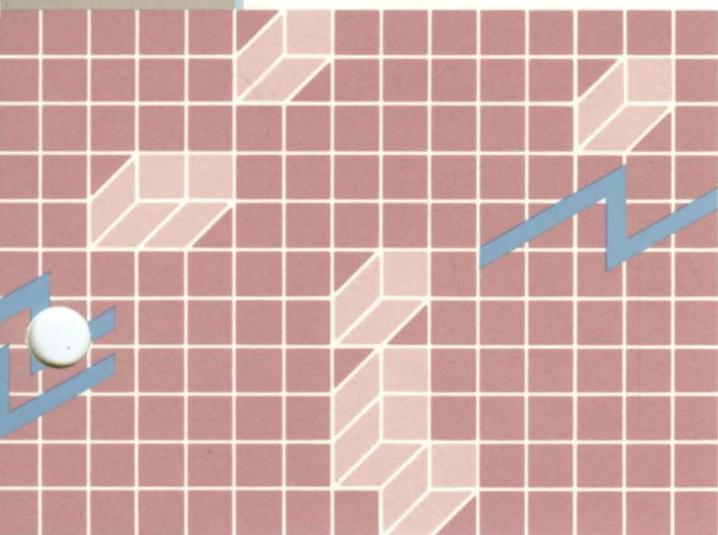


*Generic Print System
Programmer's Guide*



GENERIC PRINT SYSTEM PROGRAMMER'S GUIDE

Specifications Subject to Change.

Convergent Technologies and NGEN are registered trademarks of Convergent Technologies, Inc.

Art Designer, Chart Designer, Convergent, CT-DBMS, CT-MAIL, CT-Net, CTIX, CTOS, DISTRIX, Document Designer, The Operator, AWS, CWS, IWS, MegaFrame, MightyFrame, MiniFrame, MiniFrame Plus, Multibus, Voice/Data Services, Voice Processor, and X-Bus are trademarks of Convergent Technologies, Inc.

IBM is a registered trademark of International Business Machines Corporation.

Microsoft, MS, GW, and XENIX are trademarks of Microsoft Corporation.

UNIX is a trademark of Bell Laboratories.

First Edition (July 1986) 09-00985-01

Copyright © 1986 by Convergent Technologies, Inc., San Jose, CA. Printed in USA.

All rights reserved. Title to and ownership of the documentation contained herein shall at all times remain in Convergent Technologies, Inc., and/or its suppliers. The full copyright notice may not be modified except with the express written consent of Convergent Technologies, Inc.

CONTENTS

1	OVERVIEW.....	1-1
	USING THIS MANUAL.....	1-1
	RELATED DOCUMENTATION.....	1-4
2	INTRODUCTION TO THE GENERIC PRINT SYSTEM	2-1
	PROGRAMMING IN THE GPS ENVIRONMENT.....	2-1
	GPS Byte Streams.....	2-3
	The Generic Print Access Method.....	2-5
	Generic Print System Requests.....	2-6
	Output Functions.....	2-6
	Status and Control Functions.....	2-6
	AN OVERVIEW OF THE GENERIC PRINT SYSTEM	
	SERVICES.....	2-8
	The GPS Routing Switch.....	2-9
	The GPS Device Driver.....	2-10
	The Spooler.....	2-12
	The Queue Manager.....	2-12
3	USING GPS BYTE STREAMS.....	3-1
	GPS BYTE STREAMS.....	3-2
	LINKING YOUR PROGRAM TO INCLUDE	
	GPS BYTE STREAMS.....	3-3
	EXAMPLE.....	3-3
4	USING THE GENERIC PRINT ACCESS METHOD...	4-1
	LINKING WITH GPAM DATA.....	4-3
	OPENING AND CLOSING A GPAM DATA	
	STREAM.....	4-8
	DATA STREAM STRUCTURE.....	4-8
	Legal Characters.....	4-10
	PAGINATION.....	4-10
	PRINT POSITION AND CHARACTER	
	PLACEMENT ON THE PAGE.....	4-11
	ASSIGNING TEXT ATTRIBUTES.....	4-12
	TIMEOUTS.....	4-13

	SENDING NON-GPAM COMMANDS TO THE PRINTING DEVICE.....	4-14
	USING GPAM FOR GRAPHICS.....	4-14
	Graphics Positioning.....	4-15
	Scaling.....	4-15
	The Graphics Routines.....	4-15
	Using CGM.....	4-17
5	USING GPS REQUESTS.....	5-1
	MAKING A PRINT REQUEST WITH GPS OUTPUT FUNCTIONS.....	5-2
	Spooled or Direct Printing.....	5-5
	Printing Mode.....	5-6
	SPECIFYING A PRINTING DEVICE.....	5-6
	GPS STATUS AND CONTROL FUNCTIONS.....	5-7
	Status Functions.....	5-7
	Controlling the Printing Device.....	5-7
	EXAMPLE.....	5-8
6	GPAM PROCEDURES.....	6-1
7	GPS ROUTING SWITCH REQUESTS.....	7-1
8	TROUBLESHOOTING.....	8-1
	STATUS CODES.....	8-1
	ROUTING SWITCH STATUS CODES.....	8-3
	SPOOLER STATUS CODES.....	8-5
	QUEUE MANAGER STATUS CODES.....	8-6
	APPENDIX A: CONVERTING NON-GPS APPLICATIONS.....	A-1
	GLOSSARY.....	G-1
	INDEX.....	I-1

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
2-1.	Three Methods of Printing with GPS..	2-4
2-2.	A Simple Printing Application.....	2-5
2-3.	A Sophisticated Printing Application	2-7
2-4.	Direct Printing with GPS.....	2-14
2-5.	Spooled Printing with GPS.....	2-15
2-6.	Printer Status Queries with GPS.....	2-16
2-7.	Issuing Printer Control Commands with GPS.....	2-17
2-8.	Spooled Printing on a Standalone Workstation.....	2-18
2-9.	Spooled Printing on a Simple Cluster	2-19
2-10.	Spooled Printing Across the Network.	2-20
4-1.	Character Placement.....	4-11
6-1.	Repositioning.....	6-37
6-2.	Color Array.....	6-52
6-3.	A Square Has Five Points.....	6-60
6-4.	Maximum X and Y Values.....	6-61

LIST OF TABLES

<u>Table</u>		<u>Page</u>
4-1	GPAM Procedure Summary.....	4-4
5-1	Summary of Routing Switch Requests..	5-3
6-1	GPAM Procedure Summary.....	6-2
7-1	Summary of Routing Switch Requests..	7-2
7-2	Format of AddGPSDevice Device Structure.....	7-44
7-3	Format of GPS Net List.....	7-46
7-4	Device Configuration Information....	7-47
7-5	Device History Information.....	7-49
7-6	Format of a History Entry.....	7-50
7-7	Document Information.....	7-51
7-8	Device Status Information.....	7-55
7-9	Job Parameters.....	7-57
7-10	Page Location Record Definition.....	7-60

1 OVERVIEW

The Generic Print System (GPS) is a comprehensive software printing system that simplifies the tasks your application must perform to output to the printer. These tasks have been made "generic" whenever possible, so that you can use GPS with a wide variety of applications and output devices. GPS works in a clustered environment, within a network, or on a standalone system.

Four dynamically-installed GPS system services pass the data to the printer or plotter for output and handle communications within the cluster or network. Your application makes requests of GPS; GPS tracks the print job and feeds back the information your application requests about the job.

To make applications programming for printing routines as free from hard-coded, device-specific operations as possible, GPS provides the following device-independent features:

- Sequential Access Method (byte stream) access to GPS devices.
- The Generic Print Access Method (GPAM) is an easy-to-use library of printing routines that you can use to access a GPS device, as well as to print text and/or graphics; GPAM provides a device-independent method to add complex text formatting, such as multiple text typefaces and styles, or graphics.
- A unique device driver that controls the device and translates the generic output your application sends to GPS into device-specific control commands. The GPS distribution disk includes one device driver for each type of supported printing device.

- A programmatic interface your application can use to check printer status or to send printer control commands (such as pause and restart).
- Identical calls to use when sending output to either a direct or a spooled device.

In addition, the Generic Print System includes the Print Manager, an interactive utility used to install and monitor printers. The Print Manager can also be used to print files. It is described in detail in the Printing Guide.

USING THIS MANUAL

This manual describes how an application program uses the Generic Print System and the Generic Print Access Method to print and to perform printer control operations.

It is assumed that the programmer has basic familiarity with programming in the CTOS environment. If this is not the case, read "Getting Started," in the CTOS Programmer's Guide. Also read Chapters 1 through 3, and Chapter 13, "Sequential Access Method," in the CTOS Manual.

Installation and configuration of the Generic Print System is described in the Printing Guide and the Generic Print System Release Notice.

In general, the Generic Print System is used in the same way whether it is running on a standalone system or on a workstation within a cluster. Throughout this manual, whenever the term cluster is specified, it is assumed that the statement also applies to standalone configurations.

GPS can be used with shared resource processors as masters. When the term master workstation is used in the manual it means either a workstation or a shared resource processor.

Chapter 2, "Introduction to the Generic Print System," provides an overview of programming in the GPS environment and describes how the Generic Print System functions.

Chapter 3, "Using GPS Byte Streams," discusses the simplest type of printing your application can perform with the Generic Print System. GPS byte streams is a quick, efficient way to output text-only documents.

Chapter 4, "Using the Generic Print Access Method," discusses using GPAM to send device-independent formatting to the printing device. GPAM allows fully formatted text and graphics output.

Chapter 5, "Using GPS Requests," describes the request and discusses how your application system can use the power of the Generic Print System to provide your user with more complex printing facilities.

Chapters 6 and 7 are reference chapters that describe the GPAM and GPS operations in great detail.

Chapter 8, "Troubleshooting," lists error codes and error conditions you may encounter.

Appendix A briefly discusses how to interface with existing non-GPS applications.

Additionally, there are example programs on the distribution disk that demonstrate the various features of GPS. These examples are referred to in this manual, but you do not need them to use the manual.

RELATED DOCUMENTATION

For a complete list of Convergent Technologies publications, see the "Guide to Technical Documentation" in the Executive Manual or similar command-line interpreter manual for your operating system.

The CTOS Operating System Manual describes the CTOS operating system. It specifies services for managing processes, messages, memory, exchanges, tasks, video, disks, keyboard, printer, timer, communications, and files. In particular, it specifies the standard file access methods: SAM, the sequential access method; RSAM, the record sequential access method; and DAM, the direct access method.

The CTOS Programmer's Guide (formerly titled the System Programmer's Guide) is a reference guide for use in programming under the CTOS operating system. It describes CTOS programming practices and introduces the system to the programmer who is using it for the first time.

The Printing Guide provides the information you need to install any supported printing device on your standalone workstation or within your cluster, and describes the Print Manager, the interface to the Generic Print System. It also describes how to use the Print Manager on a day-to-day basis to monitor the status of printing devices and to control the printer. Printer troubleshooting is also discussed.

2 INTRODUCTION TO THE GENERIC PRINT SYSTEM

The Generic Print System (GPS) consists of four dynamically installed system services that work together to control the printing device and to handle communication between application systems, the operating system, and the printers or plotters currently installed. These system services are as follows:

- Routing Switch
- Device Driver
- Spooler
- Queue Manager

The GPS system services are described in more detail later in this chapter.

A dynamically installed system service is a modular part of the operating system that resides in workstation memory and can be added or removed at any time. System services, therefore, are not a part of the operating system when it is first installed, nor are they resident after a system is started unless you make special entries in your system initialization file for their installation.

An application system can make a request of any system service at any time.

System services and the way that they work are described in detail in the CTOS Operating System Manual. Also consult the CTOS Programmer's Guide for information on interfacing with system services.

The Generic Print System handles all print requests your application makes to a device under its control. Whenever a printer is installed with

GPS, a printer name is assigned that is unique within the cluster. This printer name is all your application system needs to know in order to send a print request to any installed printer. GPS handles the routing of the request to the appropriate location, wherever the printer is in the cluster or network. In the same way, your application does not need to know whether the printing device has been set up for direct or spooled printing, or whether it is a serial or parallel device.

In addition to the GPS system services, GPS includes the following features:

- Generic Print Access Method (GPAM), a library of object modules procedures used for text and graphics formatting.
- Print Manager, an interactive utility that provides print, status, and installation facilities for GPS. Using the Print Manager is described in the Printing Guide.

PROGRAMMING IN THE GPS ENVIRONMENT

GPS provides three ways that your application can make print requests:

- using Generic Print System byte streams, which communicate with the GPS system services through the Sequential Access Method (SAM)
- using the Generic Print Access Method (GPAM), a library of object module procedures which send generic formatting commands to the printer through the Sequential Access Method

- using Generic Print System requests, which communicate directly with the GPS system services

These three methods are shown in Figure 2-1. In addition, GPS provides status and control functions that you can use whether you are using byte streams, GPAM, or direct calls to GPS as the method to transfer data to the printer.

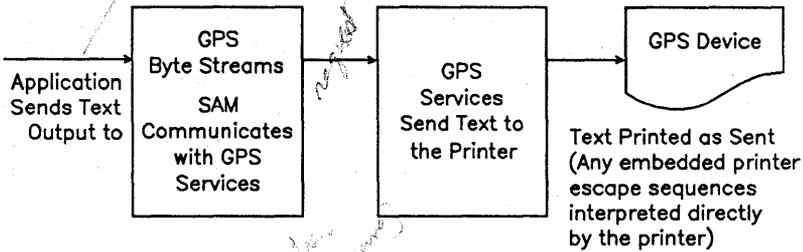
GPS BYTE STREAMS

Using GPS byte streams is the simplest way for your application to print a document. With this method, you specify the printer name for the Sequential Access Method and it passes the text to the GPS system services for you. Your application does not even have to know that any of the GPS system services exist.

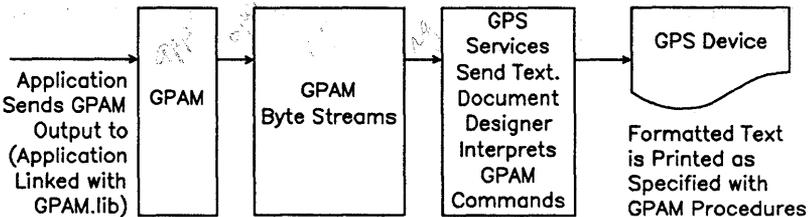
Figure 2-2 shows a simple printing application of this type. (See Chapter 3, "Using GPS Byte Streams," for more information and an example of this type of simple application.)

Open Cycle Stream [GPS Module Name]

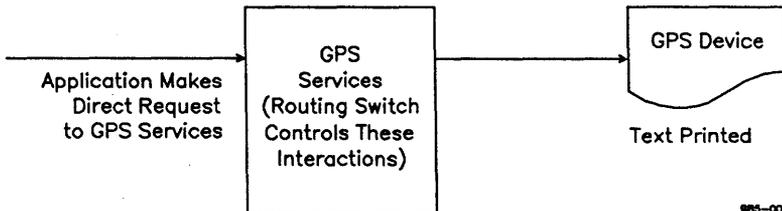
Byte Streams



GPAM

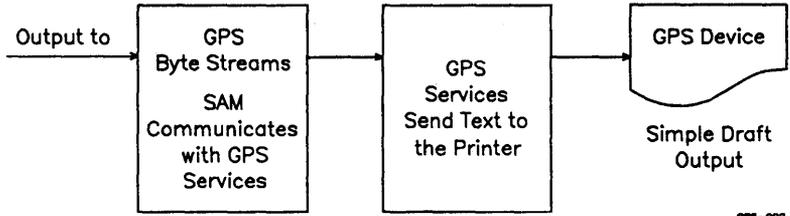


Direct Requests to GPS Services



985-001

Figure 2-1. Three Methods of Printing with GPS



985-002

Figure 2-2. A Simple Printing Application

THE GENERIC PRINT ACCESS METHOD

If you want to include formatting attributes, such as boldface text, different font styles, or graphics with your text, you can use calls to GPAM which work very much like a byte stream. When you use GPAM you include generic formatting information with the data. GPAM handles communications with the GPS system services for you in the same way that byte streams do; in addition, the generic formatting commands it provides are translated by GPS for the specific printer to which the output is sent.

All printers are not created equal; some printers may not be capable of outputting the formatting features you have specified. In these cases, GPAM is designed for graceful degradation. If the output device cannot perform a certain function, GPAM instructs it to do the next best thing. For example, if your printer cannot output boldface text, GPAM simply outputs all such text as plain text. GPAM does not reject your print request; it executes it as well as it is able, subject to hardware (and sometimes software) limitations.

(Constructing a GPAM data stream is described in Chapter 4, "Using the Generic Print Access Method." Each procedure from the Generic Print Access Method library is described in detail in Chapter 6, "GPAM Procedures.")

GENERIC PRINT SYSTEM REQUESTS

Direct calls to GPS can be made for two types of functions

- printing (output functions)
- status inquiry or printer control (status and control functions)

Output Functions

GPS byte streams or GPAM send output to the printer through the GPS system services.

Note that byte streams, GPAM, and GPS output requests must be used separately; your application should choose only one output request at a time.

It is easier to use GPS byte streams or GPAM to print than to try to make output requests directly of GPS.

Status and Control Functions

The GPS status and control functions, all of which make calls to one of the GPS services (the routing switch), can be used with any of the three output functions. You can check status and send control commands only by making these calls directly to GPS.

Using the additional functions the routing switch offers, you could, for example, display a list of printers for the user to choose from, or you could make status enquiries and send the printing device control commands like pause and restart.

Figure 2-3 shows this type of more sophisticated printing application.

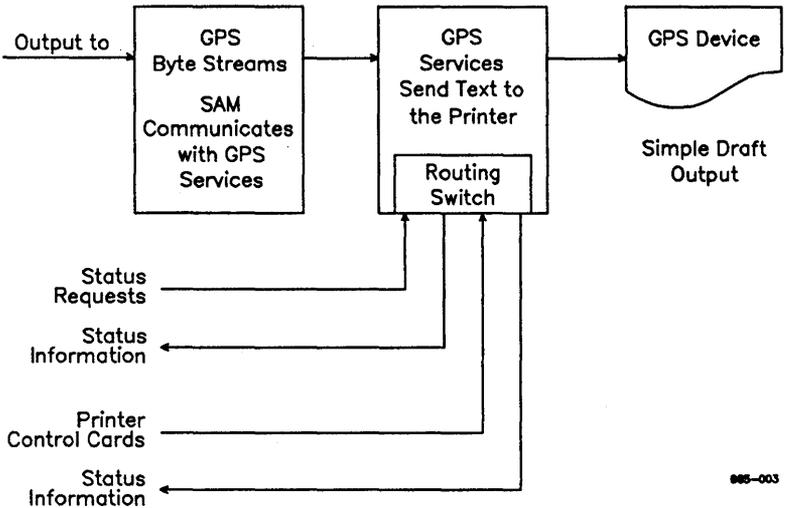


Figure 2-3. A Sophisticated Printing Application

The status and control functions also include a function to find out the users assigned default printer name.

