The graPHIGS Programming Interface: Customization and Problem Diagnosis

SC33-8130-05

The graPHIGS Programming Interface: Customization and Problem Diagnosis

SC33-8130-05

Note

Before using this information and the product it supports, read the information in Appendix B, “Notices,” on page 43.

Sixth Edition (October 2000)

This edition of the *The graPHIGS Programming Interface: Customization and Problem Diagnosis* applies to the following IBM program products: the GDDM/graPHIGS Programming Interface, Version 2, Release 2.5, program number 5688-093, AIXwindows Environment/6000 (1.3) AIXwindows/3D feature, Program Number 5601-257, and to all subsequent releases of this product until otherwise indicated in new editions.

Contents

About This Book

This manual describes the hardware supported by the IBM Personal graPHIGS Programming Interface and hardware and software requirements. It explains how to customize the IBM Personal graPHIGS Programming Interface for optional environments and plotter support.

It also provides:

- Information needed to diagnose problems in the IBM Personal graPHIGS Programming Interface and the graPHIGS API gateway
- Guidelines for locating the symptoms of the graPHIGS API problem
- Techniques for collecting the supporting data required for further analysis
- An Authorized Program Analysis Report (APAR) form
- A description of the information required for submitting the APAR

Use this manual with other manuals in the graPHIGS API library.

Who Should Use This Book

This book is intended for application programmers.

Highlighting

The following highlighting conventions are used in this book;

Bold	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
Italics	Identifies parameters whose actual names or values are to be supplied by the user.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related Publications

The following books contain information on graPHIGS API products:

- *The graPHIGS Programming Interface: Getting Started*
- *The graPHIGS Programming Interface: ISO PHIGS Subroutine Reference*
- *The graPHIGS Programming Interface: ISO PHIGS Quick Reference*
- *The graPHIGS Programming Interface: Messages and Codes*
- *The graPHIGS Programming Interface: Quick Reference*
- *The graPHIGS Programming Interface: Subroutine Reference*
- *The graPHIGS Programming Interface: Technical Reference*
- *The graPHIGS Programming Interface: Understanding Concepts*

Chapter 1. Hardware and Software Support and Requirements

This chapter provides the following information about the IBM Personal graPHIGS API:

- Hardware support
- Hardware and software requirements for the graPHIGS API gateway daemon

Hardware Support

For information on specific systems and their compatible adapters, see *PCI Adapter Placement Reference*.

The graPHIGS API supports the following:

Adapters and Processors

- POWER GXT100
- POWER GXT110P
- POWER GXT120P
- POWER GXT130P
- POWER GXT150M
- POWER GXT155L
- POWER GXT250P
- POWER GXT255P
- POWER GXT300P
- POWER GXT500
- POWER GXT500D
- POWER GXT500P
- POWER GXT550P
- POWER GXT800P
- POWER GXT800M
- POWER GXT1000
- POWER GXT2000P
- POWER GXT3000P
- POWER GXT4000P
- POWER GXT4500P
- POWER GXT6000P
- POWER GXT6500P
- POWER Gt3
- POWER Gt3i
- POWER Gt4e
- POWER Gt4
- POWER Gt4i
- POWER Gt4x
- POWER Gt4xi
- POWER GTO
- Grayscale Graphics Display Adapter
- Color Graphics Display Adapter

Displays

- IBM 16-inch Color
- IBM 6091 19-inch Color
- IBM 6091 19i-inch Color
- IBM 6091 23-inch Color
- IBM P70 17-inch Color
- IBM P72 17-inch Color
- IBM P76 17-inch Color
- IBM P90 19-inch Color
- IBM P92 19-inch Color
- IBM P200 20-inch Color
- IBM P201 20-inch Color
- IBM P202 21-inch Color
- IBM P260 21-inch Color
- IBM Powerdisplay 17 17-inch Color
- IBM Powerdisplay 20 20-inch Color

Tablets

- IBM 6093-011
- IBM 6093-012
- IBM 6093-021

Pointing Devices

- IBM RS/6000 family Mouse - 3 button (#6041)
- IBM 6094 030 Spaceball
- IBM 6094 031 Spaceball
- IBM 6094 040 Spaceball
- IBM 6094 051 Spaceball
- IBM 6094 600 Magellan

Lighted Program Function Keyboard

- IBM 6094 020 Lighted Program Function Keyboard

Dials

- IBM 6094 010 Dials

Plotters

Only operating system drivers are supported for plotters.

Keyboards for National Language Support

- IBM Chinese Keyboard (#6033)(required for Traditional Chinese Language Support)
- IBM Korean Keyboard (#6031) (required for Korean language support)
- IBM Japanese Kanji Keyboard (#6030) (required for Japanese language support)

Hardware and Software Requirements for the graPHIGS API Gateway

In addition to the minimum requirements for the graPHIGS API, **gPgated** requires the following:

RS/6000 workstation:

Host Interface Adapter, PN 09G3667 or higher.

Minimum hardware requirements for the RS/6000 to be used as a gateway are the same as the minimum requirements for installation of the operating system: 32 MB real memory, 400MB DASD, and an ASCII terminal on s1/s2. You can install up to three Host Interface Adapters (HIA) on an RS/6000. 48MB real memory is suggested for optimal performance with multiple HIAs.

IBM S/390:

- GDDM/graPHIGS API V2.2.1 and GAM/SP 3.0 or 3.1 with the current service level updates
- The following 5088 or 6098 microcode EC levels:
  - 5088 EPROM level 9056 or higher
  - 5088-11R EPROM level 5187 or higher
  - 6098 microcode 0104 or higher (1.0602 or higher recommended)
  - 6098-1R EPROM level 3846 or higher.

See *The graPHIGS Programming Interface: Technical Reference* for more information on using the graPHIGS API gateway.

Requirements for Connection to 6098 with FDDI

In addition to the minimum requirements for the graPHIGS API, the graPHIGS/GAM direct connection to the 6098 with the FDDI feature requires the following:

- 6098 Channel Control Unit with the FDDI feature
  - 6098 microcode 2.02 or higher (2.1002 or higher recommended)
- IBM S/390
  - GDDM/graPHIGS API 2.2.3 or higher, and GAM/SP 3.0 or 3.1 with the current service level updates

Chapter 2. The graPHIGS API Directories and Files

This chapter provides information about the organization of the graPHIGS API directories and files.

The following table identifies the graPHIGS API directories and the files located in those directories:

Directory	Name	Use
/usr/include	afmnc.h	non-reentrant interface include file for C compiles
	afmrc.h	reentrant interface include file for C compiles
	afmsc.h	system programmer interface include file for C compiles
	afmnpas.inc	non-reentrant interface include file for Pascal compiles
	afmrpas.inc	reentrant interface include file for Pascal compiles
	phigs.fortran	PHIGS include file for ISO PHIGS FORTRAN applications
	ginteger.inc	include file for Pascal compiles
	afmgp.h	long names for graPHIGS API functions and enumerations defined by these functions
	afmrcp.h	include file for all rcp codes
	phigs.h	include file for ANSI/ISO PHIGS C binding
/usr/lib	libgP.a	archive containing the graPHIGS API
	librgP.a	archive containing the reentrant interface to the graPHIGS API
	libgPA.a	the graPHIGS API shared library
	libgksco.a	archive containing the graPHIGS GKS-CO Interface
	libgppipe.a	graPHIGS API soft graphics pipeline code
	librengppex_*.a	graPHIGS API rasterizer code for each 3D graphics adapter and soft graphics
	librengppex_soft.a	graPHIGS API soft graphics rasterizer code
	librengppex_mag.a	graPHIGS API rasterizer code for POWER GXT500 and POWER GXT500D
/usr/bin	cvtgdf	GDF conversion utility
	gPinit	script file to start a remote graPHIGS API nucleus
	gPterm	script file to stop a remote graPHIGS API nucleus
	gPhost	script file to add and delete host names permitted to access a remote graPHIGS API nucleus
	gPafut	script file to run the archive file utility
	gPq	script file to inquire the status of remote graPHIGS API nuclei
	gPtutor	graPHIGS API tutorial
	chgPcon	add/change/delete a run time connection profile entry for a graPHIGS API gateway daemon
	lsgPcon	inquire run time connection profile information from a graPHIGS API gateway daemon

