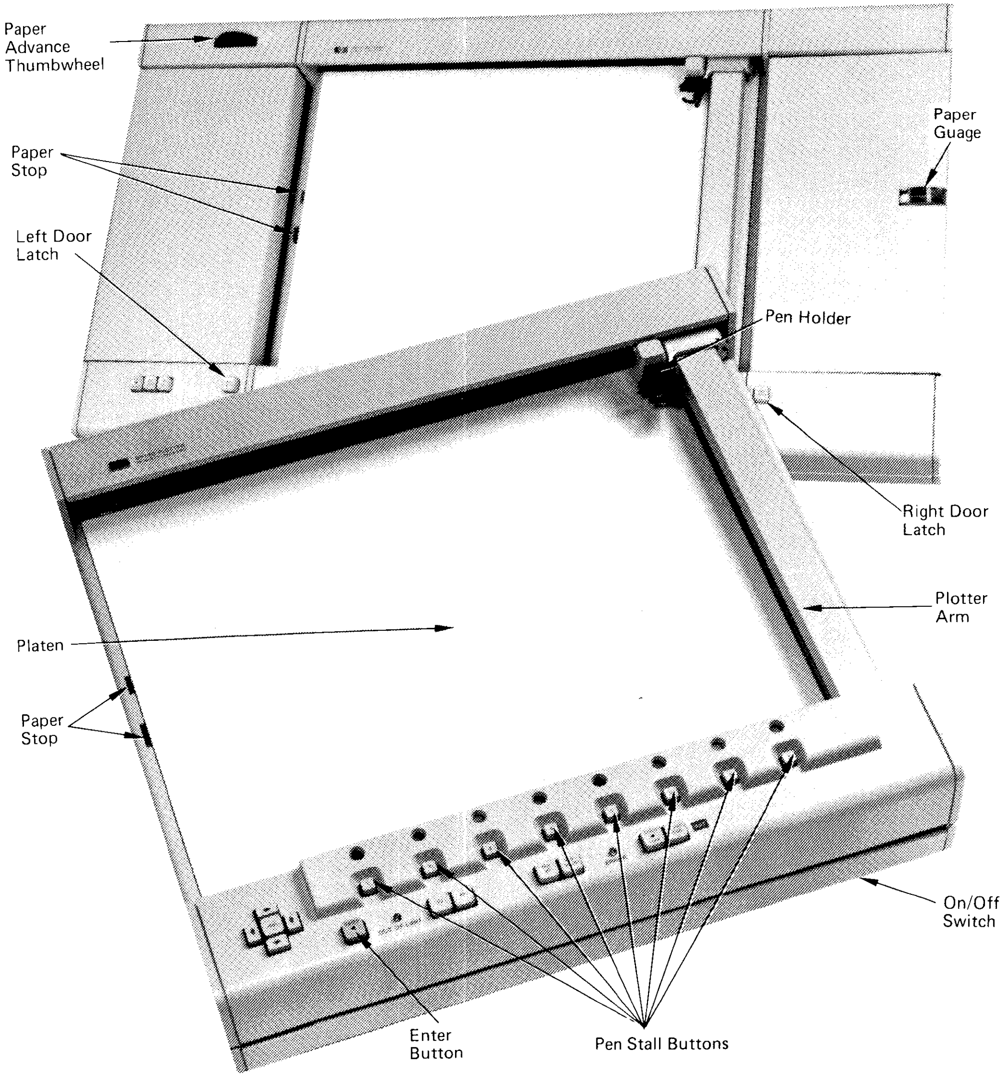


Figure 3-1. Two Front Views 7220C and 7220T

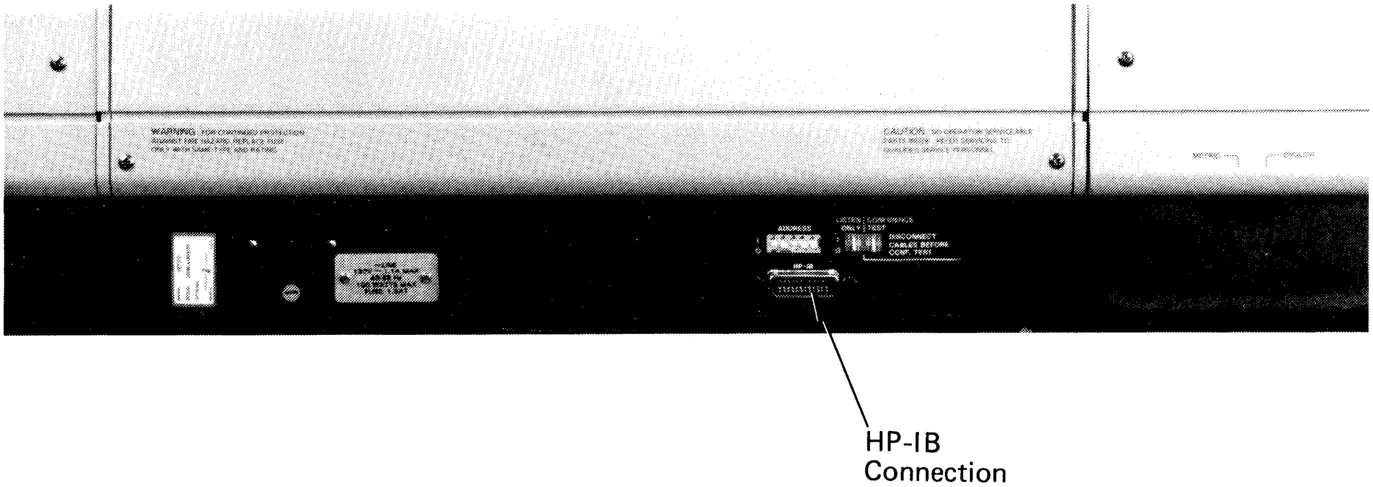


Figure 3-2. Rear View

Power ON/OFF

The on/off switch is located on the lower right front of the plotter.

1. Press the button in to turn the power on. Pressing the button a second time turns the power off.

2. When the switch is on, the pen holder on the plotter arm moves to the lower right corner.

Loading Pens

1. Select a pen and remove its cap.

2. Place the tip of the pen into the rubber cap at the bottom of the stall you want to use. Push the pen in gently until it snaps into place. Repeat this procedure for any other pen stall.

3. To remove pen from the stall, press down on lever to the right of the pen. Grasp the pen firmly between thumb and forefinger and pull gently out of the pen stall.

Loading Paper (7220T Four Pen)

1. Set the on/off switch to off. Remove pen from the pen holder so ink does not get on your paper.

2. Press down on the paper stop on the left side of the platen. Open both the left and the right paper supply modules by pressing the door latch buttons.

3. Position roll paper between hubs in right supply module and align hub tabs with roll notches. Paper should feed across top of roll.

4. Place paper sprocket holes on sprockets at end of roller. Hold paper while closing door. Make sure paper is not on top of paper guide at front edge of platen.

5. Advance paper with thumbwheel until taut.

6. Turn plotter switch to on and press ADV HALF button to advance the paper half a page.

Loading Paper (7220C)

1. Set the on/off switch to on. Raise the paper stop on left side of the platen by pushing the upper portion of the stop with a pencil or pointed object.

2. Press the Chart Load button. The Chart Load and Out of Limit lights should go on and the plotter arm and pen holder is moved to the upper right corner of the platen.

3. Place a sheet of paper on platen with bottom under plastic lip and left side against the paper stop. Press Chart Hold push button. The Chart Load light should go out and Out of Limit light stays on. Smooth paper down using the back of your hand.

For more information on the 7220C/T plotter, refer to the Graphics Plotter Operating and Programming Manual, P/N 07220-90053.

