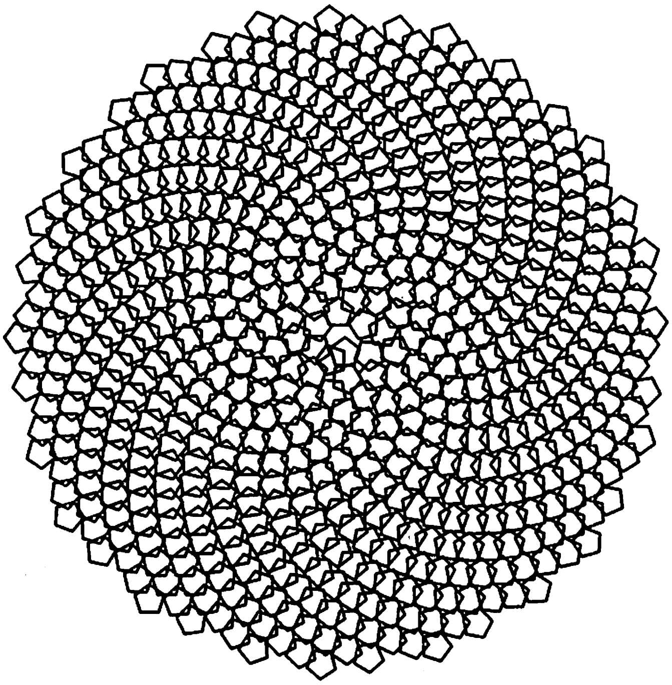
GDDM

Diagnosis and Problem Determination Guide

SC33-0326-3

IBM



Front Cover Pattern: Electronic Sunflower

The pattern on the front cover was produced by a GDDM program. The program to produce this pattern, and many variations of the pattern, is published in:

- *GDDM Application Programming Guide*
- *GDDM Base Programming Reference*

SC33-0326-3
File No. S370/4300-37

GDDM

Diagnosis and Problem Determination Guide

Program Numbers

GDDM/VMXA, 5684-007	Version 2 Release 3
GDDM/MVS, 5665-356	Version 2 Release 3
GDDM/VSE, 5666-328	Version 2 Release 3
GDDM-PGF, 5668-812	Version 2 Release 1 Modification 1
GDDM Interactive Map Definition, 5668-801	Version 2 Release 1 Modification 1
GDDM-IVU, 5668-723	Version 1 Release 1 Modification 1
GDDM-GKS, 5668-802	Version 1 Release 1 Modification 1
GDDM-CSPF, 5688-013	Version 1 Release 1
GDDM-REXX, 5664-336	Version 1 Release 1
GDDM-PCLK, 6242913 (U.S.A. part number)	Version 1.1
GDDM-OS/2 Link, 5688-113	Version 1.0

Licensed Programs

IBM

Fourth Edition (June 1990)

The following paragraph does not apply to the United Kingdom or any 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 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 edition applies to the following series of IBM GDDM (Graphical Data Display Manager) licensed programs:

Program number	Program name	Version	Release	Modification
5665-356	GDDM/MVS	2	3	0
5666-328	GDDM/VSE	2	3	0
5684-007	GDDM/VMXA	2	3	0
5664-336	GDDM-REXX	1	1	0
5668-723	GDDM-IVU	1	1	1
5668-801	GDDM Interactive Map Definition	2	1	1
5668-802	GDDM-GKS	1	1	1
5668-812	GDDM-PGF	2	1	1
5688-013	GDDM-CSPF	1	1	0
5688-113	GDDM-OS/2 Link	1.0		
6242913 (part number)	GDDM-PCLK	1.1		

and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

Consult the latest edition of the applicable IBM system bibliography for current information on this product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this book. If the form has been removed, send your comments either to:
International Business Machines Corporation, Department 6R1H, 180 Kost Road, Mechanicsburg, Pennsylvania 17055-0786, U.S.A.
or to:
IBM United Kingdom Laboratories Limited, Information Development,
Mail Point 95, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

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

Special notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The following terms, denoted by an asterisk (*), used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

GDDM
graPHIGS
IBM
InfoWindow
Operating System/2
OS/2
Personal System/2
PS/2

The following terms, denoted by a double asterisk (**), used in this publication, are trademarks of other companies:

DEC	Digital Equipment Corporation
Tektronix	Tektronix Inc.

Preface

What this book is about

This book is intended to help the customer diagnose problems that may occur when using IBM* Graphical Data Display Manager (GDDM*). It primarily contains problem determination procedures for GDDM running on VM/CMS, MVS/TSO, CICS, IMS/VS, DOS/VSE, and VMXA systems.

Unless specifically stated otherwise, the information in this book must not be used for programming purposes.

However, this book also provides the following types of information, which are explicitly identified where they occur:

Product-sensitive programming interface

Installation exits and other product-sensitive interfaces are provided to allow the customer installation to perform tasks such as product tailoring, monitoring, modification, or diagnosis. They are dependent on the detailed design or implementation of the product. Such interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of maintenance.

End of Product-sensitive programming interface

General-use programming interface

General-use programming interfaces are provided to allow the customer to write programs that use the services of GDDM. Unlike product-sensitive programming interfaces, they do not have significant dependencies on detailed product design and implementation.

End of General-use programming interface

If you need to know more about where programming interface information is described, or about the definitions of the different types of information in the GDDM library, you should read the *GDDM Library Guide and Master Index*.

* IBM trademark. For a complete list of trademarks, see page iii.

Who this book is for

This book is for application and systems programmers who use GDDM, and for IBM service personnel.

What you need to know to understand this book

This book assumes that you are familiar with GDDM, that you have some knowledge of debugging application and system problems, and that you understand the task that your application program is intended to perform.

If you are not familiar with GDDM, you should read some of the books in the GDDM library, starting with the *GDDM General Information* manual. There is a list of the books in the GDDM library on page ix. Other books you may need are listed on page x.

How to use this book

If you meet a problem while using GDDM, start by reading Chapter 1. Then read other chapters and appendixes, as appropriate.

Conventions

Throughout this book, the first (leftmost) byte or bit of a sequence is designated as byte or bit 0.

In programming syntax:

[ ] indicates optional items.
{ } indicates selections or choices.

Book structure

- Chapter 1, Diagnostic procedures — helps you decide whether a problem is within GDDM.
- Chapter 2, Debugging aids — describes the help that GDDM makes available to you.
- Chapter 3, GDDM tracing — explains how GDDM tracing can be started, and gives examples of GDDM traces.
- Chapter 4, GDDM Interactive Map Definition diagnosis — describes an additional diagnosis aid that is available with GDDM Interactive Map Definition (GDDM-IMD).
- Chapter 5, GDDM-REXX diagnosis — describes an additional diagnosis aid that is available with GDDM-REXX.
- Chapter 6, GDDM-PCLK diagnosis — describes an additional diagnosis aid that is available with GDDM-PCLK.
- Chapter 7, GDDM-OS/2 Link diagnosis — describes an additional diagnosis aid that is available with GDDM-OS/2 Link.

- Chapter 8, Reporting GDDM problems to IBM — lists the information that you should give when you report a suspected GDDM defect to IBM, and explains how to report the problem. A problem report form is included in the chapter.
- Appendix A, Data areas — contains information about some GDDM data areas (control blocks and tables) and is intended to help you diagnose program failures.
- Appendix B, Abend codes — contains a list of the abend codes issued by GDDM.
- Appendix C, Message-to-module cross-reference — relates GDDM error messages to the modules that issue them.
- Appendix D, Trace-string grammar — gives the formal definition of the grammar and syntax for sets of TRCESTR statements.
- The Glossary defines terms used in the book.
- Index.

Changes from the third edition

This book includes the functions introduced with GDDM Version 2 Release 3.

All references to GDDM/VM (program number 5664-200) have been removed because the product is functionally stabilized at Version 2 Release 2. GDDM/VM is not included with GDDM Version 2 Release 3, but continues to be available for customers with VM/SP Releases 4 and 5. If you use VM/SP Release 6, or you need any of the functional enhancements in GDDM Version 2 Release 3, you should change to GDDM/VMXA.

GDDM-OS/2 Link diagnosis is now in a separate chapter.

The information from Technical Newsletter (TNL) SN33-6352 has been included in this book.

Notes:

1. Technical changes to this book are indicated by a vertical line to the left of the change, as on this page for example.
2. Some of the information in this book was previously published in other GDDM books.

Changes from the second edition

This book includes the functions introduced with GDDM Version 2 Release 2; for example:

- High performance alphanumerics (HPA)
- GDDM-CSPF
- GDDM-PCLK
- Tracing in the VMXA environment.

More information on GDDM-IVU, GDDM-REXX, and GDDM-GKS has been added.

Changes from the first edition

This book now includes the functions introduced with GDDM Version 2 Release 1 Modification 1; for example:

- You can now specify the high-resolution input generator (HRIG) and composite document presentation data stream (CDPDS) action functions in TRCESTR statements.
- The components CDU, GKS, and IVU have been added to those you can specify in the component event function in TRCESTR statements.
- You can now use tracing in the VSE Batch environment.

This book also covers diagnosing problems in GDDM-IVU, GDDM-REXX, and GDDM-GKS.

Bibliography

GDDM library

Introduction

GDDM General Information and brochures	GBOF-0058	
GDDM General Information	GC33-0319	
GDDM If you make business presentations... (brochure)	GC33-0455	
GDDM If you're an engineer... (brochure)	GC33-0456	
Introducing GDDM-OS/2 Link (flyer)	GC33-0696	
GDDM Release Guide	GC33-0320	
GDDM Library Guide and Master Index	GC33-0595	
GDDM-PGF Better Charts (poster)	GC33-0529	

General

GDDM Interactive Map Definition	SC33-0338	
GDDM-REXX Guide	SC33-0478	
GDDM Image View Utility	SC33-0479	
GDDM-PCLK Guide	—	

User's guides

GDDM Guide for Users	SC33-0327	
GDDM-PGF Interactive Chart Utility	SC33-0328	
GDDM Image Symbol Editor	SC33-0329	
GDDM-PGF Vector Symbol Editor	SC33-0330	
GDDM-PCLK Reference Summary (booklet)	SX33-6067	
GDDM-CSPF User's Guide	SC33-0552	
GDDM Typefaces and Shading Patterns	SC33-0554	

Programming

GDDM Application Programming Guide (two volumes)	SC33-0337	
GDDM Base Programming Reference (two volumes)	SC33-0332	
GDDM Base Programming Reference Summary (booklet)	SX33-6053	
GDDM-PGF Programming Reference	SC33-0333	
GDDM-PGF Programming Reference Summary (booklet)	SX33-6054	
GDDM-GKS Programming Guide and Reference	SC33-0334	

Systems

GDDM Installation and System Management for MVS	GC33-0321	
GDDM Installation and System Management for VM	GC33-0323	
GDDM Installation and System Management for VSE	GC33-0322	
GDDM Performance Guide	SC33-0324	

Diagnosis

GDDM Messages	SC33-0325	
GDDM Diagnosis and Problem Determination Guide	SC33-0326	◄ **this book**

books

Related publications

In addition to the GDDM library, you may need to refer to one or more of these manuals:

VM

VM/System Product Interpreter Reference, SC24-5239

VM/SP System Programmer's Guide, SC19-6204

VM/XA System Product Interpreter Reference, SC23-0197

CICS

CICS Library Guide, GC33-0356 (for complete information about the books that are available for your release of CICS.)

3179-G

3179-G Color Graphics Display Station Description, GA18-2177

3192-G

3192 Display Station Description, GU21-6249

3270-family devices

3270 Information Display System Configurator, GA27-2849
3270 Information Display System Data Stream Programmer's Reference, GA23-0059
8775 Display Terminal Component Description, GA33-3044

3270-PC/G and 3270-PC/GX workstations

Introducing the IBM 3270 Personal Computer/G and /GX Ranges of Work Stations, GA33-3157
3270-PC/G Personal Computer/G and /GX Ranges of Work Stations; Planning Guide, GA33-3158
3270-PC/G Guide to Operations, GA33-3140
3270-PC/GX Guide to Operations, GA33-3139
Graphics Control Program User's Guide and Reference, SC33-0207 (for IBM 3270-PC/G and /GX)
Graphics Control Program Version 3.2 User's Guide, SC33-0368 (for IBM 3270-PC AT/G and /GX)
Graphics Control Program Version 3.2 User's Reference, SC33-0372 (for IBM 3270-PC AT/G and /GX)

3472-G

IBM InfoWindow 3472: User's Guide*, GA18-2917

PS/2*

IBM Personal System/2 Display Adapter 8514/A: Technical Reference Manual*, S68X-2248

* IBM trademark. For a complete list of trademarks, see page iii.

3274

3274 Control Unit Description and Programmer's Guide, GA23-0061
3274 Control Unit Planning, Setup and Customization Guide, GA23-2827

3174

3174 Subsystem Control Unit Customization Guide, GA23-0214

3174 ASCII device support

3174 Asynchronous Emulation Adapter: Description and Reference, GA27-3872

3174 Establishment Controller: Customizing Guide, GA23-0214

3174 Establishment Controller: AEA Description and Reference, GA27-3872

3174 Establishment Controller: Terminal User's Reference for Expanded Functions, GA23-0332

Image devices

3193 Display station

Description, GA18-2364
Setup Instruction, GA18-2366
Operator's Guide, GA18-2365
Problem Solving Quick Check Guide, GA18-2443
Problem Solving Guide, GA18-2444

3117 Scanner

IBM 3117 Scanner and IBM 3117 PC Adapter Guide to Operations, GA18-2477
IBM 3117 Scanner and Extension Unit Guide to Operations, GA18-2478
IBM 3117 Scanner Hardware Maintenance and Service, SY18-2159
IBM 3117 Scanner Technical Reference, SC18-2105

3118 Scanner

Scanner Guide to Operations, GA18-2475
High Speed Adapter Guide to Operations, GA18-2476
IBM 3118 Scanner Hardware Maintenance and Service, SY18-2158
High Speed Adapter Hardware Maintenance and Service, SY18-2167
Scanner Technical Reference, SC18-2104
High Speed Adapter Technical Reference, SC18-2117

Other printers

3812 and 3816 Printers

Intelligent Printer Data Stream Reference, S544-3417
IBM 3812 and 3816 Pageprinters: IPDS Handbook, GA34-2082
IBM 3812 and 3816 Pageprinters: Programming Reference for 3270 Information Display System Attachment, GA34-2081

4224 Printer

Printer Product and Programming Description Manual, GC31-2551
Operating Instructions, GC31-2546
Guide to Operations, GC31-3621

4234 Printer

Model 11: Operation Instructions, GC31-3736

4250 Printer

Operator's Guide, GA33-1551

5550 Multistation

(Available in Japanese only)

5550 Japanese 3270-PC User's Guide, N:SC18-2059
How To Use 5550 Japanese 3270-PC, N:SC18-2060
5550 Japanese 3270-PC/G User's Guide, N:SC18-2071
How To Use 5550 Japanese 3270-PC/G, N:SC18-2072
5550 Small Cluster User's Guide, N:SC18-2092
How To Use 5550 Small Cluster, N:SC18-2091
5550 Small Cluster/Graphics User's Guide, N:SC18-2107
How To Use 5550 Small Cluster/Graphics, N:SC18-2108
5550 3270 Kanji Emulation Description, N:SC18-2020
5550 3270 Kanji Emulation Operator's Guide, N:SC18-2021

GDDM/graPHIGS

*Messages and Codes for graPHIGS**, SC33-8105.

Non-IBM devices

Ask your device supplier for information about applicable publications.

* IBM trademark. For a complete list of trademarks, see page iii.

Contents

Chapter 1. Diagnostic procedures

This chapter shows the structure of GDDM* Version 2 Release 3 and helps you determine whether a suspected problem is in GDDM.

Structure of GDDM Version 2 Release 3

GDDM is an IBM* licensed program that enables you to display text, images, graphics, and alphanumerics on several types of IBM displays, workstations, printers, and plotters. Version 2 Release 3 of GDDM enables you to use also these non-IBM graphics devices:

DEC** VT240, VT241, VT330, VT340

Tektronix** 4105, 4205, 4207, 4208, 4209.

GDDM consists of three base programs (one for each system supported) and a series of associated programs.

The base programs are:

- GDDM/MVS, licensed program number 5665-356
- GDDM/VSE, licensed program number 5666-328
- GDDM/VMXA, licensed program number 5684-007

each of which includes the Image Symbol Editor.

The associated programs are:

- GDDM-PGF, licensed program number 5668-812, which includes:
  - Presentation Graphics routines
  - Interactive Chart Utility
  - Vector Symbol Editor.
- GDDM Interactive Map Definition (GDDM-IMD), licensed program number 5668-801.
- GDDM Image View Utility (GDDM-IVU), licensed program number 5668-723.
- GDDM-REXX, licensed program number 5664-336.
- GDDM-GKS, licensed program number 5668-802.
- GDDM-CSPF, licensed program number 5688-013.
- GDDM-PCLK, Version 1.1 6242913 (part number).
- GDDM-OS/2 Link Version 1.0, licensed program number 5688-113.

The GDDM base programs and GDDM-PGF, GDDM-GKS, and GDDM-IVU each have a National Language feature:

The release levels for all the programs listed here are given on page ii.

* IBM trademark. For a complete list of trademarks, see page iii.

** Trademark. For a complete list of trademarks, see page iii.

You can find more information about GDDM, its associated licensed programs, and its utility programs, in the other publications listed in "Bibliography" on page ix.

Determining whether the problem is in GDDM

The remainder of this chapter shows you how to determine whether your problem is in GDDM. If the problem is in GDDM, go to Chapter 8, "Reporting GDDM problems to IBM" on page 117, which tells you how to report the problem to your IBM Support Center representative.

For some problems, you may find that you do not have enough information to enable the IBM Support Center Representative to help you. You may be asked to investigate the problem further by running a GDDM trace or by using the GDDM-IMD, GDDM-REXX, GDDM-PCLK, or GDDM-OS/2 Link diagnostics facility. Each of these aids is described in this book.

Initial diagnosis

If you get an unexpected result while using GDDM, the cause of the problem could be:

- Device definitions
- The application program
- A GDDM utility program (for example, ICU, Vector Symbol Editor, Image Symbol Editor)
- The GDDM print utilities
- The output device or control unit
- The workstation customization
- The operating system, subsystem, or telecommunications access method
- The way GDDM has been installed
- GDDM Base, GDDM-PGF, GDDM-IVU, GDDM-REXX, GDDM-GKS, GDDM-PCLK, GDDM-OS/2 Link, or GDDM-CSPF.

Outlined below are the symptoms you are most likely to meet, together with some checks you can make that may find the cause.

Unexpected output

Incorrect output

If your output is not correct:

- The device token, if you have used one, may be wrongly defined or you may have used the wrong device token for the device.
- There may be a problem with your nickname table; for example, the nickname may be directing the output to the wrong device.

- Check the application program you are using. Is it device-independent? If not, you may be running it on the wrong device.
- Check that the devices you are using are correctly defined to the system.
- Check that your DSOPEN statement has been correctly coded. If you are using GDDM-GKS, the equivalent is the open workstation (GOPWK) function call.
- Check that all the symbol sets required by the job are available.
- In GDDM-IVU, a user-defined map for a menu or help panel might contain an error. If your installation does not use standard maps, run a test using the standard maps. If the error is not reproduced, check the user-defined maps.
- For GDDM-GKS, check that your ADMMDFT GKSWS macro or statement gives the correct device token for the physical device.

No output

If you do not get any output, check that:

- Your output device (terminal, plotter, printer) is suitable for graphics or image output
- Your device and control unit are at the correct level of microcode
- Your device and control unit are correctly configured
- Your devices are correctly connected, and the connections are not loose
- Your devices are switched on and "online"
- The devices are correctly defined to the system
- Your graphics are in segments, if your output device is a printer or plotter.

Device checks/sense codes

If you receive these, the most likely causes are:

- The device is not defined, or is incorrectly defined, to the subsystem or telecommunications access method.
- The device token or a nickname is causing the wrong data stream to be built.
- The device, the control unit, or both may be at an incorrect level of microcode.
- The device, the control unit, or both may be incorrectly configured.

Note: For TSO, the device checks and sense codes may be normal, resulting from line-by-line reshow causing an incomplete data stream transmission before a complete reshow.

GDDM error messages

Most GDDM error messages begin with the prefix ADM. Other prefixes are:

AEM	GDDM-IMD
EAK	GDDM-CSPF
ERX	GDDM-REXX
GQD	GDDM-PCLK
GQF	GDDM-OS/2 Link.

There are many causes for these messages. Check the message in the *GDDM Messages* manual, and take any action recommended there.

Messages that start AFM are from GDDM-graPHIGS*. Check them in the *Messages and Codes for graPHIGS* manual.

Abends

If you get an abend, and diagnosis shows that the error is in GDDM, you might be asked to submit an APAR. See Chapter 8, "Reporting GDDM problems to IBM" on page 117 for information on this.

For a list of abend codes, see Appendix B, "Abend codes" on page 155.

Where to look for further advice

If you think you have isolated the problem, but have not fixed it, this section suggests possible sources of advice.

Problems with application programs

If you think that your problem may be caused by an error in the application program, but are not sure, the following may help you:

- Documentation supplied with the application program.
- Chapter 2, "Debugging aids" on page 27 describes the structure of GDDM error messages and the GDDM call statements FSEXIT and FSQERR, which can help you diagnose errors in application programs.
- Chapter 3, "GDDM tracing" on page 37 describes the TRCESTR default keyword and FSTRCE call statements, which you can use to record a trace. You may do this if you suspect an internal GDDM error.
- The *GDDM Base Programming Reference* manual, the *GDDM-PGF Programming Reference* manual, and the *GDDM Release Guide*.

  These tell you the correct parameters and values to use with GDDM and GDDM-PGF call statements.
- The *GDDM Messages* manual.

  This gives information about the messages issued by GDDM.

* IBM trademark. For a complete list of trademarks, see page iii.

Problems with GDDM utility programs

The GDDM utility programs are:

- Image Symbol Editor (part of GDDM Base)
- Interactive Chart Utility (part of GDDM-PGF)
- Vector Symbol Editor (part of GDDM-PGF)
- Image View Utility (GDDM-IVU)
- Interactive Map Definition (GDDM-IMD)
- Central Slide and Plot Facility (GDDM-CSPF)
- GDDM-PCLK
- GDDM-OS/2 Link
- GDDM-GKS
- GDDM-REXX.

If there is a problem in one of these programs, use the online help information or tutorial, or look at the appropriate book to check that the utility is being used correctly.

If any error messages appear, check them in the appropriate book or the *GDDM Messages* manual.