Making calls to GPS is described in Chapter 5, "Using GPS Requests." The requests themselves are described in detail in Chapter 7, "GPS Routing Switch Requests."

AN OVERVIEW OF THE GENERIC PRINT SYSTEM SERVICES

Four different system services make up the Generic Print System. GPS is modular, so the print system can be tailored to specific needs. Only those modules that are needed are installed, and not all modules are needed for every workstation in a cluster. Several installations of the same system service can also exist within each cluster.

The following is a brief description of the four GPS system services:

- The routing switch routes requests to the designated printing device and coordinates the work of the entire Generic Print System.

The routing switch also handles printer control commands (such as pause and restart), and status enquiries. Routing switches within a cluster or network communicate with each other to pass requests or status information.

A routing switch is installed on each workstation to which a printer is attached. In addition, a routing switch is usually also installed on the master workstation of each cluster.

- The device driver controls the device and translates the Generic Print Access Method byte stream it receives into device-specific commands. Device drivers are usually unique for each type of printing device.

A device driver is installed on the workstation to which the device it controls is attached.

- The spooler gives a print request to the Queue Manager to place in the print queue. When the printing device is free to print, the spooler asks for the print request back again.

The spooler is installed on each workstation to which a spooled device is attached.

- The Queue Manager handles all queues of jobs for the cluster. It is used by other applications, such as Batch or SNA/RJE, as well as by GPS.

A Queue Manager must be installed on the master workstation of each cluster where spooled devices are installed, or on any standalone system where you want to have spooled printing. Installation of these services and the configuration of printers and plotters for use with GPS is described in the Printing Guide.

Figures 2-4 through 2-10 at the end of this chapter show how output data and status information flow from service to service within the print system.

THE GPS ROUTING SWITCH

The GPS routing switch is the focal point of the Generic Print System. When an application makes a print request, the routing switch sends the request to the device driver for the specified device and monitors the printing process. Printer control commands or requests for status information are also handled by the routing switch.

The routing switch keeps a list of installed printing devices for your standalone workstation or cluster. It polls this list regularly to check printer activity.

A routing switch must be installed on each workstation to which a printer is attached. If printers are to be shared within a cluster, or shared with another node in a CT-Net network, then there must also be a routing switch at the master workstation, whether or not a printer is attached to it.

The routing switch acts as the request interface between your application and the Generic Print System. Requests are described in Chapter 5, "Using GPS Requests."

THE GPS DEVICE DRIVER

There is a specific device driver for every installed device in your print system. Device drivers are written specifically for each type of device to drive the printer and to translate a device-independent GPAM data stream into device-specific commands.

Your application system communicates with the device drivers indirectly, through the routing switch.

Despite the specific capabilities of the printing device, the output produced depends on the device driver used. Output can be as follows:

- draft quality output, without special formatting such as boldface or underscored text
- letter quality output, which includes formatting such as boldface, underscore, superscript, and subscript text
- graphics

Both text and graphics can be printed on the same page within a document.

The Generic Print System is distributed with device drivers for several supported printing devices.

There is a generic device driver, called LptSimpleDD.Run, that can be used with most printers if you only want to produce draft quality output.

There is also a daisy wheel driver, DaisyDD.Run, that works with several of the daisy wheel printers and produces letter quality text.

Additionally, there are drivers for publication-quality laser printers.

Information on specific supported printing devices and device drivers can be found in the Printing Guide.

THE SPOOLER

The spooler receives print requests from the routing switch for those printing devices it controls.

The spooler makes a request that the Queue Manager place a temporary copy of the document in the queue for the specified printer. Since queues are priority-based, the request may or may not be placed at the end of the queue. When the printer is free, the spooler requests the next job in the queue.

One spooler must be installed on each workstation to which one or more spooled printing devices are attached.

THE QUEUE MANAGER

The Queue Manager is a GPS system service that is also used by other applications such as Batch. The Queue Manager queues and dequeues files as requested.

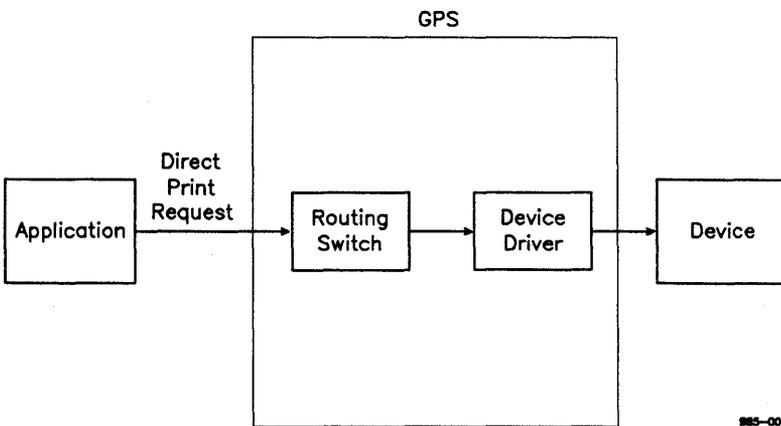
For non-GPS applications, the Queue Manager references a file called Queue.Index to establish queues at installation time. GPS, by contrast, automatically creates queues as needed.

Your application does not need to make requests directly to the Queue Manager.

The GPS Queue Manager has the ability to dynamically establish and remove queues without referencing the Queue.Index file.

For each standalone system or cluster of workstations, only one Queue Manager is installed. Within a cluster of workstations, the Queue Manager is installed at the master workstation.

When the Queue Manager is installed, the number of queues specified must be large enough to accommodate all the queues used by non-GPS applications (those defined in Queue.Indexb), plus those that will be used by the print system.



985-004

Figure 2-4. Direct Printing with GPS

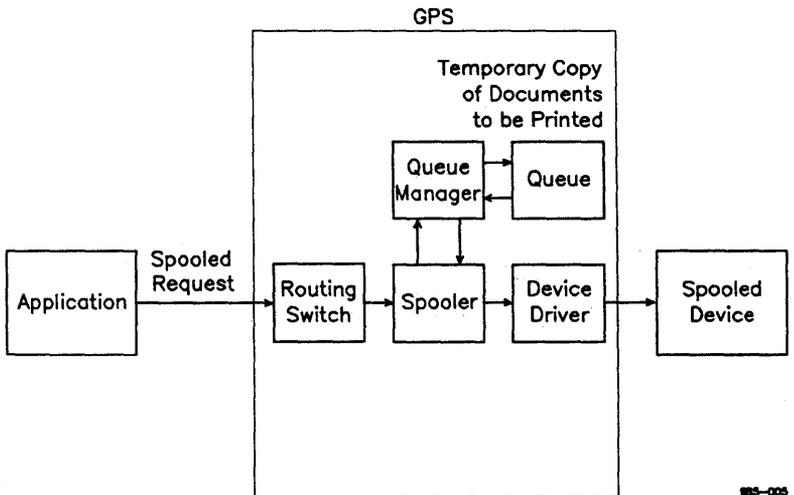
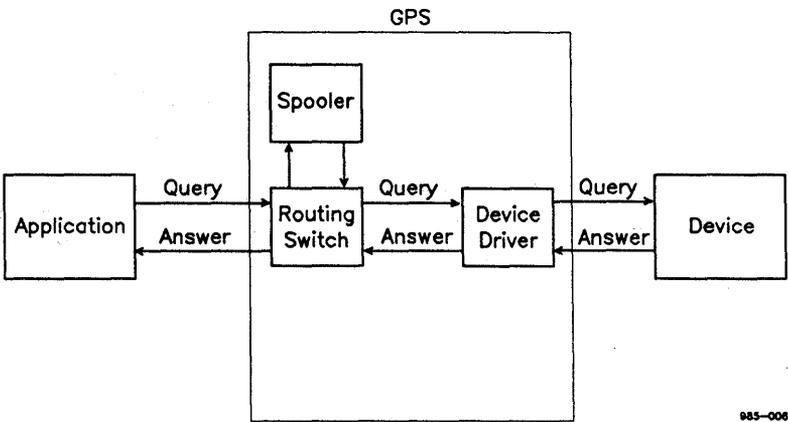
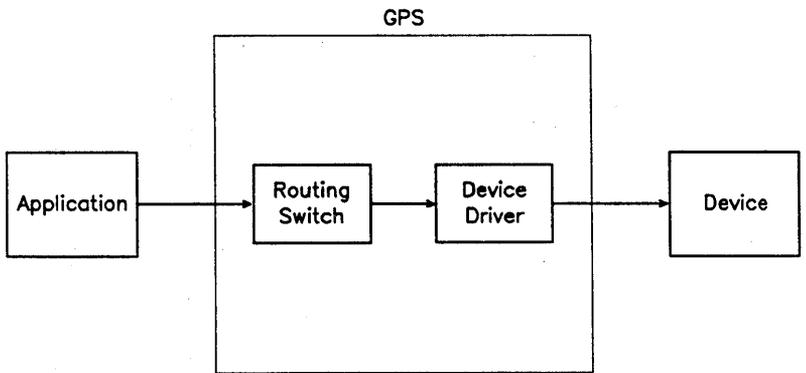


Figure 2-5. Spooled Printing with GPS



985-006

Figure 2-6. Printer Status Queries with GPS



SES-007

Figure 2-7. Issuing Printer Control Commands with GPS

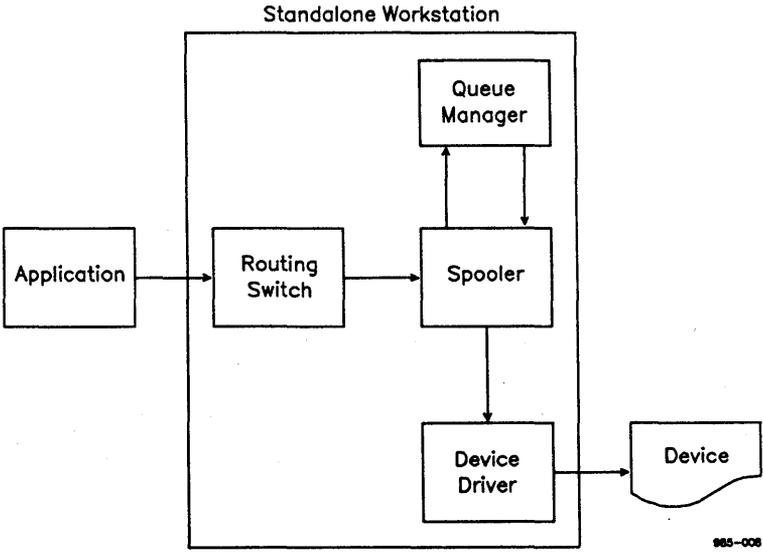


Figure 2-8. Spooled Printing on a Standalone Workstation

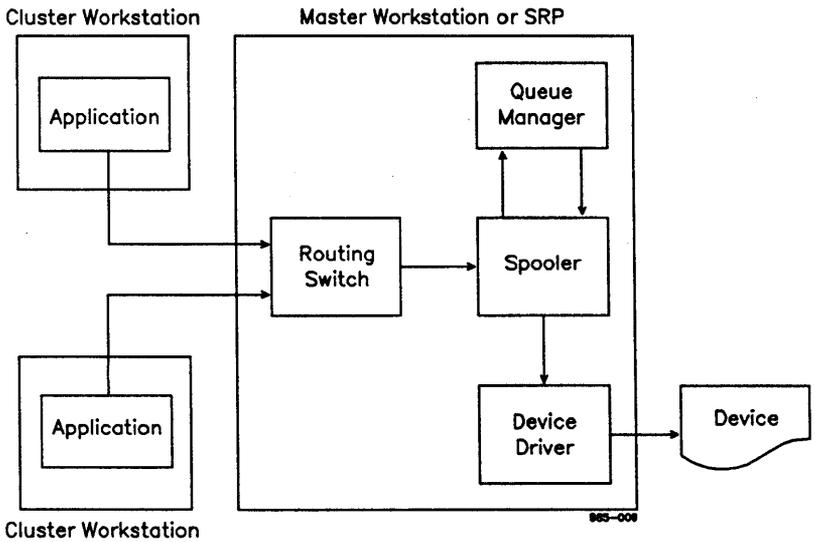


Figure 2-9. Spooled Printing on a Simple Cluster

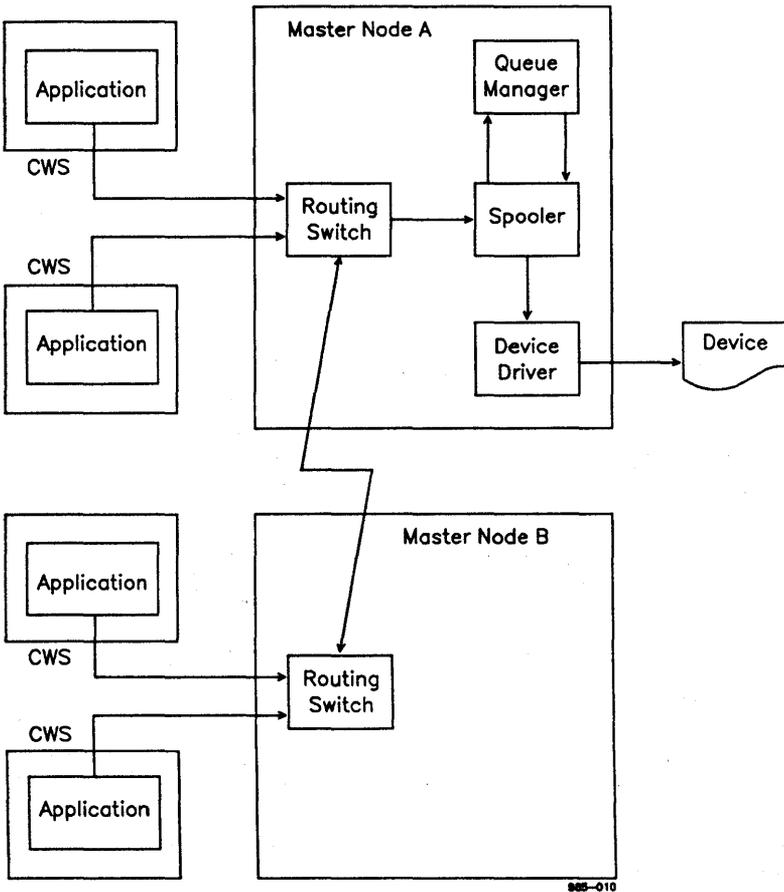


Figure 2-10. Spooled Printing Across the Network

3 USING GPS BYTE STREAMS

The CTOS Sequential Access Method (SAM) provides device-independent calls an application can use to access real devices, such as the video display, keyboard, or printer. These calls send a generic stream of bytes, called a byte stream, through SAM to the specified device.

The simplest way for your application to use the Generic Print System is to print through GPS byte streams (part of the Sequential Access Method).

This method passes ASCII characters to the printing device for printing. The device driver passes the output to the printer interpreting the characters required for simple draft output.

You may want to use this type of printing, for example, if your application creates files that do not contain special formatting, but consists of characters, line feeds, and form feeds.

Using the Sequential Access Method for all kinds of devices is explained in detail in the CTOS Manual. Note that printer byte streams and spooler byte streams as described in the CTOS Manual are not GPS byte streams. Using GPS byte streams is described below.

NOTE

If you want to convert an application not originally written to use the Generic Print System, the most expedient mode of conversion is to print the output currently produced through GPS byte streams. See Appendix A, "Converting Non-GPS Applications," for more information.

GPS_BYTE_STREAMS

To use GPS byte streams when you make the call to `OpenByteStream`, specify the GPS printer name enclosed in square brackets, as shown below:

```
[LaserJet]
```

Note that you do not have to give the printer location; only the printer name is required. In addition, your application does not need to know whether the device is set up for direct or spooled printing. The Generic Print System handles all this.

(You can find out this type of printer information by using the `GetGPSDeviceConfig` call, described in Chapter 7.)

Specifically, the calls used to send output to a GPS device are as follows:

<code>OpenByteStream</code>	Opens the byte stream and specifies the GPS Byte Stream Work Area.
<code>WriteBsRecord</code>	Writes a specified count of bytes to the GPS Byte Stream Work Area.
<code>WriteByte</code>	Writes one byte only to the GPS Byte Stream Work Area.
<code>CloseByteStream</code>	Closes the byte stream.
<code>ReleaseByteStream</code>	Releases abnormal closes and fatal errors.

Note that to set the printing mode with GPS you should use the routing switch request, `SetImageModeGPS`, rather than `SetImageMode` as described in the CTOS Manual.

An example has been included on the distribution disk to show how a simple Copy program might use byte streams to open a file and copy it to a GPS device.

LINKING YOUR PROGRAM TO INCLUDE GPS BYTE STREAMS

You should be sure that when you link your program that the version of `CTOS.lib` and `SAMGen.obj` that you are using includes GPS byte streams. See your Generic Print System Release Notice for more information.

Also note that SAM can be customized to include the capability to handle only those device types you specify. For example, if you intend to use only GPS, you may want to remove entries for printer byte streams and spooler byte streams (both for non-GPS printing devices) that are no longer applicable. See "Building a Customized SAM," in the CTOS Programmer's Guide for more information.

EXAMPLE

A sample program illustrating some of the concepts discussed in this chapter can be found on your distribution disk.

4 USING THE GENERIC PRINT ACCESS METHOD

The Generic Print Access Method (GPAM) is a library of object module procedures (GPAM.lib) provided for use with GPS. GPAM handles communications with the GPS system services for you in the much the same way that byte streams do. In addition, it provides the following:

- generic object module routines that can be translated by each GPS device driver into device-specific formatting commands
- the following character formatting attributes:
 - boldface
 - underscore
 - overstrike
 - color
 - superscript and subscript
 - font
- graphics formatting
- enables you to mix graphics and text on the same page
- controls character placement within the page; for example, you can specify line feed measurements and placement of the character relative to the left margin
- defines printing parameters, such as number of copies and priority

You can specify graphics output with GPAM in one of two ways:

- Use a series of GPAM graphics routines:
 - lines can be specified in any color as solid, dashed, dotted, or dash-dot
 - closed polygons can be specified with separate interior and exterior colors, hollow, solidly filled, or filled with diagonal lines or crosshatching
- Use the GPAM write routine, GPAMWriteRecord, to write out an existing CGM file produced with the Graphics library routine, WriteCGM. (The graphics library is available separately.)

NOTE

You can print graphics and text with GPAM. GPS byte streams can only be used to output text.

With GPAM, as with GPS byte streams, you do not have to give the printer location, only the printer name is required. In addition, your application does not usually need to know whether the device is set up for direct or spooled printing. The Generic Print System handles all this. GPAM data streams can also be directed to a file.

When a device driver receives a GPAM data stream that includes formatting it cannot produce, those commands are ignored and the document is printed according to the capabilities of the device. For example, if a document including text and graphics is sent to a text-only printer, the text is printed and the graphics are not.

Table 4-1 includes a list of the GPAM procedures with a brief description of each.

You can use GPAM in conjunction with calls to GPS. GPS requests are discussed in Chapter 5, "Using GPS Requests."