7221A, B, S Plotter

HP7221 Front and Rear Views

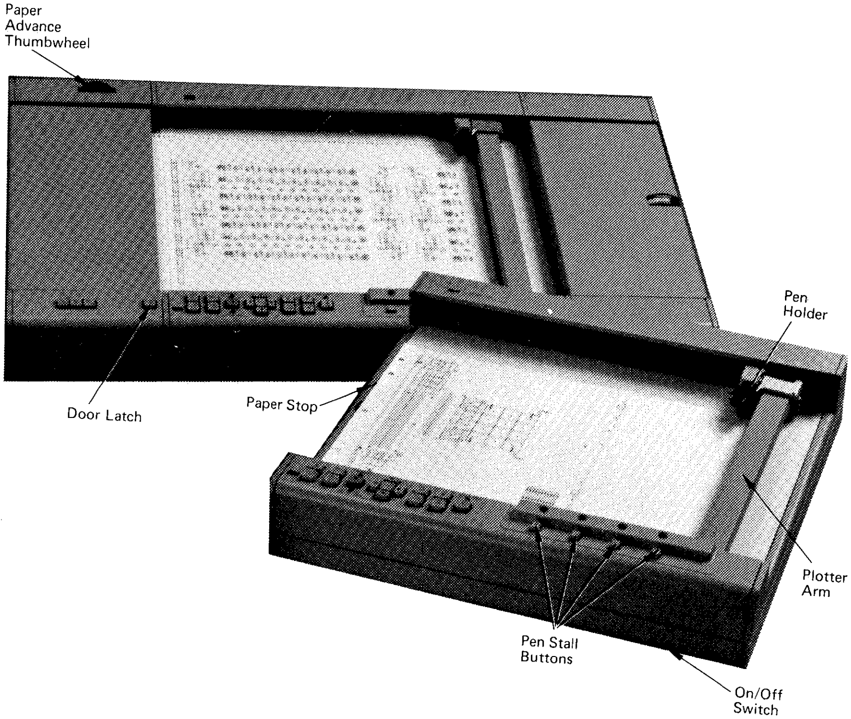


Figure 3-3. Two Front Views 7221C and 7221T

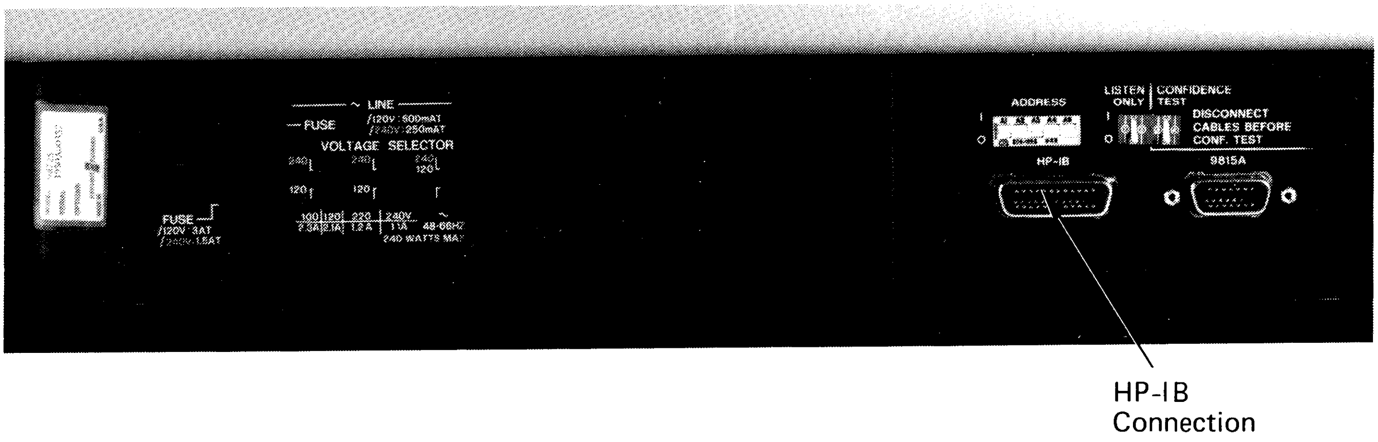


Figure 3-4. Rear View

Power ON/OFF

The on/off switch is located on the lower right front of the plotter.

1. Press the button in to turn the power on. Pressing the button a second time turns the power off.

2. When the switch is on, the following occurs: the On Line light and the Out of Limit light are now on and the fan motor starts running.

Loading Pens

1. Select a pen and remove its cap.

2. Place the tip of the pen into the rubber cap at the bottom of the stall you want to use. Push the pen in gently until it snaps into place. Repeat this procedure for any other pen stall.

3. To remove pen from the stall, press down on lever to the right of the pen. Grasp the pen firmly between thumb and forefinger and pull gently out of the pen stall.

Loading Paper (7221S)

1. Set the on/off switch to off. Remove pen from the pen holder so ink does not get on your paper.

2. Press down on the paper stop on the left side of the platen. Open both left and right paper supply modules by pressing the door latch buttons.

3. Position roll paper between hubs in right supply module and align hub tabs with roll notches. Paper should feed across top of roll.

4. Place paper sprocket holes on sprockets at end of roller. Hold paper while closing door. Make sure paper is not on top of paper guide at front edge of platen.

5. Advance paper with thumbwheels until taut.

6. Turn plotter switch to on, and press ADV HALF button to advance the paper half a page.

Loading Paper (7221A, B)

1. Set the on/off switch to on. Raise the paper stop on left side of the platen by pushing the upper portion of the stop with a pencil or pointed object.

2. Press the Chart Load button. The Chart Load and Out of Limit lights should go on and the plotter arm and pen holder is moved to the upper right corner of the platen.

3. Place a sheet of paper on platen with bottom under plastic lip and left side against the paper stop. Press Chart Hold push button. The Chart Load light should go out and Out of Limit light stays on. Smooth paper down using back of your hand.

For more information on the 7221A, B, S plotter, refer to the Graphics Plotter Operating and Programming Manual, P/N 07221-90014.

7221C, T Plotter

HP7221 Front and Rear Views

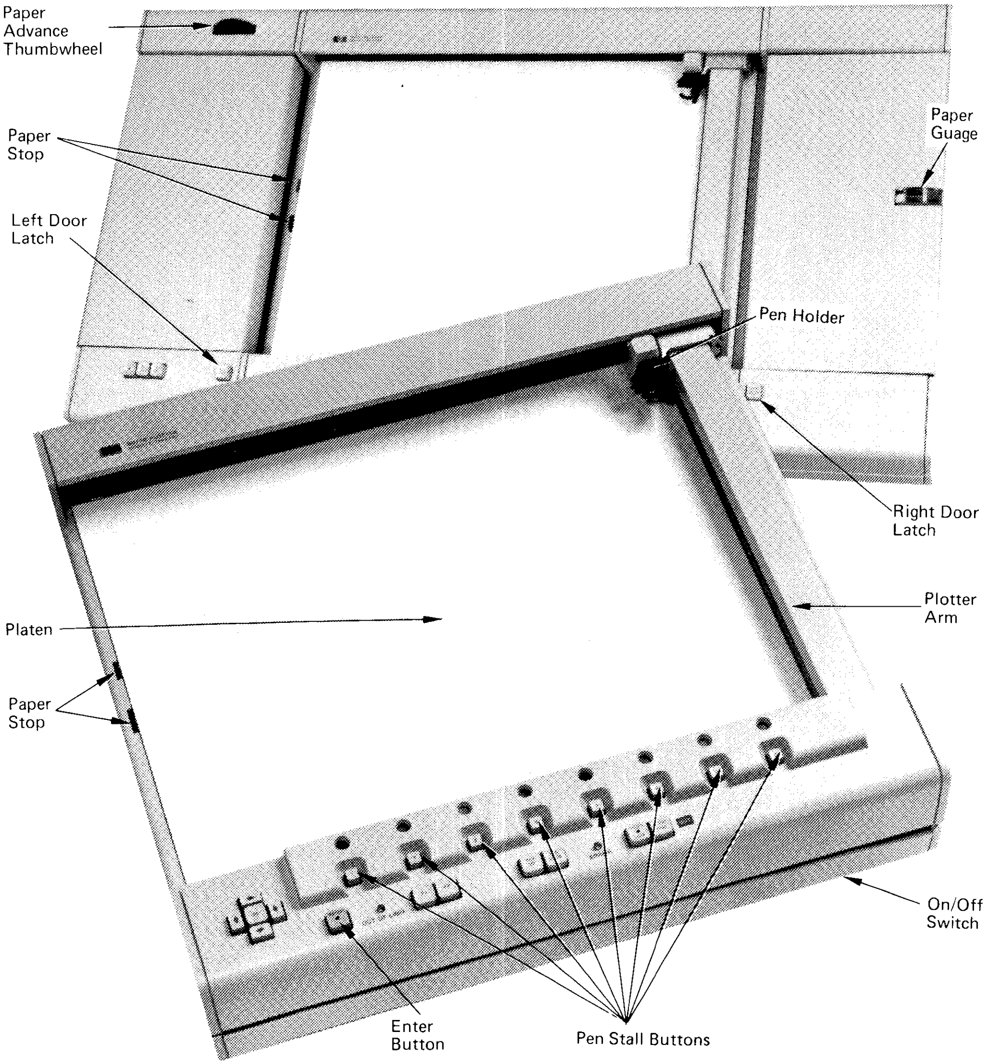


Figure 3-5. Two Front Views 7221C and 7221T

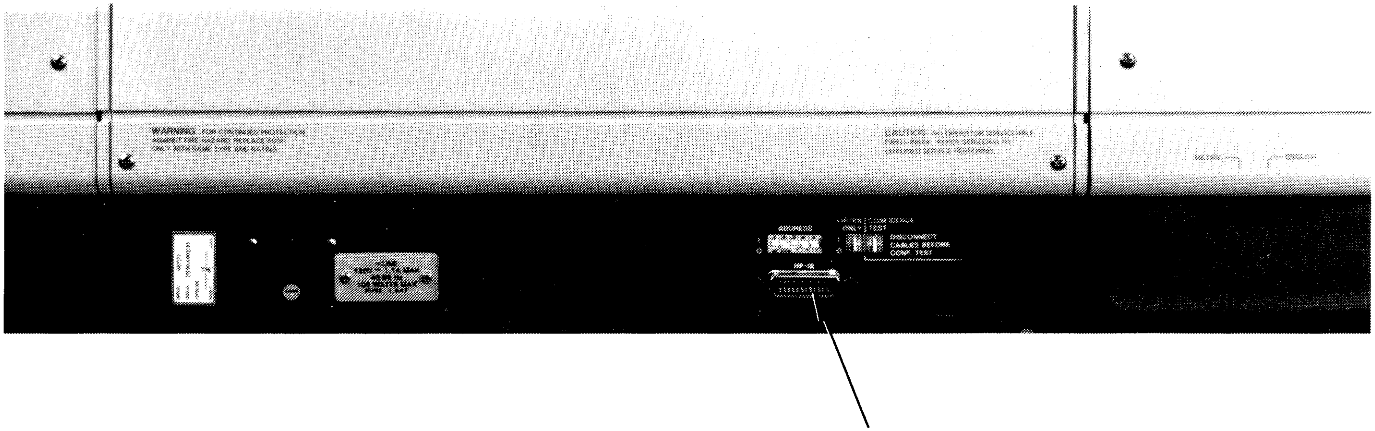


Figure 3-6. Rear View 7221

Power ON/OFF

The on/off switch is located on the lower right front of the plotter.