Problems with GDDM print utilities

If you suspect a problem with a GDDM print utility, inspect the system console log. Your system operator should have access to this, and may well be able to tell from it the cause of the problem.

If you try to use a symbol set that you have not made available to your GDDM print utilities, you will get errors. This is a common cause of problems, so check that the symbol sets you have used are available to the print utility.

Problems with the hardware

Hardware problems are most likely to occur during installation testing. The most common indication that the problem is hardware-related is that no graphics are displayed and the message `ADM0275 GRAPHICS CANNOT BE SHOWN` appears. The problem is usually caused by an incorrect device definition or an inappropriate controller configuration.

If you think you have a hardware problem, refer to the *GDDM Installation and System Management* manual for your system. Ensure that the terminals you are using can show graphics and check whether they require any additional features to do it. More information on hardware-related problems is given in "What to do if things go wrong" on page 7.

If you are using GDDM-IVU, check that the device being used supports GDDM image functions. This information is listed for all devices in the *GDDM Installation and System Management* manual for your system.

You may also need to refer to the component description or customization manual for the devices involved.

Problems with customizing workstations

Failures that occur in a 3270-PC/G or 3270-PC/GX workstation may be caused by a problem with the customization of the workstation. If you suspect that the problem is of this type, check the *GDDM Installation and System Management* manual, in the Appendix "Preparing your workstations".

You should also check the customization procedures in the *GCP User's Guide and Reference* manual.

Problems with the operating system

If you think you have an operating system problem, refer to the procedures given in the installation manual for the operating system or subsystem. Check the minimum software levels required by GDDM; these are in the *GDDM Installation and System Management* manual. You should also check any error messages in the appropriate manuals.

Problems with the installation of GDDM

The most likely symptoms of a problem with the installation of GDDM are:

- A program does not run.
- A program terminates abnormally.
- Several different programs fail.

Incorrect installation of GDDM may also mean that graphics are not shown on some, or all, of your devices that are capable of showing them.

If you suspect a problem with the installation of GDDM, look at "What to do if things go wrong" on page 7. Also check in the *Program Directory*, issued with the distribution tape for GDDM and any features you may be installing, for late changes or extra information.

Problems within GDDM

The problem might be in GDDM, its associated programs, or an application program. If you believe the problem to be in one of these areas, report the error to IBM giving specific information. Chapter 8, "Reporting GDDM problems to IBM" on page 117 deals with this.

What to do if things go wrong

If you find that you cannot get GDDM to work satisfactorily when you use it, there are several things you should check before calling for assistance.

- Check in the section "Common installation errors and pitfalls" on page 15 for the symptoms of your problem. If you cannot find your problem described in that section, continue with the following checks.
- Check that you have the correct levels of:
  - System
  - Subsystem
  - Access method
  - Controller microcode
  - Control program (where appropriate)
  - Hardware features (specifically for graphics).

  Check any relevant notes, restrictions, or prerequisites that are mentioned in the *GDDM Installation and System Management* manual for your system.
- If an error message appears, either on a display screen or on your console log, you should look it up in the *GDDM Messages* manual.
- If an abnormal termination (abend) occurs, you should check Appendix B, "Abend codes" on page 155, which contains details of all GDDM abends.
- If you are installing GDDM, check through the steps you have completed so far, and examine the console log for any unusual messages. Also review the entire installation process from the start, and check for errors. In particular check:
  - Your preinstallation planning instructions
  - The VTAM bind parameters
  - Your CICS table entries
  - Your IMS table entries
  - Any defaults you have changed
  - The *Program Directory* and the PSP (preventive service planning) "bucket" for late information.
- If you have some terminals working successfully on GDDM, compare the definitions for the successful terminals against the one you are now diagnosing.

If these checks show up no unusual circumstances, check your hardware; it could be that they are not set up correctly to show GDDM graphics. The section "Checking hardware characteristics" on page 8 gives you advice on doing this.

If your problem remains, check with your IBM Support Center. Chapter 8, "Reporting GDDM problems to IBM" on page 117 contains information about this.

Checking hardware characteristics

To determine your problem, you may need to check your hardware characteristics. The following sections give you advice on doing this, and may enable you to isolate the problem to a specific terminal or controller. Checking a screen will require about one minute, a controller another ten to fifteen minutes.

3179-G and 3192-G graphics diagnosis

First check that you have a 3179 Model G. No 3179 terminals, other than the 3179-G, show graphics. Next refer to the documentation provided with the terminal, *IBM 3179 Color Display Station Operator Reference and Problem Solving Guide*, GA18-2180.

Similar considerations apply to the 3192-G color display station.

3472-G graphics diagnosis

First check that you have a 3472 Model G. No 3472 terminals, other than the 3472-G, show graphics. Next refer to the documentation provided with the terminal, *IBM InfoWindow* 3472: User's Guide*, GA18-2917.

3270 graphics diagnosis

This is applicable only to 3278 and 3279 terminals.

Check that the terminal is built to display graphics.

To test for installed graphics features (program symbol sets 2 and 4), hold the Alternate key down and press the TEST key on the lower left-hand row of the keyboard. Type "/8" and press the ENTER key.

If graphics are available, you will see the PS fields A through F with a symbol. Absence of a PS (program symbol) set is signified by a period. Triple-plane PS stores are marked with a red and white triangle.

To leave Test mode, press PF3.

Absence of the PS feature means that you need to have graphics capabilities installed on your terminal. Ask your IBM representative to verify the terminal configuration. If you receive the stick man message "operation not permitted" in the Operator Information Area, check the controller for graphics (see the next section), and then review the screen functions again.

3270 EDCB verification

This section is not applicable to 3179-G color display stations or 3270-PC workstations.

3270-devices that have been configured for extended functions (more correctly known as SFAP — structured field and attribute processing) should normally

* IBM trademark. For a complete list of trademarks, see page iii.

operate with an extended DCB (device control block) allocated by the controller and created during customization.

As part of problem determination, perform the following test to confirm that the device is operating with an extended DCB. This test is more fully documented in the Error Codes Appendix of the *3274 Control Unit Description and Programmer's Guide*.

On any 3278 or 3279 display attached to the same controller as the display or printer being diagnosed:

1. Enter TEST Mode by holding down the ALT key and pressing the TEST key.
2. Type AA/6 to display the DCB for the device in question; AA is the coax port number in question (00 through 31). (If the device being used for the test is the port in question, /6 will suffice.)
3. Press the ENTER key. The first X'40' (64 decimal) bytes of the DCB in question will be displayed on lines 3 through 6. The first two bytes of the displacement from the start of the control block of the data being displayed will appear on line 2 (initially 00).
4. Continue to press the ENTER key. Line 2 should change to 04, 08, 0C, 10, 14, 18, 1C for each pressing. Lines 3 through 6 will change to display further bytes of the DCB.

If the test display stops at 0C (with the keyboard inhibited with the minus function indicator on the fifth pressing of the ENTER key), the device is not operating with an extended DCB.

The reason for this lies with the controller, or the customization, or the features installed on the device. TEST /8 (displays), described above, or TEST 5 (printers), described below, can be used to verify the features installed on the device.

If the test display stops at 1C, the device is operating with an extended DCB.

General printer diagnosis

If you have a problem when using a printer, check that the printer has been set up with the correct page size. The printer page size is checked and changed using the printer operator panel; for more information, refer to the *Operating Instructions* for the printer. It is not important what lines per inch (lpi) or characters per inch (cpi) settings are used, if the resulting page size is correct.

Page depth in inches = maximum page length (MPL) / lpi
Page width in inches = maximum print position (MPP) / cpi

The other printer settings can be set as you like.

If device tokens are being used, the page size defined to the printer must be at least as large as that defined to GDDM by the device token. (It does not matter if the printer has a larger page size defined to it than that defined by the device token.)

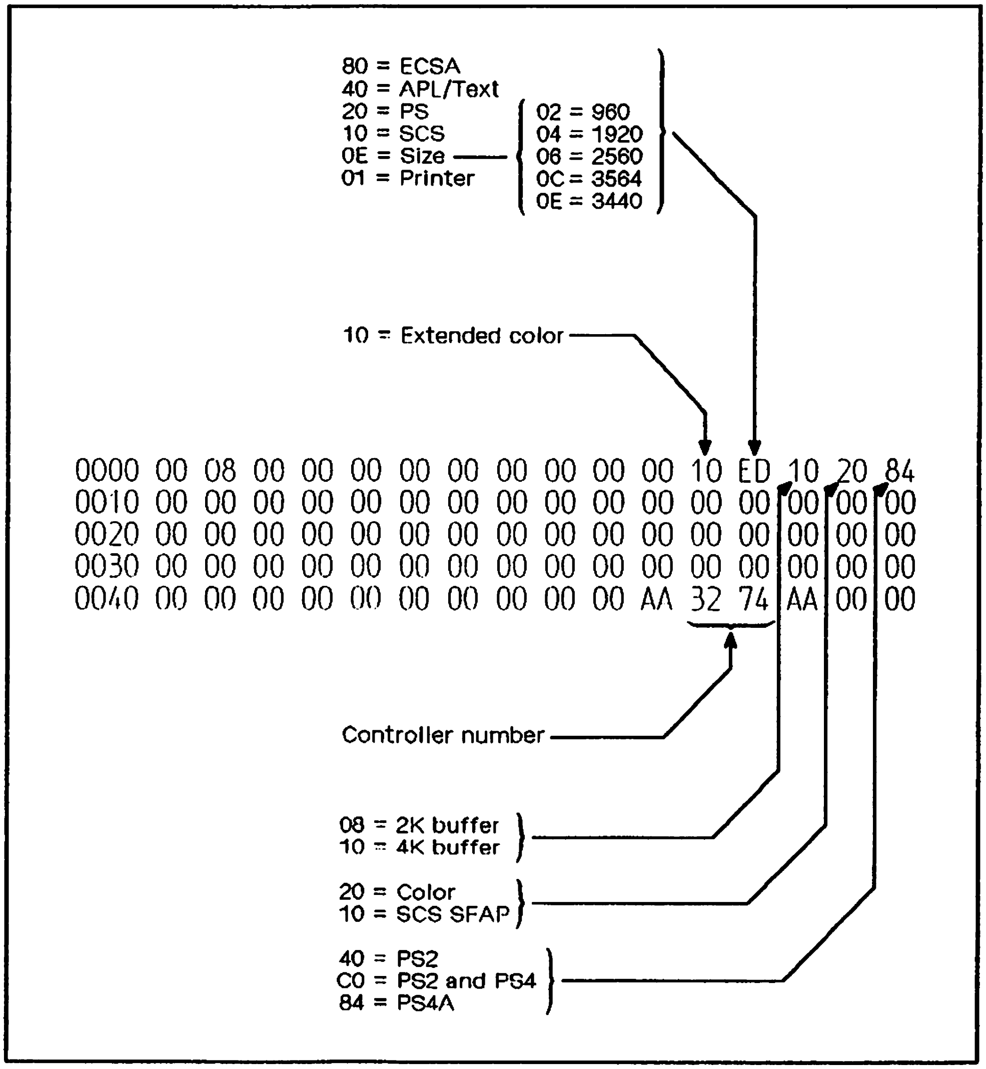
If A4 device tokens are being used on the 4224 printer, the printer must be in full-page mode when the 4224 ASF (auto sheet feed) attachment is used. To check the mode, run test 307 and verify that 001 is displayed when the ALT key and the 9 key are pressed at the same time.

3287 printer diagnosis

To test for installed graphics features on a 3287 printer, you must use the test procedure to print the print control information area (PCIA). To produce the PCIA:

1. Press and hold down the TEST button
2. Press and release the 5 button
3. Release the TEST button.

An example of the PCIA is shown below. For full details, see the maintenance information manual that is provided with the printer.



```
0000 00 08 00 00 00 00 00 00 00 00 00 10 ED 10 20 84
0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 AA 32 74 AA 00 00
```

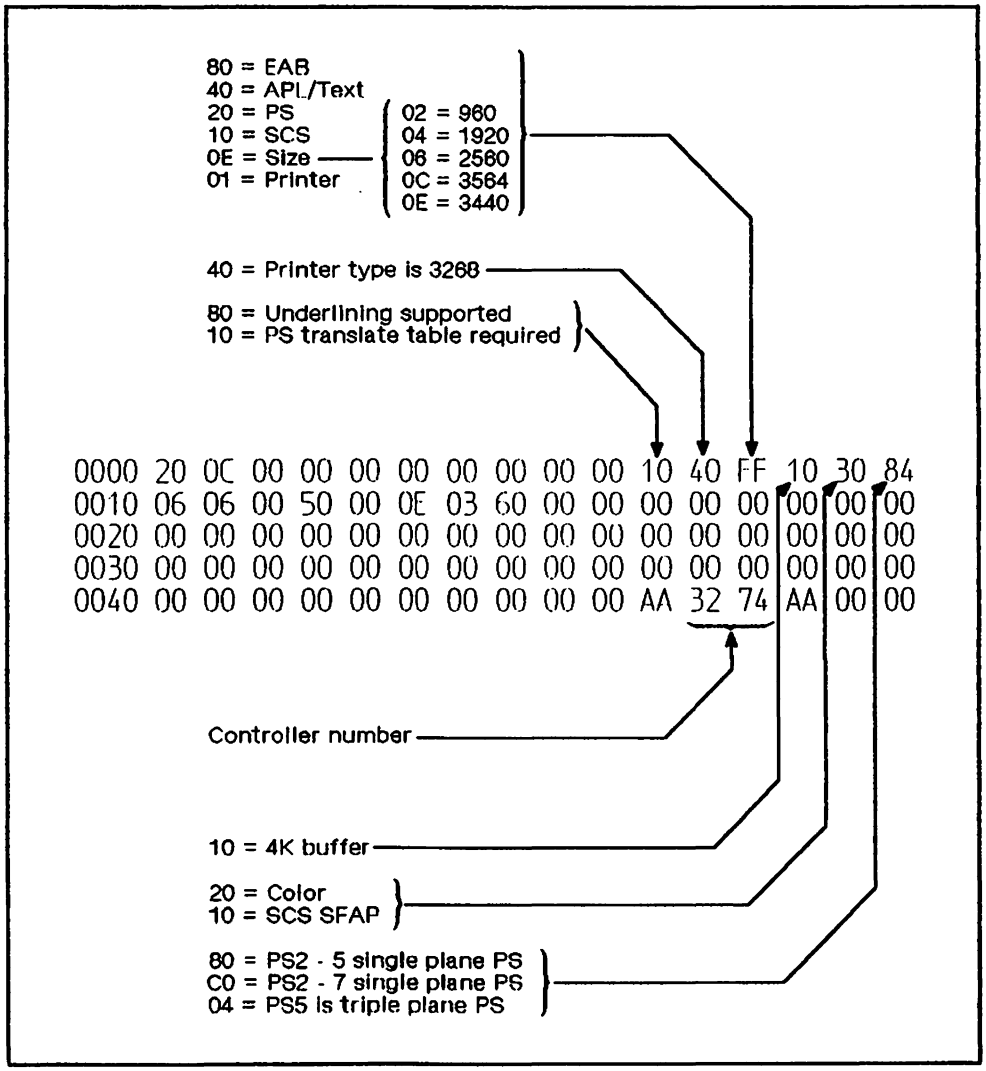
3268 printer diagnosis

To test for installed graphics features on a 3268 printer, you must use the test procedure to print the print control information area (PCIA). To produce this PCIA:

1. Press and hold down the TEST button
2. Press and release the 4 button
3. Release the TEST button.

An example of the PCIA is shown below. For full details, see the maintenance information manual that is provided with the printer.

Note: To enable use of the triple plane PS, language switch 1 must be in the On position.



```
0000 20 0C 00 00 00 00 00 00 00 00 10 40 FF 10 30 84
0010 06 06 00 50 00 0E 03 60 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 AA 32 74 AA 00 00
```

80 = EAB
40 = APL/Text
20 = PS
10 = SCS
0E = Size
01 = Printer
02 = 960
04 = 1920
06 = 2560
0C = 3564
0E = 3440
40 = Printer type is 3268
80 = Underlining supported
10 = PS translate table required
Controller number
10 = 4K buffer
20 = Color
10 = SCS SFAP
80 = PS2 - 5 single plane PS
C0 = PS2 - 7 single plane PS
04 = PS5 is triple plane PS

3193 display station diagnosis

Check the stand-alone TEST procedure and the device set-up procedure. The set-up procedure enables you to change many attributes of the display; these may affect how the application runs. The setup procedures include:

- Whether extended attributes are supported, such as blink or reverse video.
- How many hardware partitions are available in each of the two logical terminals.
- The size of each logical terminal's viewport.
- The volume of the alarm.

Another potential problem is that Image is only supported on Logical Terminal 1 (LT-1).

3274 controller diagnosis

To permit graphics, a 3274 must be configured for graphics. Models eligible for graphics are:

- Any 3274 Model 31
- 3274-1A
- 3274-1C
- 3274-1D
- 3274-51C with enough storage.

Models that are not eligible for graphics but that can be upgraded to be graphics-compatible are:

- Any 3274 Model 21
- 3274 Model 1B.

Ensure that you have the correct model and enough storage (96K bytes are required for 3278, 3279, 3287, but later devices may need more storage; 3274 Model 31s all have sufficient storage). Then check that the diskettes that have been customized are Configuration C or D for 3278, 3279, and 3287, or D or T for 3270-PC/G and /GX, 3290, and 5550. If they are not, get the level of support and customize as described in the *IBM 3274 Control Unit Planning, Setup, and Customization Guide.*

Specific questions, given in the Guide, must be answered to support graphics. These may be any of the following. (The answers you must give are shown.)

```
Q.121 Is this correctly selected for your language requirement?

Q.161 Color
A. 1..Yes

Q.162 SFAP
A. 1..Yes

Q.163 Extended Character Set Adapter (ECSA).
A. 1..for every device on the controller that
      has ECSA, but not greater than quantity in Q.112.
```

```
Q.164 PS
A. 1..Yes

Q.165 Decompression.
Normal recommendation is:
A. 0..for 3274 A and D models,
   1..for 3274 C models.

Q.166 Attribute Select Keyboard.
A. C..This encompasses all options.

Q.176 Decompression.
Normal recommendation is:
A. 1..for BSC Enhanced Communication Option (Distributed
          Function Terminals) applied to 3270-PC/G and
          3270-PC/GX.  This is known as WACK support.
```

If you already have the 3274 correctly customized, use the modification procedure described in the *IBM 3274 Control Unit Planning, Setup, and Customization Guide* to verify that the above options are defined. Then complete the documentation card held with the controller for future reference.

3174 controller diagnosis

To permit graphics on ASCII graphics terminals, a 3174 controller with an asynchronous emulation adapter (AEA) is required. The 3174 licensed internal code must be at Configuration Support Release B2 or later.

The 3174 must be customized to suit the specific ASCII graphics terminal you are using. This customization creates an AEA "Station Set" that defines the type and characteristics (for example, baud rate) of the terminal. Any local terminal-setup options must match the characteristics defined in the Station Set.

For terminals other than the DEC VT241 and Tektronix 4205, User Defined Tables (UDTs) are required in the 3174 AEA configuration. The graphics query reply field in the UDT must match one of the GDDM device token names for ASCII graphics terminals. These are:

DEC240
DEC241
DEC330
DEC330M
DEC340
DEC340M

TEK4105
TEK4205
TEK4205M
TEK4207
TEK4207M
TEK4208
TEK4208M
TEK4209
TEK4209M

Sample UDTs are given in the *GDDM Installation and System Management* guides.

Notes:

1. To use the full screen for GDDM graphics, set the alternate screen size in the 3174 AEA UDTs as follows:

DEC displays	0	(24 row only)
Tektronix 4105/4205	1	(30 row)
Tektronix 4207/4208/4209	2	(32 row)

2. The graphics input lineout for Tektronix terminals should be set to 400 msecs in the 3174 AEA UDTs.
3. Tektronix graphics displays require XON/XOFF flow control to avoid corruption of graphics data. Configure this using question 731 in the 3174 AEA station set as well as in the display.
4. DEC VT330 and VT340 terminals should be set to VT300 mode with the status line set to `HOST WRITABLE`.

For details about defining the AEA Station Set (customization of the 3174, questions 721 and onward) and the UDT, see:

3174 Establishment Controller: Customizing Guide
3174 Establishment Controller: AEA Description and Reference
3174 Establishment Controller: Terminal User's Reference for Expanded Functions.

Non-IBM devices diagnosis

For non-IBM devices, refer to any applicable publications or consult your device supplier.

For ASCII graphics terminals, see "3174 controller diagnosis" on page 14.

Common installation errors and pitfalls

The following section contains a list of common (and some not so common) errors and pitfalls that you may meet during the installation process. If you have a problem, check whether it is described below. Problems are listed in the following categories:

Problems associated with abends (on page 16):
The host software program is terminated with an abend code displayed or printed.

Problems associated with incorrect output (on page 17):
The output did not correspond to that expected or did not appear.

Problems associated with messages (on page 18):
An unexpected message was met.

Problems involving system performance (on page 22):
A degradation in performance occurred that could not be accounted for.

Problems involving device checks (on page 23):
A device "PROG" code or other code was displayed in the device's Operator Information Area.

Problems associated with SNA sense codes (on page 24):
An unexpected SNA sense code was encountered.

Problems associated with GCP (on page 25):
An unexpected GCP problem was encountered.

Within each category, the errors and pitfalls are listed in order of the code, message number, or other characteristic associated with the error.

Check for your problem in any of the categories that may seem appropriate.

Problems associated with abends

For a list of abend codes, see Appendix B, "Abend codes" on page 155.

User abend code 1064

Symptoms:
User abend code 1064.

Possible causes:
This abend applies to GDDM features, for example GDDM-PGF or GDDM-IVU. It occurs when an attempt is made to use a feature that is:

- Not installed
- Not reinstalled when necessary
- Not installed correctly
- In an area of storage that cannot be accessed by the user.

It can also be caused when a GDDM licensed program is installed and ADMGLIB is not the last parameter in the GLOBAL command parameter list.

Problem resolution:
If it is an installation problem, install or reinstall the feature correctly in an area that the user can access. If it is an ADMGLIB problem, make ADMGLIB the last parameter in the GLOBAL command parameter list.

User abend code 1201, 2201, or G201

Applicable devices:
3179-G1, 3179-G2, 3270-PC/G or 3270-PC/GX, or 5550.

Problem resolution:
This abend occurs if an incomplete alphanumerics defaults module (ADMDATRN) is used with a 3179-G1, 3179-G2, or 3472-G color display station, or with a 3270-PC/G or /GX workstation or a 5550 multistation. Refer to "Compatibility with previous releases" in the *GDDM Installation and System Management* manual for your system.