LINKING WITH GPAM

To use GPAM, you must specify GPAM.lib in the Libraries field of the Linker command form when you link your application.

The linking is done automatically from the [Sys]<sys> directory if GPAM.lib is there.

Table 4-1
GPAM PROCEDURE SUMMARY
(Page 1 of 4)

GPAMBeginBold	Prints all subsequent characters as boldface text.
GPAMBeginBound	Adjusts proportional text generated by an application to conform to the limitations of an output device.
GPAMBeginItalic	Prints all subsequent characters as italic text.
GPAMBeginOverstrike	Indicates that all subsequent text is to be overstruck with a specified character.
GPAMBeginPage	Causes a new physical page to begin and defines a series of attributes about the new page.
GPAMBeginRaster	Indicates raster graphics output is to begin.
GPAMBeginTransparent	Indicates that all subsequent output until GPAMEndTransparent is called is to be passed through to the output device and is not to be processed in any way by the output device driver.
GPAMBeginUnderscore	Indicates that all subsequent text is to be underscored.
GPAMClose	Closes the specified GPAM data stream.

Table 4-1
GPAM PROCEDURE SUMMARY
(Page 2 of 4)

GPAMEndBold	Terminates bold printing.
GPAMEndBound	Terminates bound printing.
GPAMEndItalic	Terminates italic printing.
GPAMEndOverstrike	Terminates overstriking.
GPAMEndRaster	Terminates raster graphics output.
GPAMEndTransparent	Terminates transparent output.
GPAMEndUnderscore	Terminates underscoring.
GPAMLineFeed	Causes the output position to move vertically down by an amount equal to the most recent value set by GPAMSetSLD.
GPAMOpen	Opens a GPAM data stream and specifies all document level parameters, for example number of copies.
GPAMPlaceCharacter	Places a single character on the current line an absolute distance from the left side of the page.
GPAMRelease	Abnormally closes the output stream in the case of error or an unsuccessful GPAMClose.
GPAMReposition	Causes subsequent output to be placed starting at the location specified.

Table 4-1
GPAM PROCEDURE SUMMARY
(Page 3 of 4)

GPAMSetColor	Specifies which color should be used for all subsequent text.
GPAMSetFont	Specifies the font name, its width, and its height.
GPAMSetSLD	Specifies the vertical distance to move for each subsequent GPAMLineFeed or Return (the single line distance).
GPAMSetTranslateTable	Specifies the name of the translate table (12 bytes or less) to use to map internal 8 bit character codes into the codes to be presented to the output device.
GPAMSetVerticalOffset	Used for subscripts and superscripts.
GPAMWhiteSpace	Puts a specified amount of white space at the current position.
GPAMWriteRecord	[text and graphics] Writes the specified bytes to the open byte stream.
GPAMGrBeginGraphics	Used to specify the beginning of any series of GPAM Graphics routines or GPAMWriteRecord routines containing CGM-format graphics information.

Table 4-1
GPAM PROCEDURE SUMMARY
(Page 4 of 4)

GPAMGrColorTable	Sets the colors to be used with graphics routines.
GPAMGrEndGraphics	Terminates the definition of graphics output.
GPAMGrFillColor	Sets the color used by GPAMGrPolygon for interior fills.
GPAMGrHatchIndex	Sets the hatch pattern used by GPAMGrPolygon to fill the figure.
GPAMGrInteriorStyle	Sets the style used by GPAMGrPolygon to fill the figure.
GPAMGrLineColor	Sets the color used by GPAMGrPolyLine.
GPAMGrLineType	Sets the line type used by GPAMGrPolyline.
GPAMGrPerimeterColor	Sets the color used by GPAMGrPolygon to draw the polygon perimeter.
GPAMGrPolygon	Draws a closed polygon.
GPAMGrPolyLine	Draws a sequence of connected lines.

OPENING AND CLOSING A GPAM DATA STREAM

To begin a GPAM data stream, you must use the procedure GPAMOpen, then follow with other GPAM routines. GPAMOpen specifies the device or file to which you want to print and gives print request parameters, for example, document name, time to begin print, number of copies, timeout, and priority.

In effect, GPAM opens a GPS byte stream, then writes data in GPAM format to the specified device driver. The driver translates the generic commands and data is printed as specified. GPAM requires a 300-byte work area and a buffer area from 1024 to 4800 bytes in size.

For normal circumstances, the data stream is closed with the procedure GPAMClose. In the case of an unsuccessful close or a fatal GPS error, you should use GPAMRelease to close the output stream. GPAMRelease does not attempt to write out any partially full buffers.

DATA STREAM STRUCTURE

You can use GPAM procedures to describe your page, line spacing, character placement within the line, and character attributes such as bold, underscore, superscript or subscript.

Once these attributes are set, GPAM data is output similarly to byte stream data, with the procedure GPAMWriteRecord.

A typical sequence that would begin a page, print out some boldface and underscored characters, then change line spacing and print out normal characters might be the following:

GPAMOpen

GPAMBeginPage

GPAMBeginBold

GPAMBeginUnderscore

GPAMWriteRecord

GPAMEndUnderscore

GPAMEndBold

GPAMWriteRecord

GPAMWriteRecord

GPAMClose

A complete example showing construction of a GPAM document is included on the distribution disk.

Once a GPAMBeginGraphics call has been made, all calls must also be Graphics calls, until you end the graphics sequence with a call to GPAMEndGraphics.

When including text and graphics on the same page it is recommended that all text be written in a block, then use GPAMReposition before writing an area of graphics data. The device driver buffers up to a full page of data to handle information.

LEGAL CHARACTERS

Data written to the output stream may contain any of the 256 characters included in the CTOS Standard Character Set. See the CTOS Operating System Manual for a complete list of legal characters.

The Form Feed (0Ch), Carriage Return (0Dh), Horizontal Tab (09h), and Return (0Ah) can all be included as a part of the text written to the output stream. For each, the device driver generates the appropriate action.

Note that return (0Ah) is the preferred method for starting a new line. (0Ch and 0Dh are supported for compatibility with ASCII character sets.) Form feed is not the preferred method for starting a new page; use GPAMBeginPage instead.

Depending on the device driver, the inclusion of some characters, for example, the vertical tab (0Bh), may cause printer action. This should only be included if you really want the printer to space vertically on the page.

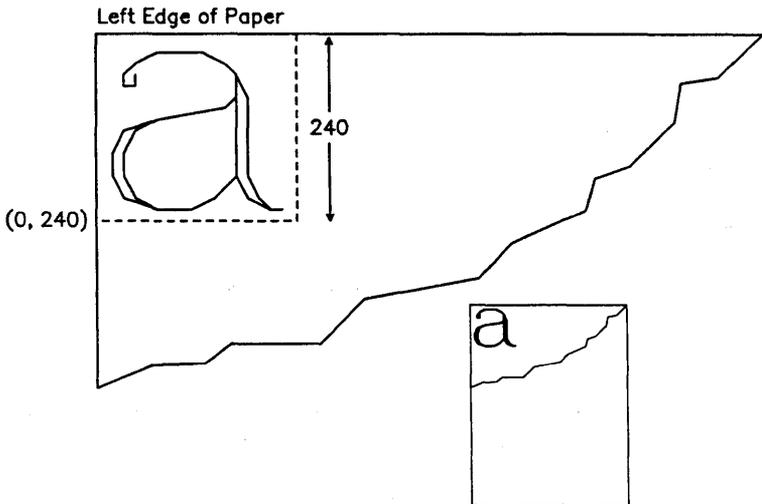
PAGINATION

Page set up is specified with the procedure GPAMBeginPage. At this time you can set parameters such as output bin, special form, portrait or landscape aspect, page length, and page width.

If you have form feeds within any text you write out, the printer will take the appropriate action. The last parameters set for GPAMBeginPage are carried over for that new page.

PRINT POSITION AND CHARACTER PLACEMENT ON THE PAGE

Page coordinates are specified for GPAM in 20ths of a point. The coordinate base specifies the left edge of the character and the base line. For example, the first character on a page would usually appear at the position 0,240. See Figure 4-1.



985-011

Figure 4-1. Character Placement

You can set line feed distance with GPAMSetSLD.

For most line feeds, you should use Return (0Ah). GPAMLinefeed moves the print position down one line without any horizontal movement.

Carriage Return (0Dh) moves the print position back to the beginning of the line without any vertical movement.

The procedure `GPAMReposition` can reposition the print position to anywhere in the page.

Character placement within a line can be set with `GPAMPlaceCharacter`. This routine requires fewer parameters than `GPAMReposition` and is, therefore, more efficient to use if no change in vertical position is desired.

ASSIGNING TEXT ATTRIBUTES

You can specify that text be boldface, underscored, superscript, subscript, a specified color or font, or overstruck by a specified character.

With bold, overstrike, and underscore, the attribute can be turned on and off, for example, `GPAMBeginBold/GPAMEndBold`.

Superscript and subscript are set with the procedure `GPAMSetVerticalOffset`, which moves the current print position up or down a fraction of the current line feed distance (set with `GPAMSetSLD`). The distance moved depends on the device driver. If you choose not to use the default value, however, you can set the superscript or subscript to be up to 8 times the value of the default distance.

To terminate superscript or subscript you must use `GPAMSetVerticalOffset` again to return the offset to 0 (the baseline). You do this by assigning the appropriate positive (+) or negative (-) value.

`GPAMSetColor` sets the color of subsequent text. The colors are defined with `GPAMColorTable`.

GPAMSetFont specifies a font name, width, and height. The font height is equivalent to the font point size. If you choose not to use the default width of a font, you can set width and height separately.

TIMEOUTS

If you want to get control back during your GPAM output session (to issue a printer status request for example), you can also set a timeout with GPAMOpen. When the device driver times out, it cancels the current command.

Generally, when you get a timeout from the device driver, you should immediately check printer status, then, to resume processing, reissue the command that timed out.

To avoid hanging-up the system, make a timeout call as a general caution after each GPAM call, for example, after GPAMBeginBold, after GPAMEndBold, and so on.

Note that with GPAMWriteRecord, the current command is not cancelled. A number of bytes may have been already transferred to the device.

If you get a timeout after a GPAMWriteRecord, you should look at the number of bytes returned to determine where in the write procedure the timeout occurred. Then you can reissue the call specifying a write for the number of remaining bytes to print.

SENDING NON-GPAM COMMANDS TO THE PRINTING DEVICE

Note that if you want to send device-specific commands not supported with GPAM, a special procedure (`GPAMBeginTransparent`) is provided so that the commands can be sent within your GPAM data stream.

`GPAMBeginTransparent` sets GPAM's data mode to transparent. In that case, subsequent data from `GPAMWriteRecord` is passed as is to the printing device without any translation by the GPS device driver. To reset GPAM's data mode back to normal, call the `GPAMEndTransparent` routine.

USING GPAM FOR GRAPHICS

With GPAM you can easily include both figures and alphanumeric characters on the same page. You can use GPAM procedures to write all your text, then use `GPAMReposition` to change the print position and write out the figures.

When the page is printed out, the device driver buffers the entire page, then prints it out line-by-line, with all text and graphics.

The printing of all graphics must be preceded by a call to `GPAMGrBeginGraphics`, which should itself be preceded by a call to `GPAMReposition`. Subsequent calls may be either of the following:

- GPAM graphics routines
- `GPAMWriteRecord` routines that write out an existing CGM file originally created using the `Graphics.lib` routine, `WriteCGM`

End a sequence of calls to graphics routines with a call to `GPAMGrEndGraphics`.

GRAPHICS POSITIONING

Whenever a call to GPAMGrBeginGraphics is made, the picture is placed by default with its upper-left corner at the current print position. To change this, you must make a call to GPAMReposition before making the GPAMGrBeginGraphics call.

NOTE

Pictures cannot cross page boundaries.

SCALING

When the call to GPAMGrBeginGraphics is made, the width and height of the graphics area is specified. The device driver automatically scales graphics output to fit the space designated.

THE GRAPHICS ROUTINES

With graphics routines you can set a color palette and create lines or closed polygons with a variety of different attributes.

GPAMGrColorTable sets the color palette. The color you define in the palette can be subsequently selected when you draw lines or polygons.

Lines can be specified in any color as solid, dashed, dotted, or dash-dot. Use GPAMGrPolyLine to specify the line coordinates. Up to 8191 points can be specified.

GPAMGrLineColor specifies the color for subsequent lines, from the colors array specified in GPAMGrColorTable.

GPAMGrLineType specifies the line as one of the following:

solid	_____
dashed	-----
dotted
dash dot	-.-.-.-
line dash dash	_-_-_-_-
line dash	__-__-__-
line blank	__ _ _ _
line blank blank	_ _ _ _

Some devices may use other patterns for some of the line types defined above. You should experiment with your output device to determine what line types it supports.

Closed polygons can be specified with separate interior and exterior colors, hollow, solidly filled, or filled with diagonal lines or crosshatching.

Use GPAMGrPolygon to specify the array of coordinates to be connected to specify the boundaries of the polygon.

The GPAMGrInteriorStyle routine sets whether the subsequent polygons have a visible border and whether the interior fill will be hollow, solid, or crosshatched.

If a perimeter exists, GPAMGrPerimeterColor selects the color of the perimeter line for any subsequent polygons.

GPAMGrHatchIndex sets interior hatching for either a narrow or wide diagonal or crosshatching.

GPAMGrFillColor sets the interior colors for the polygon.

A typical sequence of graphics routines might be as follows:

```
GPAMOpen
  GPAMReposition
  GPAMGrBeginGraphics
    GPAMGrSetColorTable
    GPAMGrInteriorStyle
    GPAMGrHatchIndex
    GPAMGrPolygon
  GPAMGrEndGraphics
GPAMClose
```

A sample GPAM graphics routine is included on the distribution disk.

USING CGM

If you want to print files that were created using the Graphics.lib routine (WriteVDM), use GPAMWriteRecord.

A typical sequence of calls would be as follows:

```
GPAMRepostion
GPAMGrBeginGraphics
  GPAMWriteRecord
GPAMGrEndGraphics
```


5 USING GPS REQUESTS

Most application systems use GPAM or GPS byte streams to print. If, however, you want to add status and control functions to your application, or if you want more direct control over the printing process, you can use the services of the Generic Print System by making direct requests of the routing switch.

A large portion of the routing switch's function is the generation, collection, presentation and control of GPS data. The information about each GPS device is collected and maintained by the routing switch installed at the same workstation as that device. This routing switch disseminates the information to the other routing switches in the cluster and network. Data describing GPS devices throughout the cluster and network is provided to an application by the nearest routing switch.

You can use routing switch operations, as described below, to enhance your application's printing facilities:

- to get a list of printing devices that can be legally specified, or to get the name of the user's designated default device
- to check and report on the status of any printing device or print request
- to send printer control commands, such as pause and restart, from your application to the printing device

- to add a check for such conditions as printing speed, whether the printing device can handle graphics output, whether the device is spooled, or whether there is enough room on the spooler volume for the spooled print request you are about to make

You can use routing switch requests to print instead of using byte streams or GPAM to copy information.

Table 5-1 gives a summary and description of the routing switch requests. The requests are described in detail in Chapter 7, "GPS Routing Switch Requests."

MAKING A PRINT REQUEST WITH GPS OUTPUT FUNCTIONS

Most applications use GPS byte streams or GPAM to print, rather than communicating directly with any of the GPS system services. Such applications use only the GPS status and printer control requests, discussed below in "Status Functions," and "Controlling the Printing Device."

If you should use direct GPS requests to print, a typical sequence would be as follows:

OpenGPSFile

SetGPSParams

WriteGPSFile

CloseGPSFile

Whenever you do an OpenGPSFile you should also use the request SetGPSParams to control queuing and printing of the document. SetGPSParams sets such parameters as number of copies, start and stop location, and priority.

Table 5-1
SUMMARY OF ROUTING SWITCH REQUESTS
(Page 1 of 2)

CloseGPSFile	Terminates an output session, closes the open device driver and releases associated hardware.
DeleteGPSJob	Causes a document queued at a device to be deleted.
GetGpsDefaultPrinter	Used by application programs to select the default GPS device as specified in the logged in user file.
GetGPSDevHistory	Provides a list of documents that have been printed by a particular device, indicates when the printing started and finished, and the status of the document upon termination.
GetGPSDeviceConfig	Returns configuration information about a particular device.
GetGPSDocInfo	Used to query GPS about the status of documents that are printing or are waiting to be printed at a selected device.
GetGPSNetList	Returns a list of all GPS output devices accessible from the workstation running the application.
GetGpsSplVolStatus	Returns the number of 512-byte pages remaining on the spooling volume.

Table 5-1
SUMMARY OF ROUTING SWITCH REQUESTS
(Page 2 of 2)

GetGPSStatus request	Returns status information for a particular output device.
OpenGPSFile	Opens a GPS device for output.
ProvideGPSPassword	Provides a password so that printing of a password protected document, printed in security mode can start.
QueueGPSFile	Allows an application to specify the name of add a document file that is to be added to a spooled generic device's queue.
SetGPSControl	Provides the mechanism for applications to send control commands to generic print devices.
SetGPSParams	Provides information to the GPS system about a job that has just been opened with the OpenGPSFile request.
SetGPSRestart	Provides the mechanism for applications to send restart control commands to generic output devices that have previously been paused.
SetImageModeGPS	Sets normal, image, or binary mode for a GPS device.
WriteGPSFile	Transfers a number of bytes from memory to the printer.

Whenever you do an `OpenGPSFile` you should also use the request `SetGPSParams` to control queuing and printing of the document. `SetGPSParams` sets such parameters as number of copies, start and stop location, and priority.

If you specify security mode for printing when you call `SetGPSParams`, the printing device pauses before printing to wait for a password to be entered. To supply the password, your application should use `ProvideGPSPassword`.

SPOOLED OR DIRECT PRINTING

In most cases, your application does not need information about whether the device is a direct or spooled device in order to print. If you want to know this information, you can use `GetGPSDevConfig` to query GPS about the device.

You should be aware that requests for multiple copies cannot be processed by a direct printing device. However, since GPS always tries to do the best it can with your request, it does accept such a request to a direct device and prints one copy only.

For spooled printing devices, as an alternative to the sequence outlined above, you can use `QueueGPSFile`. With this request the normal GPS process of queuing a file by copying a temporary file to the spooler volume is bypassed and the file is opened directly by the spooler. However, the file must be accessible from the spooler for the device specified. For example, a file on a local file system could not be queued for a device on the master workstation with `QueueGPSFile`, because the spooler at the master cannot open the file.

PRINTING MODE

You can set the printing mode using `SetImageModeGPS`. Printing mode controls the way the printing device interprets line control characters and escape sequences, thereby controlling your ability to pass printer specific escape sequences through to the printing device with your output stream. See the entry for `SetImageModeGPS` in Chapter 7, "GPS Routing Switch Requests," for more information on printing modes.

SPECIFYING A PRINTING DEVICE

You can print to any GPS device by specifying the device name when you make your print request.