Directory	Name	Use
	gPgated	the graPHIGS API gateway daemon that allows a GDDM/graPHIGS API shell to communicate with remote graPHIGS API nuclei on workstation(s)
	ls6098	inquire connection information for the 6098 with FDDI feature
	mkgPcon	make or break a connection with a 6098 with FDDI feature
/usr/lpd	cgm2hp2	HP GL2 plotter support
	cgm2hp2.readme	HP GL2 readme file
	cpsC1	CalComp** plotter support
	config.ccp	CalComp configuration file
	cpsI1	IBM plotter backend
	cpsV1	Versatec** plotter support
	PLOTPARM	Versatec plot parameter files
	README.ccp	CalComp readme file
	README.VERSA	Versatec readme file
	*.DAT	Versatec data files
	*.PAT	Versatec plot files
	gpifo.a	plotting routines to link with ispc
	gpico.a	plotting routines to link with ispc
	gdf_script	script file to process a GDF file and output a Versatec plot file
	ftpplot	script file to send a plot file to the Versatec vps node via ftp
	cadam_script	script file to process CADAM drawings and print out to the gdf print queue
	gpimain.o	plotting routines to link with ispc
	pentbl.fil	sample GDF color to plotter color mapping table
/usr/lpp/gai/buc_* /usr/lpp/gai/mca_* /usr/lpp/gai/pci1410*	load3dm3	graPHIGS API code for each 3D graphics adapter
	GPpipeline	graPHIGS API code for each 3D graphics adapter
	GPrasterizer	graPHIGS API code for each 3D graphics adapter
/usr/lpp/graPHIGS/bin	gP	command to start a remote graPHIGS API nucleus
	gpterm	command to stop a remote graPHIGS API nucleus
	gphost	command to add or delete host names permitted to access a remote graPHIGS API nucleus
	gpq	command to inquire the status of remote graPHIGS API nuclei
	gPafutil	command to run the gPafutil archive utility
	gPI	graPHIGS API run time code for MP system support
	loadafmae0	graPHIGS API run time code
	loadafmat0	graPHIGS API run time code
	loadafmh0	graPHIGS API run time code

Directory	Name	Use
	loadafmn0	graPHIGS API run time code
	loadxd	graPHIGS API run time code
	loadafmu0	graPHIGS API run time code
	loadcor	graPHIGS API run time code
	loads3d	graPHIGS API soft graphics code
	loadgpshm	graPHIGS API soft graphics code
	ls6098	inquire connection information for the 6098 with FDDI feature
	mkgPcon	make or break a connection with a 6098 with FDDI feature
/usr/lpp/graPHIGS/lib	libgPz.a	graPHIGS API shared library
/usr/lpp/graPHIGS/fonts	afm*.sym	geometric text fonts
/usr/lpp/graPHIGS/etc	runivp	installation verification script
	ivp	installation verification program
	PROFILE	sample graPHIGS API profile
	README.gP	graPHIGS API readme file
	Xdefaults	sample X defaults file
	extensions	run time information
/usr/lpp/graPHIGS/samples		graPHIGS API sample programs
/usr/lpp/graPHIGS/clients/gPtutor		graPHIGS API tutorial
/usr/lpp/graPHIGS/clients/fonteditor		graPHIGS API font editor
/usr/lpp/graPHIGS/clients/slicer_iso		slicing and contouring demo
/usr/lpp/graPHIGS/clients/gPdbg		sample graPHIGS API debugger
/usr/lpp/info/data/graPHIGS		graPHIGS documentation

Chapter 3. Customizing the graPHIGS API

Customizing for Optional Environments

The graPHIGS API is contained within three separate packages:

- PEX_PHIGS.dev
- PEX_PHIGS.rte
- PEX_PHIGS.adt

The device-dependent software is contained in the **PEX_PHIGS.dev** package. The graPHIGS run-time environment is contained in the **PEX_PHIGS.graPHIGS.rte** package. The graPHIGS application development toolkit is contained in the **PEX_PHIGS.graPHIGS.adt** package. The packages may contain optionally installable subsystems.

You select the options you want to install during the **smit** installation process. A list of the environments will be displayed. You can choose from this list the ones you want to install:

PEX_PHIGS.dev

PEX_PHIGS.dev.mca.8fbc	PEX/graPHIGS GXT1000 Device Dependent Software
PEX_PHIGS.dev.mca.8ee3	PEX/graPHIGS GT4 Device Dependent Software
PEX_PHIGS.dev.mca.8f61	PEX/graPHIGS GXT800M Device Dependent Software
PEX_PHIGS.dev.buc.00004002	PEX/graPHIGS GXT500 Device Dependent Software
PEX_PHIGS.dev.pci.14103c00	PEX/graPHIGS GXT250P/GXT255P Device Dependent Software
PEX_PHIGS.dev.pci.14105400	PEX/graPHIGS GXT500P/GXT550P Device Dependent Software
PEX_PHIGS.dev.pci.14105e00	PEX/graPHIGS GXT800P Device Dependent Software
PEX_PHIGS.dev.pci.14106e01	PEX/graPHIGS GXT4000P Device Dependent Software
PEX_PHIGS.dev.pci.14107001	PEX/graPHIGS GXT6000P Device Dependent Software
PEX_PHIGS.dev.pci.14108e00	PEX/graPHIGS GXT3000P Device Dependent Software
PEX_PHIGS.dev.pci.1410b800	PEX/graPHIGS GXT2000P Device Dependent Software

PEX_PHIGS.graPHIGS.rte

PEX_PHIGS.graPHIGS.rte.base	graPHIGS Base Runtime Environment (includes C and Pascal header files)
PEX_PHIGS.graPHIGS.rte.pipe	graPHIGS Pipeline Runtime Environment
PEX_PHIGS.graPHIGS.rte.6098	graPHIGS 6098 Support
PEX_PHIGS.graPHIGS.rte.soft	graPHIGS Soft Runtime Environment
PEX_PHIGS.graPHIGS.rte.rnuc	graPHIGS Remote Nucleus Support
PEX_PHIGS.graPHIGS.rte.plot	graPHIGS IBM/CALCOMP/VERSATEC Plotter Support

PEX_PHIGS.graPHIGS.fnt

PEX_PHIGS.graPHIGS.fnt.JP	graPHIGS Kanji (Japanese) Fonts
PEX_PHIGS.graPHIGS.fnt.KR	graPHIGS Hangul (Korean) Fonts
PEX_PHIGS.graPHIGS.fnt.SC_EUC	graPHIGS Simplified Chinese Font, EUC based
PEX_PHIGS.graPHIGS.fnt.TW	graPHIGS Traditional Chinese Fonts
PEX_PHIGS.graPHIGS.fnt.uni	graPHIGS Unicode Fonts

PEX_PHIGS.graPHIGS.adt

PEX_PHIGS.graPHIGS.adt.include	graPHIGS Application Development Toolkit Include Files
PEX_PHIGS.graPHIGS.adt.gks	graPHIGS Application Development Toolkit GKS Library
PEX_PHIGS.graPHIGS.adt.samples	graPHIGS Application Development Toolkit Samples
PEX_PHIGS.graPHIGS.adt.tutor	graPHIGS Application Development Toolkit Tutorial
PEX_PHIGS.graPHIGS.adt.clients	graPHIGS Application Development Toolkit Clients

When installation is complete, the system displays this message:

```
The installation of n completed successfully
```

where *n* is the environment option you chose.

Customizing for File Format Conversion

When you select **PEX_PHIGS.graPHIGS.rte.plot**, the **cvtgdf** utility is installed in the **/usr/bin** directory. This utility converts a file that is in GDF format (that is, a file generated by a GDF workstation on RS/6000) to ADMGDF format (the format produced by a GDF workstation running under VM or MVS). For more information on using this utility, see Appendix C in *The graPHIGS Programming Interface: Technical Reference*.

Customizing for a Remote graPHIGS API Nucleus

When you select the **PEX_PHIGS.rte.rnuc** option, the files necessary for remote graPHIGS API nucleus support are installed. The files **gPinit**, **gPterm**, **gPhost** and **gPq** are installed and linked to the **/usr/lpp/graPHIGS/bin** directory. In addition, the four corresponding script files are installed and linked to the **/usr/bin** directory. **gPterm** is used to terminate the remote graPHIGS API nucleus. **gPhost** is used to add or delete host names permitted access to the remote graPHIGS API nucleus. **gPq** is used to inquire the status of remote graPHIGS API nuclei.

The **/etc/services** file contains the base port number to be used by remote graPHIGS API nuclei. The default base port number is 8000. If this port number is already reserved for another purpose in the **/etc/services** file, a warning is issued. The **/etc/services** file is *not* automatically modified to reserve port 8000. However, the default base port number remains 8000 until you modify it in the **/etc/services** file.

To change the base port number, change the following line in the **etc/services** file:

```
graPHIGS        8000/tcp
```

where 8000 is the default base port number. To change the base port number to, for example, 9000, change the above line as follows:

```
graPHIGS        9000/tcp
```

Note: For remote graPHIGS API nucleus support to work over a network, the base port number must be the same for all IBM RS/6000 workstations that are using remote graPHIGS nuclei.

See *The graPHIGS Programming Interface: Technical Reference* for more information on remote graPHIGS API nuclei.

Customizing for Plotters

The graPHIGS API provides support for IBM, Hewlett Packard, CalComp, and Versatec plotters when **PEX_PHIGS.graPHIGS.rte.plot** is installed.

Customizing for IBM Plotters

Follow the steps below to enable output to an IBM plotter attached to an IBM RS/6000.

