Professional Graphics Series

# IBM RT PC Plotting System Programmer's Guide

Programming Family

IBM

Personal
Computer
Software

59X8583

International Business Machines Corporation Armonk, New York 10504

# IBM Program License Agreement

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THIS PACKAGE. OPENING THIS PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE UNOPENED AND YOUR MONEY WILL BE REFUNDED.

IBM provides this program and licenses its use in the United States and Puerto Rico. Title to the media on which this copy of the program is recorded and to the enclosed copy of the documentation is transferred to you, but title to the copy of the program is retained by IBM or its supplier, as applicable. You assume responsibility for the selection of the program to achieve your intended results, and for the installation, use and results obtained from the program.

## LICENSE

You may:

a. use the program on only one machine at any one time except as otherwise specified by IBM in the enclosed Program Specifications (available for your inspection prior to your acceptance of this Agreement);

b. copy the program into machine readable or printed form for backup or modification purposes only in support of such use. (Certain programs, however, may include mechanisms to limit or inhibit copying. They are marked "copy protected");

c. modify the program and/or merge it into another program for your use on the single machine. (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement.); and,

d. transfer the program with a copy of this Agreement to another party only if the other party agrees to accept from IBM the terms and conditions of this Agreement. If you transfer the program, you must at the same time either transfer all copies whether in printed or machine-readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the program contained or merged into other programs. IBM will grant a license to such other party under this Agreement and the other party will accept such license by its initial use of the program. If you transfer possession of any copy, modification or merged portion of the program, in whole or in part, to another party, your license is automatically terminated.

You must reproduce and include the copyright notice on any copy, modification, or portion merged into another program.

You may not reverse assemble or reverse compile the program without IBM's prior written consent.

You may not use, copy, modify, or transfer the program, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this Agreement.

You may not sublicense, assign, rent or lease this program.

## TERM

The license is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies, modifications and merged portions in any form.

## LIMITED WARRANTY AND DISCLAIMER OF WARRANTY

IBM warrants the media on which the program is furnished to be free from defects in materials and workmanship under normal use for a period of 90 days from the date of IBM's delivery to you as evidenced by a copy of your receipt.

IBM warrants that each program which is designated by IBM as warranted in its Program Specifications, supplied with the program, will conform to such specifications provided that the program is properly used on the IBM machine for which it was designed. If you believe that there is a defect in a warranted program such that it does not meet its specifications, you must notify IBM within the warranty period set forth in the Program Specifications.

ALL OTHER PROGRAMS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT IBM OR AN IBM AUTHORIZED REPRESENTATIVE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IBM does not warrant that the functions contained in any program will meet your requirements or that the operation of the program will be uninterrupted or error free or that all program defects will be corrected.

THE FOREGOING WARRANTIES ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

## LIMITATIONS OF REMEDIES

IBM's entire liability and your exclusive remedy shall be as follows:

1. With respect to defective media during the warranty period:
   a. IBM will replace media not meeting IBM's "Limited Warranty" which is returned to IBM or an IBM authorized representative with a copy of your receipt.
   b. In the alternative, if IBM or such IBM authorized representative is unable to deliver replacement media which is free of defects in materials and workmanship, you may terminate this Agreement by returning the program and your money will be refunded.
2. With respect to warranted programs, in all situations involving performance or nonperformance during the warranty period, your remedy is (a) the correction by IBM of program defects, or (b) if, after repeated efforts, IBM is unable to make the program operate as warranted, you shall be entitled to a refund of the money paid or to recover actual damages to the limits set forth in this section.

   For any other claim concerning performance or nonperformance by IBM pursuant to, or in any other way related to, the warranted programs under this Agreement, you shall be entitled to recover actual damages to the limits set forth in this section.

IBM's liability to you for actual damages for any cause whatsoever, and regardless of the form of action, shall be limited to the greater of $5,000 or the money paid for the program that caused the damages or that is the subject matter of, or is directly related to, the cause of action.

In no event will IBM be liable to you for any lost profits, lost savings or other incidental or consequential damages arising out of the use of or inability to use such program even if IBM or an IBM authorized representative has been advised of the possibility of such damages, or for any claim by any other party.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

## SERVICE

Service from IBM, if any, will be described in Program Specifications or in the statement of service, supplied with the program, if there are no Program Specifications.

IBM may also offer separate services under separate agreement for a fee.

## GENERAL

Any attempt to sublicense, assign, rent or lease, or, except as expressly provided for in this Agreement, to transfer any of the rights, duties or obligations hereunder is void.

This Agreement will be construed under the Uniform Commercial Code of the State of New York.

# Program Specification

## IBM RT Personal Computer Plotting System Licensed Program (55X8925)

### Statement of Limited Warranty

The IBM RT Personal Computer[1] Plotting System Licensed Program is warranted to conform to this Program Specification when properly used in its designated operating environments.

Any other documentation with respect to this licensed program is provided for information purposes only and does not extend or modify this IBM RT PC Plotting System Licensed Program Program Specification.

The IBM RT PC Plotting System Licensed Program Program Specification may be updated from time to time. Such updates may constitute a change to these specifications.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

This limited warranty and the 90-day program media warranty are contained in the IBM Program License Agreement supplied with this product. These warranties are available to all licensees of the IBM RT PC Plotting System Licensed Program. The limited warranty period is until February 1, 1988, or until six months after written notice by IBM that the warranty period has been terminated, whichever is sooner.

### Statement of Function Warranted

The IBM RT Personal Computer Plotting System Licensed Program (55X8925) is a subroutine library of functions designed to assist the user in developing programs to produce various types of charts. This subroutine library may also be used to develop interactive graphics applications.

The highlights of this licensed program are:

- Supports area, bar, line, pie, scatter, schedule, step, and text-only chart types
- Allows graphic and text annotation
- Supplies graphic input functions
- Permits linear and logarithmic axes
- Allows for a combination of chart types on a single chart
- Allows changing from one chart type to another chart type with a single command
- Provides a program interface via language bindings to IBM RT PC FORTRAN 77, IBM RT PC BASIC Interpreter and Compiler (compiler only), IBM RT PC Pascal, and the C compiler provided with the IBM RT PC AIX[2] Operating System
- Allows device independence through the use of IBM RT PC Virtual Device Interface.

[1] RT, RT PC, and RT Personal Computer are trademarks of IBM

[2] AIX i a trademark of IBM

## Specified Operating Environment

### Machine Requirements

The minimum machine requirements are:

- An IBM RT PC with a display (for example, the IBM RT PC Advanced Color Graphics Display, IBM RT PC Advanced Monochrome Graphics Display, the IBM Personal Computer Display, or an equivalent display).

**Note:** The number of users on the IBM RT PC AIX Operating System Licensed Program, Version 1.1, (74X9995), the number and type of tasks, and the application requirements may expand the requirements beyond these minimums.

### Programming Requirements

The IBM RT PC AIX Operating System Licensed Program, Version 1.1, (74X9995) is a prerequisite for program execution.

## Statement of Service

Program service for valid program-related defects in the IBM RT PC Plotting System Licensed Program is available to all IBM RT PC Plotting System Licensed Program licensees until February 1, 1988, or until six months after written notice by IBM that the warranty period has been terminated, whichever is sooner. However, service will be provided only for the current update level and for the prior release for ninety (90) days following release of the current level of update.

Each licensee's access to program service is determined by the marketing channel through which the license was obtained. For example, in the United States and Puerto Rico, if the IBM RT PC Plotting System Licensed Program license was obtained through:

- An authorized IBM Personal Computer dealer.

  Requests for program service should be made through your dealer.

- The IBM North-Central Marketing Division or the IBM South-West Marketing Division.

  Your company will have established a technical support location to interface to IBM central service through an IBM Support Center, and your request for program service should be made through your company's technical support location.

If the IBM RT PC Plotting System Licensed Program is obtained through transfer of license from another party under the conditions of the IBM Program License Agreement supplied with this product, the new licensee may obtain program service through the access arrangement provided for the original licensee.

When a license is transferred, if the original license was obtained through the IBM North-Central Marketing Division or the IBM South-West Marketing Division, the old licensee is responsible for contacting their IBM marketing representative to make arrangements to transfer service entitlement to the new licensee; the new licensee must also establish a qualified technical support location to interface to IBM central service.

IBM does not guarantee service results or that the program will be error free, or that all program defects will be corrected.

IBM will respond to a reported defect in an unaltered portion of a supported release of the licensed program by issuing: defect correction information such as correction documentation, corrected code, or notice of availability of corrected code; a restriction; or a bypass.

Corrected code is provided on a cumulative basis on diskettes; no source code is provided. Only one copy of the corrections with supporting documentation will be issued to the licensee, or the agent of the licensee, reporting the defect. IBM will authorize various agents such as the IBM Personal Computer dealers and the IBM North-Central Marketing Division or IBM South-West Marketing Division customer's technical support locations to make and distribute a copy of the corrections if needed, to each IBM RT PC Plotting System Licensed Program licensee which they serve.

IBM will notify authorized IBM Personal Computer dealers, IBM marketing and service representatives, and IBM North-Central Marketing Division and IBM South-West Marketing Division customer's technical support locations if and when an update is made available. Program updates contain all currently available changes for the licensed program.

Licensees may request available updates to this licensed program, if any, prior to the program service termination date. As with defect corrections, IBM will authorize various agents such as IBM Personal Computer dealers and the IBM North-Central Marketing Division and IBM South-West Marketing Division customer's technical support locations to distribute a copy of the update, if needed, to each IBM RT PC Plotting System Licensed Program licensee which they serve.

The total number of copies of an update distributed to IBM RT PC Plotting System Licensed Program licensees within a customer's location may not exceed the number of copies of the IBM RT PC Plotting System Licensed Program licensed to the customer.

IBM does not plan to release updates of IBM RT PC Plotting System Licensed Program code on a routine basis for preventative service purposes. However, should IBM determine that there is a general need for a preventative service update, it will be made available to all licensees through the same process that is utilized to distribute general IBM RT PC Plotting System Licensed Program updates, as described above.

Following the discontinuance of all program services, this program will be distributed on an "As Is" basis without warranty of any kind either express or implied.

June 1986 Printed in U.S.A.
(GC23-0954-1)

IBM Corporation
Industry Systems Products
472 Wheelers Farms Road
Milford, Connecticut 06460

Professional Graphics Series

# IBM RT PC Plotting System Programmer's Guide

Programming Family

Personal
Computer
Software

## First Edition (July 1986)

Changes are made periodically to the information herein; these changes will be incorporated in new editions of this publication.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

**International Business Machines Corporation provides this manual "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this manual at any time.**

Products are not stocked at the address given below. Requests for copies of this product and for technical information about the system should be made to your authorized &ibm. dealer.

A reader's comment form is provided at the back of this publication. If the form has been removed, address comments to IBM Corporation, Department 997, 11400 Burnet Road, Austin, Texas 78758. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

# About This Book

This book explains how to use the IBM RT Personal Computer Plotting System, a library of routines which enables you to create charts within an application program. It explains the features of Plotting System and how to use them. Pseudocode for several programs is provided at the end of Chapter 1.

## Who Should Read This Book

This book is written for people who write graphics application programs. To use this product, you should be familiar with the IBM RT PC AIX[1] Operating System and with the general use of graphics software. In addition, you should know how to program in one or more of the following languages:

- Pascal
- C
- BASIC
- FORTRAN.

[1] AIX is a trademark of IBM.

# Before You Begin

Before you begin using Plotting System, ensure that you have the minimum requirements for both hardware and software.

## Hardware Requirements

The minimum hardware required to use Plotting System is one of the graphics output devices supported by the IBM RT PC Graphics Development Toolkit.

## Software Requirements

The IBM RT PC Graphics Development Toolkit device drivers must be installed before using Plotting System. These device drivers are provided as part of the IBM RT PC Multi-User programs. The description of the device drivers is contained in *IBM RT PC Graphics Development Toolkit*, Appendix C, "Graphics Drivers."

IBM RT PC Plotting System provides language bindings for C, BASIC, Pascal, and FORTRAN. The IBM RT PC C Compiler is included with the IBM RT PC AIX Operating System. In order to use BASIC, Pascal, or FORTRAN, you must have one of the following language compilers:

- IBM RT PC BASIC Interpreter and Compiler
- IBM RT PC Pascal
- IBM RT PC FORTRAN 77.

# How To Use This Book

The purpose of this publication is to describe how to use Plotting System to create charts for application programs. The purpose and function of the routines are given and discussion about selecting and using various types of charts is presented.

The material on each routine in Chapter 3, "Plotting System Routines", is described in generic terms. The specific syntax for the routines is provided in the Language Reference Booklets for each language.

## Organization

This book is organized into three chapters, four appendixes, a glossary, and an index as follows:

- **Chapter 1,** “Using Plotting System”, explains how to use the features of Plotting System to create several types of charts. Pseudocode examples are given to assist the programmer in incorporating Plotting System into the application program.
- **Chapter 2,** “Selecting A Chart”, aids the programmer in choosing among the alternative types of charts available and shows how the charts are implemented by Plotting System.
- **Chapter 3,** “Plotting System Routines”, contains descriptions of each routine included in Plotting System.
- **Appendix A,** “Installing Plotting System”, contains instructions for installing Plotting System.
- **Appendix B,** “Error Codes”, is a list of error codes generated by Plotting System.
- **Appendix C,** “Plotting System Defaults”, is an alphabetical list of the default for each routine.
- **Appendix D,** “ASCII Character and Control Codes”, is provided for your reference.
- The **Glossary** defines many of the technical terms that appear in the Plotting System Programmer’s Guide.
- The **Index** shows you where to find information about the topics discussed in this manual.

**Note:** Language reference booklets are provided for each of the languages supported by Plotting System. Each booklet is a quick reference to the syntax needed to include each routine in an application program.

## Highlighting

Throughout this manual, bold italics have been used to highlight the first occurrence of new terms. In the descriptions of Plotting System routines, parameter names are italicized. Computer output and user input are shown in monospace type and in color.

## Related Information

The following books contain related information that you might find useful.

*IBM RT PC Graphics Development Toolkit*
This book provides an overview of the Graphics Development Toolkit operation including specific information about how to incorporate Toolkit routines into graphic application programs. Included with the Toolkit are language reference booklets that support C, BASIC, FORTRAN, and Pascal programming languages.

*IBM RT PC Graphical File System Programmer's/User's Guide*
This book is a guide to the installation and use of the Graphical File System. It contains information about Metafile concepts and elements, interactive interpretation of metafiles and programming with the Graphical File System.

*IBM RT PC Graphics Terminal Emulator User's Guide*
This book is a guide to the installation and use of the Graphics Terminal Emulator. It contains information about defining communications protocol to allow communications with a host computer, transferring data to and from a host computer, and redirecting graphics output from the host computer to a plotter, printer, or other output device.

*IBM RT PC C Language Guide and Reference*
This book provides information for writing, compiling, and running C Language programs.

*IBM RT PC BASIC Language Reference*
This book is an encyclopedia-type manual. It contains syntax and format for BASIC language commands, statements, and functions, arranged in alphabetical order.

*IBM RT PC BASIC Language Handbook*
This book contains general information about using BASIC. Some sections of the book provide help for those unfamiliar with BASIC. Other sections contain information on advanced subjects for the experienced programmer.

*IBM RT PC Pascal Compiler Fundamentals*
This book presents the fundamentals of the IBM RT PC Pascal Compiler Version 1.00.

*IBM RT PC Pascal Compiler Language Reference*
This book is designed as a reference tool and contains information about specific keywords, library routines, and metacommands used in writing programs.

*IBM RT PC FORTRAN 77*
This book is designed as a reference tool and discusses the implementation of FORTRAN 77.

*IBM RT PC Using and Managing the AIX Operating System*
This manual contains information for using the AIX Operating System commands, working with the file system, developing shell procedures, and performing such system management tasks as creating and mounting file systems, backing up the system and repairing file system damage.

*IBM RT PC AIX Operating System Commands Reference*
This manual describes the AIX Operating System commands, including the following:

- The proper syntax for each command with the acceptable flags and arguments

- Examples showing proper usage of the commands.

*IBM RT PC Messages Reference*
This manual lists the errors you may see on your display, and shows how to respond to the messages.

*IBM RT PC AIX Operating System Technical Reference*
This manual gives details about the AIX Operating System, the file system, files, special files, miscellaneous files, and writing device drivers.

*IBM RT PC Installing and Customizing the AIX Operating System*
This guide explains how to install the programs contained in the IBM RT PC AIX Operating System Licensed Program Product, as well as how to install other licensed programs products. This guide also explains how to customize the AIX Operating System.

*IBM RT PC User Setup Guide*
This guide contains information about identifying, unpacking, and connecting devices and testing the system setup.

## Ordering Additional Copies of This Book

To order additional copies of this publication (without the program diskettes), use either of the following sources:

- To order from your IBM representative, use Order Number SV21-8059.
- To order from your IBM dealer, use Part Number 55X8926.

A binder and language reference booklets for C, BASIC, FORTRAN, and Pascal are included with the order.

## About the IBM RT PC Professional Graphics Series

The IBM RT PC Professional Graphics Series consists of several compatible programs and libraries that provide you with a comprehensive graphics package. Designed to run under the IBM RT PC AIX Operating System, the series provides programmers with a versatile way to create, modify, store, and output high-quality graphic images.

Using the series, you can develop custom graphics applications that meet the needs of the business, scientific, engineering, and research environments.

# The Series Components

The IBM RT PC Professional Graphics series includes:

- IBM RT PC Graphics Development Toolkit
- IBM RT PC Plotting System
- IBM RT PC Graphical File System
- IBM RT PC Graphics Terminal Emulator.

The following pages briefly describe the series components. See the individual manuals for more information.

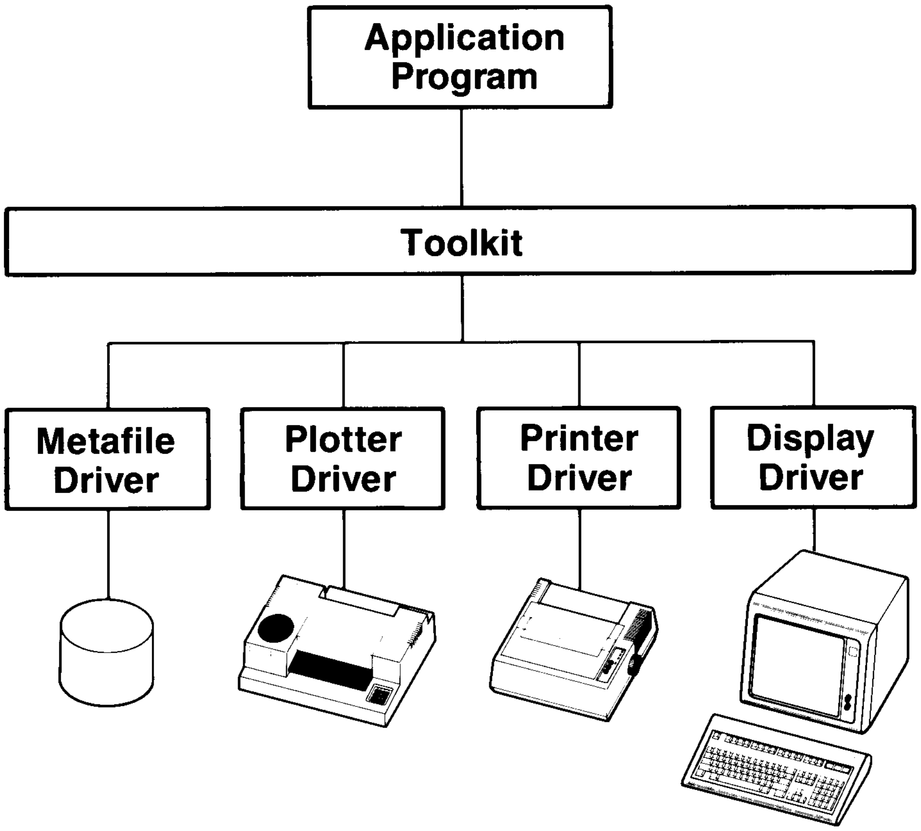
## IBM RT PC Graphics Development Toolkit

At the heart of the Professional Graphics Series is the Graphics Development Toolkit, including its device drivers. These device drivers, part of the IBM RT PC Multi-User programs, must be installed in order to use any of the other graphics products in the series. The Toolkit controls the exchange of information (commands and data) between the high-level, device-independent applications and the device-dependent drivers. These drivers, in turn, control the operation of a variety of input and output devices, such as printers, plotters, and displays.

The Toolkit converts coordinates from a writing surface to the viewing surface coordinate systems of individual graphics devices. In addition, it manages all supported devices connected to the system and fulfills program requests for device information.

A library of language *bindings* (interfaces) is also contained in the Graphics Development Toolkit. Using this product, you can develop application programs that access the Toolkit directly. Language Reference Booklets for C, Pascal, FORTRAN 77, and BASIC are provided with the software.

The following illustration shows the interface between an application program and the Graphics Development Toolkit.



## IBM RT PC Plotting System

Plotting System consists of subroutines that allow you to create professional-quality charts. The types of charts available include area, bar, line, pie, scatter, schedule, step and text-only charts.

Plotting System provides bindings to several major languages. The Toolkit, although transparent to the programmer, is the foundation of Plotting System.

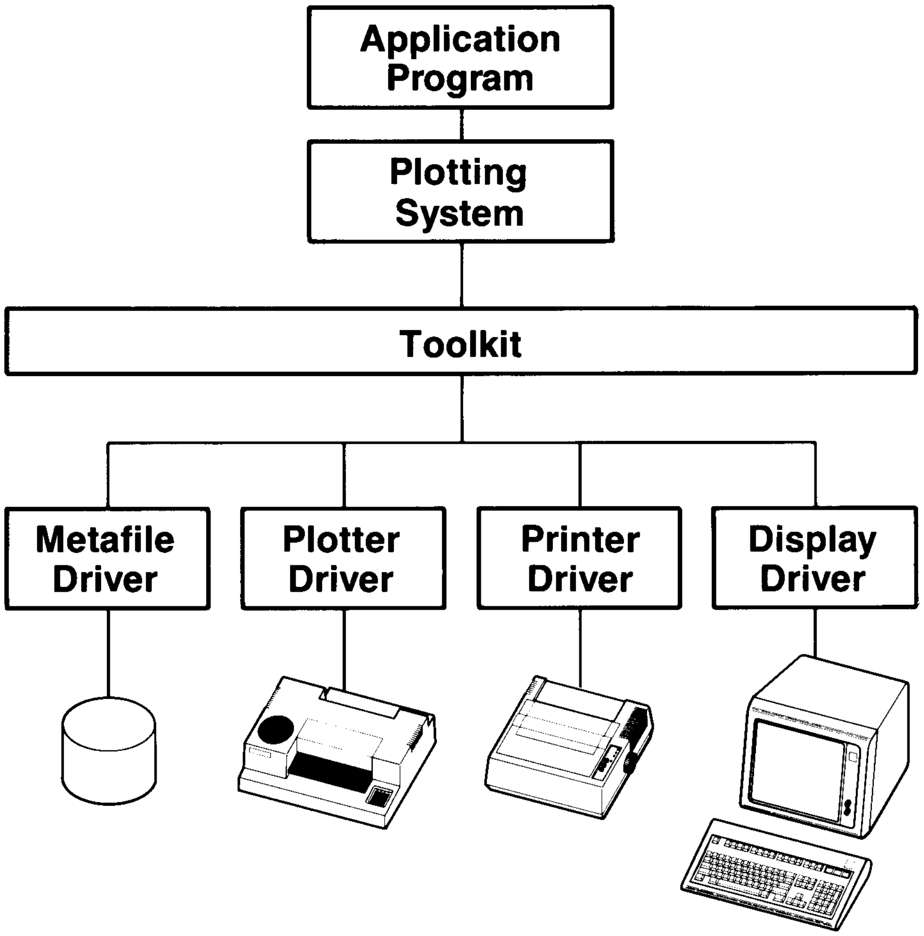
Because Plotting System provides a comprehensive set of default values, you can create charts with a minimum number of steps. A built-in chart dimensioning procedure ensures that the sizing and spacing of chart components are consistent and proportional. Multiple charts or chart types can appear on a single display surface.

Most chart attributes can be changed. These include:

- Line style for line and step charts
- Fill pattern for area, pie and bar charts
- Color
- Text height and font
- Logarithmic axes
- Axis range
- Tick-spacing.

Plotting System includes facilities for both graphic and text annotation. In addition, it contains device inquiry capabilities that allow you to use device-independent features. You can create interactive graphics applications by using input capabilities, such as choice, locator and string.

The following illustration shows an application program interface with Plotting System.



When programming with Plotting System, include calls to Plotting System routines in your application program. The external references are resolved when the compiled program is linked to the subroutine library.

To use Plotting System, you should be an experienced programmer familiar with the BASIC, C, FORTRAN or Pascal languages.

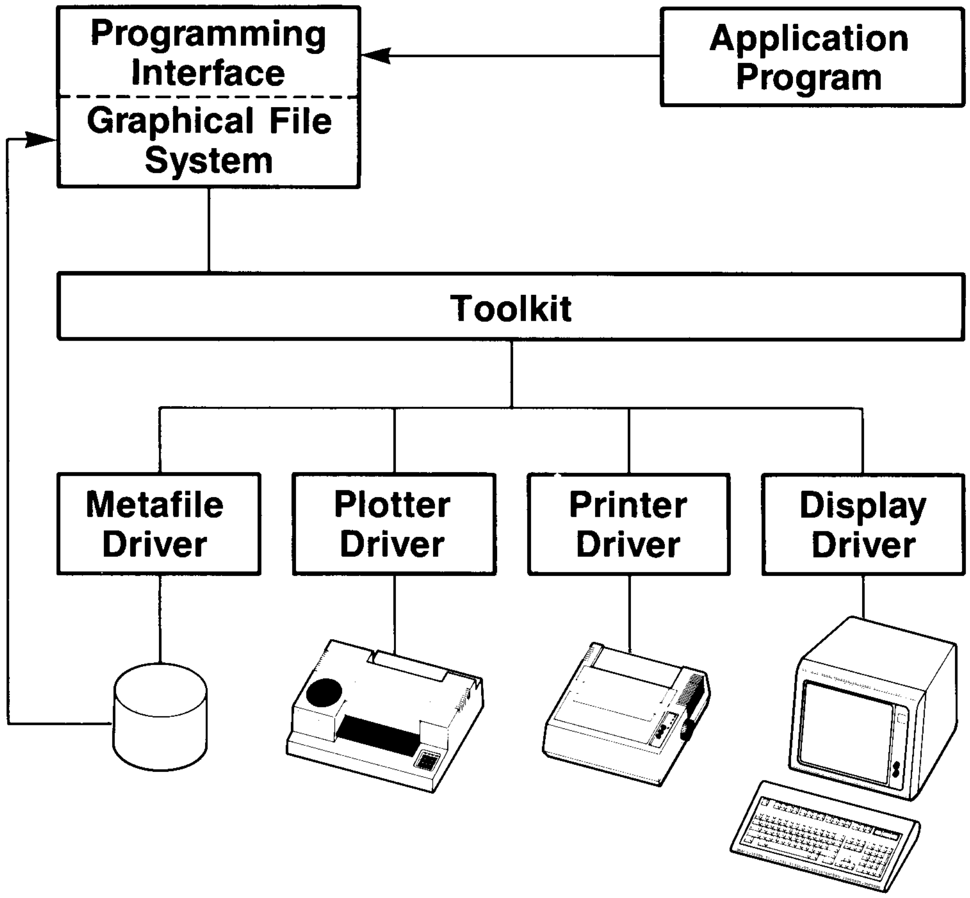
## IBM RT PC Graphical File System

The Graphical File System allows you to retrieve and interpret *metafiles* that have been generated by the Metafile Device Driver within the Graphics Development Toolkit. Each metafile contains a series of pictures or graphics images that have been stored in a special metafile format. This format promotes standardization and portability between different kinds of graphics output devices.

You can use the Graphical File System as a *programming interface* or an *interactive interface.*

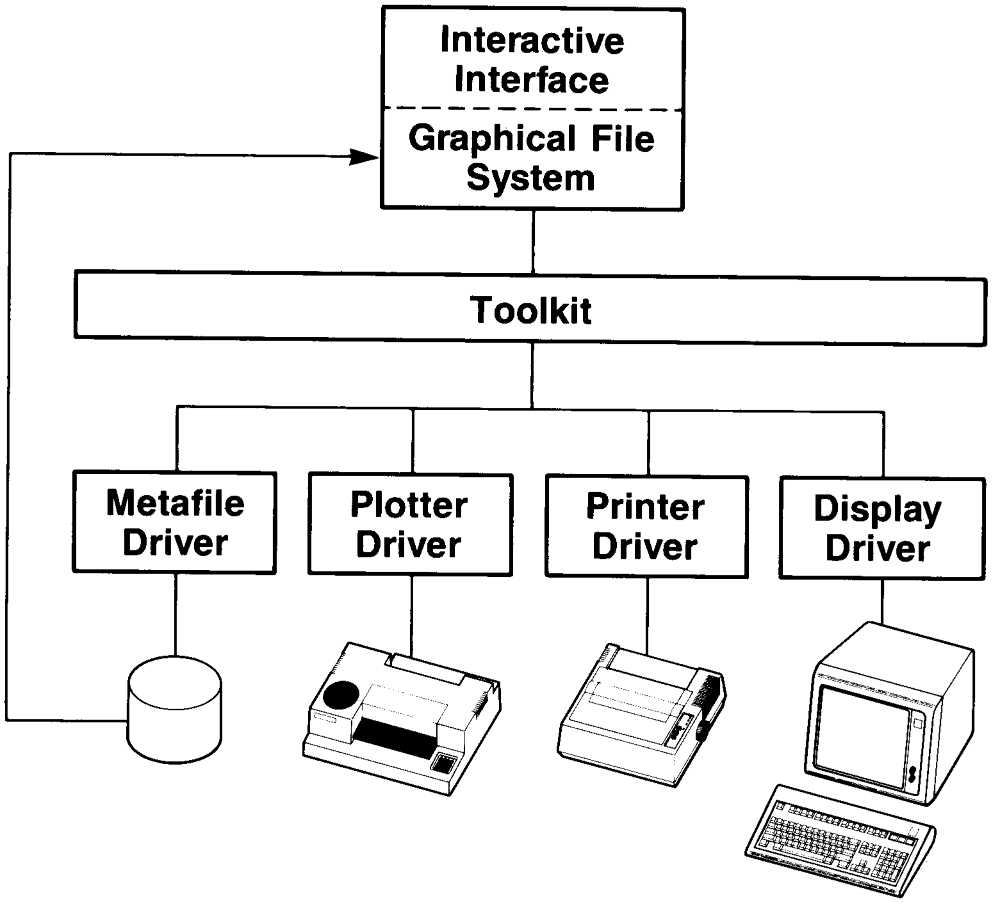
The programming interface consists of subroutines called by an application program. These subroutines are in the form of functions that retrieve metafile pictures and interpret individual *elements* within a picture. The programming interface requires programming experience in BASIC, C, FORTRAN or Pascal.

A diagram of the programming interface is shown in the following illustration.



The interactive interface, on the other hand, is an easy-to-use, icon-driven program. By selecting icons, you can perform metafile interpretation functions. To operate the interactive interface, you need no special knowledge of either graphics or programming.

The interactive interface is diagramed in the following illustration.



## IBM RT PC Graphics Terminal Emulator

Graphics Terminal Emulator allows you to emulate most of the functions of the Tektronix[2] 4014 graphics terminal, most functions of the Tektronix 4014 with the Extended Graphics Module, and selected features of the Tektronix 4105. Most functions of the Lear Siegler ADM[3]-3A are also supported.

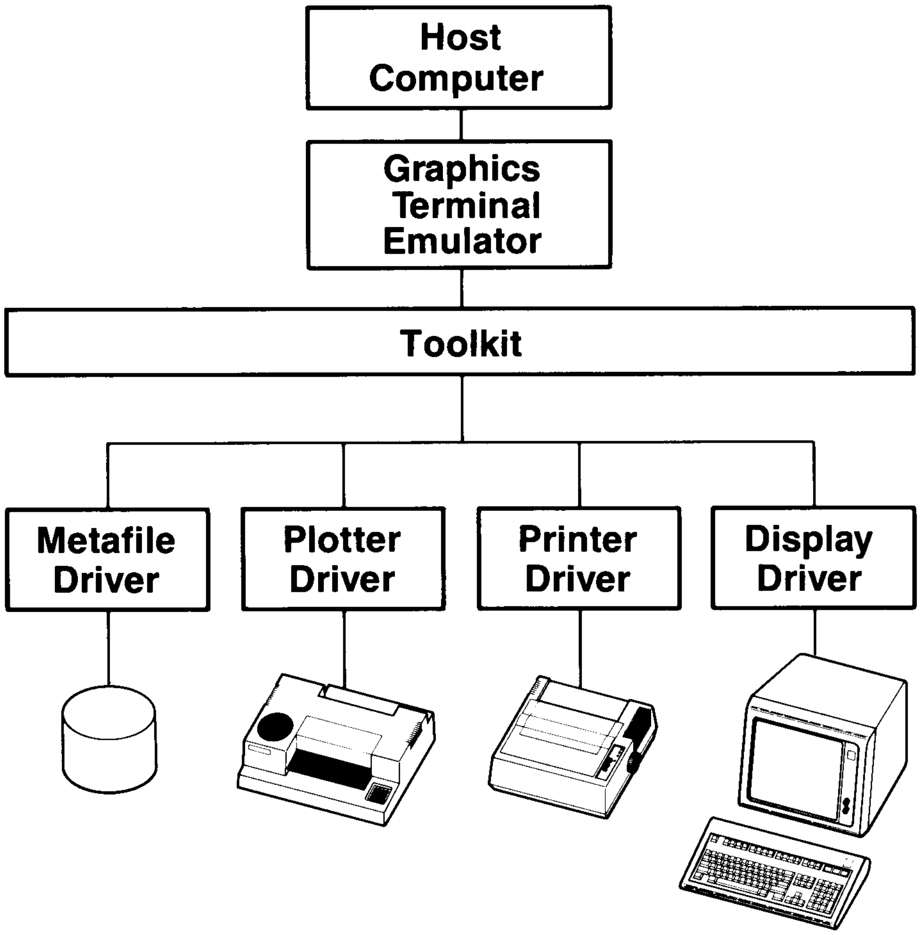
[2] Tektronix is a trademark of Tektronix, Inc.

[3] ADM is a trademark of Lear Siegler, Inc.

Graphics Terminal Emulator allows you to access host-computer software, send and receive files, and print both alphanumeric and graphic output. In addition, it reproduces the effects of the many special function keys available on the Tektronix family of supported terminals.

Because Graphics Terminal Emulator is an icon-driven system, no special knowledge of programming is required to use it. Experience on the supported terminals, however, is necessary to create graphics.

The operation of Graphics Terminal Emulator involves an external host computer, as shown in the following illustration.



# Contents

# Contents

# Chapter 1. Using Plotting System

# CONTENTS

# About This Chapter

Chapter 1 explains how to use Plotting System to generate charts. It provides a general overview of the features of Plotting System and describes how to use them to create many different charts. At the end of the chapter, pseudocode examples are presented to help you to write your code.

# Introduction

The IBM RT PC Plotting System can help you create many different types of charts. This chapter describes the features of Plotting System and provides some examples of its use.

Part of Plotting System's versatility is that it supports four programming languages: Compiled BASIC, C, FORTRAN 77, and Pascal. Specific routines in Plotting System have names in simple English (generic names) which are used to refer to the routines throughout the text. The same routines have different "syntax" names depending on the language in which the routine is being used.

For example, the four syntax names for the routine with the generic name of Open Plotting System are **popnps** in BASIC, **p_opnps** in C, **popnps** in FORTRAN, and **p_open_plotting_system** in Pascal. The routines are organized alphabetically within the Language Reference Booklets by their generic names.

# Plotting System Features

The following features can eliminate many tedious graphic programming tasks, providing you with the means to design several styles of charts:

- *Chart dimensioning* provides consistent and proportional sizing and spacing of chart components.
- *Chart components* and their related *attributes* can be controlled individually with little attention to the order in which they appear in the program.
- *Default values* provide the most appropriate choice when a chart component or attribute is not specified.

## Chart Dimensioning

As a graphics user, you rely on chart components to represent real-world relationships. The accuracy with which related components are sized and positioned determines the effectiveness of the graphics tool. For this reason, the Plotting System takes an interrelated approach to all chart dimensioning. Once you decide the height and width of a chart's display surface, all other dimensions are then set relative to the area defined.

Any portion of the physical device's display surface can be specified as the display surface for your chart. The view area is contained within the display surface. It contains the chart output plus its titles. Several charts can be created on a single display surface by specifying multiple view areas. The position and size of chart components are specified relative to the view area or the axis system of the chart itself.

For more information about chart dimensioning, see "Controlling the Plotting Environment" in this chapter. Examples showing the procedure to follow when dimensioning a chart are in "Pseudocode Examples." also in this chapter. The actual routines used to set chart components are described in the Language Reference Booklets.

## Chart Components

Plotting System provides a flexible structure and an efficient set of building blocks for chart creation. The following chart components and their related attributes are controlled individually and can be defined in any order.

- Data sets are numeric data that can be translated into graphic form such as a bar, line, or pie.
- Titles are separate character strings associated with a main chart title, chart subtitle, and axis titles. The height and font of each title can be specified.
- Labels include the axis tick labels that define unit divisions for the vertical and horizontal axes, legend labels, and the labels associated with each pie slice within a pie chart. The font and height of labeling text can be defined.
- Graphics primitives are lines, markers, circles, rectangles, arcs, and arrows added to improve the appearance or clarity of a chart.

- Notation is additional character strings that provide remarks about the chart. Notation can appear anywhere within the view area and in any available font and height.

- Frames of specifiable thickness can be set to enclose the view area or a specific chart area.

- Gridding can be generated within the area defined by chart axes. When called for by the program, grid lines appear at each axis tick mark.

For examples showing the procedure to follow when defining chart components, see "Pseudocode Examples" in this chapter. The actual routines used to set chart components are described in the Language Reference Booklets.

## Default Values

Plotting System's default set reduces possibilities for error and simplifies the task of writing source code. If no specifications are provided, the defaults will produce chart components which are sized and spaced in proportion to each other. The default set also assures that two different data sets charted on the same X/Y-axis system are distinguishable by assigning each a different primitive style.

The plotting environment, all defined data sets, and related attribute sets can be reset to the initial default conditions by calling the Reset Plotting System to Default routine. This routine is described in each of the Language Reference Booklets. Actual default values are listed in Appendix C, "Plotting System Defaults."

# Ordering Certain Routines

In most cases, the order in which you use the Plotting System routines in your programs is unimportant. However, the following are exceptions:

- The Open Plotting System routine must be called before any other Plotting System routine.

- The Close Plotting System routine must be the last routine called.

- An output device must be opened, using the Assign Plotting System Output Device routine, before any output is attempted. Output occurs when the Output Currently Defined Chart routine, one of the immediate action graphic primitive routines (Arc, Arrow, Circle, Polyline, Polymarker, or Rectangle), or the Define Notation String and Location routine is called.

- An input device must be opened, using the Assign Plotting System Input Device routine, before graphic input is attempted. (Graphic input is accomplished by calling the Request Choice, Request Locator, and Request String routines. This is the only time an input device must be opened.)

In addition, the output device must be opened before the input attempt if the input is to be echoed, as when using Request Locator or Request String. For example, by opening the display as both the input and output device and calling Request Locator, you can use the graphics cursor to position the tip and end points of an arrow.

- When the same device is opened for both input and output, the Close Input/Output Device routine must be called twice before the Plotting System is closed.

- Any chart attributes must be set before the Output Currently Defined Chart routine is called.

- Any primitive attributes must be set before the primitive routine is called, since primitives and notation are immediate action routines. For example, Set Notation Color must be called before Define Notation String and Location if the color is to have an effect on the notation string.

- If primitives are to be used in conjunction with a chart, the Output Currently Defined Chart routine should be called before the primitive routine. The chart area is cleared before the chart is output.

# Generating a Chart

The following text will explain the process of generating a chart using the Plotting System routines.

## Initializing Plotting System

The first Plotting System routine included in your program must be the Open Plotting System routine. This control routine initializes Plotting System to default conditions.

The default values simplify coding and reduce the possibility of error. Using the default values, you can generate a chart with the minimum amount of code. The default values remain in effect until you change them by using Plotting System routines. Appendix C, "Plotting System Defaults", contains a list of all Plotting System default values and the routines used to change the default values.

## Assigning Input/Output Devices

*Control routines* are used to control the Plotting System environment. This section discusses control routines that are used to assign, open, and close the input and output devices. For information about other control routines, see "Controlling the Plotting Environment" in this chapter. For a complete list of control routines, see "List of Routines by Function" in Chapter 3.

An output device must be assigned before output can occur. You assign an output device with the Assign Plotting System Output Device routine. Assign a string with a maximum of eight characters to each device. The string must correspond with the variables set in the environment. Refer to "Setting Environmental Parameters" in Appendix A for more details.

If your application requires user input, you must assign an input device with the Assign Plotting System Input Device routine before any input is attempted.

When appropriate, the same device can be specified for both graphics input and output.

If you are using a display for output, the Clear Display Surface routine clears the screen. If you are using a plotter for output, this routine prompts for new paper. If you are using a printer for output, this routine displays all pending graphics and advances to the top-of-form setting.

The Close Input/Output Device routine must be called:

- To close any open output device before another output device is opened
- To close any open input device before another input device is opened.

**Note:** It is not necessary to close an output device before opening an input device.

## Defining the Data Sets

The next step in generating your chart is defining the necessary data sets. Plotting System requires that the numeric data be translated into graphic representation – a bar, a line, a pie, a step, a scatter, or a schedule – and be stored in one of three types of data sets:

- X/Y data sets
- Sequenced data sets
- Schedule data sets.

Four data set definition routines are used to define or undefine data sets. Three of these routines define the data sets listed above. The fourth can be used to undefine, or erase, any data set.

Each data set is identified by a unique data set identification number. You assign the identification number when you define a data set. If the same number is assigned to more than one data set, only the most recent assignment is retained.

The following sections explain the data set definition routines.

### X/Y Data Sets

Two arrays are required to supply values for X/Y data sets: one array contains the independent X-axis coordinates, the other contains the dependent Y-axis coordinates. Each element in the array of X-axis coordinates is associated with exactly one element in the array of Y-axis coordinates.

The Define X/Y Data Set routine defines an X/Y data set. Four parameters are used to define an X/Y data set:

- The data set identification number
- A variable indicating the number of elements in each array
- The name of the array of X-axis coordinates
- The name of the array of Y-axis coordinates.

X/Y data sets usually are used to generate X/Y-axis charts, such as area, bar, line, scatter, and step charts. If a pie chart is defined, only the Y data is used.

When generating a vertically oriented chart, the independent X-array is plotted relative to the horizontal, or X, axis. When generating a horizontally oriented chart, the independent X-array is plotted relative to the vertical, or Y, axis.

## Sequenced Data Sets

Sequenced data sets are indexed by a sequence of integers. The sequence number begins with 1 and increases by 1, until the last number in the sequence is equal to the number of items in the data set.

The Define Sequenced Data Set routine defines a sequenced data set. It requires three parameters:

- The data set identification number
- A variable indicating the number of elements contained in the array
- An array of data points.

Though sequenced data sets can be used to generate area, line, bar, scatter, and step charts, they usually are used to generate pie charts. Each slice of the pie is associated with a different element in the sequence.

When a pie chart is plotted, Plotting System evaluates each element within the array as a percentage of the whole and defines the pie slice accordingly. If an X/Y chart is being generated, the independent axis scale is (1,2,. . .,n).

## Schedule Data Sets

Schedule data sets are used to generate schedule charts. Each bar of the schedule chart requires a unique schedule data set.

The Define Schedule Chart Data Set routine defines a schedule data set. Four parameters are needed to define a schedule data set:

- The data set identification number
- A character string representing the start date of the scheduled task
- A character string representing the end date of the scheduled task
- A variable specifying the percent completion of the scheduled task.

## Undefining a Data Set

Plotting System provides a fourth data set definition routine, Undefine a Data Set, which erases all memory of a data set. It is invoked when you wish to delete a currently defined data set.

The following tables provide a quick reference to each of the data set types.

| Data Set Types | | | |
|---|---|---|---|
| **Chart Type** | **X/Y Data Set** | **Sequenced Data Set** | **Schedule Data Set** |
| Bar chart | Independent and dependent data; charted as a set of bars. | Dependent data is charted as a set of bars; independent data implies numeric order. | No chart output. |
| Line chart | Independent and dependent data; charted as a line. | Dependent data is charted as a line; independent data implies numeric order. | No chart output. |
| Pie chart | Dependent data; charted as a pie. Pie slices are in numeric order and slice numbers become the identifier for slice attributes. | Dependent data; charted as a pie. Pie slices are in numeric order and slice numbers become the identifier for slice attributes. | No chart output. |
| Scatter chart | Independent and dependent data; charted as a set of markers. | Dependent data is charted as a set of markers; independent data implies numeric order. | No chart output. |
| Schedule chart | No chart output. | No chart output. | Dates and percentage complete data are charted as a schedule bar. |
| Step chart | Independent and dependent data; charted as a stepped line. | Dependent data is charted as a stepped line; independent data implies numeric order. | No chart output. |

## Assigning the Chart Type

Now that your data set is defined, you need to assign the chart type to the data set.

Plotting System routines are available to define bar, line, pie, scatter, schedule, and step charts. These routines are attribute set definition routines.

When you want to generate a bar chart, call the Define Data Set Chart Type to be Bar routine.

When you want to generate a line chart, call the Define Data Set Chart Type to Be Line routine.

When you want to generate a pie chart, call the Define Data Set Chart Type to Be Pie routine.

When you want to generate a scatter chart, call the Define Data Set Chart Type to Be Scatter routine.

When you want to generate a schedule chart, call the Define Data Set Chart Type to Be Schedule routine.

When you want to generate a step chart, call the Define Data Set Chart Type to Be Step routine.

When you want to generate an area chart, call either the Define Data Set Chart Type to Be Line or the Define Data Set Chart Type to Be Step routine. Then, identify the area to be filled with the Fill between Two Data Sets routine.