If you want to provide a list of printers for the user to choose from you can use `GetGPSNetList`. This request returns all printing devices accessible from the workstation from which it is sent. The number received is limited by the size of the buffer you specify.

If you want to always print to a specified default printing device, you can use `GetGPSDefaultPrinter` to find out what printing device the user has chosen for his or her default device.

`GetGPSDefaultPrinter` is an object module procedure and requires that the user has made the following entry in his User Configuration file:

```
:GPSDefaultPrinter:PrinterName
```

where `PrinterName` is the GPS name assigned to the printing device at installation.

GPS STATUS AND CONTROL FUNCTIONS

STATUS FUNCTIONS

With the GetGPSStatus request your application can request status on any specific GPS device. The information returned includes the state of the device (for example, paused for font change), and the name and owner of the document currently printing.

With GetGPSDocInfo you can get additional information about a specific print request, such as the priority of the job, time to print, current page printing, and whether or not the job was specified to be printed in security mode.

For a specific device, GetGPSDevHistory returns a limited list of completed documents and gives such information as the time the job completed. The length of this list varies depending on the capabilities of the device driver used by the device. Typically it lists the last five jobs.

CONTROLLING THE PRINTING DEVICE

For most printer control commands, you use the SetGPSControl request. With this you can pause the printing device temporarily, cancel the current job, or pause the device for alignment. To restart, use SetGPSRestart.

If the device must pause for font or forms changes, this is done automatically by GPS. The font or form to be used is set with SetGPSParams. To restart after an automatic pause, use SetGPSRestart.

You can also delete a queued job from the print queue with DeleteGPSJob.

EXAMPLE

There is an example of how to use Generic Print System requests on the distribution disk.

6 GPAM PROCEDURES

This chapter gives a detailed description of each of the procedures included in the Generic Print Access Method (GPAM) library. See Chapter 4, "Using the Generic Print Access Method," for an overview and examples of GPAM.

Table 6-1
GPAM PROCEDURE SUMMARY
(Page 1 of 4)

GPAMBeginBold	Prints all subsequent characters as boldface text.
GPAMBeginBound	Adjusts proportional text generated by an application to conform to the limitations of an output device.
GPAMBeginItalic	Prints all subsequent characters as italic text.
GPAMBeginOverstrike	Indicates that all subsequent text is to be overstruck with a specified character.
GPAMBeginPage	Causes a new physical page to begin and defines a series of attributes about the new page.
GPAMBeginRaster	Indicates raster graphics output is to begin.
GPAMBeginTransparent	Indicates that all subsequent output until GPAMEndTransparent is called is to be passed through to the output device and is not to be processed in any way by the output device driver.
GPAMBeginUnderscore	Indicates that all subsequent text is to be underscored.
GPAMClose	Closes the specified GPAM data stream.

Table 6-1
GPAM PROCEDURE SUMMARY
(Page 2 of 4)

GPAMEndBold	Terminates bold printing.
GPAMEndBound	Terminates bound printing.
GPAMEndItalic	Terminates italic printing.
GPAMEndOverstrike	Terminates overstriking.
GPAMEndRaster	Terminates raster graphics output.
GPAMEndTransparent	Terminates transparent output.
GPAMEndUnderscore	Terminates underscoring.
GPAMLineFeed	Causes the output position to move vertically down by an amount equal to the most recent value set by GPAMSetSLD.
GPAMOpen	Opens a GPAM data stream and specifies all document level parameters, for example number of copies.
GPAMPlaceCharacter	Places a single character on the current line an absolute distance from the left side of the page.
GPAMRelease	Abnormally closes the output stream in the case of error or an unsuccessful GPAMClose.
GPAMReposition	Causes subsequent output to be placed starting at the location specified.

Table 6-1
GPAM PROCEDURE SUMMARY
(Page 3 of 4)

GPAMSetColor	Specifies which color should be used for all subsequent text.
GPAMSetFont	Specifies the font name, its width, and its height.
GPAMSetSLD	Specifies the vertical distance to move for each subsequent GPAMLineFeed or Return (the single line distance).
GPAMSetTranslateTable	Specifies the name of the translate table (12 bytes or less) to use to map internal 8 bit character codes into the codes to be presented to the output device.
GPAMSetVerticalOffset	Used for subscripts and superscripts.
GPAMWhiteSpace	Puts a specified amount of white space at the current position.
GPAMWriteRecord [text and graphics]	Writes the specified bytes to the open byte stream.
GPAMGrBeginGraphics	Used to specify the beginning of any series of GPAM Graphics routines or GPAMWriteRecord routines containing CGM-format graphics information.

Table 6-1
GPAM PROCEDURE SUMMARY
(Page 4 of 4)

GPAMGrColorTable	Sets the colors to be used with graphics routines.
GPAMGrEndGraphics	Terminates the definition of graphics output.
GPAMGrFillColor	Sets the color used by GPAMGrPolygon for interior fills.
GPAMGrHatchIndex	Sets the hatch pattern used by GPAMGrPolygon to fill the figure.
GPAMGrInteriorStyle	Sets the style used by GPAMGrPolygon to fill the figure.
GPAMGrLineColor	Sets the color used by GPAMGrPolyLine.
GPAMGrLineType	Sets the line type used by GPAMGrPolyline.
GPAMGrPerimeterColor	Sets the color used by GPAMGrPolygon to draw the polygon perimeter.
GPAMGrPolygon	Draws a closed polygon.
GPAMGrPolyLine	Draws a sequence of connected lines.

GPAMBeginBold (pGPAMWA) : ErcType

DESCRIPTION

GPAMBeginBold indicates that all subsequent text is to be printed as bold characters. Bold printing continues until turned off by **GPAMEndBold**.

Bold printing can be specified in combination with overstriking and/or underscoring.

You can change the font with **GPAMSetFont** while bold printing is active without resetting **GPAMBeginBold**.

A call to **GPAMBeginBold** while bold printing is active is ignored by the output device driver.

PROCEDURAL INTERFACE

GPAMBeginBold (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMBeginBound (pGPAMWA, WXLEFT, WXRright, bJustify
: ErcType

DESCRIPTION

GPAMBeginBound and GPAMEndBound enables the device driver to make adjustments in received proportional text to account for device-specific limitations.

Applications that generate proportional text use the Font Service to allocate space for each character in a line based on the width information found in the font database. Some printers have width information different from that used by the application because of limitations in the printer's font options.

GPAMBeginBound specifies a pair of X-coordinates, between which the device driver is requested to place all subsequent characters. The third parameter specifies the type of justification to take place.

The justification requested by GPAMBeginBound does not occur until the matching GPAMEndBound operation occurs.

Bounding operations are intended to bracket a string of text produced with GPAMWriteRecord and GPAMWhiteSpace. The text specified by the white space operations may be adjusted to suit the requirements of justification.

Any operation causing a change in the current Y-coordinate immediately causes a GPAMEndBound operation.

PROCEDURAL INTERFACE

GPAMBeginBound (pGPAMWA, wXLeft, wXRight,
bJustify) : ErcType

where

PGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wXLeft is the leftmost position (in 1440ths/inch units), between which microjustification is to be applied.

wXRight is the rightmost position (in 1440ths/inch units), between which microjustification is to be applied.

bJustify is a byte variable with 4 possible values:

- 0 Left justify the text against wXLeft
- 1 Right justify the text against wXRight
- 2 Left and right justification between wXLeft and wXRight
- 3 Center text between wXLeft and wXRight

GPAMBeginItalic (pGPAMWA) : ErcType

DESCRIPTION

GPAMBeginItalic causes all subsequent text to be printed as italic characters. Italic printing continues until turned off by **GPAMEndItalic**.

Italic printing may be specified in combination with bold, overstriking, and/or underscoring.

Nested calls to **GPAMBeginItalic** have no defined meaning. Thus, a call to **GPAMBeginItalic** while italic printing is already active, is ignored by the device driver.

GPAMSetFont can be used to change the font while italic printing is active.

PROCEDURAL INTERFACE

GPAMBeginItalic (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte work area for use by GPAM procedures.

GPAMBeginOverstrike (pGPAMWA, bBypass, bChar) :
ErcType

DESCRIPTION

GPAMBeginOverstrike indicates that all subsequent text is to be overstruck with the specified character. Overstriking continues until turned off by **GPAMEndOverstrike**.

You can specify whether or not spaces are to be overstruck.

GPAMBeginOverstrike can be called while overstriking is already in progress, which causes any new bypass conditions to replace the previous ones. The next (third) call to **GPAMEndOverstrike** terminates overstriking; it does not revert back to the earlier bypass conditions.

Overstriking can be specified in combination with bold and/or underscoring. Note that the bypass conditions on overstriking and underscoring are applied independently of each other.

GPAMSetFont may change the font while overstriking is active.

Subscripts and superscripts (created via use of **GPAMSetVerticalOffset**) are overstruck in the baseline position.

For proportionally spaced text, the overstrike character is centered at the same point as the overstruck character.

PROCEDURAL INTERFACE

GPAMBeginOverstrike (pGPAMWA, bBypass, bChar) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

bBypass is 0 to indicate overstrike all character positions except blanks.

is 1 to indicate overstrike all character positions.

bChar is the character to be used for overstriking. A bChar value of 00h results in use of the printer's default overstrike character.

GPAMBeginPage (pGPAMWA, pbParams, cbParams) :
 ErcType

DESCRIPTION

GPAMBeginPage causes a new physical page to begin and defines a series of attributes about the new page.

You must precede your first page of output with GPAMBeginPage.

With GPAMBeginPage you can specify the form name to be used for subsequent printing. For devices where manual intervention is needed to change the form, the output device driver creates a status entry indicating that a change to form FormName is required, and then pauses to wait for a Resume command to be sent via the SetGPSRestart request. If no form name is specified, then the current form is used.

GPAMBeginPage also specifies whether the operator must align the form. If forms alignment is specified, the output device driver prints the page contained in the file FormName.Align.

Alignment pages are single-page GPAM documents and can contain any of the GPAM formatting controls except for forms name and forms alignment. They are the first page of a document. A status entry is made indicating that forms alignment is needed. The output device driver then pauses until it is restarted by SetGPSRestart. If SetGPSRestart indicates resume, then the output device driver proceeds. If SetGPSControl indicates align forms, the output device driver reprints the page and pauses again as above.

When a process requests restarting of a printer at a given page number (via SetGPSRestart), the most recent such declarations are used to invoke the form name and/or forms alignment routines. Once loading of the desired form and/or alignment is confirmed, printing begins at the desired page number. Alignment pages are not counted as numbered pages for purposes of positioning the document during a restart.

PROCEDURAL INTERFACE

GPAMBeginPage (pGPAMWA, pbParams, cbParams) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

pbParams

cbParams describes an area containing the new parameters to use for this page. If cbParams is 0, then the parameter settings from the previous page are used. If cbParams is 0 and this is the first page, the defaults listed below are used.

The Params data area has the following structure:

DESCRIPTION

0	bSides	1	<p>is 0 to indicate use default setting.</p> <p>is 1 to indicate print on one side.</p> <p>is 2 to indicate print on both sides.</p>
1	fStagger	1	<p>is TRUE to indicate that the page should be staggered in the output bin.</p> <p>is FALSE to indicate that the page should be placed in the same position in the output bin as the previous page (that is, not staggered). This is the default.</p>
2	wBinIn	2	<p>is the number of the input bin to be used for sheet feeder selection. A value of 0 indicates use the default bin. This is the default.</p>
4	sbFormName	13	<p>is the form name to be used. If the size is 0, then the default form is used. The default is to use the default form.</p>

-
- | | | | |
|----|-----------|----|---|
| 17 | fAlign | 1 | <p>is TRUE to indicate operator forms alignment is needed before printing this page.</p> <p>is FALSE to indicate that operator forms alignment is not needed. This is the default.</p> |
| 18 | bAspect | 1 | <p>is 0 to indicate use default aspect, which is the portrait mode. This can be overridden with WPgLength and WPgWidth.</p> <p>is 1 to indicate use landscape aspect (where the main axis runs the width of the page).</p> <p>is 2 to indicate use portrait aspect (where the main axis runs the length of the page).</p> |
| 19 | sbPageNum | 13 | <p>is the page number of this page and is used during restart. If the size is 0, then this page is assigned a number equal to its ordinal position in the document (this is the default).</p> |

It is recommended that processes using strings as page numbers should explicitly set this value for all pages.

- | | | | |
|----|-----------|---|---|
| 45 | bQuality | 1 | is 0 to indicate use default setting.

is 1 to indicate draft /data processing quality.

is 2 to indicate use memo quality.

is 3 to indicate use letter quality. |
| 46 | wPgLength | 2 | is the length of the page in 1440ths of an inch. Zero is the default and means to use the device driver's default length. |
| 48 | wPgWidth | 2 | is the width of the page in 1440ths of an inch. Zero is the default and means to use the device driver's default width. |

GPAMBeginTransparent (pGPAMWA) : ErcType

DESCRIPTION

GPAMBeginTransparent indicates that all subsequent output is to be passed through to the output device and is not to be processed in any way by the output device driver. Transparent output is turned off with a call to **GPAMEndTransparent**.

GPAMWriteRecord should be the only GPAM routine called between **GPAMBeginTransparent** and **GPAMEndTransparent**.

NOTE

This routine is intended for advanced functions such as sending programs to programmable devices, and may cause the GPAM output stream to lose its device independence.

PROCEDURAL INTERFACE

GPAMBeginTransparent (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMBeginUnderscore (pGPAMWA, bBypass,
bUnderscore) : ErcType

DESCRIPTION

GPAMBeginUnderscore indicates that all subsequent text is to be underscored. Underscoring continues until turned off by **GPAMEndUnderscore**.

Single or double underscoring can be selected.

To indicate whether or not spaces are to be underscored, one of two bypass conditions can be selected. Note that the bypass conditions on overstriking and underscoring are applied independently of each other.

GPAMBeginUnderscore may be called while underscoring is already in progress, which causes any new bypass conditions to replace the previous ones. The next call to **GPAMEndUnderscore** terminates underscoring; it does not revert back to the earlier parameter values.

GPAMSetFont may change the font while underscoring is active.

Subscripts and superscripts (created with **GPAMSetVerticalOffset**) are underscored in the baseline position.

For proportionally spaced text, underscoring is adjusted so as to stay within the horizontal boundaries of the text.

PROCEDURAL INTERFACE

GPAMBeginUnderscore (pGPAMWA, bBypass,
bUnderscore) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

bBypass is 0 to indicate underscore all character positions except blanks.

is 1 to indicate underscore all character positions.

bUnderscore is 1 to indicate single under-scoring.

is 2 to indicate double under-scoring.

GPAMClose (pGPAMWA) : ErcType

DESCRIPTION

GPAMClose closes the GPAM output stream identified by the memory address of the GPAM work area. GPAMClose writes any partially filled buffers and waits for all write operations to complete.

In the case of spooled output this means that the output has all been successfully spooled to disk; it does not indicate that printing or plotting has completed.

After calling GPAMClose the process can reuse the GPAM work area and buffer area. If an error occurs during GPAMClose, then the output stream is closed and a status code is returned.

PROCEDURAL INTERFACE

GPAMClose (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMEndBold (pGPAMWA) : ErcType

DESCRIPTION

GPAMEndBold terminates bold printing.

PROCEDURAL INTERFACE

GPAMEndBold (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMEndBound (pGPAMWA) : ErcType

DESCRIPTION

GPAMEndBound terminates bound printing, causing the justification in the preceding **GPAMBeginBound** operation to be applied.

PROCEDURAL INTERFACE

GPAMEndBound (pGPAMWA)

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMEndItalic (pGPAMWA) : ErcType

DESCRIPTION

GPAMEndItalic terminates italic printing.

PROCEDURAL INTERFACE

GPAMEndItalic (pGPAMWA)

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMEndOverstrike (pGPAMWA) : ErcType

DESCRIPTION

GPAMEndOverstrike terminates overstriking.

PROCEDURAL INTERFACE

GPAMEndOverstrike (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMEndTransparent (pGPAMWA) : ErcType

DESCRIPTION

GPAMEndTransparent terminates transparent output.

PROCEDURAL INTERFACE

GPAMEndTransparent (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMEndUnderscore (pGPAMWA) : ErcType

DESCRIPTION

GPAMEndUnderscore terminates underscoring.

PROCEDURAL INTERFACE

GPAMEndUnderscore (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMLineFeed (pGPAMWA) : ErcType

DESCRIPTION

GPAMLineFeed causes the output position to move vertically down by an amount equal to the most recent value set by GPAMSetSLD. No horizontal movement takes place.

NOTE

Return causes both CarriageReturn and GPAMLineFeed. It is the preferred method.

LineFeed is a one-byte control in IBM DCA Final Form Text. It has been included here so that documents coming in from IBM through a gateway process can retain the semantics of vertical motion only.

PROCEDURAL INTERFACE

GPAMLineFeed (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

```
GPAMOpen (pGPAMWA, sGPAMWA, pbFileSpec,  
          cbFileSpec, pbPassword, cbPassword,  
          pBufferArea, sBufferArea, pbParams,  
          cbParams) : ErcType
```

DESCRIPTION

GPAMOpen must be the first GPAM procedure called.

This routine specifies all document level parameters, including the following:

- identifier of desired device or device class
- number of copies desired

You can avoid the potential delay that occurs with spooled printing when an entire temporary file is created before printing begins, by specifying that if the printing device is free, you want printing or plotting to start immediately. Note that this option is acted upon by the spooler and not by the output device driver.

PROCEDURAL INTERFACE

```
GPAMOpen (pGPAMWA, sGPAMWA, pbFileSpec,  
          cbFileSpec, pbPassword, cbPassword,  
          pBufferArea, sBufferArea, pbParams,  
          cbParams) : ErcType
```

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

sGPAMWA is 300 (this may change in subsequent versions of GPAM and GPS).

pbFileSpec
cbFileSpec describe a device or file specification.

pbPassword
cbPassword describe a device, volume, directory or file password.

pBufferArea
sBufferArea describe a memory area provided for exclusive use of GPAM procedures. This area must be at least 1024 bytes and word-aligned. If the buffer is greater than 4800 bytes, then only the first 4800 bytes are be used.

pbParams
cbParams describe an area containing the parameters to use for this document. If cbParams is 0, then all default parameter settings will be used.

The Params data area has the following structure:

DESCRIPTION

0	sbDocName	93	is the name of the document. (This is the name that displays in the Print Manager.) A size of zero means to use the default, which is no DocName.
93	sbUserName	31	is the client user name. A size of zero means to use the default, which is no UserName.
124	fSecurity	1	is TRUE to indicate the output should be printed in security mode. The output device driver makes a status entry indicating that a password is needed, and then pauses.

is FALSE to indicate the output should not be printed in security mode. This is the default.

-
- 125 fSupressPage 1 is TRUE to indicate that the output device driver should not do a form feed prior to commencing output.
- is FALSE to indicate that the output device driver should do a form feed prior to commencing output. This is the default.
- 126 fSuppressBanner 1 is TRUE to indicate that the output device driver should not print a banner or the notice file.
- is FALSE to indicate that the output device driver should print a banner and the notice file. This is the default.
- 129 qTimeCreated 4 is the date and time the output was created. A zero value means to use the default, which is the current date and time. (Date/time formats are discussed in the CTOS Operating System Manual.)