1. Define and enable the port to which the plotter is attached.

   Issue the **smit** command to set up a port for the plotter. See *AIX 5L Version 5.2 Commands Reference* for more information on the **smit** command.

   Set up the plotter as a TTY device.

   ```
   Example: smit
            Devices
            TTY
            Add a TTY
              - choose tty rs232 Asynchronous Terminal
              - select the appropriate adapter (e.g., sa0, sa1)
              - set the following values:
                  PORT number                          [s1]  (or s2, etc.)
                  Enable Login                         disable
                  BITS per Character                   8
                  Number of STOP BITS                  1
                  FLOW CONTROL to be used              [xon]
                  STATUS of device at boot time        [available]
              - set BAUD rate and PARITY fields to the
                appropriate values for the plotter
   ```

   The **smit** command returns a message indicating what TTY device name you have defined (e.g., *tty0*). This name will be used in later steps.

2. If you will be using the queuing system to schedule plots, you need to define a print queue for the plotting backend programs. If queuing will not be used, this step can ignored. Refer to *The graPHIGS Programming Interface: Technical Reference* for details on executing the plotting programs both with and without queuing.

   Issue the **smit** command to define a print queue and backend program to be used by the **qdaemon** queue manager.

   ```
   Example: smit
            Print Spooling
            Local Printer Queues
            Add a Print Queue
            other User Defined Backend
              - set the following values:
                  NAME of queue to be added          [gdf]
                  NAME of device to be added         [gdfplt]
                  BACKEND PROGRAM pathname           [/usr/lpd/cpsI1 -tty0]
              - the queue and device names may be set to whatever
                you choose. The backend program may be one of
                the three provided by graPHIGS (cpsI1, cpsC1,
                or cpsV1) or one of your own (see the section
                below on customizing the system for non-IBM
                plotters).
              - any options which will always be passed to the
                backend may be added following the backend
                program pathname (e.g., tty0 in the above
                example).
   ```

For information on using the plotter run time environment, refer to Appendix C in *The graPHIGS Programming Interface: Technical Reference*.

IBM Plotters Supported

Category	Model	Description
Pen Plotters	6180	8 pen, A size
	6182	8 pen, A & B size, auto-feed
	6184	8 pen, C & D size

Category	Model	Description
	6186-1	8 pen, A-E size
	6186-2	8 pen, A-E size, roll-feed
	7371	2 pen, A size
	7372	6 pen, A & B size
	7374	8 pen, A-D size
	7375-1	8 pen, A-E size
	7375-2	8 pen, A-E size, roll-feed

Customizing for CalComp and Versatec Plotters

When **PEX_PHIGS.graPHIGS.rte.plot** is installed, CalComp and Versatec plotting backend programs are installed on your system. Refer to CalComp and Versatec instructions for information on how to set up and run these backend programs. CalComp instructions are found in the file **/usr/lpd/README.ccp**, and Versatec instructions are in the file **/usr/lpd/README.VERSA** Refer to *The graPHIGS Programming Interface: Technical Reference* for additional information.

CalComp Plotters Supported

Category	Model	Description
Pen Plotters	1023	8 pen, 'D' cut sheet 30/42 ips
	1025	8 pen, 'E' cut sheet 30/42 ips
	1026	8 pen, 'E' cut sheet/roll 30/42 ips
	1043 DM	8 pen, 'E' cut sheet/roll 24 ips
	1044	8 pen, 'E' cut sheet/roll 24 ips
	1076C	4 pen, 'E' cut sheet 50/70 ips
	1077	4 pen, 'E' cut sheet/roll 50/70 ips
Electrostatic Raster Plotters	57424	24-inch wide, 400 dpi — monochrome
	57436	36-inch wide, 400 dpi — monochrome
	57444	44-inch wide, 400 dpi — monochrome
	58424	24-inch wide, 400 dpi — color
	58436	36-inch wide, 400 dpi — color
	58444	44-inch wide, 400 dpi — color
Thermal Raster Monochrome (2-Color) Plotters	52224	DrawingMaster 24-inch wide, 200 dpi
	52236	DrawingMaster 36-inch wide, 200 dpi
Thermal Raster Color Plotters	5902A	PlotMaster 'A', 200 dpi
	5902AE	PlotMaster 'A4', 200 dpi
	5912	Colorview 'B'/'A3', 200 dpi
	5913	Colorview 'B'/'A3', 300 dpi

Versatec Plotters Supported

Category	Model	Description
Electrostatic Plotters	V80	11-inch — monochrome
	7222	22-inch — monochrome
	7422	22-inch — monochrome
	7224	24-inch — monochrome
	7424	24-inch — monochrome
	7236	36-inch — monochrome
	7244	44-inch — monochrome
	7444	44-inch — monochrome
	C2552	11-inch — color
	C2562	11-inch — color

Category	Model	Description
	C2558	11-inch — color
	CE3224	24-inch — color
	CE3424	24-inch — color
	CE3225	24-inch(A1) — color
	CE3425	24-inch(A1) — color
	CE3236	36-inch — color
	CE3436	36-inch — color
	CE3244	44-inch — color
	CE4444	44-inch — color
	ECP42	42-inch — color
	8924	24-inch
	8936	36-inch
	8944	44-inch
Thermal Plotters	C2756-vv	11-inch
	C2766-vv	11-inch
	C2726	A size
	C2736	A4 size
	C2756	B size
	C2766	A size
Laser Plotters	8836	36-inch

Customizing for Plotters with HP GL2 support

Many new plotters include support for HP GL2** language and by default set XOFF/XON on. If you are configuring your plotter queue for cpsl1, you need to disable XOFF/XON support at the plotter. If you are configuring your plotter queue for HP GL2 support (using the cgm2hp2 backend), you need to set XON/XOFF on in the TTY port definition.

Follow the steps below to enable output to a plotter supporting HP GL2 that is attached to an IBM RS/6000.

1. Define and enable the port to which the plotter is attached.

   Issue the **smit** command to set up a port for the plotter. See *AIX 5L Version 5.2 Commands Reference* for more information on the **smit** command.

   Set up the plotter as a TTY device.

   Example:
   ```
   smit
            Devices
            TTY
            Add a TTY
              - choose tty rs232 Asynchronous Terminal
              - select the appropriate adapter (e.g., sa0, sa1)
              - set the following values:
                  PORT number                          [s1]  (or s2, etc.)
                  Enable Login                         disable
                  BITS per Character                   8
                  Number of STOP BITS                  1
                  FLOW CONTROL to be used              [xon]
                  STATUS of device at boot time        [available]
              - set BAUD rate and PARITY fields to the
                appropriate values for the plotter
   ```

   The **smit** command returns a message indicating what TTY device name you have defined (e.g., *tty0*). This name will be used in later steps.

2. If you will be using the queuing system to schedule plots, you need to define a print queue for the plotting backend programs. If queuing will not be used, this step can ignored. Refer to *The graPHIGS Programming Interface: Technical Reference* for details on executing the plotting programs both with and without queuing.

Issue the **smit** command to define a print queue and backend program to be used by the **qdaemon** queue manager.

Example:
```
smit
        Print Spooling
        Local Printer Queues
        Add a Print Queue
        other User Defined Backend
          - set the following values:
              NAME of queue to be added          [plot]
              NAME of device to be added         [cgm]
              BACKEND PROGRAM pathname           [/usr/lpd/cgm2hp2 -f /dev/tty0]
          - the queue and device names may be set to whatever
            you choose.
          - any options which will always be passed to the
            backend may be added following the backend
            program pathname (e.g., tty0 in the above
            example).
```
For information on using the plotter run time environment, refer to Appendix C in *The graPHIGS Programming Interface: Technical Reference*.

Industry Standard Plotting Interface (ISPI)

The graPHIGS API plotting routines can make Industry Standard Plotting Interface (ISPI) calls, instead of generating IBM, CalComp, or Versatec plotting data. This function is invoked by passing *-ispi* as an option to your backend program.

If you have an ISPI package, you can link-edit this with the graPHIGS API plotting object code to produce your own backend program. Your ISPI package must include the following industry standard subroutines:

PLOTS	Initializes software
NEWPEN	Changes current pen
SYMBOL	Draws text
PLOT	Draws lines, moves pen, terminates software

The following FORTRAN unit numbers are used by the graPHIGS plotting routines:

unit 6	log file
unit 8	input GDF file
unit 10	pentbl.fil (if *-pens* is specified)

The graPHIGS API plotting routines that can be link-edited with your ISPI package are contained in the following files:

- **/usr/lpd/gpimain.o**
- **/usr/lpd/gpifo.a**
- **/usr/lpd/gpico.a**

All three files must be linked with your ISPI package. Information on link-edit procedures can be found in the documentation for your compiler or run time environment or in the *AIX 5L Version 5.2 Commands Reference* under the **ld** command.