For a complete list of attribute set definition routines, see “Attribute Set Definition Routines” in Chapter 3.

If you want to generate a text-only chart, no data sets are defined. Independent chart component routines are used to create text-only charts. For information on generating text-only charts, see “Using Independent Chart Components” in this chapter.

## Planning the Chart

Next, the appearance of your chart should be considered, which will be affected by several aspects of the chart design. These include:

- Size of the actual plotted data, such as bars and lines, without associated text
- Proportion of the chart to the page
- Proportion of the chart to the associated text
- Position of the chart on the plotting surface
- Composition of all chart components.

A rough sketch of the desired chart may help you anticipate and correct potential problems with your design. When planning your chart remember that the appearance depends on these factors:

- Shape of the chart
- Width of the lines
- Size of the text
- Colors assigned to the foreground and background elements
- Textures created by line style and fill patterns.

You also need to consider which areas of the chart are specified by control routines, which by attribute set definition routines, which by text definition routines, and which by independent chart component routines. Figure 1-1 illustrates all four.

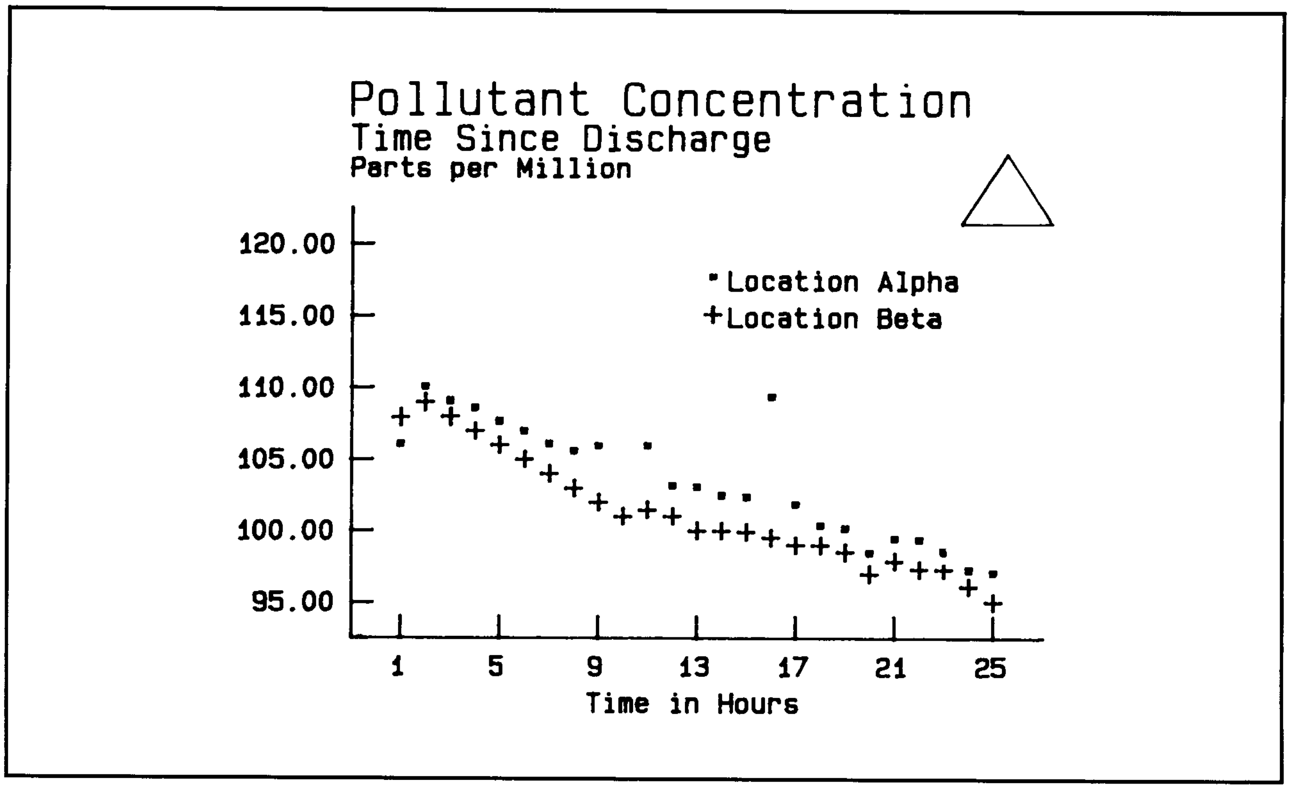


**Figure 1-1. Chart Definition Routines**

The axis extents (size), color, and chart orientation (horizontal or vertical) are defined using control routines. These routines determine the basic characteristics of a chart by controlling the Plotting System environment.

The type of chart is defined using attribute set definition routines. This chart is a scatter chart.

The contents of the titles and the appearance of the text is defined using text definition routines. These routines control the font type, height, alignment and location of the titles and legend accompanying the chart.

The chart's polymarker is defined using independent chart component routines. The polymarker's color and style are defined by these routines. These routines control the appearance of notations and graphic primitives, such as the triangle in the upper right corner of the chart in Figure 1-1.

See "List of Routines by Function" in Chapter 3 for a complete list of each routine type.

## Controlling the Plotting Environment

Plotting System divides the physical display space of a device into three areas:

- Display surface
- View area
- Chart area.

Plotting System provides routines to control the appearance of these areas with Control routines.

### Display Surface

The display surface is the portion of the physical output device on which output from one or more charts is displayed. The size of the display surface is specified in inches or millimeters. The Set Display Surface Units routine indicates whether the display surface dimensions are defined in inches or millimeters. The Set Display Surface Size routine specifies the height and width of the display surface.

Once the dimensions of the display surface are established, all other dimensions are set relative to these dimensions. By default, the display surface is the entire physical device display area. A smaller display surface can be specified using Set Display Surface Size.

The (0.0, 0.0) point of the display surface is the lower-left corner of the device space. If the specified display surface size dimensions are larger than the physical device surface, Plotting System scales down the display surface size, but retains the specified height-to-width ratio. For example, if the available device surface is 8 × 12 inches and the requested display surface is 10 × 10 inches, the resulting display surface size is 8 × 8 inches. Figure 1-2 shows a defined display surface.

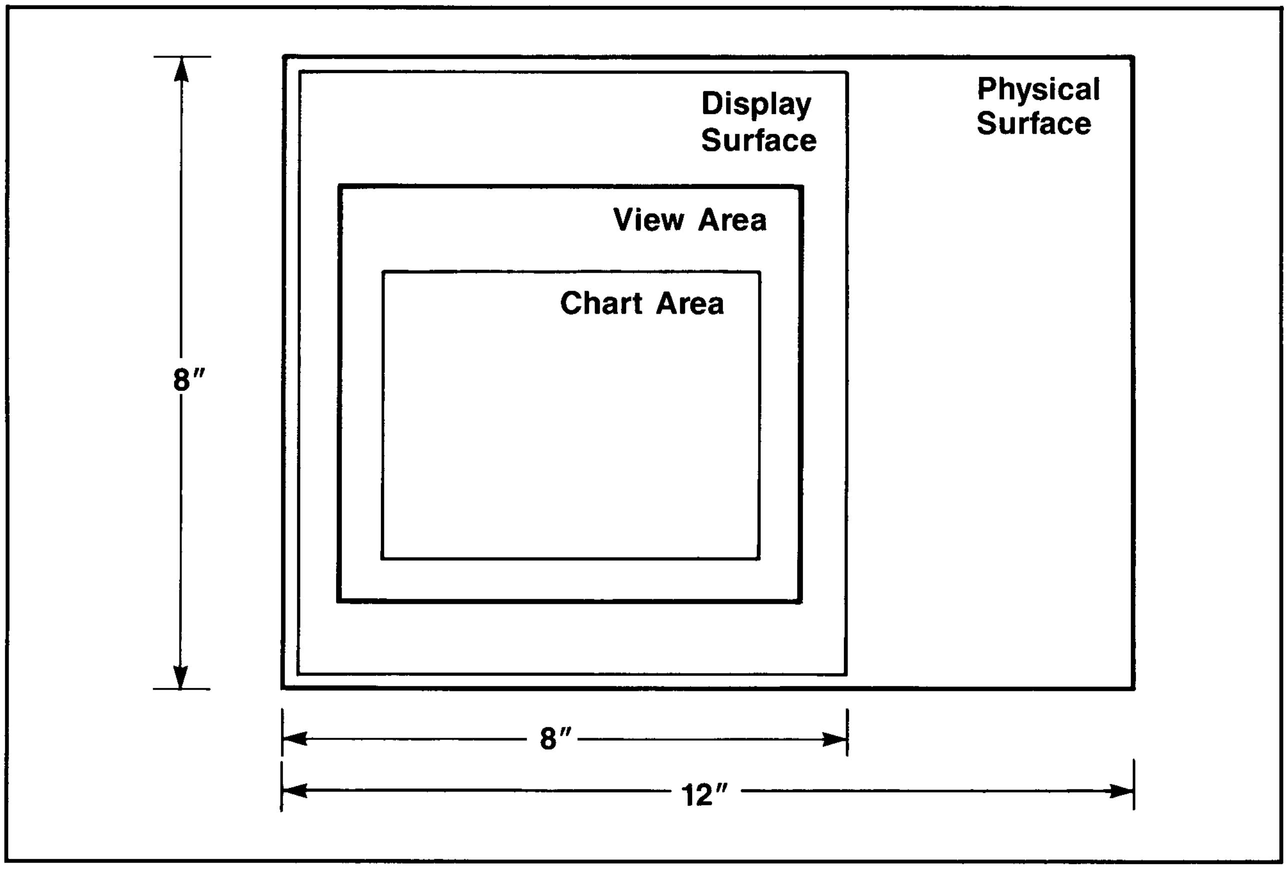


**Figure 1-2. Display Surface**

## View Area

The view area is the portion of the display surface designated for the plotted data of a chart and its title and legend. By default, the view area fills the entire display surface.

You can change the dimensions of the view area by invoking the Set View Area Extents routine. View area extents are defined as a percentage of the display surface. X and Y coordinates are entered for the lower-left corner and upper-right corner of the desired view area.

The referenced coordinate space sets the (0.0, 0.0) point at the lower-left corner of the display surface and the (100.0, 100.0) point at the upper-right corner of the display surface.

By defining view area extents, you can reduce the size of the chart in relation to the display surface or create multiple charts on a single display surface.

You should consider the desired shape and size of your output when establishing the dimensions of the display surface and view area. For example, if the specified output device is a plotter with 14 × 11-inch paper, the display surface and view area could be as large as 14 × 11.

However, if you are creating a chart for insertion into a document, you may want to reduce the size of the display surface to 11 × 8½ inches. If you want to place two charts on the same page, you may want to specify two view areas, each 5½ × 8½ inches, for this display surface. See "Generating a Bar and a Pie Chart" in this chapter for an example of how this is done.

## Chart Area

The chart area is the portion of the view area that contains the plotted data.

Because both the graphic and text components of a chart must be contained within the view area, the space requirements of chart titles, axis titles, and other text determine the size of the chart area.

All output is contained within the view area and all components of the chart should appear in their entirety and in relative proportion. For this reason, the percentage of the view area available for the chart area is determined by the space needed for text output, such as the chart title, subtitle, and axis titles. In other words, the larger the text, the smaller the available chart area space.

If no text output is defined, the chart area fills the entire view area. If titles are specified, the chart area is scaled down to permit the titles to be printed within the view area.

In addition to changing the text size, another way to change the size of the chart area is to change the size of the view area. This is done with the Set View Area Extents routine.

## View and Chart Area Frames

A frame of definable thickness can be drawn around the view area with the Set Existence of View Area Frame routine.

For X/Y-axis and schedule charts, you can specify a visible frame of definable thickness with the Set Existence of Chart Area Frame routine.

## Foreground and Background Colors

Plotting System contains routines that allow you to specify foreground and background colors for your chart. However, the number of color options available depends on your output device.

The Set Color Representation routine allows you to define your own colors according to the output device you are using.

The Set Background Color routine is used to define the background color for the chart.

The Set Foreground Color routine controls the color of the title, subtitle, legend, pie slice labels, view area frame, chart area frame, axis tick marks, axis tick labels, axis titles, grid lines, and filled area outline for data sets and pie slices.

## Displaying the Axes, Labels, and Grid Lines

The visibility of chart axes is specified by the Set Existence of Axis routine. For charts with defined axes, you can specify axis labels and axis tick labels with the Set Axis Labels and Set Axis Tick Label Type routines, respectively.

The extents of the axes are specified with the Set Axis Extents routine. Either linear or logarithmic axes can be specified with the Set Axis Type routine. The Set Axis Indent routine specifies whether data on either axis is to be indented.

You have the option of setting visible gridding with the Set Existence of Grid Lines routine. Grid lines that extend from each tick mark can be specified for either or both axes.

## Vertical or Horizontal Orientation

For X/Y-axis charts, you have the option of specifying a vertical or horizontal chart orientation. The Set Vertical or Horizontal Chart routine is used to specify the orientation. A vertical orientation is the most common form. This orientation has the data placed from left to right. The horizontal orientation arranges data from bottom to top.

### Assigning a Baseline

On an X/Y-axis chart, data is usually plotted relative to the independent axis. When a baseline is specified with the Set Baseline routine, data is plotted relative to the baseline.

A deviation bar chart is created when a baseline is specified for a bar chart. The baseline is used for the base value for all bars. Sharp data deviations, such as profit/loss or negative/positive differences, are easily recognized when plotted on opposite sides of a baseline value. All stacking of data sets occurs from the baseline.

## Defining the Attribute Set

The characteristics of your plotted data are defined by attribute set definition routines. For example, one attribute set definition routine indicates the line style of a line representing a data set on a line chart and another routine specifies the color of the line.

### X/Y-Axis Chart Attributes

When an X/Y-axis chart is defined, one set of attributes is associated with each data set to be represented on the chart.

The unique number that identifies each data set is used to link the data set to an attribute set. The attribute set specifies the characteristics of the data set when it is plotted on an X/Y-axis chart.

A character name can be associated with the attribute set of an X/Y data set with the Name Attribute Set routine. This character string becomes the legend name for the X/Y-axis chart.

### Pie Chart Attributes

When a pie chart is defined, up to 16 sets of attributes are associated with the data set to be represented on the chart. Each attribute set specifies the characteristics of one slice of the pie chart. Each attribute set is associated with a variable from 1 to 16.

A character string can be assigned to each attribute set associated with a pie chart by the Name Attribute Set routine. The character strings are mapped one to one with each pie slice and are used as pie slice labels.

If a pie chart is defined for a view area, no other type of chart can be output for the same view area.

### Set Explode of Pie Slice

The Set Explode of Pie Slice routine calls attention to a particular pie slice by moving it away from the center of the pie by a distance of 20 percent of the radius. The attribute set identification number is used to identify the slice to be exploded.

## Schedule Chart Attributes

A total of two attribute sets are associated with all defined schedule data sets. Attribute set 1 specifies the characteristics of the segments of the bars which represent the completed portion of a task. Attribute set 2 specifies the characteristics of the segments of the bars that represent the portion to be completed. Attribute sets 1 and 2 together determine the appearance of the bars on the schedule chart which represent the schedule data sets.

A character string can be assigned to each of the attribute sets associated with a schedule data set with the Name Attribute Set routine. The names appear in the legend area of the schedule chart.

## Attribute Destinations

When defining attributes, remember that the destination of the attribute differs according to the specified chart type.

When a line chart is defined, the destination of the attribute is the line representing the identified data set.

When a bar chart is defined, the destination of the attribute is the set of bars representing the identified data set.

When a step chart is defined, the destination of the attribute is the stepped line representing the identified data set.

When a scatter chart is defined, the destination of the attribute is the set of markers representing the identified data set.

When a pie chart is defined, the destination of the attribute is the pie slice representing the identified element of the data set.

When a schedule chart is defined, the destination of the attribute is the portion of the bars representing the complete or to-be-completed portion of the task.

### Defining Attributes

The appearance of the *output primitives* composing each type of chart can be varied by defining the attribute sets.

### Defining the Appearance of Output Primitives

An output primitive is a line, bar, marker, pie slice, or fill area. Four *attribute set definition routines* define the appearance of the output primitives used to represent data values. These routines assign an output primitive characteristic to an attribute set.

Color is assigned to an attribute set with the Set Output Primitive Color for Data Set routine. The specified color is used for:

- Fill color for pie, bar, and area filling
- Line color for line and step charts
- Marker color for scatter charts.

Outline is assigned to an attribute set with the Set Output Primitive Outlining for Data Set routine. The specified outline is used for:

- Outlining bars for bar and schedule charts
- Outlining pie slices for pie charts.

Style is assigned to an attribute set with the Set Output Primitive Style for Data Set routine. The specified style is used for:

- Line style for line and step charts
- Fill style for bars for bar and schedule charts
- Fill style for pie slices for pie charts
- Fill style for area charts
- Marker style for scatter charts.

Width is assigned to an attribute set with the Set Output Primitive Width for Data Set routine. The specified width is used for:

- Bar width for bar and schedule charts
- Line width for line charts.

Other data attribute definition routines control the appearance of data sets when plotted in conjunction with other data sets.

### Stacking Data Sets

By calling the Stack a Set of Data Sets routine, data sets can be stacked above each other on X/Y-axis charts.

The Stack a Set of Data Sets routine allows you to create a stacked bar, line, step, or scatter chart. You can build more than one stack, each stack composed of a number of data sets. The input required by this routine includes an array that lists the identifiers for each data set to be stacked, the number of elements in that array, and a new identifier that is used to reference the stack.

The order given to the data sets within the array that defines the stack is maintained in the stack itself; the first array element is at the base of the stack. For example, if the stack definition array listed the sequence of data sets as 1, 3, and 5, the resulting stack shows data set 1 at the bottom, data set 3 in the middle, and data set 5 on top. When multiple stacks are defined, the stacks are displayed in ascending order according to the stack number, as in Figure 1-3.

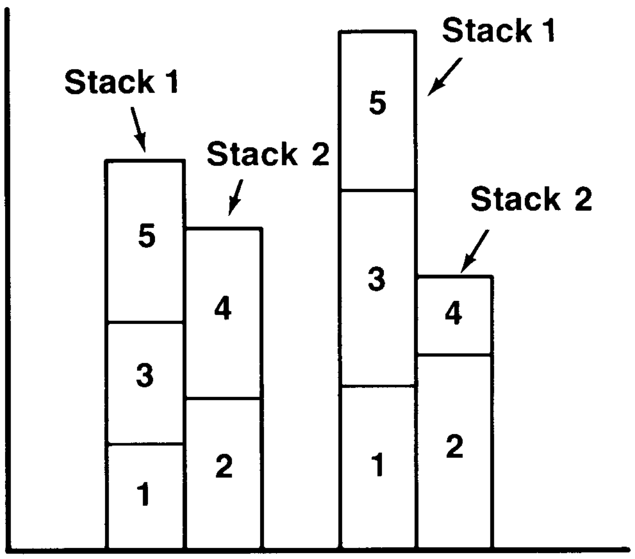


**Figure 1-3. Stacking Data Sets**

### Filling between Data Sets

The Fill between Two Data Sets routine is used to create an area chart. This routine causes an area defined by a set of boundaries to be filled with a solid color or shading pattern. The data sets that establish the boundaries must be defined as either line or step charts to create the necessary solid lines.

### Data Set Visibility

If you do not want a data set to be visible on the chart, such as when a floating bar effect is desired, define several data sets that can be stacked on top of each other. A data set can then be made invisible by invoking the Set Visibility of Data Set routine.

If you wish to show select portions of a pie chart, you can use the Set Visibility of Data Set routine to set the visibility of individual pie slices.

The following tables explain how the attribute set definition routines discussed in the preceding pages affect the various chart types.

| Effect of Attribute Routines on Chart Types | | | |
|---|---|---|---|
| **Attribute Routines** | **Bar Chart** | **Pie Chart** | **Line Chart** |
| Name Attribute Set | Assigns character string to set of bars. Name appears in chart legend. | Assigns character string to pie slice. Name appears adjacent to pie slice. | Assigns character string to line. Name appears in chart legend. |
| Fill between Two Data Sets | N/A | N/A | Identifies data set as one area boundary. Fill style and color are assigned by first data set. |
| Set Output Primitive Color for Data Set | Assigns fill color of bars. | Assigns fill color of pie slice. | Assigns line color. Also assigns fill color when Fill between Two Data Sets has been called. |
| Set Output Primitive Style for Data Set | Assigns fill style of bars. | Assigns fill style of pie slice. | Assigns line style. Also assigns fill style when Fill between Two Data Sets has been called. |

**Effect of Attribute Routines on Chart Types**

| Attribute Routines | Scatter Chart | Schedule Chart | Step Chart |
|---|---|---|---|
| Name Attribute Set | Assigns character string to distributed markers. Name appears in chart legend. | Assigns character string to either complete or to-be-completed portion of schedule bar(s). Name appears in legend. | Assigns character string to stepped line. Name appears in legend. |
| Fill between Two Data Sets | N/A | N/A | Identifies data set as one area boundary. Fill style and color are assigned by first data set. |
| Set Output Primitive Color for Data Set | Assigns marker color. | Assigns fill color for either complete or to-be-completed portion of schedule bar(s). | Assigns color for stepped line. Assigns fill color when Fill between Two Data Sets has been called. |

**Effect of Attribute Routines on Chart Types**

| **Attribute Routines** | **Bar Chart** | **Pie Chart** | **Line Chart** |
|---|---|---|---|
| Set Output Primitive Width for Data Set | Assigns bar width. | N/A | Assigns line width. |
| Set Output Primitive Outlining for Data Set | Sets outlining for bars in foreground or primitive color. | Sets outlining for pie slice in foreground or primitive color. | N/A |
| Set Explode of Pie Slice | N/A | Indicates pie slice should be moved away from center of pie. | N/A |
| Set Visibility of Data Set | Sets visibility of bar data set. Creates floating bar effect when bars are stacked. | Sets visibility of pie slice. | Sets visibility of line data set. Invisible line determines position of next visible line when lines are stacked. |
| Stack a Set of Data Sets | Stacks specified sets of bars. | N/A | Stacks specified lines. |

**Effect of Attribute Routines on Chart Types**

| **Attribute Routines** | **Bar Chart** | **Pie Chart** | **Line Chart** |
|---|---|---|---|
| Set Output Primitive Width for Data Set | N/A | Assigns schedule bar width. | Assigns width of stepped line. |
| Set Output Primitive Outlining for Data Set | N/A | Sets outlining for either complete or to-be-completed portion of schedule bar(s) in foreground or primitive color. | N/A |
| Set Explode of Pie Slice | N/A | N/A | N/A |
| Set Visibility of Data Set | Sets visibility of set of markers. Invisible marker set determines position of next visible set when scatter charts are stacked. | N/A | Sets visibility of stepped line. Invisible stepped line determines position of next visible stepped line when stacked. |
| Stack a Set of Data Sets | Stacks specified markers. | N/A | Stacks specified stepped lines. |

## Defining the Titles

Plotting System provides *text definition routines* to enable you to define the chart titles. The Define Title String routine defines the chart title and the Define Subtitle String routine defines the chart subtitle.

Figure 1-4 shows a chart with a title, a subtitle, and axis titles.

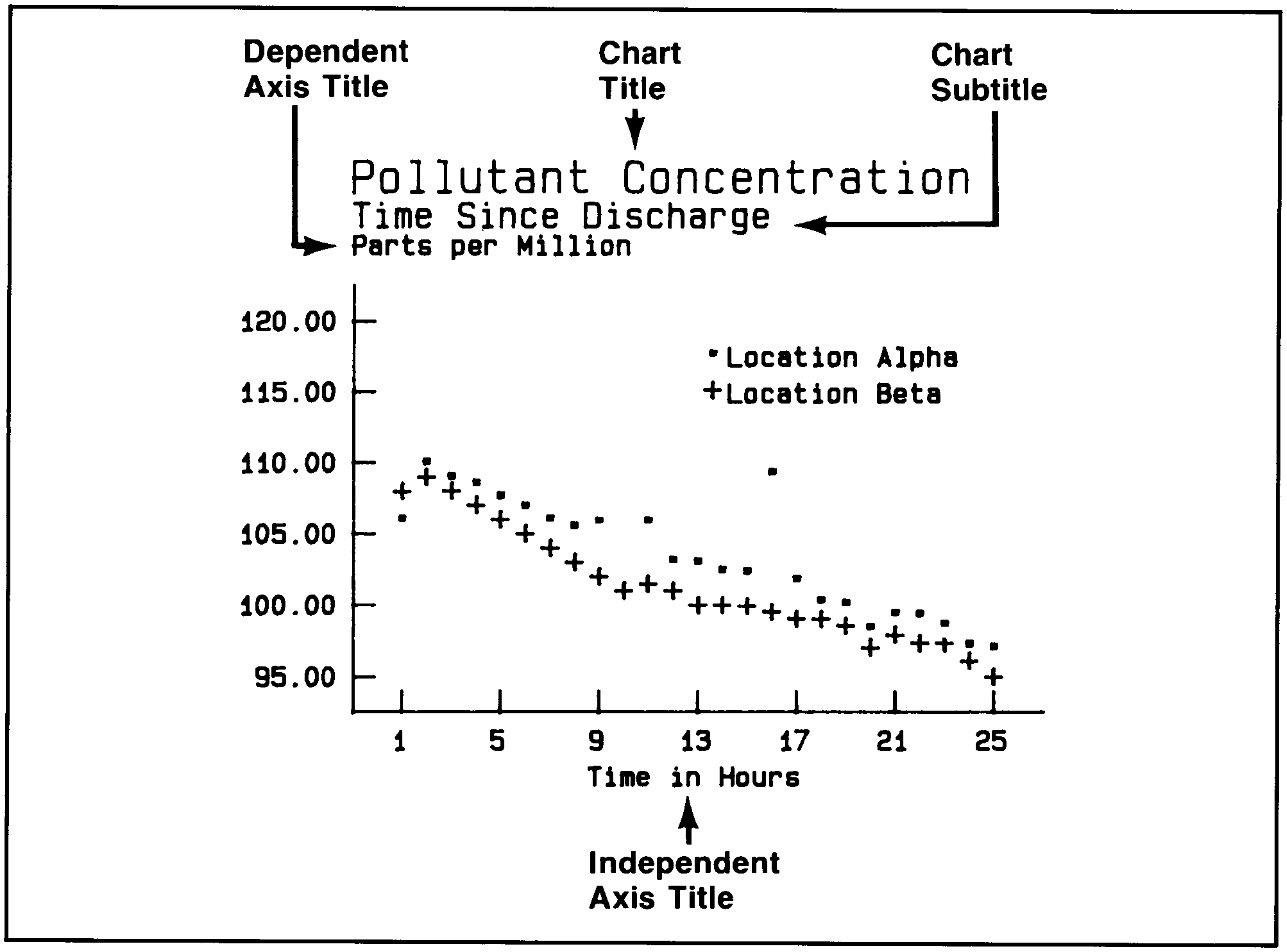


**Figure 1-4. Chart Titles**

Character strings defined as titles for pie charts are centered at the top of the viewing area. Character strings defined as titles for other chart types appear left-justified over the vertical axis. Subtitles appear directly below the title.

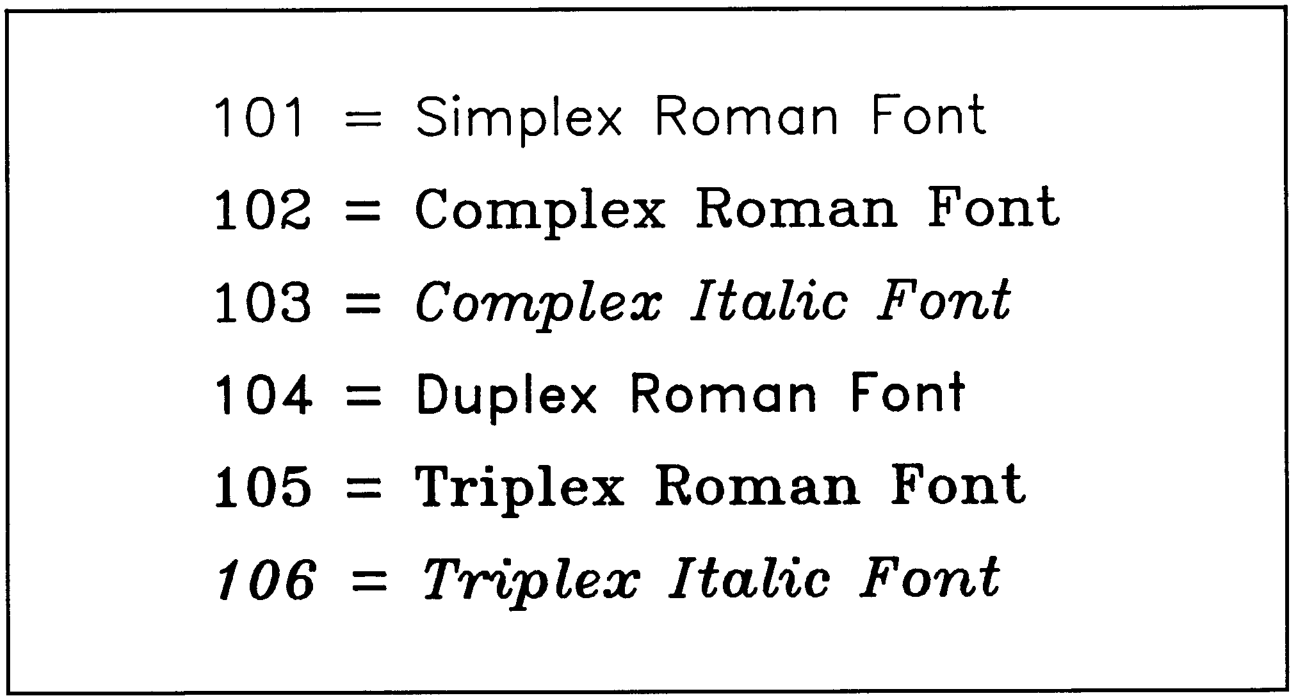
Axis titles can be defined for X/Y-axis charts with the Define Axis Title String routine. The title for the vertical axis is left-justified above the chart. The title for the horizontal axis is centered below the axis line.

The height and font of the title, subtitle, and axes titles can be specified using text definition routines.

The title, subtitle, and axis title heights are specified by the Set Title Height, Set Subtitle Height, and Set Axis Title Height routines. If the axis title height is not specified, the default value is five percent of the height of the view area. If the subtitle height is not specified, the default value is eight percent of the view area. If the title height is not specified, the default value is ten percent of the height of the view area.

The title, subtitle, and axis title fonts are specified by the Set Title Font, Set Subtitle Font, and Set Axis Title Font routines. The hardware font or one of six software fonts shown in Figure 1-5 can be specified. Plotting System will look in the current directory for a specified font. If the font files are contained in a different directory, you can direct Plotting System to that directory by setting an environment variable. Refer to “Setting Environmental Parameters” in Appendix A for more detailed instructions.

If the font is not specified, the hardware font is used. Figure 1-5 shows examples of the Plotting System fonts.



101 = Simplex Roman Font

102 = Complex Roman Font

103 = *Complex Italic Font*

104 = Duplex Roman Font

105 = Triplex Roman Font

*106 = Triplex Italic Font*

**Figure 1-5. Plotting System Fonts**

Plotting System's default values cause the titles to be written without any overlapping. If a specified title is too long to fit within the viewing area, Plotting System rescales the title. Plotting System adheres as closely as possible to your specifications for title or subtitle height.

To maintain appropriate chart proportions, Plotting System first fits the defined subtitle string into the vertical view area space, assuring that at least 1/3 of the space is still available for the chart area, within which all graphic data output will be contained. If the defined subtitle does not fit, all currently defined text strings (titles, axes titles, axes labels) are scaled to a size that enables all text to be displayed.

After resolving the vertical spacing, Plotting System evaluates the horizontal dimensions of the subtitle in relation to other currently defined text strings and the required chart area size. Text is scaled to a size that leaves at least half of the horizontal dimension of the view area available for the chart area. If text size is at a minimum and the defined subtitle still does not fit, Plotting System positions the subtitle to accommodate as much of the string as possible. Any chart elements which are too large to be contained within the view area are clipped.

Since most devices are not capable of scaling text to any size when using the hardware font, you should be aware of the available options of your device when defining title strings.

All output is contained within the view area; this includes all text output as well as the chart area that contains all the graphic data output. Thus, the space requirements of the chart title, subtitle, and axis titles affect the size of the actual plotted data.

The length and height of the title strings determine the percentage of the output space available for the chart area. As shown in Figure 1-6, when title strings are not specified, the chart area fills the entire view area. When titles are specified, the chart area is scaled down to a size that allows all title strings to be displayed without overwriting characters, as shown in Figure 1-7.

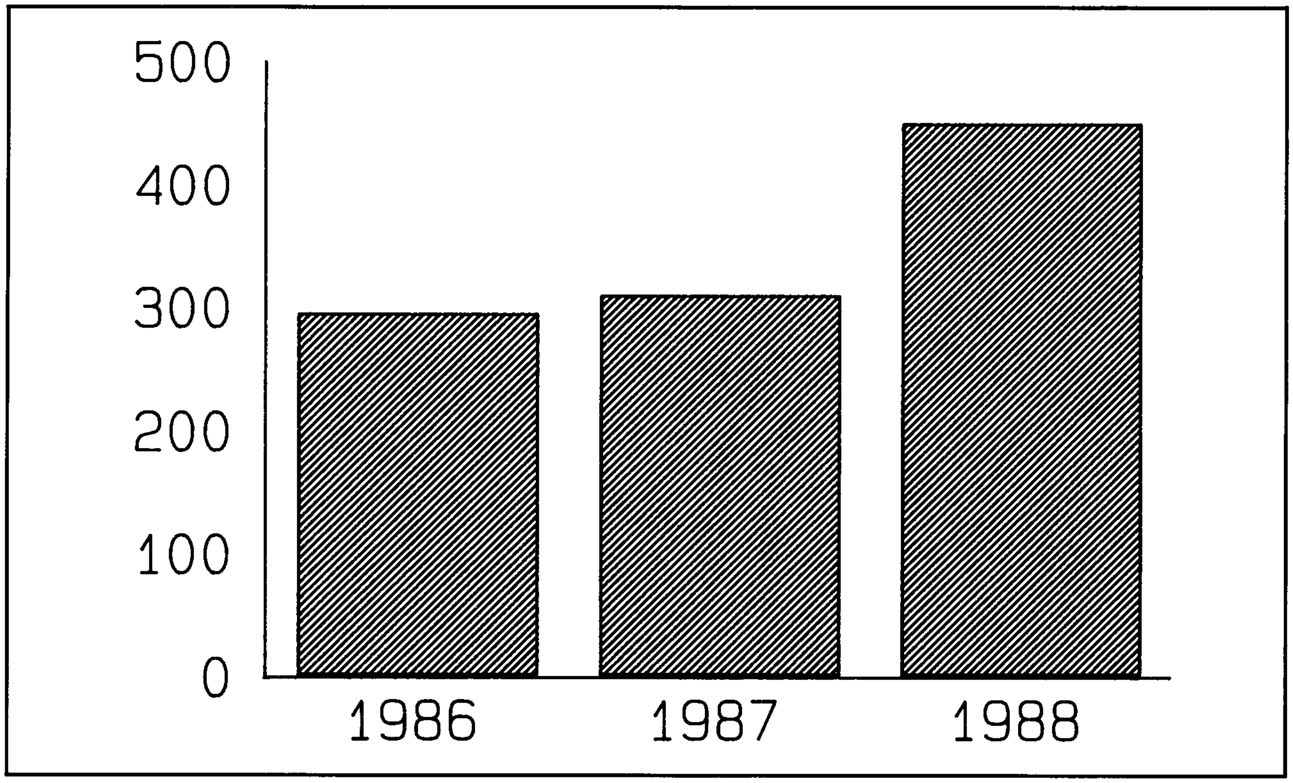


**Figure 1-6. Chart Without Titles**

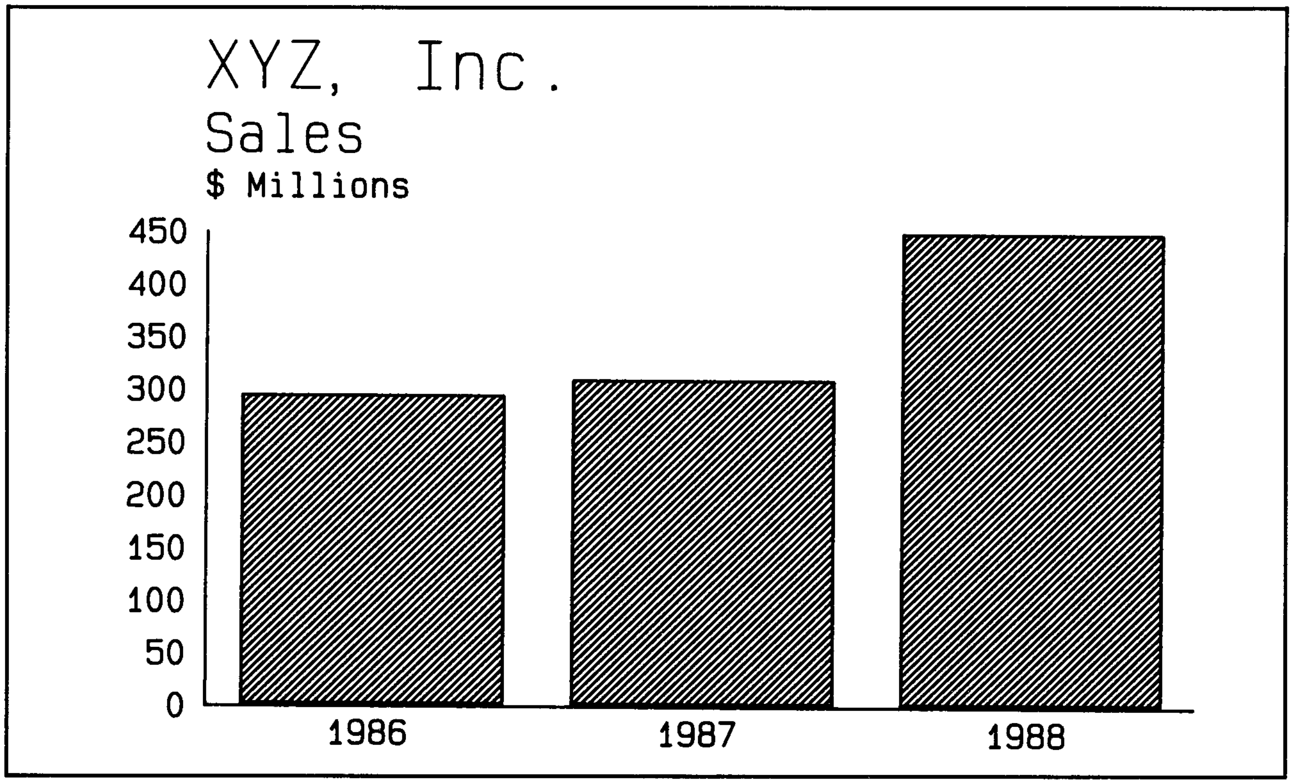


**Figure 1-7. Chart With Titles**

## Defining the Legend

Text definition routines are used to specify the legend, which you can position anywhere within the chart area. Unlike the creation of titles, the addition of a legend does not affect the size of the plotted data or the chart area. The size of the legend box is determined by the number, the length, and the height of the character strings to be placed within the legend. The character strings composing the legend are defined with the Name Attribute Set routine.

The Set Existence and Location of Legend routine allows you to enter coordinate values that define the desired base point of the legend within the view area. The Set Legend Alignment routine offers vertical and horizontal orientation options that position the legend relative to the defined base point. Figure 1-8 is an example of a chart with a legend.

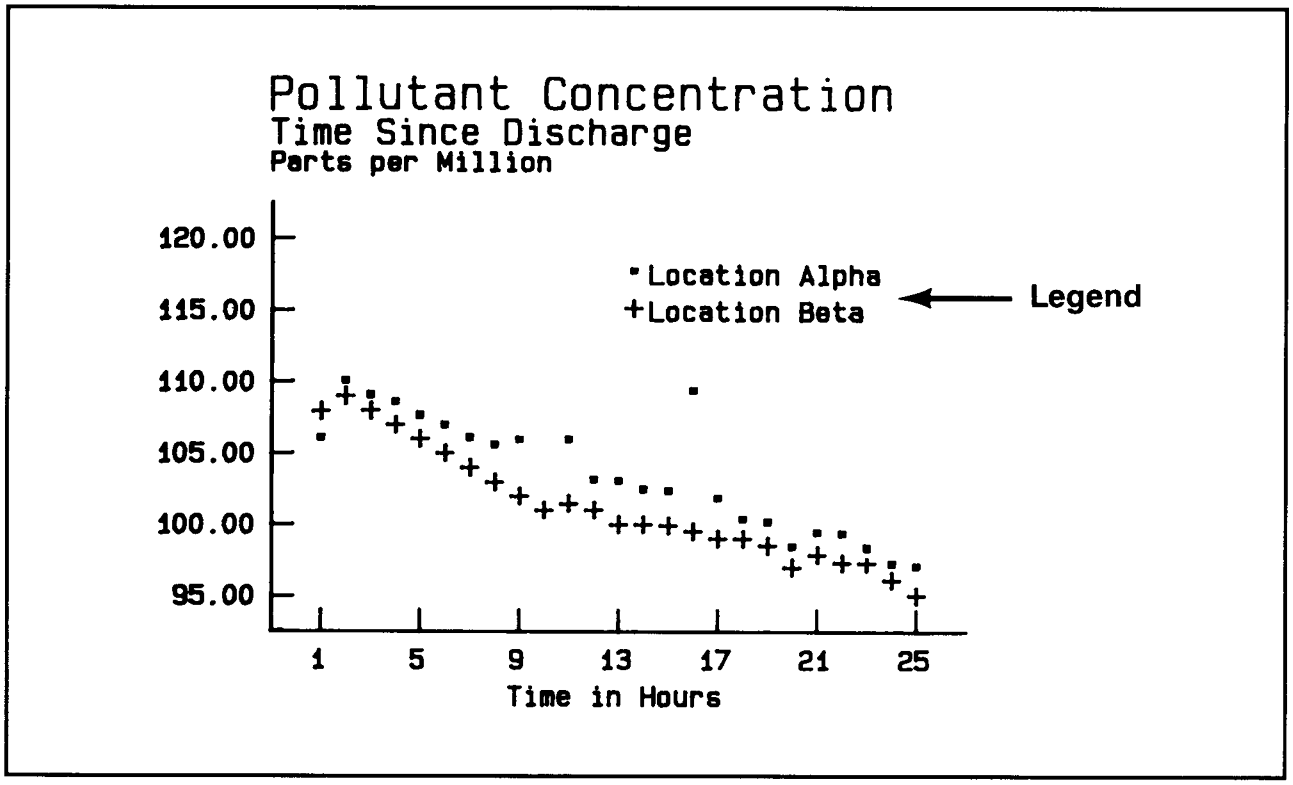


**Figure 1-8. Chart with Legend**

The Set Legend Height and Set Legend Font routines are used to specify the height and font of the legend. The default legend height is five percent of the view area. The default legend font is the hardware font. The same software fonts are available for the legend and the titles.

## Displaying the Chart

Chart output, for chart types other than text-only charts, is controlled by *control routines.*

When the Output Currently Defined Chart routine is called, the chart is output to the defined view area on the current output device. On most CRTs and plotters, the chart appears immediately. On printers, however, the chart must be stored internally in the computer and then sent to the printer.

Output to a printer does not take place until either the Clear Display Surface or Close Plotting System Output Device routine is called.

When you want to generate a two-page chart on the printer or plotter, you must call the Clear Display Surface routine after the first call to the Output Currently Defined Chart routine.

When more than one type of chart is defined at the time Output Currently Defined Chart is called, the output priority is:

- When a pie chart is defined, it is the only chart drawn. If more than one pie chart is defined, only the most recently defined pie chart is plotted.
- When a pie chart has not been defined and a schedule chart is defined, a schedule chart representing all defined schedule data sets is drawn.
- When neither a pie nor a schedule chart is defined, other chart types are plotted on a single set of axes. Because of this, any number of bar, step, line, and scatter charts can be defined for a single view area.

The elements of the chart are drawn in the following order:

- Axes
- Tick marks, grid lines, and axis tick labels (all drawn at the same time)
- Graphic representation (bars, lines, pie slices, etc.)
- Title
- Subtitle
- Axis titles
- Legend.

If a chart type has not been defined for a data set, a line chart is produced when the Output Currently Defined Chart routine is called.

If no data sets have been defined, only the default axes are drawn.

## Using Independent Chart Components

Plotting System features *independent chart component routines* that place graphic primitives and notation anywhere within the view area.

Graphic primitives are basic graphic components, such as polylines, polymarkers, arcs, arrows, circles, and rectangles. Their color and style can be specified. Notation is used to produce text-only charts and to add remarks to charts that contain graphic representation, such as bars and lines.

Graphic primitives and notation are plotted as soon as they are defined, rather than after the Output Currently Defined Chart routine is called. Often, graphic primitives and notation are added after a chart has been produced and evaluated. The specified size of graphic primitives and notation does not affect and is not affected by the size of the data representation (the bars, pies, lines, and so forth) nor by the size of the chart and axis titles. Therefore, they will overlay any other chart elements that occupy the same space they occupy.

### Generating Graphics Primitives

Separate routines are available to generate polylines, polymarkers, and filled arcs, arrows, circles, and rectangles. These routines are the Polyline, Polymarker, Arc, Arrow, Circle, and Rectangle routines. Figure 1-9 shows each of the graphics primitives.