1. Press the button in to turn the power on. Pressing the button a second time turns the power off.
2. When the switch is on, the following will occur: the On Line light and the Out of Limit light are now on and the fan motor starts running.

Loading Pens

1. Select a pen and remove its cap.
2. Place the tip of the pen into the rubber cap at the bottom of the stall you want to use. Push the pen in gently until it snaps into place. Repeat this procedure for any other pen stalls.
3. To remove pen from the stall, press down on lever to the right of the pen. Grasp the pen firmly between your thumb and forefinger and pull gently out of the pen stall.

Loading Paper (7221T)

1. Set the on/off switch to off. Remove pen from the pen holder so ink does not get on your paper.

2. Press down on the paper stop on the left side of the platen. Open both left and right paper supply modules by pressing the door latch buttons.

3. Position roll paper between the hubs in right supply module and align hub tabs with roll notches. Paper should feed across the top of roll.

4. Place paper sprocket holes on sprockets at end of roller. Hold paper while closing door. Make sure paper is not on top of paper guide at front edge of platen

5. Advance paper with thumbwheel until taut.

6. Turn plotter switch to on, and press ADV HALF button to advance the paper half a page.

Loading Paper (7221C)

1. Set the on/off switch to on. Raise the paper stop on left side of the platen by pushing the upper portion of the stop with a pencil or pointed object.

2. Press the Chart Load button. The Chart Load and Out of Limit lights should go on and the plotter arm and pen holder is moved to the upper right corner of the platen.

3. Place a sheet of paper on platen with bottom under plastic lip and left side against the paper stop. Press Chart Hold push button. The Chart Load light should go out and Out of Limit light stays on. Smooth paper down using back of your hand.

For more information on the 7221C/T plotter, refer to the Graphics Plotter Operating and Programming Manual, P/N 07221-90024.

7225A, B Plotter
(17604A Personality Module)

HP7225 Front and Rear Views

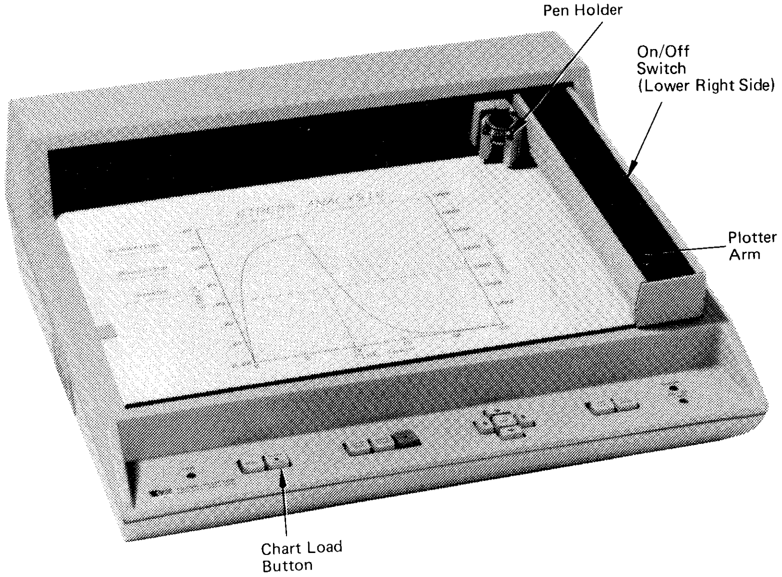
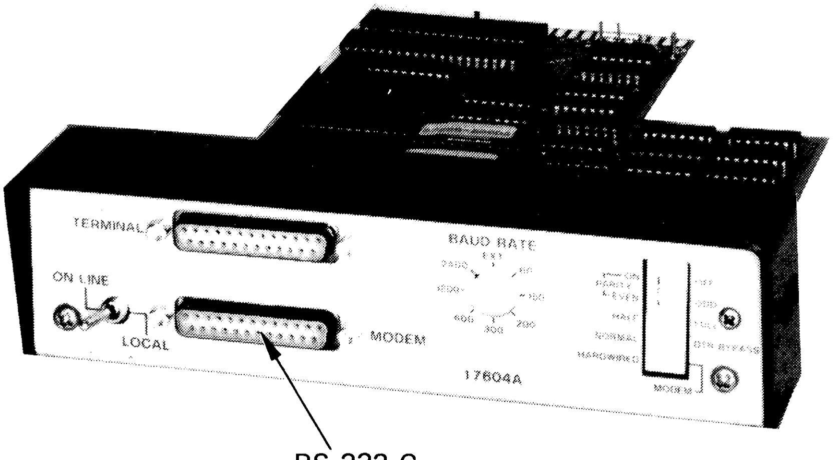


Figure 3-7. Front View



RS-232-C
Connector

Figure 3-8. Rear View

Power ON/OFF

The on/off switch is located on the lower right side towards the rear of the plotter.

1. Press the button in to turn the plotter on. Pressing the button again turns the power off.

2. When the switch is on, the pen holder on the plotter arm moves to the upper right corner and Chart Hold is activated.

Loading Pens

1. Select a pen and remove the cap.

2. When the plotter arm has stopped moving, hold the metal carrier ring up with one hand while pressing the pen into the holder with the other hand.

Loading Paper

1. Press the Chart Load button. This releases the paper hold-down mechanism and moves the plotter arm to the upper right corner of the platen.

2. Position the paper squarely against the ridge at the bottom on the platen. Smooth out the paper with the back of your hand to avoid smearing the natural oil from your fingertips on the paper.

3. Press the Chart Hold button to activate the paper hold-down mechanism.

For more information on the 7225A, B Graphics Plotter and 17604A Personality Module refer to the Operating and Programming Manual, P/N 17604-90000.