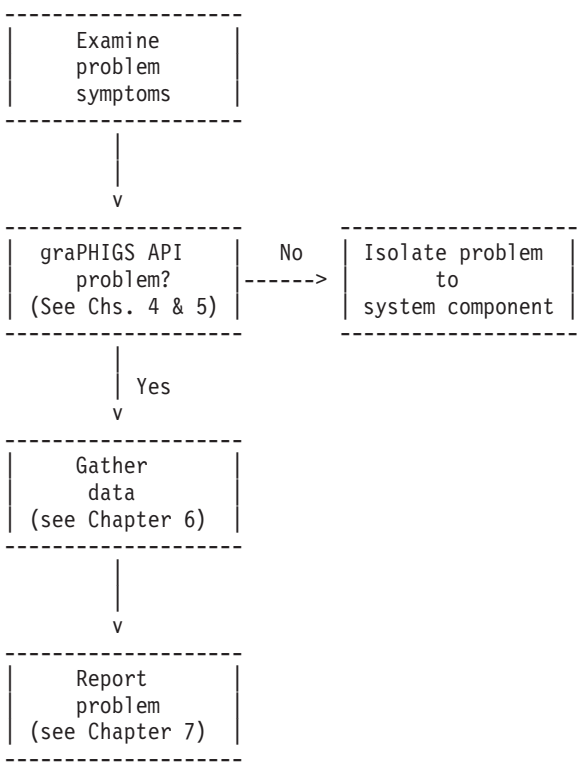
Example:
```
     xlf /usr/lpd/gpimain.o /usr/lpd/gpifo.a /usr/lpd/gpico.a  \
         /example/ispilib.o -lm -lqb -o
filename
```

where **/example/ispilib.o** is the name of your ISPI library object file, and *filename* is the name of the executable file you are creating. *Filename* can then be run directly, or used as a backend program for a print queue.

See Appendix C in *The graPHIGS Programming Interface: Technical Reference* for more information on plotting using the ISPI interface.

Chapter 4. Overview of Problem Diagnosis for the graPHIGS API

The following chart provides a broad outline of the problem diagnosis procedure for the graPHIGS API. Bypass this chapter if you know your problem is within the graPHIGS API and if you have isolated the symptoms of your program failure.



This chapter contains the basic procedures used to identify problems in the hardware and operating systems that the graPHIGS API uses. First, examine the following areas to determine whether they function properly. Programs using the graPHIGS API can fail because of problems with any of the following:

- The application program
- The installation of the graPHIGS API
- The hardware
- The operating system

After reviewing these problems, gather as much supporting data as possible to help you isolate the symptoms of the failure. You will include this information in your report to IBM.

Programming Problems

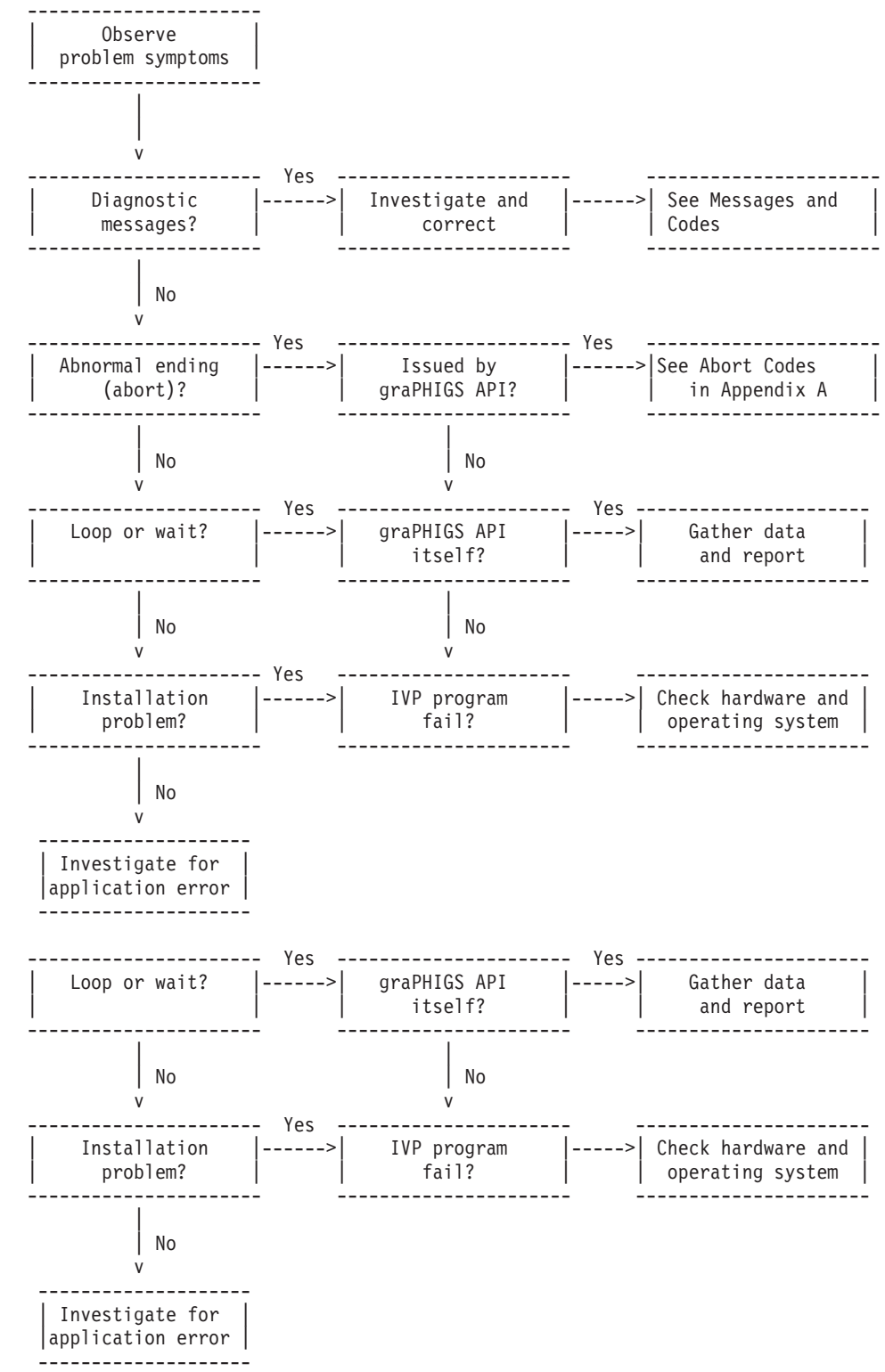
Programming problems create symptoms in many ways. Messages on the display or in the error file, abnormal termination, or improper operation are general categories of program failures that are discussed in this section.

Diagnosis begins by observing the symptoms of the suspected problem. These symptoms include:

- Error messages
- Aborts

- Incorrect output
- Blank displays
- Similar indications that your program is operating abnormally

The following flowchart is a quick reference for the diagnosis process. The brief descriptions provided in the chart are discussed in detail in the sections that follow.



Is the graPHIGS API Sending You Diagnostic Messages?

Errors Logged in Your Error File

The graPHIGS API ordinarily defaults to logging diagnostic messages in a designated error file. The graPHIGS API designated error file is found in the current working directory at the time of the error. Investigate all diagnostic messages logged in this file to determine whether they point to your observed symptoms.

The name of the error file is chosen by the application with the Open graPHIGS (**GPOPPH**) subroutine. If no error file is specified (e.g., if your application specified blanks for the file name) or if the error file is not usable (e.g., if the user does not have write authority in the current directory), messages are written to **stderr**, which defaults to the console.

Message Format

Diagnostic messages are displayed in the error file as follows:

```
1                       2       3         4
Fri Jul 10 11:17:49 1992 GPLT    AFM0063   LINETYPE VALUE < ONE
```

These messages contain the following information:

1 - Date and Time
Displays the date and time of the error.

2 - Subroutine
Name of the graPHIGS API subroutine associated with the error.

The graPHIGS API subroutine name is provided to help identify a graPHIGS API subroutine call that may have caused the error. For most errors, this name is the graPHIGS API subroutine that your application called.

The subroutine name field contains asterisks if the error is not directly related to any API subroutine call (such as an asynchronous hardware error).

An asterisk follows the subroutine name to indicate that the identified subroutine name may not be the actual subroutine call issued by your application that may have caused the error. For example, the error may be detected long after the invoked graPHIGS API subroutine has returned to your application and other subroutines have been called. The graPHIGS API provides the name of the subroutine that was most likely invoked when the error occurred. The related subroutine name (followed by an asterisk) is provided in the error message. Refer to *The graPHIGS Programming Interface: Messages and Codes* for a table of related subroutines.

3 - Message Number
Identifier of the message found in the manual *The graPHIGS Programming Interface: Messages and Codes*

4 - Message Text
A brief summary of the error condition. *The graPHIGS Programming Interface: Messages and Codes* manual contains the following information about the error:

- An explanation of why the error was issued
- A statement of what action the graPHIGS API has taken, if any, as a result of the error
- A statement outlining actions the programmer can take to resolve the problem

Note: To isolate the sequence of the graPHIGS API subroutine calls that generated the error, it may be necessary to rerun the application with the graPHIGS API trace active.