User abend codes 2053 and 2054

Symptoms:
User abend codes 2053 or 2054.

Applicable subsystems:
IMS/VS.

Problem resolution:
These abends occur instead of messages ADM0001, ADM0002, or ADM0003. Register 15 at the time of the abend will locate the corresponding message text. Look up the message in the *GDDM Messages* manual.

Problems associated with incorrect output

Thick black lines on 3812 output

Symptoms:
Thick black lines appear on 3812 output.

Applicable subsystems:
VM.

Applicable devices:
3812.

Problem resolution:
Either the required font has not been loaded or the wrong level of the VM3812 program is in use.

User session logoff

Symptoms:
Missing Interrupt conditions and consequent user session logoff.

Applicable subsystems:
VM.

Applicable devices:
3179-G1, 3179-G2, 3472-G, 3270-PC/G, or 3270-PC/GX.

Problem resolution:
For GDDM/VMXA systems, the matching message is HCPMHT2150I; this message is described in "Message HCPMHT2150I" on page 22.

Problems associated with messages

Messages beginning ADM

Symptoms:

`ADM.....` Any message beginning with the letters ADM.

Problem description:

The message may appear on the display screen, on printer output, or in a console log.

Problem resolution:

Look up the message in the *GDDM Messages* manual. The manual contains an explanation of the message and an indication of what to do. Also, check below for additional information relating to the specific messages.

Message ADM0275

Symptoms:

```
ADM0275 GRAPHICS {(IMAGE)} CANNOT BE SHOWN, REASON CODE n
```

Problem resolution:

Look up the message in the *GDDM Messages* manual.

If the reason code shows that system tables are at fault, check the system tables, including VTAM bind definitions where appropriate.

If the reason code shows that the device is at fault, read the section "Checking hardware characteristics" on page 8.

If you are trying to display graphics using the GDDM-PCLK program, check the GDDM-PCLK procopt. Also make sure that you "hot-key" and do not press ENTER when the PC is opened by the host application program. See the *GDDM Messages* manual for more details.

Message ADM0275, reason code 9

Symptoms:

```
ADM0275 GRAPHICS {(IMAGE)} CANNOT BE SHOWN, REASON CODE 9
```

Applicable devices:

3270-PC/G and /GX workstations, 4224 printer.

Problem resolution:

Message ADM0275, reason code 9, will occur for a 3270-PC/G or /GX when the device has PS support but no graphics support.

Even when such a device has been customized with graphics support, it will dynamically suppress the graphics support when it is SNA-attached and its VTAM MODEENT SRCVPAC specification is incorrect. Refer to "Checking a VTAM network" in the *GDDM Installation and System Management* manual for your system.

Messages beginning AEM

Symptoms:

`AEM.....` Any message beginning with the letters AEM.

Applicable subsystems:

All except IMS/VS.

Problem description:

The message may appear on the display screen, on printer output, or in a console log. It comes from GDDM Interactive Map Definition.

Problem resolution:

Look up the message in the *GDDM Messages* manual. The manual contains an explanation of the message and an indication of what to do.

Message DFS0089

Symptoms:

```
DFS0089I OUTPUT EXCEEDS BUFFER SIZE. LTERM .... NODE ....
```

Applicable subsystems:

IMS/VS.

Applicable devices:

3270 displays.

Applicable environment:

Local, non-SNA attachment.

Problem resolution:

For local non-SNA displays on IMS/VS, the OUTBUF parameter must be coded to define a buffer large enough to hold a complete output message. The maximum buffer size of 32 000 is recommended for such displays. For further details, refer to the *GDDM Installation and System Management for MVS* manual and the *GDDM Performance Guide*.

Message DFS2078

Symptoms:

```
DFS2078 O/P REJECTED SENSE nnnnnnnn
```

Applicable subsystems:

IMS/VS.

Problem description:

Message displayed on IMS Master Console.

Problem resolution:

This message contains the sense/status bytes returned by VTAM. Check at the device to see if a device PROG error code has been displayed in the Operator Information Area. Then check the sections "Problems involving device checks" on page 23 and "Problems associated with SNA sense codes" on page 24 for information relating to the specific codes.

Check the appropriate VTAM and hardware manuals for further information on the sense code.

Message DFS971I

Symptoms:

```
DFS971I I/O ERROR NODE ...., nnnnnnnn, xxxx
```

Applicable subsystems:
IMS/VS.

Problem description:
Message displayed on IMS Master Console.

Problem resolution:
This message relates to an I/O error communicating with a device. "nnnnnnnn" represents sense/status bytes returned by VTAM. Check the section "Problems associated with SNA sense codes" on page 24 for information relating to the specific code.

See the appropriate VTAM manuals for more information on the sense code.

Message DMKDID546I

Symptoms:

```
DMKDID546I INTERRUPTION <PENDING|CLEARED> ........
```

Applicable subsystems:
VM.

Applicable devices:
3179-G1, 3179-G2, 3472-G, 3270-PC/G, or 3270-PC/GX.

Problem resolution:
Highly complex graphic output can sometimes incur a significant length of processing time in 3179-G1, 3179-G2, or 3472-G color display stations, or in 3270-PC/G or /GX workstations before they indicate the completion of an I/O operation. If the time involved exceeds the host system's "missing interrupt time interval" an error condition may be raised and the user session logged off. On VM/CMS, the default interval is 30 seconds. See the *GDDM Performance Guide*, indexed under "time-outs VM/CMS," for information on increasing this interval.

Messages beginning EAK

Symptoms:
`EAK.....` Any message beginning with the letters EAK.

Problem description:
The message may appear on the display screen, on printer output, or in a console log. It comes from GDDM-CSPF.

Problem resolution:
Look up the message in the *GDDM Messages* manual. The manual contains an explanation of the message and an indication of what to do.

Messages beginning ERX

Symptoms:

`ERX.....` Any message beginning with the letters ERX.

Problem description:

The message may appear on the display screen, on printer output, or in a console log. It comes from GDDM-REXX.

Problem resolution:

Look up the message in the *GDDM Messages* manual. The manual contains an explanation of the message and an indication of what to do.

Messages beginning GQD

Symptoms:

`GQD.....` Any message beginning with the letters GQD.

Problem description:

The message may appear on the display screen, on printer output, or in a console log. It comes from GDDM-PCLK.

Problem resolution:

Look up the message in the *GDDM Messages* manual. The manual contains an explanation of the message and an indication of what to do.

Messages beginning GQF

Symptoms:

`GQF.....` Any message beginning with the letters GQF.

Problem description:

The message may appear on the display screen, on printer output, or in a console log. It comes from GDDM-OS/2 Link.

Problem resolution:

Look up the message in the *GDDM Messages* manual. The manual contains an explanation of the message and an indication of what to do.

Message IST211I

Symptoms:

```
IST211I NCP SLOWDOWN INITIATED FOR ....
```

Applicable subsystems:

| CICS on MVS, or IMS/VS.

Applicable access methods:

VTAM remote.

Problem resolution:

In a remote SNA configuration, graphic data streams can affect response times for non-GDDM users, and can initiate NCP SLOWDOWN conditions. You should review your NCP generation, as described in "Tuning and customization by subsystem" in Chapter 2 of the *GDDM Performance Guide*, paying particular attention to buffer, PACING, and VPACING parameters. If in doubt, specify PACING=(2,1) and VPACING=2 on relevant SNA PU macros in your NCP generation.

Message HCPMHT2150I

Symptoms:

```
HCPMHT2150I devtype addr AN INTERRUPT IS PENDING
```

Applicable subsystems:
VM.

Applicable devices:
3179-G1, 3179-G2, 3270-PC/G, 3472-G, or 3270-PC/GX.

Problem resolution:
Highly complex graphic output can sometimes incur a significant length of processing time in 3179-G1, 3179-G2, or 3472-G color display stations, or in 3270-PC/G or /GX workstations before they indicate the completion of an I/O operation. If the time involved exceeds the host system's "missing interrupt time interval" an error condition may be raised and the user session logged off. On VM/CMS, the default interval is 30 seconds. See the *GDDM Performance Guide*, indexed under "time-outs VM/CMS," for information on increasing this interval.

Problems involving system performance

Line time-outs

Symptoms:
Line time-outs.

Applicable devices:
3179-G1, 3179-G2, 3472-G, 3270-PC/G or 3270-PC/GX, or 5550.

Applicable access methods:
Remote (link-attached) non-SNA environments.

Problem resolution:
Communication line time-outs may occur for a 3179-G1, 3179-G2, or 3472-G color display station, or for a 3270-PC/G or /GX workstation, or a 5550 multistation if the device is BSC-attached but the associated 3274 controller has not been configured with WACK support. For more information refer to "Preparing your workstations" in the *GDDM Installation and System Management* manual for your system.

Missing interrupt conditions

Symptoms:
Missing interrupt conditions and consequent user session logoff.

Applicable subsystems:
VM.

Applicable devices:
3179-G1, 3179-G2, 3472-G, 3270-PC/G, or 3270-PC/GX.

Problem resolution:
For GDDM/VMXA systems, refer to the description in "Message HCPMHT2150I"above.

NCP SLOWDOWN conditions for non-GDDM users

Symptoms:
Poor response times for non-GDDM users, with NCP SLOWDOWN conditions initiated.

Applicable subsystems:
| CICS on MVS, or IMS/VS.

Applicable access methods:
VTAM remote.

Problem resolution:
Refer to the description in "Message IST211I" on page 21.

Problems involving device checks

Machine check 207

Symptoms:
Machine check 207.

Applicable devices:
3270-PC/G and /GX workstations.

Problem resolution:
This machine check may be indicated on a 3270-PC/G or /GX if required patches or customization options have not been applied to the associated 3274 controller. See the *GDDM Installation and System Management* manual for your system.

Device check PROG752

Symptoms:
PROG752.

Applicable devices:
3270 family displays and printers.

Problem resolution:
This device PROG error code is usually caused by a bad device definition. Check that the terminal definition in the GDDM System Definition data base matches the device characteristics (as configured, where appropriate). Under IMS, also check that:

- The terminal definition and device characteristics match the IMS TERMINAL macro SIZE operand and, where applicable, the VTAM bind definition (in particular, the screen size)
- The VTAM bind definition conforms to those shown in *GDDM Installation and System Management for MVS* manual.

Program check

Symptoms:
Program checks in mixed fields.

Applicable devices:
5550 Multistations.

Problem resolution:
Changes in data stream in fields that include a mixture of single-byte and double-byte character sets (SBCS/DBCS) may result in program checks on 5550 systems. If the subsystem and communications functions permit, you can avoid this problem by using a sufficiently large transmission buffer.

Problems associated with SNA sense codes

Sense code 1005

Symptoms:
Sense code 1005.

Applicable subsystems:
IMS/VS, or TSO, or CICS on MVS.

Applicable devices:
3270 family displays and printers.

Problem resolution:
This SNA sense code is usually caused by a bad device definition. Check that the terminal definition in the GDDM System Definition data base matches the device characteristics (as configured, where appropriate). Under IMS, also check that:

- The terminal definition and device characteristics match the IMS TERMINAL macro SIZE operand and, where applicable, the VTAM bind definition (in particular, the screen size)
- The VTAM bind definition conforms to those shown in *GDDM Installation and System Management for MVS* manual.

Sense code 800A

Symptoms:
Sense code 800A.

Applicable subsystems:
IMS/VS, or TSO, or CICS on MVS.

Applicable devices:
3270 family displays and printers.

Applicable access methods:
VTAM in a remote non-SNA environment.

Problem resolution:
This SNA Sense code can arise if you try to use GDDM on a non-SNA remote VTAM-attached terminal on IMS/VS.

GDDM is supported on IMS/VS through VTAM, only for local or SNA-attached terminals.

Problems associated with GCP errors

User symptoms

GCP hangs with X clock in the indicator row during a nonretained host graphics outgoing data stream. SY005 component informational 08690000 occurs unnecessarily on a /GX during a nonretained host graphics outgoing data stream. Drawing performance of a nonretained host graphics outgoing data stream may be (adversely or favorably) affected by this problem. The problem occurs only if the segment storage exceeds 86KB and can be bypassed by recustomizing with a different segment store value (but could then still occur with a different data stream if the segment store remains above 86KB). Because segment storage is limited to 63KB, this problem can never occur on a /GX with GCP 1.12 or GCP 2.10.

Chapter 2. Debugging aids

General-use programming interface

GDDM provides several facilities for an application programmer to help with debugging:

- Error messages

  Normally, when an error occurs, an error message is sent to the user console.

  There are exceptions to this:

  For the **TSO Print Utility**, error messages are sent to the system console. The system operator has access to this.

  Batch programs have no console. When you are running batch programs, you must ensure that the application program you are using checks for errors. If you are using GDDM in a VSE Batch environment, all error messages are saved in the VSE Recorder file.

  The **Composite Document Print Utility** prints error messages on an extra page (or pages) at the end of the document.

  If you are using **GDDM-GKS**, error messages are normally written to the GKS error file. Details of this and exceptions to it are given in the *GDDM Installation and System Management* manual for your system.

  A full list of the error messages with explanations is given in the *GDDM Messages* manual.

- Most recent error query. Following a program request, GDDM returns an error record for analysis by the programmer.
- User-exit facility. Control is passed to an exit whenever an error occurs above a specified threshold.
- Trace. GDDM trace is described in Chapter 3, "GDDM tracing" on page 37.

  If you are using GDDM-REXX, you can also use the tracing facilities in GDDM-REXX and those in REXX itself. These facilities are described in Chapter 5, "GDDM-REXX diagnosis" on page 93.

GDDM error messages

Almost every call made to GDDM can result in an error if the parameters passed to GDDM are incorrect or if the call cannot be processed correctly. When an error occurs, GDDM sends a pair of error messages to the user terminal. These messages give the error number, the error severity letter, and the message text (possibly including inserted variable text).

Here is an example of such a pair of messages:

```
ADM0055 E DSUSE, AT X'4E0202FE'
ADM0082 E DEVICE DOES NOT EXIST
```

The first message gives the name of the incorrect call and its address in main storage, and the second describes the error.

Each message has an associated letter to indicate the severity level:

Severity	Letter	Meaning
0	I	Information
4	W	Warning
8	E	Error
12	S	Severe error
16	U	Unrecoverable error.

Here are examples of the second message of the pair, for each level of severity:

```
ADM0258 I PRINT GENERATED ON 22 MAY 1988 AT 09.14.43 BY SMITH

ADM0156 W COORDINATE OUTSIDE PICTURE SPACE

ADM0082 E DEVICE DOES NOT EXIST

ADM0415 S DEVICE INACTIVE OR OFFLINE, X'061'

ADM0001 U GDDM STORAGE ANCHOR IS INVALID OR HAS NOT BEEN INITIALIZED
```

After issuing the error messages, GDDM returns control to the application program and execution continues with the next statement.

You can write a routine to gain control when an error occurs, using the FSEXIT call, described later in this chapter. If you use FSEXIT, you can specify an error threshold; when the error is less severe than this threshold, the exit routine is not called.

Besides issuing the two messages, GDDM builds an error record that may be accessed by the program. The error record is described on page 29. You can write a routine using the call FSQERR (Query Last Error) to return to the program the error record that reflects the most recent error. FSQERR is described on page 31.

It is good practice to test the error return code field after all critical GDDM calls (for example, DSOPEN) or groups of calls and to invoke user-written error-handling routines as required.

Error record structure

Note: The error record is subject to change from release to release of GDDM, or as a result of maintenance.

The error record has a length of 160 bytes, and contains both numeric and alphabetic information. The format is:

Offset	Field	Data type	Field description
0	Severity	fullword binary	0 - 16, the error return code.
4	Error number	fullword binary	Specifies which error occurred.
8	Function name	8 characters	Name of the failing function.
16	Message length	fullword binary	Length of message text.
20	Message text	80 characters	Text of the error message.
100	Entry-point function code	fullword binary	GDDM request code parameter.
104	Parameter list pointer	fullword binary	Address of the user's parameter list.
108	Return address	fullword binary	Return address to user's program.
112	Arithmetic insert 1	fullword binary or floating point	Numeric message insert. Depends on the error.
116	Arithmetic insert 2	fullword binary or floating point	Numeric message insert. Depends on the error.
120	Character insert 1	20 characters	Character message insert. Depends on the error.
140	Character insert 2	20 characters	Character message insert. Depends on the error.

The fields are:

1. Severity

   A fullword binary integer denoting the error severity:

0	Informative (or no error)
4	Warning
8	Error (function call ignored)
12	Severe error (resultant state unpredictable)
16	Unrecoverable (not passed to an error exit).

2. Error number

   A fullword binary integer identifying the error. The numbers correspond to the error-message numbers (listed in the *GDDM Messages* manual) without the three-letter prefix. The number is zero if no error has occurred.

3. Function name

   Two fullwords (eight characters) containing the name of the function whose invocation caused the error. If the error exit threshold is zero or less, the function name is the GDDM function called. For FSQERR, the field contains blanks if no error has occurred since the last call to FSQERR, or since initialization.

4. Message length

   A fullword binary integer containing the length of the message, excluding trailing blanks. The maximum length is 80. The length is zero if there is no error.

5. Message text

   The text of the error message associated with the error number, padded with trailing blanks, if necessary, to fill the 80-character length.

6. Entry-point function code

   A fullword binary integer (the request control parameter (RCP) code) representing the GDDM function invoked. (Refer to the *GDDM Base Programming Reference* manual for the list of RCP codes.) If the error-exit threshold is zero or less, the RCP code is the last GDDM function called. For FSQERR, the number is zero if no error has occurred since the last call to FSQERR, or if no error has occurred since initialization.

7. Parameter list pointer

   A fullword containing the contents of Register 1 when the call was made. This pointer enables you to obtain the parameters for the call that generated the error. If the error-exit threshold is zero or less, the information in this field is that for the last GDDM function called. Note that this value may not be relevant if the parameter-list area has been reused by the application program. This may happen when FSQERR is invoked.

8. Return address

   A fullword containing the contents of Register 14 when the call was made. This pointer enables you to obtain the storage address of the call that generated the error. If the error-exit threshold is zero or less, the information relates to the last GDDM function called.

9. Arithmetic insert 1

   A fullword binary integer or fullword floating-point number whose content depends on the error. Inserts are shown in the appropriate error messages, which are listed and described in the *GDDM Messages* manual. This field is zero if no error has occurred.

10. Arithmetic insert 2

    As arithmetic insert 1, described above.

11. Character insert 1

    Five fullwords (20 characters); the contents depend on the error, as noted for the arithmetic inserts. The field is blank if no error has occurred.

12. Character insert 2

    As character insert 1, described in item 11.

CALL FSQERR – query last error

This call returns to the program the error record for the most recent error. Informational messages are not considered to be errors, so the error reported is of warning level or above. Here is a typical call:

```
DCL 1 ERROR_RECORD,
      2 SEVERITY FIXED BIN(31),
      2 NUMBER FIXED BIN(31),
      2 FUNCTION_NAME CHAR(8),
      2 MSG_LENG FIXED BIN(31),
      2 MSG_TEXT CHAR(80),
      2 FUNCTION_CODE FIXED BIN(31),
      2 PARMLIST_PTR POINTER,
      2 RET_ADDR POINTER,
      2 ARITH_INSERT1 FIXED BIN(31),
      2 ARITH_INSERT2 FIXED BIN(31),
      2 CHAR_INSERT1 CHAR(20),
      2 CHAR_INSERT2 CHAR(20);

CALL FSQERR(RECORD_LENGTH,ERROR_RECORD);
                /* RETURN WHOLE ERROR RECORD FOR */
                /* THE MOST RECENT ERROR         */
```

The first parameter (in the example, RECORD_LENGTH) specifies in bytes the length of the second parameter (in the example, ERROR_RECORD). The second parameter is the name of a variable where GDDM returns all or part of the error record. The example returns the complete error record. You could use it in a program to analyze errors, or to present the error messages in some unusual format. The program might want to maintain, on auxiliary storage, a record of errors.

More commonly, you might want to test whether a particular GDDM call (or group of calls) had executed successfully. FSQERR returns the most recent error *since the last FSQERR*. It is not usually enough to place an FSQERR after the call in question. You may be given an error record corresponding to a GDDM call made some time before. To ensure that the error record (if any) corresponds to the particular call you want to verify, you must execute an FSQERR call just before the one you want to test (except for the first GDDM call in the program).

```
DCL 1 ERROR_RECORD,
      2 SEVERITY FIXED BIN(31),
      2 ERROR_NUMBER FIXED BIN(31);

/*******************************/
/* CLEAR ERROR RECORD (IF ANY) */
/*******************************/
CALL FSQERR(8,ERROR_RECORD);    /* CLEAR PREVIOUS ERROR RECORD        */
/*******************************/
/* EXECUTE CALL TO BE CHECKED  */
/*******************************/
CALL ASDFMT(7,8,DFMT_ATTRS);    /* REDEFINE PAGE'S ALPHA FIELDS       */
/*******************************/
/* QUERY  ERROR                */
/*******************************/
CALL FSQERR(8,ERROR_RECORD);    /* SEE IF ASDFMT RESULTED IN AN ERROR */

IF SEVERITY > 4
   THEN GOTO ABORT;             /* IF ALPHA REDEFINE FAILED, END RUN  */

    Continue normal processing...
```

You request (and declare) only that part of the error record in which you are interested.

CALL FSEXIT – specify error exit and threshold

This call specifies a user routine that will gain control when an error of specified severity occurs. This is a typical call:

```
CALL FSEXIT(DIAG66,8);          /* GIVE CONTROL TO ROUTINE DIAG66 IF AN */
                                /* ERROR OF SEVERITY 8 OR HIGHER OCCURS */
```

If an application program is using the nonreentrant interface, the named routine is passed one parameter: the GDDM error record, described on page 29. If the reentrant or system-programmer interface is used, the routine is passed two parameters. The first is the application anchor block (AAB), previously passed by the application program to GDDM; the second is the GDDM error record.

FSEXIT has these characteristics:

- In PL/I programs, the name of the error exit routine must be declared as an external entry, otherwise GDDM is unable to pass the error record as a parameter.
- In COBOL programs, user error exits cannot be specified, but FSEXIT can still be used to specify a threshold for the default error exit.
- If no error exit is explicitly specified (by calling FSEXIT), the default error exit applies. This exit is called following all errors of severity 4 or higher (8 or higher on IMS). It presents the error message to the user console and returns control to the program.
- The default error exit can be specified in an FSEXIT call by setting the first parameter to zero.