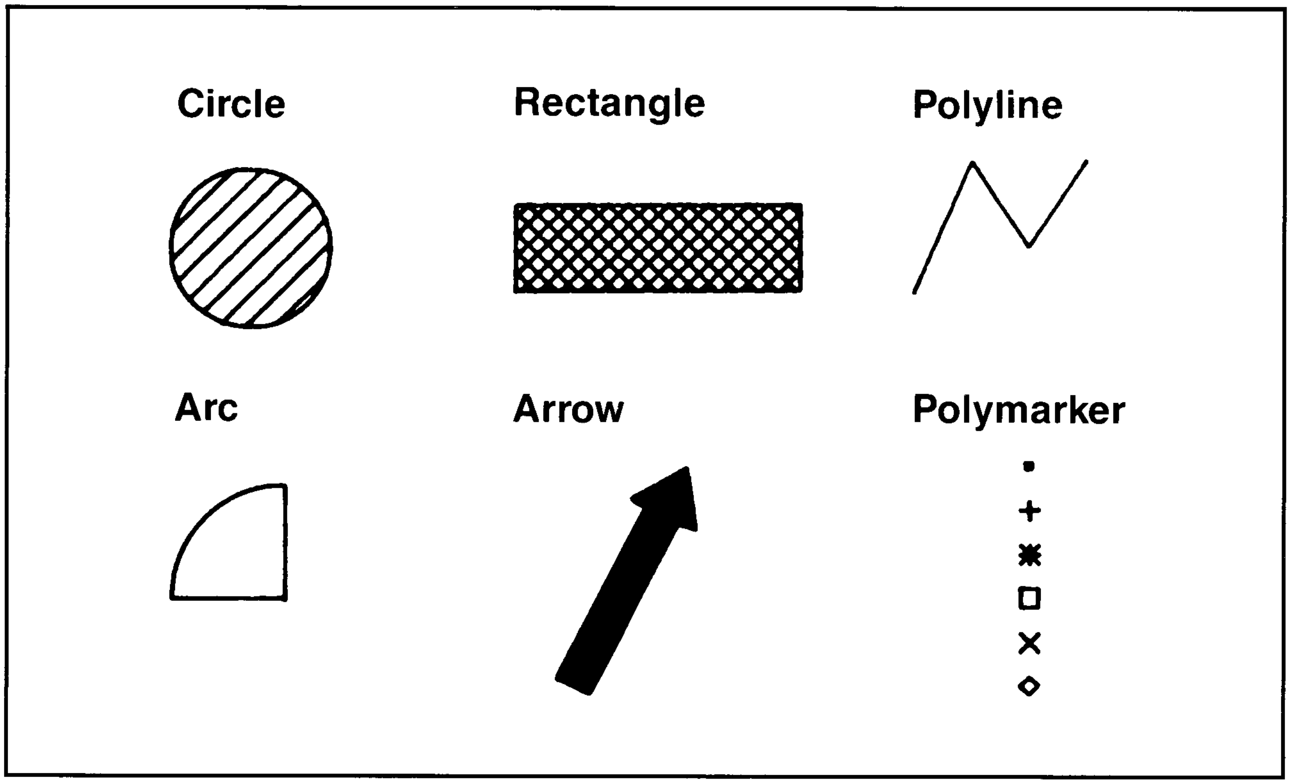


**Figure 1-9. The Graphics Primitives**

## Style Options

Several style options can be specified when you are enhancing a chart with polylines or polymarkers. The Set Polyline Type routine allows you to choose from a range of line styles including solid lines, several variations of dashed lines, dotted lines, and lines composed of a combination of dashes and dots. Options for the Set Polymarker Style routine include the asterisk, the dot, the plus, the diamond, the x, and the square bullet.

Several fill styles can be specified when filling arcs, arrows, circles, or rectangles. Fill styles for the Set Fill Style routine include solid fill, unidirectional lines, and crosshatched lines. A choice of three spacing options exists for both the unidirectional and crosshatched lines.

The Set Polyline Color routine specifies color for polylines. The Set Polymarker Color routine specifies color for markers. Set Fill Color specifies the fill color for arcs, arrows, circles, and rectangles.

When the output device is a raster display, the area defined by the graphic primitive is erased with the background color before the primitive is drawn.

### Defining Notation

Notation can be added anywhere on a chart with the Define Notation String and Location routine.

The height at which the notation appears on the chart can be specified with the Set Notation Height routine. The default value for notation height is five percent of the view area.

The font in which the notation appears on the chart can be specified with the Set Notation Font routine. The default is the hardware font. The same software fonts are available for notation, titles, and legends.

The color of the notation string can be specified with the Set Notation Color routine.

Alignment can be set according to a defined base point location of the string with the Set Notation Alignment routine.

## Resetting Plotting System

*Control routines* are used to return the plotting environment to the original default values.

The Reset Plotting System Viewing Environment routine resets the display surface and view area to the full physical device display surface.

The Reset All Data Sets routine deletes all data sets and their associated data attributes.

The Reset Plotting System to Default routine resets Plotting System to start-up conditions.

## Closing Plotting System

The last step in generating a chart is closing Plotting System. The Close Input/Output Device routine must be called to close all open devices before calling the Close Plotting System routine. Close Plotting System is the last Plotting System routine called.

If you want to do additional plotting after the Close Plotting System routine has been called, you must reinitialize Plotting System by calling the Open Plotting System routine again.

# Using Graphic Input Routines

Three *graphic input routines* are supported by Plotting System. The graphic input routines let you wait until the program is actually being run to enter such values as chart titles and colors. You can also make other choices, such as where an arrow will begin and end, and which output device to use.

Using graphic input routines, one application program can create several different charts. The program obtains input from an operator by controlling the activity of one or more logical input devices, which deliver input values to the program.

The Plotting System graphic input routines are:

- Request Choice
- Request Locator
- Request String.

The input device is activated and waits for you to terminate the input process with a device-specific action. The input value which was entered is then returned. For example, the location of the graphics cursor is returned when you type an alphanumeric key.

The input method chosen determines the type of logical input value the device delivers. The input methods and the values they provide are described in the following text.

## Request Choice

The Request Choice routine returns an integer value indicating one of a set of alternatives. Pushing a function key or responding to a menu selection using numeric keys are common forms of choice input. Choice input frequently is used for selecting the style and color of chart components.

### Request Locator

The Request Locator routine reports coordinate information from a device representing a location on the display surface. An example of a locator device is an input device such as a mouse used to position a graphics primitive or notation string.

### Request String

The Request String routine returns text from the console keyboard. String input could be used to define a chart title or request a yes/no response.

## Using Inquiry Routines

Four *inquiry routines* are supported by the Plotting System:

- The Inquire on Data Set routine allows you to obtain information about a data set.
- The Inquire on Device routine allows you to obtain information on a device's capabilities.
- The Inquire on Plotting System Error routine returns the number of the last error encountered. See the source code listings for the Line Chart program in one of the Language Reference Booklets for examples of Inquire on Plotting System Error usage.
- The Inquire on Software Version routine returns the version and level of the Plotting System software.

## Pseudocode Examples

The following text presents a series of examples using Plotting System routines. In each example, structured English pseudocode is used instead of the actual syntax used when programming in BASIC, C, FORTRAN 77 and Pascal. These generic names for the Plotting system routines are the same names used for the routines throughout the Language Reference Booklets.

The generic names are used so that the purpose of each routine and the general organization of the program are easily recognized. Throughout the text, they appear in lowercase letters with initial capitals.

**Note:** The language bindings in the Language Reference Booklets are arranged alphabetically for quick reference according to the generic names. The actual source code listings for the pseudocode examples are located at the end of each booklet. Files containing this source code are on your master diskettes.

Plotting System streamlines the source coding procedure so that a useful chart can be specified with a minimal number of routines.

## Generating a Line Chart

Suppose a department manager wants to evaluate the costs incurred by his department relative to other departments. A line chart plotting dollars spent per department would graphically portray the relationships. In the chart below, four departments are plotted along the X-axis and the dollars spent by each are plotted along the Y-axis.

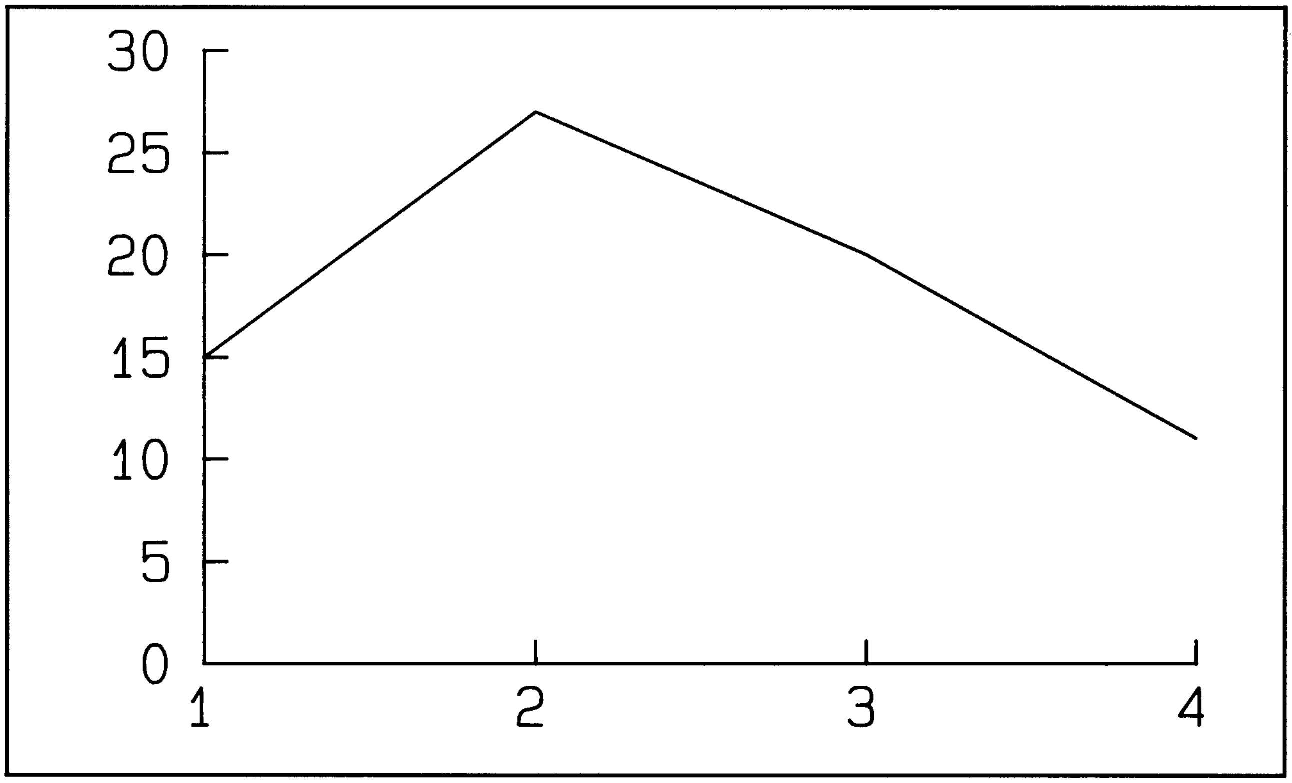


**Figure 1-10. Line Chart**

The pseudocode example in the following text shows how the data values are defined for two arrays containing coordinates for four data points. XRAY provides X-axis coordinates and YRAY provides Y-axis coordinates. By default, the axis system of this chart is just large enough to accommodate the data.

The example also shows the routines used to create the line chart. Plotting System is opened and the display is assigned as the output device. The data set is defined to be X/Y and the chart is defined to be a line chart. The chart is sent to the display, the display is closed, and Plotting System is closed.

```
DATA XRAY /1.0, 2.0, 3.0, 4.0/
DATA YRAY /15.0, 27.0, 20.0, 11.0/
Open Plotting System
Assign Plotting System Output Device (“DISPLAY”)
Define X/Y Data Set (1, 4, XRAY, YRAY)
```

In the preceding line of code, “1” refers to Data Set l, and “4” is the number of data points for the set.

```
Define Data Set Chart Type to Be Line (1)
Output Currently Defined Chart
Close Input/Output Device (“DISPLAY”)
Close Plotting System
```

Two of the lines used in the preceding example could have been replaced by one. Plotting System provides arithmetic values (1, 2, 3, . . . ,n) for the independent axis by default. Thus, an identical chart could be created by:

- Eliminating the DATA XRAY statement
- Replacing Define X/Y Data Set with Define Sequenced Data Set.

Also, calling the Define Data Set Chart Type to Be Line routine could be eliminated because a line chart is created by default. However, to improve the clarity of this example, these calls are included.

## Generating a Bar Chart

Now, suppose the department manager wants to enhance this chart so that he can present it to four departments: Engineering, Documentation, Administration, and Marketing. He decides to plot the data as a bar chart rather than a line chart, with each department represented by a different bar. He also adds titles and axis tick labels so that each staff can recognize its related bar and cost figure.

Figure 1-11 shows what the enhanced chart looks like.

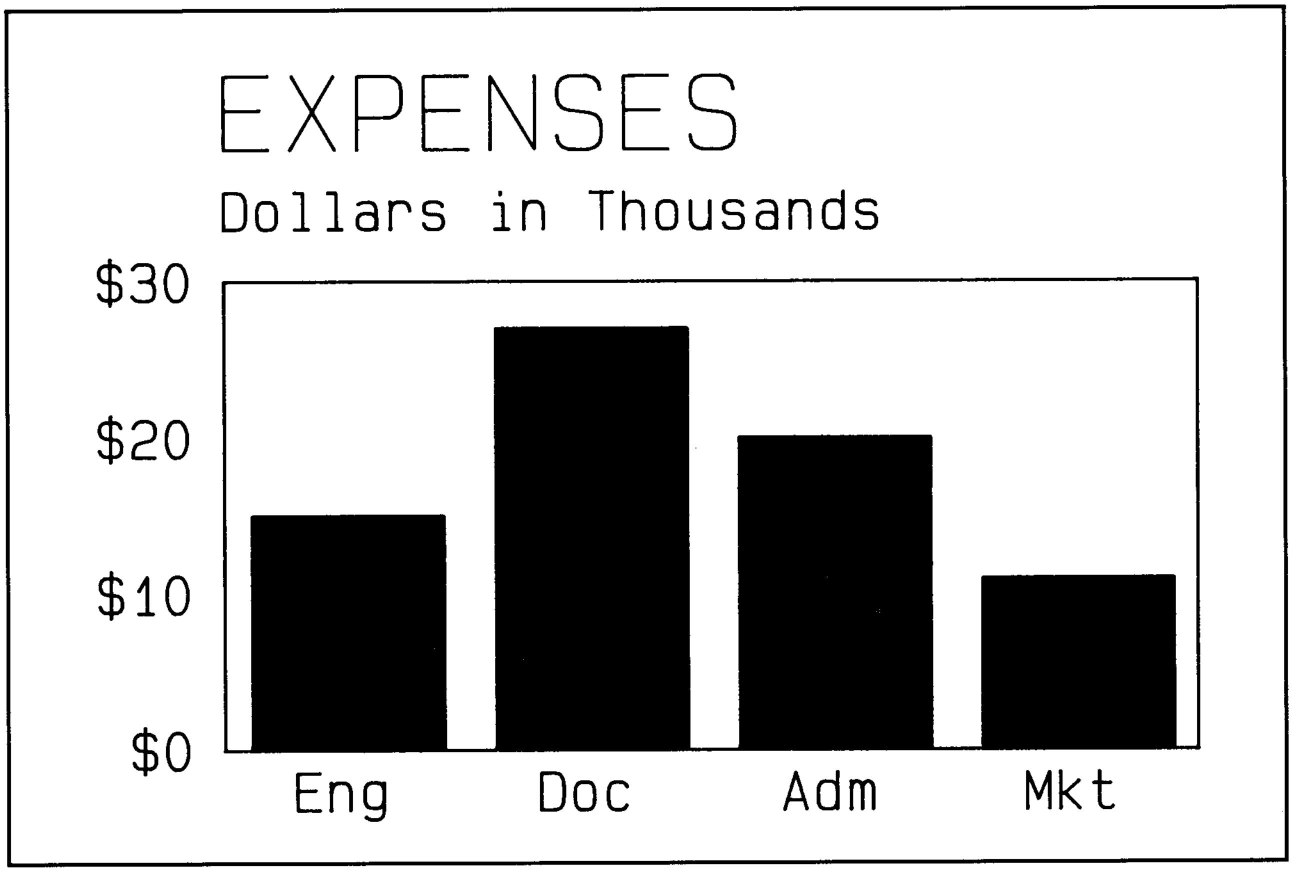


**Figure 1-11. Bar Chart**

The first four lines of the pseudocode program show that the bar chart was created using the same data values, arrays, and output device used to create the line chart in the previous example.

```
DATA XRAY /1.0, 2.0, 3.0, 4.0/
DATA YRAY /15.0, 27.0, 20.0, 11.0/
Open Plotting System Assign Plotting System Output Device ("DISPLAY")
```

However, the bar chart has a chart title and a title for the dependent axis. Also, the axis tick labels (Eng, Doc, Adm, and Mkt) are specified for the independent axis. Only the independent axis labels are being set here. Default values are used for all text height and font specifications.

**Define Title String (“EXPENSES”)**
**Define Axis Title String (2, “Dollars in Thousands”)**
**Set Axis Labels (1, “/Eng/Doc/Adm/Mkt//”)**

In the following command, the dollar sign labels for the dependent axis are specified by entering:

- 0,0 for the parameters related to the independent axis
- 3 to specify dollar signs as the dependent axis tick labels
- 1 to indicate an integer value for the numeric data.

Plotting System ignores the zeros because character labels already have been set for the independent axis.

**Set Axis Tick Label Type (0, 0, 3, 1)**

The chart area frame and the view area frame are specified as visible and as 0.1 percent of the horizontal view area width.

**Set Existence of Chart Area Frame (1, 1)**
**Set Existence of View Area Frame (1, 1)**

The data set is defined as a bar chart, the solid fill style is chosen, and the chart is sent to the display. Then, the display and Plotting System are closed.

**Define X/Y Data Set (1, 4, XRAY, YRAY)**
**Define Data Set Chart Type to Be Bar (1)**
**Set Output Primitive Style for Data Set (1, 2)**
**Output Currently Defined Chart**
**Close Input/Output Device (“DISPLAY”)**
**Close Plotting System**

## Generating a Multiple Bar Chart

Now, suppose the department manager showed the labeled chart to his director. Appreciating the graphic insight it provided, the director wants to use the chart when asking for a budget increase at the next meeting of the Board of Directors. However, she wants the chart to reflect cost data from the previous year as well as the current year.

Figure 1-12 shows what the director has in mind.

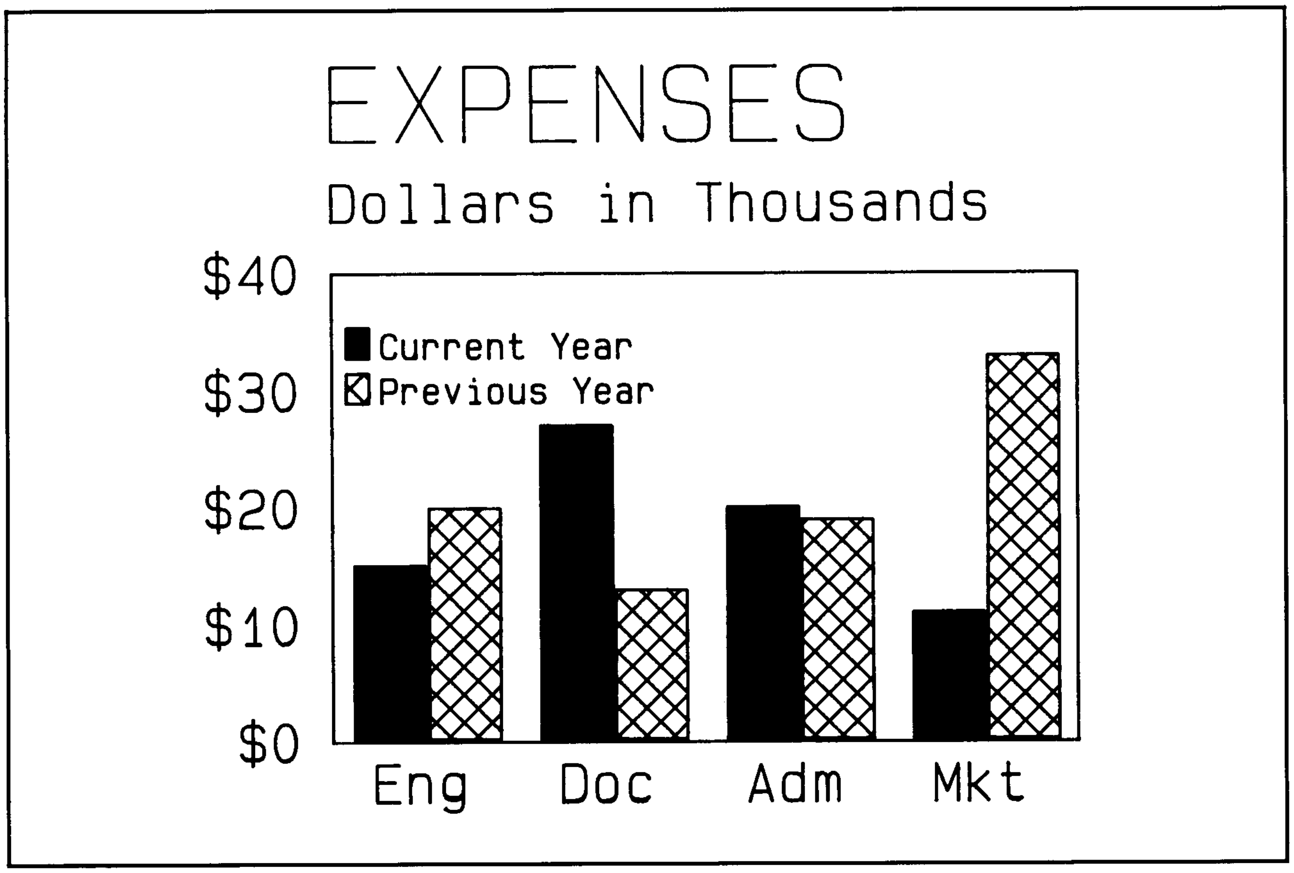


**Figure 1-12. Multiple Bar Chart**

To reflect the previous year cost data, another array with its corresponding data values is added before opening the Plotting System and assigning the output device.

**DATA XRAY /1.0, 2.0, 3.0, 4.0/**
**DATA YRAY /15.0, 27.0, 20.0, 11.0/**
**DATA YRAY1 /20.0, 13.0, 19.0, 33.0/**
**Open Plotting System**
**Assign Plotting System Output Device ("DISPLAY")**

The display surface defined fills a letter-size piece of paper (11 inches × 8½ inches).

**Set Display Surface Size (11.O, 8.5)**

Titles, axis labels, axis tick labels, chart area frame, and view area frame are set as before.

**Define Title String ("EXPENSES")**
**Define Axis Title String (2, "Dollars in Thousands")**
**Set Axis Labels (1, "/Eng/Doc/Adm/Mkt//")**
**Set Axis Tick Label Type (0, 0, 3, 1)**
**Set Existence of Chart Area Frame (1, 1)**
**Set Existence of View Area Frame (1, 1)**

Both data sets are defined to be bar charts and related attributes are specified so they are distinguishable when plotted. As before, bars with solid fill represent this year's data. But now, bars with crosshatched fill style distinguish last year's data. A different color is used for each data set.

**Define X/Y Data Set (1, 4, XRAY, YRAY)**
**Define X/Y Data Set (2, 4, XRAY, YRAY1)**
**Define Data Set Chart Type to Be Bar (1)**
**Define Data Set Chart Type to Be Bar (2)**
**Set Output Primitive Style for Data Set (1, 2)**
**Set Output Primitive Color for Data Set (1, 2)**
**Set Output Primitive Style for Data Set (2, 8)**
**Set Output Primitive Color for Data Set (2, 3)**

To show which set of bars represents this year's data and which represents last year's data, names are assigned that will appear in a legend area in the upper-left corner of the chart area. The height of the legend is set, also. The chart is sent to the display. The display and Plotting System are closed.

**Name Attribute Set (1, "Current Year")**
**Name Attribute Set (2, "Previous Year")**
**Set Legend Height (3)**
**Set Existence and Location of Legend (1, 0, 0.0, 70.0)**
**Output Currently Defined Chart**
**Close Input/Output Device ("DISPLAY")**
**Close Plotting System**

## Generating a Bar and a Pie Chart

The Board of Directors approves the budget increase. They want to send a modified version of the chart to the stockholders, showing the total cost of the increase, as well as the department breakdown.

Therefore, the department manager is asked to provide a pie chart that shows the percentage of the total cost spent by each department. The two charts are to be placed side-by-side on the same sheet of paper. Figure 1-13 is the department manager's chart.

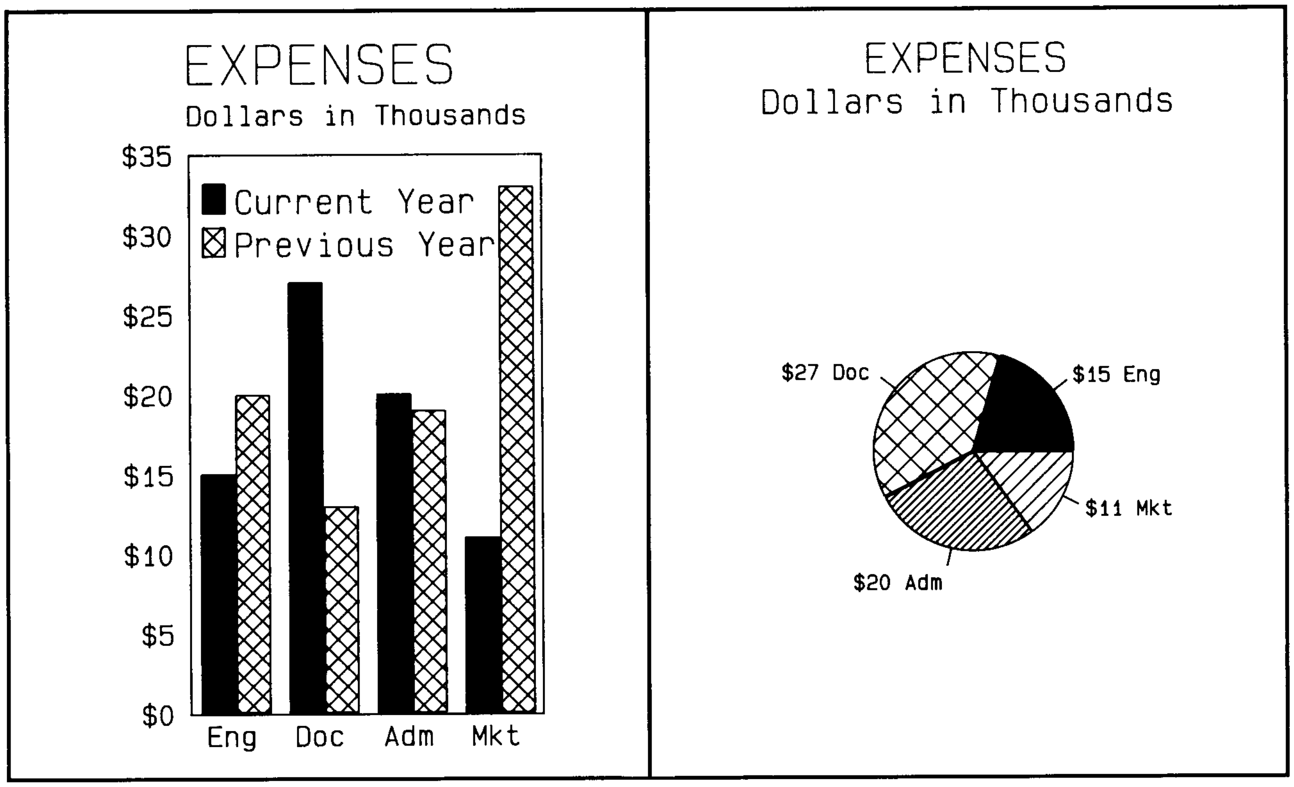


**Figure 1-13. Combined Bar and Pie Chart**

The same initial code that was used for the other multiple bar chart is used again here, with one notable exception. To fit both charts on the same page, the Set View Area Extents routine is used. With this routine, a view area is specified for the bar chart. This view area is 50 percent the size of the display surface used to create the bar chart in the previous example. It appears on the left side of the display surface.

```
DATA XRAY /1.0, 2.0, 3.0, 4.0/
DATA YRAY /15.0, 27.0, 20.0, 11.0/
DATA YRAY1 /20.0, 13.0, 19.0, 33.0/
Open Plotting System Assign Plotting System Output Device (“DISPLAY”)
Set Display Surface Size (11.0, 8.5)
Set View Area Extents (0.0, 0.0, 50.0, 100.0)
```

Again, the same code is used to define the titles, axis labels, and so forth, that was used for the other multiple bar chart. This time, however, a pie chart must be created before the display and Plotting System are closed.

```
Define Title String (“EXPENSES”)
Define Axis Title String (2, “Dollars in Thousands”)
Set Axis Labels (1, “/Eng/Doc/Adm/Mkt//”)
Set Axis Tick Label Type (0, 0, 3, 1)
Set Existence of Chart Area Frame (1, 1)
Set Existence of View Area Frame (1, 1)
Define X/Y Data Set (1, 4, XRAY, YRAY)
Define X/Y Data Set (2, 4, XRAY, YRAY1)
Define Data Set Chart Type to Be Bar (1)
Define Data Set Chart Type to Be Bar (2)
Set Output Primitive Style for Data Set (1, 2)
Set Output Primitive Color for Data Set (1, 2)
Set Output Primitive Style for Data Set (2, 8)
Set Output Primitive Color for Data Set (2, 3)
Name Attribute Set (1, “Current Year”)
Name Attribute Set (2, “Previous Year”)
Set Legend Height (3)
Set Existence and Location of Legend (1, 0, 0.0, 80.0)
Output Currently Defined Chart
```

A view area is specified for the pie chart. This view area consists of the remaining 50 percent of the display surface not used by the bar chart. It is positioned on the right side of the display surface.

```
Set View Area Extents (50.0, 0.0, 100.0, 100.0)
```

A subtitle string is defined and the chart type is defined as a pie chart. Also, colors are set for pie slices 3 and 4.

**Define Subtitle String (“Dollars in Thousands”)**
**Define Data Set Chart Type to Be Pie (1)**
**Set Output Primitive Color for Data Set (3, 2)**
**Set Output Primitive Color for Data Set (4, 3)**

Each pie slice is labeled.

**Name Attribute Set (1, “Eng”)**
**Name Attribute Set (2, “Doc”)**
**Name Attribute Set (3, “Adm”)**
**Name Attribute Set (4, “Mkt”)**

The pie chart is sent to the display. The display and Plotting System are closed.

**Output Currently Defined Chart**
**Close Input/Output Device (“DISPLAY”)**
**Close Plotting System**

## Adding Graphic Input Routines

Suppose the department manager wants to enhance the pie chart by using an arrow to highlight the pie slice that represents engineering expenses. To illustrate the graphic input routines, assume he wants to input a chart title, position the arrow, choose a color for the arrow, and then output the chart to the plotter to create the chart shown in Figure 1-14.

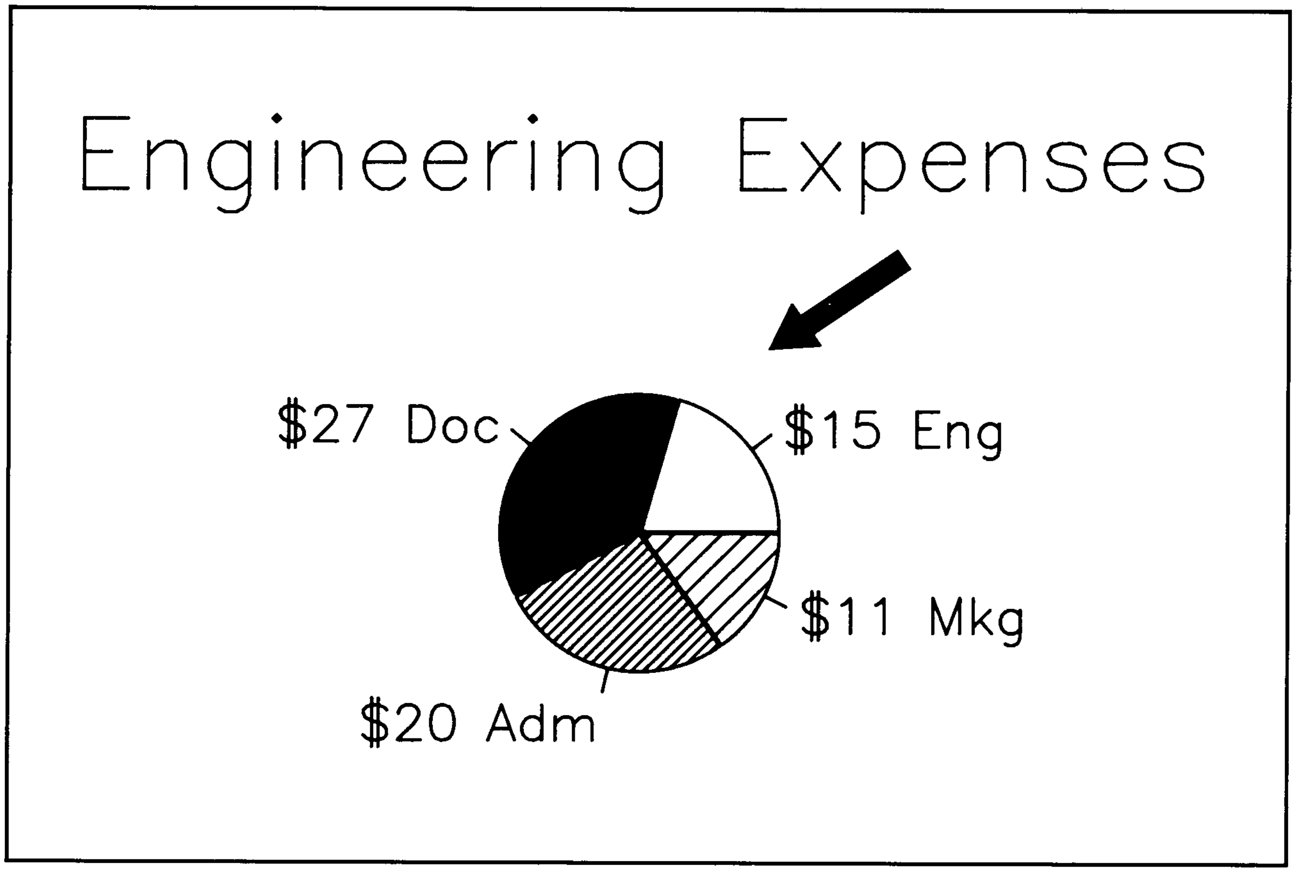


**Figure 1-14. Pie Chart with Arrow for Emphasis**

Before going into the steps for producing the pie chart with the arrow, an overview of the entire process is necessary because two new concepts are introduced here.

The first new concept deals with switching back and forth between two view areas during program execution. In this example, prompts and questions are included in the program. These prompts and questions appear at the top of the screen as the program is run, telling what actions to take and giving an opportunity to make choices that will affect the chart's appearance.

For this to happen, the display surface must be divided into two sections. The top of the display, view area (0.0, 90.0) to (100.0, 100.0), is used to show the prompts and questions, and echo responses. The bottom portion, (0.0, 0.0) to (100.0, 89.0), is used to display the chart and echo the graphics cursor in the Request Locator routine. Switching back and forth between the two sections is accomplished by calling the Set View Area Extents routine.

The second new concept is that of opening the display as both an input and output device simultaneously. This is necessary in order to echo the responses to the Request Locator and Request String routines.

**Note:** Opening an input device is necessary only when using one or more of the Request routines.

The Request Locator routine allows the graphics cursor to be used to position the arrow tip and end points. The movement of the graphics cursor is echoed on the display by Plotting System. The final position is returned to the tip and end parameters, respectively, as output by the Request Locator routine.

The Request String routine allows the chart title to be entered at run time.

Before the chart is sent to the plotter, the display is closed as an output device and the plotter is opened as an output device.

Now look at the steps used to create the pie chart with the arrow. The same array and data values that were used to create the pie chart in the previous example are used again. However, for this chart a statement (DATA ECHOPT) is added to define the location on the display surface where the title string will be echoed in response to the Request String routine.

A data initiation statement (DATA INIT) is added to mark the spot where the graphics cursor will initially appear. (The graphics cursor is moved with an input locator, such as the cursor control keys on the keyboard.) Then, Plotting System is opened.

**DATA YRAY /15.0, 27.0, 20.0, 11.0/**
**DATA ECHOPT /1.0, 0.0/**
**DATA INIT /50.0, 80.0/**
**Open Plotting System**

The display is opened for output.

**Assign Plotting System Output Device ("DISPLAY")**

The view area extents are set for the area that will contain the pie chart, leaving space at the top of the screen for prompts and questions.

The view area is framed and a sequenced data set is defined, rather than an X/Y data set as in the previous example. This can be done because only one array must be interpreted.

**Set View Area Extents (0.0, 0.0, 100.0, 89.0)**
**Set Existence of View Area Frame (1, 1)**
**Define Sequenced Data Set (1, 4, YRAY)**

The next step is to define YRAY as a pie chart and to assign labels and label types to the pie slices.

**Define Data Set Chart Type to Be Pie (1)**
**Name Attribute Set (1, "Eng")**
**Name Attribute Set (2, "Doc")**
**Name Attribute Set (3, "Adm")**
**Name Attribute Set (4, "Mkt")**
**Set Axis Tick Label Type (0, 0, 3, 1)**

Since the pie chart must be seen so the prompt/question area can be defined and the arrow positioned correctly, the chart is sent to the display.

**Output Currently Defined Chart**

With the chart on the display, the next step is to use the graphic input routines to annotate the chart. The display is opened for input. The view area is set to the portion of the display surface that will contain the prompts and questions.

**Assign Plotting System Input Device ("DISPLAY")**
**Set View Area Extents (0.0, 90.0, 100.0, 100.0)**

The color of the chart title is set, Plotting System is told to prompt for the title, and the title string is defined. This title appears in the prompt area when it is typed, but does not appear on the chart until it is printed or plotted.

**Set Notation Color (3)**
**Define Notation String and Location (1, 1.0, 50.0,**
**"Enter chart title:                         ")**
**Request String (1, 1, ECHOPT, 24, STRING)**
**Define Title String (STRING)**

Next, Plotting System is told to prompt for the arrow color. The Request Choice routine is used to select a color when one of the function keys **(F1-F10)** is pressed. These function keys correspond to the color index of the output device being used, as explained in Set Color Representation in Chapter 3, "Plotting System Routines."

When a function key is pressed, the value returned is used by the Set Fill Color routine to set the arrow fill color. However, this color does not appear until the arrow's location is defined and the chart is output.

```
Define Notation String and Location (1, 1.0, 50.0,
   "Use F1-F10 keys to select arrow color")
Request Choice (1, COLOR)
Set Fill Color (COLOR)
```

Plotting System is told to prompt for the arrow tip position. The view area is set to the portion of the display surface that contains the pie chart. The Request Locator routine is used to cause the graphics cursor to appear. This routine allows the use of an input locator device, such as the cursor control keys, to position the graphics cursor.

```
Define Notation String and Location (1, 1.0, 50.0,
   "Position cursor for arrow tip                ")
Set View Area Extents (0.0, 0.0, 100.0, 89.0)
Request Locator (1, 1, INIT, TIP, KEY)
```

Next, the view area is set to the portion of the display surface that contains the prompts. Plotting System is told to prompt for the arrow end position.

```
Set View Area Extents (0.0, 90.0, 100.0, 100.0)
Define Notation String and Location (1, 1.0, 50.0,
   "Position cursor for arrow end                ")
```

Then, the view area extents are set to the display area that contains the pie chart. The Request Locator routine is used again to cause the graphics cursor to appear and to select the location of the arrow end.

```
Set View Area Extents (0.0, 0.0, 100.0, 89.0)
Request Locator (2, 1, TIP, END, KEY)
```

The pie chart and the arrow, with a solid fill style, are sent to an output device.

```
Output Currently Defined Chart
Set Fill Style (2)
Arrow (1, END, TIP)
```

To provide a choice of output devices, the view area extents are set to the question area and Plotting System is told to ask if the chart will be sent to the plotter.

```
Set View Area Extents (0.0, 90.0, 100.0, 100.0)
Define Notation String and Location (1, 1.0, 50.0,
   "Output chart to plotter? y or n                    ")
Request String (1, 1, ECHOPT, 2, STRING)
```

Regardless of the response, the display will no longer be used. Therefore, the display is closed as both an input and output device. It must be closed twice.

```
Close Input/Output Device ("DISPLAY")
Close Input/Output Device ("DISPLAY")
```

Plotting System is told that y means to set the view area to the entire display surface size, open the plotter as an output device, send the pie chart and the arrow to the plotter, and then close the plotter as an output device.

```
IF (STRING="y") THEN
   Set View Area Extents (0.0, 0.0, 100.0, 100.0)
   Assign Plotting System Output Device ("PLOTTER")
   Output Currently Defined Chart
   Arrow (1, END, TIP)
   Close Input/Output Device ("PLOTTER")
ENDIF
```

The last step, whether the answer is y or n, is to close the Plotting System.

```
Close Plotting System
```

## Generating a Text-only Chart

Now, suppose the department manager wants to produce the text-only chart identifying his four departments as illustrated in Figure 1-15.

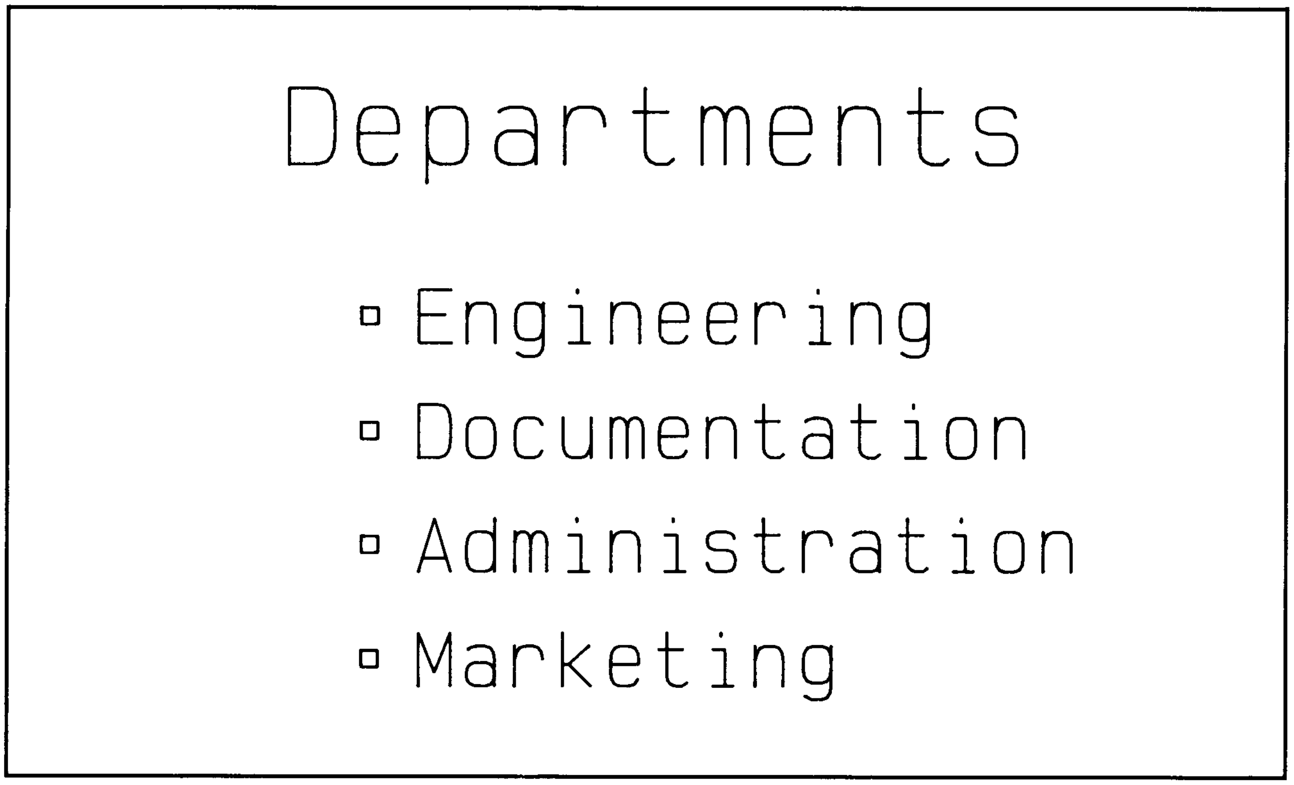


**Figure 1-15. Text-Only Chart**

The first step is to position four bullets ($), one for each department.

**DATA MARKER1 /20.0, 60.0/**
**DATA MARKER2 /20.0, 45.0/**
**DATA MARKER3 /20.0, 30.0/**
**DATA MARKER4 /20.0, 15.0/**

Next, Plotting System is opened and the display is chosen as the output device.

**Open Plotting System**
**Assign Plotting System Output Device ("DISPLAY")**

Then, a display surface that fills a letter-size (11 inches × 8½ inches) piece of paper is defined.

**Set Display Surface Size (11.0, 8.5)**

Now, the alignment, height, and color of the first string, which will serve as the chart title, are set.

**Set Notation Alignment (2, 1)**
**Set Notation Height (10)**
**Set Notation Color (2)**

The notation string and its location are defined.

**Define Notation String and Location (1, 50.0, 80.0, "Departments")**

The next task is to set the alignment, height, and color of the other strings.

**Set Notation Alignment (1, 2)**
**Set Notation Height (7)**
**Set Notation Color (3)**

Next, the other strings and their locations are defined.

**Define Notation String and Location (1, 25.0, 60.0, "Engineering")**
**Define Notation String and Location (1, 25.0, 45.0, "Documentation")**
**Define Notation String and Location (1, 25.0, 30.0, "Administration")**
**Define Notation String and Location (1, 25.0, 15.0, "Marketing")**

A polymarker style and color are set to emphasize each of the departments. The style chosen is the bullet that precedes each department name. Then, the locations of the bullets are set, corresponding to the four marks set at the beginning of the program.

**Set Polymarker Style (4)**
**Set Polymarker Color (2)**
**Polymarker (1, 1, MARKER1)**
**Polymarker (1, 1, MARKER2)**
**Polymarker (1, 1, MARKER3)**
**Polymarker (1, 1, MARKER4)**

**Note:** Calling the Output Currently Defined Chart routine is not necessary for this chart. If this routine is called, only the X/Y axes appear.