-
- 133 qReqOutputStartTime 4
is the date/time after which outputting can begin. A zero value indicates as soon as possible. This is the default. (Date/time formats are discussed in the CTOS Operating System Manual.)
- 137 wCopies 2
is the number of copies to be output. This parameter is used only for spooled printing. For direct printing, only one copy of the document is produced regardless of this parameter's value. Default is one copy.
- 39 wTimeout 2
is the time, in tenths of a second, that GPAM will wait for a response from GPS to a write request. 0FFFFh indicates no timeout. This is the default.
- 141 fSingleSheet 1
is TRUE to cause pauses between pages (single sheet feed).

is FALSE to indicate that continuous feed paper is being used. This is the default.

142 wTimeToPrint 2 is the approximate time in minutes for the job to print. Default is zero.

144 bPriority 1 is the queue entry priority. (See the CTOS Operating System Manual for more information.) Default is five.

GPAMPlaceCharacter (pGPAMWA, bChar, wPosition) :
ErcType

DESCRIPTION

GPAMPlaceCharacter places a single character on the current line an absolute distance from the left side of the page.

The same rules about special characters that apply to **GPAMWriteRecord** also apply to **GPAMPlaceCharacter**.

PROCEDURAL INTERFACE

GPAMPlaceCharacter (pGPAMWA, bChar, wPosition) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

bChar is the character to be output.

wPosition is the position in 1440ths of an inch relative to the left side of the page where the character should be placed.

GPAMRelease (pGPAMWA) : ErcType

DESCRIPTION

GPAMRelease abnormally closes the output stream identified by the GPAM work area. Unlike GPAMClose, GPAMRelease does not write out partially full buffers. GPAMRelease should be used following an unsuccessful GPAMClose, or following a fatal GPS error.

PROCEDURAL INTERFACE

GPAMRelease (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMReposition (pGPAMWA, wHorizontal, wVertical) :
ErcType

DESCRIPTION

GPAMReposition specifies a starting location for subsequent output in 1440ths of an inch relative to the left and top edges of the current physical page. Allowable values are 0 to 65534 (45.5 inches). A value of 0FFFFh indicates use the current position.

GPAMReposition is generally used only to place a figure on the page before a call to GPAMBeginGraphics, or after a sequence of graphics calls.

When output is repositioned to a point that is not flush with the left edge of the physical page, subsequent lines (caused by Return, CarriageReturn, FormFeed, or a call to GPAMBeginPage) start at the indented position. Thus GPAMReposition has the effect of establishing a new left margin.

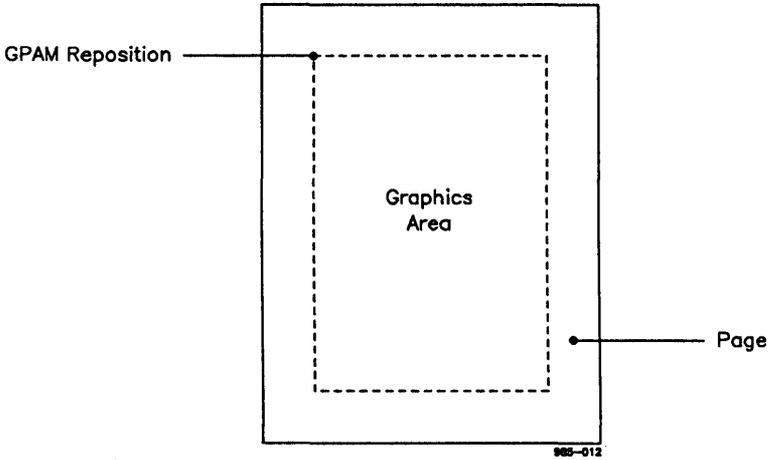


Figure 6-1. Repositioning

Repositioning is only allowed within the current physical page. The output device driver may cause a new page to begin when the selected device cannot accommodate the amount of output provided on a single page.

Not all device drivers support repositioning.

PROCEDURAL INTERFACE

GPAMReposition (pGPAMWA, wHorizontal, wVertical) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wHorizontal is the location to which repositioning will be done. It is specified in 1440ths of an inch, relative to the left edge of the current physical page. Value 0FFFFh indicates use the current horizontal position.

wVertical is the location to which repositioning will be done. It is specified in 1440ths of an inch, relative to the top edge of the current physical page. Value 0FFFFh indicates use the current vertical position.

GPAMSetColor (pGPAMWA, wColor) : ErcType

DESCRIPTION

GPAMSetColor specifies which color should be used for all subsequent text. On devices requiring operator action to change colors, the output device driver makes a status entry indicating that a change to color wColor is needed and then pauses to wait for a restart command given with GPSRestart.

PROCEDURAL INTERFACE

GPAMSetColor (pGPAMWA, wColor) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wColor is the number of the color to be used for subsequent output. (See GPAMGrColorTable for numbers.)

GPAMSetFont (pGPAMWA, pbFontName, cbFontName,
wWidth, wHeight) : ErcType

DESCRIPTION

GPAMSetFont specifies the font identification (40 bytes or less), its width, and its height. Because the font identification and width are separately specified, it is possible to use the same font at different pitches.

On devices requiring manual changing of the font, the output device driver makes a status entry indicating that a change to font FontName is required and then pauses to wait for a Restart command given with SetGPSRestart.

Font width and height are specified in 1440ths of an inch.

(See the Printing Guide for more information on fonts.)

PROCEDURAL INTERFACE

GPAMSetFont (pGPAMWA, pbFontName, cbFontName,
wWidth, wHeight) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

pbFontName
cbFontName is the name of the font to be used (maximum is 40 bytes).

wWidth is the width of the font. This parameter is used for fixed pitch fonts only. A value of 0 selects the default width for the named font.

wHeight is the height of the font. A value of 0 selects the default height for the named font. The height of the font is its point size.

GPAMSetSLD (pGPAMWA, wSLDDistance) : ErcType

DESCRIPTION

GPAMSetSLD specifies the vertical distance to move for each subsequent **GPAMLineFeed** or **Return** (the single line distance). The value is specified in 1440ths of an inch. The default is determined by the device driver.

If the value specified is not supported by the output device, then the next smaller value that is supported is used. If a smaller value is not supported, then the next larger value that is supported is used.

When the first character of output is provided for a new line, the most recent **GPAMSetSLD** value is applied and cannot be changed for that line, until the next **GPAMLineFeed**, **FormFeed**, or **Return**.

PROCEDURAL INTERFACE

GPAMSetSLD (pGPAMWA, wSLDDistance) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wSLDDistance is the vertical distance in 1440ths of an inch that will be used for subsequent **GPAMLineFeed** or **Return** operations.

GPAMSetVerticalOffset (pGPAMWA, intLevel) :
ErcType

DESCRIPTION

GPAMSetVerticalOffset is used for subscripts and superscripts.

A positive **intLevel** value causes subsequent characters to be superscripted by moving the output position vertically up a fraction of the single line distance (specified via **GPAMSetSLD**). Negative values work similarly for subscripting.

Values of **intLevel** greater than 1 or less than -1 have the effect of nested superscripts and subscripts. For example, an **intLevel** value of 1 causes normal superscripting, while a value of 2 has the effect of nested superscripts ("a superscripted superscript").

PROCEDURAL INTERFACE

GPAMSetVerticalOffset (pGPAMWA, intLevel) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

intLevel is a 16-bit signed integer. Allowed values are negative 8 through positive 8, with 0 indicating a return to the baseline position. Positive values indicate superscripting (that is, the printing position will be moved upward). Negative values indicate subscripting.

If a given output device does not support the number of levels specified, then the next level closer to the baseline position is selected.

GPAMWhiteSpace (pGPAMWA, wWidth) : ErcType

DESCRIPTION

GPAMWhiteSpace puts wWidth 1440ths of an inch of white space at the current position, and advances the current position by wWidth. If underscore or overstrike are activated, they are applied to the white space.

PROCEDURAL INTERFACE

GPAMWhiteSpace (pGPAMWA, wWidth) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wWidth is the length of the white space in 1440ths of an inch.

GPAMWriteRecord (pGPAMWA, pb, cb, pcbRet) :
ErcType

DESCRIPTION

GPAMWriteRecord writes the specified bytes to the open byte stream identified by the memory address of the **GPAMWA**.

GPAMWriteRecord is similar to **WriteBSRecord** with the exception that it precedes all application-generated data with a length field. This ensures that no application-generated data will be mistaken for a **GPAM** escape sequence.

The data written to the output stream may contain any of the possible 256 character codes.

The following single character codes cause special action whenever they are present in the text; they have no action within graphics data:

- **CarriageReturn** (0Dh) causes the output pointer to move to the left most position on the current line.

Note that this position cannot be flush with the left edge of the physical page if **GPAMReposition** has been used to establish a different left margin.

- **LineFeed (0Ah)** acts the same as **CarriageReturn** followed by **GPAMLineFeed**. This is the preferred method of starting a new line.
- **FormFeed (0Ch)** acts much like a call to **GPAMBeginPage** but does not alter any of the current page attribute settings. In addition, the current printer state is not replicated for restart purposes. Thus, **FormFeed** should not be used unless default printer conditions are acceptable during restart.

Note that use of some codes within a text area, if not translated to graphic characters by the translate table in the output device driver, can result in a variety of actions when sent to the output device. For example, **VerticalTab (0Bh)** and the **HorizontalTab (09h)** make some printers space vertically or horizontally on the page. GPAM does not prevent such codes from being sent to the device driver, but the results are not guaranteed by GPAM and may be device-specific.

PROCEDURAL INTERFACE

GPAMWriteRecord (pGPAMWA, pb, cb, pcbRet) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

pb is the memory address of the data to be written.

cb is the count of bytes to be written. The maximum is 508 bytes.

pcbRet is the memory address of the word to which the count of data bytes successfully written is returned.

GPAMGrBeginGraphics (pGPAMWA, wWidth, wHeight,
fClear) : ErcType

DESCRIPTION

GPAMGrBeginGraphics indicates that graphics output is to begin. Subsequently, all GPAM routines called before **GPAMGrEndGraphics** must be either graphics routines (**GPAMGr** routines) or **GPAMWriteRecord** routines containing VDM-format graphics information.

The application can specify graphic output by using **GPAMWriteRecord** to write out the image of a VDM file (which may have been created using calls to the **WriteVDM** routine in **Graphics.Lib**), or by calling other **GPAMGr** routines.

GPAMGrBeginGraphics also specifies the width and height of the graphics area, thus allowing the output device driver to automatically scale the graphics.

The picture is placed with the upper-left corner at the current print position. Thus, in many cases **GPAMReposition** should be used before calling **GPAMGrBeginGraphics** to establish the desired starting location.

Any currently active formatting controls (for example, **Bold**) are not applied to the graphics.

Pictures cannot cross page boundaries.

PROCEDURAL INTERFACE

GPAMGrBeginGraphics (pGPAMWA, wWidth, wHeight,
fClear) : Erctype

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wWidth is the width of the picture in 1440ths of an inch. Values up to 65535 (45.5 inches) can be specified. A value of zero means to use the device driver's default page width.

wHeight is the height of the picture in 1440ths of an inch. Values up to 65535 (45.5 inches) can be specified. A value of zero means to use the device driver's default page height.

fClear is true if the area specified by **wWidth** and **wHeight** should be cleared of all information before the picture is printed.

GPAMGrColorTable (pGPAMWA, wIndex, pbColors,
cbColors) : ErcType

DESCRIPTION

GPAMGrColorTable sets the color palette to the values in the passed color array beginning at the passed index value (for example, if wIndex 115 is specified, then the color values from 115 through 255 can be specified). For each index location, colors are expressed as consecutive integer intensities of red, green and blue, where 0 is 0% and 1000 is 100%. You can define .1% increments.

PROCEDURAL INTERFACE

GPAMGrColorTable (pGPAMWA, wIndex, pbColors,
cbColors) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wIndex is the palette index number of the first palette to change. Allowed values are 0 to 255.

pwColors
cwColors describe the array of consecutive Red, Green and Blue values to place in the color palette.

	wRed	wGreen	wBlue	
0	1000	0	0	= Red
1	0	1000	0	= Green
2	0	0	1000	= Blue
3	1000	1000	1000	= White
4	0	0	0	= Black

885-013

Figure 6-2. Color Array

GPAMGrEndGraphics (pGPAMWA) : ErcType

DESCRIPTION

GPAMGrEndGraphics terminates the definition of graphics output.

PROCEDURAL INTERFACE

GPAMGrEndGraphics (pGPAMWA) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

GPAMGrFillColor (pGPAMWA, wColor) : ErcType

DESCRIPTION

GPAMGrFillColor sets the color used by GPAMGrPolygon for interior fills.

PROCEDURAL INTERFACE

GPAMGrFillColor (pGPAMWA, wColor) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wColor is the color index number to use for filling. Allowed values are 0 to 255.

GPAMGrHatchIndex (pGPAMWA, wHatchIndex) : ErcType

DESCRIPTION

GPAMGrHatchIndex sets the hatch pattern used by GPAMGrPolygon to fill the figure. This value is used as an index into a set of predefined hatch types. It is significant only when GPAMInteriorStyle sets the style to hatch.

PROCEDURAL INTERFACE

GPAMGrHatchIndex (pGPAMWA, wHatchIndex) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wHatchIndex is the index into a list of hatch types to use for polygon filling. Allowed values are 0 to 6.

Crosshatch Definition

- 1 = Narrow Diagonal
- 2 = *** Not Available ***
- 3 = Wide Diagonal
- 4 = Narrow Crosshatch
- 5 = *** Not Available ***
- 6 = Wide Crosshatch

GPAMGrInteriorStyle (pGPAMWA, wInteriorStyle, fPerimeterVis) : ErcType

DESCRIPTION

GPAMGrInteriorStyle sets the style used by **GPAMGrPolygon** to fill the figure. The **wInteriorStyle** parameter is used as an index into a set of predefined interior styles. **fPerimeterVis** determines if the polygon perimeter is to be drawn or not.

PROCEDURAL INTERFACE

GPAMGrInteriorStyle (pGPAMWA, wInteriorStyle, fPerimeterVis) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wInteriorStyle is the index into a list of interior styles to use for polygon filling. Allowed values are 0 to 3.

fPerimeterVis is TRUE if the perimeter is to be visible and FALSE otherwise.

Interior Style Definition

- 0 = Hollow
- 1 = Solid
- 2 = Hatch
- 3 = *** Not Available ***

GPAMGrLineColor (pGPAMWA, wColor) : ErcType

DESCRIPTION

GPAMGrLineColor sets the color used by GPAMGrPolyLine.

PROCEDURAL INTERFACE

GPAMGrLineColor (pGPAMWA, wColor) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wColor is the color index number to use for line drawing. Allowed values are 0 to 255.

GPAMGrLineType (pGPAMWA, wLineType) : ErcType

DESCRIPTION

GPAMGrLineType sets the line type used by GPAMGrPolyline. This value is used as an index into a set of predefined line types.

PROCEDURAL INTERFACE

GPAMGrLineType (pGPAMWA, wLineType) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wLineType is the index into a list of line types to use for line drawing. Allowed values are 0 to 8.

Line Type Definition

0 = Solid	_____
1 = Dashed	-----
2 = Dotted
3 = Dash Dot	-. - . - . -
4 = * Not Available *	
5 = Line Dash Dash	_ - - _ - -
6 = Line Dash	- - - - -
7 = Line Blank	- - - - -
8 = Line Blank Blank	- - - - -

GPAMGrPerimeterColor (pGPAMWA, wColor) : ErcType

DESCRIPTION

GPAMGrPerimeterColor sets the color used by GPAMGrPolygon to draw the polygon perimeter.

PROCEDURAL INTERFACE

GPAMGrPerimeterColor (pGPAMWA, wColor) : ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

wColor is the color index number to use for drawing the polygon perimeter. Allowed values are 0 to 255.

GPAMGrPolygon (pGPAMWA, pbPoints, cbPoints) :
ErcType

DESCRIPTION

GPAMGrPolygon draws a closed polygon. The polygon is described as a series of points in an array with consecutive points connected, and the last point connected to the first. The GPAMGrInteriorStyle routine sets whether or not the polygon will be outlined, and the GPAMPerimeterColor routine defines the outline color. The values set in GPAMGrFillColor, GPAMGrInteriorStyle and GPAMGrHatchIndex determine the interior contents of the polygon.

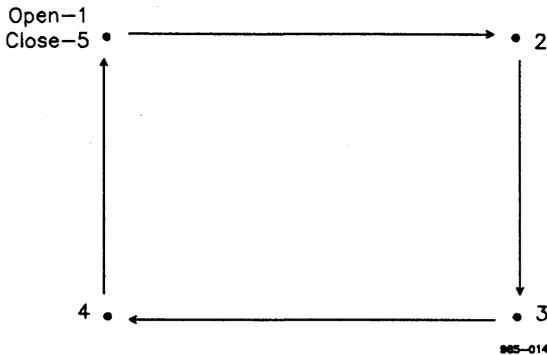


Figure 6-3. A Square Has Five Points

PROCEDURAL INTERFACE

GPAMGrPolygon (pGPAMWA, pbPoints, cbPoints) :
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

pbPoints
cbPoints describe the array of points used to draw the lines. The limit is 8191 points (32764 bytes).

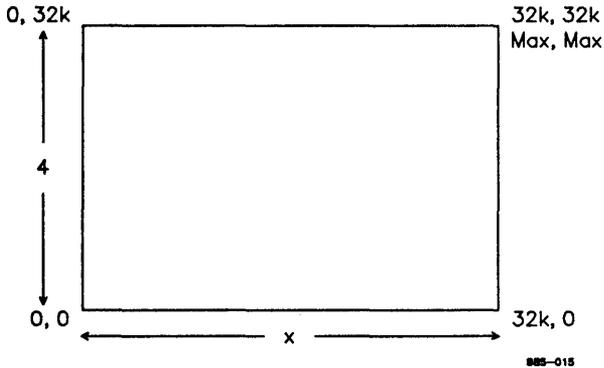


Figure 6-4. Maximum X and Y Values

GPAMGrPolyLine (pGPAMWA, pbPoints, cbPoints):
ErcType

DESCRIPTION

GPAMGrPolyLine draws a sequence of connected lines, described as a series of points in an array. The values set in GPAMGrLineColor and GPAMGrLineStyle determine the color and type of the line.

PROCEDURAL INTERFACE

GPAMGrPolyLine (pGPAMWA, pbPoints, cbPoints):
ErcType

where

pGPAMWA is the memory address of a 300-byte memory work area for use by GPAM procedures.

pbPoints
cbPoints describe the array of points used to draw the lines. The limit is 8191 points (32764 bytes).

7 GPS ROUTING SWITCH REQUESTS

This chapter gives a detailed description of each of the requests you can make of the Generic Print System routing switch. See Chapter 5, "Using GPS Requests" for an overview and examples.