  To ensure the correct data type for this parameter, use this call:

  ```
  CALL FSEXIT(BINARY(0,31),8); /* CALL DEFAULT EXIT TO PRESENT ERROR    */
                               /* MESSAGES IF SEVERITY IS 8 OR MORE     */
  ```

  This call suppresses messages of warning level. Only messages of severity 8 or more are sent to the terminal.
- You can call the default exit after every GDDM call. The effect of this is to send a trace to the terminal of all the GDDM calls that have been executed. This is the statement needed:

  ```
  CALL FSEXIT(BINARY(0,31),0); /* CALL DEFAULT EXIT AFTER EVERY  */
                               /* CALL TO TRACE THE PROGRAM FLOW */
  ```

There are more details of this call in the *GDDM Base Programming Reference* manual.

Here is an example of an error exit routine:

```
DCL DERROR EXTERNAL ENTRY;
CALL FSEXIT(DERROR,8);

/*        .         */
/*        .         */
/*        .         */

DERROR: PROC(ERROR_RECORD) OPTIONS(COBOL);
DCL DCODE FIXED BIN (31) EXTERNAL;        /* COMMUNICATE WITH APPLN.    */
DCL 1 ERROR_RECORD,                       /* GDDM ERROR RECORD.         */
      2 SEVERITY FIXED BIN (31),          /* SEVERITY.                  */
      2 NUMBER   FIXED BIN (31),          /* ERROR MESSAGE NUMBER.      */
      2 FUNCTION CHAR(8),                 /* GDDM FUNCTION GIVING ERROR.*/
      2 MSGLEN   FIXED BIN (31),          /* LENGTH OF MESSAGE TEXT.    */
      2 MSGTEXT  CHAR(80),                /* MESSAGE TEXT.              */
      2 RCP      FIXED BIN (31),          /* GDDM RCP.                  */
      2 PLISTPTR FIXED,BIN (31),          /* PARAMETER LIST POINTER.    */
      2 RETADDR  FIXED BIN (31),          /* RETURN ADDRESS.            */
      2 AI1      FIXED BIN (31),          /* MESSAGE INSERT 1.          */
      2 AI2      FIXED BIN (31),          /* MESSAGE INSERT 2.          */
      2 CI1      CHAR(20),                /* CHARACTER MESSAGE INSERT 1.*/
      2 CI2      CHAR(20);                /* CHARACTER MESSAGE INSERT 2.*/
IF FUNCTION = 'DSOPEN'                    /* DSOPEN HAS FAILED BECAUSE  */
 & NUMBER =  97 THEN                      /* THERE IS NOT A PLOTTER.    */
  DCODE = 4;                              /*                            */
ELSE IF FUNCTION = 'GSLOAD'               /* GSLOAD HAS FAILED WITH AN  */
 & NUMBER = 303 THEN                      /* UNRECOGNIZED FILE FORMAT.  */
  DCODE = 8;                              /*                            */
END DERROR;                               /******************************/
```

Instead of declaring the error routine to be an external entry, you may choose to execute an FSQERR call to obtain the error record:

```
CALL FSEXIT(EERROR,8);                    /* SPECIFY ERROR EXIT.        */

/*        .         */
/*        .         */
/*        .         */
                                          /*                            */
EERROR: PROC(DUMMY);                      /* TRAP GDDM ERROR.           */
DCL DUMMY CHAR(*);                        /* NOT USED FOR INTERNAL RTN.*/
DCL DCODE FIXED BIN (31) EXTERNAL;        /* COMMUNICATE WITH APPLN.    */
DCL 1 ERROR_RECORD,                       /* GDDM ERROR RECORD.         */
     2 SEVERITY FIXED BIN (31),           /* SEVERITY.                  */

/*        .         */
/*        .         */
/*        .         */

     2 CI2      CHAR(20);                 /* CHARACTER MESSAGE INSERT 2*/

CALL FSQERR(160,ERROR_RECORD);            /* GET ERROR RECORD STRUCTURE*/
IF FUNCTION = 'DSOPEN' THEN               /* DSOPEN FOR PLOTTER HAS     */
  DCODE = 4;                              /* FAILED.                    */
ELSE IF FUNCTION = 'GSLOAD' THEN          /* GSLOAD HAS FAILED.         */
  DCODE = 8;
END EERROR;
```

Other methods

GDDM defaults mechanism

You can tell GDDM to return error information in a control block, instead of sending messages to the terminal. You specify your requirement using the GDDM ERRFDBK external default. This can be done with a SPINIT call or an ESEUDS call, or in the GDDM defaults module. Details are given in the *GDDM Base Programming Reference* manual.

Information returned in register 15

If you are using a programming language that allows you access to registers, you can get error information from register 15. On return from a call to GDDM, the top half of this register contains the error severity code and the bottom half the error number.

Reentrant and system programmer interfaces

Error information, consisting of an error code and a severity code, is supplied by GDDM in the application anchor block (AAB). Details are given in the *GDDM Base Programming Reference, Volume 2* manual.

End of General-use programming interface

Chapter 3. GDDM tracing

Product-sensitive programming interface

You can choose to run a GDDM trace as part of your diagnosis procedure, or your IBM Support Center Representative may ask you to run a trace, to provide more information to help solve your problem. In the latter instance, you may be provided with trace statements to include in your program.

Version 2 of GDDM provides an improved system of tracing. The methods used previously are still valid, and are described later in this chapter, but the recommended way to run GDDM trace is to write a set of statements using the GDDM default keyword, **TRCESTR**, like this:

```
[label] ADMMDFT TRCESTR='IF TSI THEN DO                          '
        ADMMDFT TRCESTR='  FLOW ;                                '
        ADMMDFT TRCESTR='  FULLIO ;                              '
        ADMMDFT TRCESTR='  TIME                                  '
        ADMMDFT TRCESTR='END                                     '
```

This chapter describes what you can do with TRCESTR, how to invoke it, and how to write TRCESTR statements. It describes the functions you can specify and the format of trace output. Examples of TRCESTR statements, together with the output produced by them, are given. The chapter then describes the methods of tracing that were previously available and that may still be used. Finally, there is a section that tells you how to locate GDDM control blocks, including the in-storage trace table, from a dump.

Notes:

1. A GDDM trace can produce large amounts of output. Try to be as restrictive as possible when specifying the part of the program to be traced and the amount of output needed.
2. For help in analyzing the data produced by the trace, see also *3270 Information Display System: Data Stream Programmer's Reference* and *Intelligent Printer Data Stream Reference.*

The TRCESTR default keyword

When tracing with TRCESTR, you can specify:

- The level at which the trace is to run (module, component, subcomponent, API)
- A component or module to be traced
- Whether to trace on entry to, or exit from, the item being traced, or both
- The number of times a particular set of conditions must occur before a trace record is produced
- Particular occurrences or ranges of events to be traced

- Your own abend codes
- The type of trace records to be produced
- Whether the contents of floating-point, or general-purpose registers, or both are to be included in the trace output
- Whether to include storage use in your trace output
- A full or partial I/O trace
- 5080 tracing.

Starting trace

This section tells you how to start tracing in each of these system environments:

- TSO and the TSO print utility
- CMS
- CICS
- IMS/VS
- VSE batch.

It also covers tracing in an application program.

Tracing under TSO

The most flexible way to start tracing is with an external default file. To use this method:

1. Ensure that you have allocated a suitable data set to receive the trace output. It is not necessary to supply DCB characteristics for this data set.

   The trace records are written using a sequential file with a default ddname of ADMTRACE, which should be allocated to a suitable data set or SYSOUT destination before GDDM is invoked.

   If the output is going to a tape, the data set must be allocated with DSORG=PS.

2. Place your TRCESTR statements in an external defaults data set. The default file name for the external defaults data set is ADMDEFS.

3. Include in your job stream this DD statement:

   ```
   //ADMDEFS DD DSN=your-external-defaults-dataset
   ```

4. Run the application program or utility that you want to trace.

Remember to disable tracing when you no longer require it. You can do this by erasing the TRCESTR statements from your defaults file, or by turning them into comments. To do this, place an asterisk (*) in column 1 of each TRCESTR default statement.

Tracing in the TSO print utility environment

When tracing in the TSO print utility, trace records are written using sequential files. The trace output for the utility subtask (ADMOPST) associated with any one printer is written using a sequential file with a ddname the same as the LUNAME of the printer. These ddnames should be assigned to suitable data sets or SYSOUT destinations in the job control language (JCL) for the utility. In this way, the trace output from the utility can be limited to that associated with a specific printer.

Tracing under CMS

The most flexible way to start tracing is with an external default file. To use this method:

1. Place your trace statements in your defaults file, PROFILE ADMDEFS, and ensure that the file is on your A-disk.
2. Specify a destination for the trace output.

   The trace records are normally written to a sequential file with the file-identifier ADM00001 ADMTRACE A1. If you want to change the file-name or file-type used, you can do so by specifying the ADMMDFT CMSTRCE external default in the source format defaults file. If the ADMMDFT CMSTRCE value is set to blank, the trace records are written to the virtual printer, using spool file processing. To direct trace output to the virtual printer, you should include the following default specification in the external defaults file:

   ```
   [label] ADMMDFT CMSTRCE=(,)
   ```

   If you do not have enough space on your A-disk, you should spool your trace output to a virtual printer.
3. Run the application program or utility that you want to trace.

Remember to disable tracing when you have finished. You can do this by erasing the TRCESTR statements from your defaults file, or by commenting them out. To comment them out, place an asterisk (*) in column 1 of each TRCESTR default statement.

Tracing under CICS

With CICS, it is generally not possible to create a permanent external defaults file, like that available to TSO or VM users. However, a restricted external defaults file mechanism is available to CICS users, mainly for debugging purposes.

To run GDDM trace under CICS:

1. Determine the CICS identification of the terminal on which you intend to run GDDM; you can get this information using the CEMT transaction.
2. Allocate a data set to receive the trace output. On MVS CICS systems, if the output is going to a tape, the data set must be allocated with DSORG=PS.

   The trace records are written by standard CICS services to a single transient data destination, with a default name of ADMT. This must be defined in the

CICS Destination Control Table (DCT), in a way to suit the installation's requirements. Typically, the destination would be defined as an extra partition destination, which would direct the trace records to an external data set for printing later. This destination name can be changed by altering the ADMMDFT CICTRCE default.

3. Create a temporary storage queue, containing the replacement defaults. This queue must have a name of the form ADMDxxxx, where **xxxx** is the CICS terminal-identifier. (The prefix ADMD can be changed in a defaults module or through a SPINIT call statement, using the default CICDFPX=aaaa.)

   If you are suitably authorized, you can create a temporary storage queue using the CECI transaction. This transaction is documented in the *CICS/VS Application Programmer's Reference Manual (Command Level)*, which includes a discussion of the security and authorization aspects of the transaction.

   The replacement defaults must be in source format, as described in the *GDDM Base Programming Reference* and *GDDM Installation and System Management* manuals.

   For example, to turn trace on, enter the following transactions:

   ```
   CECI DELETEQ TS QUEUE('ADMDxxxx')
   CECI WRITEQ  TS QUEUE('ADMDxxxx')
                   FROM(' ADMMDFT TRCESTR="FLOW"')
   ```

   (Remember that you must have a blank before ADMMDFT and additional quotes round imbedded strings.)

   If you have more than one TRCESTR statement, you must code a FROM statement for each ADMMDFT statement, thus:

   ```
   CECI DELETEQ TS QUEUE('ADMDxxxx')
   CECI WRITEQ  TS QUEUE('ADMDxxxx')
                   FROM(' ADMMDFT TRCESTR="IF TSI THEN FLOW"')
                   FROM(' ADMMDFT TRCESTR="PARTIO"')
   ```

4. Run the application program or utility that you want to trace.

Remember to disable tracing when you have finished. The defaults established remain applicable until the Temporary Storage Queue is deleted, either explicitly, or automatically at CICS termination. They are not normally retained after CICS termination.

Tracing under IMS/VS

With IMS/VS systems, you cannot use an external source defaults file. You must therefore include your trace statements in the external defaults module (ADMADFI). This module is either link-edited with your application program or GDDM utility program, or is in your library data set. Changing ADMADFI is described in the *GDDM Installation and System Management* manual for IMS/VS systems.

Run a GDDM trace as follows:

1. Allocate a data set to receive the trace output. If the output is to go to a tape, the data set must be allocated with DSORG=PS.

   The trace records are written using a sequential file with a default name of ADMTRACE. This should be allocated to a suitable data set or SYSOUT destination before GDDM is invoked.

2. Ensure that your trace statements are included in the external defaults module.

3. Run the application program or utility that you want to trace.

Remember to disable trace, when you no longer want to run it, by restoring the defaults module to its original state.

Tracing in the VSE Batch environment

The most flexible way to start tracing is to include tracing defaults in the job stream, immediately after the invocation of GDDM. To use this method:

1. Ensure that you have allocated a suitable VSAM data set to receive the trace output.

2. Include in your job stream the following statement:

   ```
   // DLBL ADMTRACE,'your_output_file_name',,VSAM
   ```

3. Include any TRCESTR statements (and other defaults or nicknames) immediately after the invocation of GDDM.

Remember to disable tracing when you no longer require it. You can do this by erasing the TRCESTR statements from the job stream, or by commenting them out. To comment them out, place an asterisk (*) in column 1 of each TRCESTR default statement.

Tracing in an application program

Tracing can be initiated by an application call, for example, ESEUDS. Details of this method are given in "Other tracing methods" on page 75.

Note: The trace statements required for tracing from an application call differ from those required when using an external defaults file or module:

- The trace statements in an external file or module may be spread over several lines or ADMMDFT statements.
- A single trace statement in an application call must conform to the grammar for a complete program. For example, all parts of an IF THEN ELSE statement must be within a single application call. However, a trace specification may be built up from any number of separate application calls; these will be appended to the trace program formed from an external file, if there is one.

Tracing multiple instances of GDDM

If you are running under TSO or VM/CMS, the GDDM TRCESHR default facility allows you to trace more than one instance of GDDM to the same trace file. This is particularly useful in a windowing environment, where you may have more than one application to be traced at the same time. You may also want to trace combinations of task managers and applications.

The facilities provided to support tracing are:

- If only one instance of GDDM is to be traced, tracing will proceed as usual.
- If two or more GDDM instances are to be traced and each instance uses a different trace file, there is no contention for the trace files and tracing will proceed as usual. Each instance may be directed to use a different trace file with suitable trace output defaults.

  For example, in one instance of GDDM use:

  ```
  CMSTRCE=(INST0001,ADMTRACE)
  ```

  and in the second instance use:

  ```
  CMSTRCE=(INST0002,ADMTRACE)
  ```

- If two GDDM instances are traced, and both instances try to use the same trace file, the second GDDM instance to attempt tracing will fail, because the trace file can only be shared serially.
- You can code a GDDM **TRCESHR** default statement to specify that a trace file is to be shared among instances of GDDM. If an instance of GDDM is initialized with this default set, both it and subsequently initialized instances of GDDM that specify tracing to the same trace file will use the one file.

  The format of the TRCESHR statement is:

  ```
  [label] ADMMDFT TRCESHR=YES|NO
  ```

  Use **ADMMDFT TRCESHR=YES** to specify that a trace file is to be shared.

  There must be a blank before **ADMMDFT**.

  Note: Specifying this default means that the instance of GDDM is to be a coordinator for shared use of the trace file. The coordinating instance must not be terminated before all other instances sharing the trace file have been terminated.

Trace file records are prefixed with a three-digit number that identifies the instance of GDDM that generated the record. The numbers are assigned to the instances of GDDM in the order that the instances begin to use the trace file. Page 71 shows an example of such trace output.

Trace keywords

The table below shows the source syntax and defaults for each of the keywords associated with tracing.

Keyword	Meaning of the keyword	Source syntax of the keyword	GDDM default
TRCESTR	Trace options	**TRCESTR='xxxxxxxxx'**	None
TRCEWID	Trace output width	**TRCEWID={SINGLE|DOUBLE}**	SINGLE
TRTABLE	Trace table size, in-core	**TRTABLE=***n*	100
TRCESHR	Trace share	**TRCESHR={NO|YES}**	NO
TRACE	Trace word value	**TRACE={0|***n***}**	0
CICTRCE	Trace output transient data name	**CICTRCE=***aaaa*	ADMT
IMSTRCE	Trace output ddname	**IMSTRCE=***aaaaaaaa*	ADMTRACE
TSOTRCE	Trace output ddname	**TSOTRCE=***aaaaaaaa*	ADMTRACE
CMSTRCE	Trace output (filename,filetype)	**CMSTRCE=(***aaaaaaaa,bbbbbbbb***)**	(ADM00001, ADMTRACE)

Details of coding TRCESTR statements begin below. Full details of the syntax of TRCESTR statements are in Appendix D, "Trace-string grammar" on page 193.

Details of the TRCEWID statement are on page 59.

The other keywords are described in the *GDDM Base Programming Reference, Volume 2.*

Coding TRCESTR statements

A set of TRCESTR statements consists of one or more

```
    ADMMDFT TRCESTR='xxxxxxxxxxxxx'
```

statements in a defaults module or file. Note that:

- You must have a space before ADMMDFT.
- If you are including your TRCESTR statements in a default *file*, rather than a default *module*, you can begin the statements with DEFAULT instead of ADMMDFT.
- Each character string may be up to 256 characters long.
- All the TRCESTR statements in the defaults module or file are used to make up a single trace program; this program must conform to the syntax defined in Appendix D.

A set of trace statements may consist of one action or several actions to be executed whenever the trace processor is invoked. It can be as simple as:

```
[label] ADMMDFT TRCESTR='FLOW'
```

This statement specifies that trace is to be invoked and that standard trace statements (known as flow statements) are to be produced in the trace output. Every module boundary will be traced on entry and exit.

Remember that the ADMMDFT statement must *not* begin in column 1.

You can restrict the amount of trace output produced. If, for example, you want trace output from the ICU only, you can specify:

```
[label] ADMMDFT TRCESTR='IF ICU THEN FLOW'
```

You can specify several actions in one statement. For example, the statement:

```
[label] ADMMDFT TRCESTR='IF ICU THEN FLOW PARTIO TIME'
```

requests a partial transmission trace from the ICU, with standard output. Specifying TIME causes a time stamp to be included in the trace output for each traced module.

The general form of a statement is:

```
[label] ADMMDFT TRCESTR='IF event THEN action(s) ELSE action(s)'
```

The event and action functions that you can specify are defined in "Functions available with the TRCESTR keyword" starting on page 46.

Compound statements are allowed. Use semicolons to separate all statements within the compound statement. The previous example could be written:

```
[label] ADMMDFT TRCESTR='IF ICU THEN DO FLOW; PARTIO; TIME END'
```

IF statements may be nested:

```
[label] ADMMDFT TRCESTR='IF ESI THEN IF COUNT(4) THEN DO FLOW ;
        ADMMDFT TRCESTR='PARTIO; TIME END'
```

They may be nested within compound statements:

```
[label] ADMMDFT TRCESTR='IF TSI THEN                          '
        ADMMDFT TRCESTR=' DO                                  '
        ADMMDFT TRCESTR='  FLOW ;                             '
        ADMMDFT TRCESTR='  IF NAME(''ADMLN*'') THEN           '
        ADMMDFT TRCESTR='   DO                                '
        ADMMDFT TRCESTR='    LIST(100,200);                   '
        ADMMDFT TRCESTR='    PARTIO(ON)                       '
        ADMMDFT TRCESTR='   END ;                             '
        ADMMDFT TRCESTR='  TIME                               '
        ADMMDFT TRCESTR=' END                                 '
```

Note: If you specify a character string in a TRCESTR statement, such as ADMLN* in the example above, you must add extra quotes, either by using double quotes or by adding another pair of single quotes. The relational expression in an IF statement can consist of a combination of events. For example:

```
[label] ADMMDFT TRCESTR='IF CMPNT & (ICU | FSM) THEN FLOW'
```

This example can also be written:

```
[label] ADMMDFT TRCESTR='IF CMPNT AND (ICU OR FSM) THEN FLOW'
```

Signed numeric values can be expressed as binary numbers (B'011100'), decimal numbers (−123445) or hexadecimal numbers (X'789AB').

The contents of an address (%), a general-purpose register (GR), or a floating-point register (FR), may be referred to in a relational expression or as a parameter:

```
[label] ADMMDFT TRCESTR='IF ENTRY & ESI THEN LIST((1 GR +4)%,4)'
        ADMMDFT TRCESTR='IF 13 GR >= X''0F1B2'' THEN FLOW'
```

Note: If you use % to access an address for which trace does not have read access, an abnormal termination will occur in module ADMATPX for the unacceptable address. You can trace on a component RCP code (described in the *GDDM Base Programming Reference* manual). For example, to trace entry and exit parameters for ASREAD:

```
[label] ADMMDFT TRCESTR='DO IF CMPNT THEN                          '
        ADMMDFT TRCESTR='   IF (1 GR +4)%%=X''C100000'' THEN  '
        ADMMDFT TRCESTR='   PARMSF   END                           '
```

In this example, nesting ensures that the address is evaluated *only* after CMPNT is satisfied, thus eliminating the risk of an abend.

All the relational expressions in an IF statement are evaluated on every occasion (no jumping code), thus the following two examples are equivalent:

```
[label] ADMMDFT TRCESTR='IF COUNT(4) & LIMIT(2,12) THEN FLOW'
        ADMMDFT TRCESTR='IF LIMIT(2,12) & COUNT(4) THEN FLOW'
```

If you want the effect that would be produced by jumping code, you must explicitly code nested IF statements:

```
[label] ADMMDFT TRCESTR='IF COUNT(4) THEN IF LIMIT(2,12) THEN FLOW'
        ADMMDFT TRCESTR='IF LIMIT(2,12) THEN IF COUNT(4) THEN FLOW'
```

You can use the following operators in TRCESTR statements. Where words are used, uppercase and lowercase letters are both acceptable.

+	
−	
*	
/	
&	and
|	or
¬	not
=	eq
¬=	ne
<	lt
>	gt
<=	le
>=	ge

The % sign may be used to indicate the contents of an address, as described on page 45. It is also used as a token on the PRINT function, as described on page 51.

Note: If you are compiling or assembling the TRCESTR statements, ampersands (&) within the statements must be repeated. For example:

```
[label] ADMMDFT TRCESTR='IF CMPNT && ICU THEN FLOW'
```

You can put a CLEAR statement at the start of your set of TRCESTR statements and a FORCE statement at the end. These control the priority if you have more than one set of TRCESTR statements. The next section of this chapter gives examples of these functions.

Functions available with the TRCESTR keyword

In TRCESTR statements you can use:

- A CLEAR function
- A FORCE function
- Action functions
- Event functions
- Control functions.

These are described in the following sections. Later in the chapter you will find working examples, showing sample trace output.