Overview of Message and Error Numbers

The following is a general outline of diagnostic messages and error numbers. See *The graPHIGS Programming Interface: Messages and Codes* for further error information and for help in debugging problems reported in the error file.

Message Numbers

The message number provided in each message identifies each unique error message text and identifies the additional information in the error message descriptions that is provided in *The graPHIGS Programming Interface: Messages and Codes*. The message numbers are arranged in groups of related messages.

Messages 1-899	Base graPHIGS API messages for errors that are device independent. These messages generally identify errors in the parameters passed to the graPHIGS API subroutines.
Messages 1000-1299	System Service messages generated by errors in the internal service utilities.
Messages 1300-1399	Distributed Application Process (DAP) messages generated from errors while processing a distributed application.
Messages 2000-2999	Device-dependent messages for errors detected during processing for a specific workstation.

Error Numbers

One of the parameters provided to your first level error handling routine is an error number. See the **GPEHND** subroutine in *The graPHIGS Programming Interface: Subroutine Reference* for a complete description of the interface and parameters passed to your error handling routine. The error number identifies the error that has occurred and reveals the severity of the error. This can assist you in determining how your error handling routine should react to the error.

The error number provided to your error handling routine is related to, but is separate from, the message number of the error message text associated with the error. For most errors, the error number and the message number are the same number. Only for the Device Support errors (900-999) are the error numbers different from the message numbers. For the Device Support errors, the error numbers are in groups that indicate the severity of the error. See the section on Errors 900-999 DEVICE SUPPORT or, for more information, *The graPHIGS Programming Interface: Messages and Codes*). Using this grouping, your application's error handling routine can process workstation errors in general categories without having to deal with each workstation-specific error that can occur across many different workstations. The message number associated with each Device Support error is unique. It identifies the unique message text associated with the reported error.

The graPHIGS API error numbers are divided into the following ranges:

ERRORS 1-899 - BASE graPHIGS API

The errors in this range are device independent; they are common to all devices. Most errors are found in this category, including errors generated by parameter validation. Note that for errors in this range, the message number is the *same as* the error number. There are two types of system actions in this category: warning and ignoring function.

Warning	A default action may be taken, causing the result to differ slightly from the intended action.
Ignoring function	No graphical data or state list is modified nor is any workstation affected. The resulting effect is as though the subroutine was never called.

ERRORS 900-999 - DEVICE SUPPORT

The errors in this range are detected during processing for a specific workstation. The types of errors detected include I/O errors, workstation capacity exceeded errors, and device or host service errors. Note that for errors in this range, the message number is *different* from the error number. The device support errors are subdivided into the following error classes:

900-909 - Informational	A minor error has been detected or a previous error condition has been cleared. Processing continues to normal completion although default actions may be taken as indicated by the message.

910-919 - Warning	An inconsistent state has been detected. This condition is not severe. No information has been lost and the device support may be able to clear the inconsistency at a later time. The requested function has been logically completed, and processing continues to normal completion although default actions may be taken as indicated by the message.
920-929 - Error	The device support was unable to complete the requested function, either logically or physically. If the requested operation was a non-structure modification function, the device support will attempt to back out of the operation so the environment can return to its previous state. Non-structure modification functions are effectively ignored since the action has been reset. Structure modifications cannot be removed from the workstation that detects the error because they are device independent and may be completed on other workstations. Therefore, it will probably be impossible for the application to eliminate inconsistency without closing and reopening the workstation.
930-939 - Severe Error	An inconsistency has been detected between an actual device and the current graPHIGS API state (workstation state list or graphic data content). After an error of this type, the workstation state is unknown. The application should not invoke any function involving this workstation except **Close Workstation** or any inquiry function. This is only a recommendation—other subroutine calls are not prevented but can produce unpredictable results. The application may attempt to return the workstation to a determinate state by closing and reopening the workstation.
940-949 - Terminating Error	The device support has detected an unrecoverable error and cannot accept any further requests. For example, this type of error is generated if the **Open Workstation** function cannot be completed or if corrupted control blocks are detected. If open, the workstation should be closed immediately by your application.

ERRORS 1000-1299 - SYSTEM SERVICE ERROR

These messages are generated by the graPHIGS API service utilities. Examples of these errors include initialization, incorrect number of parameters passed on a subroutine, storage request failures, and file I/O errors. Note that for errors in this range, the message number is the *same as* the error number.

The state of the system after the error, is dependent on the error conditions. Some errors have no effect on the system; other errors may be unrecoverable, causing the application to terminate.

Inquiry Errors in the errind parameter

If an error is detected during the processing of an inquire subroutine call, a diagnostic message is not returned to the error file; rather, an error number is returned as the error indicator *errind* parameter.

Your application may be relying on information assumed to have been returned by a graPHIGS API inquiry subroutine. If the information requested by an inquiry is unavailable for any reason, the *errind* parameter is returned as a non-zero value. Make sure that the application always checks for zero value before using the requested information.

Error numbers are listed with the *errind* parameter description of each inquiry subroutine in *The graPHIGS Programming Interface: Subroutine Reference*. (Note that in addition to the errors listed with each subroutine description, it is possible to get additional errors reported when the error occurs within an internal service of the graPHIGS API). See *The graPHIGS Programming Interface: Messages and Codes* for a description of the error numbers provided by the graPHIGS API in the *errind* parameter.

National Language Support

If an input method (IM) error is detected during the processing of a graPHIGS API string device, a system service diagnostic message is returned to the error file.

See *The graPHIGS Programming Interface: Technical Reference* for more information on ensuring that font files have been correctly installed.

Handling Error Messages

It is possible that graPHIGS API is attempting to send error messages to you, but your application is suppressing them. Your application (and the user) can control where error messages are displayed, or whether they are displayed at all. Your application can provide recovery routines invoked as error handling routines for processing the error.

Using Specify an Error Exit and Error Threshold (**GPEXIT**), you can define a level 1 error handler in your system. However, the graPHIGS API will log the error only upon your request, which is done by issuing Error Logging (**GPELOG**). When an error condition occurs, the level 1 error handler is invoked. An error number indicating the severity of the error is passed to the error handler. You can request that the message be logged.

You can have your application handle errors in its own way. Instead of logging the error, it can take other actions. See *The graPHIGS Programming Interface: Writing Applications* for a detailed description of error processing and handling.

Has an Abnormal Termination (ABORT) Occurred?

If an error is severe enough, the graPHIGS API terminates processing. Termination occurs using the **ABORT** service. **ABORT** completion codes indicate that an abnormal condition, generally one that is unrecoverable, has occurred. An **ABORT** completion code may also indicate an error in the way the application is using the graPHIGS API.

graPHIGS API

When the graPHIGS API issues an abort, the message `AFM1204 graPHIGS API ABORT CODE=xxx` is written to **stderr**, that provides the **ABORT** code for the error that occurred. See Appendix A to determine the issuing module and the reason for the abort.

Other

A core file results when various errors occur in a terminated process. Common errors are *memory violations* and *illegal instructions*. The core file is written in the process's working directory if there is sufficient space and the user has write authority.

Is There Evidence of a Loop or a Wait?

The use of a separate shell and nucleus has made it more difficult to determine which subroutine call caused a loop or wait. Synchronize (**GPSYNC**) should be used in conjunction with the graPHIGS API trace facility to show which subroutine has caused the loop or wait. To avoid having to place calls to**GPSYNC** after every graPHIGS API subroutine in your application, refer to *The graPHIGS Programming Interface: Writing Applications* for a method of using **GPSYNC** in a second-level error handling routine to force requests to the nucleus after every graPHIGS API subroutine.

Is Your System Installed Correctly?

When the graPHIGS API is installed, files are put in the file system. Check to be sure that these files exist in each of the specified directories.

To help rule out system and hardware problems, run a program that you know works correctly. Review your application program to make sure that all subroutines are used correctly. Verify that all values that are

not explicitly set by your application have an acceptable default. See *The graPHIGS Programming Interface: Writing Applications* for possible programming pitfalls.

Chapter 5. Problem Determination for gPgated, the graPHIGS API Gateway Daemon

If you encounter a problem when attempting to run a graPHIGS API application or connect to a graPHIGS API gateway, follow this procedure:

Step 1.

Check that **gPgated**, the graPHIGS API gateway daemon, is running and that there is a run time connection profile entry which associates an IBM S/370 device address with the target remote graPHIGS API nucleus. Use the **lsgPcon** command to verify that the entry exists. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **lsgPcon** command.) This may return one of a number of different messages depending on your connection, nucleus, and general workstation state. Messages and appropriate actions are as follows:

Message	Action
lsgPcon AFM0602 No connection exists on path *hostname:nucid*	Go to **Step 2**.
device address hostname:nucid DEFINED	Verify that this is the run time connection profile entry desired. If so, go to **Step 3**. If not, go to **Step 2**.
device address hostname:nucid OPEN	This device address is in use. You may wish to try another device address.
lsgPcon unknown hostname *hostname*	Go to **Step 4**.
lsgPcon AFM0604 gPgated *hostname* NOT STARTED OR NOT RESPONDING	Go to **Step 5**.
gPhost AFM0604 NUCLEUS gP *xxx* NOT STARTED OR NOT RESPONDING	Go to **Step 6**.