Finally, the display and Plotting System are closed.

```
Close Input/Output Device ("DISPLAY")
Close Plotting System
```

The actual calls and parameters needed for each of the routines referred to in the preceding pseudocode examples are included in the Language Reference Booklets for BASIC, C, FORTRAN 77, and Pascal.

Source program listings of these examples are included at the end of each binding. Source code for these programs is on the master diskettes.

# Chapter 2. Selecting A Chart

## CONTENTS

# About This Chapter

This chapter is an introduction to using charts and graphs for data representation and analysis. It explores the relationship between Plotting System's options and some general principles surrounding the representation of data by charts. This is done in order to help you to choose the right type of chart for your application and design it to suit your particular audience. All the graphs shown in this chapter were generated by Plotting System.

# About Charts

Charts are used extensively to represent data in scientific, business, engineering, and many other fields, since charts are a powerful and time-saving way of organizing complex data into easily understandable form. An effective chart makes the characteristic features of a data set, such as its volume, organization, range, and precision, obvious at a glance. The first step in effective chart design is to choose the best among many possibilities to display your particular set of data. To narrow down your choices, consider the main purpose of your chart. Some typical ones are:

- To make a comparison
- To show the relationship of parts to a whole
- To show change in relation to time.

Evaluate the nature of the data, the medium of presentation, the purpose of the chart, and the audience when selecting the chart type.

The following sections contain a few basic rules for matching a type of chart to the category of information you are presenting. Many other possibilities exist, limited only by your creativity. Plotting System enables you to draw the four types of charts listed below:

- X/Y-axis charts
- Pie charts
- Schedule charts
- Text-only charts.

These charts are illustrated in Figure 2-1.

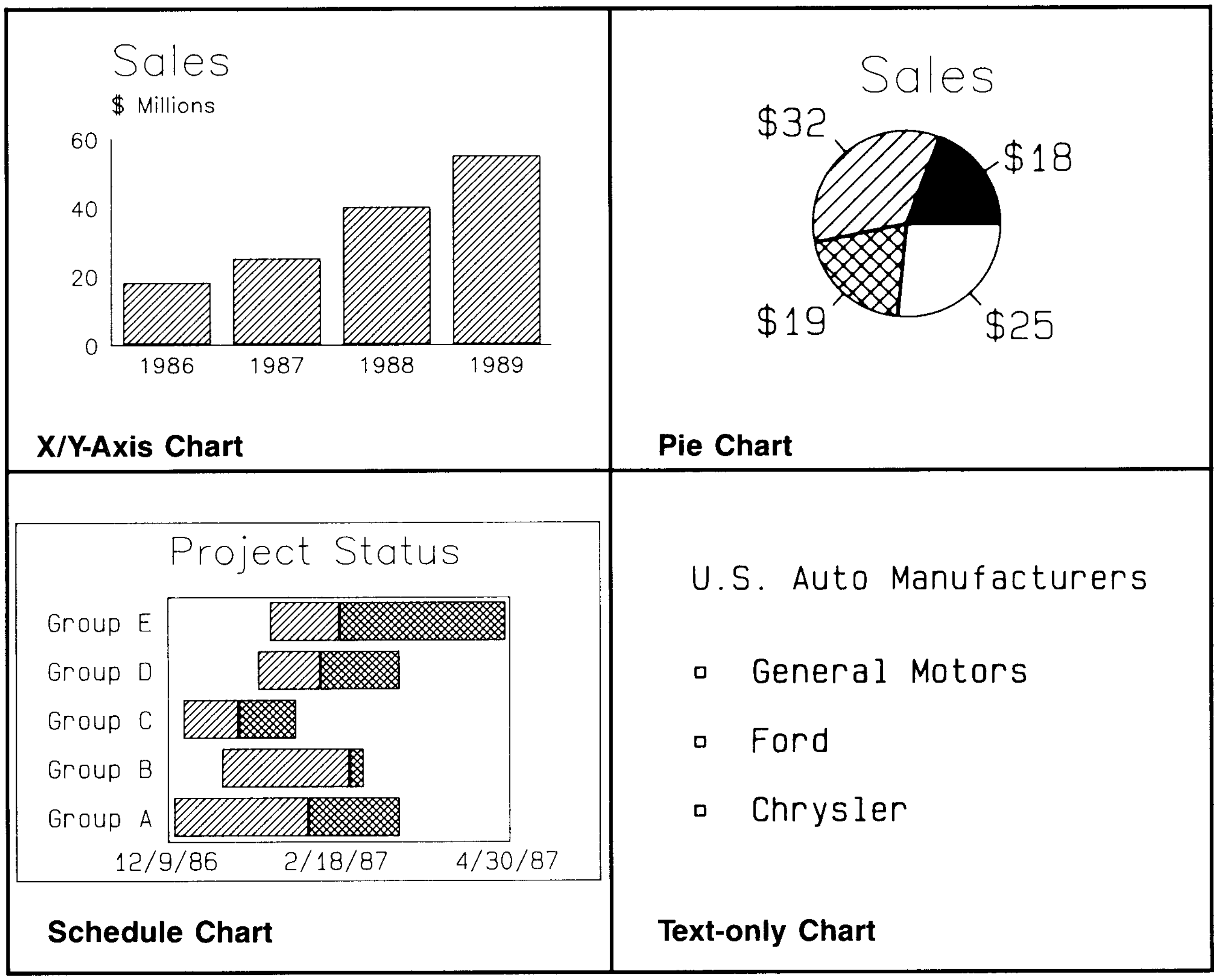


**Figure 2-1. Types of Charts**

# X/Y-Axis Charts

An X/Y-axis chart is a good choice when your data compares two variables, such as dollar growth against time, mileage against distance, or percentage change against time. One variable is plotted along the independent X-axis and the other is plotted relative to the dependent Y-axis. Conventionally, the horizontal axis is the independent X-axis and the vertical axis is the dependent Y-axis.

Five types of X/Y-axis charts are implemented by Plotting System:

- Bar charts
- Line charts
- Step charts
- Area charts
- Scatter charts.

Plotting System also allows you to create several variations of X/Y-axis charts, which are discussed at the end of this section.

## Bar Charts

A bar chart effectively represents small amounts of data, usually eight or fewer data values, each of which is represented by a single bar. The length of a bar is determined by the dependent variable.

### Vertical Bar Chart

Figure 2-2 illustrates a *vertical bar chart,* or simple column chart. It represents only one set of data.

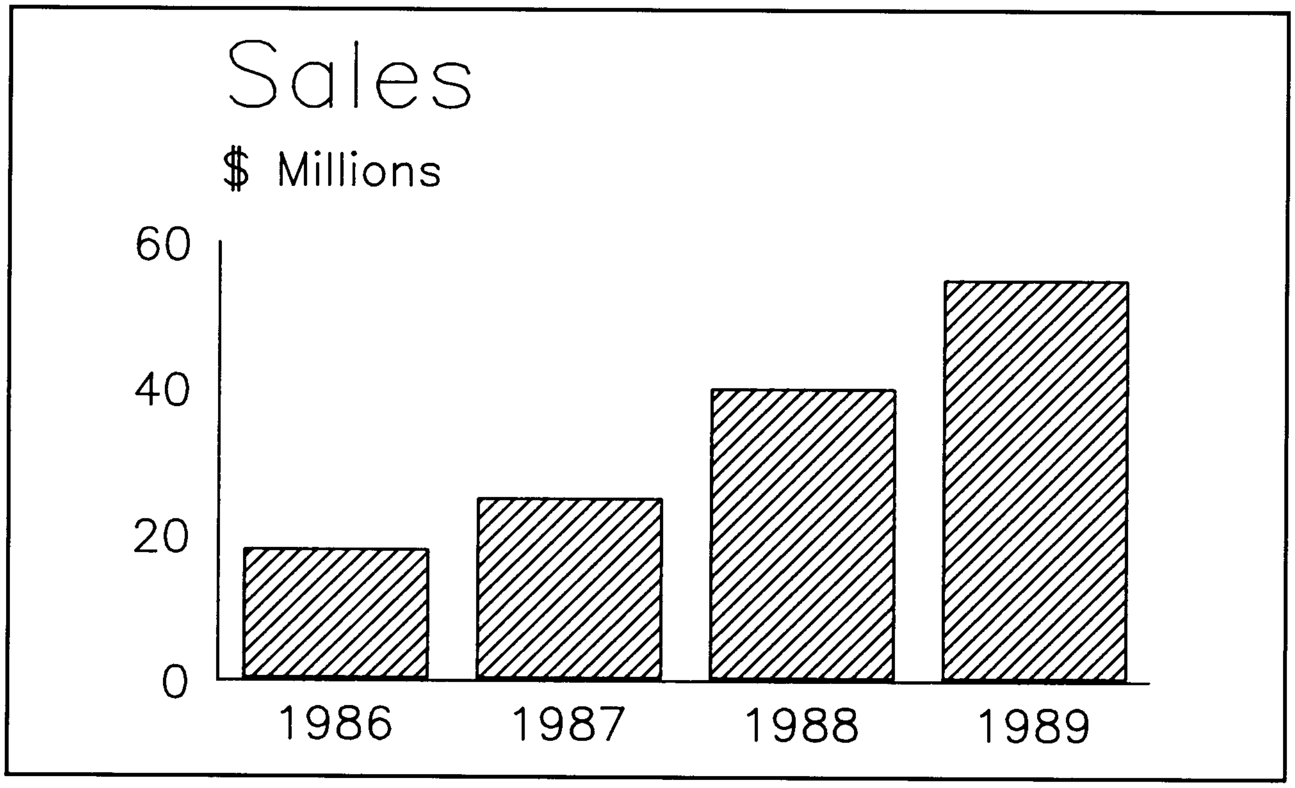


**Figure 2-2. Vertical Bar Chart**

## Multiple Bar Chart

Two or more data sets with the same independent values may be placed in the same bar chart for easy comparison. Figure 2-3 represents two sets of data and is called a *multiple bar chart* or *clustered bar chart.* Bars representing each set of data are clustered at the same tick mark.

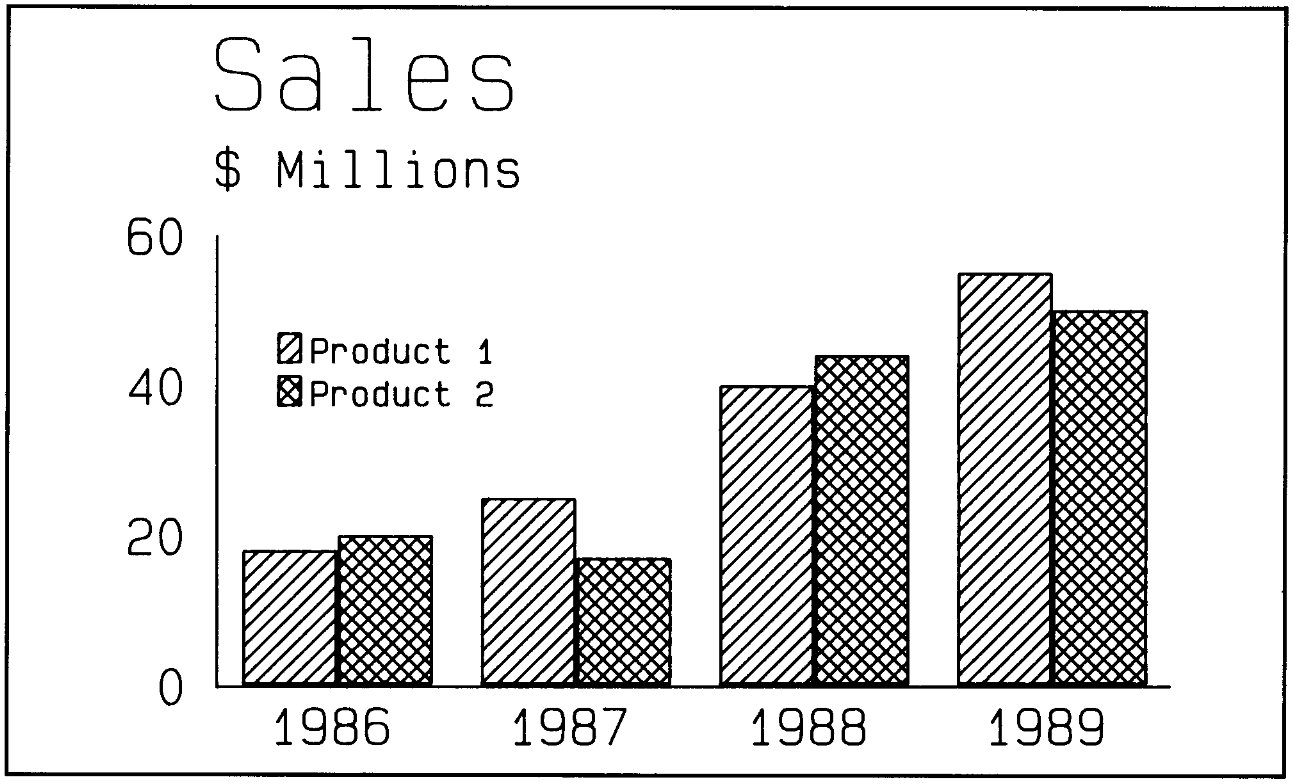


**Figure 2-3. Multiple Bar Chart**

## Stacked Bar Chart

If you need to portray the sum of several different data sets in one bar, the bars can be stacked, as shown in Figure 2-4. This bar chart represents four data sets and is called a *stacked bar chart.* Either the sum of the data sets or the total of each set can be read from the numbered dependent axis.

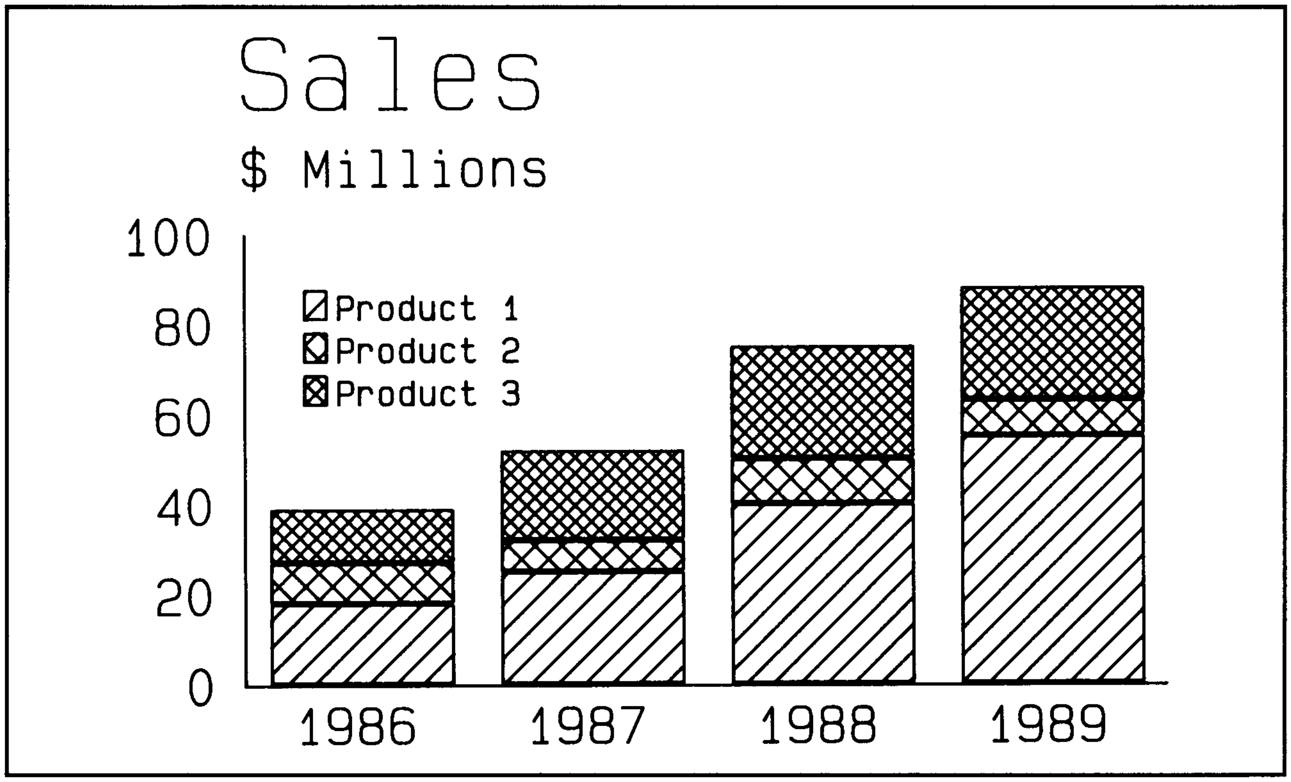


**Figure 2-4. Stacked Bar Chart**

For more information, see "Stack a Set of Data Sets" in Chapter 3.

## Floating Bar Chart

You can direct the reader's eyes to a particular range of information with a *floating bar chart.* By specifying that one or more of the data sets in the stacked bar chart are not visible, you can create the effect of floating bars, shown in Figure 2-5.

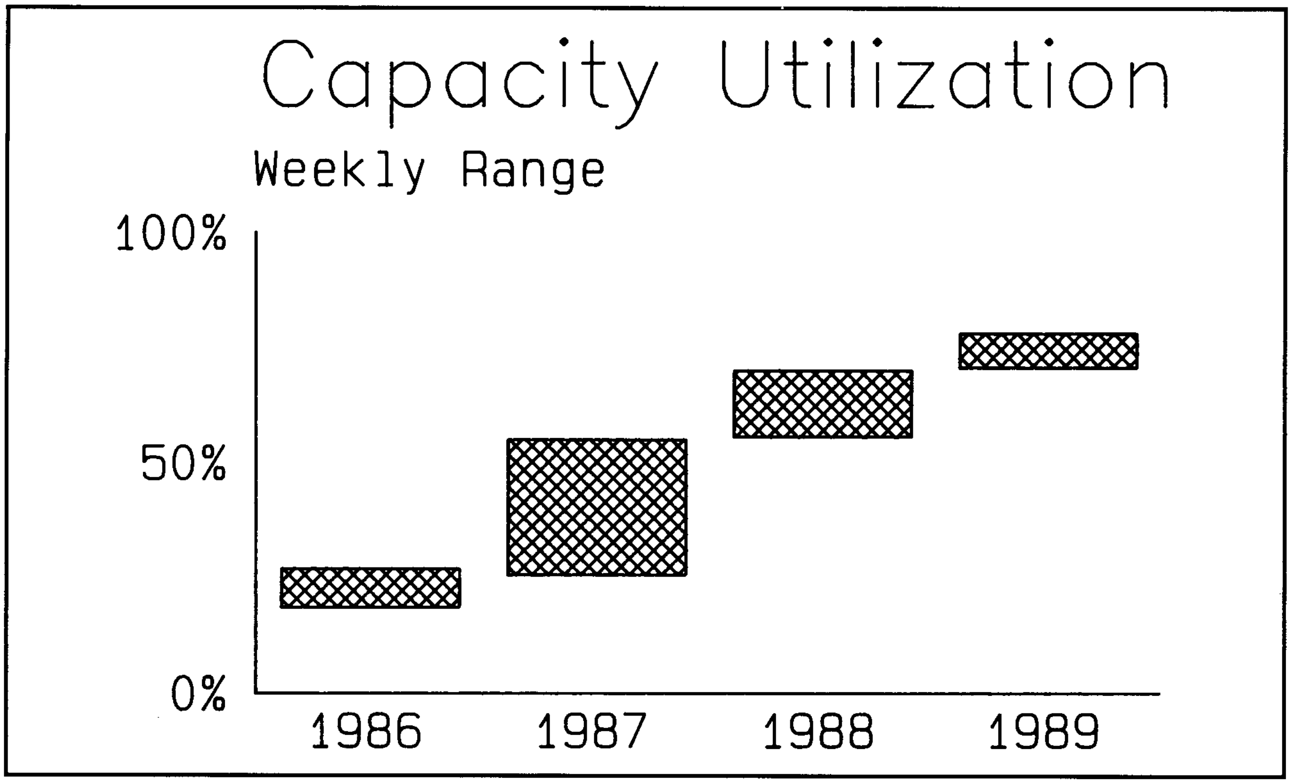


**Figure 2-5. Floating Bar Chart**

For more information, see "Set Visibility of Data Set" and "Stack a Set of Data Sets" in Chapter 3.

## Deviation Bar Chart

The *deviation bar chart* is still another variation of the bar chart. Normally data is plotted relative to the independent axis. In a deviation bar chart, the bars extend to either side of a baseline which is drawn parallel to the independent axis. This type of chart is especially useful in representing positive-negative or profit-loss data.

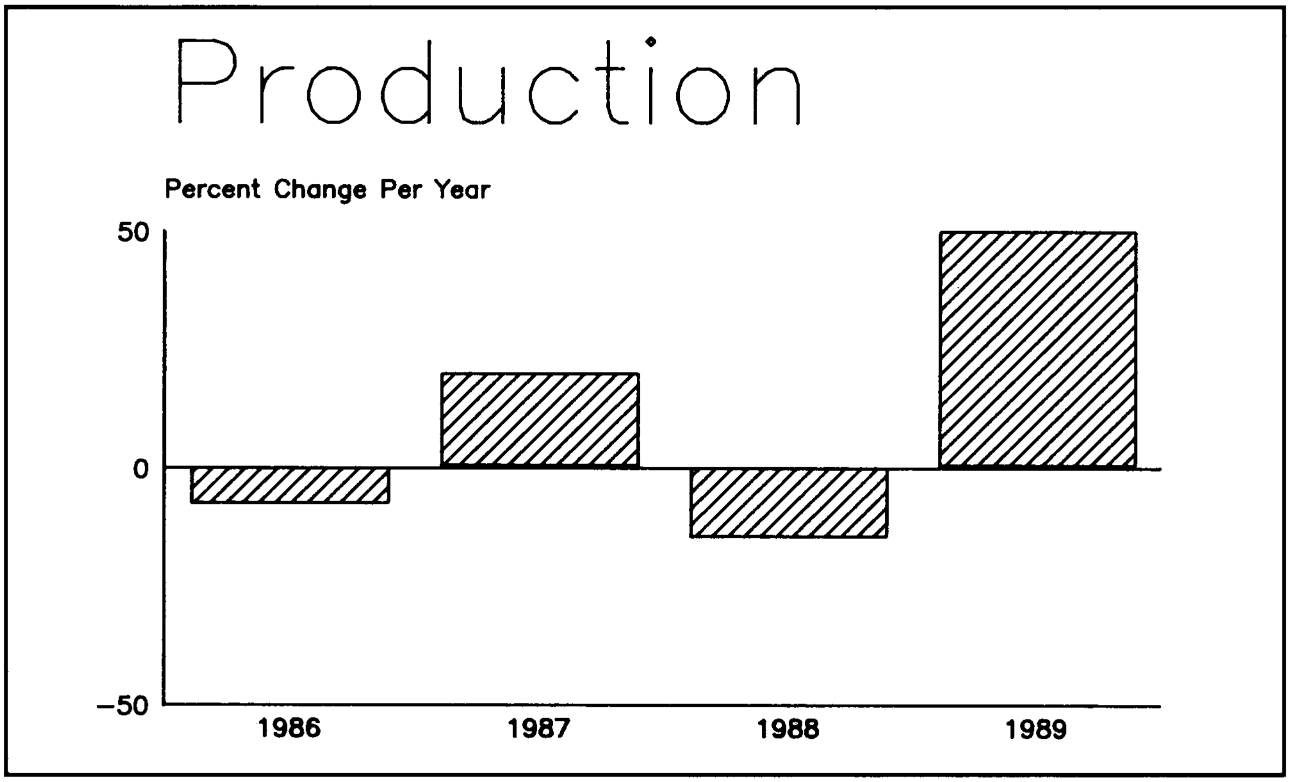


**Figure 2-6. Deviation Bar Chart**

For more information, see “Set Baseline” in Chapter 3.

## 100-Percent Bar Chart

This type of bar chart, shown in Figure 2-7, is called a *100-percent bar chart* because each element stacked in the bar represents a percentage of the complete set. By grouping several 100-percent bars on a single chart, you can compare a number of complete sets of data.

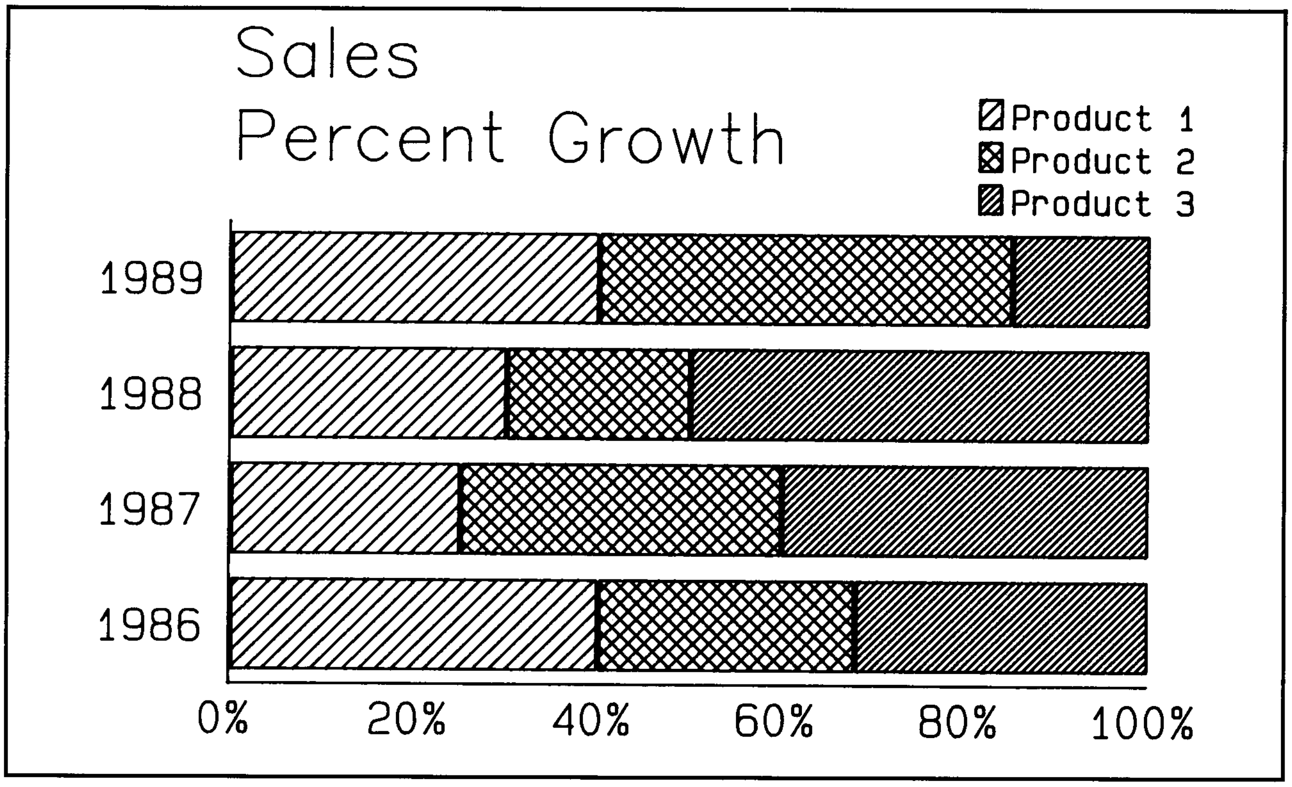


**Figure 2-7. 100% Bar Chart**

For more information, see “Stack a Set of Data Sets” in Chapter 3.

# Line Charts

A *line chart* is simply a set of X/Y data points connected by line segments. When a concentrated number of data points is plotted, the connected line resembles a curve.

## Simple Line Chart

If you need to show the trend or direction indicated by your data set, a *simple line chart* may be the most effective. Projections over time and other kinds of estimates can also be plotted on a line chart as shown in Figure 2-8.

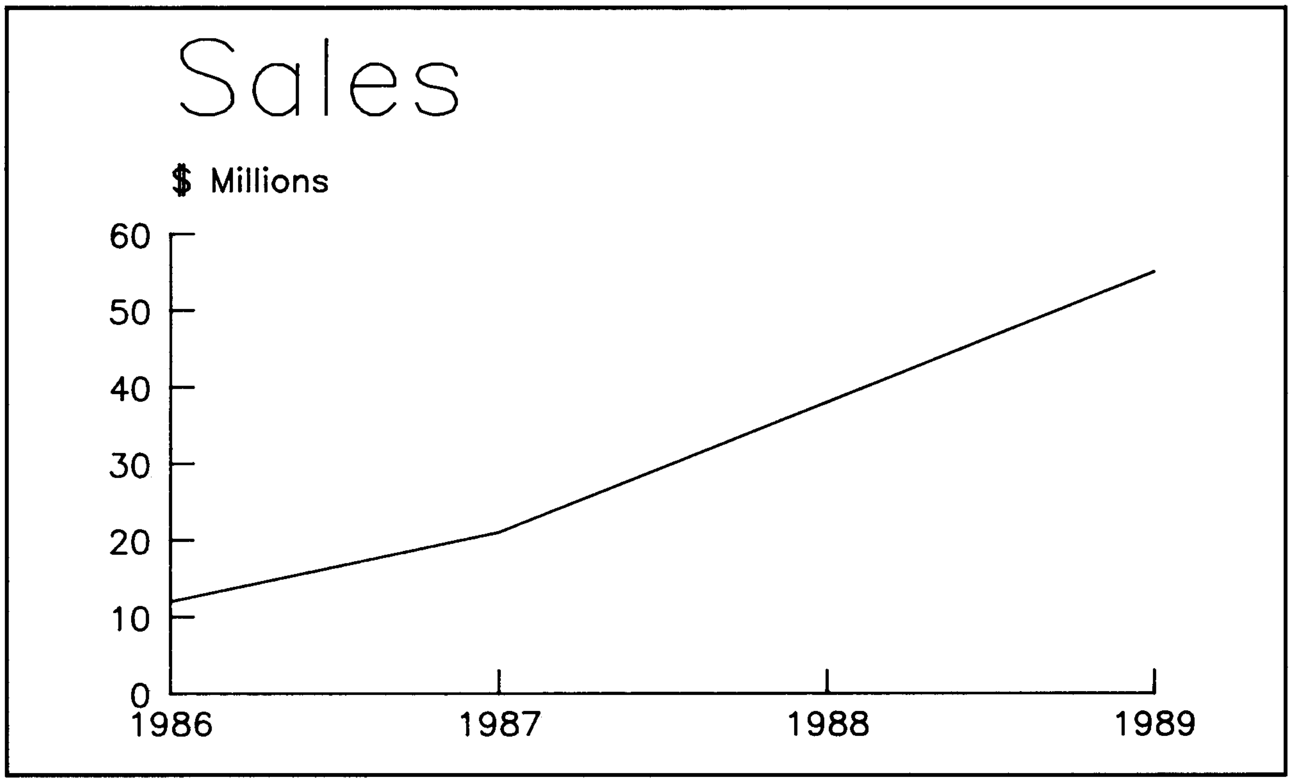


**Figure 2-8. Simple Line Chart**

## Multiple Line Chart

Because the line chart is such a simple graphic form, clarity may be maintained when several lines are combined in a single chart. A *multiple line chart,* such as the one shown in Figure 2-9, is an effective format for comparing several variables, especially when each line is assigned an easily distinguished line style or color and identified by a legend.

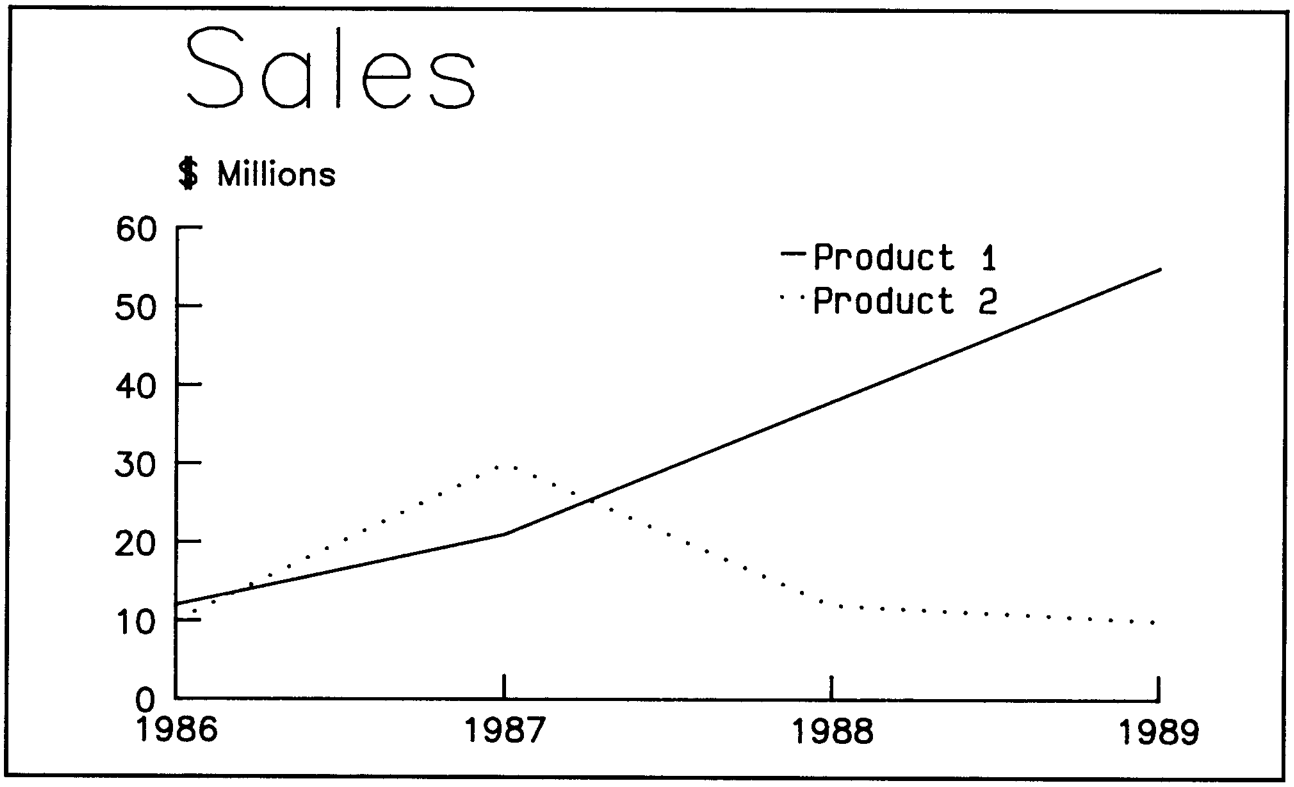


**Figure 2-9. Multiple Line Chart**

## Step Charts

In a *step chart,* data points are connected with lines set at right angles to the axes. Step charts are appropriate when presenting data values with definite limits and no intermediate values.

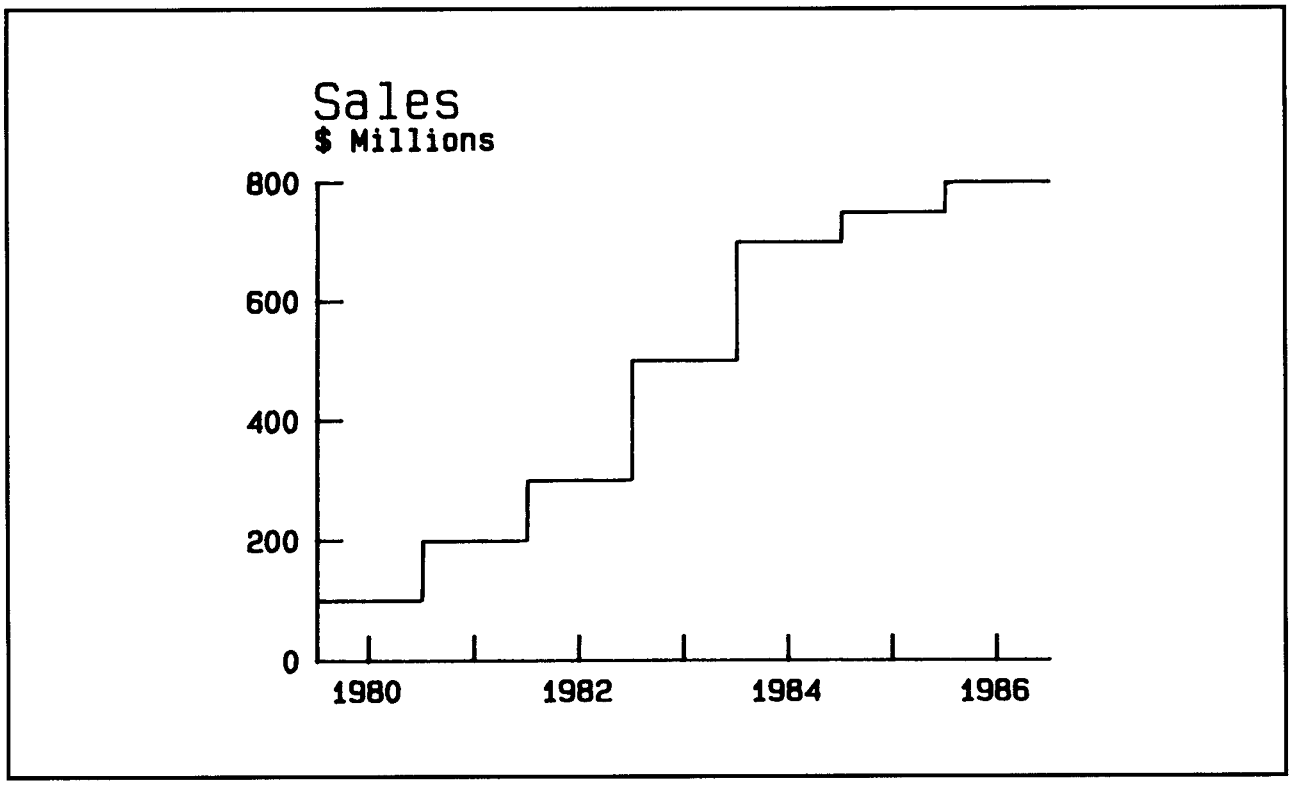


**Figure 2-10. Step Chart**

## Area Charts

If you need to emphasize a volume or a total quantity, an *area chart* can be created. An area chart is a line or step chart with the areas above, below, or between individual lines filled with a distinguishable color or pattern.

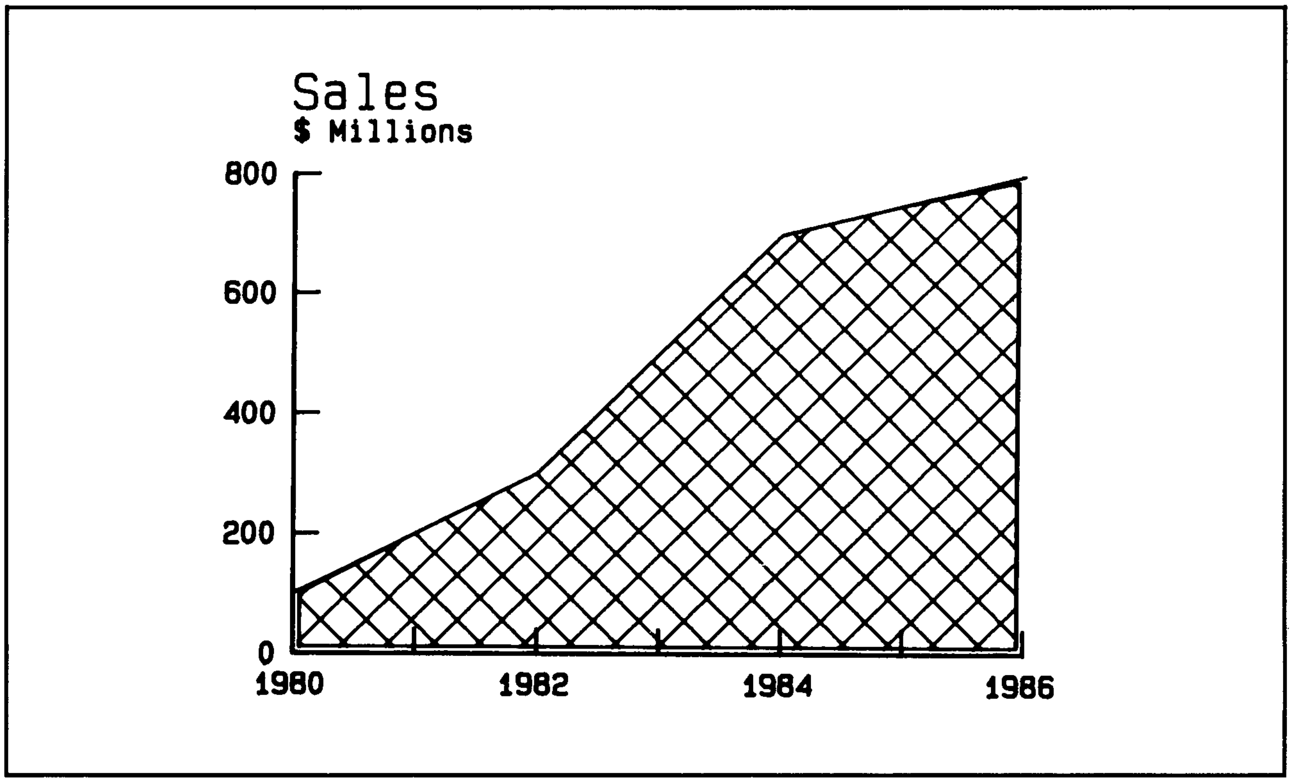


**Figure 2-11. Area Chart**

For more information see “Fill between Two Data Sets” in Chapter 3.

## Scatter Charts

A *scatter chart* represents a distribution of data points over a large area, and can illustrate well any variation between two sets of data.

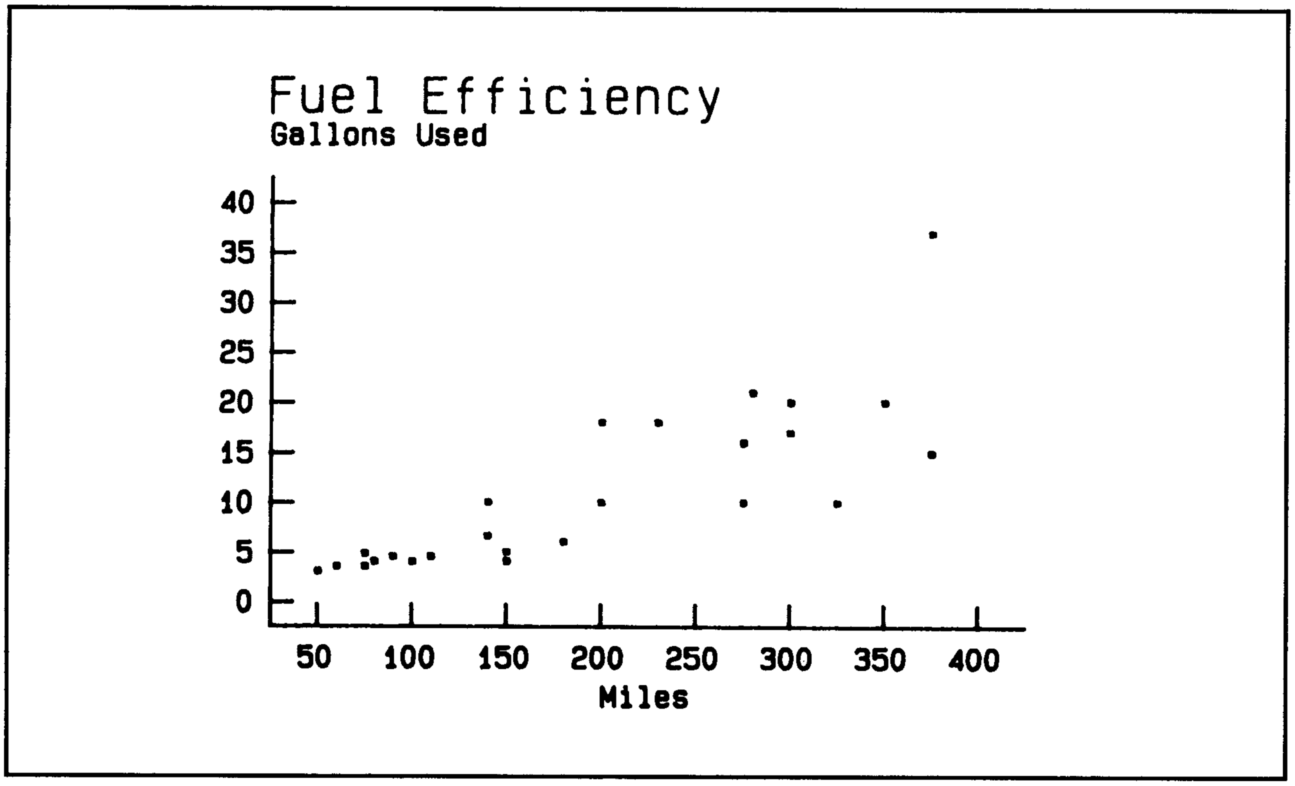


**Figure 2-12. Scatter Chart**

## X/Y-Axis Chart Variations

An X/Y-axis chart can be varied in two ways:

- shifting the dependent axis from horizontal to vertical
- plotting two or more types of X/Y-axis chart on the same axes.

### Vertical or Horizontal Charts

You can specify that the information represented on an X/Y-axis chart extend either vertically or horizontally from the independent axis.

For example in the *vertical bar chart* shown in Figure 2-13, the data set for each of four years has been represented in bars that extend vertically from the independent axis.