Table 7-1

SUMMARY OF ROUTING SWITCH REQUESTS

(Page 1 of 2)

CloseGPSFile	Terminates an output session, closes the open device driver and releases associated hardware resources.
DeleteGPSJob	Causes a document queued at a device to be deleted.
GetGPSDevHistory	Provides a list of documents that have been printed by a particular device, indicates when the printing started and finished, and the status of the document upon termination.
GetGPSDeviceConfig	Returns configuration information about a particular device.
GetGPSDocInfo	Used to query GPS about the status of documents that are printing or are waiting to be printed at a selected device.
GetGPSNetList	Returns a list of all GPS output devices accessible from the workstation running the application.
GetGpsSplVolStatus	Returns the number of 512-byte pages remaining on the spooling volume.

Table 7-1

SUMMARY OF ROUTING SWITCH REQUESTS

(Page 2 of 2)

GetGPSStatus request	Returns status information for a particular output device.
OpenGPSFile	Opens a GPS device for output.
ProvideGPSPassword	Provides a password so that printing of a password protected document, printed in security mode can start.
QueueGPSFile	Allows an application to add a document file to a spooled generic device's queue.
SetGPSControl	Provides the mechanism for applications to send control commands to generic print devices.
SetGPSParams	Provides information to the GPS system about a job that has just been opened with the OpenGPSFile request.
SetGPSRestart	Provides the mechanism for applications to send restart control commands to generic output devices that have previously been paused.
SetImageModeGPS	Sets normal, image, or binary mode for a GPS byte stream.
WriteGPSFile	Transfers a number of bytes from memory to the printer.

CloseGPSFile

Request

CloseGPSFile (jh): ErcType

DESCRIPTION

The CloseGPSFile request terminates an output session, closes the open device driver and releases associated hardware resources.

Note that if your application is also using GPAM routines, you do not need to use this request since GPAM does it for you.

PROCEDURAL INTERFACE

CloseGPSFile (jh): ErcType

where

jh is a job handle returned from an OpenGPSFile operation.

CloseGPSFile

(continued)

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	2
1	RtCode	1	0
2	nReqPbCb	1	0
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	803Bh
12	jh	2	

DeleteGPSJob

Request

DeleteGPSJob (pbDeviceName, cbDeviceName,
pbDocumentName, cbDocumentName):
ErcType

DESCRIPTION

The DeleteGPSJob request causes a document queued at a device to be deleted.

PROCEDURAL INTERFACE

DeleteGPSJob (pbDeviceName, cbDeviceName,
pbDocumentName, cbDocumentName):
ErcType

where

pbDeviceName
cbDeviceName are a pb/cb pair pointing to the device name.

pbDocumentName
cbDocumentName are a pb/cb pair that point to a string which identifies a document.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	2
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	803Dh
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pbDocumentName	4	
30	cbDocumentName	2	

GetGpsDevHistory (pbDeviceName, cbDeviceName,
pBufferRet, sBufferMax): ErcType

DESCRIPTION

The **GetGPSDevHistory** provides a list of which documents have been printed by a particular device, when the printing started and finished, and the status of the document upon termination. The number of documents in this list is finite and defined by the device driver when it is built.

Documents must be closed before they become history. **GetGPSDevHistory** does not list jobs currently open.

PROCEDURAL INTERFACE

GetGpsDevHistory (pbDeviceName, cbDeviceName,
pBufferRet, sBufferMax): ErcType

where

pbDeviceName
cbDeviceName are a pb/cb pair pointing to the device name.

pBufferRet
sBufferMax are a pb/cb pair pointing to a buffer area in which the device history will be returned. The format of this data structure is defined in Table 7-5.

GetGPSDevHistory(continued)

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	803Fh
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pBufferRet	4	
30	sBufferMax	2	

```
GetGPSDeviceConfig (pbDeviceName, cbDeviceName,  
                    pBufferRet, sBufferMax):  
                    ErcType
```

DESCRIPTION

The `GetGPSDeviceConfig` request returns configuration information about a particular device.

PROCEDURAL INTERFACE

```
GetGPSDeviceConfig (pbDeviceName, cbDeviceName,  
                    pBufferRet, sBufferMax):  
                    ErcType
```

where

`pbDeviceName`
`cbDeviceName` are a pb/cb pair pointing to the device name.

`pBufferRet` is the memory address of the buffer into which to place the configuration information (see Table 7-4).

`sBufferMax` is the size of the buffer. If the buffer is too small, the configuration information is truncated.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8040h
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pBufferRet	4	
30	sBufferMax	2	

GetGPSDocInfo

Request

GetGpsDocInfo (qeh, pbDeviceName, cbDeviceName,
pBufferRet, sBufferMax,
psBufferRet, pQehRet,
pQehNextRet): ErcType

DESCRIPTION

The GetGPSDocInfo request is used to query GPS about the status of documents that are printing, or are waiting to be printed at a selected device.

If no more entries exist for the selected device, this request returns the status message `ercRsEndOfQueue`.

If the queue is not empty, status messages are returned directly from the device.

For direct print devices, only one entry can be returned.

PROCEDURAL INTERFACE

GetGpsDocInfo (qeh, pbDeviceName, cbDeviceName,
pBufferRet, sBufferMax,
psBufferRet, pQehRet, pQehNextRet):
ErcType

where

qeh is the 32-bit queue entry handle returned from a previous GetGPSDocInfo operation. A 0 indicates the first entry in the queue.

pbDeviceName
cbDeviceName are a pb/cb pair pointing to the device name.

pBufferRet
sBufferMax are a pb/cb pair pointing to a memory area into which the document information will be returned. The format of this information is defined in Table 7-7, which appears at the end of this chapter.

psBufferRet is a pointer to a word into which the count of bytes returned will be placed.

pQehRet is a pointer to a 4-byte memory area where the queue entry handle for this entry is returned.

pQehNextRet is a pointer to a 4-byte memory area where the next queue entry handle is returned.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	4
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8041h
12	qeh	4	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pBufferRet	4	
30	sBufferMax	2	
32	psBufferRet	4	
36	ssBufferRet	2	2
38	pQehRet	4	
42	sQehRet	2	4
44	pQehNextRet	4	
48	sQehNextRet	2	4

GetGPSNetList (iDevice, pBufferRet, sBufferMax,
psBufferRet): ErcType

DESCRIPTION

The **GetGPSNetList** request is to be used by application programs to interrogate the routing switch for a list of all GPS output devices accessible from the workstation running the application.

Because the number of devices accessible can be large, this request is designed to be called several times in sequence so as to allow receipt of a complete net list when the return buffer is not large enough to contain the entire list.

If iDevice is set to 0, the buffer will receive a concatenation of two data structures (described in Tables 7-2 and 7-3), up to the maximum buffer size as set by sBufferMax. This form provides the total number of devices available to the routing switch.

If iDevice is set to $n \neq 0$, then the buffer will receive a data structure in the form of Table 7-2, starting with device n and repeated for device $n+1$, $n+2$, ..., up to the maximum allocated space or until the end of the net list, whichever is less.

PROCEDURAL INTERFACE

GetGPSNetList (iDevice, pBufferRet, sBufferMax,
psBufferRet): ErcType

where

iDevice is a word defining the index of the first device to be returned in the buffer defined below.

pBufferRet is the memory address of the buffer into which to place the configuration information (see Tables 7-2 and 7-3).

sBufferMax is the size of the buffer. If the buffer is too small, the configuration information is truncated.

psBufferRet points to a word into which the count of bytes returned is placed.

GetGPSNetList

(continued)

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	0
3	nRespPbCb	1	2
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8042h
12	iDevice	2	
14	reserved1	2	
16	reserved2	2	
18	reserved3	2	
20	pBufferRet	4	
24	sBufferMax	2	
26	psBufferRet	4	
30	ssBufferRet	2	2

GetGPSSplVolStatus

Request

GetGpsSplVolStatus (pbDeviceName, cbDeviceName,
pCountRet) : ErcType

DESCRIPTION

The **GetGpsSplVolStatus** request is used by the client to interrogate the **GpsSpooler** as to the number of 512-byte pages remaining on the spooling volume.

PROCEDURAL INTERFACE

GetGpsSplVolStatus (pbDeviceName, cbDeviceName,
pCountRet) : ErcType

where

pbDeviceName
cbDeviceName are a pb/cb pair pointing to the device name.

pCountRet is a pointer to a double word in memory which will receive the count of free 512-byte pages on the volume used by the spooler.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8044h
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pCountRet	4	
30	sCountRet	2	4

GetGPSStatus (pbDeviceName, cbDeviceName,
pStatusRet, sStatusMax): ErcType

DESCRIPTION

The **GetGPSStatus** request is to be used by application programs to interrogate the routing switch for information regarding the status of a particular output device. This request does not use the job handle convention of **WriteGPSFile** or **CloseGPSFile**, since it is possible and desirable to be able to get the status of a device, independent of whether a job is open on that device.

PROCEDURAL INTERFACE

GetGPSStatus (pbDeviceName, cbDeviceName,
pStatusRet, sStatusMax): ErcType

where

pbDeviceName

cbDeviceName

describe a character string of the form [devname], where devname is an alphanumeric string of up to 12 characters.

pStatusRet

is the memory address of the buffer into which to place the device-status information. The format of this information is defined in Table 7-8.

sStatusMax

is the size of the buffer. If the buffer is too small, the configuration information is truncated.

GetGPSStatus

(continued)

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8045h
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pStatusRet	4	
30	sStatusMax	2	

OpenGPSFile

Request

OpenGPSFile (pJhRet, pbDeviceName, cbDeviceName,
pbPassword, cbPassword): ErcType

DESCRIPTION

The OpenGPSFile procedure opens a GPS device for output. The job handle returned by OpenGPSFile is used to refer to the job in subsequent operations such as WriteGPSFile and CloseGPSFile. In the case of direct output, the device driver reserves the required hardware resources at this time and establishes an interrupt handler for the device.

Note that if your application is also using GPAM routines, you do not need to use this request, since GPAM does it for you.

OpenGPSFile opens a device only. Use OpenByteStream to open a disk file for printing.

PROCEDURAL INTERFACE

OpenGPSFile (pJhRet, pbDeviceName, cbDeviceName,
pbPassword, cbPassword): ErcType

where

pJhRet is the memory address of a word to which the job handle is returned.

pbDeviceName
cbDeviceName describe a character string of the form [devname], where devname is an alphanumeric string of up to 12 characters.

pbPassword
cbPassword describe a device, volume, directory or file password.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	2
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8048h
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pbPassword	4	
30	cbPassword	2	
32	pJhRet	4	
36	sJhMax	2	2

ProvideGPSPassword

Request

ProvideGPSPassword (pbDeviceName, cbDeviceName,
pbPassword, cbPassword):
ErcType

DESCRIPTION

If a document was submitted to GPS in security mode, (see fSecurity variable in job parameters data structure, Table 7-10), a password is required before the spooler can access and print it. When the document is ready to be printed, the GPS system will pause and require the user to provide the proper password via the ProvideGPSPassword request.

PROCEDURAL INTERFACE

ProvideGPSPassword (pbDeviceName, cbDeviceName,
pbPassword, cbPassword):
ErcType

where

pbDeviceName
cbDeviceName describe a character string of the form [devname], where devname is an alphanumeric string of up to 12 characters.

pbPassword
cbPassword describe a device, volume, directory or file password.

ProvideGPSPassword(continued)

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	2
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8049h
12	reserved1	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pbPassword	4	
30	cbPassword	2	

QueueGPSFile

Request

QueueGPSFile (pbFileName, cbFileName,
pbDeviceName, cbDeviceName, fDelete,
pJobParameters, sJobParameters):
ErcType

DESCRIPTION

The QueueGPSFile procedure allows an application to add a document file to a spooled generic device's queue.

This request is provided to eliminate the overhead of copying the document from the user's file to another file that is created for temporary use by the GPS spooler. This is not a ByteStream type operation, and has two major restrictions:

1. The generic output device referenced by the request must be a spooled device.
2. The specified file must be accessible by the spooler. Note that this is not always possible because the file may be on a local file system, while the spooler may be on a master, another cluster workstation, or a remote node, and therefore cannot access the file.

Note that if your application is also using GPAM routines, you do not need to use this request since GPAM does it for you.

This request is very useful with very large documents. The request points to the file itself, instead of making a temporary copy of the file. For average-sized files, use WriteGPSFile.

PROCEDURAL INTERFACE

QueueGPSFile (pbFileName, cbFileName,
pbDeviceName, cbDeviceName, fDelete,
pJobParameters, sJobParameters):
ErcType

where

pbFileName
cbFileName is the name of a file containing document data that is to be added to the queue of an output device for subsequent output.

pbDeviceName
cbDeviceName is the device name for the GPS output device that the document file is to be output upon.

fDelete is set to TRUE if this user-defined file is to be deleted by the GPS after it has been output.

pJobParameters
sJobParameters define a buffer containing a data structure which specifies that various control and information parameters are to be associated with this document. This data structure is defined in Table 7-9.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	3
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	804Ah
12	fDelete	1	
13	reserved	1	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pbFileName	4	
30	cbFileName	2	
32	pJobParameters	4	
36	sJobParameters	2	

```
SetGPSControl (bCommand, bSubCommand,  
              pbDeviceName, cbDeviceName):  
              ErcType
```

DESCRIPTION

SetGPSControl provides the mechanism for applications to send control commands to generic print devices. Using this request, the application may request that the output be halted or paused, that the output be canceled, or that the output device driver repeat the forms alignment page and pause. (Restart commands are sent to generic print devices by the SetGPSRestart request, which is defined elsewhere in this chapter.)

SetGPSControl does not use a job handle for identification of the device driver due to the fact that with spooled output it is necessary to provide control commands for documents that were created earlier and are no longer open.

PROCEDURAL INTERFACE

```
SetGPSControl (bCommand, bSubCommand,  
              pbDeviceName, cbDeviceName):  
              ErcType
```

where

bCommand has the following values:

0 = PAUSE: Halt the output of data and save it for possible future printing.

- 1 = CANCEL: Halt the output of data and inform the device that no more data will be forthcoming. Discard any currently buffered data and any any future write commands until a proper start-up command is received.
- 2 = ALIGN Output the alignment form and pause.
- 3 = JOBCHANGE: Define a print-job change. (See definition below for JOBCHANGE.) This JOBCHANGE command is to be used only after all data for the previous job has been sent to the device driver, and before any data for the next document has been sent to the device driver.
- 4 = NOP-1: Reserved.
- 5 = NOP-2: Reserved.

bSubCommand has the following values when **bCommand = 3 (JOBCHANGE)**:

- 0 = Invalid
- 1 = Beginning of a banner document.
Implies end of any previously existing print job, since a banner does not have page numbers.
- 2 = Beginning of document.
- 3 = End of existing job.
- 4 = End of this copy.
- 5 = Document password required.
- 6 = Document password no longer required.
- 7 = Existing job ended due to error.
- 8 = Beginning of error-message document.

pbDeviceName
cbDeviceName describe a character string of the form [devname], where devname is an alphanumeric string of up to 12 characters.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	804Bh
12	bCommand	1	
13	bSubCommand	1	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	

SetGPSParams

Request

SetGPSParams (jh, pJobParameters, sJobParameters):
ErcType

DESCRIPTION

The SetGPSParams request provides information to the GPS system about a job that has just been opened with the OpenGPSFile request. GPS printing can be performed without using this SetGPSParams request, but it is strongly recommended that it be used with all OpenGPSFile requests. For certain GPS-print facilities, the SetGPSParams request must be used. The information provided by this request,

1. Controls the queuing and printing of this file.
2. Enhances the information returned by subsequent requests for output-status information.

This request, if present, must follow an OpenGPSFile request, and must precede any WriteGPSFile request for that job.

Note that if your application is also using GPAM routines, you do not need to use this request since GPAM does it for you.

PROCEDURAL INTERFACE

SetGPSParams (jh, pJobParameters, sJobParameters):
ErcType

where

jh is a job handle returned from an
OpenGPSFile operation.

pJobParameters
sJobParameters define a buffer containing a data
structure which specifies that
various control and information
parameters are to be associated
with this document. This data
structure is defined in Table
7-9. It also displays in the
Print Manager and is used by the
spooler.

SetGPSPParams**(continued)**

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	2
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	804Dh
12	jh	2	
14	pJobParameters	4	
18	sJobParameters	2	

```
SetGPSRestart (pbDeviceName, cbDeviceName,  
               pDocLocation, sDocLocation, wCopy):  
               ErcType
```

DESCRIPTION

The SetGPSRestart request provides the mechanism for applications to send restart control commands to generic output devices that have previously been paused. A generic device may be requested to continue its paused output, or to resume its output at a different position in the document. The total number of pages per copy and the total number of copies to be output may be changed by this request.

For direct printing and plotting, restart may only commence at the current byte in the output stream; any ending-page specification is ignored. Spooled output may use any of the restart options.

SetGPSRestart does not use a job handle for identification of the device driver due to the fact that with spooled output it is necessary to provide control commands for documents which were created earlier and are no longer open.

PROCEDURAL INTERFACE

SetGPSRestart (pbDeviceName, cbDeviceName,
pDocLocation, sDocLocation, wCopy):
ErcType

where

pbDeviceName
cbDeviceName describe a character string of the
form [devname], where devname is an
alphanumeric string of up to 12
characters.

pDocLocation
sDocLocation is a buffer containing Document-
Location records that define:

1. where in the document
output is to be restart-
ed; or
2. which pages of the docu-
ment are to be printed.
This buffer is to contain
two PageLocation records.
The first defines the
Restart "Beginning Page,"
and the second defines
the Restart "Ending
Page." See Table 7-10
for the definition of the
PageLocation record and
its values and uses.

Valid PageLocation record specification-class values for use with this request are as follows:

Beginning Page	Ending Page
First/Last	First/Last
Logical	Logical
Physical	Physical
Backup Restart	Pages Beyond
Original Page	Original Page
Continue Restart	
Next-Page Restart	
TopOfPage Restart	

If the "Ending Page" specification is of Original Page type, then the location specified is only for purposes of restarting. Otherwise, the specified pages redefine which pages of the document are to be printed for each copy.

wCopy redefines the Current-Copy value for this document-output job. A value of zero indicates no change.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	8
1	RtCode	1	0
2	nReqPbCb	1	2
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	804Eh
12	wCopy	2	
14	reserved2	2	
16	reserved3	2	
18	reserved4	2	
20	pbDeviceName	4	
24	cbDeviceName	2	
26	pDocLocation	4	
30	sDocLocation	2	

SetImageModeGPS (jh, mode): ErcType

DESCRIPTION

The SetImageModeGPS request sets normal, image, or binary mode for a GPS byte stream.

Mode controls the way the printing device interprets line control characters and escape sequences, thereby controlling your ability to pass printer-specific escape sequences through to the printing device with your output stream.

- Normal mode converts any tab symbols in the stream of characters passed to the printing device into spaces and maps all line feeds to the line control specifications set at device installation time.
- Image mode suppresses all character conversions, except pre-GPS printer or spooler escape sequences. (See the Executive Manual.)
- Binary mode passes all characters to the printer with no interpretation at all.