Step 2.

Use the **chgPcon** command to add a run time connection profile entry, associating an IBM S/370 device address with the target remote nucleus. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **chgPcon** command.)

Step 3.

Use the **gPhost** command to determine whether the graPHIGS API gateway daemon is allowed to communicate with the target remote nucleus. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **gPhost** command.)

If the host on which the graPHIGS API gateway daemon is running is *not* in the list returned by **gPhost**, use the **gPhost** command to add the host. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **gPhost** command.)

Then go to **Step 7**

Step 4.

You have a TCP/IP configuration problem. The host where the graPHIGS API gateway daemon is running is not defined. Contact your network administrator.

Step 5.

The graPHIGS API gateway daemon does not appear to be running. Check to see if the specified host is the desired gateway. If so, use the **gPgated** command to restart the graPHIGS API gateway daemon. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **gPgated** command.)

Step 6.

Use the **gPq** command to determine whether the remote graPHIGS API nucleus is running on

your workstation. If there is no response, execute the **gPinit** command to start the remote graPHIGS API nucleus. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **gPq** and **gPinit** commands.)

Then go to **Step 3**

Step 7.
Restart your GDDM/graPHIGS API application.

Expected Behavior Under Problem Conditions

Under problem conditions, several types of system failures are possible. Following is a list of potential problems and the expected default system behavior:

- **Application abnormal termination**
  - On MVS and MVS/XA, GAM closes the connection to your graPHIGS nucleus, and then the workstation and its resources are freed.
  - On VM and VM/XA, GAM may not get control to close the connection, so the workstation and its resources may appear to ″hang″ until the application is restarted. GAM open processing causes the workstation and its resources to be freed and then open processing proceeds.
- **IBM S/370 failure**

  This failure does not affect the graPHIGS API gateway daemon or any connections. If a 'system reset' is required, warning messages are recorded at the host where the graPHIGS API gateway daemon is running.
- **IBM 5088 or 6098 failure**

  Depending on the kind of failure, there may be no greater impact than an IBM S/370 failure, as described above. Run the utility program **panel20** at the host where the graPHIGS API gateway daemon is running. (See the *5080 Graphics System Installation, Operation, and Problem Determination* for more information about the **panel20** command.)

  If the link status code is 0, no action is required. If the status for a connection is not as expected, the graPHIGS API gateway daemon may need to be restarted with the **gPgated** command. (See *The graPHIGS Programming Interface: Technical Reference* for more information on the **gPgated** command.)

  If the IBM 5088 or 6098 is restarted, *not-ready-to-ready* interrupts are received at the IBM S/370.
- **Host Interface Adapter (HIA) failure**

  This is indicated by a link status of 2 (adapter error) in the utility program **panel20**. This is a terminating error and requires the graPHIGS API gateway daemon to be terminated and the adapter to be reconfigured. This may be done simply by shutting down, or by the following command sequence:

  ```
  rmdev -l hia0
  mkdev -l hia0
  ```

  Error log information from the **errpt** command should be reported to your service representative. Once the graPHIGS API gateway daemon is restarted and the connections are re-established, *not-ready-to-ready* interrupts are received at the IBM S/370.
- **gPgated failure**

  In the event of a graPHIGS API gateway daemon abend, there is a ″warmstart″ option that allows the gateway to be restarted with the same run time connection profile that existed at the time of failure. Once the connections are re-established, *not-ready-to-ready* interrupts are received at the IBM S/370.
- **TCP/IP network failure**

  Depending on the nature of the failure, one of two things may occur:
  - If a specific link is broken, it appears as if it were a workstation failure as described below, with the potential for more than one workstation to be involved
  - The following error may be logged in the gateway transaction file:

    ```
    AFM0593 COMMUNICATION ERROR: MAJOR 7, MINOR
    errno
    ```

In either case, the IBM S/370 device address associated with the connection receives an asynchronous error (Attention-UC).

- **Personal graPHIGS remote nucleus or user workstation failure**

  The following error may be logged in the gateway transaction file:

  ```
  AFM0204 NUCLEUS CONNECTION FAILED
  DEVICE ADDRESS
  xxx, NUCLEUS ':0' MAJOR 7, MINOR
  errno
  ABNORMALLY terminated
  ```

  The associated IBM S/370 device address receives an asynchronous error (Attention-UC) and the connection remains `OPEN`. At this point, the application should react to the error and disconnect the nucleus.

Using the Trace Facility

The graPHIGS API gateway daemon is capable of using the system trace facility. If detailed problem determination is required, your support representative may ask you to generate a trace file.

To generate a trace file, follow these steps:

1. Log on as super user to the host at which the graPHIGS API gateway daemon is running.
2. Invoke the trace command as follows:

   ```
   trace -aj280,281,282
   ```

3. Run the application to the point of error.
4. Turn trace off by issuing the following command:

   ```
   trcstop
   ```

5. Format the trace data by issuing the following command:

   ```
   trcrpt -o trace.file
   ```

See The graPHIGS API Trace Facility for more information on the graPHIGS API trace facility.

Chapter 6. Collecting Supporting Documentation

This chapter documents product implementation information provided by the graPHIGS API for collecting support documentation. See Notices for information on the use of product implementation information.

If you decide that your problem is caused by the graPHIGS API, you need to gather supporting documentation so that IBM can correct your problem. This chapter focuses on collecting the necessary information. For more information on tracing the graPHIGS API gateway daemon see Using the Trace Facility.

The graPHIGS API Trace Facility

Facilities

The graPHIGS API provides an internal tracing capability to assist in isolating problems. This facility enables you to track the internal flow at the component level, as well as at key internal processing points. The resulting trace file can be used either to help isolate the routine generating the error, or as part of the data submitted to IBM in your problem report.

Background

A few terms need to be defined before you start to use the trace facility. The term *shell* refers to the graPHIGS API shell that is linked with your application. The term *nucleus* refers to the graPHIGS API nucleus, which may or may not be linked with your application. If the nucleus is linked with your application, it is called a *private nucleus* If the shell is connected to a nucleus that is running as a separate process, it is called a *remote nucleus*. For more information on the graPHIGS API shell/nucleus concept, see *The graPHIGS Programming Interface: Understanding Concepts*.

Specification

Tracing is controlled through the trace control word, a fullword integer that defines the kind of trace produced. If the shell is to connect to a private nucleus, trace control may be specified through:

- The Internal Trace Control (**GPTRCE**) subroutine (see *The graPHIGS Programming Interface: Subroutine Reference* for more information).
- A default specification in the External Defaults File (EDF) or Application Default Interface Block (ADIB) (see Setting Trace Defaults).

If the shell is to connect to a remote nucleus, trace control for the shell is specified the same way as listed above. Trace control for the remote nucleus may be specified as follows:

- Through a default in the External Defaults File (EDF) in the working directory of the remote nucleus (the current directory of the user who starts the nucleus).

  Note: The default specification must be set up before starting the remote nucleus.
- By specifying a parameter on the command to start the remote nucleus that will override what may have been set in the EDF file. See *The graPHIGS Programming Interface: Technical Reference* for a complete discussion on remote nuclei.

Trace Control Word

By default, the trace control word is set to `0`, which means that tracing is turned off. The trace control word has the following format:

Byte 0	**Flags**	
	1...	Set to 1 to make the trace word unchangeable. If this bit is set, any further changes to the trace word is ignored, including subsequent subroutine calls to **GPTRCE**. This bit can be used to “force” a setting of trace (typically using a default specification). This may be of use if an application already contains **GPTRCE** subroutine calls and if the source of the application is not readily available. A trace word with this bit set in the user-defined External Defaults File (EDF) takes priority over any trace word set in the Application Default Interface Block (ADIB). This allows the user control over the application setting of trace. See Setting Trace Defaults for more information. Specifying a parameter to initiate trace on the command to start a remote nucleus will override the trace word set in the External Defaults File (EDF).
	.000 0000	Reserved
Byte 1	Reserved	
Byte 2	Trace Qualifier	Reserved
Byte 3	Trace Level	
	0000 0000	Stop component and data entry
	1	Start trace of component entry and exit
	.... .1..	Start trace of data entry and exit

Note: Large quantities of trace output may be generated when the trace facility is used.

Format of Trace File Output

The trace file output for a remote graPHIGS API nucleus does not provide you with meaningful data. It is provided only to supply additional information for reporting a problem to IBM.