HP7245 Front and Rear Views

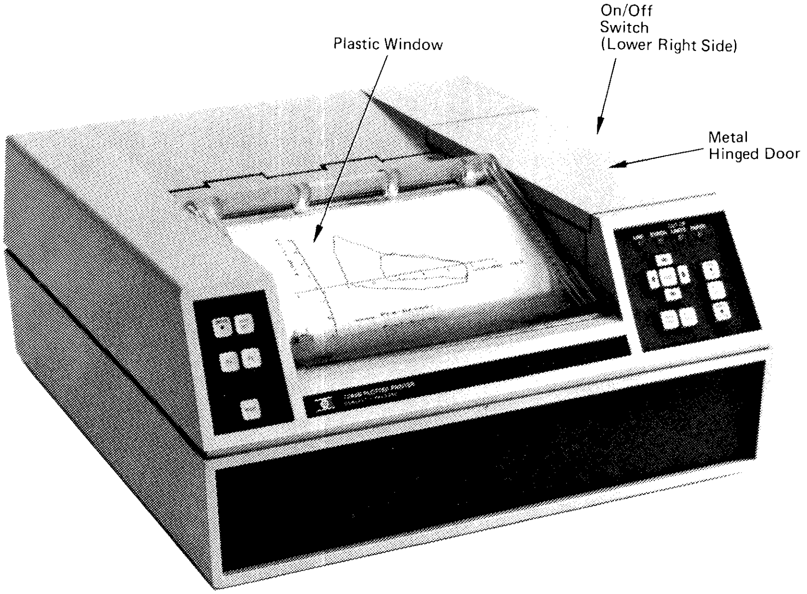


Figure 3-9. Front View

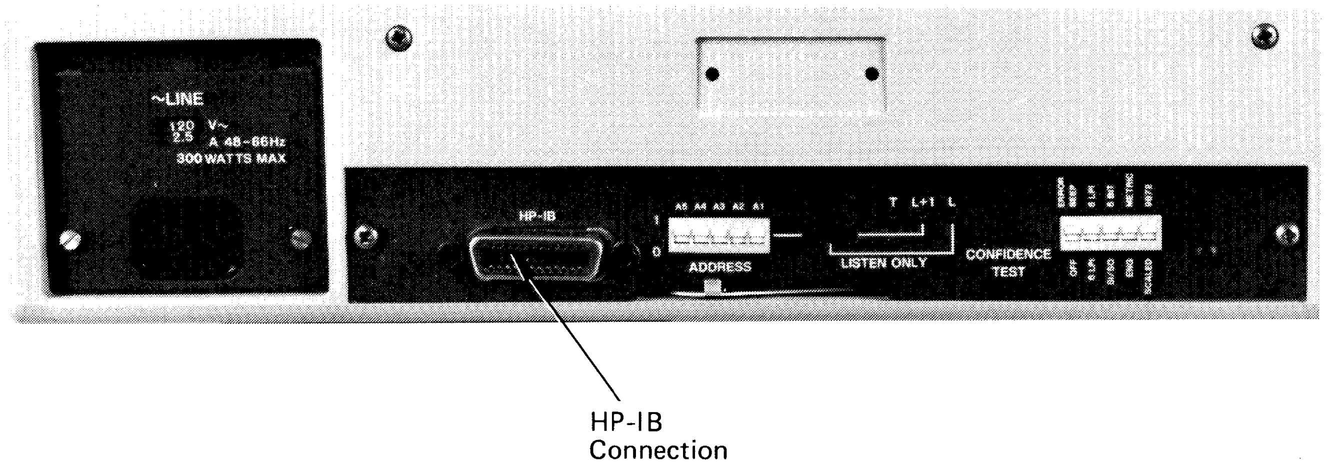


Figure 3-10. Rear View

Power ON/OFF

The on/off switch is located to the right rear of the plotter.

1. To turn the plotter on set the switch to 1. Setting the switch to 0 turns the plotter off.

2. The Line indicator light remains on and the pen/print head moves to the left margin of the paper. The paper advances until it finds the paging hold.

Changing the Pen/Print Head

1. The pen/print head is a long-life device. When it needs to be changed, refer to the 7245A Plotter/ Printer Operating and Programming Manual, P/N 07245-90001 for instructions.

Loading Paper

1. Open plastic window on front and metal hinged door on the top right side of the plotter.

2. Raise paper drive assembly all the way up, then pull lower idler assembly towards front of plotter.

3. Insert roller into paper roll, place paper roll into cradle with the proper side out and seat roller.

4. Pull paper up and over the paper drive with at least six inches of paper extending beyond sprockets. Place the paper holes on left and right sprockets on the platen and then with rear sockets.

5. Close and seat the drum protector, lower paper drive back into place, and close the window and door.

See the HP7245A Plotter/Printer Operating and Programming Manual, P/N 07245-90001, for additional information.

7470A Plotter

HP7470 Front and Rear Views

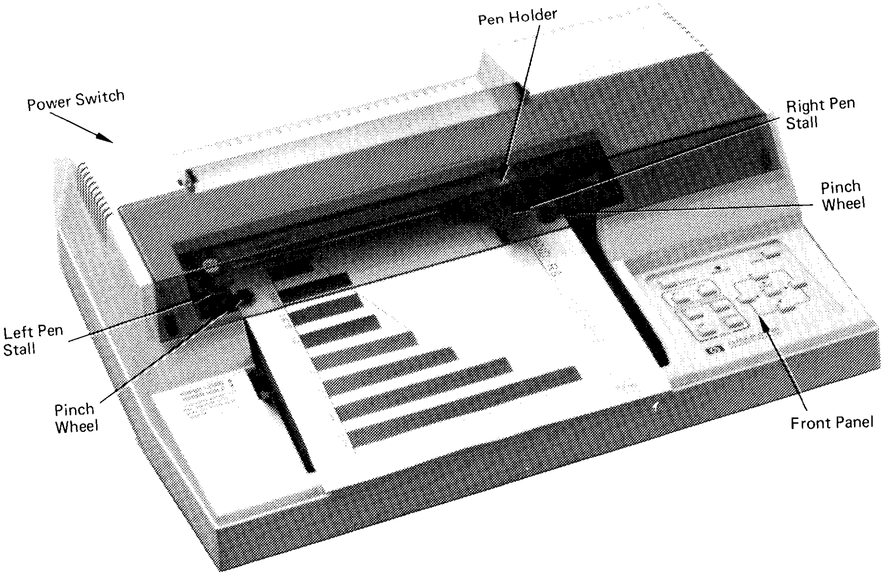


Figure 3-11. Front View

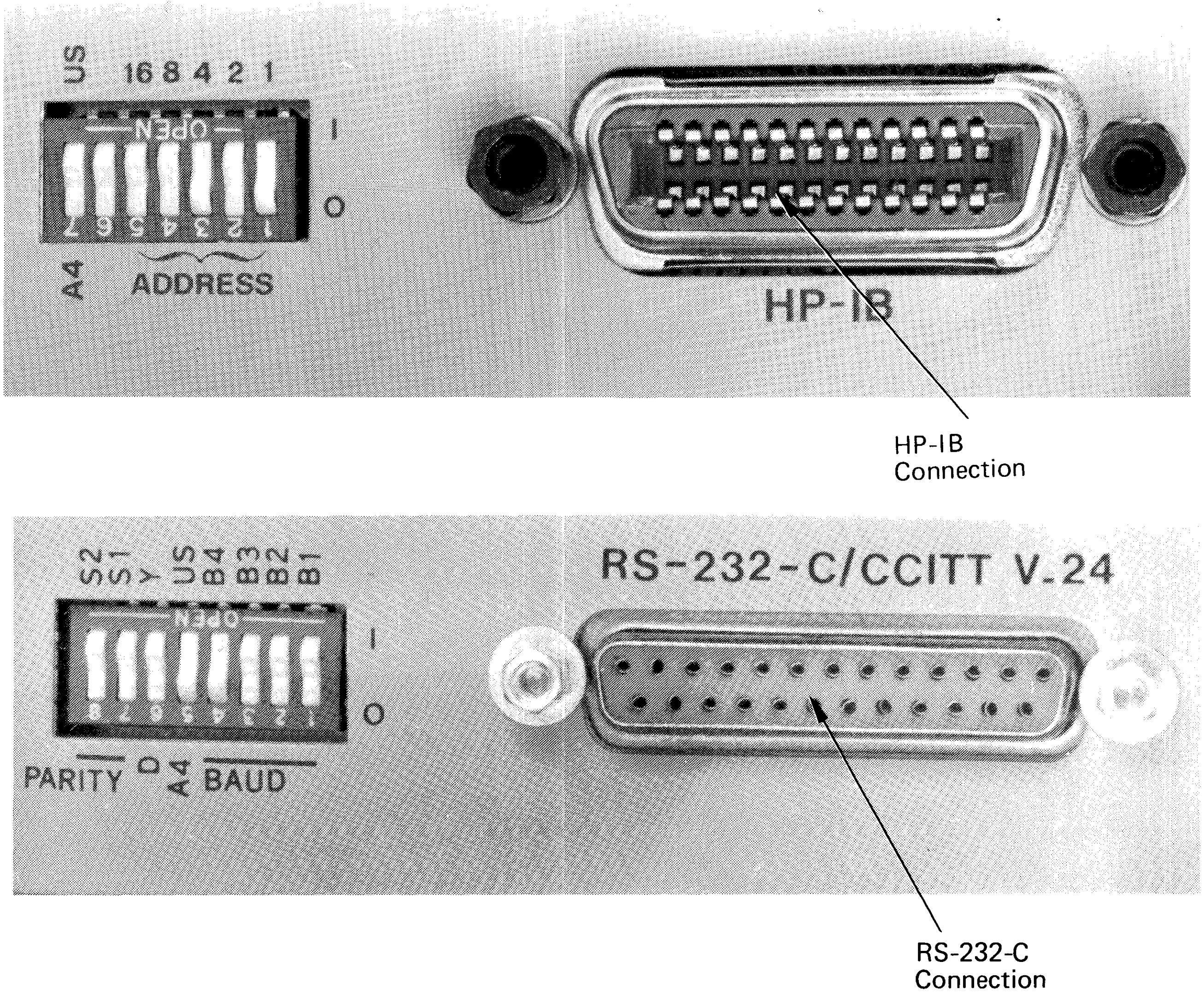


Figure 3-12. Rear View

Power ON/OFF

The on/off switch is located at the left rear of the plotter.

1. Set the switch to on.

2. The Error light comes on momentarily and the pen holder moves to the left pen stall. If there is no pen in the left stall, the pen holder moves to the right stall. When a pen is found, or if no pens are installed, the pen holder returns to a point near the right plotting limit.

Loading Pens

1. Raise the protective cover to gain access to the pen stalls.

2. Select a pen and remove its cap.

3. Place the pen tip in the hole at the base of the stall and press down and in gently until the pen snaps into place. Repeat for the second pen.

4. Lower the protective cover.

Loading Paper

1. Move the Paper Loading Lever (below the left pen stall) to the Paper Load position. This raises the pinch wheels and the paper stop and stores and caps the pen.

2. Place a sheet of paper on the platen surface so that it is against the paper stop and the left edge of the platen.

3. Move the Paper Loading Lever to the Paper Hold position. This lowers the pinch wheels and the paper stop to secure the paper.