CLEAR

You can start your set of TRCESTR statements with a CLEAR statement:

```
[label] ADMMDFT TRCESTR='CLEAR              '
        ADMMDFT TRCESTR='IF ENTRY THEN FLOW '
```

If you specify CLEAR, any preceding TRCESTR statements are ignored, unless FORCE has already been invoked. You would normally only use CLEAR in an application call for tracing, for example, an ESSUDS call.

FORCE

At the end of your set of statements, you can specify FORCE:

```
[label] ADMMDFT TRCESTR='IF ENTRY THEN FLOW '
        ADMMDFT TRCESTR='FORCE              '
```

This causes subsequent TRCESTR statements and subsequent calls to FSTRCE to be ignored, even if they contain the CLEAR keyword. That is, FORCE overrides all application FSTRCE calls and non-defaults-file tracing.

Action functions

The action functions that you can specify are summarized in this table:

Function	**Arguments**
ABEND	(*number*)
CDPDS	[({**ON**|OFF|*control*})]
CGMREP	[({**ON**|OFF|*control*})]

Function	Arguments
DSOPEN	[({ON|OFF|*control*})]
FLOW	none
FREGS	none
FULLIO	[({ON|OFF|*control*})]
FULLTCA	[({ON|OFF|*control*})]
GREGS	none
HRIG	[({ON|OFF|*control*})]
LIST	(*low_address*[, *length*])
NULL	none
PARMSF	[(*number*)]
PARTIO	[({ON|OFF|*control*})]
PRINT	("*control_string*" [, *remaining_parameters*])
STGREP	[({ON|OFF|*control*})]
TIME	none
5080IO	[({ON|OFF|*control* }[, *number*])]

All the actions for which you can specify a control parameter are *latched*, that is, the action is inactive until you specify it to be ON, and then remains ON until you specify it to be OFF. Details of the control parameter are given on page 58.

The order in which the actions take effect (that is, place information in the trace output file) is fixed to minimize duplication. For example, if two separate conditions are satisfied, both of which call for a time stamp, only one time stamp is placed in the trace file. The order in which the actions are evaluated is:

PRINT	These actions take effect in the order in which
LIST	they are evaluated by the trace interpreter.
FLOW	
PARMSF	
TIME	
GREGS	
FREGS	
STGREP	
ABEND	
CDPDS	The effect of these latching actions is
CGMREP	not directly related to their evaluation
DSOPEN	by the trace interpreter.
HRIG	
PARTIO	
FULLIO	
FULLTCA	
5080IO	

A detailed description of each action function is given below.

ABEND

ABEND(*number*)

GDDM is terminated abnormally with the specified abend number. The *number* must be a decimal number in the range 1 through 999.

CDPDS

CDPDS[({ON|OFF|*control*})]

If ON, this specifies that the composite-document presentation data stream (CDPDS) input is to be included in the trace output. The structure and content of the CDPDS are described in the *GDDM Base Programming Reference, Volume 2.*

If the header of a CDPDS structured field is invalid, the output includes a note of this and the first eight bytes of the field. If a field is less than eight bytes long, the output includes a note of this and the short record.

The *control* function is described on page 58.

Page 73 gives an example of trace output containing CDPDS input.

CGMREP

CGMREP[({ON|OFF|*control*})]

If set to ON, this specifies that descriptions of the CGM orders being processed during a CGLOAD or CGSAVE API call are to be included in the trace output.

The *control* function is described on page 58.

DSOPEN

DSOPEN[({ON|OFF|*control*})]

If ON, nickname processing within a DSOPEN call is traced so it can be checked.

The *control* function is described on page 58.

FLOW

FLOW

This specifies that standard trace records are to be included in the trace output. Standard trace records are those produced at all module entry and exit points.

FREGS

FREGS

This specifies that the contents of the four floating-point registers (FREGS) are to be included in the trace output.

FULLIO

FULLIO[({ON|OFF|*control*})]

If ON, this specifies that a full transmission trace is required. The full I/O buffer is included in the trace output for each transmit operation and each receive operation.

The *control* function is described on page 58.

FULLTCA

FULLTCA[({ON|OFF|*control*})]

If ON, this specifies that a full transmission trace with terminal services interface control area (TCA) blocks is required.

The key blocks in the TCA are the terminal descriptor block (TDB) and the terminal request block (TRB); these are described on pages 149 and 152, respectively.

The *control* function is described on page 58.

GREGS

GREGS

This specifies that the contents of the sixteen general-purpose registers (GREGS) are to be included in the trace output.

HRIG

HRIG[({ON|OFF|*control*})]

If ON, this specifies that the output data stream directed to a page printer is to be included in the trace output.

The *control* function is described on page 58.

Page 74 gives an example of trace output containing this data.

LIST

LIST(*low_address*[, *length*])

This specifies an area of storage to be listed in dump format.

The *low_address* parameter shows the target location of the beginning of the required storage area, and the optional *length* parameter specifies the length, in bytes, of the storage area to be listed. If the *length* is less than 1 or is not specified, 4 bytes are listed.

Note: If you try to display storage to which trace does not have read access, an abnormal termination occurs in module ADMATPX for the unacceptable address.

NULL

NULL

The NULL function performs no action. It is provided for use in nested IF THEN ELSE statements.

PARMSF

PARMSF[(*number*)]

This specifies that parameters of any external (API) calls are to be included in the trace output, in formatted notation. For any component calls other than API calls, or if the *number* parameter is set to zero, PARMSF produces the RCP (request control parameter) code and function title only. Subcomponent and module level tracing does not produce any output from PARMSF.

The optional *number* parameter specifies the maximum number of bytes that will be listed for each traced parameter. If it is not specified, a default limit of 80 bytes is applied. If *number* is negative, zero is assumed.

PARTIO

PARTIO[({ON|OFF|*control*})]

If ON, this specifies that a partial transmission trace is required. The data in the I/O buffer, up to a maximum of 16 bytes, is included in the trace output for each transmit operation and each receive operation. If the I/O buffer data exceeds 16 bytes, the first 16 bytes are included in the trace output with an indication that the data has been truncated and the length of the full data.

The *control* function is described on page 58.

PRINT

PRINT("*control_string*"[, *remaining_parameters*])

This causes a single line to be inserted in the trace output file.

The *control_string* parameter is a character string defining the format of the line. It specifies literal text and sufficient format controls for the following parameters. Format controls should be separated by the underscore character '_', which becomes a blank in the output.

The format controls are:

```
%JW.nc
```

% is a token indicating a substitution format control

J is an optional control either

+ specifying right justified, the default.

− specifying left justified.

W is an optional control that specifies the minimum field width. If no value is given, the PRINT function chooses a suitable value. If the value specified is too small, it is overridden with the value the PRINT function would have used, had no value been specified.

n is an optional control that specifies:

- For character strings, the maximum number of characters to be printed. If not specified, the default is all the characters in the string.
- For floating-point numbers, the number of digits to be printed to the right of the decimal point. If not specified, the default is 1. This is so that a floating-point number cannot be mistaken for an integer.

c is a control character that *must* be specified. It determines the type of formatting performed. Values may be:

B or b Binary format.

The corresponding parameter should be a fullword. Output has a single leading zero to minimize the binary string length.

D or d Decimal format.

The corresponding parameter should be a fullword.

E or e Exponential format.

The corresponding parameter should be in short form floating point format.

F or f Floating format.

The corresponding parameter should be in short form floating point format.

S or s Character string format.

The corresponding parameter should be a character string.

X or x Hexadecimal format.

The corresponding parameter should be a fullword.

The decimal point is only required if the optional *n* control is specified.

The *remaining_parameters* are the items that are to be formatied as directed by the control string and then sent to the trace file.

If there are more *remaining_parameters* than format controls, the excess parameters are ignored.

If there are more format controls than *remaining_parameters*, a single * is put in the trace file for each missing parameter.

Here are some examples of PRINT:

- `PRINT("Trace_example")` produces:

  ```
  Trace example
  ```

- `PRINT("%s","trace_example")` produces:

  ```
  trace example
  ```

- `PRINT("%20s","trace_example")` produces:

  ```
         trace example

  1---5----0----5----0----5----0 - (This line indicates where
                                    the characters appear.)
  ```

- `PRINT("%-20s","trace_example")` produces:

  ```
  trace example

  1---5----0----5----0----5----0
  ```

- `PRINT("%-20.10s","trace_example")` produces:

  ```
  trace example

  1---5----0----5----0----5----0
  ```

- `PRINT("Data_in_reg_%d_is_%x", 5, 5 GR)` produces:

  ```
  Data in reg 5 is 02447E
  ```

  if the data in general-purpose register 5 is X'02447E'.

Notes:

1. All spaces within a character string must be written as _ in the TRCESTR string.
2. Because they are within a TRCESTR statement, the *control_string*, and each of the *remaining_parameters* that is a character string, must be within either double quotes or a pair of single quotes.

STGREP

STGREP[({ON|OFF|*control*})]

If set to ON, this specifies that a storage use report is to be included in the trace output.

The report is normally included in the trace file on termination or immediately before a GDDM-initiated abend.

If a *control* parameter with a numerical value greater than 1 is included in the statement, any storage report already started is included in the trace file immediately and a new storage report is started.

The *control* function is described on page 58.

TIME

```
TIME
```

This specifies that a time stamp is to be included in the trace output. The time stamp includes date and time. Time is shown in two formats: hh:mm:ss and the same value expressed in seconds.

5080IO

5080IO[({**ON**|OFF|*control*}[, *number*])]

If ON, this specifies that calls to the 5080 Graphics System workstation (using GDDM/graPHIGS) are to be traced. The format of the trace entries is the same as the format for parameter trace entries for calls to GDDM base. The calls using GDDM/graPHIGS have GP as the first two letters of the call name.

The *number* parameter is optional. It specifies a limit for the number of bytes of data to be displayed. If this parameter is not specified, a default limit of 80 bytes is applied.

Event functions

Event functions, when used with an IF statement, restrict the amount of trace output produced by the action functions. That is, they qualify the conditions that, when satisfied, cause the specified trace actions.

The event functions that you can specify are summarized in this table:

Function	Syntax
ALL	ALL
Component	AIC|CDU|CSI|ESI|FSM|GKS|ICU|IMC|IMD|IMS|ISE|IVU|NUM|PGR|QPU|TSI|VSE
COUNT	(*number*)
ENTRY	ENTRY
EXIT	EXIT
Level	**MOD**|SCMPNT|CMPNT|API
LIMIT	LIMIT(*low_limit*[, *high_limit*])
NAME	NAME("*module_name*")
RANGE	RANGE(*low_event_number*[, *high_event_number*])

If you do not specify any event functions, the action functions are executed at the entry and exit of each module.

ALL

```
ALL
```

This specifies that the specified actions are to be executed at the entry and exit of each module.

Component

```
AIC|CDU|CSI|ESI|FSM|GKS|ICU|IMC|IMD|IMS|ISE|IVU|NUM|PGR|QPU|TSI|VSE
```

This specifies the component to be traced. If you specify the module or subcomponent level, those boundaries will be traced within the component you specify with this parameter. If you do not specify a component, the default is ALL.

You can specify more than one component by putting an 'or' sign (|) between the codes for the components you want to trace. The components and their associated name prefixes are as follows:

Code	**NAME**	**Component**
AIC	ADMAxxxx	Application Program Interface Component
CDU	ADM4xxxx	Composite Document Print Utility
CSI	ADMYxxxx	Common Services Interface
ESI	ADMExxxx	Environmental Services Interface
FSM	ADMDxxxx	Full Screen Manager
GKS	ADMJxxxx	Graphical Kernel System
ICU	ADMPxxxx	Interactive Chart Utility
IMC	ADM3xxxx	Image Manager Component
IMD	AEMxxxxx	GDDM Interactive Map Definition
IMS	ADMKxxxx	Information Management System
ISE	ADMIxxxx	Image Symbol Editor
IVU	ADM5xxxx	GDDM Image View Utility
NUM	ADMNxxxx	Numerical Processing Routines
PGR	ADMBxxxx	Presentation Graphics Routines
QPU	ADMOxxxx	Queued Print Utility
TSI	ADMLxxxx	Terminal Services Interface
VSE	ADMVxxxx	Vector Symbol Editor.

You can use the NAME prefixes, given in the table, with the NAME function (described on page 57) to be more selective in your trace operation. For example, specifying NAME("ADMA*") is equivalent to specifying AIC.

COUNT

COUNT(*number*)

This specifies that trace records are to be produced every time the number of occurrences of the specified event matches the value of the *number* parameter. If you do not specify any event functions, the entry and exit of each module is considered to be a specified event.

In the following example, trace records are produced every fourth time module ADMEF* is entered:

```
[label] ADMMDFT TRCESTR='IF ENTRY & NAME("ADMEF*") THEN     '
        ADMMDFT TRCESTR='     IF COUNT(4) THEN FLOW         '
```

ENTRY

ENTRY

This specifies that tracing of entries to units at the specified level is required.

EXIT

EXIT

This specifies that tracing of exits from units at the specified level is required.

Level

MOD|SCMPNT|CMPNT|API

These values specify the level at which tracing is to be carried out:

API	Application interface level
CMPNT	Component boundaries
SCMPNT	Subcomponent boundaries
MOD	Module boundaries.

MOD is the default value. The lower levels of tracing include the higher levels, so that MOD includes tracing at subcomponent, component, and API levels.

LIMIT

LIMIT(*low_limit*[, *high_limit*])

This specifies limits for tracing occurrences of a particular set of conditions. Trace records are produced every time the number of occurrences of the specified event lies between the values of the *low_limit* and *high_limit* parameters, inclusive. If you do not specify any event functions, the entry and exit of each module is considered to be a specified event.

The second parameter is optional; if it is omitted a value of infinity is used.

In the following example, trace records are produced for the third, fourth, and fifth times that module ADMEF* is entered:

```
[label] ADMMDFT TRCESTR='IF ENTRY & NAME("ADMEF*") THEN      '
        ADMMDFT TRCESTR='     IF LIMIT(3,5) THEN FLOW        '
```

NAME

NAME("*module_name*")

The *module_name* parameter is a character string that either:

1. Specifies a module name.
2. Consists of the first part of a module name followed by the character *. This specifies all modules whose names start with the characters given. For example:

   ```
   ADME* specifies ADMEAAA through ADMEZZZ
   ```

   Notes:

   a. Only one * is allowed and it must be the last character in the string.
   b. Because it is within a TRCESTR statement, the *module_name* must be within either double quotes or a pair of single quotes.

   The first part of each module name with the component that contains it is given in the table on page 55.

RANGE

RANGE(*low_event_number*[, *high_event_number*])

This specifies that trace records are to be produced for events with sequence numbers that lie between the values of the *low_event_number* and *high_event_number* parameters, inclusive.

Each module entry and exit is given an event sequence number; you can use these numbers to limit the trace output to the area that you are most interested in. If you omit the second parameter, a value of infinity is used.

Only those events that have numbers in the specified range, and meet the requirements of any action function specified, will appear in the trace output.

For example, trace output produced by the statement:

```
[label] ADMMDFT TRCESTR='FLOW'
```

may include trace records numbered from 1 through 900. If you see that the problem you are examining is shown in records 210 through 230, you can limit future trace output to this area by using the statement:

```
[label] ADMMDFT TRCESTR='IF RANGE(190,250) THEN FLOW'
```

Note: You should broaden the range specified from just the area that you are interested in; trace record numbers may not correspond exactly from run to run because the modules may be loaded in a different order.

Control function

({**ON**|OFF|*control*})

Some of the action functions described above have an optional control function associated with them. You can specify one of the following for this function:

ON The trace interpreter takes the action specified.

This is the default value for the control function.

OFF The trace interpreter does not take the action specified.

control You can specify a *control* relational expression. (The grammar of relational expressions is defined in Appendix D, "Trace-string grammar" on page 193.) The trace interpreter evaluates the expression:

- If the value is zero, or logical false, the action specified is not taken. This is equivalent to OFF.
- If the value is not zero, or logical true, the action specified is taken. This is equivalent to ON. For the STGREP action, values of greater than one cause a storage report to be produced immediately and a new storage report to be begun.

For example, *control* can be the name of an event function. The results of the following statements are the same:

```
[label] ADMMDFT TRCESTR='IF AIC OR FSM THEN STGREP(ON) ELSE STGREP(OFF)'
        ADMMDFT TRCESTR='STGREP(AIC|FSM)'
```

The TRCEWID keyword

Using the TRCEWID default keyword, you can specify the width of the trace output. The valid options are:

```
[label] ADMMDFT TRCEWID=SINGLE
        ADMMDFT TRCEWID=DOUBLE
```

The default is SINGLE, which produces 4-word hex output. DOUBLE produces 8-word hex output. This option reduces the amount of paper used.

End of Product-sensitive programming interface

Format of trace output

Trace output begins with the trace control word and a GDDM build identification. Unless a partial transmission trace is the only option specified, a formatted listing of the GDDM defaults table is be produced. Here is an explanation of some of the information contained in trace output.

Trace record type

A mnemonic shows the type of trace record:

CPNIN Shows that a GDDM component has been entered to perform the function listed.

CPNOUT Shows that a GDDM component has been left after performing the function listed.

ERFIN Shows that on entry to a GDDM component, the standard GDDM error feedback areas contained error information as listed.

ERFOUT Shows that on leaving a GDDM component, the standard GDDM error feedback areas contained error information as listed.

SUBIN Shows that the GDDM subcomponent listed has been entered.

SUBOUT Shows that the GDDM subcomponent listed has been left.

MODIN Shows that the GDDM module listed has been entered.

MODOUT Shows that the GDDM module listed has been left.

MSGOUT Is produced on leaving the message processor in GDDM. It contains the text of any message produced.

Trace explanation

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

CPNIN, CPNOUT

The trace explanation contains:

- The request control parameter (RCP) in mnemonic and hexadecimal format. The first byte of the RCP identifies the component being invoked, as follows:

X'00'	Application Interface Component (AIC)
X'04'	Terminal Services Interface Component (TSI)
X'08'	Environmental Services Interface Component (ESI)
X'0C'	Full Screen Manager (FSM)
X'10'	Presentation Graphics Routines (PGR)
X'14'	Interactive Chart Utility (ICU)
X'18'	Image Symbol Editor
X'1C'	Common Services Interface Component (CSI)
X'20'	Vector Symbol Editor
X'24'	Output Print Utility (OPU)
X'28'	IMS/VS Interactive Utility Scheduler (IMS)
X'2C'	Numerical Processing Routines (NMR)
X'30'	GDDM Interactive Map Definition (GDDM-IMD)
X'38'	GDDM Graphical Kernel System (GDDM-GKS)
X'3C'	Image Manager Component (IMC)
X'40'	Composite Document Print Utility (CDU)
X'44'	GDDM Image View Utility (GDDM-IVU).

- An interpretation of the RCP.
- For calls to the Environmental Services Interface (ESI), for storage, program load/delete, or abend processing, an interpretation of the parameters to those services.

ERFIN, ERFOUT

The trace explanation contains:

- The associated error message number, severity, and text. The string '(INTERNAL)' is produced for error codes that are purely internal to GDDM.

SUBIN, SUBOUT, MODIN, MODOUT

The trace explanation contains:

- The module identification, including the module name.
- For SUBOUT or MODOUT, the value of Register 15, if it appears to be a return code.

Examples of trace output from TRCESTR statements

The following pages show examples of output obtained from various sets of trace statements. In each case, the statements used are given.

The output from each trace begins with a listing of the default table. This is included in the first example, but thereafter is omitted. Other records may also be omitted, for the sake of clarity. Where records are omitted, this is shown.