The beginning of each trace file for a shell contains a number of standard entries including the trace control word, the graPHIGS API defaults table contents, and additional information. The following example illustrates the format for standard data:

```
+-------------------------------------------------------------------------------+
|                                                                               |
| ORIGIN - /u/gus                                                               |
|                                                                               |
| TRACE WORD = '00000001'X   BUILDID = V2R1.0                                   |
|                                                                               |
| AFMTDFT  - GENERAL DEFAULTS TABLE COMMON SECTION                              |
| 20024BF8                        44465420  00000000    *        DFT     *     |
| 20024C00    20025108  41040100  00000064  00000001    *  Q A      d    *     |
| 20024C10    00000064  20025110  00000000  00000000    *   d  Q         *     |
| 20024C20    20025110  00000001  0000FFEC  0000FFEC    *  Q             *     |
| 20024C30    00004000  00000020  00000001  00000000    *  @             *     |
| 20024C40    00000000  00000000                        *                *     |
|          - SUBSYSTEM SPECIFIC DEFAULTS                                        |
| 20024C48                        00000000  20202020    *                *     |
| 20024C50    41464D54  52414345  41464D44  45465320    *AFMTRACEAFMDEFS *     |
| 20024C60    41464D30  30303031  41464D54  52414345    *AFM00001AFMTRACE*     |
| 20024C70    50524F46  494C4520  41464D44  45465320    *PROFILE AFMDEFS *     |
| 20024C80    41464D55  54312020  20202020  20202020    *AFMUT1          *     |
| 20024C90    20202020  20202020  20202020  20202020    *                *     |
| 20024CA0    20202020  20202020  20202020  20202020    *                *     |
| 20024CB0    20202020  20202020  20202020  41464D54    *            AFMT*     |
| 20024CC0    52414345  20202020  20202020  20202020    *RACE            *     |
| 20024CD0    20202020  20202020  20202020  20202020    *                *     |
```

```
| 20024CE0    20202020  20202020  20202020  20202020    *               *       |
| 20024CF0    20202020  20202020  20202020  20202020    *               *       |
| 20024D00    50524F46  494C4520  20202020  20202020    *PROFILE        *       |
+-------------------------------------------------------------------------------+
```

Trace output begins with the origin, the trace control word, and the graPHIGS API level identification. The origin is the path in which the process is currently running. The file also produces a formatted listing of the graPHIGS API defaults table.

When the appropriate level of tracing is enabled, one or more trace records are created for each entry and exit at the specified level(s)

Format of the Trace Record

Each trace record contains the following information:

Event Sequence Number

While trace is active (that is, while the trace control word is not 0), each module entry and exit is counted as a “trace event.” Each event is assigned a sequence number (starting at 1) which is listed in the trace output.

Trace Record Type

A mnemonic is used to indicate the type of trace record, as follows:

CPNIN Shows that a graPHIGS API component has been entered to perform the function listed.
CPNOUT Shows that a graPHIGS API component has exited after performing the function listed.
TRDIN Shows the input parameters being traced within a graPHIGS API component.
TRDOUT Shows the return parameters being traced within a graPHIGS API component.

Trace Explanation

The contents of the trace explanation vary according to the trace record types, as follows:

CPNIN, CPNOUT Trace explanation contains:

- The Request Control Parameter (RCP) in mnemonic and hexadecimal format.
- An interpretation of the RCP.

Setting Trace Defaults

The user may specify trace default values through an external file containing the User Default Specifications (UDS) or by the application programmer in an Application Default Interface Block (ADIB). Priorities of the trace word are discussed under Trace Control Word.

User Default Specifications (UDS) Processing

User Default Specifications (UDS) are accessed through a file called the External Defaults File (EDF). These default specifications appear at the top of the trace file, and can be specified without compiling or rebuilding the application.

The External Defaults File (EDF), which contains records consisting of the User Defined Specifications, must be named PROFILE *or* must be specified in the **gPPROFILE** environment variable. When the Open graPHIGS (**GPOPPH**) subroutine is called or when a remote graPHIGS API nucleus is started, the graPHIGS API searches for a PROFILE in this order:

1. **gPPROFILE** environmental variable

   The **gPPROFILE** environment variable allows you to use as the External Defaults File (EDF) an alternate PROFILE or a PROFILE not in your current directory.

If the **gPPROFILE** environmental variable is defined as a valid file name, then that file is used as the External Defaults File (EDF). If the **gPPROFILE** environmental variable is defined as a valid directory name, then that directory is searched for a PROFILE. If a PROFILE is found, then it is used as the External Defaults File (EDF).

If the **gPPROFILE** environmental variable is not defined, is defined with an invalid file name or directory name, or there is no PROFILE in the defined valid directory name, the search continues.

For more information on setting environment variables, see the *AIX 5L Version 5.2 Commands Reference*.

2. Current directory

   The current directory is searched for a PROFILE. If there is no PROFILE in the current directory, the search continues.

3. **/usr/lpp/graPHIGS/etc** directory

   The graPHIGS API provides a default PROFILE in the **/usr/lpp/graPHIGS/etc** directory.

For more information on setting User Default Specifications, see *The graPHIGS Programming Interface: Technical Reference*.

Operating System Defaults

The following table shows the graPHIGS API defaults that can be changed within the operating system environment and the format of the user default specifications.

The following default descriptions for the operating system environment are listed in alphabetical order by the user default specification parameter:

AIXTRCE=(aaaaaaaa...,bbbbbbbb,cccccccc)
: A string of up to 50 characters and two strings of up to 8 characters, which are the filepath, filename, and file extension used by the graPHIGS API for trace output.

COMMENT=(cccccccc,cccccccc,.....)
: A comment in the form of a list of strings of 8 or less non-blank characters, which is ignored by the graPHIGS API default processing. The list must not contain more than 8000 such strings.

TRACE={0|n}
: An integer that is the default value of the trace control word at initialization. The value may be specified either as a decimal integer or as an Assembler language hexadecimal constant. The format of the trace control word is described in Trace Control Word.

Meaning of Default	Syntax of the AFMMDFT Options	graPHIGS API Default
Comments for module identification	COMMENT=(cccccccc, cccccccc, ...)	N/A
Trace output (for graPHIGS shell) filepath filename file extension	AIXTRCE=(aaaaaaaa..., bbbbbbbb, cccccccc)	Current working path 'AFMTRACE' N/A
Trace output (for graPHIGS remote nucleus) filepath filename file extension	AIXTRCE=(aaaaaaaa..., bbbbbbbb, cccccccc)	Current working path 'AFMTRACE' 'NUC'
Trace word value	TRACE={0|n}	0

For a complete description of all entries in the External Defaults File (EDF), see *The graPHIGS Programming Interface: Technical Reference*.

Application Default Interface Block (ADIB) Processing

Application Default Specifications (ADS) are accessed through the Application Default Interface Block (ADIB) during Open graPHIGS processing. The ADIB must be specified on the Open graPHIGS (**GPOPPH**) subroutine call as the second parameter. If no ADIB options are specified, this parameter should be *0*.

For a complete description of all entries in the ADIB, see *The graPHIGS Programming Interface: Technical Reference*.

Documenting Your graPHIGS API Problem

Reviewing the following questions before contacting IBM can aid the preparation of your written report.

- Exactly what were you doing when the problem occurred? What sequence of calls to the graPHIGS API led to the problem?
- Does the problem occur repeatedly or does it seem to be a one-time occurrence?
- Does the problem occur predictably or unexpectedly?
- Can you re-create the problem?
- What are the effects of the problem on your overall system operation?

Chapter 7. Reporting graPHIGS API Program Failures

This chapter describes the steps involved in reporting a suspected graPHIGS API failure to IBM.

Central Service is available for the graPHIGS API until discontinued by IBM upon six months notice. However, IBM does *not* guarantee that all errors or program defects will be corrected.

Each licensee's access to program service is determined by the marketing channel through which the product was obtained.

In the United States and Puerto Rico

If obtained through one of the IBM Authorized Personal Computer Dealers or IBM Authorized Industry Remarketers, requests for program service should be made through your dealer/remarketer.

If obtained through the U.S. Marketing and Services (M&S) — Market Operations, requests for program service should be made to IBM Central Service. Your IBM Representative can assist you in completing the request.

Outside the United States and Puerto Rico

Contact your local IBM Representative or your Authorized IBM Supplier for information on program services.

In all cases, the licensee may report problems directly to IBM by sending appropriate information to the address given below.

Obtain an Authorized Problem Analysis Report (APAR) form and complete it.