4. Transparencies are loaded in the same way as paper.

For further information on the HP7470A plotter, refer to the Operator's Manual, P/N 07470-90002.

7580A/7585A * Drafting Plotter

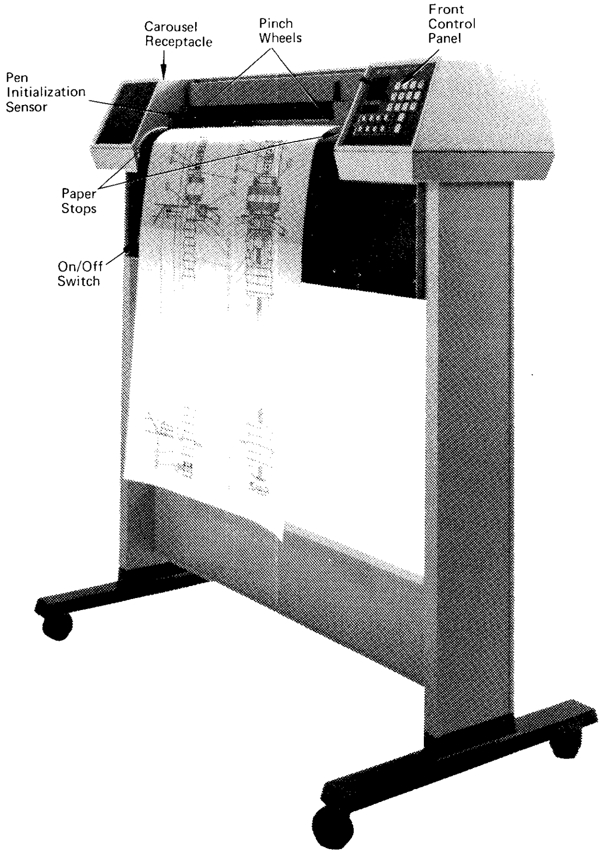


Figure 3-13. Front View

* This plotter is the same as the 7580A except it handles paper 12 inches wide.

Power ON/OFF

The on/off switch is loaded on the front, lower left panel.

1. To turn the power on press the switch to I (ON) position. The 0 (OFF) position turns the power off.

2. When the switch is set to on, the Line and Chart Load lights come on and remain on. The pen moves approximately two inches to the right, then returns to pen initialization sensor at left end on the pen carriage arm.

Loading Pens and Carousels

1. There are three types of pens available: fiber tip, roller ball, and drafting. A separate pen carousel is supplied for each type of pen.

2. Select the pen type and color to be loaded into the corresponding pen carousel and remove the cap.

3. Holding the carousel in one hand, place a pen so the tip flange is resting on the edge of the cap and the pen tip is over the opening.

4. Press the pen down as you straighten it. Be careful that the tip clears the cap and the pen collar fits snugly into the notch on the hub of the carousel.

5. Pull the plunger down and release it to assure the cap seals the pen tightly without binding.

6. Repeat this procedure for the remaining seven pens.

7. Insert pen carousel into carousel receptacle on the top left of the plotter. Rotate slightly until it drops into position on the spindle. No force is required.

Loading Paper

1. Set the Line switch to on. Raise the carriage cover. Position the paper so that its left edge is flush against both the front and back paper stops and hold in this position.

2. Manually position the right pinch wheel to align its scribe mark with the right paper edge.

3. Close the carriage cover and press Chart Hold to lower the pinch wheels. (The plotter will not operate with the carriage cover raised.)

For more information on the HP7580A Drafting Plotter refer to the Operating and Programming Manual, P/N 07580-90000 and Operators Manual, P/N 07580-90002.

9872A/B/S Plotter

HP9872 Front and Rear Views

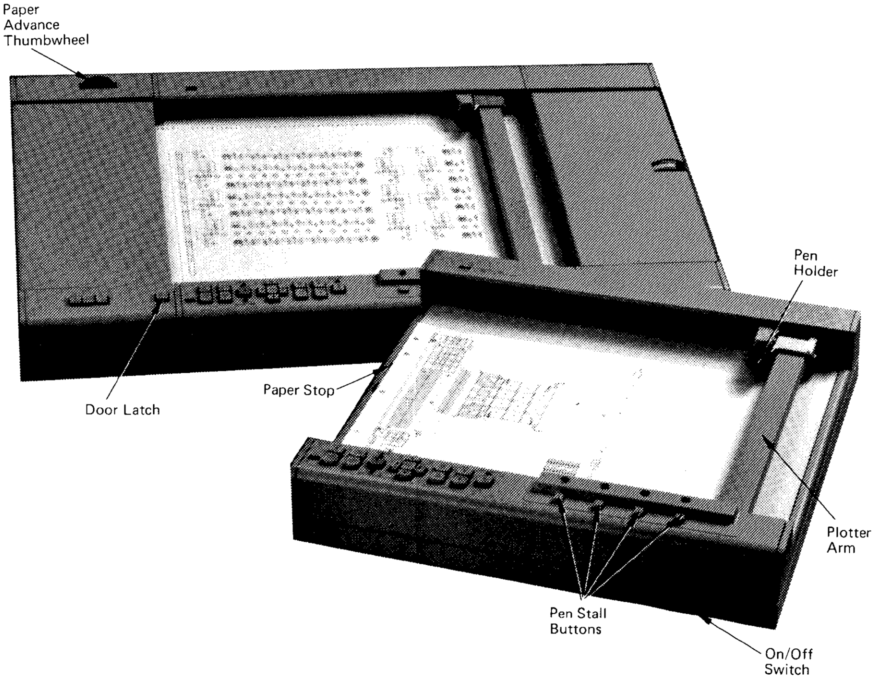
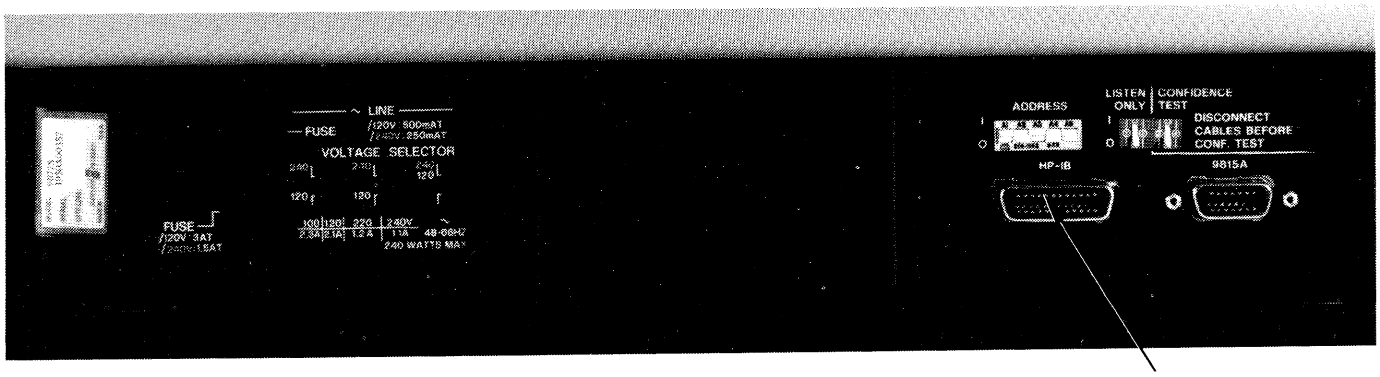


Figure 3-14. Front View



HP-IB Connection

Figure 3-15. Rear View

Power ON/OFF

The on/off switch is located at the lower right front of the plotter.

1. Press the button in to turn the power on. Pressing the button a second time turns the power off.

Loading Pens

1. Select a pen and move its cap.
2. Place the tip of the pen into the rubber cap at the bottom of the stall you want to use. Push the pen in gently until it snaps into place. Repeat this procedure for any other pen stall.
3. To remove pen from the stall, press down on lever to the right of the pen. Grasp the pen firmly between thumb and forefinger and pull gently out of the pen stall.

NOTE: Make sure that you place the right color of pen in the pen slot that matches the pen number you selected in HPDRAW.

Loading Paper (9872B)

1. Press the Chart Load button to release the paper hold mechanism and move the plotter arm to the upper right corner of the platen. The Chart Load light comes on.

2. Place a sheet of paper on the platen surface so that it is against the ridge at the bottom of the platen and against the paper stop on the left side of the platen. Smooth the paper.

3. Press the Chart Hold button to activate the hold-down mechanism. The Chart Load light goes off. Again smooth the paper.

Transparencies are loaded in the same way as paper.

Loading Paper (9872S)

1. Store pens to avoid getting ink on the paper. Turn off the plotter.

2. Press down on the paper stop on the left side of the platen and open both the left and the right paper supply modules by pressing the door latch buttons.

3. Hold the paper roll in your left hand and load the roll between hubs in supply module with paper feeding across the top of the roll, aligning hub tabs with roll notches.

4. Feed paper across table and under arm.

5. Engage paper sprocket holes and sprockets at each end of take-up roller. Hold paper on roller while closing door.

6. Close supply module door. Make sure paper is not on top of front edge guide.

7. Advance paper with thumbwheel until taut. Turn plotter on and advance paper at least once with front panel buttons. The page advance option is automatically turned on when paper is loaded successfully.