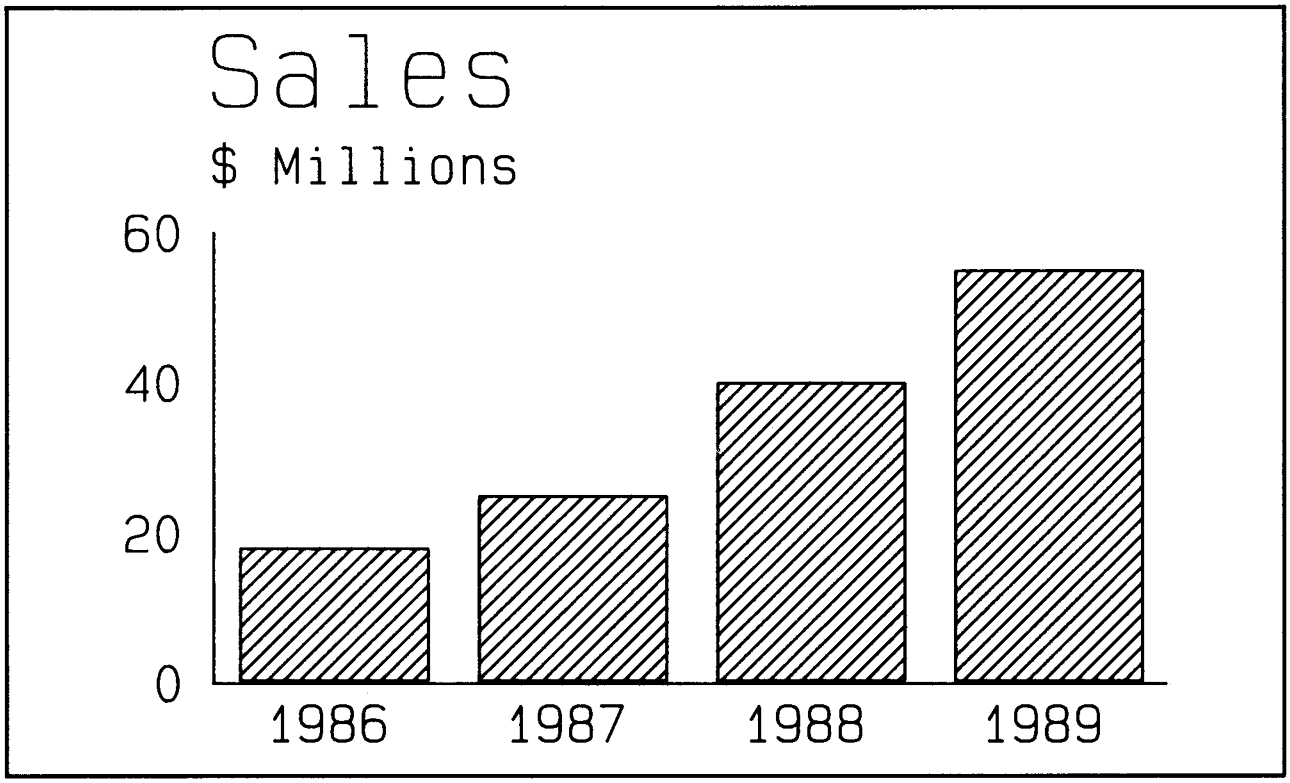


**Figure 2-13. Vertical Bar Chart**

However, you could create a *horizontal bar chart,* with bars extending horizontally from the independent axis, as shown in Figure 2-14.

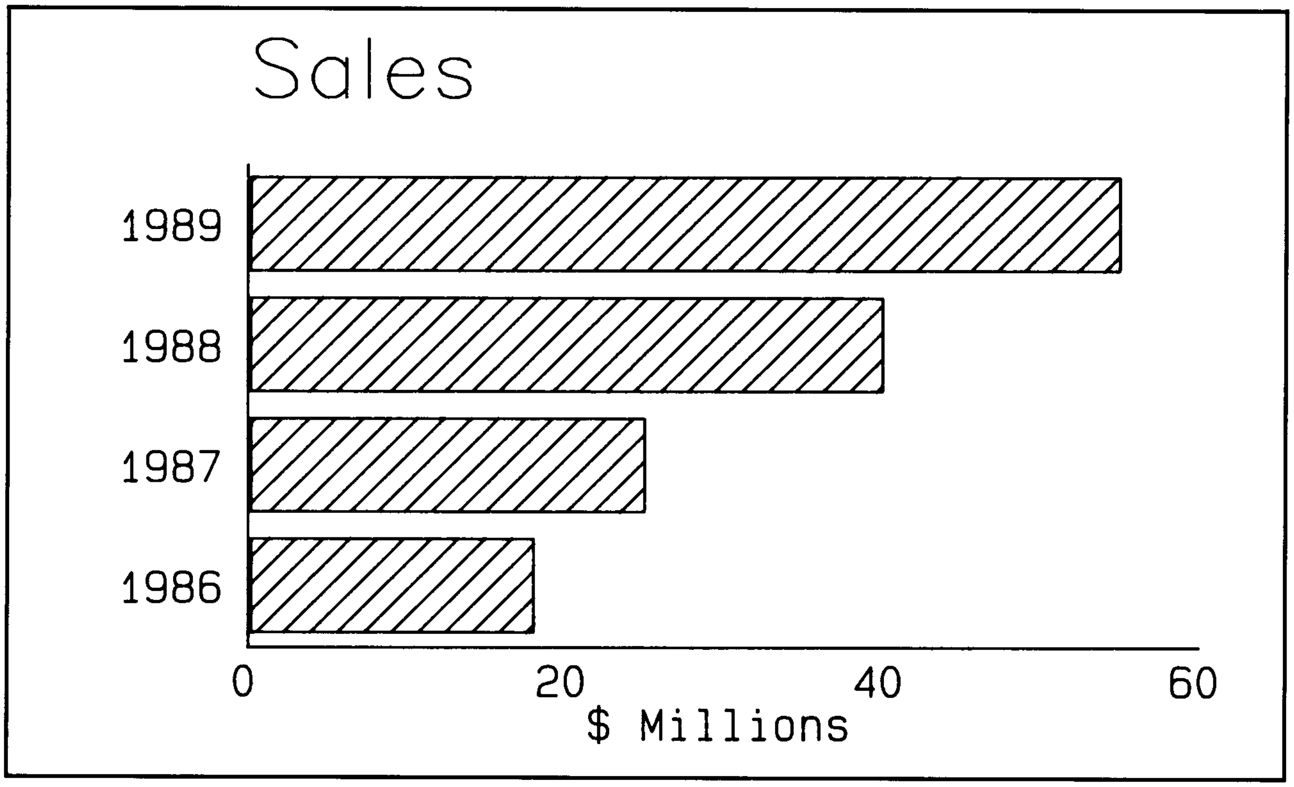


**Figure 2-14. Horizontal Bar Chart**

Switching vertical and horizontal axes with regard to dependent and independent variables can also be done for line, scatter, step and area charts. For more information, see "Set Vertical or Horizontal Chart" in Chapter 3.

## Multiple X/Y-Axis Charts On One Set of Axes

Two or more types of X/Y-axis charts can be plotted on the same set of axes. For example, a line and a bar chart can be plotted on the same set of axes to show the difference between two sets of data, as shown in Figure 2-15. In this case, a line chart was plotted onto the underlying bar chart, and effectively shows actual versus projected sales data.

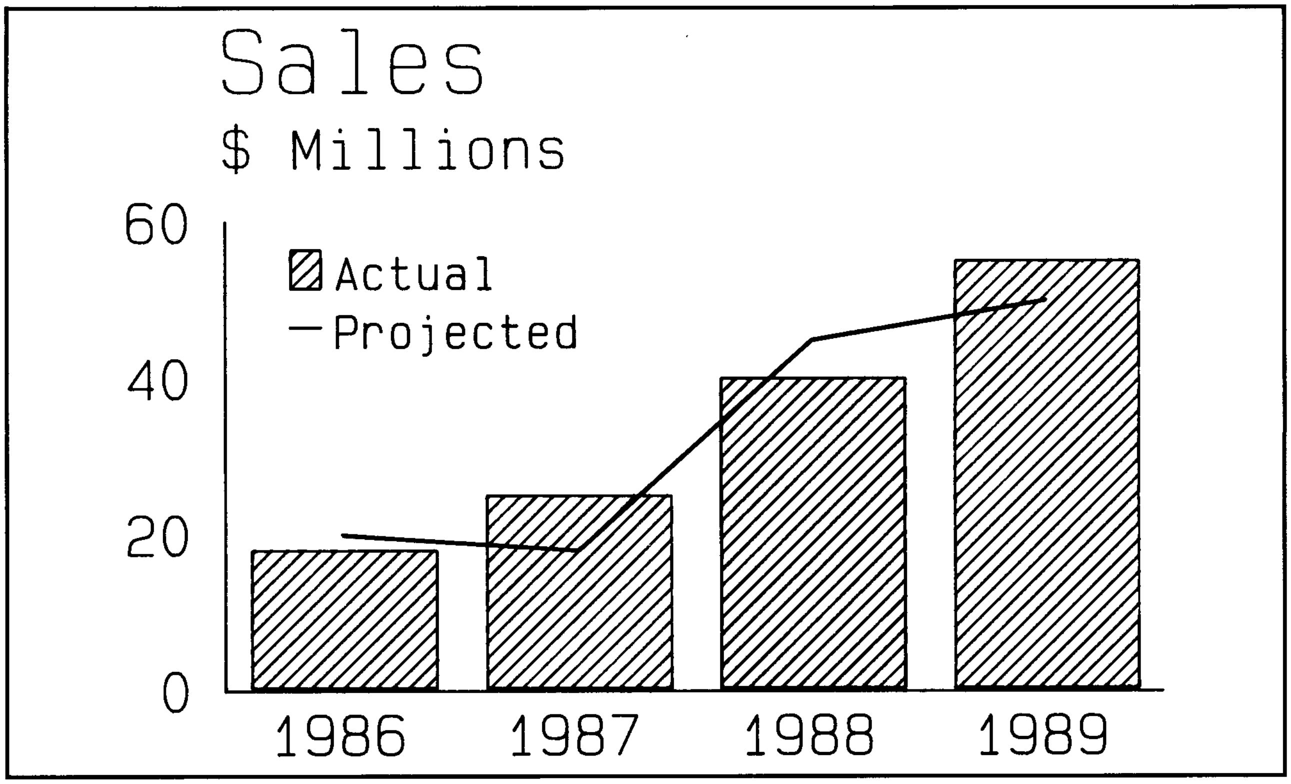


**Figure 2-15. Multiple Chart Types on One Set of Axes**

# Pie Charts

A *pie chart* illustrates the relation of component parts to the whole. Each slice of the pie is proportional in angle and area to the relative size of the data represented.

In Figure 2-16, both pie charts portray equivalent data sets for different years. Plotting two pie charts allows comparison of the market segments. In this case, two pie charts highlight the differences more effectively than a clustered bar chart because only two years of data are represented.

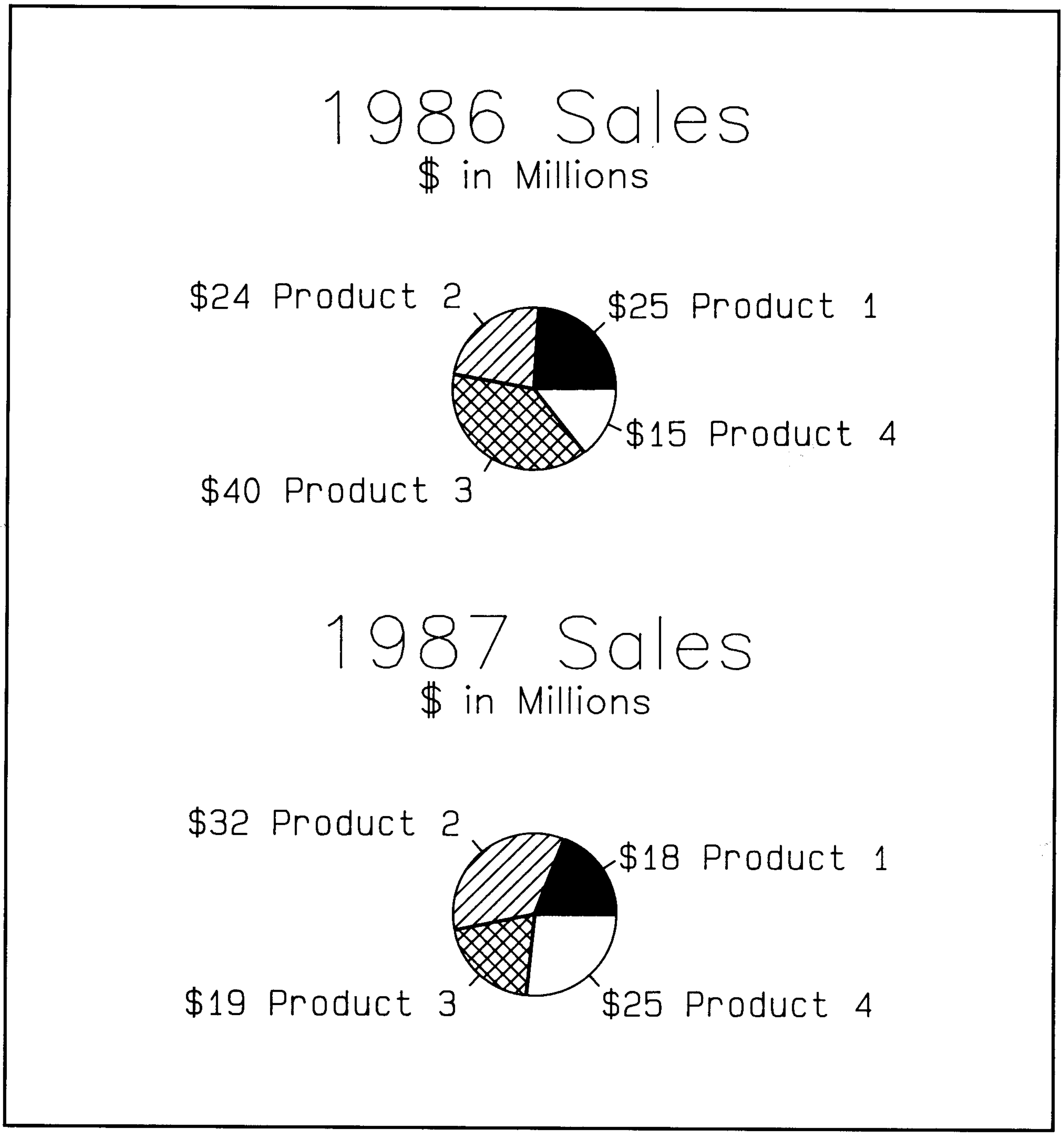


Figure 2-16. Pie Charts

## Exploded Pie Charts

A variation of the pie chart, the *exploded pie chart,* is useful to emphasize the importance of one segment in relation to the others within the pie. When a pie is exploded, the pie slice representing the element to be emphasized is moved away from the center. This is shown in Figure 2-17.

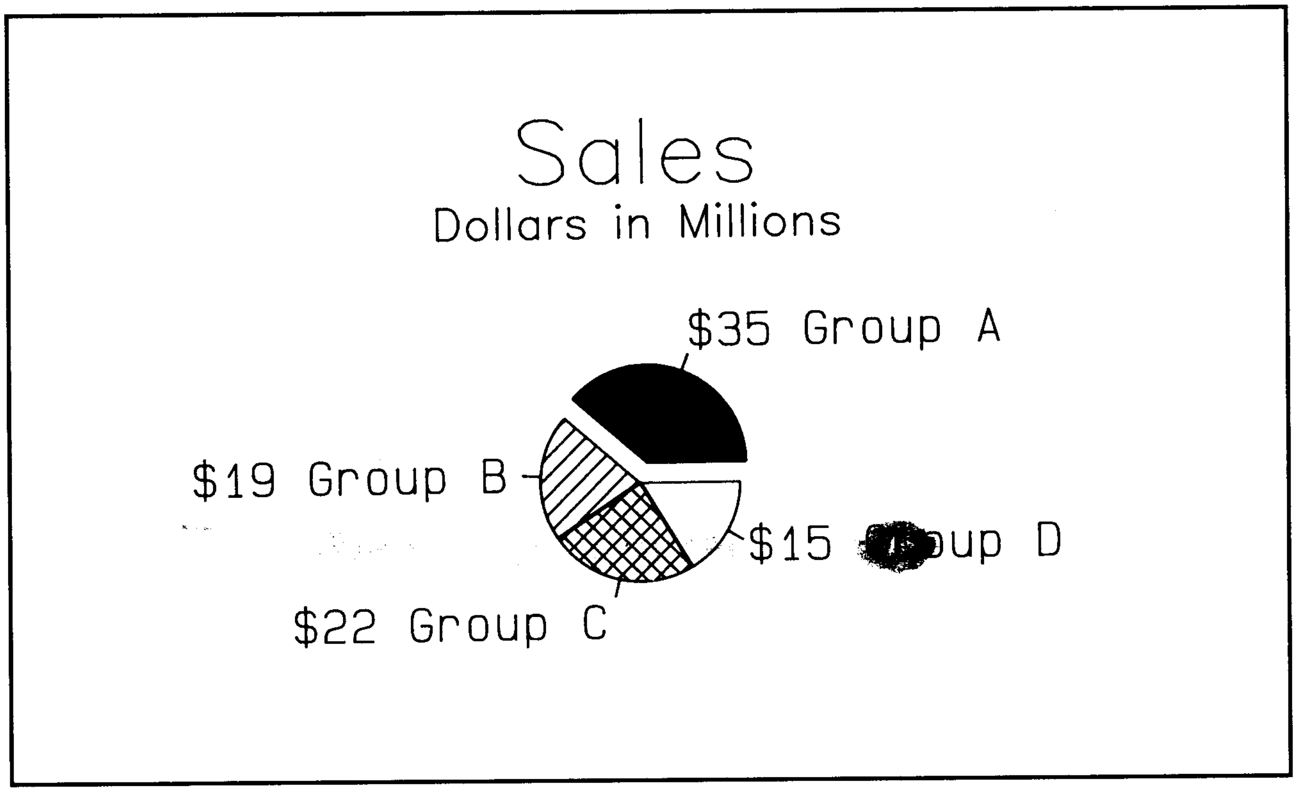


**Figure 2-17. Exploded Pie Chart**

For more information, see “Set Explode of Pie Slice” in Chapter 3.

# Schedule Charts

A *schedule chart* illustrates task status, planning, and degree of task completion. The independent variable is a set of projects or tasks, each represented by a horizontal bar. The dependent variable is some measure of time. Such a chart is shown in Figure 2-18.

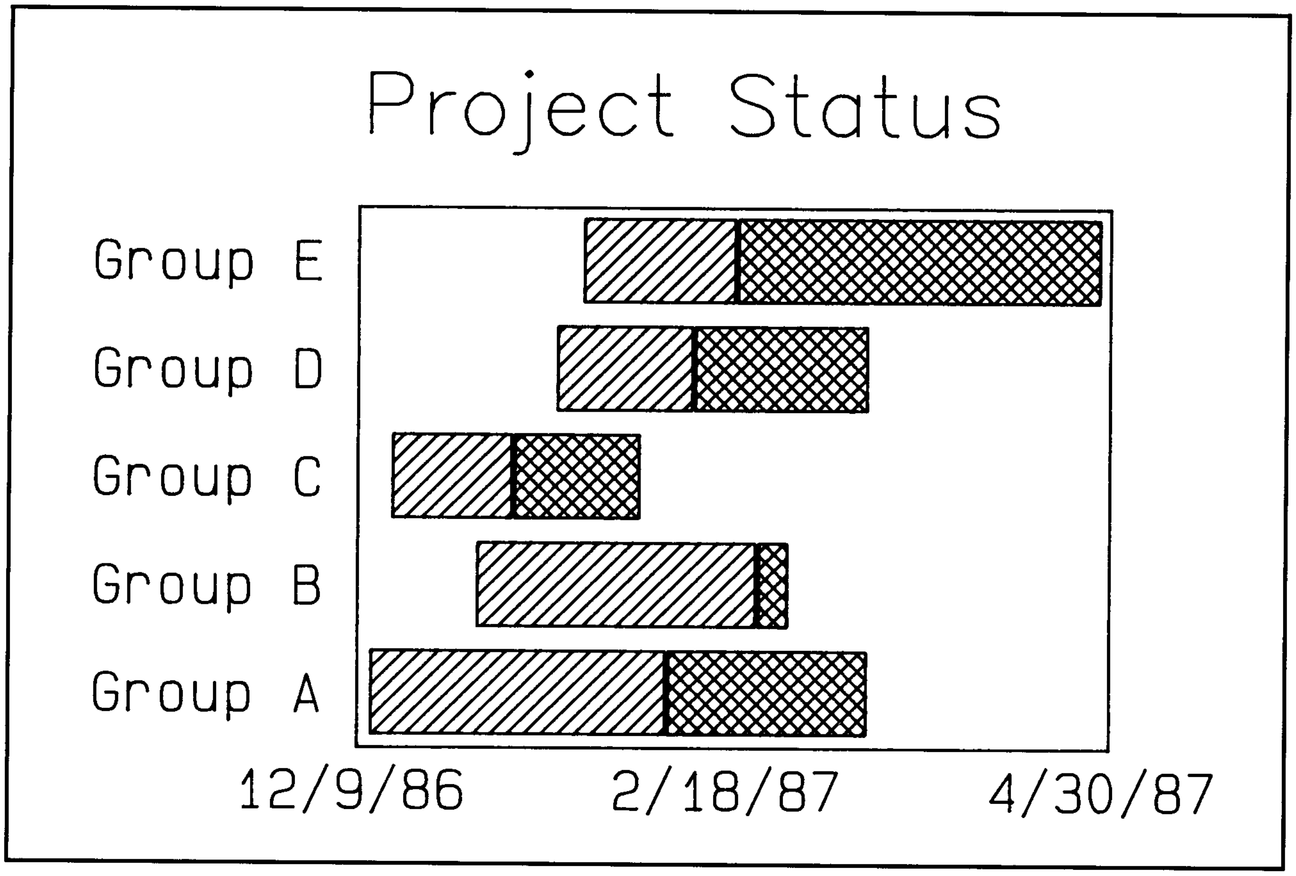


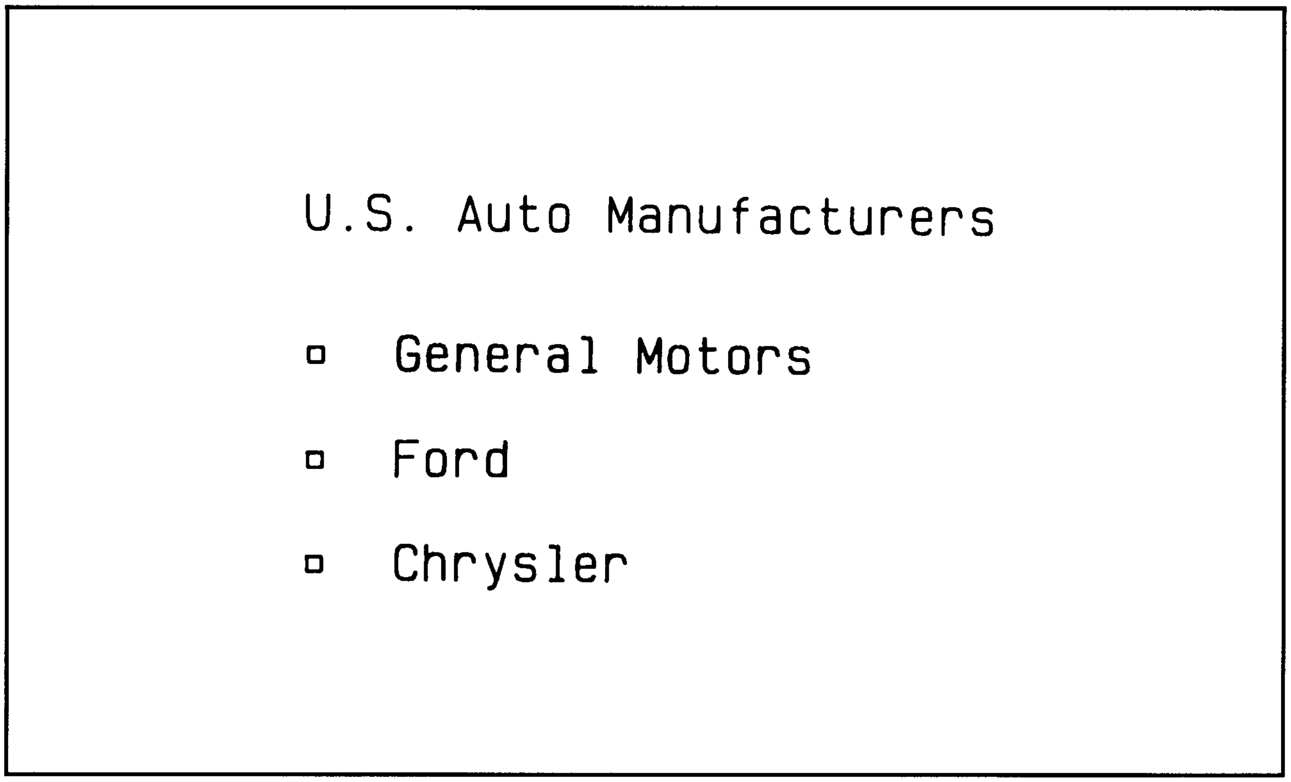
**Figure 2-18. Schedule Chart**

The beginning of each bar is set at the start date of the task, and the end point at the projected finish date. The bars are divided into two segments: the left side represents the completed portion of the task and the right side the portion to be completed. In addition, a vertical line can be drawn on the chart to identify the current date.

For more information, see "Set Date" in Chapter 3.

# Text-Only Charts

A *text-only chart* is composed of words and other graphic symbols. It can be used to enhance the appearance of a document or when organizing an oral presentation. Text-only charts can be enhanced by the use of bullets (large dots), underlining and distinct coloring. To produce a text-only chart, the routines which handle notation strings (Set Notation Alignment, Set Notation Font, Set Notation Color, Set Notation Height) are called without the use of axes, titles or labels. Figure 2-19 shows a typical text-only chart.



U.S. Auto Manufacturers

□ General Motors

□ Ford

□ Chrysler

**Figure 2-19. Text-Only Chart**

# Chapter 3. Plotting System Routines

## CONTENTS

# About This Chapter

This chapter contains descriptions of each routine included in Plotting System. It is intended to provide information about the purpose and operation of the routines. These descriptions are given in terms which are generic with regard to programming languages. Specific syntax for inclusion of the routines in actual programs is given in the Language Reference Booklets provided for supported languages. The routines are arranged in alphabetical order by the generic name for the routine.

# List of Routines by Function

The following list contains the routines provided by Plotting System arranged in groupings by their function. It is provided as a cross-reference to use with the alphabetically arranged descriptions which follow. The groups are:

- Attribute Set Definition Routines
- Control Routines
- Data Set Definition Routines
- Graphic Input Routines
- Independent Chart Component Routines
- Inquiry Routines
- Text Definition Routines.

## Attribute Set Definition Routines

The following routines define (select) the type of chart, and set appropriate attributes for the selected chart type which affect the appearance of the chart:

| **Generic Name** | **Page No.** |
|---|---|
| Define Data Set Chart Type to Be Bar | 3-23 |
| Define Data Set Chart Type to Be Line | 3-24 |
| Define Data Set Chart Type to Be Pie | 3-25 |
| Define Data Set Chart Type to Be Scatter | 3-27 |
| Define Data Set Chart Type to Be Schedule | 3-28 |
| Define Data Set Chart Type to Be Step | 3-29 |
| Fill between Two Data Sets | 3-39 |
| Name Attribute Set | 3-48 |
| Set Explode of Pie Slice | 3-99 |
| Set Output Primitive Color for Data Set | 3-112 |
| Set Output Primitive Outlining for Data Set | 3-114 |
| Set Output Primitive Style for Data Set | 3-116 |
| Set Output Primitive Width for Data Set | 3-118 |
| Set Visibility of Data Set | 3-130 |
| Stack a Set of Data Sets | 3-131 |

## Control Routines

The following routines control the overall plotting environment by:

- Initializing, resetting and terminating the Plotting System
- Assigning (selecting) and closing the display surface to devices; clearing the display surface
- Displaying the currently defined chart.

| **Generic Name** | **Page No.** |
|---|---|
| Assign Plotting System Input Device | 3-15 |
| Assign Plotting System Output Device | 3-16 |
| Clear Display Surface | 3-18 |
| Close Input/Output Device | 3-19 |
| Close Plotting System | 3-20 |
| Open Plotting System | 3-50 |
| Output Currently Defined Chart | 3-51 |
| Reset All Data Sets | 3-64 |
| Reset Plotting System to Default | 3-65 |
| Reset Plotting System Viewing Environment | 3-66 |
| Set Axis Extents | 3-67 |
| Set Axis Indent | 3-69 |
| Set Axis Labels | 3-72 |
| Set Axis Tick Label Type | 3-74 |
| Set Axis Type | 3-79 |
| Set Background Color | 3-81 |
| Set Baseline | 3-82 |
| Set Color Representation | 3-84 |
| Set Date | 3-86 |
| Set Display Surface Size | 3-87 |
| Set Display Surface Units | 3-89 |
| Set Existence of Axis | 3-92 |
| Set Existence of Chart Area Frame | 3-94 |
| Set Existence of Grid Lines | 3-96 |
| Set Existence of View Area Frame | 3-98 |
| Set Foreground Color | 3-102 |
| Set Vertical or Horizontal Chart | 3-128 |
| Set View Area Extents | 3-129 |

## Data Set Definition Routines

The following routines define arrays of data to be schedule, sequenced or X/Y data sets. A fourth routine of this group erases a previously defined data set.

| **Generic Name** | **Page No.** |
|---|---|
| Define Schedule Chart Data Set | 3-32 |
| Define Sequenced Data Set | 3-34 |
| Define X/Y Data Set | 3-37 |
| Undefine a Data Set | 3-133 |

## Graphic Input Routines

The following routines activate the locator, choice and string graphics input modes:

| **Generic Name** | **Page No.** |
|---|---|
| Request Choice | 3-59 |
| Request Locator | 3-60 |
| Request String | 3-62 |

## Independent Chart Component Routines

The following routines provide graphics primitives and notations which may be displayed anywhere within the view area. These primitives and notations operate independently of the basic chart and are displayed as soon as they are defined.

Graphics primitives include lines, rectangles, markers, circles, arcs and arrows, which are often added to a chart to enhance its appearance or clarity.

Notations include character strings (other than chart titles, axis titles, tick labels and legend text), and are often used to add remarks to a chart.

| Generic Name | Page No. |
|---|---|
| Arc | 3-12 |
| Arrow | 3-14 |
| Circle | 3-17 |
| Define Notation String and Location | 3-30 |
| Polyline | 3-53 |
| Polymarker | 3-55 |
| Rectangle | 3-57 |
| Set Fill Color | 3-100 |
| Set Fill Style | 3-101 |
| Set Notation Alignment | 3-107 |
| Set Notation Color | 3-109 |
| Set Notation Font | 3-110 |
| Set Notation Height | 3-111 |
| Set Polyline Color | 3-120 |
| Set Polyline Type | 3-121 |
| Set Polymarker Color | 3-122 |
| Set Polymarker Style | 3-123 |

## Inquiry Routines

The following routines allow inquiry as to the size and attributes of a data set, capabilities of a device, the last error generated and the version number of Plotting System.

| Generic Name | Page No. |
|---|---|
| Inquire on Data Set | 3-41 |
| Inquire on Device | 3-43 |
| Inquire on Plotting System Error | 3-46 |
| Inquire on Software Version | 3-47 |

## Text Definition Routines

The following routines define text components including the chart title, subtitle, axis titles and legend labels. They are commonly used to add remarks to a chart.

| Generic Name | Page No. |
|---|---|
| Define Axis Title String | 3-21 |
| Define Subtitle String | 3-35 |
| Define Title String | 3-36 |
| Set Axis Title Font | 3-76 |
| Set Axis Title Height | 3-78 |
| Set Existence and Location of Legend | 3-90 |
| Set Legend Alignment | 3-103 |
| Set Legend Font | 3-105 |
| Set Legend Height | 3-106 |
| Set Subtitle Font | 3-124 |
| Set Subtitle Height | 3-125 |
| Set Title Font | 3-126 |
| Set Title Height | 3-127 |

# How Routines Are Described

For each routine, information has been organized into the following seven topics:

**Purpose:** The “Purpose:” entry for a routine describes the function or purpose of that routine.

**Format:** The “Format:” entry for a routine shows the name of that routine followed by both its input and output parameters, in parentheses.

Request Choice
(input device, choice)

In the above example, “Request Choice” is the name of the routine while “input device” and “choice” are parameters.

**Input:** The Input parameters are arrays or variables to which you must assign a value before calling the Plotting System routines. Each input parameter listed in the “Format:” entry is described in “Input:”.

**Output:** Output parameters are arrays or variables that receive values from the Plotting System routine that is called by your application program. Each output parameter listed in the “Format:” entry is described in “Output:”.

**Related Routines:** Routines whose functions are related to the routine under discussion are listed in this entry.

**Status:** Each Plotting System routine returns a status code to the application program at the completion of the routine. A zero means no error, and a -1 indicates error.

**Remarks:** The “Remarks:” entry for a routine contains information that is particularly important to understanding the operation of a routine. It includes qualifications and notes about the effects the routine being described has on other routines and on Plotting System.

**Note:** Arrays are described in this book as having a given number of elements. Each element of the array is briefly described following the array description. Refer to the Language Reference Booklets for a definition of data types for each array.

# Arc

**Purpose:** Generates a filled arc.

**Format:** Arc
(coordinate, center, radius, start, end)

**Input:**

**coordinate**
Location on the display surface from which *center* is referenced and *radius* is calculated.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**center[2]**
Array of X and Y coordinates of the center of the arc. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.*

**radius**
Radius of the arc as a percentage of the horizontal dimension of the area specified by *coordinate.*

**start**
Starting angle in degrees. The range is 0.0 - 360.0. Zero degrees is at the three o'clock position. Positive angles move in a counterclockwise direction. Angles ⟩ 360.0 are modulo 360.0; for example, 450.0 becomes 90.0.

**end**
Ending angle in degrees. The range is 0.0 - 360.0. Zero degrees is at the three o'clock position. Positive angles move in a counterclockwise direction. Angles ⟩ 360.0 are modulo 360.0; for example, 450.0 becomes 90.0.

**Output:** None

**Related Routines:** Set Fill Color, Set Fill Style

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate*. The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate*.

The arc is output as soon as this routine is called.

Any part of the arc that is specified to lie outside the view area is clipped.

When the output device is a raster display, the area defined by the arc parameters is erased with the background color before the arc is drawn.

# Arrow

**Purpose:** Generates a filled arrow.

**Format:** Arrow
(coordinate, start, end)

**Input:**

**coordinate**
Location on the display surface from which *start* and *end* are referenced.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**start[2]**
Array of X and Y coordinates of the arrow tip point. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.*

**end[2]**
Array of X and Y coordinates of the arrow end point. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.*

**Output:** None

**Related Routines:** Set Fill Color, Set Fill Style

**Status:**
 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

The arrow is output as soon as this routine is called.

Any part of the arrow that is specified to lie outside the view area is clipped.

When the output device is a raster display, the area defined by the arrow parameters is erased with the background color before the arrow is drawn.

# Assign Plotting System Input Device

**Purpose:** Opens the graphics input device.

**Format:** Assign Plotting System Input Device
(device)

**Input:** **device []**
Graphics input device. A string of up to eight characters.

**Output:** None

**Related Routines:** Assign Plotting System Output Device, Close Input/Output Device, Request Choice, Request Locator, Request String

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine must be called before attempting graphics input.

In some cases, the same device may be used for both graphics input and output, such as the display.

This routine opens *device.*

Before opening a new input device, you must close the currently opened input device.

When using an input device for locator input, such as a mouse, an output display device must also be assigned to echo input device action.

# Assign Plotting System Output Device

**Purpose:** Opens the graphics output device.

**Format:** Assign Plotting System Output Device
(device)

**Input:** **device []**
Graphics output device. A string of up to eight characters.

**Output:** None

**Related Routines:** Assign Plotting System Input Device, Close Input/Output Device

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine must be called before the Output Currently Defined Chart routine is called; otherwise, output is not visible.

This routine opens *device.*

In some cases, the same device may be used for both input and output, such as the display.

Before opening a new output device, you must close the currently opened output device.

When using an input device for locator input, such as a mouse, an output display device must also be assigned to echo input device action.

# Circle

**Purpose:** Generates a filled circle.

**Format:** Circle
(coordinate, center, radius)

**Input:** **coordinate**
Location on the display surface from which *center* and *radius* are referenced.

1=lower-left corner of the view area
2=lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**center[2]**
Array of X and Y coordinates of the center of the circle. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.*

**radius**
Radius of the circle as a percentage of the horizontal dimension area specified by *coordinate.* The value of *radius* must be ⟩ 0.0.

**Output:** None

**Related Routines:** Set Fill Color, Set Fill Style

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

The circle is output as soon as this routine is called.

Any part of the circle that is specified to lie outside the view area is clipped.

When the output device is a raster display, the area defined by the circle parameters is erased with the background color before the circle is drawn.

## Clear Display Surface

**Purpose:** Causes CRT screens to be cleared, prompts for new paper on plotters, or displays all pending graphics and advances to top-of-form on printers.

**Format:** Clear Display Surface

**Input:** None

**Output:** None

**Related Routines:** Close Plotting System, Open Plotting System, Output Currently Defined Chart

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Either this routine or Close Plotting System must be invoked after Output Currently Defined Chart to produce the image on a printer device.

When the output device is a raster display, the area defined as the display surface is erased with the background color when this routine is called.

# Close Input/Output Device

**Purpose:** Closes an input or output device.

**Format:** Close Input/Output Device
(device)

**Input:** **device []**
Device to be closed. A string of up to eight characters.

**Output:** None

**Related Routines:** Assign Plotting System Input Device, Assign Plotting System Output Device, Close Plotting System

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Close each open device before calling the Close Plotting System routine.

Close any open input device before opening a new device for input.

Close any open output device before opening a new device for output.

If the device is already closed or has never been opened, this routine is ignored.

If an input and an output device are open with the same name, the input device is closed the first time this routine is called. This routine must be called again to close the output device.

# Close Plotting System

**Purpose:** Terminates Plotting System and releases all memory allocated to Plotting System.

**Format:** Close Plotting System

**Input:** None

**Output:** None

**Related Routines:** Open Plotting System

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine must be the last Plotting System routine called. If you want to generate another chart after calling this routine, call the Open Plotting System routine again.

# Define Axis Title String

**Purpose:** Specifies the string for the axis title.

**Format:** Define Axis Title String
(axis, string)

**Input:** **axis**
Axis to be titled.

1 = independent
2 = dependent

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**string[]**
Axis title.

**Output:** None

**Related Routines:** Set Axis Title Font, Set Axis Title Height, Set Foreground Color

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

If the title is redefined, a value of 1 is returned.

**Remarks:** If this routine is not called, no axis titles are produced.

The axis title appears in the foreground color.

The vertical axis title is left-justified over the Y-axis. The horizontal axis title is centered below the X-axis.

This routine has no effect on pie or text-only charts.

To maintain appropriate chart proportions, Plotting System first fits the defined axis title string into the vertical view area space, assuring that at least one-third of the space is still available for the chart area. All graphic data output will be contained within the chart area.

If the axis title does not fit, all currently defined text strings (chart titles, subtitles, and axis titles) are scaled to a size that enables all text to be displayed.

After resolving the vertical spacing, Plotting System evaluates the horizontal dimensions of the axis title in relation to other currently defined text strings and the required chart area size. Text is scaled to a size that leaves at least half the horizontal dimension of the view area available for the chart area.

If text size is at a minimum and the defined axis title still does not fit, Plotting System positions the title to accommodate as much of the string as possible. Those portions that fall outside the view area are clipped.

# Define Data Set Chart Type to Be Bar

**Purpose:** Assigns a data set to a set of bars.

**Format:** Define Data Set Chart Type to Be Bar
(data set id)

**Input:** **data set id**
Unique data set identification number. The value of *data set id* must be 〉 0.

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Define Sequenced Data Set, Define X/Y Data Set, Output Currently Defined Chart, Stack a Set of Data Sets

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When the chart is output, bars will represent the data set identified by *data set id.* See "Assigning the Chart Type" in Chapter 1, "Using Plotting System."

When the data set identified by *data set id* is an X/Y data set, the independent and dependent data is plotted as a set of bars.

When the data set identified by *data set id* is a sequenced data set, the sequence numbers are plotted relative to the X-axis and the dependent data values are plotted relative to the Y-axis.

When the data set identified by *data set id* is a schedule data set, no chart is output.

Bars can be stacked by calling the Stack a Set of Data Sets routine. If the Stack a Set of Data Sets routine is not called, multiple bar data sets are clustered at the tick marks on the independent axis.

For a quick reference to the data set types, see the table in the "Defining the Data Sets" section in Chapter 1, "Using Plotting System."

# Define Data Set Chart Type to Be Line

**Purpose:** Assigns a data set to line representation.

**Format:** Define Data Set Chart Type to Be Line
(data set id)

**Input:** **data set id**
Unique data set identification number. The value of *data set id* must be 〉 0.

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Define Sequenced Data Set, Define X/Y Data Set, Fill between Two Data Sets, Output Currently Defined Chart, Stack a Set of Data Sets

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When the chart is produced, a line will represent the data set identified by *data set id.* See "Assigning the Chart Type" in Chapter 1, "Using Plotting System."

When the data set identified by *data set id* is an X/Y data set, the independent and dependent data is charted as a line.

When the data set identified by *data set id* is a sequenced data set, the sequence numbers are plotted relative to the X-axis and the dependent data values are plotted relative to the Y-axis.

When the data set identified by *data set id* is a schedule data set, no chart is output.

An area chart can be created by calling the Fill between Two Data Sets routine.

For a quick reference to the data set types, see the table in the "Defining the Data Sets" section in Chapter 1, "Using Plotting System."

# Define Data Set Chart Type to Be Pie

**Purpose:** Assigns a data set to pie chart representation.

**Format:** Define Data Set Chart Type to Be Pie
(data set id)

**Input:** **data set id**
Unique data set identification number. The value of *data set id* must be 〉 0.

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Define Sequenced Data Set, Define X/Y Data Set, Output Currently Defined Chart

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When the chart is generated, a pie will represent the data set identified by *data set id.* See "Assigning the Chart Type" in Chapter 1, "Using Plotting System."

Only one data set can be defined as a pie chart at a time. If more than one data set is defined as a pie chart, only the last set defined is used. All previously defined pie charts are reset to line charts.

When the data set identified by *data set id* is a sequenced data set, the dependent data is plotted as a pie. Pie slices are numbered sequentially and slice numbers become the identifiers for slice attributes.

When the data set identified by *data set id* is an X/Y data set, only the dependent data is plotted. The independent data is ignored.

When the data set identified by *data set id* is a schedule data set, no chart is output.

The destination of any attributes defined for a pie chart is identified by the slice number. Slice numbers correspond to the position of the data item in the data set.

A maximum of 16 slices can be formed. If more than 16 data values are defined, only the first 16 are displayed. Those 16 values are used to calculate the percentage value of each slice.

For a quick reference to the data set types, see the table in the “Defining the Data Sets” section in Chapter 1, “Using Plotting System.”

# Define Data Set Chart Type to Be Scatter

**Purpose:** Assigns a data set to a set of markers.

**Format:** Define Data Set Chart Type to Be Scatter
(data set id)

**Input:** **data set id**
Unique data set identification number. The value of *data set id* must be 〉 0.

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Define Sequenced Data Set, Define X/Y Data Set, Output Currently Defined Chart, Stack a Set of Data Sets

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When the chart is generated, a marker will represent each data point in the data set identified by *data set id.*

When the data set identified by *data set id* is an X/Y data set, the independent and dependent data is plotted as a set of markers.

When the data set identified by *data set id* is a sequenced data set, the sequence numbers are plotted relative to the X-axis and the dependent data values are plotted relative to the Y-axis.

When the data set identified by *data set id* is a schedule data set, no chart is output.

For a quick reference to the data set types, see the table in the "Defining the Data Sets" section in Chapter 1, "Using Plotting System."

# Define Data Set Chart Type to Be Schedule

**Purpose:** Indicates chart output is a schedule chart.

**Format:** Define Data Set Chart Type to Be Schedule

**Input:** None

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Output Currently Defined Chart, Set Date

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When the Output Currently Defined Chart routine is called, all defined schedule data sets are shown on a schedule chart.

A schedule chart usually involves a number of schedule data sets, one for each task bar.

A line can be drawn at a specified date using the Set Date routine.

For a quick reference to the data set types, see the table in the "Defining the Data Sets" section in Chapter 1, "Using Plotting System."

# Define Data Set Chart Type to Be Step

**Purpose:** Assigns a data set to step representation.

**Format:** Define Data Set Chart Type to Be Step
(data set id)

**Input:** **data set id**
Unique data set identification number. The value of *data set id* must be ⟩ 0.

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Define Sequenced Data Set, Define X/Y Data Set, Fill between Two Data Sets, Output Currently Defined Chart, Stack a Set of Data Sets

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When the chart is generated, the data points contained in the data set identified by *data set id* will be connected by a series of horizontal and vertical lines resembling steps.

When the data set identified by *data set id* is an X/Y data set, the independent and dependent data is plotted as a stepped line.

When the data set identified by *data set id* is a sequenced data set, the sequence numbers are plotted relative to the X-axis and the dependent data values are plotted relative to the Y-axis.

When the data set identified by *data set id* is a schedule data set, no chart is output.

An area chart can be created by calling the Fill between Two Data Sets routine.

For a quick reference to the data set types, see the table in the "Defining the Data Sets" section in Chapter 1, "Using Plotting System."

# Define Notation String and Location

**Purpose:** Defines a notation string at the specified location on the display surface.

**Format:** Define Notation String and Location
(coordinate, x, y, string)

**Input:**

**coordinate**
Location on the display surface from which x and y are referenced.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**x**
X coordinate of the notation location as a percentage of the area specified by *coordinate.* The range is 0.0 - 100.0.

**y**
Y coordinate of the notation location as a percentage of the area specified by *coordinate.* The range is 0.0 - 100.0.

**string[]**
Notation string.

**Output:** None

**Related Routines:** Set Notation Alignment, Set Notation Color, Set Notation Font, Set Notation Height

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

Notation strings defined with this routine can be used to annotate a chart with additional text.

The notation is output as soon as this routine is called.

Any part of the notation that is specified to lie outside the view area is clipped.

Text-only charts are created using this routine.