Example 1. Tracing a range of events

To produce normal trace output records for events 205 through 215 this statement was used:

```
[label] ADMMDFT TRCESTR='IF RANGE(205,215) THEN FLOW'
```

The event sequence numbers are shown in the following output.

| **Note:** A description of ADMTDFT is given in "ADMTDFT — general defaults table" on page 133.

```
1GDDM DIAGNOSTIC TRACE - ORIGIN PATTERSN        02 AUG 1988 11:08 AM

 TRACE WORD = '00000000'X   BUILDID = 87315

 ADMTDFT  - GENERAL DEFAULTS TABLE COMMON SECTION
 00381EC8                       C4C6E340  000001E0    *        DFT ...\*
 00381ED0    00000001  00000000  00000000  00000000    *................*
 00381EE0    00000000  00000000  00000000  000000F8    *...............8*
 00381EF0    00000158  00000170  00120000  C1040101    *............A...*
 00381F00    7FD20000  00000004  00000002  0000D570    *"K............N.*
 00381F10    000000A4  00000000  00000064  00000000    *...u............*
 00381F20    00000D96  00000400  00002000  00000008    *...o............*
 00381F30    0000D2F8  00380CA8  00000000  00000000    *..K8...y........*
 00381F40    00000004  00000000  00000000  00000000    *................*
 00381F50    C1C6E3C3  F0F3F9F5  5C404040  40404040    *AFTC0395*       *
 00381F60    5C404040  40404040  5C404040  40404040    **       *       *
 00381F70    5C404040  40404040  5C404040  40404040    **       *       *
 00381F80    5C404040  40404040  5C404040  40404040    **       *       *
 00381F90    5C404040  40404040  00000000  00380628    **       ........*
 00381FA0    50050000  00000000  00000000  00000000    *&...............*
 00381FB0    00000025  0000015F  80000000  00000000    *.......¬........*
          - OBJECT TYPE DEFAULTS
 00381FC0    C1C4D4E2  E8D4C2D3  C1C4D4C7  C7D4C1D7    *ADMSYMBLADMGGMAP*
 00381FD0    C1C4D4E2  C1E5C540  C1C4D4C3  C6D6D9D4    *ADMSAVE ADMCFORM*
 00381FE0    C1C4D4C3  C4C1E3C1  C1C4D4E3  E4E3D7C7    *ADMCDATAADMTUTPG*
 00381FF0    C1C4D4C7  C4C64040  C1C4D4C7  D2E2D440    *ADMGDF  ADMGKSM *
 00382000    C1C4D4C3  C4C5C640  C1C4D4D7  D9D6D140    *ADMCDEF ADMPROJ *
 00382010    C1C4D4C9  D4C74040  C1C4D4D7  C3404040    *ADMIMG  ADMPC   *
          - USER EXIT DEFAULTS
 00382020    00000BBB  00000000  00000000  00000BBD    *................*
 00382030    00000000  00000000                        *........        *
          - SUBSYSTEM SPECIFIC DEFAULTS
 00382038                       80000000  00000000    *        ........*
 00382040    C1C4D4E4  E3F14040  C1C4D4C4  C5C3D240    *ADMUT1  ADMDECK *
 00382050    C1C4D4D7  D9C9D5E3  C1C4D4D3  C9E2E340    *ADMPRINTADMLIST *
 00382060    C3D6D7E8  40404040  C1C4D4C9  C6D4E340    *COPY    ADMIFMT *
 00382070    C1C4D4D4  E2D34040  C1C4D4C9  D4C1C7C5    *ADMMSL  ADMIMAGE*
 00382080    C1C4D4C3  D6D34E40  C1C4D4F0  F0F0F0F1    *ADMCOL+ ADM00001*
```

```
00382090    C1C4D4E3  D9C1C3C5  D7D9D6C6  C9D3C540    *ADMTRACEPROFILE *
003820A0    C1C4D4C4  C5C6E240                        *ADMDEFS         *
ADMTNICK - NICKNAME LIST ENTRY
00380CA8                        D5C9C3D2  00000054    *        NICK....*
00380CB0    00005F08  00000000  40000000  00380CE4    *..¬..... ......U*
00380CC0    00380CE4  00380CE4  00000001  00000000    *...U...U........*
00380CD0    00000000  00000000  40404040  40404040    *........        *
00380CE0    00000003  0000001D  00000001  00000006    *................*
00380CF0    00000000  00000000  00000000              *............    *
ADMTNICK - NICKNAME LIST ENTRY
00005F08                        D5C9C3D2  0000004C    *        NICK...<*
00005F10    00005F70  00380CA8  40000000  00005F44    *..¬....y .....¬.*
00005F20    00005F44  00005F44  00000001  00000000    *..¬...¬.........*
00005F30    00000000  00000000  40404040  40404040    *........        *
00005F40    00000002  0000001B  00000001  00000000    *................*
00005F50    00000000                                  *....            *
ADMTNICK - NICKNAME LIST ENTRY
00005F70    D5C9C3D2  0000004C  00380548  00005F08    *NICK...<......¬.*
00005F80    40000000  00005FAC  00005FAC  00005FAC    * .....¬...¬...¬.*
00005F90    00000001  00000000  00000000  00000000    *................*
00005FA0    40404040  40404040  00000002  00000021    *        ........*
00005FB0    00000001  00000000  00000000              *............    *
ADMTNICK - NICKNAME LIST ENTRY
00380548                        D5C9C3D2  00000054    *        NICK....*
00380550    003805B8  00005F70  40000000  00380584    *......¬. ......d*
00380560    0038058C  0038059C  00000000  00000001    *................*
00380570    00000001  00000002  40404040  40404040    *........        *
00380580    00000000  D7C3D7D3  E3404040  5C404040    *....PCPLT   *   *
00380590    40404040  C1C4D4D7  D3D6E340              *    ADMPLOT     *
ADMTNICK - NICKNAME LIST ENTRY
003805B8                        D5C9C3D2  00000054    *        NICK....*
003805C0    00000000  00380548  40000000  003805F4    *........ ......4*
003805D0    003805FC  0038060C  00000000  00000001    *................*
003805E0    00000001  00000002  40404040  40404040    *........        *
003805F0    00000000  D7C3D7D9  E3404040  5C404040    *....PCPRT   *   *
00380600    40404040  C1C4D4D7  C3D7D9E3              *    ADMPCPRT    *

DEFAULT TRCESTR='if range(205,215) then flow '

00000205 08 CPNIN  - - - -YSCNMP ('1C100000'X) - SCAN MODULE MAP
            MODIN  - - - -ADMYR000 86105 V2R1.1
00000206 09 MODIN  - - - - ADMYSMAP 86105 V2R1.1
00000207 09 MODOUT - - - - ADMYSMAP 86105 V2R1.1
00000208 08 MODOUT - - - -ADMYR000 86105 V2R1.1
            CPNOUT - - - -YSCNMP ('1C100000'X) - SCAN MODULE MAP
00000209 07 MODOUT - - - ADMEPROV 87218 V2R2.0
00000210 06 MODOUT - - -ADMEROOV 87315 V2R2.0
            CPNOUT - - -EPSLGU ('08040480'X) - UNCOND GROUP LOAD
            (ADMDN0   AT '00A5CFD0'X)
00000211 06 MODIN  - - -ADMDSRT 87315 V2R2.0
00000212 07 SUBIN  - - - ADMDNCRT 87209 V2R2.0
00000213 08 CPNIN  - - - -ESSGIC ('080C0441'X) - COND GET INST STG
            (000176)
            MODIN  - - - -ADMEROOV 87315 V2R2.0
00000214 08 MODOUT - - - -ADMEROOV 87315 V2R2.0
            CPNOUT - - - -ESSGIC ('080C0441'X) - COND GET INST STG
            (000176 AT '00373A40'X)
00000215 08 CPNIN  - - - -ESSGIC ('080C0441'X) - COND GET INST STG
            (000176)
            MODIN  - - - -ADMEROOV 87315 V2R2.0

00001251 END OF GDDM DIAGNOSTIC TRACE  02 AUG 1988 11:08 AM
```

Example 2. Restricting the level at which tracing takes place

Here is the statement for example 2:

```
[label] ADMMDFT TRCESTR='STGREP IF API THEN PARMSF'
```

This statement requests that a storage use report and call parameters (in formatted notation) are to be included in the trace output. No number is specified in the PARMSF parameter, so the default of 80 bytes will be listed for each traced parameter. Tracing is at the application interface (API) level. The statement produced this output:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN PATTERSN      02 AUG 1988 11:10 AM

 TRACE WORD = '00000000'X   BUILDID = 87315

 *------------------------------------------------------------------*
 *                                                                  *
 *  Trace defaults and nickname table omitted for clarity           *
 *                                                                  *
 *------------------------------------------------------------------*

 DEFAULT TRCESTR='stgrep if api then parmsf   '
 00000001 01 CPNIN  SPINIT ('00050000'X) - SPI SPECIAL INIT
 PTRACE    1 CHAR   '              P8'  00000000 00000010 00000030 0000D7F8
 PTRACE    1 CHAR   '                '  00000000 00000000 00000000 00000020
 PTRACE    1 CHAR   '  L       L    H'  0000D320 0000000A 0000D3A0 000000C8
 PTRACE    1 CHAR   '  M         '      0000D468 00000000 00000000
 00000028 01 CPNOUT SPINIT ('00050000'X) - SPI SPECIAL INIT
 PTRACE    1 CHAR   ---INPUT ONLY PARAMETER-----

 00000029 01 CPNIN  GSCHAR ('0C0C0500'X) - CHARACTER STRING AT
 PTRACE    1 FLOAT             50
 PTRACE    2 FLOAT             50
 PTRACE    3 DIM                5
 PTRACE    4 CHAR   'abcde'             81828384 85
 00000467 01 CPNOUT GSCHAR ('0C0C0500'X) - CHARACTER STRING AT
 PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    3 DIM    ---INPUT ONLY PARAMETER-----
 PTRACE    4 CHAR   ---INPUT ONLY PARAMETER-----

 00000468 01 CPNIN  GSMOVE ('0C0C0400'X) - MOVE TO
 PTRACE    1 FLOAT             40
 PTRACE    2 FLOAT             60
 00000473 01 CPNOUT GSMOVE ('0C0C0400'X) - MOVE TO
 PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----

 00000474 01 CPNIN  GSLINE ('0C0C0401'X) - LINE TO
 PTRACE    1 FLOAT             70
 PTRACE    2 FLOAT             60
 00000483 01 CPNOUT GSLINE ('0C0C0401'X) - LINE TO
 PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----

 00000484 01 CPNIN  ASREAD ('0C100000'X) - READ
 PTRACE    1 FIXED  ---OUTPUT ONLY PARAMETER-----
 PTRACE    2 FIXED  ---OUTPUT ONLY PARAMETER-----
 PTRACE    3 FIXED  ---OUTPUT ONLY PARAMETER-----
 00000791 01 CPNOUT ASREAD ('0C100000'X) - READ
```

```
PTRACE    1 FIXED             0
PTRACE    2 FIXED             0
PTRACE    3 FIXED             0

00000792 01 CPNIN  FSTERM ('0C000000'X) - TERMINATION
00001173 01 CPNOUT FSTERM ('0C000000'X) - TERMINATION

STORAGE USE REPORT

LENGTH  AT ADDRESS  GOT BY   (SEQ)      FREED BY (SEQ)

000212 '0037EA50'X  ADMDSRO  00000013   ADMDSRO  00001162
000164 '0037EB40'X  ADMACFP  00000017
001024 '00378508'X  ADMERO0V 00000020                     (SMALL STORAGE POOL)
000256 '00378520'X  ADMACFP  00000021
000164 '00378638'X  ADMACFP  00000023
000164 '003786F8'X  ADMACFP  00000025
000172 '003787B8'X  ADMACFP  00000027
000012 '003804F8'X  ADMDSNN  00000036   ADMDSNN  00000039
000020 '00380910'X  ADMDSNN  00000038   ADMDSNN  00000043
000028 '00380D18'X  ADMDSNN  00000042   ADMDSNN  00000047
000028 '00005EC8'X  ADMDSNN  00000046   ADMDSDS  00000232
000084 '0037EC00'X  ADMDSDS  00000051   ADMDSDS  00000230
000624 '00379368'X  ADMDSDS  00000055   ADMDSDS  00001159
000622 '003795F0'X  ADMLIN1V 00000067   ADMLTM1V 00001143
000028 '00380D18'X  ADMYRSRL 00000071   ADMYRSRL 00001139
003478 '003721B8'X  ADMLAC1V 00000077   ADMLRL1V 00000104
001024 '00371DA0'X  ADMERO0V 00000098   ADMERO0V 00001156 (SMALL STORAGE POOL)
000172 '00371DB8'X  ADMDSDS  00000099   ADMDSDS  00001155
000125 '00378880'X  ADMDSDS  00000101   ADMDSDS  00000228
000366 '00371E80'X  ADMDSF1  00000122   ADMDSCL  00001115
000094 '00372008'X  ADMDSII  00000125   ADMDSII  00001108
000069 '00372080'X  ADMDSII  00000127   ADMDSII  00001110
001024 '00372B50'X  ADMERO0V 00000131   ADMERO0V 00001122 (SMALL STORAGE POOL)
000292 '00372B68'X  ADMDSF1  00000132   ADMDSCL  00001121
002000 '00372368'X  ADMDSQL  00000141   ADMDSQL  00000142
000144 '003720E0'X  ADMDSQP  00000147   ADMDSQP  00000148
000684 '00379878'X  ADMDSDQ  00000153   ADMDSCL  00001117
000684 '00379B40'X  ADMDSDQ  00000159   ADMDSCL  00001119
000222 '00372CA8'X  ADMDWINT 00000176   ADMDWTRM 00001013
000040 '003720E0'X  ADMDOOC  00000194   ADMDOOC  00000801
000176 '00372DA0'X  ADMDNCRT 00000214   ADMDNDEL 00000963
000176 '00372E68'X  ADMDNCRT 00000216   ADMDNDEL 00000955
001024 '00372738'X  ADMERO0V 00000219   ADMERO0V 00001037 (SMALL STORAGE POOL)
000176 '00372750'X  ADMDNCRT 00000220   ADMDNDEL 00000957
000044 '00372120'X  ADMDNCRT 00000222   ADMDNDEL 00000961
000068 '0037EC00'X  ADMDBCRT 00000251   ADMDBDEL 00000946
000072 '00378880'X  ADMDECRT 00000271   ADMDEDEL 00000941
004096 '00362110'X  ADMERO0V 00000291   ADMERO0V 00001098 (SMALL STORAGE POOL)
000708 '00362128'X  ADMDCPC  00000292   ADMDCPC  00000937
000260 '00372818'X  ADMDGI1  00000352   ADMDGI1  00001036
000420 '00372938'X  ADMDGWI  00000372   ADMDGWI  00000929
001024 '00372320'X  ADMERO0V 00000377   ADMERO0V 00001102 (SMALL STORAGE POOL)
000124 '00372338'X  ADMDGS1  00000378   ADMDGS1  00000921
004080 '0034C008'X  ADMYGQC  00000382   ADMYGQC  00000917
002024 '0034B808'X  ADMDGPAX 00000390   ADMDGPAX 00000887
001024 '00371988'X  ADMERO0V 00000395   ADMERO0V 00000907 (SMALL STORAGE POOL)
000272 '003719A0'X  ADMYGQC  00000396   ADMYGQC  00000883
000260 '003723D0'X  ADMDGSE  00000403   ADMDGSE  00000876
004080 '0034A800'X  ADMYGQC  00000411   ADMYGQC  00000863
003486 '00349A48'X  ADMDGS1  00000439   ADMDGS1  00000779
003478 '00348C98'X  ADMLAC1V 00000511   ADMLSN1V 00000689
```

```
000248 '003724F0'X   ADMDHIN  00000528   ADMDSCL  00001101
000512 '00362408'X   ADMDHIN  00000530   ADMDSCL  00001097
000024 '00372F30'X   ADMDHIN  00000532   ADMDHDF  00000910
000256 '00371AC8'X   ADMDHCO  00000535   ADMDHDF  00000906
004096 '00347C80'X   ADMER00V 00000538   ADMER00V 00000903 (SMALL STORAGE POOL)
001024 '00347C98'X   ADMDHCO  00000539   ADMDHDF  00000902
005120 '00346868'X   ADMDHCO  00000541   ADMDHSS  00000897
004560 '00345680'X   ADMDHUW  00000578   ADMDCES  00001062
002572 '00344C58'X   ADMDXSCD 00000644   ADMDNDEL 00000959
003478 '00341260'X   ADMDDSGO 00000649   ADMDDSGO 00000666
003488 '003404A8'X   ADMDDSGO 00000651   ADMDDSGO 00000668
003478 '00348C98'X   ADMLAC1V 00000699   ADMLRL1V 00000765
003478 '00341260'X   ADMDDSGO 00000719   ADMDDSGO 00000732
003488 '003404A8'X   ADMDDSGO 00000721   ADMDDSGO 00000734
000084 '00380CA8'X                       ADMACUP  00001183
000076 '00005F08'X                       ADMACUP  00001185
000076 '00005F70'X                       ADMACUP  00001187
000084 '00380548'X                       ADMACUP  00001189
000084 '003805B8'X                       ADMACUP  00001191
000128 '00380670'X                       ADMATPE  00001194
000108 '00380708'X                       ADMATPE  00001196
000108 '00380790'X                       ADMATPE  00001198
000068 '00380818'X                       ADMATPE  00001200
000128 '00380878'X                       ADMATPE  00001202
000048 '00380130'X                       ADMATPE  00001204
000828 '00379010'X                       ADMATPE  00001206
004096 '00378FF8'X                       ADMER00V 00001207
000278 '00380178'X                       ADMATPE  00001210
000048 '00380628'X                       ADMACUP  00001213
001024 '00380530'X                       ADMER00V 00001214

00001251 END OF GDDM DIAGNOSTIC TRACE  02 AUG 1988 11:10 AM
```

Note: The > symbol following PTRACE shows that more parameters exist than were traced (because of the 80 byte default limit for PARMSF).

Example 3. Requesting a full transmission trace

The statement for this is:

```
[label] ADMMDFT TRCESTR='FULLIO'
```

The trace output is:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN PATTERSN        02 AUG 1988 11:11 AM

 TRACE WORD = '00000000'X   BUILDID = 87315

 *-------------------------------------------------------------------*
 *                                                                   *
 *  Trace defaults and nickname table omitted for clarity            *
 *                                                                   *
 *-------------------------------------------------------------------*

 DEFAULT TRCESTR='fullio  '

 OUTBOUND TRANSMISSION SENT

 I/O BUFFER                          LENGTH = 00002    TIME  11:11:40
 00372E58                        7EC2               *       =B      *
```

```
OUTBOUND TRANSMISSION SENT

I/O BUFFER                          LENGTH = 00006    TIME  11:11:40
00372E58                          F3000501  FF02      *        3.....  *

INBOUND TRANSMISSION RECEIVED

I/O BUFFER                          LENGTH = 00126    TIME  11:11:40
00372E58                          88001681  86000800  *        h..af...*
00372E60    F4F1F1F2  F2F3F3F4  F4F5F5F6  F6F7F700  *411223344556677.*
00372E70    0D818704  00F0F1F1  F2F2F4F4  00258185  *.ag..0112244..ae*
00372E80    B0000910  40000000  03000000  0100F102  *.... .........1.*
00372E90    80C10380  E304C0E4  05C0FF06  80FF07C0  *.A..T.{U.{.....{*
00372EA0    FF000781  88000102  00178181  01000050  *...ah.....aa...&*
00372EB0    00200000  0A02E500  02006F09  0C0A0000  *......V...?.....*
00372EC0    1181A600  000B0100  00500018  00500020  *.aw......&...&..*
00372ED0    00068198  8000                          *..aq..          *

PREPARING PAGE FOR PARTITION

0037888C                                    00000000  *            ....*

OUTBOUND TRANSMISSION SENT

NON-ZERO RETURN CODE ON TRANSMISSION. ADMTDISP CONTROL BLOCK:

00379760    00000000  00000000  FFFF4001  00EA0004  *.......... .....*
00379770    1134A251  202000EA  00000000  00000000  *..s.............*
00379780    00000000  00000000  00000000  00000000  *................*
00379790    C8D5C4C9  D5E34040  C3D3D940  00000009  *HNDINT  CLR ....*
003797A0    00000000  00000000  FFFFFFFF  00000000  *................*

I/O BUFFER                          LENGTH = 00235    TIME  11:11:42
0034A250    F3000403  80003D06  21E24102  00400000  *3........S... ..*
0034A260    00000000  000000FF  00000000  00000040  *............... *
0034A270    00000000  00000000  00000000  00000000  *................*
0034A280    00000000  00000000  00000000  00000000  *................*
0034A290    0000009A  4000F1C1  11004F28  43E24311  *.... .1A..|..S..*
0034A2A0    009F4311  00EF4311  013F4311  018F4311  *................*
0034A2B0    01DF4311  022F4311  027F4311  02CF4311  *........."......*
0034A2C0    031F4311  036F4311  03BF4311  03E02842  *.....?.......\..*
0034A2D0    F43C03F8  41421104  5F431104  AF431104  *4..8....¬.......*
0034A2E0    D8284300  81828384  8511054F  2843E243  *Q...abcde..|..S.*
0034A2F0    11059F43  1105EF43  11063F43  11068F43  *................*
0034A300    1106DF43  11072F43  11077F43  1107CF43  *..........".....*
0034A310    11081F43  11086F43  1108BF43  11090F43  *......?.........*
0034A320    11095F43  1109AF43  1109FF43  000F4000  *..¬........... .*
0034A330    F1F21109  FF1D6011  000013              *12....-....     *

SCREEN REFRESH

OUTBOUND TRANSMISSION SENT

I/O BUFFER                          LENGTH = 00005    TIME  11:11:43
0034A250    7EF01140  40                            *=0.             *

OUTBOUND TRANSMISSION SENT

I/O BUFFER                          LENGTH = 00231    TIME  11:11:44
0034A250    F3003D06  21E24102  00400000  00000000  *3....S... ......*
0034A260    000000FF  00000000  00000040  00000000  *........... ....*
```

```
0034A270    00000000  00000000  00000000  00000000     *................*
0034A280    00000000  00000000  00000000  0000009A     *................*
0034A290    4000F1F0  11004F28  43E24311  009F4311     * .10..|..S......*
0034A2A0    00EF4311  013F4311  018F4311  01DF4311     *................*
0034A2B0    022F4311  027F4311  02CF4311  031F4311     *....."..........*
0034A2C0    036F4311  03BF4311  03E02842  F43C03F8     *.?.......\..4..8*
0034A2D0    41421104  5F431104  AF431104  D8284300     *....¬.......Q...*
0034A2E0    81828384  8511054F  2843E243  11059F43     *abcde..|..S.....*
0034A2F0    1105EF43  11063F43  11068F43  1106DF43     *................*
0034A300    11072F43  11077F43  1107CF43  11081F43     *......".........*
0034A310    11086F43  1108BF43  11090F43  11095F43     *..?...........¬.*
0034A320    1109AF43  1109FF43  000F4000  F1F21109     *.......... .12..*
0034A330    FF1D6011  000013                           *..-....         *

INBOUND TRANSMISSION RECEIVED

I/O BUFFER                          LENGTH = 00003     TIME  11:11:44
0034A250    7D4040                                     *'               *

00001251 END OF GDDM DIAGNOSTIC TRACE  02 AUG 1988 11:11 AM
```

Example 4. Including the contents of registers in your trace

You may find it helpful to include the contents of the general-purpose registers (GREGS), or the floating-point registers (FREGS), or both, in your trace output like this:

```
[label] ADMMDFT TRCESTR='IF COUNT(100) THEN FLOW GREGS FREGS'
```

The trace output for this statement is:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN PATTERSN        02 AUG 1988 11:13 AM

 TRACE WORD = '00000000'X   BUILDID = 87315

 *---------------------------------------------------------------*
 *                                                               *
 *  Trace defaults and nickname table omitted for clarity        *
 *                                                               *
 *---------------------------------------------------------------*

 DEFAULT TRCESTR='if count(100) then flow gregs fregs '

 00000100 03 CPNIN  - ESSGLU ('080C0410'X) - UNCOND GET LOCAL STG   (000125)
             MODIN  - ADMER00V 87315 V2R2.0
 GENERAL PURPOSE REGISTERS
 GPR0      =  0038167C  00381550  00372E59  00379668
 GPR4      =  00B2ACFD  00000054  0037EA50  00380D50
 GPR8      =  00381230  00379368  0037EC00  40B290C0
 GPR12     =  00B2A0BF  00381448  40B2998A  009E20C0
 FLOATING POINT REGISTERS
 FPR0     =  42000000  80000000             0   7.6293900E-06
 FPR2     =  42320000  80000000   5.000000E+01   5.0000000E+01
 FPR4     =  00000000  00000000             0               0
 FPR6     =  00000000  00000000             0               0
 00000200 04 SUBOUT - -ADMDSSST 87323 V2R2.0
 GENERAL PURPOSE REGISTERS
 GPR0      =  00000A00  0038180C  003798B4  00379668
 GPR4      =  00379668  00000009  00372B20  00B2AD25
 GPR8      =  00000012  00379368  00A60056  40A5E058
```