Once the APAR form is complete, send it and all supporting documentation to:

```
IBM Corporation
11400 Burnet Road
Austin, TX 78758 U.S.A.

Attention:  IBM graPHIGS API Programming Interface APAR Coordinator
            Department 997
```

Filling Out the APAR Form

The APAR form enables you to provide IBM with a description of your problem and your system environment. It must be completed and sent, with accompanying documentation, to the address shown above.

graPHIGS API Product Information

Field "N": *Component or Program in Error/Suspected*, identifies the graPHIGS API product that you are using. The three areas are entered as follows:

- **Component ID Number.** The graPHIGS API Component ID Number is always **5601-257**.
- **Component Level/SU.** This field identifies the level of the graPHIGS API that you are using. The format is **XYZ**, where:

  X is the Version Number (e.g., 2)

  Y is the Release Number (e.g., 2)

  Z is the Modification Level (e.g., 4)

This can be obtained from the BUILDID value found in the second line of a trace file. For example, a BUILDID value of V2R2.4 would be entered in the Component Level/SU field as `224`

- **SCP-CSP Level.** Enter `AIX` into this field.

Problem Symptom Reporting

Use the following information to describe the symptoms of your problem. Use this information when completing fields “R,” “S,” “T,” and “V” of the APAR form. Be sure to include ALL available supporting documentation for the problem you are describing. For fields “R” and “S,” refer to the keyword chart, Symptoms and Failure Keywords.

Record the following information:

R The symptom code that corresponds to the Failure Keyword.

S A Failure Keyword that best describes the failure.

T A problem abstract that summarizes the failure.

V A description of the details and circumstances surrounding the failure.

Symptoms and Failure Keywords

Select the type-of-failure keyword and symptom code for fields “R” and “S” of the APAR form from the following list:

Symptom Code	Failure Keyword	Type of Failure
AB	ABORT	The graPHIGS API detected an unrecoverable error or experienced an abnormal program termination.
DD	DOC	The graPHIGS API documentation is in error.
IN	INCORROUT	The graPHIGS API generated unexpected results.
LP	LOOP	The graPHIGS API entered an infinite loop.
MS	MSGXXX	The problem is identified in the graPHIGS API error message.
PR	PERFM	The problem relates to the graPHIGS API performance.
WS	WAIT	The graPHIGS API has entered a permanent WAIT state.

Descriptions of Type-of-Failure Keywords

The following descriptions will help you narrow down your choice of keywords. Select the word that describes your problem most accurately.

Some keywords also require qualifiers. If a qualifier is required, select it from the list that accompanies the keyword. The qualifier should be entered in “T,” the abstract field, along with the summary of the problem.

ABORT *xxx* Use this keyword if the graPHIGS API terminates abnormally. Replace *xxx* in the keyword with the abort code, if one was issued.

If you select this failure keyword, you must supply the name of the graPHIGS API routine that was in control at the time of the abort. This can be determined from the trace file or the abort dump.

DOC

This keyword applies only to graPHIGS API manuals and refers to wrong, missing, or ambiguous information. The best way to deal with erroneous documentation is to fill out and submit a Reader's Comment Form. All IBM manuals include this form in the back of the book.

If you decide to submit an APAR, use the DOC keyword. In this case, you must also supply an additional failure qualifier keyword. The failure qualifier consists of the order number and level number of the manual and the numbers of any Technical Newsletters (TNLs) that update it. Choose the appropriate failure qualifier keyword from the following list:

- **SC33-8130** - *The graPHIGS Programming Interface: Customization and Problem Diagnosis*
- **SC33-8190** - *Introducing the graPHIGS Programming Interface*
- **SC33-8119** - *The graPHIGS Programming Interface: Tutorial User's Guide*
- **SC33-8198** - *The graPHIGS Programming Interface: Getting Started*
- **SC33-8191** - *The graPHIGS Programming Interface: Understanding Concepts*
- **SC33-8192** - *The graPHIGS Programming Interface: Writing Applications*
- **SC33-8193** - *The graPHIGS Programming Interface: Technical Reference*
- **SC33-8194** - *The graPHIGS Programming Interface: Subroutine Reference*
- **SC33-8195** - *The graPHIGS Programming Interface: Quick Reference*
- **SC33-8196** - *The graPHIGS Programming Interface: Messages and Codes*
- **SC33-8140** - *The graPHIGS Programming Interface: ISO PHIGS Subroutine Reference*
- **SC28-2705** - *The graPHIGS Programming Interface: ISO PHIGS Quick Reference*
- **SC33-8111** - *The Personal graPHIGS Programming Interface: User's Guide for the Graphical Kernel System-Compatibility Option*
- **SC33-8112** - *The Personal graPHIGS Programming Interface: Subroutine Reference for the Graphical Kernel System-Compatibility Option*
- **SC33-8113** - *The Personal graPHIGS Programming Interface: Quick Reference for the Graphical Kernel System-Compatibility Option*

Note: Use MSG *xxxxxx* for missing or incorrect error messages. **DO NOT USE DOC**.

INCORROUT Use this keyword to show that the program produced wrong output, or that the expected output did not appear. In conjunction with this word, use one or more of the following failure qualifier keywords in the failure abstract:

Qualifier
Meaning

NODATA
Blank display.

LOSTDATA
Some data is visible, but some expected data is missing.

MOREDATA
Some application data has been displayed that should not be displayed.

EXTRADATA
Spurious data is displayed (for example, lines or text that is not defined by the application).

DISTORTION
The data displayed is distorted (for example, compressed or expanded along an axis).

NOUPDATE
Display was not updated as specified.

PARMERR
A returned parameter had an incorrect or invalid value (other than an inquiry parameter).

INQUIRY
Incorrect output values from an inquiry routine.

OVERLAY
Returned data has overlaid another data area.

LOOP Standard host system/subsystem techniques or the graPHIGS API trace function can be used to diagnose loop and wait states. Use this keyword if you can determine that the loop was within the graPHIGS API If you are not certain, use the trace facility described in The graPHIGS API Trace Facility.

MSG*xxxxxxx* Use this keyword when a graPHIGS API message:

- Contains incorrect information
- Is not documented or is incorrectly documented in *The graPHIGS Programming Interface: Messages and Codes*
- Is generated by a set of conditions that should not cause it to appear
- Describes an error condition that should be reported (such as certain I/O error messages).

Replace *xxxxxxx* in the keyword with the identifier of the message in error. For example, suppose message AFM0310 is in error; the MSG keyword would be MSGAFM0310.

PERFM Sometimes poor performance can be attributed to the host system/subsystem. Use this keyword only if you can determine that the graPHIGS API is responsible, or if the performance problem occurs as the result of service or a new release.

A failure qualifier keyword is required with PERFM.

Qualifier
Meaning

NOCHANGE
If no recent changes have been made, or the problem occurs in a version of the program which previously ran acceptably.

CHANGE
If the problem occurs only after service or a new release was applied.

WAIT The graPHIGS API contains a small number of calls to system or subsystem wait services. It also issues requests for system services. These tasks may produce wait states. If you don't know why you are in the wait state, verify the cause by using the trace facility described in The graPHIGS API Trace Facility.

Problem Abstract

Use field "T" of the APAR form to provide a summary of the failure. Where failures involve messages or ABORTs, use the format shown for field "S," Failure Keyword (e.g. ABORT806). Where a failure keyword has a qualifier list, include the qualifier in the abstract.

Problem Description

Use field "V" of the APAR form to provide the details of the failure. Include the following items:

- Sequence of actions and observations at the device, including expected results
- The graPHIGS API subroutine calls made prior to the failure
- Description of the processing of the application.

Supporting Information

Some of the information needed to describe a problem is generic, while some information depends on the problem you are experiencing.

General information required

- Application program - a copy of the application program so that the operations of the application can be reviewed.
- Maintenance list - a list of all service that has been applied.

Specific problems

In general, the following items might be required to solve a problem:

- A copy of the application program in which the failure occurs. This may be in the form of an executable module or the source code to an application.
- A core file. If one is sent in as supporting documentation, then the failing executable module must also be sent in.
- A synchronized graPHIGS API component trace.
- For incorrect output, a description of the calls, the data values used, and the expected output must be given.
- For a graPHIGS API message that does not result in termination, a full description of the calls used and the expected output must be given.

Appendix A. ABORT Codes for the graPHIGS API

Message 1204 of the graPHIGS API displays the following ABORT codes on **stderr**:

Hex.	Dec.	Explanation
1	1	Description table overflow
41A	1050	Invalid request code
426	1062	Local EUDS area too small
427	1063	AFME0DF entry wrong
516	1302	Invalid RCP function code
522	1314	Request exceeds maximum storage
532	1330	Invalid SSM request
533	1331	Overlap found of FAQE
6E0	1760	Invalid RCP subcomponent code
6E2	1762	Invalid QUICKCELL request
6E3	1763	Zero cellsize
6E5	1765	Structure reference pointer found where none expected

Appendix B. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept. LRAS/Bldg. 003
11400 Burnet Road
Austin, TX 78758-3498
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
(c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
AIXwindows
GDDM
IBM
RS/6000

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be the trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

The graPHIGS Programming Interface: Customization and Problem Diagnosis

Publication No. SC33-8130-05

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC33-8130-05

IBM

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department H6DS-905-6C006
11501 Burnet Road
Austin, TX 78758-3493

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in U. S. A.

SC33-8130-05