# Define Schedule Chart Data Set

**Purpose:** Defines a schedule chart data set.

**Format:** Define Schedule Chart Data Set
(data set id, start, end, percent)

**Input:**

**data set id**
Unique data set identification number. The value of *data set id* must be 〉 0.

**start[8]**
Starting date of the schedule. The format for *start* is “MM/DD/YY” where:

MM=two-digit month (1-12)
DD=two-digit day (1-31)
YY=two-digit year (00-99)
/=required delimiter.

**end[8]**
Ending date of the schedule. The format for *end* is “MM/DD/YY” where:

MM=two-digit month (1-12)
DD=two-digit day (1-31)
YY=two-digit year (00-99)
/=required delimiter.

If *start* 〉 *end, end* is set to *start* + 1 day.

**percent**
Percentage of the task that is complete. The range for *percent* is 0 - 100. Values 〈 0 are set to 0 and values 〉 100 are set to 100.

**Output:** None

**Related Routines:** Define Data Set Chart Type to Be Schedule, Set Date, Set Output Primitive Color for Data Set, Set Output Primitive Outlining for Data Set, Set Output Primitive Style for Data Set, Set Output Primitive Width for Data Set

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine defines data for a schedule chart.

Each bar of a schedule chart represents a different task and requires a unique data set identification number.

Use Set Output Primitive Color for Data Set to define the color and Set Output Primitive Style for Data Set to define the fill style of the schedule chart task bars. Attribute Set 1 defines color and fill style for the bars showing completed portions of the task, and Attribute Set 2 defines color and fill style for the bars showing the incomplete portions of the task.

The width of the schedule bar (complete and incomplete portions) is defined by setting attribute set 1. The outline color of the schedule bars can also be set with Set Output Primitive Outlining for Data Set.

# Define Sequenced Data Set

**Purpose:** Defines a sequenced data set.

**Format:** Define Sequenced Data Set
(data set id, number, values)

**Input:** **data set id**
Unique data set identification number. The value of *data set id* must be ⟩ 0.

**number**
Number of elements in the data set. The value of *number* must be ⟩ 0.

**values[]**
Data elements. The number of elements in *values* must equal *number.*

**Output:** None

**Related Routines:** Define Data Set Chart Type to Be Bar, Define Data Set Chart Type to Be Line, Define Data Set Chart Type to Be Pie, Define Data Set Chart Type to Be Scatter, Define Data Set Chart Type to Be Step, Define X/Y Data Set

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The array of dependent data values is automatically associated with an independent sequence beginning with 1, incremented by 1, and ending with *number.*

This routine can be used to define a data set for a pie chart. If the data is for a pie chart, each element of the array values is evaluated as a percentage of the total. When plotted, the total is represented by the entire pie and each element is represented by a pie slice, up to a maximum of 16 slices.

This routine can be used to define data for bar, line, scatter, or step charts when the independent axis scale is arithmetic (1, 2, 3, 4, n) or when independent axis tick marks are labeled.

# Define Subtitle String

**Purpose:** Specifies the string for the chart subtitle.

**Format:** Define Subtitle String
(string)

**Input:** **string[]**
Chart subtitle.

**Output:** None

**Related Routines:** Set Foreground Color, Set Subtitle Font, Set Subtitle Height

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

If the subtitle is redefined, a value of 1 is returned.

**Remarks:** If this routine is not called, no subtitle is output.

On pie charts, the subtitle is centered at the top of the view area. On all other chart types, the subtitle is left-justified over the vertical axis.

The subtitle appears in the foreground color.

To maintain appropriate chart proportions, Plotting System first fits the defined subtitle string into the vertical view area space, assuring that at least one-third of the space is still available for the chart area. All graphic data output will be contained within the chart area.

If the subtitle does not fit, all currently defined text strings (chart titles, subtitles, and axis titles) are scaled to a size that enables all text to be displayed.

After resolving the vertical spacing, Plotting System evaluates the horizontal dimensions of the subtitle in relation to other currently defined text strings and the required chart area size. Text is scaled to a size that leaves at least one-half the horizontal dimension of the view area available for the chart area.

If text size is at a minimum and the defined subtitle still does not fit, Plotting System positions the subtitle to accommodate as much of the string as possible. Those portions that fall outside the view area are clipped.

# Define Title String

**Purpose:** Specifies the string for the chart title.

**Format:** Define Title String
(string)

**Input:** **string[]**
Chart title.

**Output:** None

**Related Routines:** Set Foreground Color, Set Title Font, Set Title Height

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

If the title is redefined, a value of 1 is returned.

**Remarks:** If this routine is not called, no title is output.

On pie charts, the title is centered at the top of the view area. On all other chart types, the title is left-justified over the Y-axis.

The title appears in the foreground color.

To maintain appropriate chart proportions, Plotting System first fits the defined title string into the vertical view area space, assuring that at least one-third of the space is still available for the chart area. All graphic data output will be contained within the chart area.

If the title does not fit, all currently defined text strings (chart titles, subtitles, and axis titles) are scaled to a size that enables all text to be displayed.

After resolving the vertical spacing, Plotting System evaluates the horizontal dimensions of the title string in relation to other currently defined text strings and the required chart area size. Text is scaled to a size that leaves at least one-half the horizontal dimension of the view area available for the chart area.

If text size is at a minimum and the defined title still does not fit, Plotting System positions the title to accommodate as much of the string as possible. Those portions that fall outside the view area are clipped.

# Define X/Y Data Set

**Purpose:** Defines an X/Y data set.

**Format:** Define X/Y Data Set
(data set id, number, xvalues, yvalues)

**Input:**

**data set id**
Unique data set identification number. The value of *data set id* must be 〉 0.

**number**
Number of elements in the data set. The value of *number* must be 〉 0.

**xvalues[]**
Independent data elements. The number of elements in *xvalues* must equal *number.* The elements of *xvalues* must be in order of increasing value.

**yvalues[]**
Dependent data elements. The number of elements in *yvalues* must equal *number.*

**Output:** None

**Related Routines:** Define Data Set Chart Type to Be Bar, Define Data Set Chart Type to Be Line, Define Data Set Chart Type to Be Pie, Define Data Set Chart Type to Be Scatter, Define Data Set Chart Type to Be Step, Define Sequenced Data Set, Set Vertical or Horizontal Chart

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine can be used to define a data set for a bar, line, scatter, or step chart. Each element of *xvalues* is associated with the corresponding element of *yvalues.* For example, *xvalues[1]* and *yvalues[1]* define one data point.

When generating a vertically-oriented X/Y-axis chart, the data in *xvalues* is plotted relative to the horizontal axis.

When generating a horizontally-oriented X/Y-axis chart, the data in *xvalues* is plotted relative to the vertical axis.

This routine can be used to define data for pie charts. In this case, only the dependent data is represented by the pie chart.

If the elements of *xvalues* are not in increasing order, unexpected results may occur.

# Fill between Two Data Sets

**Purpose:** Defines two data sets as boundaries of an area to be filled.

**Format:** Fill between Two Data Sets
(area1, area2)

**Input:**

**area1**
Data set identification number of the first boundary of the filled area. The value of *area1* must be ≥ 0.

**area2**
Data set identification number of the second boundary of the filled area.

**Output:** None

**Related Routines:** Define Data Set Chart Type to Be Line, Define Data Set Chart Type to Be Step, Set Baseline, Set Output Primitive Color for Data Set, Set Output Primitive Style for Data Set, Set Visibility of Data Set

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The fill style assigned to the *area1* data set by the Set Output Primitive Style for Data Set routine determines the fill style used for the area chart.

The color assigned to the *area1* data set by the Set Output Primitive Color for Data Set routine determines the fill color used for the area chart.

Only data sets defined to be line or step charts can be used as boundaries for the filled area. The boundaries always appear as solid lines.

If neither *area1* nor *area2* data set exists, this routine is ignored.

If only one of the *area1* or *area2* data sets exists, the fill begins at the top of the chart area and continues to the data set that is defined.

If the value entered for the *area2* data set is zero, the area between the area1 data set and the baseline will be filled.

If an alternate baseline has not been specified, the fill will continue to the independent axis.

If either *area1* or *area2* is set to not visible, the fill will appear without the line representing that data set.

If both *area1* and *area2* are set to not visible, the defined area will be filled, but without the lines representing the data sets.

# Inquire on Data Set

**Purpose:** Returns information about the specified data set.

**Format:** Inquire on Data Set
(data set id, attribute set id, length, array, number)

**Input:**

**data set id**
Data set identification number. The value of *data set id* must be 〉 0.

**attribute set id**
Attribute identification number. The number must be valid for the given data set.

**length**
Number of elements in *array.*

**Output:**

**array[8]**
Inquiry information about the data set identified by *data set id.* The number of elements in *array* must equal *length.*

The elements of *array* are:

**array[1]**
width

**array[2]**
color

**array[3]**
style

**array[4]**
outline flag

0 = primitive color
1 = foreground color

**array[5]**
pie slice explode flag

0 = no
1 = yes

**array[6]**
data set existence flag

0=no
1=yes

**array[7]**
chart type

1=line
2=step
3=bar
4=scatter
5=schedule
6=pie

**array[8]**
number of values in data set

**number**
Number of elements actually placed in *array* by this routine.

**Related Routines:** None

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** For pie charts, *data set id* refers to the pie slice number.

For schedule chart, *data set id* = 1 refers to the segment of the task bar that represents the completed portion of the task and *data set id* = 2 refers to the segment of the task bar that represents the incomplete portion of the task.

If a schedule chart is defined and *data set id* is 〉 2, elements 1 - 6 of *array* are set to -1.

If *length* is 〈 8, only elements 1 - length of *array* are returned.

The elements of *array* for which attributes are not explicitly defined are set to -1.

# Inquire on Device

**Purpose:** Obtains information about the specified device.

**Format:** Inquire on Device
(device, array)

**Input:** **device []**
Device name. A string of up to eight characters.

**Output:** **array[19]**
Array of inquiry information about *device.*

The elements of *array* are:

**array[1]**
number of character heights available

**array[2]**
number of line types available

**array[3]**
number of line widths available

**array[4]**
number of marker types available

**array[5]**
number of hardware graphic text fonts available

**array[6]**
number of fill patterns available

**array[7]**
number of hatch styles available

**array[8]**
number of predefined colors

**array[9]**
device color capability

0=none
1=yes

**array[10]**
device fill capability

0=none
1=yes

**array[11]**
text alignment capability

0=none
1=yes

**array[12]**
device type

0=output only
1=input only
2=input/output
3=device-independent segment storage

**array[13]**
device type

0=CRT
1=plotter
2=printer
3=reserved
4=metafile output
5=other

**array[14]**
minimum character height as a percentage of the full display surface

**array[15]**
maximum character height as a percentage of the full display surface

**array[16]**
width of the display in rasters or steps (maximum addressable points)

**array[17]**
height of the display in rasters or steps (maximum addressable points)

**array[18]**
width of one raster or step unit in micrometers

**array[19]**
height of one raster or step unit in micrometers

**Related Routines:** Assign Plotting System Input Device, Assign Plotting System Output Device

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine is used only to obtain information about an open device. Inquiry on a closed device results in an error.

If *array* is declared to have fewer than 19 elements, unpredictable results may occur.

If you are using a plotter, array[1] = 0 means continuous character scaling.

# Inquire on Plotting System Error

**Purpose:** Returns the number of the actual error last encountered.

**Format:** Inquire on Plotting System Error

**Input:** None

**Output:** None

**Related Routines:** None

**Status:** Actual error code of the last error encountered.

**Remarks:** Call this routine when the value returned by any other routine is −1.

See Appendix B, “Error Codes”, for a list of error numbers and messages.

# Inquire on Software Version

**Purpose:** Returns the version and level of Plotting System software.

**Format:** Inquire on Software Version
(version, level)

**Input:** None

**Output:** **version**
Software version number.

**level**
Software level number.

**Related Routines:** None

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** None

# Name Attribute Set

**Purpose:** Assigns a character name to an attribute set.

**Format:** Name Attribute Set
(attributes, name)

**Input:** **attributes**
Attribute set identification number. The value of *attributes* must be 〉 0.

**name[]**
Attribute set name.

**Output:** None

**Related Routines:** Set Legend Font, Set Legend Height, Set Existence and Location of Legend

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The value of *name* is used for the legend for bar, line, scatter, schedule, and step charts.

The value of *name* is used for pie slice labels for pie charts.

The height of the labels is set by the Set Legend Height routine. The font of the labels is set by the Set Legend Font routine.

Because only one attribute set is associated with each X/Y data set, the attribute set identification number for an X/Y data set is the same as the data set identification number.

When a pie chart is defined, one attribute set is associated with each slice within the pie chart (1 - 16). The names defined for attribute sets 1 - 16 are mapped one to one with pie slices 1 - 16.

For a schedule chart, the name defined for attribute set 1 is associated with the segment of the task bar that represents the completed portion of the task. The name defined for attribute set 2 is associated with the segment of the task bar that represents the incomplete portion of the task.

The length of *name* has no limit. However, very long names which extend out of the view area will not appear on the chart.

# Open Plotting System

**Purpose:** Initializes Plotting System and sets the plotting environment to the default values.

**Format:** Open Plotting System

**Input:** None

**Output:** None

**Related Routines:** Reset Plotting System to Default

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine must be the first Plotting System routine called.

The viewing environment is set to the Plotting System default values. See Appendix C, “Plotting System Defaults” for a list of the default values.

# Output Currently Defined Chart

**Purpose:** Plots the currently defined chart.

**Format:** Output Currently Defined Chart

**Input:** None

**Output:** None

**Related Routines:** Clear Display Surface, Set Output Primitive Color for Data Set, Set View Area Extents, Stack a Set of Data Sets

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine causes the currently defined chart to be produced by the output device.

If more than one type of chart is defined, the following priority is followed:

- If a pie chart is defined, a pie chart is the only chart output.
- If a schedule chart is defined and no pie chart is defined, the output will be a schedule chart.
- If neither a pie nor a schedule chart is defined, all defined X/Y-axis charts are produced on the same set of axes.

Multiple bar data sets are clustered around independent-axis tick marks unless the Stack a Set of Data Sets routine is called to stack the data sets.

If a chart type is not defined for a data set, a line chart is produced. If no data sets are defined, just the axes are drawn.

The chart is output to the view area defined by the Set View Area Extents routine. On all devices except plotters and printers, the view area is erased before the chart is drawn.

## Output Currently Defined Chart

When this routine is called, the chart is output to the current view area on the current output device. On most CRTs and plotters the chart appears immediately, but on printers the chart must be stored internally in the computer and then output all at once to the printer. Output on printers does not take place until either the Clear Display Surface or Close Plotting System Output Device routine is called. When you want to output a chart on two pages of the printer or plotter, you must call the Clear Display Surface routine after the first call to this routine.

The elements of the chart are drawn in the following order:

- Axes
- Tick marks, grid lines, and axis tick labels (all drawn at the same time)
- Graphic representation (bars, lines, pie slices, and so forth)
- Title
- Subtitle
- Axis titles
- Legend.

All elements except the graphic representation are displayed in the foreground color. You can set the color of the graphic representation using the Set Output Primitive Color for Data Set routine.

# Polyline

**Purpose:** Generates a polyline.

**Format:** Polyline
(coordinate, count, points)

**Input:**

**coordinate**
Location on the display surface from which *points* is referenced.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**count**
Number of points on the polyline.

**points[]**
Array of X and Y coordinates of the points on the polyline. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.* The number of elements in *points* must equal *count* * 2. The elements of the array *points* must be in the sequence (X1, Y1, X2, Y2,. . ., Xn, Yn).

**Output:** None

**Related Routines:** Set Polyline Color, Set Polyline Type

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

This routine generates a set of connected line segments using the coordinates specified by the array *points* as starting and end points for the line segments.

The polyline is output as soon as this routine is called.

Any part of the polyline that is specified to lie outside the view area is clipped.

When you want to draw one line segment, *points* must contain two sets of X and Y coordinates or four elements.

When you want to draw two line segments, *points* must contain three sets of X and Y coordinates or six elements.

When you want to draw the triangle shown below, *points* must contain four sets of X and Y coordinates: first the coordinates of point A, then the coordinates of point B, then the coordinates of point C, then the coordinates of point A again.

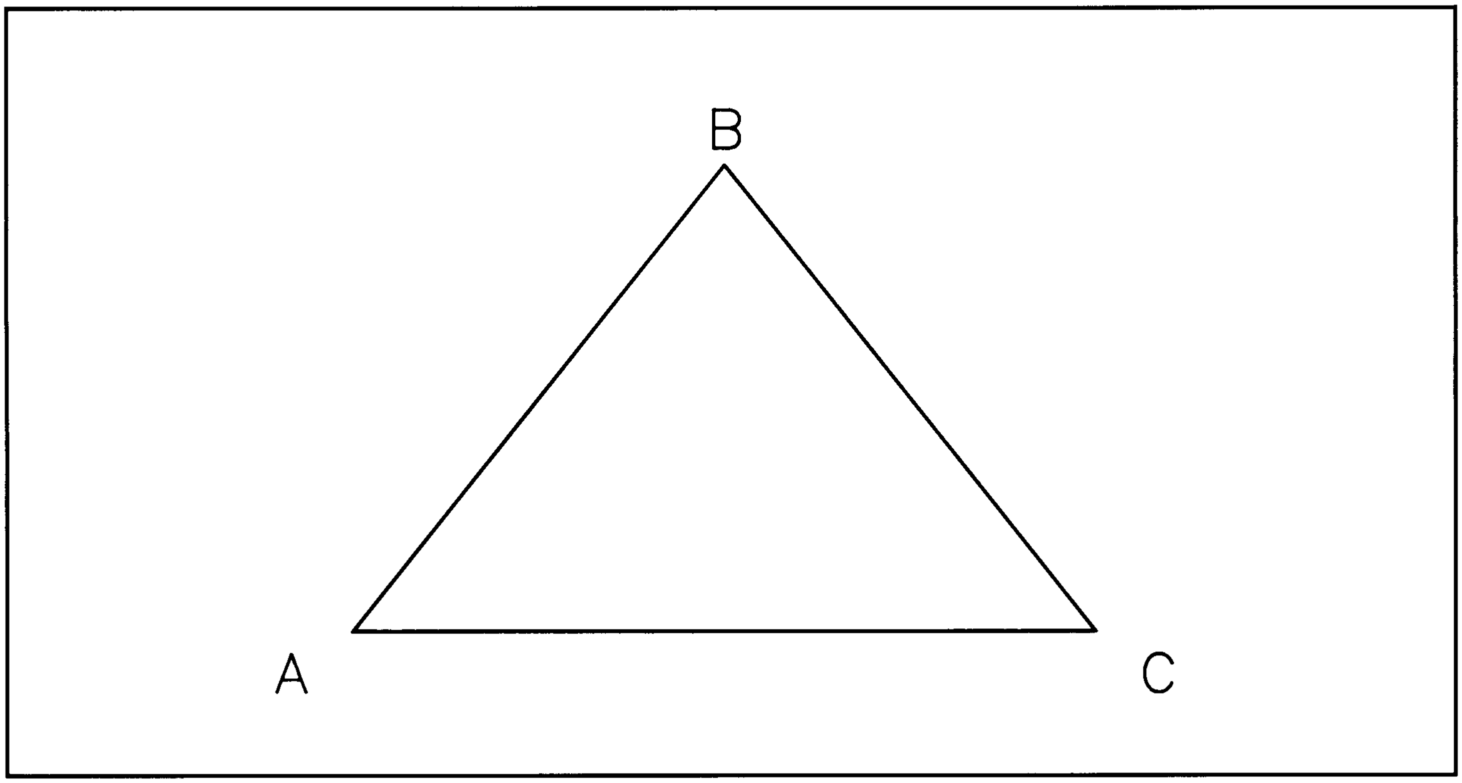


**Figure 3-1. Triangle Drawn Using the Polyline Routine**

# Polymarker

**Purpose:** Generates a set of markers at specified locations.

**Format:** Polymarker
(coordinate, count, points)

**Input:**

**coordinate**
Location on the display surface from which *points* is referenced.

1=lower-left corner of the view area
2=lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**count**
Number of markers.

**points[]**
Array of X and Y coordinates of the markers. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.* The number of elements in points must equal *count* * 2. The elements of the array *points* must be in the sequence (X1, Y1, X2, Y2,. . ., Xn, Yn).

**Output:** None

**Related Routines:** Set Polymarker Color, Set Polymarker Style

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

# Polymarker

This routine generates markers at the coordinates specified by the array *points*.

The markers are output as soon as this routine is called.

Any marker specified to lie outside the view area is clipped.

# Rectangle

**Purpose:** Generates a filled rectangle.

**Format:** Rectangle
(coordinate, corner1, corner2)

**Input:** **coordinate**
Location on the display surface from which *corner1* and *corner2* are referenced.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**corner1[2]**
Array of X and Y coordinates of one corner of the rectangle. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.*

**corner2[2]**
Array of X and Y coordinates of the opposite corner of the rectangle. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.*

**Output:** None

**Related Routines:** Set Fill Color, Set Fill Style

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

Any portion of the rectangle that is specified to lie outside the view area is clipped.

The rectangle is output as soon as this routine is called.

When the output device is a raster display, the area defined by the rectangle parameters is erased with the background color before the rectangle is drawn.

The *corner1* and *corner2* parameters can be any two opposite corners.

# Request Choice

**Purpose:** Activates the choice device and pauses while you input a selection.

**Format:** Request Choice
(input device, choice)

**Input:** **input device**
Initial input choice. The range for *input device* is 1 - 10. Values ⟨ 1 are set to 1 and values ⟩ 10 are set to 10. The default selection is indicated by *choice.*

**Output:** **choice**
Returned choice selection. The range of values returned is 1 - 10.

**Related Routines:** Assign Plotting System Input Device

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** On devices with keyboards but no function keys, the numeric keypad or the digits 1 - 0 represent choices 1 - 10, with the zero digit representing choice 10.

An input device must be open.

# Request Locator

**Purpose:** Enables Plotting System to receive graphic input.

**Format:** Request Locator
(style, coordinate, initial cursor, final cursor, key)

**Input:**

**style**
Type of feedback to be used for the input action.

1 = normal cursor
2 = rubberband line
3 = rubberband box

Values ⟨ 1 are set to 1 and values ⟩ 3 are set to 3.

**coordinate**
Location on the display surface from which *initial cursor* and *final cursor* are referenced.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**initial cursor[2]**
Array of X and Y coordinates of the initial location of the cursor. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.* Values ⟨ 0.0 are set to 0.0 and values ⟩ 100.0 are set to 100.0.

**Output:**

**final cursor[2]**
Array of X and Y coordinates of the final location of the cursor. The values returned are between 0.0 and 100.0.

**key**
Character. Key used to end the graphic input process. For example, to end the graphic input process with a slash "/", the value returned would be ASCII Code 047.

**Related Routines:** Assign Plotting System Input Device, Assign Plotting System Output Device

**Status:**

0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:**

The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

This routine allows you to enter a graphic position.

A rubberband line is a line that stretches from an initial cursor position to a new cursor position. A rubberband line is useful in positioning chart components such as graphic primitives and legends.

A rubberband box indicates an area change. A rubberband box is useful in determining the size of chart components such as legends and rectangular graphics primitives.

The final location of the cursor is returned as X/Y coordinates in *final cursor.* The key used to end the input is returned in *key.*

The graphic cursor is displayed until a key or button is pressed to end the input action. You can move the cursor using the device specified in the Assign Plotting System Input Device routine.

When using an input device for locator input, an output device must also be assigned to echo input device action.

For rubberband lines or boxes, *inital cursor* is the fixed point from which the line or box extends.

For information on the values returned in key, see Appendix D, "ASCII Character and Control Codes."

# Request String

**Purpose:** Enables Plotting System to receive an input string from the keyboard.

**Format:** Request String
(coordinate, echo, echopt, maxlen, string)

**Input:**

**coordinate**
Location on the display surface from which *echopt* is referenced.

1 = lower-left corner of the view area
2 = lower-left corner of the chart area

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**echo**
Echo mode on/off indicator.

0 = input characters are not echoed.
1 = input characters are echoed at *echopt.*

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**echopt[2]**
Array of X and Y coordinates of the position where the characters are echoed if *echo* = 1. The range is 0.0 - 100.0. The coordinates are expressed as percentages of the area specified by *coordinate.* Values ⟨ 0.0 are set to 0.0 and values ⟩ 100.0 are set to 100.0.

**maxlen**
Maximum length of *string.* Values ⟨ 1 are set to 1.

**Output:**

**string []**
The string input from the keyboard.

**Related Routines:** Assign Plotting System Input Device, Assign Plotting System Output Device, Output Currently Defined Chart

**Status:**

0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The coordinates (0.0, 0.0) indicate the lower-left corner of the area specified by *coordinate.* The coordinates (100.0, 100.0) indicate the upper-right corner of the area specified by *coordinate.*

This routine activates the keyboard and accepts any characters up to but not including the string terminator.

The characters read from the keyboard are returned as *string,* up to *maxlen.* The string terminator is not included in *string.*

If echo = 1, an output device must be open.

When the output device is a CRT, this routine can be used as a "wait" function. Calling this routine after calling the Output Currently Defined Chart routine causes the application to pause. You can then examine the output before the application continues. To send the output to the specified output device, press any key.

# Reset All Data Sets

**Purpose:** Deletes all data sets and associated attributes.

**Format:** Reset All Data Sets

**Input:** None

**Output:** None

**Related Routines:** None

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All memory containing data set information is released.

# Reset Plotting System to Default

**Purpose:** Resets Plotting System to start-up conditions.

**Format:** Reset Plotting System to Default

**Input:** None

**Output:** None

**Related Routines:** None

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All data sets are deleted.

See Appendix C, "Plotting System Defaults", for a list of the default values.

# Reset Plotting System Viewing Environment

**Purpose:** Resets viewing environment to full display area and view area.

**Format:** Reset Plotting System Viewing Environment

**Input:** None

**Output:** None

**Related Routines:** None

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Any chart drawn after this routine is called will fill the output device's display surface.

# Set Axis Extents

**Purpose:** Sets the minimum and maximum values of an axis and defines the interval between axis tick labels.

**Format:** Set Axis Extents
(axis, minval, maxval, step)

**Input:**

**axis**
Axis identifier.

1 = independent axis
2 = dependent axis

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**minval**
Beginning value for *axis.*

**maxval**
Ending value for *axis.*

**step**
Increment between ticks. The value of *step* must be ⟩ 0.

**Output:** None

**Related Routines:** None

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The range of the displayed data is determined by *minval* and *maxval.* The increment between axis tick labels and, therefore, the number of axis tick labels on an axis is controlled by *step.*

This routine has no effect on pie, schedule, or text-only charts.

Call this routine to set an axis range and increment different from the default axis range and tick increment. The default axis range includes the minimum and maximum data values. The default tick increment is such that the axis tick labels do not overlap.

# Set Axis Indent

**Purpose:** Establishes whether the data on either axis is to be indented.

**Format:** Set Axis Indent
(independent, dependent)

**Input:** **independent**
Independent axis indent on/off indicator.

0=data not indented (default)
1=data indented

**dependent**
Dependent axis indent on/off indicator.

0=data not indented (default)
1=data indented

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**Output:** None

**Related Routines:** None

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The independent axis for bar and step charts is always indented.

This routine has no effect on pie, schedule, or text-only charts.

Axes should be indented when the graphic representation might overlap the vertical axis or the chart frame. Examples of charts with and without indented axes follow.

If the axes are not indented as shown in Figure 3-2, a chart's origin is the lower-left corner of the chart area. The first horizontal axis label is centered below the vertical axis, while the first vertical axis label is centered adjacent to the left end of the horizontal axis.

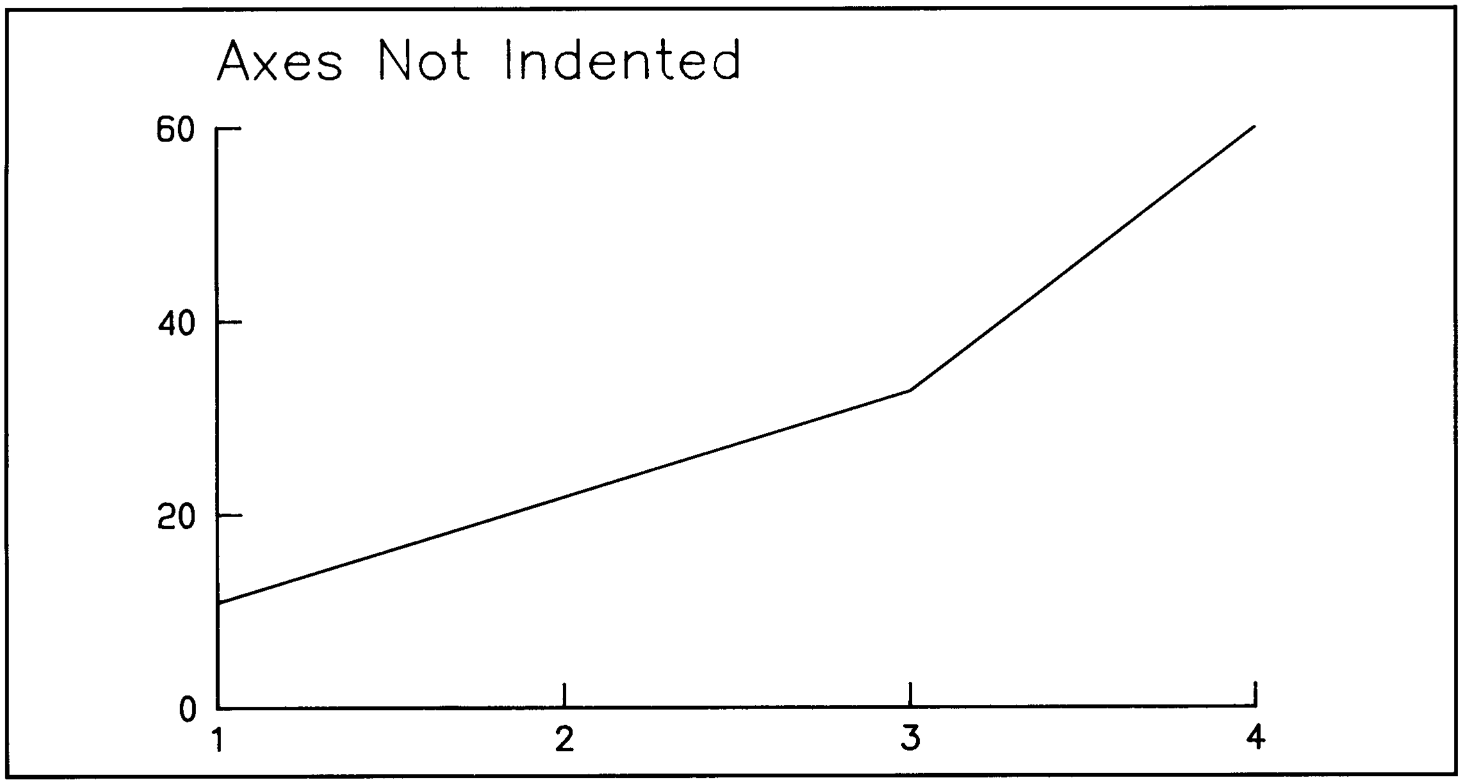


**Figure 3-2. Axes Not Indented**

If an axis is indented, as shown in Figure 3-3, the axis labels are indented by placing the first label to the right or above the origin of the chart area. The data is indented one-half a standard tick interval.

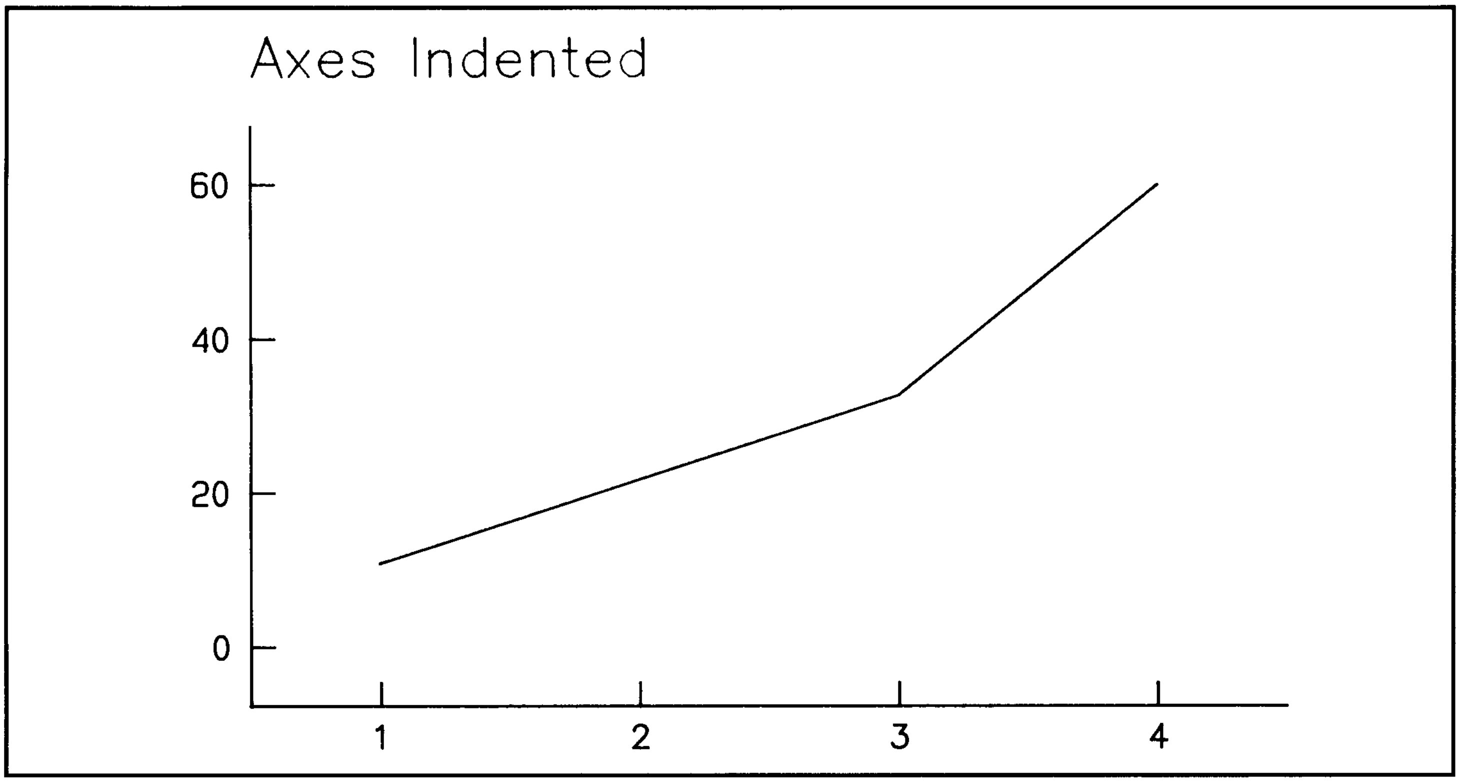


**Figure 3-3. Axes Indented**

# Set Axis Labels

**Purpose:** Defines a set of labels to annotate the tick marks on the independent or dependent axis.

**Format:** Set Axis Labels
(axis, string)

**Input:** **axis**
Axis to be labeled.

1 = independent axis
2 = dependent axis

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**string[]**
Axis tick labels to be placed at each tick mark.

**Output:** None

**Related Routines:** Set Axis Tick Label Type, Set Axis Title Font, Set Axis Title Height, Set Axis Type, Set Foreground Color

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** A single-character delimiter is used to signal the beginning of the label string and to separate individual labels. Plotting System uses the first character of the label string (after the quotation mark) as the delimiter.

Subsequent labels are separated by one delimiter. A blank space between two delimiters indicates no label for the corresponding tick mark. The string is terminated with two delimiters. Strings not properly terminated cause unpredictable results.

An example of a label string follows:

"/one/two/ /four//"

Standard numeric axis tick labels are used if this routine is not called.

Call this routine if you want to annotate the tick marks on either axis with labels other than numeric axis labels, such as months, days, or departments. The labels can be any length, but long labels may overlap if they are specified for a short axis.

Axis tick labels are the same height and font as the axis title and display in the foreground color.

For area, bar, line, and scatter charts, the labels defined by *string* are used as the axis tick labels for the specified axis, one label for each axis tick mark. For schedule charts, the labels defined by *string* are used as the task names.

This routine has no effect on pie or text-only charts.

If labels are defined for an axis, the values specified for that axis by the Set Axis Tick Label Type and Set Axis Type routines are ignored.

# Set Axis Tick Label Type

**Purpose:** Defines type and precision of axis tick labels.

**Format:** Set Axis Tick Label Type
(indep_type, indep_prec, dep_type, dep_prec)

**Input:** **indep_type**
Independent axis tick label type. Label type indicates the type of numeric data.

1 = value (default)
2 = percent (%)
3 = dollar ($)

Values 〈 1 are set to 1 and values 〉 3 are set to 3.

**indep_prec**
Independent axis tick label precision type. Precision type indicates the format of the numeric data.

1 = integer (default)
2 = real (with two decimal positions)
3 = real (with three decimal positions)
4 = real (with four decimal positions)
5 = real (with five decimal positions)

Values 〈 1 are set to 1 and values 〉 5 are set to 5.

**dep_type**
Dependent axis tick label type. Label type indicates the type of numeric data.

1 = value (default)
2 = percent (%)
3 = dollar ($)

Values 〈 1 are set to 1 and values 〉 3 are set to 3.

**dep_prec**
Dependent axis tick label precision type. Precision type indicates the format of the numeric data.

1=integer (default)
2=real (with two decimal positions)
3=real (with three decimal positions)
4=real (with four decimal positions)
5=real (with five decimal positions)

Values ⟨ 1 are set to 1 and values ⟩ 5 are set to 5.

**Output:** None

**Related Routines:** Set Axis Title Font, Set Axis Title Height, Set Foreground Color

**Status:** 0 if no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Percent and dollar label types generate the respective character with the label. The dollar sign is placed before the numeric value. The percent sign is placed after the numeric value.

Pie chart labels are specified using only the dependent axis parameters. The independent axis parameters are ignored.

Axis tick labels are the same height and font as the axis title. Axis tick labels appear in the foreground color.

The default for axis label precision is integer (1) in most instances. However, axis labels will be output as real numbers with 5-digit accuracy if the difference between the maximum axis value and minimum axis value is less than 0.10 and greater than 0.00001. Also, the default for axis label precision is integer (1) unless the maximum axis value is greater than 999,999,999 or the minimum axis value is less than -999,999,999. In this case, axis labels will be output as real numbers in scientific notation with four places for accuracy.

If a logarithmic axis is specified via Set Axis Type, the axis label precision is integer (1), unless the difference between the maximum and minimum axis values is less than 1.0. In this case, the default will be real (2) with two places of accuracy. Logarithmic axes with minimum axis values less than 1.0 also feature real (2) precision. Axis tick labels are displayed in scientific notation with four places of accuracy. These exceptions hold even if the user has set integer precision, because real precision is needed to differentiate between axis label ticks when data falls into the indicated ranges.

# Set Axis Title Font

**Purpose:** Sets the character font for the axis title.

**Format:** Set Axis Title Font
(axis, font)

**Input:**

**axis**
Axis to be titled.

1 = independent
2 = dependent

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**font**
Character font.

1 = Hardware font (default)
101 = Simplex Roman software font
102 = Complex Roman software font
103 = Complex Italic software font
104 = Duplex Roman software font
105 = Triplex Roman software font
106 = Triplex Italic software font

Values ⟨ 1 are set to 1. Values between 1 and 101 are hardware fonts. Values between the maximum number of fonts supported and 101 are set to 1. Values ⟩ 106 are set to 1.

**Output:** None

**Related Routines:** Name Attribute Set, Set Axis Labels, Set Axis Title Height

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The font specified is used for the axis title and axis tick labels.

For pie chart, the specified dependent axis title font is used for the pie slice labels. Pie slice labels are assigned with the Name Attribute Set routine.

If a software font (101-106) is to be used, the corresponding font table must be in your current directory or a path to its location must be set. See “Setting Environmental Parameters” in Appendix A. See Figure 1-5 in Chapter 1, “Using Plotting System”, for an example of each software font.

# Set Axis Title Height

**Purpose:** Sets the character height of the axis title.

**Format:** Set Axis Title Height
(axis, height)

**Input:**

**axis**
Axis to be titled.

1 = independent
2 = dependent

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**height**
Character height of the axis title as a percentage of the view area. The range for height is 1 - 20. Values ⟨ 1 are set to 1 and values ⟩ 20 are set to 20. The default value for height is 5.

**Output:** None

**Related Routines:** Name Attribute Set, Set Axis Labels, Set Axis Title Font

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Because the graphic and text output display proportionally within the view area, the size of the plot is affected by the size of the title, subtitle, and axis title. The size of the plot is reduced so the axis titles do not overlap other titles or the plot.

Axis tick labels display at the same height and in the same font as the axis titles.

For pie charts, this routine sets the height for pie slice labels. Pie slice labels are assigned by the Name Attribute Set routine.

Consider the spacing and size of the axis tick labels when you specify the font and height of the axis titles. If a large height is chosen for the axis title, the axis tick labels may overlap or be too closely spaced.

# Set Axis Type

**Purpose:** Assigns linear or logarithmic axis type.

**Format:** Set Axis Type
(independent, dependent)

**Input:**

**independent**
Independent axis type.

1 = linear (default)
2 = logarithmic (base 10)

**dependent**
Dependent axis type.

1 = linear (default)
2 = logarithmic (base 10)

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**Output:** None

**Related Routines:** None

**Status:**

0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Call this routine to plot data along a logarithmic axis.

Logarithmic axes are useful when comparing data that varies greatly.

The physical distance between tick marks on a linear axis represents equal amounts.

The physical distance between tick marks on a logarithmic axis does not represent equal amounts. If the first tick mark is labeled 10 and the second 100, the physical distance between the origin and 10 and 10 and 100 is the same.

This routine has no effect on pie, schedule, or text-only charts.

The Plotting System's logarithmic axis is base 10; thus, the first tick mark might be labeled 10 and the second labeled 100 with an equal distance lying between 10 and the origin, and 10 and 100.

If a negative value is provided for an axis extent and a log (2) axis type is chosen, the log of the negative value will approach 0 on the output axis.

When a logarithmic axis is specified, the axis label precision is integer (1), unless the difference between the maximum and minimum axis values is less than 1.0. In this case, the default will be real (2) with two places of accuracy. The axis label precision for logarithmic axes with minimum axis values less than 1.0 is also real (2). Axis labels will be output in scientific notation with four places of accuracy. These exceptions hold even if the user has set integer precision because real precision is needed in the indicated data ranges to differentiate between axis label ticks.

# Set Background Color

**Purpose:** Sets the background color index.

**Format:** Set Background Color
(color)

**Input:** **color**
Background color index. The range for *color* depends on the output device. The default value for *color* is 0. Values ⟨ 0 are set to 0.

**Output:** None

**Related Routines:** Set Color Representation

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Since the number of available colors depends on the output device, this routine produces device-dependent results.

Background color is useful for CRT and film recorder devices where a backdrop is required. Many printers and plotters cannot support a change of background color.

The Set Color Representation routine can be used to map indexes to new colors.

# Set Baseline

**Purpose:** Specifies the location of the baseline.

**Format:** Set Baseline
(value)

**Input:** **value**
Data value for the baseline. The default for *value* is 0.0.

**Output:** None

**Related Routines:** Stack a Set of Data Sets

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The baseline is drawn for all charts. If the baseline value falls outside the data value range, the axis range is extended to include the baseline.

This routine has no effect on pie, schedule, or text-only charts.

When a baseline is specified, stacked data is plotted relative to this line. Nonstacked data is plotted relative to the independent axis according to absolute values. Important data deviations such as profit/loss differences are easily recognized when plotted on opposite sides of a baseline.

Here are two examples of charts with different baselines. Figure 3-4 shows a chart with the default baseline, which is on the independent axis. Figure 3-5 shows a chart whose baseline is set at position 25.0 on the dependent axis.