For more information on the HP9872B/S plotter, refer to the Operating and Programming Manual, part number 09872-90008.

9872C, T Plotter

HP9872 Front and Rear Views

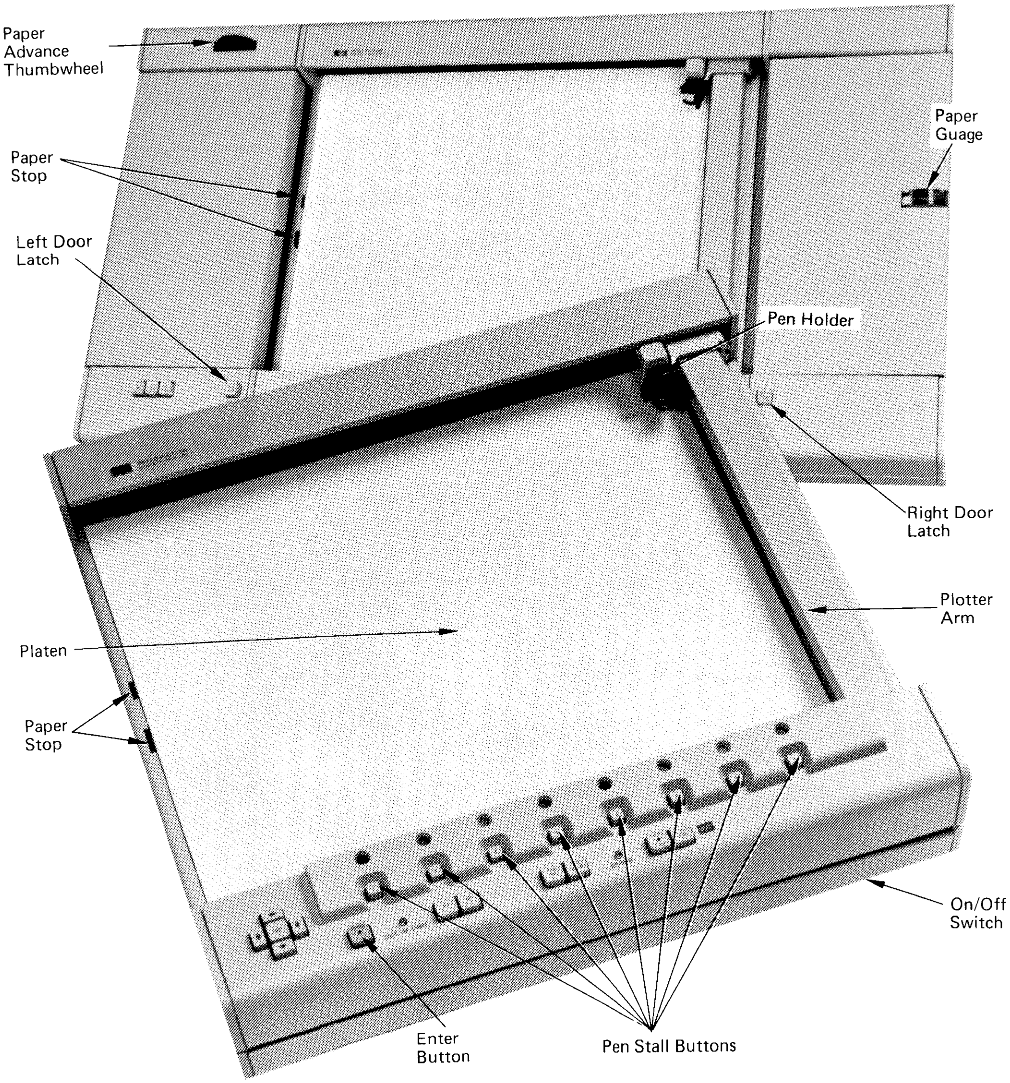


Figure 3-16. Two Front Views 9872C and 9872T

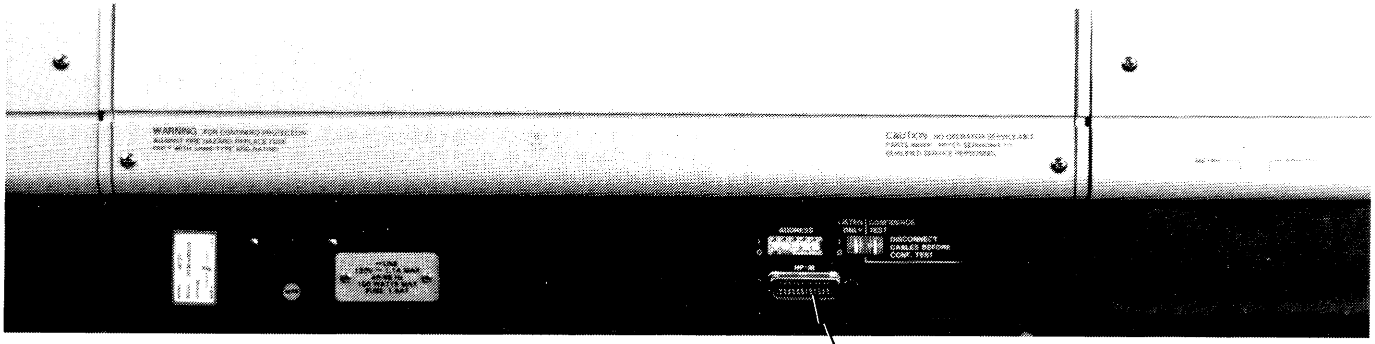


Figure 3-17. Rear View

Power ON/OFF

The on/off switch is located at the lower front of the plotter.

1. Press the button in to turn the plotter on. Pressing the switch again turns the plotter off.

2. When the switch is on, the pen holder on the plotter arm moves to the lower right corner.

Loading Pens

1. Select a pen and remove its cap.

2. Place the tip of the pen into the rubber cap at the bottom of the stall you want to use. Push the pen in gently until it snaps into place. Repeat this procedure for any other pen stall.

3. To remove pen from the stall, press down on lever to the right of the pen. Grasp the pen firmly between your thumb and forefinger and pull gently out of the pen stall.

Loading Paper (9872T)

1. Set the on/off switch to off. Remove pen from the pen holder so ink does not get on your paper.

2. Press down on the paper stop on the left side of the platen. Open both left and right paper supply modules by pressing the door latch buttons.

3. Position roll paper between hubs in right supply module and align hub tabs with roll notches. Paper should feed across top of roll.

4. Engage paper sprocket holes on sprockets at end of roller. Hold paper while closing door. Make sure paper is not on top of paper guide at front edge of platen.

5. Advance paper with thumbwheel until taut.

6. Turn plotter switch to on, and press ADV HALF button to advance the paper half a page.

Loading Paper (9872C)

1. Set the on/off switch to on. Raise the paper stop on left side of the platen by pushing the upper portion of the stop with a pencil or pointed object.

2. Press the Chart Load button. The Chart Load and Out of Limit lights should go on and the plotter arm and pen holder is moved to the upper right corner of the platen.

3. Place a sheet of paper on platen with bottom under plastic lip and left side against the paper stop. Press Chart Hold push button. The Chart Load light should go out and Out of Limit light stays on. Smooth paper down using back of your hand.

For more information on the HP9872C/T plotter, refer to the Graphics Plotter Operating and Programming Manual, P/N 09872-90011.

Graphics Terminals

You must have a graphics terminal to use HPDRAW. The three supported graphics terminals are the HP2623A, HP2647A/F, and HP2648A. Many keys on the three keyboards are located in different positions; however, the keys perform the same operations in HPDRAW.

The major difference in the keyboards is the CTRL key on the HP2623A keyboard. This key is labelled CNTL on the HP2647 and HP2648 terminals.

HP2647A, F and HP2648A Graphics Terminals

Some of the basic features that you need to be familiar with of the HP2647 and HP2648 graphics terminals are illustrated below. Keys and functions that relate specifically to HPDRAW are explained in Section 1.

For a complete explanation of using your terminal, refer to the HP2647A User's Manual, P/N 02647-90001 or the HP2648A User's Manual, P/N 02648-90001.

Power ON/OFF

The on/off is located at the left rear of the terminal.