```
GPR12    =  00A5F057  00381718  60A5F6D8  00B26130
FLOATING POINT REGISTERS
FPR0    =  43B65526  A13F5000   2.917322E+03   2.9173220E+03
FPR2    =  40BFBDAC  98000000   7.489879E-01   7.4898800E-01
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00000300 02 MODOUT -ADMDSRT 87315 V2R2.0
GENERAL PURPOSE REGISTERS
GPR0     =  00000000  00A64C94  00000000  00381EC8
GPR4     =  0000DAD4  0037F164  0037EA50  00380D50
GPR8     =  00A65116  00379368  0037F164  60A64600
GPR12    =  00380D50  00381230  70A64B64  00B27F50
FLOATING POINT REGISTERS
FPR0    =  43B65526  A13F5000   2.917322E+03   2.9173220E+03
FPR2    =  40BFBDAC  98000000   7.489879E-01   7.4898800E-01
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00000400 10 SUBOUT - - - - -ADMDGPA 87268 V2R2.0
GENERAL PURPOSE REGISTERS
GPR0     =  0037377C  00AAA908  0C0C000C  445B252C
GPR4     =  00AB15C8  00AAEE05  00000000  0C0C030F
GPR8     =  003735D8  00379368  00AAA6C6  40AA86C8
GPR12    =  00AA96C7  000057A8  50AAA65C  00AB6C50
FLOATING POINT REGISTERS
FPR0    =  432D9296  80000000   7.291616E+02   7.2916170E+02
FPR2    =  42200000  00000000   3.200000E+01   3.2000000E+01
FPR4    =  00000000  00000000              0               0
FPR6    =  44800000  00000000   3.276800E+04   3.2768000E+04

00000500 08 SUBOUT - - - -ADMDGREG 88012 V2R2.0
GENERAL PURPOSE REGISTERS
GPR0     =  00000001  00AAEBFC  00362128  00000009
GPR4     =  00362128  00AAEE05  003819C4  00000080
GPR8     =  00373070  00379368  00AADE06  50AABE08
GPR12    =  00AACE07  000055B0  50AAEA1C  00AC82D0
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00000600 11 MODIN  - - - - - ADMDHUW 87240 V2R2.0
GENERAL PURPOSE REGISTERS
GPR0     =  00FFFFFF  00005788  00000002  00000003
GPR4     =  000000D8  00000000  000057F1  0000FFFF
GPR8     =  0000000C  00379368  00000000  50BF6DC0
GPR12    =  00BBB699  00005728  50BF7452  00B8C6C8
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00000700 06 MODOUT - - -ADMLAC1V 86104 V2R1.1
GENERAL PURPOSE REGISTERS
GPR0     =  00BBE307  003819B8  00000000  003819B8
GPR4     =  00000000  0037EA50  00372A58  00BBD697
GPR8     =  00BBE307  003795F0  00000018  60A4F9AA
GPR12    =  00BBB699  000053D0  60A4FA90  00A8A6E0
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
```

```
00000800 08 MODOUT - - - -ADMDOBI 86100 V2R1.0
GENERAL PURPOSE REGISTERS
GPR0    =   0000DB0C  00B13A58  003819C8  00000009
GPR4    =   00B136C0  00381548  003819C8  00000001
GPR8    =   00372D80  00379368  0037EA50  60B136E8
GPR12   =   00A5F057  00005458  40B139F4  00BA1E28
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00000900 21 SUBIN  - - - - - - - - - -ADMDHFR 86100 V2R1.0
GENERAL PURPOSE REGISTERS
GPR0    =   00362128  0037E478  00A5CE44  00000009
GPR4    =   00B0A7B0  00000000  00A59A7C  00000001
GPR8    =   003793E8  00379368  00347068  60B0A7D8
GPR12   =   00AA96C7  0037E420  50B0A7FE  00B0A310
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00001000 07 SUBIN  - - - ADMDHTE 86100 V2R1.0
GENERAL PURPOSE REGISTERS
GPR0    =   0000DB0C  00381A28  00A5CE44  00000009
GPR4    =   00B0AAD8  00381548  00381A28  00000001
GPR8    =   00A5CE6C  00379368  00000000  60B27F78
GPR12   =   00A5F057  000053D0  60B280C2  00B0AAD8
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0
00001100 06 MODOUT - - -ADMER00V 87315 V2R2.0
            CPNOUT - - -ESSFI  ('080C0840'X) - FREE INSTANCE STG
GENERAL PURPOSE REGISTERS
GPR0    =   0000DB0C  00381A28  00A5FCCC  00000001
GPR4    =   00A64D08  00381548  0C100000  00381A40
GPR8    =   00380D50  00379368  00373194  50B27330
GPR12   =   00A5F057  003819C8  50B27518  009E20C0
FLOATING POINT REGISTERS
FPR0    =  00000000  00000000              0               0
FPR2    =  00000000  00000000              0               0
FPR4    =  00000000  00000000              0               0
FPR6    =  00000000  00000000              0               0

00001251 END OF GDDM DIAGNOSTIC TRACE  02 AUG 1988 11:13 AM
```

Example 5. Tracing calls to the 5080 Graphics System workstation

The following statement specifies a maximum of 200 bytes per record to be displayed in the trace output.

```
[label] ADMMDFT TRCESTR='5080IO(ON,200)'
```

The trace output for this statement is:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN HPV20          05 SEP 1988 11:31 AM
 TRACE WORD = '00000000'X   BUILDID = 87315

*-----------------------------------------------------------------*
*                                                                 *
*  Trace defaults and nickname table omitted for clarity          *
*                                                                 *
*-----------------------------------------------------------------*

 DEFAULT TRCESTR='5080IO(ON,200)  '

  CALL TO GDDM/GRAPHIGS FOLLOWS..
 PTRACE INPUT PASS  GPOPPH ('0C000101'X)
 PTRACE    1 CHAR   'PHIGSERR'          D7C8C9C7 E2C5D9D9
 PTRACE    2 FIXED               0
  CALL TO GDDM/GRAPHIGS FOLLOWS..
 PTRACE INPUT PASS  GPEHND ('0C002C01'X)
 PTRACE    1 FIXED      10,868,920
 PTRACE OUTPUT PASS GPEHND ('0C002C01'X)
 PTRACE    1 FIXED  ---INPUT ONLY PARAMETER-----
  CALL TO GDDM/GRAPHIGS FOLLOWS..
 PTRACE INPUT PASS  GPOPWS ('0C000301'X)
 PTRACE    1 FIXED               1
 PTRACE    2 CHAR   'IBM5080 '          C9C2D4F5 F0F8F040
 PTRACE    3 CHAR   'GDDM5080'          C7C4C4D4 F5F0F8F0

*-----------------------------------------------------------------*
*                                                                 *
*  Trace records omitted for clarity                              *
*                                                                 *
*-----------------------------------------------------------------*

  CALL TO GDDM/GRAPHIGS FOLLOWS..
 PTRACE INPUT PASS  GPCLST ('0C001302'X)
 ADMDGP5 CALLED WITH GDF CODE=
 002C66D0    00000004                                   *....           *
 ADMDGP5 CALLED WITH GDF CODE=
 002C66D0    0000003E                                   *....           *

 ADMDGP5 CALLED WITH GDF CODE=
 002C66D0    00000033                                   *....           *
 ADMDGP5 CALLED WITH GDF CODE=
 002C66D0    000000C3                                   *...C           *
  CALL TO GDDM/GRAPHIGS FOLLOWS..
 PTRACE INPUT PASS  GPOPST ('0C001301'X)
 PTRACE    1 FIXED       2,898,616
  CALL TO GDDM/GRAPHIGS FOLLOWS..
 PTRACE INPUT PASS  GPAN2  ('0C00060B'X)
 PTRACE    1 FLOAT       1,934.437     2,072.062
 PTRACE    2 DIM                 4
 PTRACE    3 CHAR   'HHHH'              C8C8C8C8
  CALL TO GDDM/GRAPHIGS FOLLOWS..
```

```
PTRACE INPUT PASS  GPCLST ('0C001302'X)

ADMDGP5 CALLED WITH GDF CODE=
002C66D0    000000C1                                        *...A           *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0    000000C1                                        *...A           *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0    000000C1                                        *...A           *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0    000000C1                                        *...A           *

ADMDGP5 CALLED WITH GDF CODE=
002C66D0    00000071                                        *....           *
 CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS  GPOPST ('0C001301'X)
PTRACE    1 FIXED      2,898,616
 CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS  GPDPL2 ('0C000611'X)
PTRACE    1 DIM                5
PTRACE    2 FIXED              2
PTRACE    3 FLOAT              0            0        3,840           0
PTRACE    3 FLOAT          3,840    4,095.999            0   4,095.999
PTRACE    3 FLOAT              0            0
PTRACE    4 FIXED              2            2            2           2
PTRACE    4 FIXED              0
 CALL TO GDDM/GRAPHIGS FOLLOWS..
PTRACE INPUT PASS  GPCLST ('0C001302'X)

*--------------------------------------------------------------------*
*                                                                    *
*  Trace records omitted for clarity                                 *
*                                                                    *
*--------------------------------------------------------------------*

000004693 END OF GDDM DIAGNOSTIC TRACE  05 SEP 1988 11:31 AM
```

Example 6. Tracing in multiple instances of GDDM

The number in columns 2 through 4 shows in which instance of GDDM the event that caused the trace occurred.

```
1001  DIAGNOSTIC TRACE - ORIGIN JHERROD         04 AUG 1988 10:32 AM
 001
 001 TRACE WORD = '07F80003'X   BUILDID = 87315
 001
 001 00000001 01 CPNIN  FSINIT ('0C000001'X) - INITIALIZATION
 001             CPNOUT FSINIT ('0C000001'X) - INITIALIZATION
 001
 001 00000025 01 CPNIN  DSOPEN ('0C000200'X) - OPEN DEVICE
 001             CPNOUT DSOPEN ('0C000200'X) - OPEN DEVICE
 001
 001 00000237 01 CPNIN  DSUSE  ('0C000202'X) - DEVICE USAGE
 001             CPNOUT DSUSE  ('0C000202'X) - DEVICE USAGE
 001
 001 00000241 01 CPNIN  FSQURY ('0C040501'X) - EXTENDED QUERY DEVICE
 001             CPNOUT FSQURY ('0C040501'X) - EXTENDED QUERY DEVICE
 001
 001 00000259 01 CPNIN  WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
 001             CPNOUT WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
 001
 001 00000267 01 CPNIN  DSOPEN ('0C000200'X) - OPEN DEVICE
 001             CPNOUT DSOPEN ('0C000200'X) - OPEN DEVICE
 001
 001 00000381 01 CPNIN  DSUSE  ('0C000202'X) - DEVICE USAGE
 001             CPNOUT DSUSE  ('0C000202'X) - DEVICE USAGE
 001
 001 00000385 01 CPNIN  ASDFLD ('0C080700'X) - DEFINE ALPHA FIELD
 001             CPNOUT ASDFLD ('0C080700'X) - DEFINE ALPHA FIELD
 001
 001 00000475 01 CPNIN  ASFCOL ('0C080502'X) - SET FIELD COLOR
 001             CPNOUT ASFCOL ('0C080502'X) - SET FIELD COLOR
 *-------------------------------------------------------------------*
 *                                                                   *
 *  Trace records omitted for clarity                                *
 *                                                                   *
 *-------------------------------------------------------------------*
 001
 001 00002633 01 CPNIN  WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
 001             CPNOUT WSCRT  ('0C2C0000'X) - CREATE OPERATOR WINDOW
 001
1002  DIAGNOSTIC TRACE - ORIGIN JHERROD         04 AUG 1988 10:32 AM
 002
 002 TRACE WORD = '07F80003'X   BUILDID = 86139
 002
 002 00000001 01 CPNIN  SPINIT ('00050000'X) - SPI SPECIAL INIT
 002             CPNOUT SPINIT ('00050000'X) - SPI SPECIAL INIT
 002
 002 00000039 01 CPNIN  DSOPEN ('0C000200'X) - OPEN DEVICE
 002             CPNOUT DSOPEN ('0C000200'X) - OPEN DEVICE
 002
 002 00000149 01 CPNIN  DSUSE  ('0C000202'X) - DEVICE USAGE
 002             CPNOUT DSUSE  ('0C000202'X) - DEVICE USAGE
 002
 002 00000161 01 CPNIN  CSCCRT ('14040000'X) - CREATE CHART
 002             CPNOUT CSCCRT ('14040000'X) - CREATE CHART
 002
 001 00002671 01 CPNIN  WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
 001             CPNOUT WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
 001
 002
 002 00003897 01 CPNIN  FSTERM ('0C000000'X) - TERMINATION
 002             CPNOUT FSTERM ('0C000000'X) - TERMINATION
 002
 002 00004172 END OF GDDM DIAGNOSTIC TRACE  04 AUG 1988 10:33 AM
 001
 001 00002867 01 CPNIN  WSDEL  ('0C2C0100'X) - DELETE OPERATOR WINDOW
 001             CPNOUT WSDEL  ('0C2C0100'X) - DELETE OPERATOR WINDOW
 001
 001 00002879 01 CPNIN  WSSWP  ('0C2C0900'X) - SET WINDOW PRIORITIES
 001             CPNOUT WSSWP  ('0C2C0900'X) - SET WINDOW PRIORITIES
 001
 001 00002885 01 CPNIN  ASCPUT ('0C080603'X) - SET CHARACTER CODES
 001             CPNOUT ASCPUT ('0C080603'X) - SET CHARACTER CODES
 001
 001 00002895 01 CPNIN  WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
 001             CPNOUT WSIO   ('0C100008'X) - WINDOWED DEVICE I/O
 001
 001 00003301 01 CPNIN  ASREAD ('0C100000'X) - READ
 001             CPNOUT ASREAD ('0C100000'X) - READ
 001
```

```
001 00003319 01 CPNIN  FSTERM ('0C000000'X) - TERMINATION
001             CPNOUT FSTERM ('0C000000'X) - TERMINATION
001
001 00003892 END OF GDDM DIAGNOSTIC TRACE  04 AUG 1988 10:33 AM
```

Example 7. Trace output containing CDPDS input

If you specify CDPDS in your TRCESTR statements, the trace output contains fields in this form:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN BMASTERS         03 AUG 1988  3:20 PM

 TRACE WORD = '00000000'X   BUILDID = 87315

*--------------------------------------------------------------------*
*                                                                    *
*  Trace defaults and nickname table omitted for clarity             *
*                                                                    *
*--------------------------------------------------------------------*

 CDPDS structured field ...
 00362FF0    0058D3EE  EE000000  C3C4D7C4  E2409799   *..L.....CDPDS pr*
 00363000    9684A483  85844082  A8400000  00000000   *oduced by ......*
 00363010    00000000  40404040  40404040  D6D540F0   *....        ON 0*
 00363020    F961F1F1  61F8F640  40F1F57A  F2F44040   *9/11/86  15:24  *
 00363030    C2E840C2  D4C1E2E3  C5D9E240  E5D4F3F8   *BY BMASTERS VM38*
 00363040    F1F24040  40404040                       *12              *
 CDPDS structured field ...
 00362FF0    0062D3A8  A8000000  E2E2E240  40404040   *..Lyy...SSS     *
 00363000    00000601  01510100  08210200  80000000   *................*
 00363010    08210600  80000000  08210300  80004000   *.............. .*
 00363020    05234100  0005230A  00000523  06000005   *................*
 00363030    23070000  05230100  00052308  00000523   *................*
 00363040    02800005  230F0000  05230300  00051801   *................*
 00363050    0800                                     *..              *
 CDPDS structured field ...
 00362FF0    0010D3A8  AF000000  F1404040  40404040   *..Ly....1       *
 CDPDS structured field ...
 00362FF0    0010D3A8  C9000000  F0404040  40404040   *..LyI...0       *
 CDPDS structured field ...
 00362FF0    0033D3AB  8A000000  002B0C02  8500E3F1   *..L.........e.T1*
 00363000    C4F0C2C1  E2C50C02  8600C3F1  E2F0D7D9   *D0BASE..f.C1S0PR*
 00363010    F1F20424  05010D1F  0505005A  003C0000   *12.........!....*
 00363020    090003                                   *...             *
 CDPDS structured field ...
 00362FF0    0014D3A6  AF000000  00003840  3840002F   *..Lw....... . ..*
 00363000    D0003DE0                                 *}..\            *
 CDPDS structured field ...
 00362FF0    0012D3B1  AF000000  01090000  00000000   *..L.............*
 00363000    0000                                     *..              *
 CDPDS structured field ...
 00362FF0    001CD3A6  6B000000  03430108  4B000038   *..Lw,...........*
 00363000    40384009  4C02002F  D0003DE0             * . .<...}..\    *
 CDPDS structured field ...
 00362FF0    0020D3AC  6B000000  01170000  00000000   *..L.,...........*
 00363000    00002D00  00000000  00000000  002D0001   *................*
 CDPDS structured field ...
 00362FF0    0016D3A6  9B000000  00003840  3840002F   *..Lw....... . ..*
 00363000    D0003DE0  0000                           *}..\..          *
 CDPDS structured field ...
 00362FF0    0010D3A9  C9000000  F0404040  40404040   *..LzI...0       *
```

```
CDPDS structured field ...
00362FF0    0010D3A8  9B000000  F1404040  40404040    *..Ly....1       *
CDPDS structured field ...
00362FF0    0046D3EE  9B000000  2BD304D3  069004C7    *..L......L.L...G*
00363000    061804C5  007803F1  0106DBE3  8889A204    *...E...1...This.*
00363010    C9007804  DB89A204  C9007803  DB8104C9    *I....is.I....a.I*
00363020    007808DB  A2819497  938504C9  007806DB    *....sample.I....*
00363030    86899385  02F8                            *file.8          *
CDPDS structured field ...
00362FF0    0010D3A9  9B000000  F1404040  40404040    *..Lz....1       *
CDPDS structured field ...
00362FF0    0010D3A9  AF000000  F1404040  40404040    *..Lz....1       *
CDPDS structured field ...
00362FF0    0010D3A9  A8000000  E2E2E240  40404040    *..Lzy...SSS     *

00001075 END OF GDDM DIAGNOSTIC TRACE  03 AUG 1988  3:20 PM
```

Notes:

1. If the field header is invalid, an entry appears in the form:

   ```
   CDPDS field header invalid ...
   ```

   Only the first eight bytes of the structured field are included in the output.

2. If there are less than eight bytes in a record, an entry appears in the form:

   ```
   CDPDS short record ...
   ```

   The data in the short record is included in the output.

Example 8. Trace output containing HRIG output

If you specify HRIG in your TRCESTR statements, the trace output contains fields in this form:

```
1GDDM DIAGNOSTIC TRACE - ORIGIN BMASTERS         03 AUG 1988  2:36 PM

 TRACE WORD = '00000000'X   BUILDID = 87315

*--------------------------------------------------------------------*
*                                                                    *
*  Trace defaults and nickname table omitted for clarity             *
*                                                                    *
*--------------------------------------------------------------------*

 Begin Document - First Page
 TRANSMIT FIRST
  XMIT-----
 00363FE8                        5A0058D3  EEEE0000    *        !..L....*
 00363FF0    00D7D9C9  D5E340C7  C5D5C5D9  C1E3C5C4    *.PRINT GENERATED*
 00364000    40D6D540  F1F140E2  C5D740F1  F9F8F640    * ON 11 SEP 1986 *
 00364010    40F37AF3  F040D7D4  40C2E840  C2D4C1E2    * 3:30 PM BY BMAS*
 00364020    E3C5D9E2  40404040  40404040  40404040    *TERS            *
 00364030    40404040  40404040  40404040  40404040    *                *
 00364040    40                                        *                *
 TRANSMIT MIDDLE
  XMIT-----
 00363FE8                        5A0010D3  A8A80000    *        !..Lyy..*
 00363FF0    00C4D6C3  40404040  40                    *.DOC            *
 TRANSMIT MIDDLE
  XMIT-----
```

```
00363FE8                        5A0010D3  A8AF0000    *        !..Ly...*
00363FF0    00F14040  40404040  40                    *.1              *
TRANSMIT MIDDLE
 XMIT-----
00363FE8                        5A0010D3  A8C90000    *        !..LyI..*
00363FF0    00C1C5C7  40404040  40                    *.AEG            *
TRANSMIT MIDDLE
 XMIT-----
00363FE8                        5A0028D3  B18A0000    *        !..L....*
00363FF0    001C0000  00010000  00FFFFFF  FFFFFFFF    *................*
00364000    FFE3F1C4  F0C2C1E2  C5C3F1E2  F0D7D9F1    *.T1D0BASEC1S0PR1*
00364010    F2                                        *2               *
TRANSMIT MIDDLE
 XMIT-----
00363FE8                        5A0012D3  EE9B0000    *        !..L....*
00363FF0    002BD304  D1FFFF04  C0FFFF                *..L.J...{..     *
TRANSMIT MIDDLE
 XMIT-----
00363FE8                        5A0046D3  EE9B0000    *        !..L....*
00363FF0    002BD304  D3011804  C7010404  C5001403    *..L.L...G...E...*
00364000    F10106DB  E38889A2  04C90014  04DB89A2    *1...This.I....is*
00364010    04C90014  03DB8104  C9001408  DBA28194    *.I....a.I....sam*
00364020    97938504  C9001406  DB868993  8502F8      *ple.I....file.8 *
TRANSMIT MIDDLE
 XMIT-----
00363FE8                        5A0010D3  A99B0000    *        !..Lz...*
00363FF0    00E3E7E3  40404040  40                    *.TXT            *
End   Document
TRANSMIT LAST
 XMIT-----
00363FE8                        5A0010D3  A9A80000    *        !..Lzy..*
00363FF0    00C4D6C3  40404040  40                    *.DOC            *