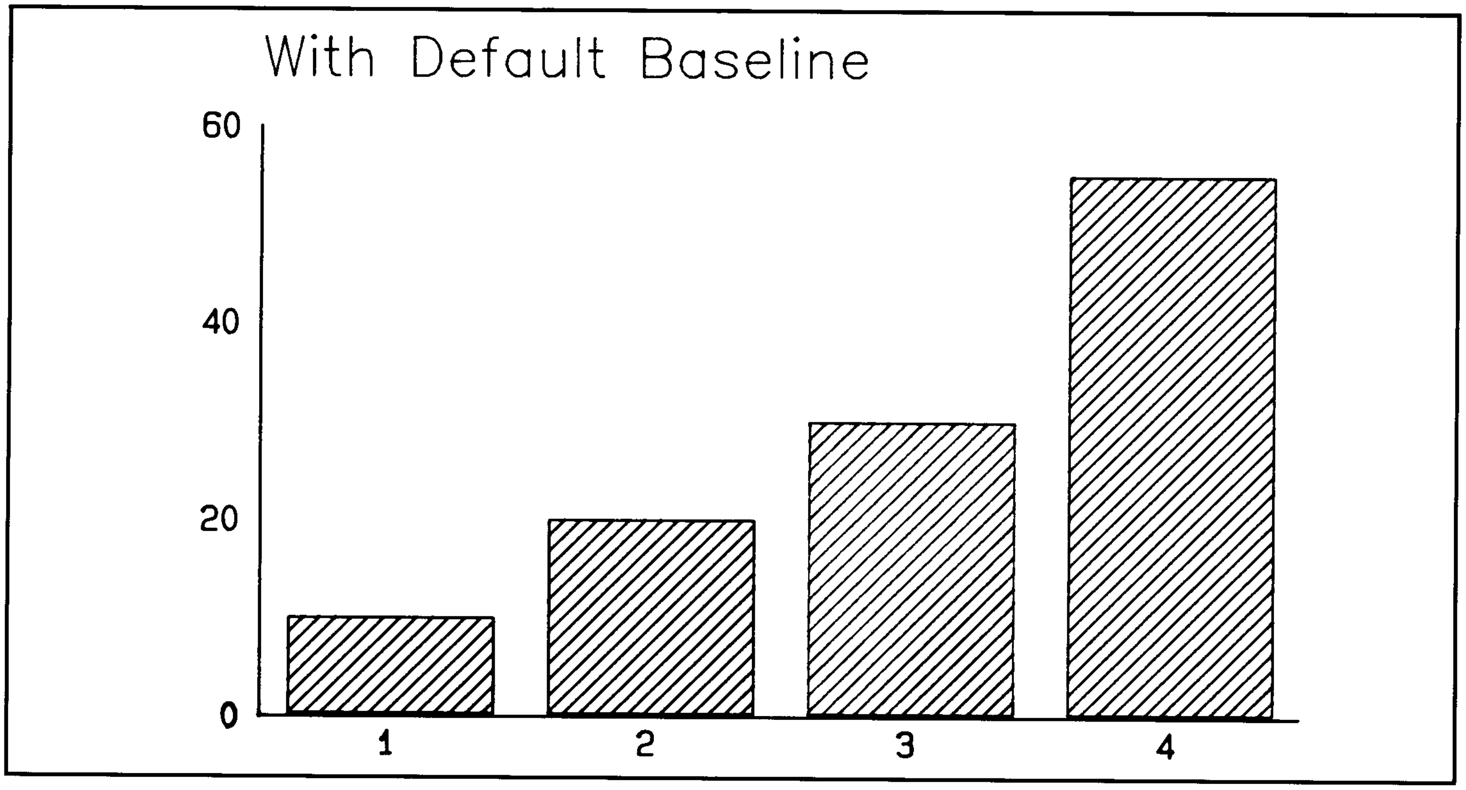


**Figure 3-4. Default Baseline**

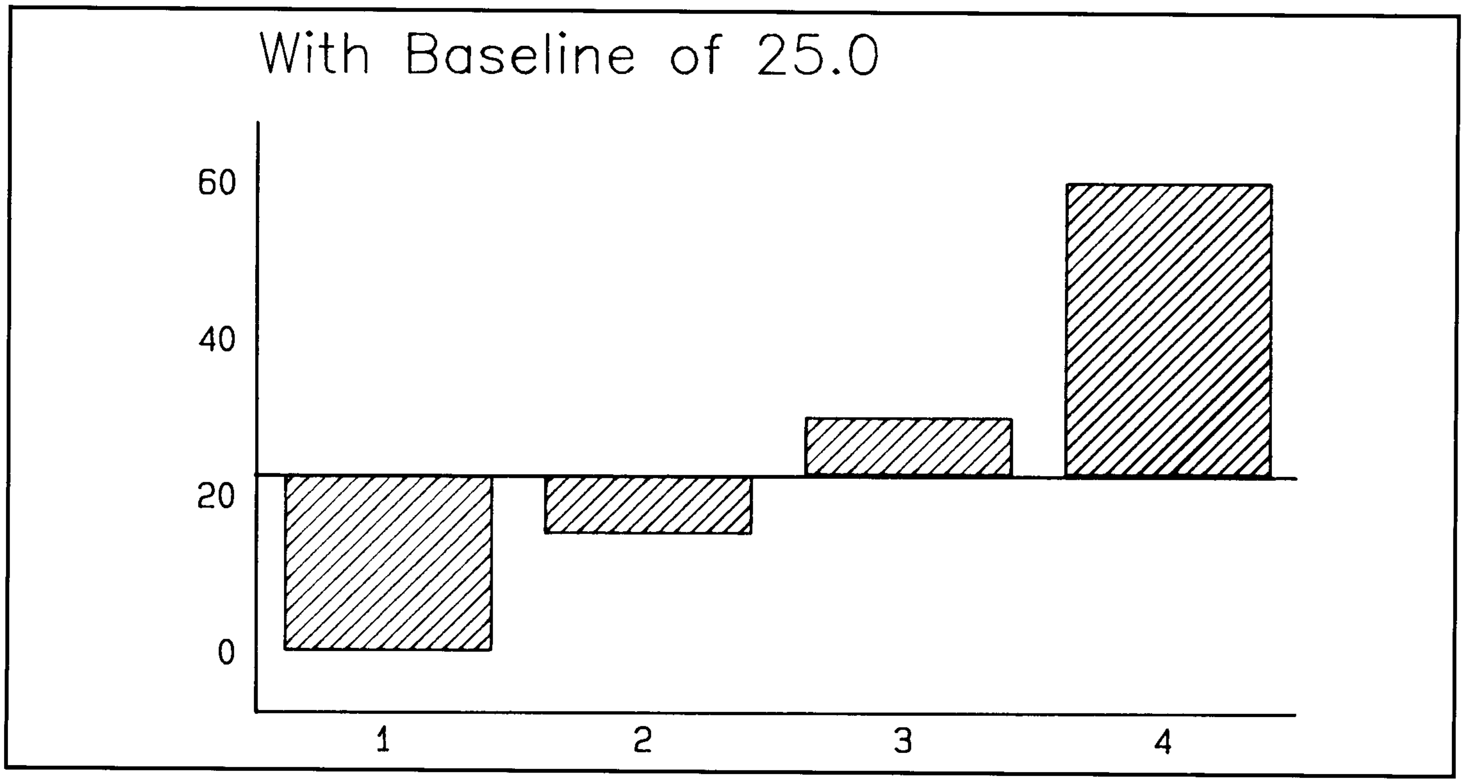


**Figure 3-5. Baseline Set at 25.0**

## Set Color Representation

**Purpose:** Sets the color index to the desired color.

**Format:** Set Color Representation
(index, red, green, blue)

**Input:**

**index**
Color number selected from the color table. The range is 0 to the maximum supported by the device.

| Color Table | | | | |
|---|---|---|---|---|
| **Index** | **Color** | **Red** | **Green** | **Blue** |
| 0 | Black | 0% | 0% | 0% |
| 1 | White | 100% | 100% | 100% |
| 2 | Red | 100% | 0% | 0% |
| 3 | Green | 0% | 100% | 0% |
| 4 | Blue | 0% | 0% | 100% |
| 5 | Yellow | 100% | 100% | 0% |
| 6 | Cyan | 0% | 100% | 100% |
| 7 | Magenta | 100% | 0% | 100% |

**red**
Percentage of red intensity requested. The range is 0.0 - 100.0.

**green**
Percentage of green intensity requested. The range is 0.0 - 100.0.

**blue**
Percentage of blue intensity requested. The range is 0.0 - 100.0.

**Output:** None

**Related Routines:** Set Background Color, Set Fill Color, Set Foreground Color, Set Notation Color, Set Output Primitive Color for Data Set, Set Polyline Color, Set Polymarker Color

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** You can change a color in the table by changing the percentages of red, green, and blue intensities associated with *index.*

This routine is used to map indexes to new colors.

At least two color indexes are provided: foreground and background. If the desired index is outside of the device's capabilities, then the closest device index is set. If a color intensity of less than 0 is requested, it is mapped to 0. If a color intensity greater than 100.0 is requested, it is mapped to 100.0.

To change the appearance of a color, you must select the desired levels of the three color components (red, green, and blue) that make up the index. This can be used to create non-default colors such as brown or orange. The new color will be visible only on devices that support color definition.

# Set Date

**Purpose:** Sets the current date for schedule charts.

**Format:** Set Date
(date)

**Input:** **date[8]**
Character array. Current date. The format for date is "MM/DD/YY" where:

MM=two-digit month (1-12)
DD=two-digit day (1-31)
YY=two-digit year (00-99)
/=required delimiter.

**Output:** None

**Related Routines:** Define Data Set Chart Type to Be Schedule, Define Schedule Chart Data Set

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine places a vertical line on a schedule chart at the specified date.

This routine is used only for schedule charts.

# Set Display Surface Size

**Purpose:** Specifies the height and width of the display surface.

**Format:** Set Display Surface Size
(width, height)

**Input:**

**width**
Display surface width.

**height**
Display surface height.

**Output:** None

**Related Routines:** Set Display Surface Units

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The values of *height* and *width* indicate the display surface size in inches or millimeters as specified in the Set Display Surface Units routine.

The range of valid values for *width* and *height* depends on the display device used. The default values are the full display surface of the current output device.

The lower-left corner of the display surface defined by *height* and *width* is located at the lower-left corner of the full display surface.

If the size specified exceeds the actual physical size of the device, the images created are scaled down to a size that fits on the device and preserves the requested shape of the chart output.

For example, if the actual device dimensions are 8 by 12 inches and the requested size is 10 by 10 inches, then the display surface size is set to 8 by 8 inches to preserve the one-to-one width-to-height ratio. This is shown in Figure 3-6.

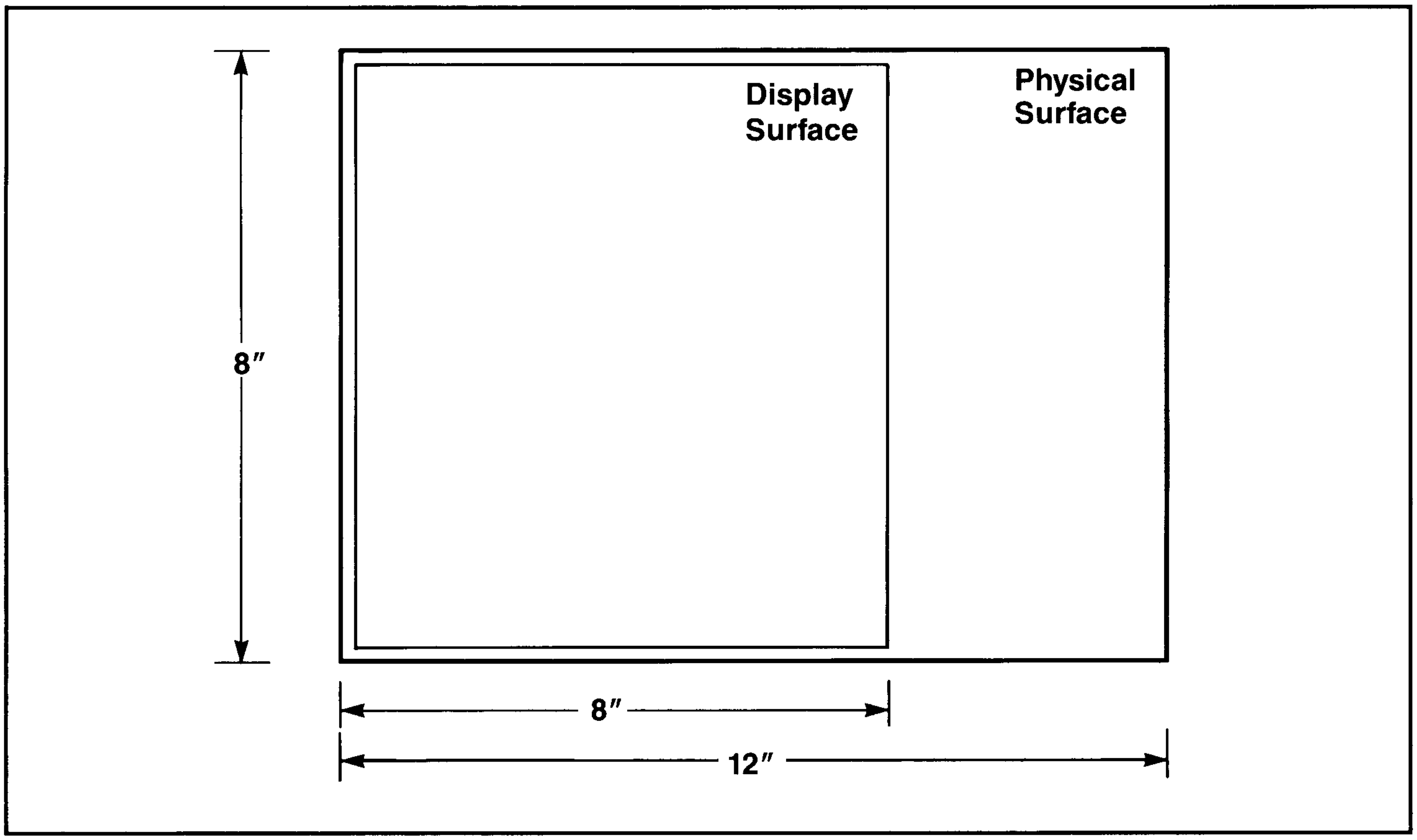


**Figure 3-6. Requested Size Too Large**

## Set Display Surface Units

**Purpose:** Sets display surface units.

**Format:** Set Display Surface Units
(units)

**Input:** **units**
Unit types.

1 = inches (default)
2 = millimeters

Values ⟨ 1 are set to 1 and values ⟩ 2 are set to 2.

**Output:** None

**Related Routines:** Set Display Surface Size

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine specifies the type of units (inches or millimeters) to be used to define the size of the display surface in the Set Display Surface Size routine.

# Set Existence and Location of Legend

**Purpose:** Indicates a legend's visibility and location within a chart area.

**Format:** Set Existence and Location of Legend
(visibility switch, visbox, x, y)

**Input:** **visibility switch**
Visibility on/off switch.

0=legend not visible (default)
1=legend visible

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**visbox**
Legend box frame on/off switch.

0=frame not visible (default)
1=frame visible

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**x**
X coordinate of the legend location as a percentage (0.0 - 100.0) of the width of the chart area. Values ⟨ 0.0 are set to 0.0 and values ⟩ 100.0 are set to 100.0.

**y**
Y coordinate of the legend location as a percentage (0.0 - 100.0) of the length of the chart area. Values ⟨ 0.0 are set to 0.0 and values ⟩ 100.0 are set to 100.0.

**Output:** None

**Related Routines:** Set Foreground Color, Set Legend Alignment, Set Legend Font, Set Legend Height

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** If this routine is not called, no legend is displayed.

The size of the legend box is determined by the number of legend strings, the length of the legend strings, and the character height of the legend text.

The legend appears in the foreground color.

This routine has no effect on pie or text-only charts.

The basepoint of the legend is defined within the chart area coordinate system. However, the entire legend box can lie outside the chart area. For example, the basepoint can be set as (100.0, 100.0). The alignment can be set at horizontal=right and vertical=bottom by the Set Legend Alignment routine.

In Figure 3-7 for example, the lower line of the legend will intersect the chart area. If the entire legend box fits within the view area, the legend appears in the output. If the entire legend will not fit within the view area, no legend is output.

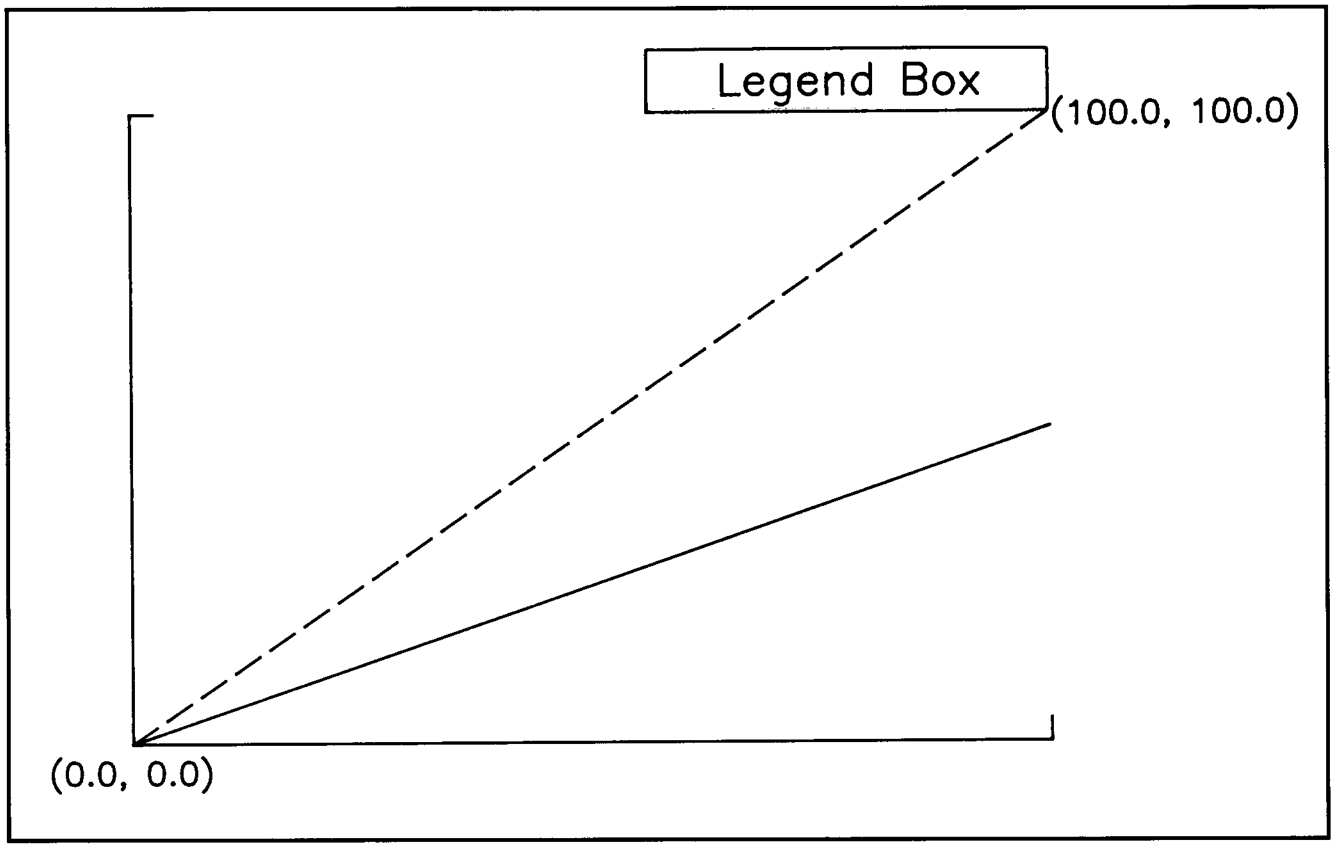


**Figure 3-7. Legend Box Outside Chart Area**

# Set Existence of Axis

**Purpose:** Assigns visibility of axes and axis labels.

**Format:** Set Existence of Axis
(i1, i2, d1, d2)

**Input:**

**i1**
Independent axis on/off indicator.

0=axis not visible
1=axis visible (default)

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**i2**
Independent axis tick label on/off indicator.

0=labels not visible
1=labels visible (default)

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**d1**
Dependent axis on/off indicator.

0=axis not visible
1=axis visible (default)

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**d2**
Dependent axis tick label on/off indicator.

0=labels not visible
1=labels visible (default)

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**Output:** None

**Related Routines:** Set Axis Labels, Set Axis Tick Label Type

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** When an axis is turned off, the axis line is not displayed. When axis tick labels are turned off, the labels and tick marks for that axis are not displayed.

This routine has no effect on pie, schedule, or text-only charts.

# Set Existence of Chart Area Frame

**Purpose:** Assigns the visibility and thickness of the frame surrounding the chart area.

**Format:** Set Existence of Chart Area Frame
(onoff, width)

**Input:** **onoff**
Chart area frame on/off indicator.

0=frame not visible (default)
1=frame visible

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**width**
Width of the frame. The value of *width* is specified in units of one-tenth of a percent of the horizontal dimension of the view area. For example, *width* = 15 defines a line with a width of 1.5% of the horizontal view area. The range for *width* is 1 - 100. Values ⟨ 1 are set to 1 and values ⟩ 100 are set to 100.

**Output:** None

**Related Routines:** Set Existence of Axis, Set Foreground Color

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The chart area of X/Y-axis charts is defined by the axes. The extents of the frame are determined by the length of the axes. If the axes are not displayed, the frame is drawn completely around the chart area. If only one axis is displayed, the frame is drawn on the remaining three sides.

The frame width specified will be supported to the best ability of the device. Regardless of the frame width specified, the chart axis will be drawn as a single line.

If you want thick axes to correspond with a thick frame, call the Set Existence of Axis routine and turn off the axes. When the Set Existence of Chart Area Frame routine is called and the frame is turned on, the frame will complete all four sides of the chart area frame in the desired width.

The chart area frame appears in the foreground color.

This routine has no effect on pie charts.

# Set Existence of Grid Lines

**Purpose:** Assigns the visibility and type of grid lines for the independent and dependent axes.

**Format:** Set Existence of Grid Lines
(igrid, istyle, dgrid, dstyle)

**Input:**

**igrid**
Independent axis grid on/off indicator.

0=axis grid not visible (default)
1=axis grid visible

Values 〈 0 are set to 0 and values 〉 1 are set to 1.

**istyle**
Independent axis grid style indicator.

1=grid lines solid
2=grid lines dotted

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**dgrid**
Dependent axis grid on/off indicator.

0=axis grid not visible (default)
1=axis grid visible

Values 〈 0 are set to 0 and values 〉 1 are set to 1.

**dstyle**
Dependent axis grid style indicator.

1=grid lines solid
2=grid lines dotted

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**Output:** None

**Related Routines:** Set Foreground Color

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Axis grid lines are drawn at right angles to the specified axis and are placed at each tick mark.

Grid lines appear in the foreground color.

This routine has no effect on pie or text-only charts.

# Set Existence of View Area Frame

**Purpose:** Assigns the visibility and thickness of the frame surrounding the view area.

**Format:** Set Existence of View Area Frame
(onoff, width)

**Input:** **onoff**
View area frame on/off indicator.

0=frame not visible (default)
1=frame visible

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**width**
Width of the frame. The value of *width* is specified in units of one-tenth of a percent of the horizontal dimension of the view area. For example, *width* = 15 defines a line with a width of 1.5% of the horizontal view area.

The range for *width* is 1 - 100. Values ⟨ 1 are set to 1 and values ⟩ 100 are set to 100.

**Output:** None

**Related Routines:** Set Foreground Color

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The view area frame surrounds the entire view area and is drawn with a solid line in the foreground color.

The frame width specified will be supported to the best ability of the device. This will not affect the chart axis which will be drawn as a single line.

If you want thick axes to correspond with a thick frame, call the Set Existence of Axis routine and turn off the axes. When the Set Existence of View Area Frame routine is called and the frame is turned on, the frame will complete all four sides of the view area frame in the desired width.

# Set Explode of Pie Slice

**Purpose:** Moves a pie slice out from the center of the pie by 20 percent of the radius.

**Format:** Set Explode of Pie Slice
(slice, flag)

**Input:** **slice**
Number of the slice to be exploded. The range of values for *slice* is 1 - 16. Values 〈 1 are set to 1 and values 〉 16 are set to 16.

**flag**
On/off switch indicating if the slice is to be exploded.

0=slice not exploded (default)
1=slice exploded

Values 〈 0 are set to 0 and values 〉 1 are set to 1.

**Output:** None

**Related Routines:** Define Data Set Chart Type to Be Pie

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The *slice* number corresponds to the Nth value in the sequence data set defining the pie chart.

This routine affects pie charts only.

# Set Fill Color

**Purpose:** Assigns a fill color to an output primitive.

**Format:** Set Fill Color
(color)

**Input:** **color**
Color index. The value for *color* must be ≥ 0. Values 〈 0 are set to 1. The default value for *color* is 1.

**Output:** None

**Related Routines:** Arc, Arrow, Circle, Rectangle, Set Color Representation, Set Fill Style

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Starting with color index 0, the unique colors of the open device are assigned successive numbers until no more unique colors exist. For requested color indexes greater than the device maximum, the maximum device value is used. In the case of a monochrome device, the background color is 0. The foreground color is any value greater than 0.

All arcs, arrows, circles, and rectangles defined after this routine is called are filled according to the specified fill color. If a hollow fill style is specified by the Set Fill Style routine, the outline appears in the specified color.

The Set Color Representation routine can be used to map indexes to new colors. See Appendix C, "Graphics Drivers", in the *IBM RT PC Graphics Development Toolkit* for color index information on specific devices.

# Set Fill Style

**Purpose:** Assigns a fill style to an output primitive.

**Format:** Set Fill Style
(style)

**Input:** **style**
Fill style index.

1=hollow (default)
2=solid
3=narrow spaced +45 degree lines
4=medium spaced +45 degree lines
5=wide spaced +45 degree lines
6=narrow spaced +45 and –45 degree lines
7=medium spaced +45 and –45 degree lines
8=wide spaced +45 and –45 degree lines
⟩8=device-dependent

Values ⟨ 1 and ⟩ the maximum supported by the device are set to 1.

**Output:** None

**Related Routines:** Arc, Arrow, Circle, Rectangle

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All arcs, arrows, circles, and rectangles defined after this routine is called are filled according to the specified fill style.

# Set Foreground Color

**Purpose:** Sets the foreground color index.

**Format:** Set Foreground Color
(color)

**Input:** **color**
Foreground color index. The range of values for *color* depends on the output device. The default value for *color* is 1. Values 〈 0 are set to 1.

**Output:** None

**Related Routines:** Set Color Representation

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The number of colors available depends on the output device. Chart elements controlled by this parameter are:

- Axes
- Axis labels, tick marks, and titles
- Chart area frame
- Filled area outline for data sets or pie slices
- Grid lines
- Legend
- Pie slice labels
- Subtitle
- Title
- View area frame.

The Set Color Representation routine can be used to map indexes to new colors.

# Set Legend Alignment

**Purpose:** Specifies alignment of the legend.

**Format:** Set Legend Alignment
(horiz, vert)

**Input:** **horiz**
Horizontal alignment indicator.

1 = left (default)
2 = center
3 = right

Values ⟨ 1 are set to 1 and values ⟩ 3 are set to 3.

**vert**
Vertical alignment indicator.

1 = bottom (default)
2 = center
3 = top

Values ⟨ 1 are set to 1 and values ⟩ 3 are set to 3.

**Output:** None

**Related Routines:** Set Existence and Location of Legend, Set Legend Font, Set Legend Height

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The legend coordinates specified by the Set Existence and Location of Legend routine are used as the base point for aligning the legend. Using horizontal and vertical alignment specified by this routine, the legend box is placed relative to the base point. For example, *horiz* = 3 and *vert* = 1 indicates that the lower-right corner of the legend is placed at the base point.

This routine has no effect on pie charts.

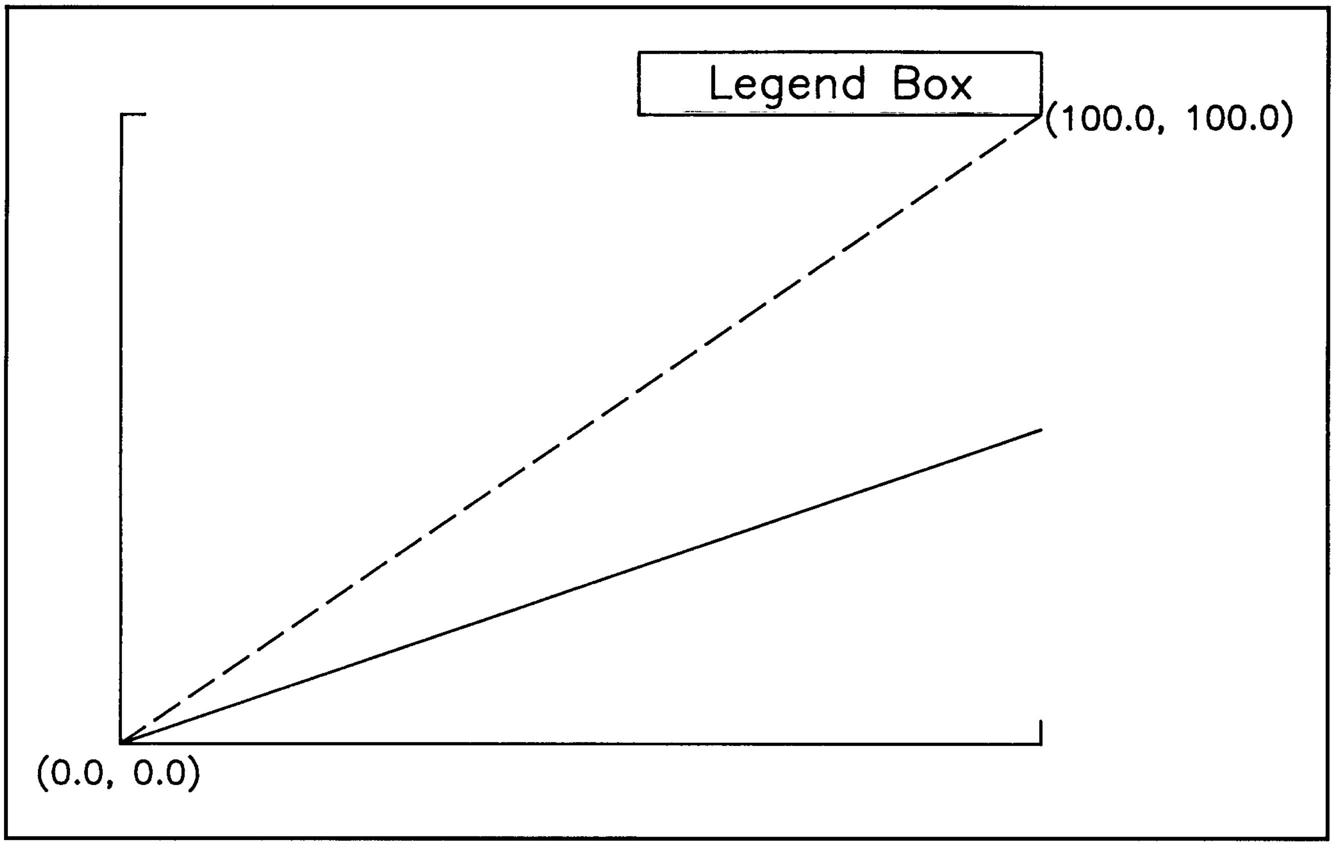


**Figure 3-8. Legend Placed Relative to Base Point**

# Set Legend Font

**Purpose:** Sets the character font for the legend text strings.

**Format:** Set Legend Font
(font)

**Input:** **font**
Character font.

1 = Hardware font (default)
101 = Simplex Roman software font
102 = Complex Roman software font
103 = Complex Italic software font
104 = Duplex Roman software font
105 = Triplex Roman software font
106 = Triplex Italic software font

Values less than 1 are set to 1. Values between 1 and 101 are hardware fonts. Values between the maximum number of fonts supported and 101 are set to 1. Values greater than 106 are set to 1.

**Output:** None

**Related Routines:** Set Existence and Location of Legend, Set Legend Alignment, Set Legend Height

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** This routine has no effect on pie or text-only charts.

If a software font (101-106) is to be used, the corresponding font table must be in your current directory or a path to its location must be set. See "Setting Environmental Parameters" in Appendix A.

See Figure 1-5 in Chapter 1, "Using Plotting System", for an example of each software font.

# Set Legend Height

**Purpose:** Sets the character height of the legend text strings.

**Format:** Set Legend Height
(height)

**Input:** **height**
Character height of the legend text strings as a percentage of the view area. The range for *height* is 1 - 20. Values ⟨ 1 are set to 1 and values ⟩ 20 are set to 20. The default value for *height* is 5.

**Output:** None

**Related Routines:** Name Attribute Set, Set Existence and Location of Legend, Set Legend Alignment, Set Legend Font

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** In contrast to titles, subtitles, and axis titles, the size of the chart area is not modified to accommodate the legend. However, the length and height of legend text strings affect the size of the legend area.

This routine has no effect on pie or text-only charts.

Text strings are assigned using the Name Attribute Set routine.

# Set Notation Alignment

**Purpose:** Specifies alignment of the notation string.

**Format:** Set Notation Alignment
(horiz, vert)

**Input:** **horiz**
Horizontal alignment indicator.

1 = left (default)
2 = center
3 = right

Values 〈 1 are set to 1 and values 〉 3 are set to 3.

**vert**
Vertical alignment indicator.

1 = bottom (default)
2 = center
3 = top

Values 〈 1 are set to 1 and values 〉 3 are set to 3.

**Output:** None

**Related Routines:** Define Notation String and Location, Set Notation Color, Set Notation Font, Set Notation Height

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The notation coordinates specified by the Define Notation String and Location routine serve as a base point. By specifying *horiz* and *vert,* the notation string is placed relative to the base point. For example, if *horiz* = 3 and *vert* = 1 when this routine is called, the lower-right corner of all subsequent notation strings is placed at the base point. See Figure 3-9.

Text can be centered by specifying *horiz* =2 and *vert* =2.

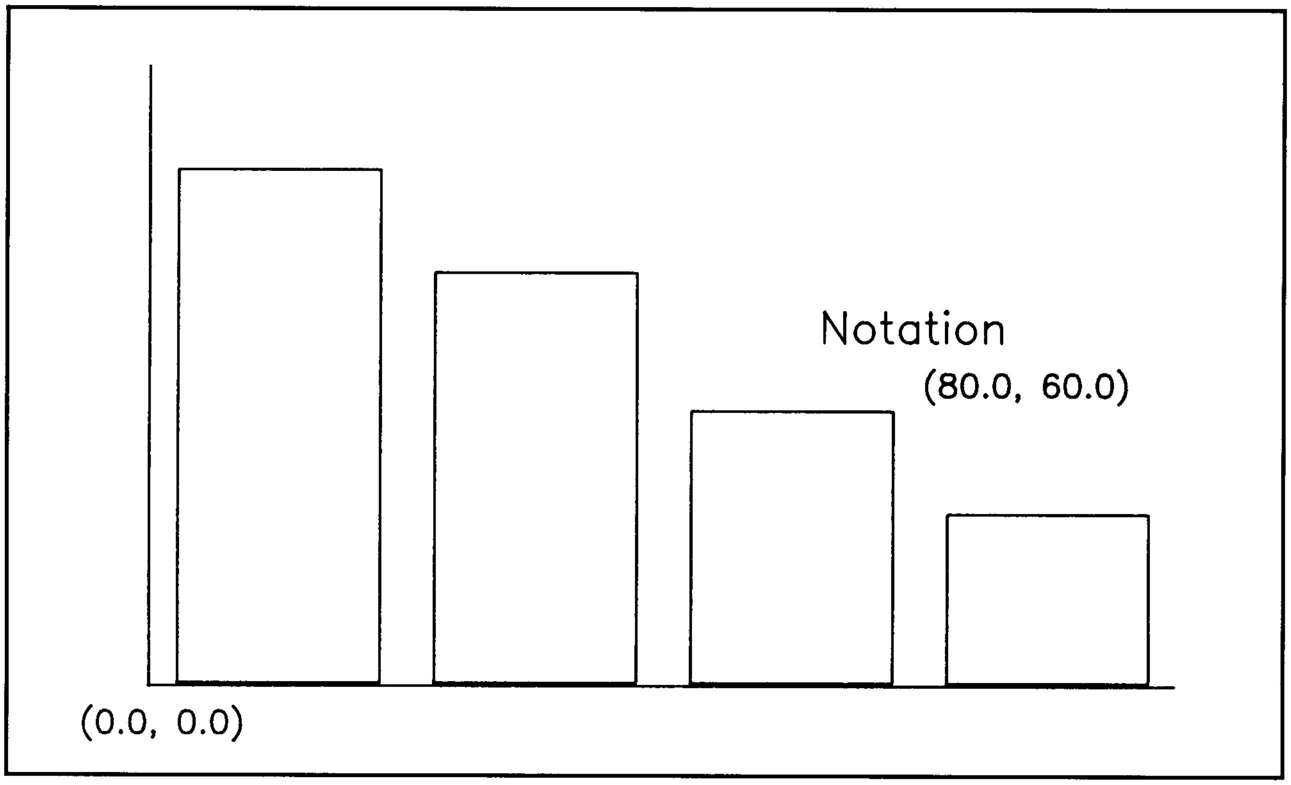


**Figure 3-9. Notation Alignment**

# Set Notation Color

**Purpose:** Assigns a color to a notation string.

**Format:** Set Notation Color
(color)

**Input:** **color**
Color index. The value of *color* must be ≥ 0. Values 〈 0 are set to 1. The default value for *color* is 1.

**Output:** None

**Related Routines:** Define Notation String and Location, Set Color Representation, Set Notation Alignment, Set Notation Font, Set Notation Height

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All notation strings defined after this routine is called are plotted in the specified color.

Starting with color index 0, the unique colors of the open device are assigned successive numbers until no more unique colors exist.

For requested color indexes greater than the device maximum, the maximum device value is used.

In the case of a monochrome device, the background color is color 0. The foreground color is used for all values 〉 0.

The Set Color Representation routine can be used to map indexes to new colors.

# Set Notation Font

**Purpose:** Sets the character font for a notation string.

**Format:** Set Notation Font
(font)

**Input:** **font**
Character font.

1 = Hardware font (default)
101 = Simplex Roman software font
102 = Complex Roman software font
103 = Complex Italic software font
104 = Duplex Roman software font
105 = Triplex Roman software font
106 = Triplex Italic software font

Values less than 1 are set to 1. Values between 1 and 101 are hardware fonts. Values between the maximum number of fonts supported and 101 are set to 1. Values greater than 106 are set to 1.

**Output:** None

**Related Routines:** Define Notation String and Location, Set Notation Alignment, Set Notation Color, Set Notation Height

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All notation strings defined after this routine is called are plotted in the specified font.

If a software font (101-106) is to be used, the corresponding font table must be in your current directory or a path to its location must be set. See "Setting Environmental Parameters" in Appendix A.

See Figure 1-5 in Chapter 1, "Using Plotting System", for an example of each software font.

# Set Notation Height

**Purpose:** Specifies the character height of the notation string.

**Format:** Set Notation Height
(height)

**Input:** **height**
Character height of the notation string as a percentage of the view area. The range for *height* is 1 - 20. Values ⟨ 1 are set to 1 and values ⟩ 20 are set to 20. The default value for *height* is 5.

**Output:** None

**Related Routines:** Define Notation String and Location, Set Notation Alignment, Set Notation Color, Set Notation Font

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All notation strings defined after this routine is called are plotted in the specified character height.

# Set Output Primitive Color for Data Set

**Purpose:** Assigns an output primitive color to an attribute set.

**Format:** Set Output Primitive Color for Data Set
(attributes, color)

**Input:**

**attributes**
Attribute set identification number. The value of *attributes* must be 〉 0.

**color**
Output color index. The value of *color* must be ≥ 0. The default value for color is 1.

**Output:** None

**Related Routines:** Fill between Two Data Sets, Set Color Representation

**Status:**
0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The output primitive color assigned to an attribute set by this routine is associated with the following output primitives:

- Fill color for pie slices, bars, and area filling
- Line color for line and step charts
- Marker color for scatter charts.

Because only one attribute set is associated with each X/Y data set, the attribute set identification number for an X/Y data set is the same as the data set identification number.

When a pie chart is defined, one attribute set is associated with each slice within the pie chart (1 - 16). The color assigned to each attribute set by this routine is mapped one-to-one with pie slices 1 - 16.

When a schedule chart is defined, a color can be assigned to attribute sets 1 and 2. Attribute set 1 is associated with the segment of the task bar that represents the completed portion of the task. Attribute set 2 is associated with the incomplete portion of the task. The color set for each attribute set is maintained for each bar in the schedule chart.

The line color assigned to the *area1* data set by the Fill Between Two Data Sets routine determines the fill color for an area chart.

Starting with color index 0, the unique colors of the open device are assigned successive numbers until no more unique colors exist.

For requested color indexes greater than the device maximum, the maximum device value is used.

In the case of a monochrome display screen, the background color is color 0. The foreground color is used for all values 〉 0.

Color index values are found in the "Set Color Representation" routine in this chapter. The Set Color Representation routine can be used to map indexes to new colors.

# Set Output Primitive Outlining for Data Set

**Purpose:** Assigns an output primitive outline to an attribute set.

**Format:** Set Output Primitive Outlining for Data Set
(attributes, outline)

**Input:** **attributes**
Attribute set identification number. The value of *attributes* must be 〉 0.

**outline**
Outline indicator.

0 = bars or slices outlined in primitive color (default)
1 = bars or slices outlined in foreground color

Values 〈 0 are set to 0 and values 〉 1 are set to 1.

**Output:** None

**Related Routines:** Set Output Primitive Color for Data Set

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The output primitive outline assigned to an attribute set by this routine is associated with the following output primitives:

- Bars for bar and schedule charts
- Pie slices for pie charts.

Because only one attribute set is associated with each X/Y data set, the attribute set identification number is the same as the data set identification number.

When a pie chart is defined, one attribute set is associated with each slice within the pie chart. The outline assigned to each attribute set by this routine is mapped one-to-one with pie slices 1 - 16.

When a schedule chart is defined, an outline can be assigned to attribute sets 1 and 2. Attribute set 1 is associated with the segment of the task bar that represents the completed portion of the task. Attribute set 2 is associated with the incomplete portion of the task. The outline set for each attribute set is maintained for each bar in the schedule chart.

If *outline* = 0, the outline is the same color as the color index associated with *attributes* by the Set Output Primitive Color for Data Set routine.

# Set Output Primitive Style for Data Set

**Purpose:** Assigns an output primitive style to an attribute set.

**Format:** Set Output Primitive Style for Data Set
(attributes, style)

**Input:** **attributes**
Attribute set identification number. The value of *attributes* must be 〉 0.

**style**
Output primitive style index. The value of *style* must be 〉 0.

For line and step charts:

1=solid line (default)
2=long dashed line
3=dotted line
4=dashed-dotted line
5=medium-dashed line
6=dashed line with two dots
〉6=device-dependent

For scatter charts:

1=dot (default)
2=+
3=*
4=o
5=x
6=diamond
〉6=device-dependent

For pie slice, bar, or area filling:

1=hollow (default)
2=solid
3=narrow spaced +45 degree lines
4=medium spaced +45 degree lines
5=wide spaced +45 degree lines
6=narrow spaced +45 and –45 degree lines
7=medium spaced +45 and –45 degree lines
8=wide spaced +45 and –45 degree lines
⟩8=device-dependent

Values ⟨ 1 and ⟩ the maximum value supported by the device are set to 1.

**Output:** None

**Related Routines:** Fill between Two Data Sets

**Status:**

0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The output primitive style assigned to an attribute set by this routine is associated with the following output primitives:

- Fill style for pie, bar, and area filling
- Line style for line and step charts
- Marker style for scatter charts.

The style assigned to the *area1* data set by the Fill between Two Data Sets routine determines the fill style for an area chart.

Because only one attribute set is associated with each X/Y data set, the attribute set identification number is the same as the data set identification number.

When a pie chart is defined, one attribute set is associated with each slice within the pie chart (1 - 16). The style assigned to each attribute set by this routine is mapped one-to-one with pie slices 1 - 16.

When a schedule chart is defined, a style can be assigned to attribute sets 1 and 2. Attribute set 1 is associated with the segment of the task bar that represents the completed portion of the task. Attribute set 2 is associated with the incomplete portion of the task. The style set for each attribute set is maintained for each bar in the schedule chart.

# Set Output Primitive Width for Data Set

**Purpose:** Assigns an output primitive width to a data set.

**Format:** Set Output Primitive Width for Data Set
(attributes, width)

**Input:** **attributes**
Attribute set identification number. The value of *attributes* must be 〉 0.

**width**
Width of a bar or line.

Bar width is specified as the percentage of the distance between axis tick marks that a single bar or bar cluster occupies. For example, *width* = 60 defines a bar width of 60% of the tick interval. The default value for bar width is 80% of the tick interval.

Line width is specified in units of one-tenth of a percent of the horizontal dimension of the view area. For example, *width* = 15 defines a line with a width of 1.5% of the horizontal view area. The default value for line width is 0.1%.

The range for *width* is 1 - 100. Values 〈 1 are set to 1 and values 〉 100 are set to 100.

**Output:** None

**Related Routines:** None

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The output primitive width assigned to an attribute set by this routine is associated with the following output primitives:

- Bar width for bar and schedule charts
- Line width for line and step charts.

The output primitive width specified for the first attribute set associated with a schedule data set is maintained for each bar within the schedule chart.

Because only one attribute set is associated with each X/Y data set, the attribute set identification number is the same as the data set identification number.

# Set Polyline Color

**Purpose:** Assigns a color to a polyline.

**Format:** Set Polyline Color
(color)

**Input:** **color**
Color index.

The value of *color* must be ≥ 0. Values ⟨ 0 are set to 1. The default value for *color* is 1.

**Output:** None

**Related Routines:** Polyline, Set Color Representation, Set Polyline Type

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Starting with color index 0, the unique colors of the open device are assigned successive numbers until no more unique colors exist.

For requested color indexes greater than the device maximum, the maximum device value is used.

In the case of a monochrome device, the background color is color 0. The foreground color is used for all values ⟩ 0.

All polylines defined after this routine is called appear in the specified color.

Color index values are found in the "Set Color Representation" routine in this chapter. The Set Color Representation routine can be used to map indexes to new colors.

# Set Polyline Type

**Purpose:** Assigns a line type to a polyline.

**Format:** Set Polyline Type
(type)

**Input:** **type**
Line type index.

1=solid line (default)
2=long dashed line
3=dotted line
4=dashed - dotted line
5=medium dashed line
6=dashed with two dots line
〉6=device dependent

Values 〈 1 and 〉 the maximum supported by the device are set to 1.

**Output:** None

**Related Routines:** Polyline, Set Polyline Color

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All polylines defined after this routine is called appear in the specified type.

# Set Polymarker Color

**Purpose:** Assigns a color to a set of markers.

**Format:** Set Polymarker Color
(color)

**Input:** **color**
Color index.

The value of *color* must be ≥ 0. Values ⟨ 0 are set to 1. The default value for *color* is 1.

**Output:** None

**Related Routines:** Polymarker, Set Color Representation, Set Polymarker Style

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Starting with color index 0, the unique colors of the open device are assigned successive numbers until no more unique colors exist.

For requested color indexes greater than the device maximum, the maximum device value is used.

In the case of a monochrome device, the background color is color 0. The foreground color is used for all values ⟩ 0.

All polymarkers defined after this routine is called appear in the specified color.

The Set Color Representation routine can be used to map indexes to new colors.

# Set Polymarker Style

**Purpose:** Assigns a marker style to a set of markers.

**Format:** Set Polymarker Style
(style)

**Input:** **style**
Marker type index. The value of *style* must be 〉 0.

1=dot
2=+
3=* (default)
4=o
5=x
6=diamond

Values ≤ 1 are set to 1. Values 〉 6 are device-dependent. The default is 1.

**Output:** None

**Related Routines:** Polymarker, Set Polymarker Color

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** All polymarkers defined after this routine is called appear in the specified style.