1. Set the switch to the on position and the TERMINAL READY message appears in the upper left corner of your screen.

2. Pressing the switch the opposite direction turns the terminal off.

3. The power cord is located atthe left rear of the terminal next to the on/off switch.

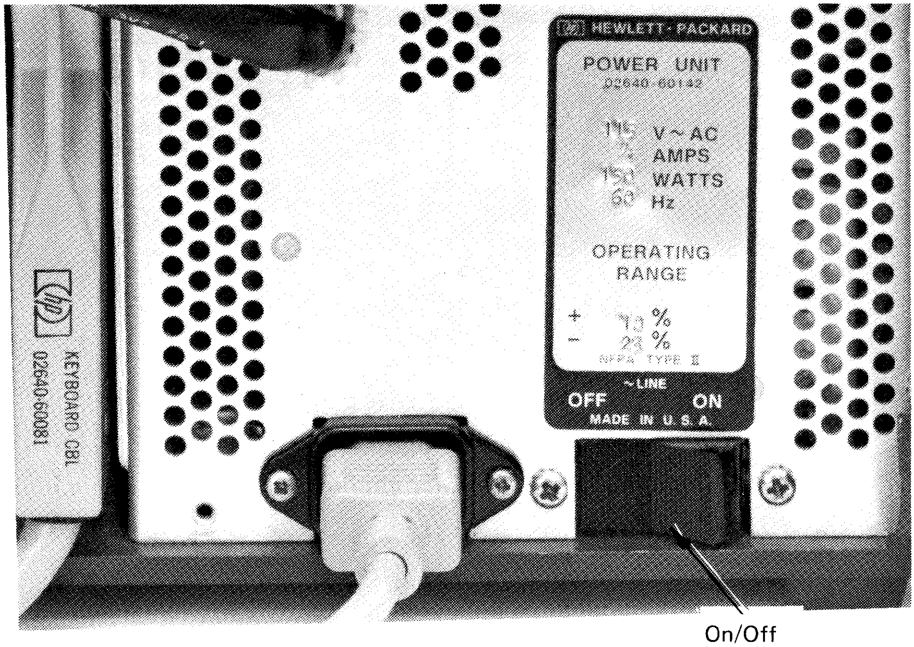


Figure 3-18. HP264X Rear View

HP2647A, F and HP2648A Keyboards

The HP2647A, F and HP 2648A keyboards are made up of six groups: an alphanumeric keyboard, a terminal control group, function keys, edit group, display control group, and graphics control group. (See the photo for the location of each group for your specific terminal.)

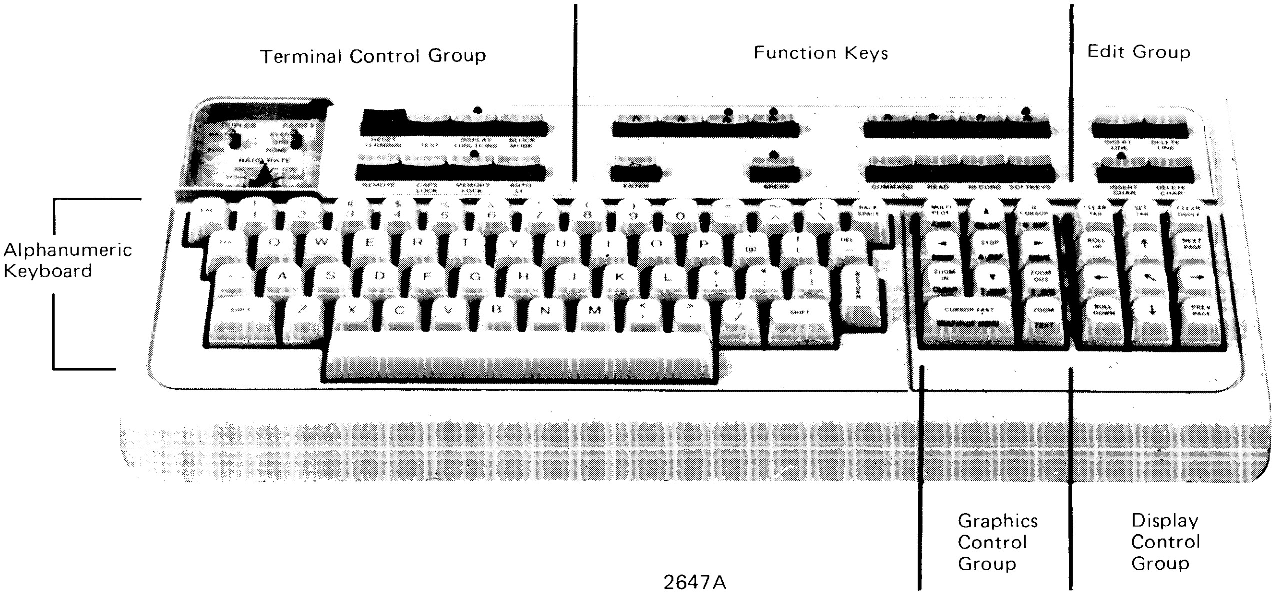


Figure 3-19. HP2647A, F Keyboard

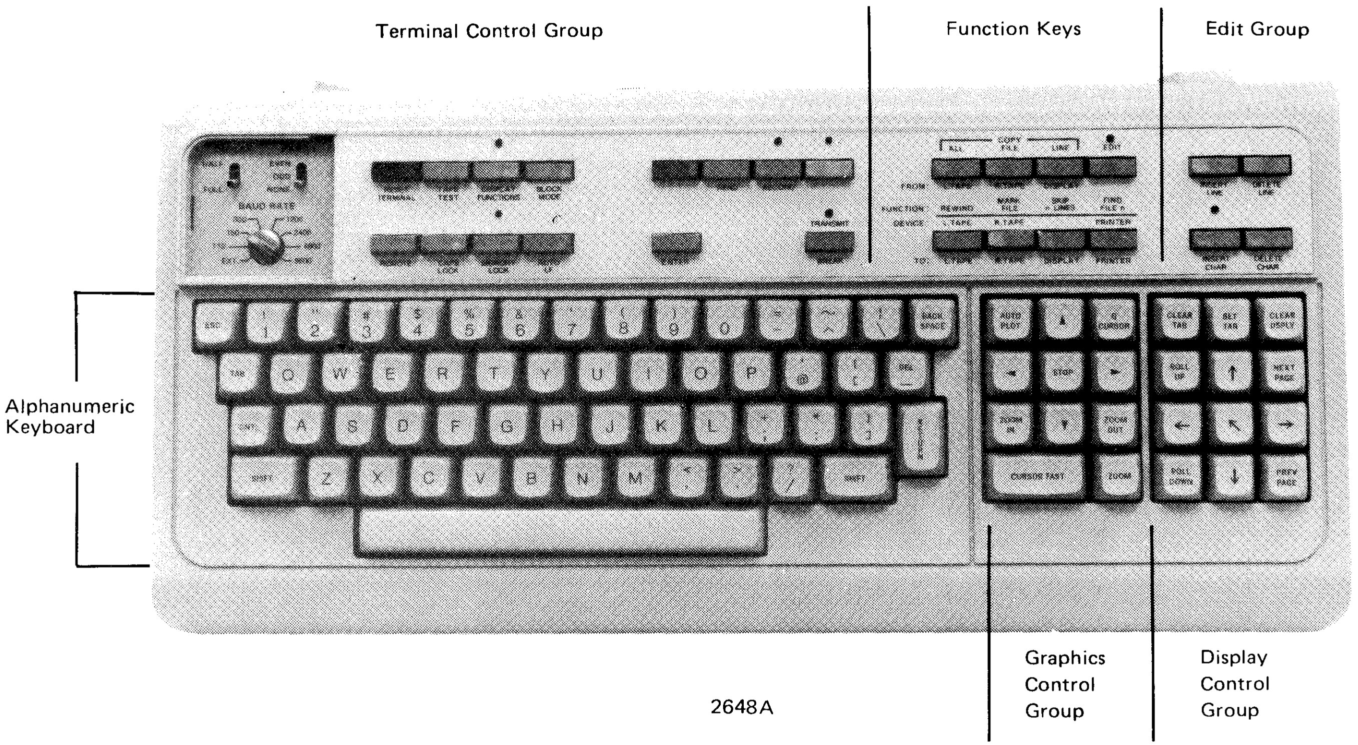


Figure 3-20. HP2648A Keyboard

Terminal Control Group

The REMOTE key is located in this group. The key is always pressed down to communicate with the computer.

The CAPS LOCK key can be either up or down. With the key down you can type all capital letters. With the key in the up position, you can type upper and lower case letters.

The remaining keys should be in the up position.

Function Keys

The function keys for the HP2647A and the HP2648A Graphics Terminals perform the same functions; however, the keys are located in different positions on the keyboards.

The function keys are numbered f1 through f8. These keys match the labels shown on the bottom of your terminal screen.

Display Control Group

The keys marked with an arrow move the cursor in the direction the arrow is pointing. By pressing these keys you can move the alphanumeric cursor to any position on the screen.

The arrow in the center "homes" the cursor (puts it at the beginning of the first field).

Graphics Control Group

The keys marked with an arrow move the graphics cursor (+) in the direction the arrow is pointing. Use these keys to place HPDRAW objects in the location you choose.

Pressing CURSOR FAST with one of the arrow keys moves the graphics cursor faster.

Alphanumeric Keyboard

The terminal keyboard works like a typewriter keyboard. See Section 1 for an explanation of the keys and how you use them with HPDRAW.

Edit Group

You use these keys to edit fields on a menu. See Section 1 for an explanation of the keys you can use with HPDRAW.

2623A Graphics Terminal

Power ON/OFF

The on/off switch is located at the right rear of the terminal.

1. Set the switch to the on position and the cursor is displayed in screen column 1, row 1. Also, the primary function key labels are displayed across the bottom of the screen.