Note that if your application is also using GPAM routines, you do not need to use this request since GPAM does it for you.

PROCEDURAL INTERFACE

SetImageModeGPS (jh, mode): ErcType

where

jh is a job handle returned from an **OpenGPSFile** operation.

mode is a code as follows:

0 for normal mode
1 for image mode
2 for binary mode

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	4
1	RtCode	1	0
2	nReqPbCb	1	0
3	nRespPbCb	1	0
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	804Fh
12	jh	2	
14	mode	2	

WriteGPSFile

Request

WriteGPSFile (jh, pBuffer, sBuffer, psDataRet):
ErcType

DESCRIPTION

The WriteGPSFile operation transfers a number of bytes from memory to the printer. WriteGPSFile returns only when the requested transfer is complete.

Note that if your application is also using GPAM routines, you do not need to use this request since GPAM does it for you.

This request is best used with small/medium files. Use QueueGPSFile for large files.

PROCEDURAL INTERFACE

WriteGPSFile (jh, pBuffer, sBuffer, psDataRet):
ErcType

where

jh is a job handle returned from an OpenGPSFile operation.

pBuffer is the memory address of the first byte of the buffer from which the data is to be written.

WriteGPSFile(continued)

sBuffer is the count of bytes to be written from memory.

psDataRet is the memory address of the word to which the count of bytes successfully written is to be returned.

REQUEST BLOCK

Offset	Field	Size (bytes)	Contents
0	sCntInfo	1	2
1	RtCode	1	0
2	nReqPbCb	1	1
3	nRespPbCb	1	1
4	userNum	2	
6	exchResp	2	
8	ercRet	2	
10	rqCode	2	8039h
12	jh	2	
14	pBuffer	4	
18	sBuffer	2	
20	psDataRet	4	
24	ssDataRet	2	2

REFERENCE TABLES

Table 7-2**FORMAT OF AddGPSDevice DEVICE STRUCTURE**
(Page 1 of 2)

Offset	Field	Size (bytes)	Contents
0	sbDeviceName	13	The name of the output device (the first byte of an "sb" string is the length of the string).
13	sbAlias	13	An alias name for the output device assigned by the Print Manager in order to avoid duplicate printer names when more than one occurrence of a particular driver exists within the network.
26	sbNode	13	CT-Net node where the device is located.

Table 7-2

FORMAT OF AddGPSDevice DEVICE STRUCTURE

(Page 2 of 2)

39	sbLocation	25	Geographic location of printer. This field can be displayed by application programs to aid the user in locating the printer to which output is directed.
64	fSpooled	1	TRUE if the device is to be controlled by the spooler.
65	fBanner	1	TRUE if the device driver is to print a banner page at the beginning of each document.
66	sbQueueNameRoot	13	A root name for the set of queues to be used for spooled device control.
79	fAtMaster	1	TRUE if device is located at the master of a cluster.
80			

Table 7-3

FORMAT OF GPS NET LIST

(Page 1 of 1)

Offset	Field	Size (bytes)	Contents
0	nPrinters	2	The number of printers in this table.
2	sRgPrinter	2	The size of the rgPrinter data structure following. nPrinters * sRgPrinters + 12 is the total required size in bytes of this data structure.
4	sbGPSVersion	8	String containing the GPS-version number.
12	rgPrinter	81*n	See Table 7-2 for rgPrinter format. There are nPrinters occurrences of rgPrinter in this table.

Table 7-4

DEVICE CONFIGURATION INFORMATION

(Page 1 of 2)

Offset	Field	Size (bytes)	Contents
0	sbDeviceName	13	The name of the output device.
13	fAlpha	1	If set to TRUE, the output device can be used for alpha-numeric output.
14	fGraph	1	If set to TRUE, the output device can be used for graphic output.
15	sbVersion	13	The first 12 characters of the version specified when the device driver was linked.
28	sbAlias	13	An alias name for the output device assigned by the Print Manager in order to avoid duplicate device names when more than one occurrence of a particular driver exists within the network.

Table 7-4

DEVICE CONFIGURATION INFORMATION

(Page 2 of 2)

41	wnSpeed	2	Output device speed in characters per second. This information is useful for timing purposes.
43	sbNode	13	CT-Net node where the device resides.
56	sbLocation	25	Geographic location of device. This field can be displayed by application programs to aid the user in locating the device to which output is directed. (For example, this field might contain the information "Software Lab" or "Printer Room.")
81	fSpooled	1	TRUE if the device is spooled.
82	fAtMaster	1	TRUE if device is located at the master of a cluster.
83	sbChannel	9	The channel device specification.
92			

Table 7-5

DEVICE HISTORY INFORMATION

(Page 1 of 1)

Offset	Field	Size (bytes)	Contents
0	cEntriesMax	1	Count of the total number of device history entries configured for this device. This value is defined when the driver is built.
1	cEntriesActual	1	Count of the number of entries that have valid device history information. Not to exceed specified cEntriesMax.
2	rgHistory	133	See Table 7-6 for the format of a History entry.

Table 7-6

FORMAT OF A HISTORY ENTRY

(Page 1 of 1)

Offset	Field	Size (bytes)	Contents
0	sbDocumentName	93	The name of the document.
93	sbUserName	31	Name of document's owner.
124	bCompletionStatus	1	Status of document completion. If bCompletionStatus = 0 Document completed normally. 1 Document was aborted by a user. 2 Document was aborted because of an error.
125	qStartTime	4	Time when the document started printing in system time format.
129	qFinishTime	4	Time when the document finished printing in system time format.

NOTES:

1. Entire structure (Table 7-5/7-6) = 135.
 2. Sufficient memory must be allocated or no information will be put into Table 7-6.
-

Table 7-7

DOCUMENT INFORMATION

(Page 1 of 4)

Offset	Field	Size (bytes)	Contents
0	fDelAftPrt	1	If set to TRUE (OFFH), the spooled file will be deleted after it is printed.
1	sbFileSpec	92	The name of the file to be printed (the first byte of an "sb" string is the length of the string).
93	sbFormName	13	The name of the form to be used. If the length is zero, the standard form will be used.
106	sbFontName	13	The name of the font to be used. If the length is zero, the devices' standard font will be used.
119	cCopies	2	The number of copies of the file that are to be printed.
121	bPrintMode	1	The printing mode, where 0 = normal 1 = binary 2 = image

Table 7-7

DOCUMENT INFORMATION

(Page 2 of 4)

Offset	Field	Size (bytes)	Contents
122	fAlignForms	1	If this is TRUE, the forms alignment option will be used.
123	fSecurityMode	1	If this is TRUE, the file will be printed in security mode.
124	reserved	5	Reserved for use by the Word Processor.
129	sbDocName	92	The name of the document being printed. This is different from sbFileSpec, which is typically a temporary file in the [scr]<spl> directory.
221	sbUserName	31	The client's user name.
252	wTimeToPrint	2	Approximate print time in minutes.
254	reserved	2	Reserved for WP.
256	timeQueued	4	The date/time that the document was queued.

Table 7-7

DOCUMENT INFORMATION

(Page 3 of 4)

Offset	Field	Size (bytes)	Contents
260	fSuppressNewPage	1	If TRUE, the device driver will not print a form-feed at the start of the document.
261	fWPPaging	1	If TRUE, the device driver will use WP page escape sequences to determine page numbers.
262	fSuppressBanner	1	If TRUE, the device driver will not print a banner or the notice file.
263	fSingleSheet	1	If TRUE, the device attached is manual feed.
264	fImmediate	1	If TRUE, this file can commence output even before the entire file has been received.
265	qFileSize	4	The total number of characters in the file.

Table 7-7

DOCUMENT INFORMATION

(Page 4 of 4)

Offset	Field	Size (bytes)	Contents
269	rgStartPage	29	A data structure defining where in the document to commence output. See Table 7-10 for a description of this data structure.
298	rgEndPage	29	A data structure defining where in the document to terminate output. See Table 7-10 for a description of this data structure.
327	bPriority	1	Queue entry priority.
328	fMarked	1	If TRUE, this entry is currently being processed.

Table 7-8

DEVICE STATUS INFORMATION

(Page 1 of 2)

Offset	Field	Size (bytes)	Contents
0	wsTable	2	Size of this table in bytes. This value corresponds to the number of bytes required to contain the entire table.
2	sbDeviceName	13	The name of the device.
15	reserved	1	Space reserved for former use.
16	bStatus	1	The device status, where 0 = idle 1 = paused 2 = printing 3 = offline 4 = down
17	fNeedFontChange	1	TRUE if a font different than the current is needed.
18	fNeedFormsChange	1	TRUE if a different form is needed.
19	fShowPauseMessage	1	TRUE if the pause message should be displayed.
20	sbPauseMessage	61	The pause message to be displayed.

Table 7-8

DEVICE STATUS INFORMATION

(Page 2 of 2)

Offset	Field	Size (bytes)	Contents
81	sbFontNeeded	21	The name of the font needed.
102	sbFormNeeded	21	The name of the form needed.
123	wColor	2	The number of the current color.
125	sbForm	21	The name of the current form.
146	sbFont	21	The name of the current font.
167	sbDocument	93	The name of the current document.
260	sbUserName	31	The client's name.
291	rgCurrentPage	29	The current page location record, see table 7-10.
320	cCurrentCopy	2	The index of the current copy being output.
322	qOutputStartTime	4	The date/time that the document was started.
326			

Table 7-9

JOB PARAMETERS

(Page 1 of 3)

Offset	Field	Size (bytes)	Contents
0	sbDocName	93	Document name to be associated with this output job.
93	sbUserName	31	Client's User-Name.
124	fSecurity	1	Set if document is to be output in security mode.
125	fSuppressPage	1	Set if there should be no page eject at the beginning of this job.
126	fSuppressBanner	1	Set if banner is not to be printed before printing the document.
127	fTypeThrough	1	Set if output is not to be buffered.
128	fImmediate	1	Set if client would desire spooled output to begin before spooling is completed.
129	qDocCreationTime	4	Time that document was created.
133	qReqOutputStartTime	4	Time after which document is to be output.

Table 7-9

JOB PARAMETERS

(Page 2 of 3)

Offset	Field	Size (bytes)	Contents
137	wcCopies	2	Number of copies of document to be output.
139	rgbStartLocation	29	PageLocation record that specifies which page of the document should begin each copy of the document. (See Table 7-10 for the definition of the PageLocation record and its values & uses.)
168	rgbEndLocation	29	PageLocation record that specifies which page of the document should end each copy of the document. (See Table 7-10 for the definition of the PageLocation record and its values and uses.)

Table 7-9

JOB PARAMETERS
(Page 3 of 3)

Offset	Field	Size (bytes)	Contents
			Valid PageLocation record specification-class values:
		<u>Beginning Page</u>	<u>Ending Page</u>
		First/Last Logical Physical	First/Last Logical Physical Pages Beyond
197	fSingleSheet	1	If TRUE, pause between pages when printing.
198	wTimeToPrint	2	Approximate print time in minutes.
200	bPriority	1	Queue entry priority.
201			

Table 7-10

PAGE LOCATION RECORD DEFINITION
(Page 1 of 1)

Each GPS Page Location record has the following structure:

Offset	Field	Size (bytes)	Contents
0	bClass	1	The value of this byte specifies the type, or class, of page specification contained in the rest of this Page Location record. The values are all explained in the following description.
1	cwPages	2	See the following description.
3	sbSection	13	See the following description.
16	sbPage	13	See the following description.
29			

Explanation:

The GPS PageLocation record is used to specify a GPS document page physically, logically or implicitly. The information that follows explains how it is used.

A user of the Generic Print System uses the PageLocation record for one of two purposes:

1. When first requesting that a document be printed, PageLocation records may be used to specify how much of the document is to be printed.
2. When restarting a print job, PageLocation records may be used to specify where to restart, and how much of the document to print.

A physical page N is the Nth page in the document. The beginning of a page is defined as anything that results in an explicit Form Feed.

A logical page is defined by a logical-section string and a logical-page-number string. These strings, if they exist, are included in documents as explicit Beginning-of-Page definitions. Currently, logical-section strings are not included in any documents. So, GPS drivers will set the logical-section string to "1" at the beginning of a document. And one will be added to the logical-section string each time that the logical-page-number string does not increase.

GPS-document pages may be specified according to the following classes:

- A. **First/Last:** If the PageLocation record is specifying a "beginning" page, it is the document's first page; if it is specifying an "ending" page, then it is the last. This class of PageLocation specification is used when the bClass field of the PageLocation record is set to 0.

B. Continue Restart:

This is one of the classes which may be used only to specify where a restart should begin; this class specifies that printing will begin at the item following the last item output. The bClass-field value for this specification is 1.

C. Next-Page Restart:

This is one of the classes which may be used only to specify where a restart should begin; this class specifies that printing will begin at the page following the last one being output. The bClass-field value for this specification is 2.

D. Logical:

A logical-page specification consists of a Section-# string and a Page-# string. Only pages that have been generated with associated # strings may be "logically" specified. When a PageLocation record contains a logical specification; the bClass field = 3, the sbSection field contains the Section-# string, and the sbPage field contains the Page-# string.

E. **Physical:** A physical-page specification consists of the page's physical index in the cwPages field of the PageLocation record. And the bClass field = 4.

F. **TopOfPage Restart:**

This is one of the classes which may be used only to specify where a restart should begin; this class specifies that printing will begin at the top of the page that was being output. The bClass-field value for this specification is 5.

G. **Backup Restart:**

This is one of the classes which may be used only to specify where a restart should begin; this class specifies that printing will begin at the top of the page that is N pages before the page that was being output. The bClass-field value for this specification is 5, and the value N is in the cwPages field of the PageLocation record.

H. **Pages Beyond:**

Only the ending page may be specified by this class of PageLocation record. This class specifies that the last page of this document to be printed is the page that is N physical pages beyond the beginning page.

The bClass-field value for this specification is 8, and the value N is in the cwPages field of the PageLocation record

I. Original Page:

This class specifies that the beginning-page or ending-page should be kept as specified in the original GPS print request. For this class, 7 is the bClass-field value.

J. Current Page:

This class specification is used to signal that the PageLocation record contains current-page status information (for example, which page is currently being printed). The page's logical Section-# and Page-#, if it has them, are in the sbSection and sbPage fields. The page's physical index is in the cwPages field, and 9 is the value in the bClassppp field.

STATUS CODES

The following list includes status codes that can occur with the printing system. For a more detailed listing of status codes, see the Status Codes Manual.

DEVICE DRIVER STATUS CODES

- 4544 Mode Value Error.
- An invalid Image-Mode value was specified.
- 4545 Buffer Too Small.
- The supplied data buffer was too small for the requested data.
- 4546 Begin-Page-Class Error.
- An invalid value was specified for "Beginning-Page" class.
- 4547 Ending-Page-Class Error.
- An invalid value was specified for "Ending-Page" class.
- 4548 Command-Value Error.
- An invalid value was specified for the Control Command.
- 4549 Sub-Command-Value Error.
- An invalid value was specified for the Control Sub-Command.

4550 to Reserved for internal use.
4551

4552 This request to GPS is not appropriate
at this time.

4553 Current operation upon this GPS device
must be completed before this request to
GPS may be made.

4554 Reserved for internal use.

4555 Reserved for internal use.

4556 Driver DeInstalling.

Request cannot be serviced, because the
device driver for the requested device
is currently being removed.

4557 Cancelled Job.

This job has been cancelled -- this data
will not be output.

4558 No Such Doc.

The specified document does not exist.

4559 to Reserved for internal use.
4566

4567 Cannot Acquire Device.

Device driver cannot acquire access to
the device's output port, because 1) GPS
has already acquired this port, or 2)
some application is currently using the
port.

4568 to Reserved for internal use.
4570

- 4571 String Too Long.
A name passed to GPS is too long.
- 4572 Transparent Mode error.
An error has occurred with an attempt to send GPAM data to the printer in transparent mode3.
- 4573 Bad Data.
The device driver has encountered data in the document that it cannot process, for example a nonprintable file has been sent to the printer for printing.
- 4574 No such font.
- 4575 Internal error.

ROUTING SWITCH STATUS CODES

- 4576 Device not installed.
The requested device is not currently installed.
- 4577 Bad request file.
You have an incorrect version of the Request file, [Sys]<Sys>Request.9.Sys. Check your distribution diskette.
- 4578 A workstation prefix is required if the routing switch is installed at a diskless workstation.
- 4579 Internal error.
- 4580 See error 4591.

- 4581 to Internal error.
4582
- 4583 An invalid request code has been detected.
- 4584 Cannot deinstall GPS when running on a single partition OS.
- 4585 Device driver has not yet identified itself, therefore this device cannot be activated.
- 4586 Duplicate device spec.
- An attempt was made to add a device that already exists.
- 4887 An attempt was made by GetGpsDocInfo to read additional entries for a direct print device.
- On direct print devices, document information is available only of the document currently printing.
- 4588 An attempt was made to read beyond the end of a device scheduling queue by the GetGPSDocInfo request.
- 4589 Internal error.
- 4590 A direct print device is already opened by another user.
- 4591 Exceeded maximum number of simultaneously open device streams.
- Either reduce the number of simultaneously open devices or increase the [Maximum number of open files (default=12)] value specified at the time of routing switch installation.

4592 Network device table is full.

Either reduce the number of devices known to the routing switch or increase the [Maximum number of network devices (default=10)] value specified at the time of routing switch installation.

4593 Unbalanced brackets on device spec.

4594 Invalid node specification.

4595 Routing switch internal error.

4596 Routing switch internal error.

4597 Routing switch internal error.

4598 The device name is not recognized.

4599 The device table is full.

Either reduce the number of devices known to the routing switch or increase the [Maximum number of local devices (default=3)] value specified at the time of routing switch installation.

SPOOLER STATUS CODES

712 The device handle in the request was unrecognized.

713 An attempt was made to supply a password when the spooler was not waiting for one.

715 An attempt was made to install a device with the same name as one already in use.

- 716 Internal error.
- 717 The spooler device table is full.
- Either reduce the number of devices or reinstall the spooler with a larger maximum number of devices.
- 718 An attempt was made to access an uninstalled device.
- 719 Internal error
- 720 An attempt was made to remove a device which was not in service.
- 721 An attempt was made to process a job with an unrecognized spooler job index.
- 722 The spooler was unable to handle a new spooling job.
- 723 GPS spooler not installed.
- The spooler is not currently installed.
- 724 Queue Manager not installed.
- 725 Illegal command.

QUEUE MANAGER STATUS CODES

- 915 The queue table is full.
- Either reduce the number of queues or reinstall the queue manager with a larger maximum number of queues.
- 916 The queue handle in the request was unrecognized.

- 917 An attempt was made to remove a queue that is no longer in service.
- 918 An attempt was made to install a queue with a file specification matching that of a queue already in service.
- 919 An attempt was made to install a queue with a name matching one already in service.
- 920 Internal error
- 921 An attempt was made to install the queue manager with an illegal installation parameter.