# Set Subtitle Font

**Purpose:** Sets the character font for the chart subtitle.

**Format:** Set Subtitle Font
(font)

**Input:** **font**
Character font.

1 = Hardware font (default)
101 = Simplex Roman software font
102 = Complex Roman software font
103 = Complex Italic software font
104 = Duplex Roman software font
105 = Triplex Roman software font
106 = Triplex Italic software font

Values less than 1 are set to 1. Values between 1 and 101 are hardware fonts. Values between the maximum number of fonts supported and 101 are set to l. Values greater than 106 are set to 1.

**Output:** None

**Related Routines:** Define Subtitle String, Set Subtitle Height

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** If a software font (101-106) is to be used, the corresponding font table must be in your current directory or a path to its location must be set. See "Setting Environmental Parameters" in Appendix A.

See Figure 1-5 in Chapter 1, "Using Plotting System", for an example of each software font.

# Set Subtitle Height

**Purpose:** Sets the character height of the chart subtitle.

**Format:** Set Subtitle Height
(height)

**Input:** **height**
Character height of the subtitle as a percentage of the view area. The range for *height* is 1 - 20. Values 〈 1 are set to 1 and values 〉 20 are set to 20. The default value for *height* is 8.

**Output:** None

**Related Routines:** Define Subtitle String, Set Subtitle Font

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Because the graphic and text output display proportionally within the view area, the size of the plot is affected by the size of the title, subtitle, and axis title. The size of the plot is reduced so the subtitle does not overlap other titles or the plot.

# Set Title Font

**Purpose:** Sets the character font for the chart title.

**Format:** Set Title Font
(font)

**Input:** **font**
Character font.

1 = Hardware font (default)
101 = Simplex Roman software font
102 = Complex Roman software font
103 = Complex Italic software font
104 = Duplex Roman software font
105 = Triplex Roman software font
106 = Triplex Italic software font

Values less than 1 are set to l. Values between 1 and 101 are hardware fonts. Values between the maximum number of fonts supported and 101 are set to 1. Values greater than 106 are set to 1.

**Output:** None

**Related Routines:** Define Title String, Set Title Height

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** If a software font (101-106) is to be used, the corresponding font table must be in your current directory or a path to its location must be set. See "Setting Environmental Parameters" in Appendix A.

See Figure 1-5 in Chapter 1, "Using Plotting System", for an example of each software font.

# Set Title Height

**Purpose:** Sets the character height of the chart title.

**Format:** Set Title Height
(height)

**Input:** **height**
Character height of the title as a percentage of the view area. The range for *height* is 1 - 25. Values ⟨ 1 are set to 1 and values ⟩ 25 are set to 25. The default value for *height* is 10.

**Output:** None

**Related Routines:** Define Title String, Set Title Font

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Because the graphic and text output display proportionally within the view area, the size of the plot is affected by the size of the title, subtitle, and axis title. The size of the plot is reduced so the title does not overlap other titles or the plot.

# Set Vertical or Horizontal Chart

**Purpose:** Sets a vertical or horizontal direction for data.

**Format:** Set Vertical or Horizontal Chart
(type)

**Input:** **type**
Orientation of the chart.

1 = vertical (default)
2 = horizontal

Values 〈 1 are set to 1 and values 〉 2 are set to 2.

**Output:** None

**Related Routines:** None

**Status:** 0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** Vertical orientation is the most common form for line charts, scatter charts, step charts, and bar charts. The bars on bar charts extend vertically from a horizontal axis.

Usually horizontal orientation is applied only to bar charts. The bars extend to the right from a vertical axis. You may specify a horizontal orientation for a line or scatter chart. The lines or markers are plotted relative to the vertical axis. Avoid this charting technique because data presented like this can be misinterpreted unless the change in orientation is pointed out.

This routine has no effect on pie, schedule, or text-only charts.

# Set View Area Extents

**Purpose:** Specifies the view area extents as a percentage of the display surface area. Extents define the portion of the display surface available for output.

**Format:** Set View Area Extents
(xmin, ymin, xmax, ymax)

**Input:**

**xmin**
X coordinate of the lower-left corner of the view area.

**ymin**
Y coordinate of the lower-left corner of the view area.

**xmax**
X coordinate of the upper-right corner of the view area.

**ymax**
Y coordinate of the upper-right corner of the view area.

Coordinates are expressed as percentages of the display surface. Values ⟨ 0.0 are set to 0.0 and values ⟩ 100.0 are set to 100.0. The default view area extents are the full display surface size.

**Output:** None

**Related Routines:** Set Display Surface Size

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** The extents are percentages of the display surface area. The lower-left corner is (0.0, 0.0). The upper-right corner is (100.0, 100.0).

Charts are drawn to a maximum size within the view area.

# Set Visibility of Data Set

**Purpose:** Assigns visibility to an attribute set.

**Format:** Set Visibility of Data Set
(attributes, visibility switch)

**Input:** **attributes**
Attribute set identification number. The value of *attributes* must be ⟩ 0.

**visibility switch**
Visibility on/off switch.

0=data set not visible
1=data set visible (default)

Values ⟨ 0 are set to 0 and values ⟩ 1 are set to 1.

**Output:** None

**Related Routines:** Stack a Set of Data Sets

**Status:** 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** For a pie chart, this routine sets the visibility of the slice number (1 - 16) which corresponds to *attributes.*

For a bar, line, scatter, or step chart, this routine sets the visibility of the data set identified by *attributes.*

This routine has no effect on schedule or text-only charts.

The effect of stacking data sets defined as not visible varies with the chart type. Data sets defined as visible bars stacked upon data sets defined as not visible create a floating bar effect. With line, scatter, and step chart types, the invisible data set is added to the other data sets in the stack to determine the placement of the next visible data set.

# Stack a Set of Data Sets

**Purpose:** Defines the order to stack data sets.

**Format:** Stack a Set of Data Sets
(index, length, array)

**Input:**

**index**
Stack identification number. The value of *index* must be 〉 0.

**length**
Number of data sets to be stacked.

**array[]**
Identification numbers of the data sets to be stacked. The order in which the data sets appear in *array* defines the order in which they are stacked. Data set identifiers for data sets which do not exist are ignored. The number of elements in *array* must equal *length.*

**Output:** None

**Related Routines:** Set Baseline, Set Visibility of Data Set

**Status:**
 0=no error
-1=error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

If the stack is redefined, a value of 1 is returned.

**Remarks:** This routine defines the order in which a set of data sets is stacked. Multiple calls to this routine cause multiple stacks to be built. Multiple stacks are clustered at the axis tick marks. Multiple stacks display in ascending order.

Data sets are stacked in the order in which they appear in *array.* If *array* [1]=1, *array* [2]=5, and *array* [3]=3, when the chart is output data set 5 is stacked on data set 1 and data set 3 is stacked on data set 5.

Stacking can be specified for data sets defined as bar, line, scatter, or step.

This routine has no effect on pie, schedule, or text-only charts.

## Stack a Set of Data Sets

The effect of stacking data sets defined as not visible varies with the chart type. Data sets defined as visible bars stacked upon data sets defined as invisible create a floating bar effect. With line, scatter, and step chart types, the invisible data set is added to the other data sets in the stack to determine the placement of the next visible data set.

By default, multiple bar data sets are clustered around the tick marks. Call this routine to stack the data sets.

The Plotting System will generate stacked and individual bars within the same bar chart. The individual bars are positioned immediately following the stacked bars, with both the stack and the individual bars clustered around the axis tick marks. The same data set cannot be specified to appear individually and within a stack.

# Undefine a Data Set

**Purpose:** Deletes a data set and its related attributes.

**Format:** Undefine a Data Set
(data set id)

**Input:** **data set id**
Data set identification number. The value of *data set id* must be > 0.

**Output:** None

**Related Routines:** Define Schedule Chart Data Set, Define Sequenced Data Set, Define X/Y Data Set

**Status:**
0 = no error
-1 = error
Actual error can be retrieved by invoking Inquire on Plotting System Error.

**Remarks:** None

# Appendix A. Installing Plotting System

Plotting System is a library of software routines that enables you to create graphics within an application program. This appendix explains how to install the IBM RT PC Plotting System. Before you install this software, the AIX Operating System and the IBM RT PC Graphics Development Toolkit device drivers must be installed. These device drivers are provided as part of the IBM RT PC Multi-User programs.

## Installation Procedure

This section describes the installation procedure. If any error messages occur during the procedure, see the *IBM RT PC Messages Reference.*

The Plotting System product includes two diskettes. Remove the first diskette from the plastic envelope at the back of the binder.

1. Make sure that no one else is using the system and that no user programs are running. If the system is not in a quiet state, problems may occur as you install the various files for your licensed product.

2. Log in as super-user or as a member of the system group. You must have super-user authority, or be a member of the system group, to install a licensed program product. See the *IBM RT PC Using and Managing the AIX Operating System* for more information.

   After you log in, you will see the # prompt.

3. Type **installp** then press **Enter.**

   ```
   # installp
   ```

4. The following message appears to remind you to make sure that the system is quiet:

```
000-123 Before you continue, you must make sure there
        is no other activity on the system. You
        should have just restarted the system, and no
        other users should be logged on. Refer to
        your messages reference book for more
        information.

        Do you want to continue with this command?
        (y or n)
```

   Type **y** and press **Enter** to continue with the installp command.

5. Insert the first program diskette in response to the prompt. Then press **Enter.**

```
Insert the program diskette into diskette drive
"/dev/rfd0" and press Enter.
```

6. In response to the prompt, type **y** to indicate that you wish to continue with the installation. Then, press **Enter.**

```
The program "Plotting System"
will be installed.

Do you want to do this? (y/n)
```

7. If a current version of this program has already been installed on your system, a message explains that the version of the program you are about to install is the same as or older than the version you already have installed on your system. Indicate whether you wish to go ahead with the installation.

```
You are about to install version "xx.xx.xxxx" of this
program. This version is the same as or older than the
version currently on your system.
Do you want to do this? (y/n)
```

   If you type **y** and press **Enter,** the installation process begins.

```
Please insert the diskette in /dev/rfd0
```

Your program diskette should already be in the diskette drive (/dev/rfd0). **Type Return** is the same as **Press Enter.** As installation continues, various files are listed on the screen as they are copied to the fixed disk.

8. When installation is complete, remove the final program diskette from the diskette drive and replace it in its protective envelope in the binder.

   The installation process has completed.

9. Log off as super-user or as a member of the system group.

You may now begin using Plotting System.

## Checking Distribution Files

Following is a list of the files on the Plotting System diskettes. These files will be transferred by the installation procedure to the directories indicated.

Plotting System requires approximately 3100 blocks on the **/usr** minidisk. See *IBM RT PC Installing and Customizing the AIX Operating System* for additional information concerning minidisk size requirements and planning.

The language libraries reside in the directory, **/usr/lpp/plot/lib.** They are:

```
cplot.a
f77plot.a
basplot.a
pasplot.a
```

The following are include files for Pascal or BASIC programs. They reside in the **/usr/include** directory.

```
pasplot.int
extrnplot.bas
```

The font files used by Plotting System reside in the **/usr/lpp/vdi/fonts** directory.

font101.tbl
font102.tbl
font103.tbl
font104.tbl
font105.tbl
font106.tbl

The sample programs contained in **/usr/lpp/plot/bin** are listed below. The code and illustrations of the charts which are generated by the programs are printed at the back of each Language Reference Booklet for the applicable language. Those followed by .bas are written in BASIC, those followed by .c are written in C, those followed by .f are written in FORTRAN, and the programs followed by .pas are written in Pascal.

bbar.bas
bbar2.bas
binput.bas
bline.bas
bpie.bas
btext.bas

cbar.c
cbar2.c
cinput.c
cline.c
cpie.c
ctext.c

f77bar.f
f77bar2.f
f77input.f
f77line.f
f77pie.f
f77text.f

pbar.pas
pbar2.pas
pinput.pas
pline.pas
ppie.pas
ptext.pas

## Setting Environmental Parameters

To use Plotting System, you must export certain environmental parameters to the AIX Operating System. This can be done from the operating system, in either the "sh" or "csh" shell. Refer to the *IBM RT PC AIX Operating System Commands Reference* for a description of the "sh" and "csh" commands. The environmental commands can also be included in ".profile" (sh) or ".login" (csh) files which are executed each time a user logs in.

1. Set the VDIPATH parameter. This parameter provides a path to the directory where the IBM RT PC Graphics Development Toolkit device driver files reside. The installation procedure for the IBM RT PC Multi-User programs places the device driver files in **/usr/lpp/vdi/drivers.** VDIPATH is set as follows:

   ```
   VDIPATH=/usr/lpp/vdi/drivers                         (sh shell)
   export VDIPATH
   ```

   (or)

   ```
   setenv VDIPATH /usr/lpp/vdi/drivers                  (csh shell)
   ```

2. Any Logical Device Names (device driver logical names) which will be referenced in the Assign Plotting System Input Device or Assign Plotting System Output Device commands must be assigned to the appropriate device driver file names. These logical names are user-selectable; Plotting System has no pre-assigned logical names. The device driver files must be located in the directory specified by VDIPATH.

   The logical device name is assigned as follows:

   ```
   PRINTERA=vdi5182                                     (sh shell)
   export PRINTERA
   ```

   (or)

   ```
   setenv PRINTERA vdi5182                              (csh shell)
   ```

3. The device driver file names referenced in Step 2 must be assigned to the system's physical devices.

   For example:

   ```
   vdi5182=/dev/tty3                                    (sh shell)
   export vdi5182
   ```

   (or)

   ```
   setenv vdi5182 /dev/tty3                             (csh shell)
   ```

   **Note:** The **devices** program must be executed from the shell to establish tty ports for your graphics output devices. This program is native to the IBM AIX Operating System, but you must be a super-user to access it. The program prompts you for the required information. Refer to the *IBM RT PC Installing and Customizing the Operating System* for information about how to run the **devices** program.

4. If you select a printer, you may also want to pipe the output through the system's spooler. To do this, enter the following in place of step 3.

   ```
   vdi5182='|print -plot lpn'                           (sh shell)
   export vdi5182
   ```

   (or)

   ```
   setenv vdi5182 '|print -plot lpn'                    (csh shell)
   ```

   where lpn is the printer device name.

5. If you intend to send output to a plotter from a console display driver that is open, you must set a special environmental parameter called MESSAGEPORT. The MESSAGEPORT parameter may also be used to select a display device for the metafile “message” output.

   The default for this parameter is **/dev/tty**. If you do not set this parameter to **/dev/hft**, prompts from the plotter (such as those to change paper and pens) will not appear on the console. The commands to set MESSAGEPORT are:

   ```
   MESSAGEPORT=/dev/hft                                          (sh shell)
   export MESSAGEPORT
   ```

   (or)

   ```
   setenv MESSAGEPORT /dev/hft                                  (csh shell)
   ```

6. If you send output to metafile (device driver file **vdimeta),** you may want to assign the output metafile file name to be used instead of the default, METAFILE.DAT.

   The commands to do this are:

   ```
   METAOUTPUT=filename                                           (sh shell)
   export METAOUTPUT
   ```

   (or)

   ```
   setenv METAOUTPUT filename                                   (csh shell)
   ```

7. Plotting System uses font files which are loaded by the installation procedures into **/usr/lpp/vdi/fonts.** You need to set the path to the fonts as follows:

   ```
   FONTS=/usr/lpp/vdi/fonts                                      (sh shell)
   export FONTS
   ```

   (or)

   ```
   setenv FONTS /usr/lpp/vdi/fonts                              (csh shell)
   ```

   If you do not set the FONTS environment parameter, Plotting System will look for the font files in the current directory. The hardware text font will be used if the font files cannot be located.

# Appendix B. Error Codes

This appendix lists the error codes that are generated by Plotting System.

The Plotting System routines always return a *status code*, whether or not the requested operation was successful. If this status code is –1, indicating that an error has occurred, the Inquire on Plotting System Error routine must be used to obtain the actual error code.

Since the Plotting System is based upon the Virtual Device Interface within the Graphics Development Toolkit, some Toolkit errors may be encountered by an application program that uses Plotting System.

## Determining Toolkit Errors

The Graphics Development Toolkit error codes are separated into two types of codes:

- General Toolkit error codes
- Specific Toolkit error codes.

To determine an error code, first locate the general error code. Then, add the specific error code to it. For example, an error code of –508 means:

| | | |
|---|---|---|
| | –500 | Error while opening file. |
| + | | |
| | –8 | Insufficient memory for requested operation. |
| = | –508 | The actual error code. |

# General Toolkit Error Codes

The following list contains a brief explanation of general Toolkit error codes that may be returned by the Inquire on Plotting System Error routine.

| Code | Meaning |
|---|---|
| –200 | Error during the file connect operation. |
| –300 | Error during the file disconnect operation. |
| –400 | Error during the copy file descriptor operation. |
| –500 | Error while opening a file. |
| –600 | Error while closing a file. |
| –700 | Error while reading a file in wait mode. |
| –800 | Error while writing to a file wait mode. |
| –1200 | Error during seek operation. |
| –1400 | Error during file delete operation. |
| –1800 | Error during dashed line output. |
| –1900 | Error during marker output. |
| –2000 | Error during text alignment output. |
| –2100 | Error during polygon output. |
| –2200 | Error during bar output. |
| –2300 | Error during arc output. |
| –2400 | Error during pie output. |
| –2500 | Error during circle output. |
| –2600 | Error initializing input device. |
| –2700 | Error moving cursor during GIN. |
| –2800 | Error terminating GIN function. |

# Specific Toolkit Error Codes

The following list contains a brief explanation of specific Toolkit error codes that may be returned by the Inquire on Plotting System Error routine. This list is followed by additional explanation and suggestions about how to respond to the error code.

| Code | Meaning |
|---|---|
| –5 | Access denied. |
| –8 | Insufficient memory for requested operation. |
| –79 | Device is busy. |
| –80 | Device or hardware not present. |
| –81 | All (driver) slots used. |
| –82 | Toolkit can't start. |
| –84 | Driver can't start. |
| –86 | Slot (driver) already open. |
| –89 | No driver file. |
| –90 | Unknown driver, (driver file does not exist). |

–5 **Access denied:**

**Cause:** A request for Monitor Mode was denied. The requested driver is only accessible from the system console.

**Action:** Check that you are requesting the correct device. Use the system console if necessary.

**Cause:** Unable to create lock file for device driver. When an application selects a device for output, a lock file is created to prevent multiple applications from mixing their output on the same device. The device driver was unable to create this lock file.

**Action:** Check the directory for permission to create a file for that port.

If the AIX Extended Services programs are installed on the system, lock files are located in directory **/usr/spool/uucp**. If Extended Services programs are not installed, lock files are located in **/usr/lpp/vdi/locks**.

–8 **Insufficient memory for requested operation:**

**Cause:** This error means that you do not have enough system memory available to perform the requested operation.

**Action:** Add more memory to the system.

–79 **Device is busy:**

**Cause:** This error is generated when a **lock** file has been created for the device you are attempting to use. A lock file is created each time you perform graphics to a device that is not assigned to the logical device /dev/tty. The naming of the lock file has the form:

```
LCK..ttyxx
```

Where xx is the identification of the logical device being requested.

If the AIX Extended Services programs have been installed on the system, the lock file is located in directory **/usr/spool/uucp**. If Extended Services programs have not been installed on the system, the lock file will be located in directory **/usr/lpp/vdi/locks**.

**Action:** a) If the lock file was created by another graphics application, then you must wait for the other graphics application to release control of the device.

b) If the file was a remnant of a previous program error, delete the lock file and run your program again.

**–80 Device or hardware not present:**

**Cause:** The physical device is not attached to the IBM RT PC.

**Action:** Attach the physical device and restart your application.

**–81 All (driver) slots used:**

**Cause:** Each application is limited to having no more than eight graphics devices open simultaneously.

**Action:** Close one of the open graphics devices.

**–82 Toolkit can't start:**

**Cause:** This error message is generated when the Inter Process Communication (IPC) of the system is full. This occurs when you have exceeded the number of system shared memory areas or semaphores configured for your system. Two shared memory areas and one semaphore group are used for each graphics application. The shared memory areas are 512K and 362 bytes. The semaphore group contains eight semaphores.

**Action:** a) Make sure that shared memory areas are at least 512K in size.

b) Rebuild the AIX Operating System with more semaphores and/or shared memory areas.

c) Remove any unused shared memory areas with the **ipcrm** command.

d) Kill any unused running applications.

**–84 Driver can't start:**

**Cause:** The **fork** command, used to start the physical graphics device driver as a new process, failed.

**Action:** a) Wait until the total number of executing processes is less than the system-imposed process limit.

b) Wait until the total number of executing processes for a single user is less than the system-imposed process limit.

c) Ensure that there is enough paging space or physical memory for the process. If necessary, increase the paging area or add more physical memory.

**-86** **Slot (driver) already open:**

**Cause:** Trying to open a device that is already open—an application can only open a driver once.

**Action:** a) Ensure that you are requesting the correct device. Refer to "Setting Environmental Parameters" in Appendix A, for information about assigning logical device names to graphics devices.

b) Ensure that you have spelled the device name correctly.

c) Determine which is the existing device **handle** and either use this identifier or close and re-open the device.

**-89** **No driver file:**

**Cause:** The name of the logical device cannot be found.

**Action:** Ensure that the logical device name is defined in the current shell environment.

**-90** **Unknown driver, (driver file does not exist):**

**Cause:** Within the Toolkit, graphics devices are referred to by logical device names, (e.g. DISPLAY, PRINTER, PLOTTER, etc.). These names must be associated with physical graphics device drivers.

**Action:** a) Check that you are requesting the correct device. Refer to "Setting Environmental Parameters" in Appendix A, for information about assigning logical device names to graphics devices.

b) Check that you have spelled the device name correctly.

c) Check that the driver you are requesting is located in the VDIPATH specified.

# Plotting System Error Codes

| | |
|---|---|
| -10001 | Output device is not open. |
| -10002 | Arc angle is ≤0.0. |
| -10003 | Arc or circle radius is ≤0.0. |
| -10004 | Memory is not available. |
| -10005 | Input device is not open. |
| -10006 | Minimum label is greater than or equal to maximum label. |
| -10007 | Step between labels is greater than difference of minimum and maximum labels. |
| -10008 | Invalid chart type is specified, internal memory error occurred. |
| -10009 | Device to be closed is not open. |
| -10010 | Non-numeric character in either start date or end date. |
| -10011 | Month is ⟨ 1 or ⟩ 12 in either start date or end date. |
| -10012 | Day is ⟨ 1 or ⟩ 31 in either start date or end date. |
| -10013 | Data set does not exist. |
| -10014 | Schedule data set does not exist. |
| -10015 | Invalid fill area is defined; a data set is less than 0. |
| -10016 | Device identifier is not a valid device name. |
| -10017 | Data set identification number is less than 1. |
| -10018 | Length of output array is less than 1. |
| -10019 | Data set identification number is not defined. |
| -10020 | Data set was not previously defined. |
| -10021 | Number of markers to be output is less than 1. |
| -10022 | Number of data points is less than 1. |
| -10023 | Number of polyline points is less than 2. |
| -10024 | Device name is not an input device name. |
| -10025 | Device name is not an output device name. |
| -10026 | Stack identification number is less than 1. |
| -10027 | Number of items in the stack is less than 1. |
| -10028 | Display surface width is less than or equal to 0.0. |
| -10029 | Display surface height is less than or equal to 0.0. |
| -10030 | Minimum x is greater than maximum x. |
| -10031 | Minimum y is greater than maximum y. |
| -10032 | Device named is not open. |
| -10033 | Arc or circle center is outside current reference coordinate system. |
| -10034 | Axis tick label step is less than or equal to 0.0. |

# Advanced Programming Notes

The following text contains a few brief notes about possible error conditions that can occur during the interaction between IBM RT PC Graphics Development Toolkit device drivers and Plotting System. This information is intended for the advanced programmer and assumes a high degree of expertise with the IBM RT PC AIX Operating System.

The Graphics Development Toolkit drivers support **signal**, the IBM RT PC AIX Operating System error handling mechanism.

In certain software/hardware error conditions, such as an illegal memory reference or an illegal CPU instruction, the AIX Operating System sends a signal to a process to tell it that an error has occurred. By default, the operating system terminates the process as a result of the error.

When an application that uses the Toolkit drivers receives the signal, the driver system is shut down. Before exiting, the I/O channels that were being used are reset.

If the application writer intercepts signals before the first Assign Plotting System Input/Output Device call, the application's signal handler will be called after the Toolkit drivers signal handler has shut down the driver system. If the application intercepts signals after an Assign Plotting System Input/Output Device call, the application's signal handler is called first. The application must call the Toolkit drivers signal handler after it is through processing the signal. See the description of the **signal** system call in the *IBM RT PC AIX Operating System Technical Reference.*

## Special Error Handling Messages

In the event of an abnormal termination of the Toolkit drivers, one of the following error messages may be displayed.

| | |
|---|---|
| SIGQUIT | –Received quit signal: Application Terminated |
| SIGILL | –Illegal Instruction: Application Terminated |
| SIGTRAP | –Trace Trap: Application Terminated |
| SIGIOT | –IOT Instruction: Application Terminated |
| SIGEMT | –EMT Instruction: Application Terminated |
| SIGBUS | –Bus Error: Application Terminated |
| SIGSEGV | –Segmentation Violation: Application Terminated |
| SIGSYS | –Bad Argument to System Call: Application Terminated |

If abnormal termination is caused by any other signal, only the words "Application Terminated" will appear on the screen.

**Note:** Floating Point Exceptions (SIGFPE) are caught by Toolkit and ignored. The program will not terminate abnormally as a result of floating point exceptions. The programmer may choose to set up a signal handler to handle these exceptions if they are important to the application.

# Appendix C. Plotting System Defaults

Plotting System makes extensive use of default values to simplify chart generation.

In the following list of paired items, the default values appear on top with the routines used to change these values indented beneath them.

## Plotting Environment

**Background color is color index 0**
Set Background Color

**Chart area frame is 0.1% of the horizontal width of the view area**
Set Existence of Chart Area Frame

**Chart area is not framed**
Set Existence of Chart Area Frame

**Chart baseline is 0.0**
Set Baseline

**Chart orientation is vertical**
Set Vertical or Horizontal Chart

**Display surface units are in inches**
Set Display Surface Units

**Foreground color is color index 1**
Set Foreground Color

**No date line is drawn**
Set Date

**No X-axis label names**
Set Axis Labels

**No Y-axis label names**
Set Axis Labels

**View area frame is 0.1% of the horizontal width of the view area**
Set Existence of View Area Frame

**View area is full screen**
Set View Area Extents

**View area is not framed**
Set Existence of View Area Frame

**X-axis grid is invisible**
Set Existence of Grid Lines

**X-axis is linear**
Set Axis Type

**X-axis is not indented**
Set Axis Indent

**X-axis is visible**
Set Existence of Axis

**X-axis label type is value**
Set Axis Tick Label Type

**X-axis labels are 5% of view area height**
Set Axis Title Height

**X-axis labels are visible**
Set Existence of Axis

**X-axis precision is integer**
Set Axis Type

**Y-axis grid is invisible**
Set Existence of Grid Lines

**Y-axis is linear**
Set Axis Type

**Y-axis is not indented**
Set Axis Indent

**Y-axis is visible**
Set Existence of Axis

**Y-axis label type is value**
Set Axis Tick Label Type

**Y-axis labels are 5% of view area height**
Set Axis Title Height

**Y-axis labels are visible**
Set Existence of Axis

**Y-axis precision is integer**
Set Axis Type

## Attribute Set Definition

**All defined data sets are visible**
Set Visibility of Data Sets

**Data set chart type is line chart**
Define Data Set Chart Type to Be Bar

**Data set chart type is line chart**
Define Data Set Chart Type to Be Line

**Data set chart type is line chart**
Define Data Set Chart Type to Be Pie

**Data set chart type is line chart**
Define Data Set Chart Type to Be Scatter

**Data set chart type is line chart**
Define Data Set Chart Type to Be Schedule

**Data set chart type is line chart**
Define Data Set Chart Type to Be Step

**Data sets are not stacked**
Stack a Set of Data Sets

**Output primitive bar width is 80% of the tick interval**
Set Output Primitive Width for Data Set

**Output primitive color is 1**
Set Output Primitive Color for Data Set

**Output primitive line width is 0.1% of the horizontal width of the view area**
Set Output Primitive Width for Data Set

**Output primitive outlining is primitive color**
Set Output Primitive Outlining for Data Set

**Output primitive style is 1**
Set Output Primitive Style for Data Set

**Pie slice(s) not exploded**
Set Explode of Pie Slice

## Chart Titles

**Chart subtitle font is hardware font**
Set Subtitle Font

**Chart subtitle height is 8% of view area height**
Set Subtitle Height

**Chart title font is hardware font**
Set Title Font

**Chart title height is 10% of view area height**
Set Title Height

**No chart subtitle**
Define Subtitle String

**No chart title**
Define Title String

**No X-axis title**
Define Axis Title String

**No Y-axis title**
Define Axis Title String

**X-axis title font is hardware font**
Set Axis Title Font

**X-axis title height is 5% of the view area height**
Set Axis Title Height **Y-axis title font is hardware font**
Set Axis Title Font

**Y-axis title height is 5% of the view area height**
Set Axis Title Height

## Legend

**Legend box frame is not visible**
Set Existence and Location of Legend

**Legend does not exist**
Set Existence and Location of Legend

**Legend horizontal alignment is left**
Set Legend Alignment

**Legend is located at the (0.0, 0.0) point of view area**
Set Existence and Location of Legend

**Legend is not visible**
Set Existence and Location of Legend

**Legend text font is hardware font**
Set Legend Font

**Legend text height is 5% of the view area height**
Set Legend Height

**Legend vertical alignment is bottom**
Set Legend Alignment

## Independent Chart Components

**Fill color is the same as the foreground color**
Set Fill Color

**Fill style is hollow**
Set Fill Style

**Notation color is the same as the foreground color**
Set Notation Color

**Notation font is hardware font**
Set Notation Font

**Notation height is 5% of the view area height**
Set Notation Height

**Notation horizontal alignment is left**
Set Notation Alignment

**Notation vertical alignment is bottom**
Set Notation Alignment

**Polyline color is the same as the foreground color**
Set Polyline Color

**Polyline type is solid**
Set Polyline Type

**Polymarker color is the same as the foreground color**
Set Polymarker Color

**Polymarker style is dot**
Set Polymarker Style

# Appendix D. ASCII Character and Control Codes

The following table of ASCII characters is provided for your reference as a programming aid. The Plotting System uses only the ASCII values 32 through 127.

| ASCII Character | Character Name | Decimal Code | Hexadecimal Code |
|---|---|---|---|
| NUL | Null | 00 | 00 |
| SOH | Start of Heading (^A) | 01 | 01 |
| STX | Start of Text (^B) | 02 | 02 |
| ETX | End of Text (^C) | 03 | 03 |
| EOT | End of Transmission (^D) | 04 | 04 |
| ENQ | Enquiry (^E) | 05 | 05 |
| ACK | Acknowledge (^F) | 06 | 06 |
| BEL | Bell (^G) | 07 | 07 |
| BS | Backspace (^H) | 08 | 08 |
| HT | Horizontal Tab (^I) | 09 | 09 |
| LF | Line Feed (^J) | 10 | 0A |
| VT | Vertical Tab (^K) | 11 | 0B |
| FF | Form Feed (^L) | 12 | 0C |
| CR | Carriage Return (^M) | 13 | 0D |
| SO | Shift Out (^N) | 14 | 0E |
| SI | Shift In (^O) | 15 | 0F |
| DLE | Data Link Escape (^P) | 16 | 10 |
| DC1 | Device Control-1 (^Q) | 17 | 11 |
| DC2 | Device Control-2 (^R) | 18 | 12 |
| DC3 | Device Control-3 (^S) | 19 | 13 |

| ASCII Character | Character Name | Decimal Code | Hexadecimal Code |
|---|---|---|---|
| DC4 | Device Control-4 (ˆT) | 20 | 14 |
| NAK | Negative Acknowledge (ˆU) | 21 | 15 |
| SYN | Synchronous Idle (ˆV) | 22 | 16 |
| ETB | End of Transmission Block (ˆW) | 23 | 17 |
| CAN | Cancel (ˆX) | 24 | 18 |
| EM | End of Medium (ˆY) | 25 | 19 |
| SUB | Substitute (ˆZ) | 26 | 1A |
| ESC | Escape | 27 | 1B |
| FS | File Separator (ˆ\) | 28 | 1C |
| GS | Group Separator (ˆ]) | 29 | 1D |
| RS | Record Separator (ˆ↑) | 30 | 1E |
| US | Unit Separator (ˆ←) | 31 | 1F |
| SP | Space, Blank | 32 | 20 |
| ! | Exclamation Point | 33 | 21 |
| " | Quotation Mark | 34 | 22 |
| # | Number Sign, Hatch | 35 | 23 |
| $ | Dollar Sign | 36 | 24 |
| % | Percent Sign | 37 | 25 |
| & | Ampersand | 38 | 26 |
| ’ | Apostrophe | 39 | 27 |
| ( | Left Parenthesis | 40 | 28 |
| ) | Right Parenthesis | 41 | 29 |
| * | Asterisk | 42 | 2A |
| + | Plus Sign | 43 | 2B |
| , | Comma | 44 | 2C |

| ASCII Character | Character Name | Decimal Code | Hexadecimal Code |
|---|---|---|---|
| – | Hyphen | 45 | 2D |
| . | Period | 46 | 2E |
| / | Slash | 47 | 2F |
| 0 | Zero | 48 | 30 |
| 1 |  | 49 | 31 |
| 2 |  | 50 | 32 |
| 3 |  | 51 | 33 |
| 4 |  | 52 | 34 |
| 5 |  | 53 | 35 |
| 6 |  | 54 | 36 |
| 7 |  | 55 | 37 |
| 8 |  | 56 | 38 |
| 9 |  | 57 | 39 |
| : | Colon | 58 | 3A |
| ; | Semicolon | 59 | 3B |
| 〈 | Less Than | 60 | 3C |
| = | Equal Sign | 61 | 3D |
| 〉 | Greater Than | 62 | 3E |
| ? | Question Mark | 63 | 3F |
| @ | At Sign | 64 | 40 |
| A |  | 65 | 41 |
| B |  | 66 | 42 |
| C |  | 67 | 43 |
| D |  | 68 | 44 |
| E |  | 69 | 45 |
| F |  | 70 | 46 |

| ASCII Character | Character Name | Decimal Code | Hexadecimal Code |
|---|---|---|---|
| G | | 71 | 47 |
| H | | 72 | 48 |
| I | | 73 | 49 |
| J | | 74 | 4A |
| K | | 75 | 4B |
| L | | 76 | 4C |
| M | | 77 | 4D |
| N | | 78 | 4E |
| O | | 79 | 4F |
| P | | 80 | 50 |
| Q | | 81 | 51 |
| R | | 82 | 52 |
| S | | 83 | 53 |
| T | | 84 | 54 |
| U | | 85 | 55 |
| V | | 86 | 56 |
| W | | 87 | 57 |
| X | | 88 | 58 |
| Y | | 89 | 59 |
| Z | | 90 | 5A |
| [ | Left Bracket | 91 | 5B |
| \ | Reverse Slash | 92 | 5C |
| ] | Right Bracket | 93 | 5D |
| ^ | Caret | 94 | 5E |
| _ | Underscore | 95 | 5F |
| ` | Accent Grave | 96 | 60 |

| ASCII Character | Character Name | Decimal Code | Hexadecimal Code |
|---|---|---|---|
| a | | 97 | 61 |
| b | | 98 | 62 |
| c | | 99 | 63 |
| d | | 100 | 64 |
| e | | 101 | 65 |
| f | | 102 | 66 |
| g | | 103 | 67 |
| h | | 104 | 68 |
| i | | 105 | 69 |
| j | | 106 | 6A |
| k | | 107 | 6B |
| l | | 108 | 6C |
| m | | 109 | 6D |
| n | | 110 | 6E |
| o | | 111 | 6F |
| p | | 112 | 70 |
| q | | 113 | 71 |
| r | | 114 | 72 |
| s | | 115 | 73 |
| t | | 116 | 74 |
| u | | 117 | 75 |
| v | | 118 | 76 |
| w | | 119 | 77 |
| x | | 120 | 78 |
| y | | 121 | 79 |
| z | | 122 | 7A |

| ASCII Character | Character Name | Decimal Code | Hexadecimal Code |
|---|---|---|---|
| { | Left Brace | 123 | 7B |
| \| | Vertical Bar | 124 | 7C |
| } | Right Brace | 125 | 7D |
| ~ | Tilde | 126 | 7E |
| DEL | Delete or Rub Out | 127 | 7F |

# Glossary

**area chart.** A chart composed of two or more line or step charts plotted on the same axes. The areas above, below, or between individual curves are filled with a color or line pattern to emphasize volume or total quantity.

**array.** A series of related data elements identified by a single name.

**attribute set.** A collection of data attributes associated with a data set. An attribute set contains information such as chart type and line or marker style and color.

**axes.** Fixed lines of reference that define boundaries of a coordinate system. Their point of intersection forms the origin, usually labeled (0,0). Each axis is divided into units marked off at regular intervals by tick marks.

**background color.** The color that determines the backdrop of a chart.

**bar chart.** A chart in which a bar represents each element of an X/Y data set. The length of a bar is determined by the dependent variable. A bar chart compares magnitude or size of elements of a data set or compares an element of a data set to the data set as a whole.

**baseline.** A reference value which defines a critical chart deviation, for example, a value which divides positive-negative or profit-loss data. When a baseline is specified, all data points are plotted relative to the baseline rather than relative to the independent axis.

**binding.** The exact calling syntax and data-type specification for parameters when calling Plotting System routines from a specific programming language.

**character string.** A series of characters enclosed in quotes. “THIS IS A CHARACTER STRING” is a character string.

**chart area.** The portion of the view area that contains graphic output.

**clipping.** The removal by Plotting System of any part of the chart text which does not fit within the defined view area.

**clustered bar chart.** A bar chart used to compare two or more data sets with the same independent values. Bars are clustered at the tick marks on the independent axis.

**color index.** An integer identifying a color. The value of a particular index depends on the device used. For example, color index 0 indicates black for CRTs and white for plotters.

**curve.** A continuous line joining data points on a line chart.

**data set.** A collection of related data elements.

**default.** A value assigned to a parameter by the Plotting System. A default value is used when you do not specify a value.

**dependent axis.** The axis along which dependent variables are plotted. In an X/Y-axis system, the dependent axis is usually the Y-axis.

**dependent variable.** A variable whose value is determined by the value of another variable.

**deviation bar chart.** A bar chart with bars or lines extending to either side of a specified baseline. Positive-negative and profit-loss data are effectively represented by a deviation chart.

**device driver.** The device-dependent part of a graphics implementation. Device drivers support physical devices, generating device-dependent output.

**device independence.** The feature that allows Plotting System routines to be used on more than one type of graphics display device.

**display surface.** The part of a device on which a visible image appears; for example, the screen of a display and the paper in a plotter.

**environmentals.** Attributes, selected by the user, that define the appearance of the displayed chart, such as the foreground and background color of a chart.

**extents.** Values that define the portion of the view area available for output and the length of the axes.

**floating bar chart.** A stacked bar chart in which not all segments of all bars are displayed.

**font.** A style and size of print for text output.

**foreground color.** The color that determines the chart nomenclature, chart and view area frames, grid and tick marks, and filled area outline.

**frame.** A line that encloses the view area or a specific chart area. The thickness of a frame can be specified.

**graphic primitives.** Basic graphic components, such as lines, markers, circles, arcs, rectangles, and arrows.

**graphics cursor.** A cursor in the shape of a crosshair, or plus sign (), that is used to enter points with the Request Locator function.

**graphics input device.** A device, such as a keyboard, mouse, joystick, tablet, plotter, or display, that transmits input data to the Plotting System.

**graphics output device.** A device, such as a printer, plotter, or display, that receives the transmitted graphics from the Plotting System.

**grid lines.** Horizontal and/or vertical lines emanating from axis tick marks.

**hardware font.** The font built into the hardware device.

**horizontal bar chart.** A bar chart with the bars arranged horizontally.

**icon.** A picture or symbol located on the screen that represents a function or action you can perform.

**independent axis.** The axis along which independent variables are plotted. In an X/Y-axis system, the independent axis is usually the X-axis.

**independent variable.** A variable whose value is determined without reference to another variable.

**joystick.** A stick that is moved in two dimensions to provide X and Y coordinated data for graphic input.

**legend.** The explanatory caption that accompanies a chart.

**legend box.** The area in a chart that contains the legend.

**line chart.** A chart derived by plotting one or more sets of data points on a coordinate surface. The data points in each set are joined together in a continuous line or curve.

**markers.** The characters that indicate data points in a scatter chart.

**metafile.** A mechanism for retaining and transporting graphic data and control information. This information contains a device-independent description of a picture.

**multiple bar chart.** A bar chart produced by plotting more than one set of data on a coordinate surface.

**multiple line chart.** A line chart produced by plotting more than one set of data on a coordinate surface.

**notation.** A character string or strings added to a chart to provide remarks about the chart. Notation can appear anywhere within the view area in any font and height.

**100-percent bar chart.** A bar chart that has two or more segmented bars with each bar totaling 100%. The divisions of the bars represent percentages of the whole.

**output primitives.** Polylines, polymarkers, arcs, arrows, circles, and rectangles.

**origin.** The point of intersection of the X- and Y-axes.

**parameter.** A value passed between a calling program and a routine. Parameters are enclosed in parentheses.

**pie chart.** A circular chart with radii that divide the circle into sectors or slices, each proportional in angle and area to the relative size of the data represented. A pie chart illustrates relation of parts to a whole.

**polyline.** A series of connected line segments.

**polymarker.** A Plotting System output primitive consisting of a series of marker symbols drawn at specified points.

**scatter chart.** A chart with each data point plotted separately, without being connected. A scatter chart portrays degree and type of relation or covariation between two variables.

**schedule chart.** A chart that illustrates task status. Horizontal bars represent each task. Each bar of the schedule chart requires a unique schedule data set.

**schedule data set.** A data set used to generate one bar of a schedule chart. The elements of a schedule data set are the start date of the schedule, the end date of the schedule, and the percent completion of the scheduled task.

**sequenced data set.** A data set composed of data elements ordered by a sequence of variables. The sequence begins with 1, increments by 1, and ends with a number equal to the number of elements within the data set.

**simple bar chart.** A bar chart produced by plotting one set of data on a coordinate surface.

**simple line chart.** A line chart produced by plotting one set of data on a coordinate surface.

**software font.** A font created through software. Text generated using software font generally takes more time to print or display than text generated with hardware font. This is because many calculations must be performed on each character before it can be drawn.

**stacked bar chart.** A bar chart in which the bars are placed side by side (horizontal bar chart) or one on top of the other (vertical bar chart). A stacked bar chart illustrates the sum of two or more data sets with the same independent values.

**step chart.** A line chart in which data points are connected with lines that lie at right angles to the axes. A step chart is used to display data with definite limits and no intermediate values.

**text-only chart.** A list of text items usually marked with bullets or other markers.

**tick label.** A character string that annotates a tick mark.

**tick marks.** Divisions along an axis.

**title.** A character string associated with a chart title, subtitle, or axis title. The font and height of a title can be specified.

**vertical bar chart.** A bar chart with the bars arranged vertically.

**view area.** The portion of the display surface designated for the output of a chart.

**Virtual Device Interface (VDI).** The interface between the device-independent and the device-dependent levels of a graphics system.

**Virtual Device Metafile (VDM).** A mechanism for saving and transporting graphics data and control information at the level of the VDI.

**X-axis.** The horizontal axis. The X-axis normally represents the independent variable.

**X/Y-axis system.** A coordinate system composed of a horizontal X-axis increasing positively toward the right and a vertical Y-axis increasing positively upwards. The axes are positioned at right angles and the point of intersection is the origin (0,0). The position of any point is defined by the displacement from the origin along first the X- and then the Y-axis.

**X/Y-axis chart.** A chart that compares two variables, such as dollar growth against time, mileage against distance, or percent change against time. One variable is plotted along the independent X-axis and the other is plotted relative to the dependent Y-axis. X/Y-axis charts are generated with X/Y data sets.

**X/Y data set.** A data set composed of two arrays of data values: one array of the independent X-axis coordinates and one array of the dependent Y-axis coordinates.

**Y-axis.** The vertical axis. The Y-axis normally represents the dependent variable.

# Index

IBM

The IBM RT PC
Family

# Reader's Comment Form

## IBM RT PC Plotting System Programmer's Guide

SV21-8059

Your comments assist us in improving our products. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

For prompt resolution to questions regarding set up, operation, program support, and new program literature, contact the authorized IBM RT PC dealer in your area.

Comments:

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 997, Building 998
11400 Burnet Rd.
Austin, Texas 78758

Fold and tape

Fold and tape

Cut or Fold Along Line

Tape

Please Do Not Staple

Tape

IBM RT PC PLOTTING SYSTEM PROGRAMMER'S GUIDE
**Book Title**

SV21-8059
**Order No.**

## Book Evaluation Form

Your comments can help us produce better books. You may use this form to communicate your comments about this book, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Please take a few minutes to evaluate this book as soon as you become familiar with it. Circle Y (Yes) or N (No) for each question that applies and give us any information that may improve this book.

Y N Is the purpose of this book clear?

Y N Is the table of contents helpful?

Y N Is the index complete?

Y N Are the chapter titles and other headings meaningful?

Y N Is the information organized appropriately?

Y N Is the information accurate?

Y N Is the information complete?

Y N Is only necessary information included?

Y N Does the book refer you to the appropriate places for more information?

Y N Are terms defined clearly?

Y N Are terms used consistently?

Y N Are the abbreviations and acronyms understandable?

Y N Are the examples clear?

Y N Are examples provided where they are needed?

Y N Are the illustrations clear?

Y N Is the format of the book (shape, size, color) effective?

### Other Comments

What could we do to make this book or the entire set of books for this system easier to use?

### Optional Information

Your name

Company name

Street address

City, State, ZIP

No postage necessary if mailed in the U.S.A.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 997, Building 998
11400 Burnet Rd.
Austin, Texas 78758

Fold and tape

Fold and tape

Cut or Fold Along Line

Tape

**Please Do Not Staple**

Tape

International Business
Machines Corporation
Department 997, Building 998
11400 Burnet Rd.
Austin, Texas 78758

Printed in the
United States of America

59X8583