2. To turn the terminal off, set the power switch to the OFF position.

3. The power cord is located at the right rear, below the power switch.

Connectors

The keyboard connector is located at the left rear of the terminal. There is a label above the connector which indicates "KYBD".

There are two additional port connectors. Port 1 is used to connect your terminal with the port that communicates with the computer.

Port 2 is used to connect another device such as a printer.

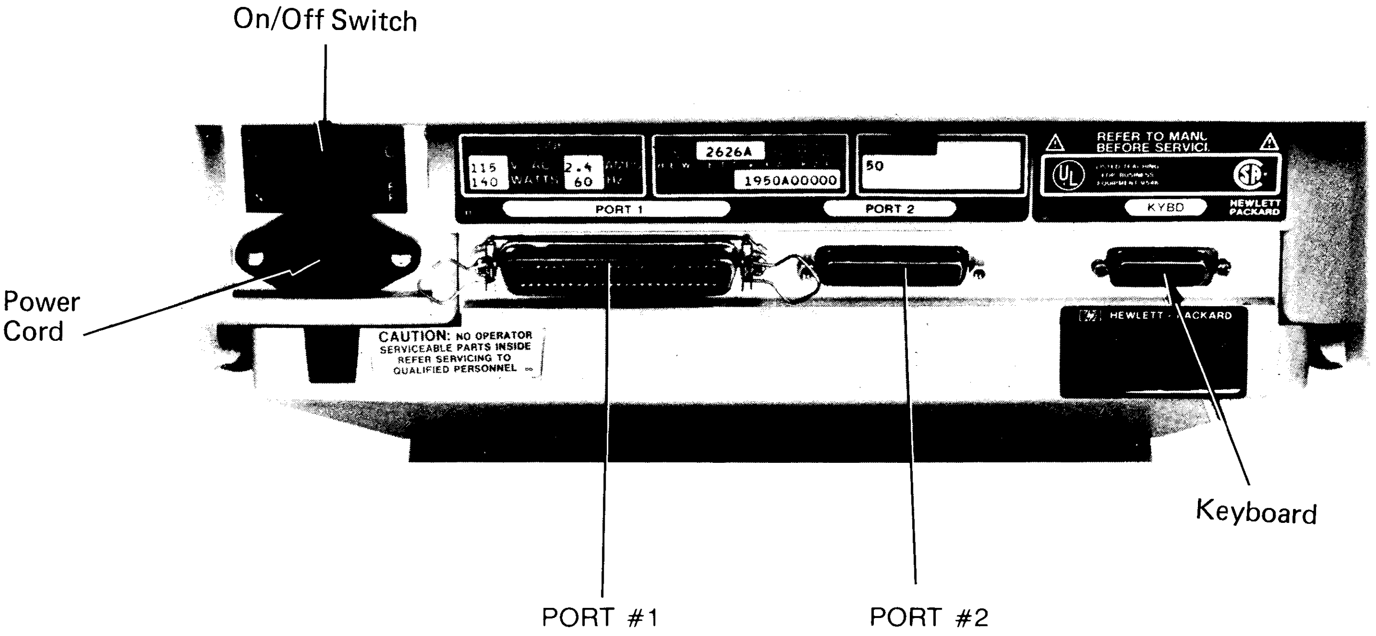


Figure 3-21. HP2623A Rear View

HP2623A Keyboard

The HP2623A keyboard consists of seven functional groups. The Character set, Graphics Control, Function Keys, Function Control Keys, and Edit group are briefly discussed here. Section 1 describes the keys you can use with HPDRAW. For additional information on the Terminal Control and Display Control groups, refer to the HP2623A User's Manual, P/N 02623-90001.

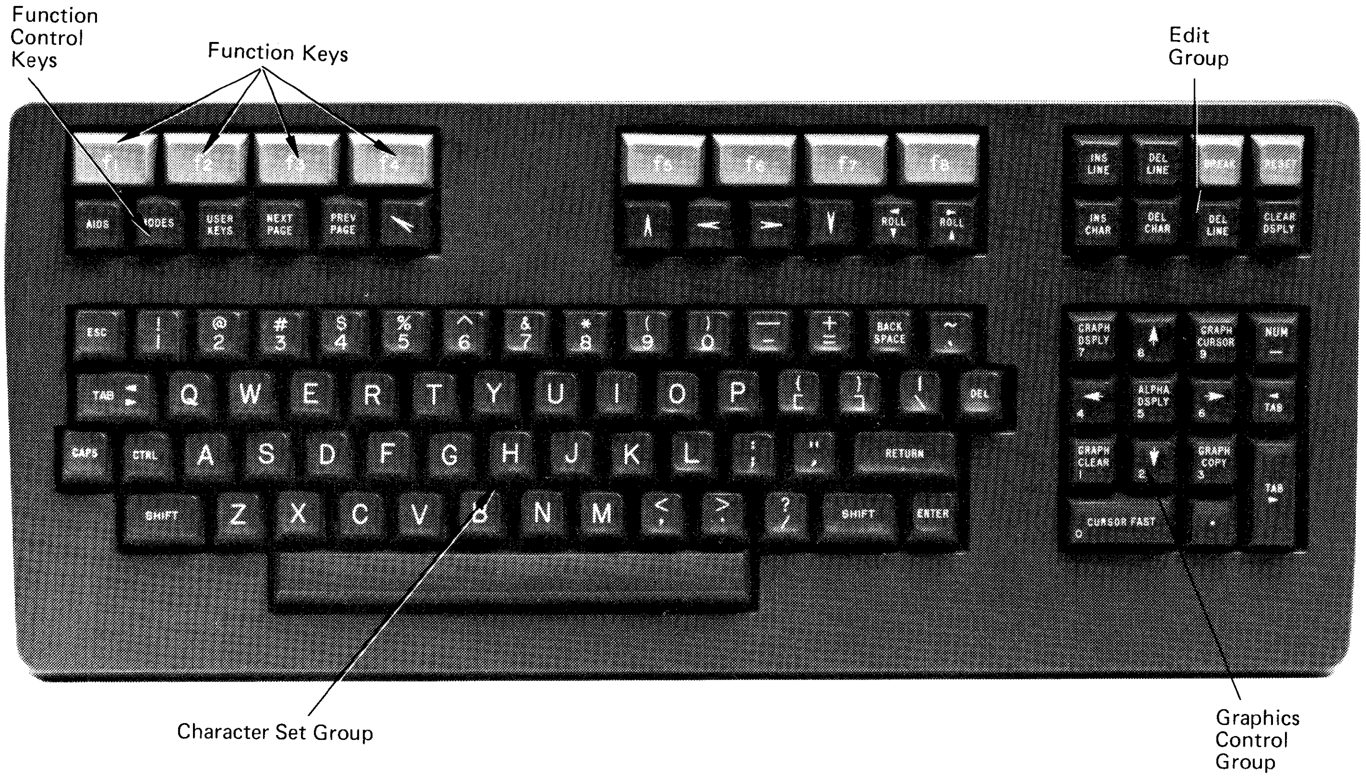


Figure 3-22. HP2623A Keyboard

Character Set Group

The Character Set Group keys are located at the right of the keyboard. They control the graphics cursor. The four keys with an arrow move the graphics cursor (+) in the direction the arrow is pointing. Use these keys to place text or drawings in the work area.

CURSOR FAST, when pressed simultaneously with one of the four cursor movement keys, moves the graphics cursor at a higher speed.

For the remaining Graphics Control keys functions, see Section 1 or the HP2623A User's Manual.

Function Keys

The Function Keys Group consists of eight keys located across the top of the keyboard. They are labelled f1 through f8. These keys are associated with the eight function key labels across the bottom of your screen. For example, the second label from the left on your screen is associated with the f2 function key. You press f2 to perform the function of that label.

Function Control Keys

This group consists of the AIDS, MODES, and USER KEYS. Use these keys to select the functions available to the function keys, including the REMOTE key. See the 2623A Users Manual for a list of the functions available.

Edit Group

Information displayed in fields can be edited by using the Edit Keys. CLEAR DSPLY clears the display from the cursor position to the end of the last field. CLEAR LINE clears the line from the cursor to the end of the line.

INS CHAR allows you to insert characters without overwriting existing characters. Place your cursor to the right of where you want your character inserted. This shifts the existing characters to the right one position for each character inserted. DEL CHAR deletes the character you position the cursor under.

For a complete explanation of using your terminal, refer to the HP2623A User's Manual, P/N 02623-90001.

INDEX

READER COMMENT SHEET

DSG/3000 Manual

32250-90001 Aug 1982

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

Is this manual technically accurate? Yes □ No □ (If no, explain under Comments, below.)

Are the concepts and wording easy to understand? Yes □ No □ (If no, explain under Comments, below.)

Is the format of this manual convenient in size, arrangement, and readability? Yes □ No □ (If no, explain or suggest improvements under Comments, below.)

Comments:

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1070 CUPERTINO,CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Publications Manager
Hewlett-Packard Company
Information Networks Division
19420 Homestead
Cupertino, California 95014

FOLD FOLD