APPENDIX A: CONVERTING NON-GPS APPLICATIONS

An application system that was not written or modified for use with the the Generic Print System probably includes device-specific codes in its output to the printer. To convert such a system you may want to do the following:

- Rewrite the system using GPAM. For long-range use with workstations, this is the best choice, since it gives you complete device independence.
- If the system was originally written for CTOS, you can replace the printer or spooler specification in any calls to `OpenByteStream` with a specification for a GPS printer name, then relink with `GPAM.lib`, and a 10.0 or higher version of `CTOS.lib`.

Since, in this case, you are leaving device-dependent information in your output stream, you must be careful that the output does indeed go to that kind of printer. For example, you can do this by adding code to check the printer type.

GLOSSARY

application system. An application system is any application program, written by either the user or vendor.

aspect. The aspect ratio is the ratio of height to width of any rectangle.

buffer. A buffer is an area of memory used to store information temporarily.

byte stream. A generic stream of bytes, called a byte stream, is sent to SAM to be routed to the specified device.

Byte Stream Work Area. A Byte Stream Work Area is a 130-byte memory work area for the exclusive use of Sequential Access Method procedures. Any number of byte streams can be open concurrently, using separate Byte Stream Work Areas.

cluster. A cluster configuration is a local resource-sharing network consisting of a master workstation and up to 16 cluster workstations.

data stream. A data stream consists of text and device-independent formatting commands sent to the printing device in much the same way that you send a GPS byte stream. The GPAM data stream is translated into device-specific commands by a device driver specific to each device.

device driver. The GPS device driver is an installed system service that controls a printing device and translates the Generic Print Access Method byte stream it receives into device-specific commands. Device drivers are usually unique for each type of printing device.

direct printing. Direct printing is printing that transfers text directly to the printer.

escape sequence. An escape sequence is a special sequence of characters that invokes special functions.

exchange. An exchange is the path over which messages are communicated from process to process (or interrupt handler to process). An exchange consists of two first-in, first-out queues; one of the processes waiting for messages, the other a queue of messages for which no process has yet waited. An exchange is referred to by a unique 16-bit integer.

job handle. A job handle is a 16-bit integer that uniquely identifies an open request. A job handle is returned by the request, `OpenGPSFile` and is used to refer to the file in subsequent operations such as `ReadGPSFile` or `WriteGPSFile`.

Generic Print Access Method. The Generic Print Access Method (GPAM) is a library of object module procedures that you can use to completely free your application from device-specific printer or plotter commands. If you want your application to offer rich formatting, such as boldface text, underlines, superscript, or graphics, you can format your output as a GPAM data stream. The data stream consists of text and device-independent formatting commands sent to the printing device in much the same way that you send a GPS byte stream. The GPAM data stream is translated into device-specific commands by a device driver specific to each device.

landscape aspect. Printing on a rectangular page where the main axis runs the width of the page (as on this page), is called the landscape aspect.

node. A node is a location in a CT-Net network of workstations. It can be a master or a standalone workstation.

object module procedure. An object module procedure is a procedure supplied as part of an object module file. It is linked with the user-written object modules of an application system.

pb/cb. A pb/cb is a 6 byte entity consisting of the 4-byte memory address of a byte string followed by the 2-byte count of the bytes in that byte string.

portrait aspect. Printing on a rectangular page where the main axis runs the length, rather than the width of the page, is called the portrait aspect.

queue manager. The Queue Manager handles all queues of jobs for the cluster and is also used by non-printing applications, such as Batch. One Queue Manager is installed on the master workstation of any cluster where spooled devices are installed.

request block. A request block is a block of memory containing request information, submitted by a client process to a system service. Request blocks are exchanged by the system service and the client process by sending the memory address of the request block. Request blocks are used most often when the client process want to continue processing rather than to wait at an exchange for the system service to return.

routing switch. A routing switch is a system service that routes requests to the designated printing device and coordinates the work of the entire Generic Print System. The routing switch also handles printer control commands (such as pause and restart), and status enquiries. Routing switches within a cluster or network communicate with each other to pass requests or status information. A routing switch is installed on any workstation to which a printer is attached and on the master workstation of any cluster.

Sequential Access Method. The CTOS Sequential Access Method provides device-independent calls an application can use to access real devices, such as the video, keyboard, or printer. These calls send a generic stream of bytes, called a byte stream, through SAM to the specified device.

spooled printing. Spooled printing is a printing process that transfers text to a print queue.

spooler. The spooler gives a print request to the Queue Manager to place in the print queue and, when the printing device is free to print asks for the print request back again. The spooler is installed on each workstation to which a spooled device is attached.

system service. A system service is like a modular part of the operating system that resides in workstation memory. Some system services can be dynamically installed (added or removed at any time). System services are not a part of the operating system when it is first installed, nor are they resident after a system is bootstrapped unless you make special entries in your system initialization file for their installation.

VDM file. A VDM file is a graphics output file originally created using the Graphics.lib routine, WriteVDM.

INDEX

- AddGPSDevice
 - format of device structure, 7-44 to 7-45
- Application
 - adding status functions to, 5-7
 - converting a non-GPS, 3-1, A-1
 - linking to include byte streams, 3-3
 - use of Queue Manager with, 2-12 to 2-13

- Boldface
 - assigning, 4-12
- Byte streams
 - description of, 2-2 to 2-3
 - use of, 3-1 to 3-3

- Carriage Return (0Dh), 4-10
- CGM
 - using files created with, 4-17
- CGM files
 - writing with GPAMWriteRecord, 4-14
- Character placement
 - specifying, 4-11 to 4-12
 - using GPAMPlaceCharacter, 4-12
- CloseGPSFile, 7-4
- Cluster workstation
 - definition of, 1-2
- Color Table
 - assigning to text with GPAMSetColor, 4-12
 - setting color table with GPAMGrSetColorTable, 4-15
 - setting with GPAMGrSetColorTable, 4-15
 - specifying for line with GPAMGrLineColor, 4-16
 - specifying for perimeter with GPAMPerimeterColor, 4-16
 - specifying interior with GPAMGrFillColor, 4-17
- Coordinate base
 - placing characters using, 4-11
- Coordinates
 - specifying page, 4-11
 - specifying polygon, 4-16
- Cross-hatching
 - specifying with GPAMGrHatchIndex, 4-17

- Daisy wheel printer
 - device driver (DaisyDD.Run), 2-11
- Data stream
 - closing with GPAMClose, 4-8
 - opening with GPAMOpen, 4-8
 - structure, 4-8
 - example of, 4-9

- DeleteGPSJob, 7-6
- Device configuration
 - querying GPS with GetGPSDevConfig, 5-5
- Device driver
 - daisy wheel (DaisyDD.Run), 2-11
 - description of, 2-8
 - generic (LptSimpleDD.Run), 2-11
 - use of, 2-10 to 2-11
- Device history
 - history entry, format of, 7-50
 - history information, 7-49
- Device status information, 7-55 to 7-56
- Document information, 7-51 to 7-54
- Draft quality output
 - definition of, 2-11
 - device driver for, 2-11

- Error codes, 8-1 to 8-7
- Errors
 - resolving with GPAMRelease, 4-8

- Fonts
 - specifying with GPAMSetFont, 4-13
- Form Feed (0Ch)
 - using GPAMBeginPage in place of, 4-10

- Generic Print Access Method (GPAM)
 - description of, 2-2, 2-5
 - linking with, 4-3
 - procedures, 4-4 to 4-7
 - summary of, 6-2 to 6-5
 - specifying graphics output with, 4-2
 - use of, 4-1 to 4-17
- Generic Print Access Method procedures
 - GPAMBeginBold, 6-6
 - GPAMBeginBound, 6-7
 - GPAMBeginItalic, 6-9
 - GPAMBeginOverstrike, 6-10
 - GPAMBeginPage, 6-12
 - GPAMBeginTransparent, 6-17
 - GPAMBeginUnderscore, 6-18
 - GPAMClose, 6-20
 - GPAMEndBold, 6-21
 - GPAMEndBound, 6-22
 - GPAMEndItalic, 6-23
 - GPAMEndOverstrike, 6-24
 - GPAMEndTransparent, 6-25
 - GPAMEndUnderscore, 6-26
 - GPAMGrBeginGraphics, 6-49
 - GPAMGrColorTable, 6-51

Generic Print Access Method procedures (cont.)

- GPAMGrEndGraphics, 6-53
- GPAMGrFillColor, 6-54
- GPAMGrHatchIndex, 6-55
- GPAMGrInteriorStyle, 6-56
- GPAMGrLineColor, 6-57
- GPAMGrLineType, 6-58
- GPAMGrPerimeterColor, 6-59
- GPAMGrPolygon, 6-60
- GPAMGrPolyLine, 6-62
- GPAMLineFeed, 6-27
- GPAMOpen, 6-28
- GPAMPlaceCharacter, 6-34
- GPAMRelease, 6-35
- GPAMReposition, 6-36
- GPAMSetColor, 6-39
- GPAMSetFont, 6-40
- GPAMSetSLD, 6-42
- GPAMSetVerticalOffset, 6-43
- GPAMWhiteSpace, 6-45
- GPAMWriteRecord, 6-46
- Generic Print System (GPS)
 - description of, 1-1, 2-1 to 2-2
 - direct calls to, 5-1 to 5-8
 - installation of, 1-2
 - programming with, 2-2
- Generic Print System Net List
 - format of, 7-46
- GetGPSDevConfig, 5-5
- GetGPSDevConfig, 7-10
- GetGPSDevHistory, 7-8
- GetGPSDocInfo, 7-12
- GetGPSNetList, 7-15
- GetGPSSplVolStatus, 7-18
- GetGPSStatus, 7-20
- GPAM Data stream. See Data stream.
- GPAM. See Generic Print Access Method.
- GPAMBeginBold, 6-6
- GPAMBeginBound, 6-7
- GPAMBeginGraphics, 4-14
- GPAMBeginItalic, 6-9
- GPAMBeginOverstrike, 6-10
- GPAMBeginPage, 4-10, 6-12
- GPAMBeginTransparent, 4-14, 6-17
- GPAMBeginUnderscore, 6-18
- GPAMClose, 4-8, 6-20
- GPAMEndBold, 6-21
- GPAMEndBound, 6-22
- GPAMEndItalic, 6-23
- GPAMEndOverstrike, 6-24
- GPAMEndTransparent, 4-14, 6-25
- GPAMEndUnderscore, 6-26
- GPAMFillColor, 4-17
- GPAMGrBeginGraphics, 6-49

- GPAMGrColorTable, 6-51
- GPAMGrEndGraphics, 6-53
- GPAMGrFillColor, 6-54
- GPAMGrHatchIndex, 4-17, 6-55
- GPAMGrInteriorStyle, 4-16, 6-56
- GPAMGrLineColor, 4-16, 6-57
- GPAMGrLineType, 6-58
- GPAMGrPerimeterColor, 6-59
- GPAMGrPolygon, 6-60
- GPAMGrPolyLine, 6-62
- GPAMGrSetColorTable, 4-15
- GPAMLineFeed, 4-11, 6-27
- GPAMOpen, 4-13, 4-8, 6-28
- GPAMPlaceCharacter, 4-12, 6-34
- GPAMPolyLine, 4-15
- GPAMRelease, 6-35
 - resolving GPS errors with, 4-8
- GPAMReposition, 4-12, 6-36
- GPAMSetColor, 4-12, 6-39
- GPAMSetFont, 4-13, 6-40
- GPAMSetSLD, 6-42
- GPAMSetVerticalOffset, 4-12, 6-43
- GPAMWhiteSpace, 6-45
- GPAMWriteRecord, 6-46
 - writing CGM files with, 4-14
- GPS Byte Streams. See **Byte streams.**
- GPS Device Driver. See **Device driver.**
- GPS PageLocation record, 7-60
- GPS Routing Switch. See **Routing switch.**
- GPS. See **Generic Print System.**
- Graceful degradation, 2-5, 4-3
- Graphics output
 - from existing CGM file, 4-2
 - specifying with GPAM, 4-2
- Graphics procedures
 - descriptions of, 4-15 to 4-17
 - GPAMGrBeginGraphics, 6-49
 - GPAMGrColorTable, 6-51
 - GPAMGrEndGraphics, 6-53
 - GPAMGrFillColor, 6-54
 - GPAMGrHatchIndex, 6-55
 - GPAMGrInteriorStyle, 6-56
 - GPAMGrLineColor, 6-57
 - GPAMGrLineType, 6-58
 - GPAMGrPerimeterColor, 6-59
 - GPAMGrPolygon, 6-60
 - GPAMGrPolyLine, 6-62
 - use of, 4-9
- Graphics
 - outputting with GPAM, 4-14
 - positioning, 4-15
 - scaling, 4-15

Hatching

- specifying with GPAMGrHatchIndex, 4-17

Horizontal tab (09h), 4-10

Installation

- Generic Print System, 1-2

- Queue Manager, 2-13

Interior color

- specifying with GPAMGrFillColor, 4-17

Job parameters, 7-57 to 7-59

Legal characters

- definition of, 4-10

Letter quality output

- definition of, 2-11

- device driver for, 2-11

Line feed

- distance, setting with GPAMSetSLD, 4-11

- using GPAMLineFeed, 4-11

- using Return (0Ah), 4-11

Lines

- specifying colors with GPAMGrLineColor, 4-16

- specifying coordinates with GPAMPolyLine, 4-15

- specifying type with GPAMGrLineType, 4-16

Logical page

- definition of, 7-61

LptSimpleDD.Run

- description of device driver, 2-11

Master workstation

- definition of, 1-3

Non-GPAM commands

- sending to printer with GPAMBeginTransparent, 4-14

Non-GPS applications

- converting, 3-1, A-1

OpenGPSFile, 5-2, 5-5, 7-22

Output

- draft quality, 2-11

- interrupting with GPAMOpen, 4-13

- letter quality, 2-11

Output device(s)

- configuration information, 7-47 to 7-48

- controlling with SetGPSControl, 5-7

- format of history entry, 7-50

- history information, 7-49

- listing default with GetGPSDefaultPrinter, 5-6

Output device(s) (cont.)

- listing with GetGPSNetList, 5-6
- restarting with SetGPSRestart, 5-7
- specifying a, 5-6

Overstriking

- assigning, 4-12

Page location record definition, 7-60 to 7-64

Page, logical

- definition of, 7-61

Page, physical

- definition of, 7-61

Pagination, 4-10

- starting with GPAMBeginPage, 4-10

Physical page

- definition of, 7-61

Polygons

- defining interior color with GPAMFillColor, 4-17
- specifying coordinates with GPAMGrCoordinate, 4-16
- specifying interior with GPAMGrInteriorStyle, 4-16

Print Manager

- description of, 1-2

Print position

- specifying , 4-11 to 4-12
- using GPAMResposition, 4-12

Print request

- checking status with GetGPSDocInfo, 5-7
- deleting with DeleteGPSJob, 5-7
- making with GPS output functions, 5-2

Printer(s)

- configuration information, 7-47 to 7-48
- controlling with SetGPSControl, 5-7
- format of history entry, 7-50
- graceful degradation, 2-5
- history information, 7-49
- listing default with GetGPSDefaultPrinter, 5-6
- listing with GetGPSNetList, 5-6
- restarting with SetGPSRestart, 5-7
- sending non-GPAM commands to, 4-14
- sending output to, 3-2 to 3-3
- specifying a, 3-2, 5-6

Printing mode

- setting with SetImageModeGPS, 5-6

Program

- converting a non-GPS, 3-1, A-1
- linking to include byte streams, 3-3

ProvideGPSPassword, 7-22

- Queue Manager
 - description of, 2-9
 - installation of, 2-13
 - Queue.Index file, 2-12
 - Queueb.Index file, 2-13
 - use of, 2-12
 - use with non-GPS applications, 2-12
- QueueGPSFile, 7-26

- Return (0Ah), 4-10
- Routing switch
 - description of, 2-8, 5-1
 - summary of requests, 5-3 to 5-4, 7-2 to 7-3
 - use of, 2-9 to 2-10
- Routing Switch requests
 - CloseGPSFile, 7-4
 - DeleteGPSJob, 7-6
 - GetGPSDevConfig, 7-10
 - GetGPSDevHistory, 7-8
 - GetGPSDocInfo, 7-12
 - GetGPSNetList, 7-15
 - GetGPSSplVolStatus, 7-18
 - GetGPSStatus, 7-20
 - OpenGPSFile, 7-22
 - ProvideGPSPassword, 7-22
 - QueueGPSFile, 7-26
 - SetGPSControl, 7-29
 - SetGPSParams, 7-33
 - SetGPSRestart, 7-36
 - SetImageModeGPS, 7-40
 - WriteGPSFile, 7-42

- SAM. See Sequential Access Method.
- Sequential Access Method (SAM)
 - description of, 1-1
- SetGPSControl, 7-29
- SetGPSParams, 5-2, 5-5, 7-33
- SetGPSRestart, 7-36
- SetImageModeGPS, 3-3, 7-40
- Spooler
 - description of, 2-9
 - opening file with QueueGPSFile, 5-5
 - use of, 2-12
- Standalone workstation
 - definition of, 1-2
- Status codes, 8-1 to 8-7
- Status functions
 - adding to applications, 5-7
 - GetGPSDevHistory, 5-7
 - GetGPSDocInfo, 5-7
- Status information, device, 7-55 to 7-56

- Subscripting
 - setting with GPAMSetVerticalOffset, 4-12
- System service,
 - definition of, 2-1

- Text
 - assigning attributes, 4-12

- Text attributes
 - assigning, 4-12

- Timeout
 - after GPAMWriteRecord, 4-13
 - setting with GPAMOpen, 4-13

- Troubleshooting, 8-1 to 8-7

- Underscoring
 - assigning, 4-12

- Workstation
 - cluster, 1-2
 - master, 1-3
 - standalone, 1-2
- WriteGPSFile, 7-42

USER'S COMMENT SHEET

Generic Print System Programmer's Guide, First Edition
09-00985-01

We welcome your comments and suggestions. They help us improve our manuals. Please give specific page and paragraph references whenever possible.

Does this manual provide the information you need? Is it at the right level? What other types of manuals are needed?

Is this manual written clearly? What is unclear?

Is the format of this manual convenient in arrangement, in size?

Is this manual accurate? What is inaccurate?

Name _____ Date _____

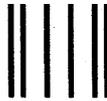
Title _____ Phone _____

Company Name/Department _____

Address _____

City _____ State _____ Zip Code _____

*Thank you. All comments become the property of
Convergent Technologies, Inc.*

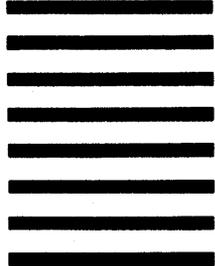


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS MAIL PERMIT NO. 1807 SAN JOSE, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Convergent Technologies
Attn: Technical Publications
2700 North First Street
PO Box 6685
San Jose, CA 95150-6685



Fold Here

Convergent

2700 North First Street
San Jose, CA 95150-6685

Printed in USA