00001075 END OF GDDM DIAGNOSTIC TRACE  03 AUG 1988  2:36 PM
```

Other tracing methods

Product-sensitive programming interface

This chapter has described how to use GDDM tracing by means of TRCESTR statements in external defaults files. You can also use TRCESTR statements for GDDM tracing in the following ways:

- Specify ADMMDFT TRCESTR statements in the source of an external defaults module. (How to create such a module is described in the *GDDM Base Programming Reference* manual.) This module is called:
  - ADMADFC on CICS
  - ADMADFI on IMS/VS
  - ADMADFT on TSO (and for the TSO Print Utility)
  - ADMADFV on VM/CMS
  - ADMADFD on VSE.
- In your application program, include calls to ESSUDS, specifying a source-format UDS for the ADMMDFT TRCESTR statements.

  The ESSUDS statement and source-format UDSs are described in the *GDDM Base Programming Reference* manual.

- In your application program, include calls to ESEUDS, specifying an encoded UDS for the ADMMDFT TRCESTR statements.

  The ESEUDS statement and encoded UDSs are described in the *GDDM Base Programming Reference* manual.

- In your application program, include calls to SPINIT, specifying in the systems programmer interface block (SPIB) an encoded UDS for the ADMMDFT TRCESTR statements.

  The SPINIT statement and the format of the SPIB are described in the *GDDM Base Programming Reference* manual.

Note: A single trace statement in an application call must conform to the grammar for a complete program. For example, all parts of an IF THEN ELSE statement must be within a single application call. However, a trace specification may be built up from any number of separate application calls; these will be appended to the trace program formed from an external file, if there is one.

There are several ways to turn on GDDM trace facilities that involve changing the value of the *trace control word* that is held internally by GDDM. The trace control word is a fullword integer that defines the kind of trace to be produced. If you do not specify it, its value is 0 (resulting in no trace activity). Other values for the trace control word are described under "The FSTRCE statement."

To change the trace control word for any GDDM utility or for an application program, do one of the following:

- Specify the ADMMDFT TRACE value in a source format defaults file. (The ADMMDFT macro is described in the *GDDM Base Programming Reference* manual.)

  Source format defaults files are normally available only on TSO or VM/CMS systems. Modified versions of this mechanism can also be used under CICS or VSE.

- Specify the ADMMDFT TRACE value in an external defaults module.

To change the trace control word from an application program:

- Include calls to FSTRCE, specifying the trace value as a parameter. This is described under "The FSTRCE statement."
- Include calls to ESSUDS, ESEUDS, or SPINIT, specifying a source-format UDS for the ADMMDFT TRACE default.

The FSTRCE statement

CALL FSTRCE(*control*)

The FSTRCE statement controls internal trace functions. It is intended for internal error diagnosis.

By default, trace is deactivated.

In coding the *control* parameter for FSTRCE, you must decide the level of trace you need to diagnose the problem:

Level 1 — Component
Level 2 — Subcomponent
Level 3 — Module.

Each lower trace level includes tracing of higher levels. For example, level 3 tracing (module tracing) also includes levels 2 and 1 (subcomponent and component tracing).

Note: Depending on the complexity of the program being diagnosed, you may get excessive quantities of trace output generated if Level 3 trace is used with all components enabled. It is therefore recommended that you initially use Level 1 trace to locate the component in error and then use Level 3 trace with all other components disabled to enable you to trace the specific module in error.

Convention: Throughout this book, the first byte or bit of a sequence is designated as byte or bit 0.

Parameter

control (fullword integer)

A fullword integer, the trace control word, whose contents determine the type of tracing and the components to be traced, and has the following format:

Bytes 0 and 1 — Flags

Bit 0 Set to 1 to make the trace word unchangeable. If this bit is set, any following calls to FSTRCE are ignored. This bit can be used to force a setting of trace (typically by using the defaults file), regardless of any FSTRCE calls in an application. This may be of use if an application already contains FSTRCE calls, and if the source of the application is not readily available.

Bits 1-15 Component disablement flags.

Set to 1 to *disable* tracing for component **n** according to the following table:

n	**Component**
1	Application Interface
2	Terminal services interface
3	Environmental services interface
4	Full screen manager
5	Presentation Graphics routines (GDDM-PGF)
6	Interactive Chart Utility (GDDM-PGF)
7	Image Symbol Editor (GDDM)
8	Common services interface
9	Vector Symbol Editor (GDDM-PGF)
10	Output print utility
11	IMS/VS Interactive Utility scheduler

12	Numerical processing routines (GDDM-PGF)
13	GDDM Interactive Map Definition (GDDM-IMD)
14	Reserved
15	Image manager.

Byte 2 — Trace qualifier

Can be set to produce abend dumps or additional trace output.

Unless otherwise stated, the trace qualifiers operate regardless of the component disablement flag settings and trace levels. The trace qualifiers in hex are:

00 No dumps:
produce trace as controlled by other flags.

02 Abend (only if the terminal services interface component trace is not disabled):
causes abnormal termination of GDDM (abend code 1405), within the module ADMLQU1x, immediately after the terminal characteristics have been determined.

03 Abend at termination:
causes abnormal termination of GDDM (abend code 1051), within the module ADMACLP, immediately before starting normal termination.

10 Partial transmission trace:
causes the first 16 bytes of terminal transmissions to be traced at the following points in execution:

For CICS, IMS/VS, and TSO, immediately before outgoing terminal transmissions and after incoming terminal transmissions.

For VM/CMS and VMXA, immediately after both incoming and outgoing terminal transmissions.

11 Full transmission trace:
as for the partial transmission trace, but including the complete terminal transmissions.

12 Full transmission trace with control blocks:
as for the full transmission trace, but including the applicable GDDM terminal control areas.

20 Storage use report:
causes a report of how GDDM storage is acquired and freed throughout processing, to be generated at termination (FSTERM), or immediately before a GDDM-generated abend.

The report lists, for each block of storage used by GDDM, its length and address, the name of the module requesting the storage and the associated event sequence number, and the name of the module releasing the storage and associated event sequence number.

The event sequence numbers tally with the sequence numbers generated in any accompanying module trace.

The report does not include the following items of storage:

- GDDM initial storage allocation
- GDDM dynamic save area stacks
- Storage acquired or freed during trace processing
- Storage acquired or freed while trace is not active.

Byte 3 – Trace level

Possible settings, in hex, are:

00	Stop component, subcomponent, and module trace
01	Start trace of component entry and exit
02	Start component trace plus subcomponent entry and exit
03	Start component and subcomponent trace plus module entry and exit.

Abend dump output

If 02 or 03 is specified as the trace qualifier, dumps are produced by abnormal termination of GDDM.

On abnormal termination of GDDM, the subsystem normally produces diagnostic messages incorporating the abend code. An exception is TSO. After receiving the message PROGRAM TERMINATED DUE TO ERROR+ you should enter a question mark (?). TSO produces a second-level message incorporating the abend code, if applicable. If you then press ENTER, without any intervening input, TSO produces a dump, if a SYSABEND or SYSUDUMP file has been previously allocated.

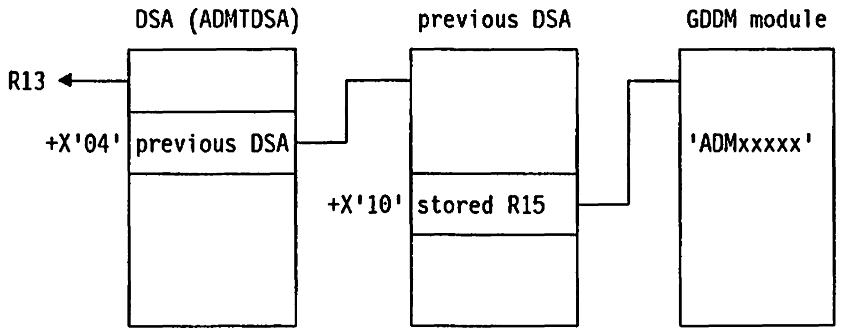
End of Product-sensitive programming interface

Locating control blocks and the in-storage trace table

The following sections define a procedure for locating the primary GDDM control blocks, including the in-storage trace table, from a dump. This procedure can be used to locate:

- The current active GDDM modules in the absence of trace output
- The current GDDM control blocks, including the in-storage trace table.

Locating the current GDDM module



1. Check whether the abend code is listed in Appendix B, "Abend codes" on page 155. If it is listed, the information provided in that appendix will help you locate the current GDDM module.

   If the abend code is not listed, continue with the next step in this procedure.

2. Locate the current value of register 13. Usually, abend dumps contain a list of REGISTERS ON ENTRY TO ABEND.

3. Locate the save area (ADMTDSA) using the address given in register 13.

   Under VSE, register 4 is loaded with the address of the save area before an abend dump.

4. Locate the preceding save area using the address given at offset X'4' in the current save area.

5. Locate the executable code that has stored its registers on entry in this save area, using the address given at offset X'10' (16 decimal) in this save area (stored register 15).

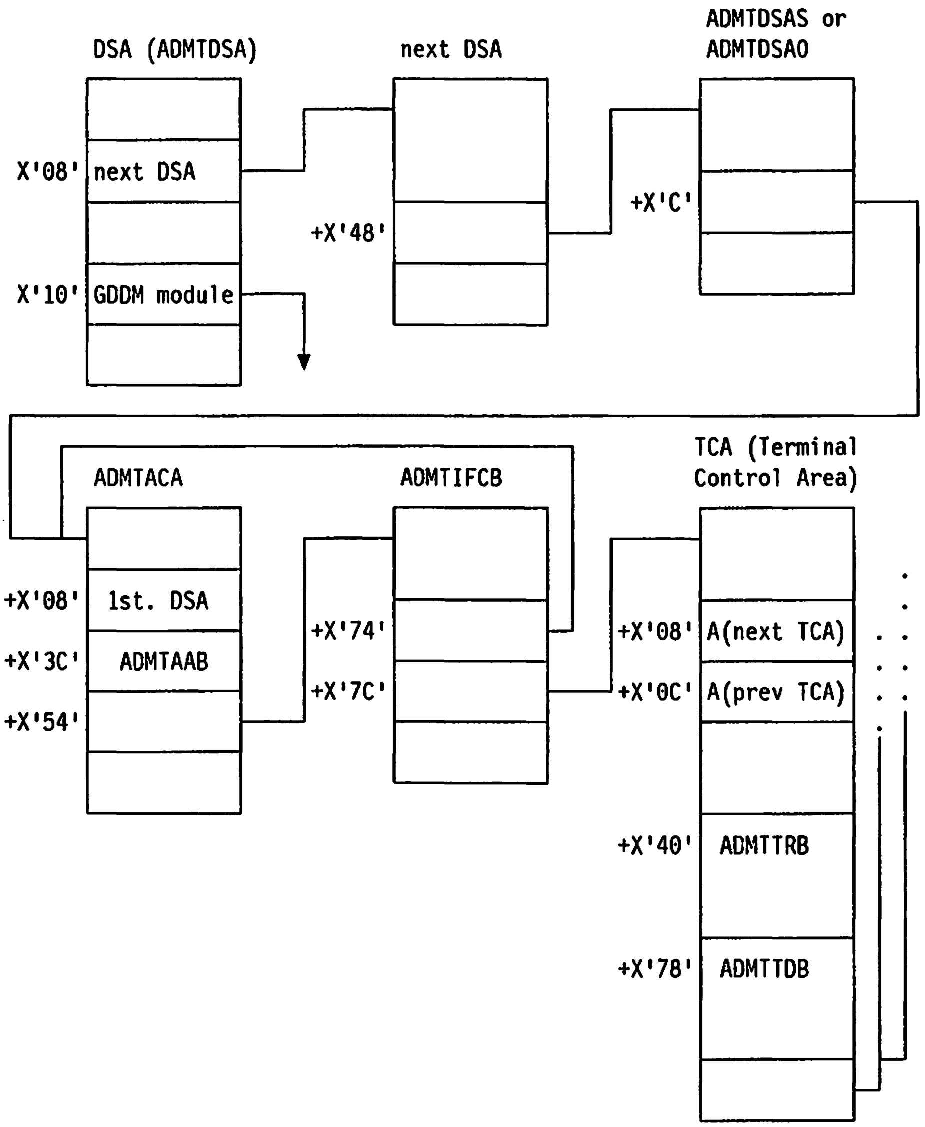
6. Determine if this executable code is a GDDM module by looking for the EBCDIC representation of the module name shortly following the start of the executable code. If it is a GDDM module, it starts with the letters ADM or AEM.

7. If the EBCDIC representation of the module name is not present or is not a GDDM module, return to step 3. If step 3 is no longer valid, go to "Locating the in-storage trace table if the save area chain is corrupted" on page 83 for another way to locate GDDM control blocks.

8. The located module should be the current GDDM module. Repeat steps 3 through 5 to get the names of the other GDDM modules that were currently active at the time of the dump.

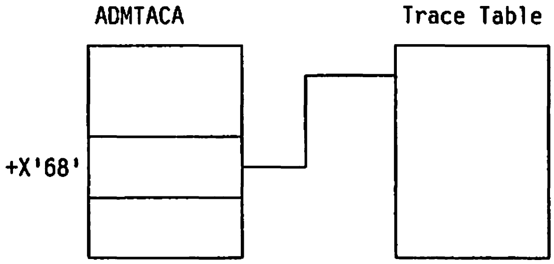
Locating the GDDM control blocks from the current module

1. From a save area used to locate a currently active GDDM module, go to the *next* save area using the address given at offset X'8' in the current save area.
2. Locate the dynamic save area stack header (ADMTDSAS or ADMTDSAO), using the address given at offset X'48' (72 decimal) in this next save area.
3. Locate the application interface component control area (ADMTACA) using the address given at offset X'0C' (12 decimal) in the stack header.
4. Locate the GDDM interface control block (ADMTIFCB) using the address given at offset X'54' (84 decimal) in the AIC control area stack header. Under VSE, register 2 is loaded with the address of ADMTIFCB before an abend dump.
5. Locate a chain of terminal control areas, each including a terminal request block (ADMTTRB) and a terminal descriptor block (ADMTTDB). The address at offset X'5C' (92 decimal) in the interface control block (ADMTIFCB) gives the start of the chain of terminal control areas; each contains a terminal request block at offset X'40' (64 decimal) and a terminal descriptor block at offset X'78' (120 decimal).



Locating the in-storage trace table

Locate the in-storage trace table using the address given in field ACATRCTB at offset X'68' (104 decimal) in the ADMTACA. Note that this field is zero and no trace table is present if trace has not been enabled, or if there is not enough storage to allocate an in-storage trace table.



Locating the in-storage trace table if the save area chain is corrupted

1. If it is not possible to locate the save area chains using the mechanisms described above, scan the dump for a data area containing the characters IFCT.

   This data area is the GDDM interface control table, and is part of the interface control block (ADMTIFCB). The characters IFCT are actually at offset X'4' in the IFCB.

2. Locate the application interface component control area (ADMTACA), using the address given at offset X'74' (116 decimal) in the IFCB.

3. Locate the initial GDDM save area, using the address given at offset X'8' in the ADMTACA.

   Verify the interface control block (IFCB) and locate the application anchor block (AAB), using the addresses given at offsets X'54' (84 decimal) and X'3C' (60 decimal), respectively, in the ADMTACA.

4. From the initial GDDM save area, locate lower-level GDDM save areas using the forward chains at offset X'8' in these save areas. Note that, using this procedure, you can chain forward into save areas that are no longer active.

5. For each of these save areas, locate the executable code that has stored its registers on entry in the save area, using the address given at offset X'10' (16 decimal) in the save area (stored register 15).

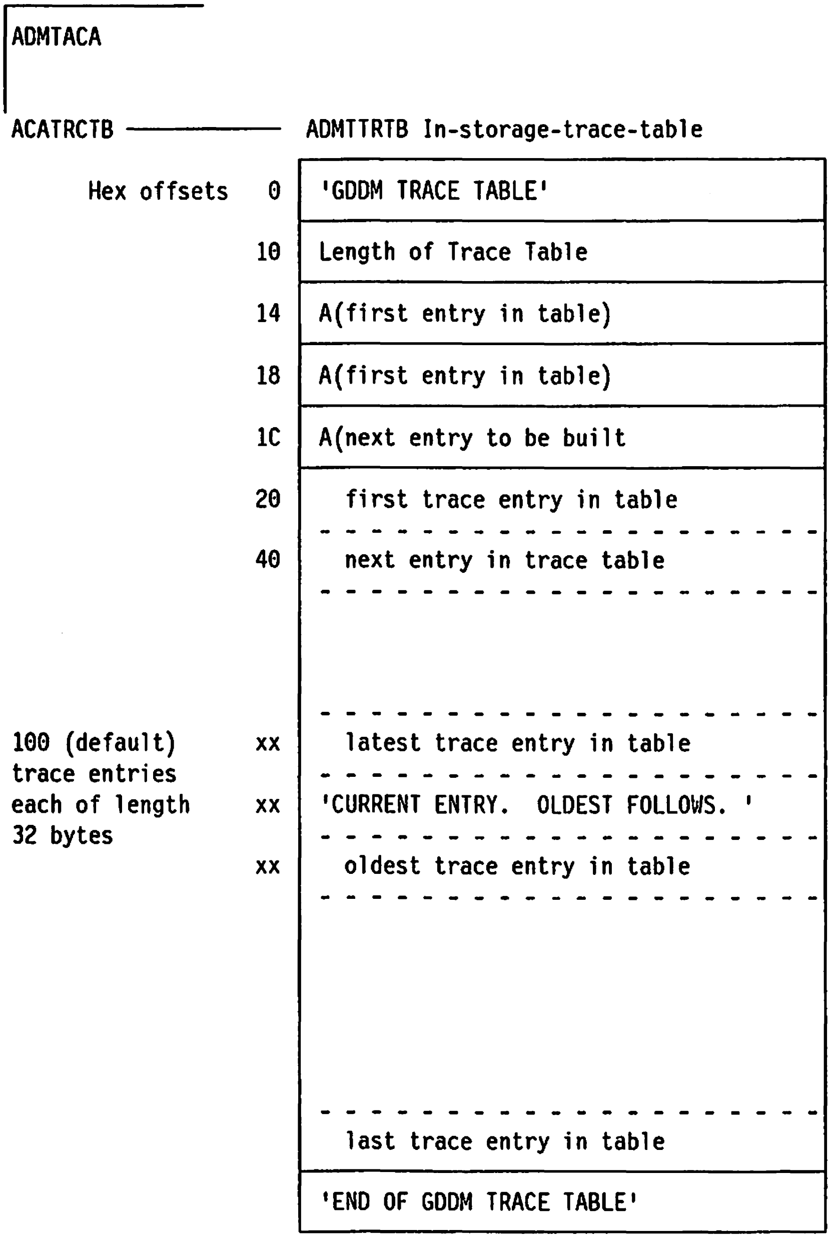
6. Determine if this executable code is a GDDM module by looking for the EBCDIC representation of the module name shortly following the start of the executable code. If it is a GDDM module, it starts with the letters ADM or AEM.

7. Locate the in-storage trace table, using the address given in field ACATRCTB at offset X'68' (104 decimal) in the ADMTACA. Note that this field is zero and no trace table is present if trace has not been enabled, or if there is not enough storage to allocate an in-storage trace table.

Format of in-storage trace table

In addition to being written to a trace data set, trace records are also written in an abbreviated format, into a cyclic in-storage trace table, the format of which is shown in the next diagram.

By default, the cyclic in-storage trace table contains room for the last 100 trace records. This default is defined by the ADMMDFT TRTABLE value in the external defaults module, and can be increased if required, up to a maximum of 1000. The in-storage trace table can be located as described on page 82.



```
ADMTACA

ACATRCTB ─────────── ADMTTRTB In-storage-trace-table

        Hex offsets   0  'GDDM TRACE TABLE'
                     10  Length of Trace Table
                     14  A(first entry in table)
                     18  A(first entry in table)
                     1C  A(next entry to be built
                     20    first trace entry in table
                     40    next entry in trace table

100 (default)        xx    latest trace entry in table
trace entries
each of length       xx  'CURRENT ENTRY.  OLDEST FOLLOWS. '
32 bytes
                     xx    oldest trace entry in table

                           last trace entry in table
                         'END OF GDDM TRACE TABLE'
```

Meaning of in-storage trace records

Each in-storage trace record is 32 bytes in size, with two possible formats, according to the trace record type. The trace record type is identified by bytes 6 and 7 of the trace record.

Format for CPNIN, CPNOUT

Bytes 00-03 Event Sequence Number (binary)

Bytes 04-05 DSA Level (EBCDIC)

Bytes 06-07 Trace Record Type (EBCDIC)

'10' CPNIN (component entry)

'11' CPNOUT (component exit)

Byte 08 Blank

Bytes 09-14 RCP (request control parameter) mnemonic (EBCDIC)

Byte 15 Blank

Bytes 16-19 RCP (request control parameter) code (binary)

Bytes 20-28 Error message identifier and severity (EBCDIC)

Bytes 29-31 Error message type (EBCDIC)

'(I)' Internal error code

'(E)' External error code

The meaning of each field is as described on page "Format of trace output" on page 59.

The following is an example of a CPNOUT trace record, in EBCDIC:

```
'xxxx0111 FSPCRT yyyyADM0130 E(E)'
```

which shows:

Bytes	Contents	Meaning
00-03	xxxx	Event sequence number xxxx (in binary)
04-05	'01'	DSA level 1
06-07	'11'	CPNOUT (component exit)
09-14	FSPCRT	Function is FSPCRT
16-19	yyyy	RCP code (in binary)
20-26	ADM0130	Error feedback contains error code 130
28	E	Error feedback is severity 'E'
29-31	(E)	Error feedback represents an external error.

Format for SUBIN, SUBOUT, MODIN, MODOUT

Bytes 00-03 Event sequence number (binary)

Bytes 04-05 DSA level (EBCDIC)

Bytes 06-07 Trace record type (EBCDIC)

- '20' SUBIN (subcomponent entry)
- '21' SUBOUT (subcomponent exit)
- '30' MODIN (module entry)
- '31' MODOUT (module exit)

Bytes 08-31 Module identification, including module name.

The meaning of each field is as described on page "Format of trace output" on page 59.

The following is an example of a MODOUT trace record, in EBCDIC:

```
'xxxx0131ADMDSRO 84223 V1R4.0   '
```

which shows:

Bytes	Contents	Meaning
00-03	xxxx	Event sequence number xxxx (in binary)
04-05	'01'	DSA level 1
06-07	'31'	MODOUT (module exit)
08-31	......	Module identification, including module name, ADMDSRO

GDDM-CSPF tracing

The trace facilities described in other parts of this chapter apply to GDDM-CSPF where it calls other GDDM functions.

For its own functions, GDDM-CSPF has a trace facility that is started as follows:

On MVS:

Start the GDDM-CSPF background task with the command:

```
S EAKBACK,PARM1=TRACE
```

The output trace is sent to the data set with ddname EAKTRACE.

On VM:

Start the GDDM-CSPF program with the command:

```
EAKVINIT filename TRACE [BY dest]
```

The output trace is sent to FILE EAKTRACE A.

Chapter 4. GDDM Interactive Map Definition diagnosis

This chapter describes an additional diagnostics facility that is supplied to help solve problems in GDDM Interactive Map Definition (GDDM-IMD). It is primarily intended for use under the direction of IBM Support Center service representatives.

Using the GDDM Interactive Map Definition diagnostics facility

You can use the GDDM-IMD Diagnostics facility for two purposes:

- To display the contents of main storage
- To trace and check operations performed on the map specification library (MSL).

You can display the contents of main storage at any time during GDDM-IMD operation.

You can turn the MSL trace on and off at any time during GDDM-IMD operation. GDDM-IMD automatically displays the applicable information whenever the MSL is accessed. If GDDM trace is active, the information is also sent to the GDDM trace file.

When you report a failure in GDDM-IMD, the IBM Support Center may ask you to repeat the operation that failed, with the MSL traces or record validation switched on. You should direct the output this generates to the GDDM trace file by turning on the GDDM trace facility.

If you are asked to submit an APAR, include any messages generated by record validation in the ERR-DESCRIPTION field in the APAR data set.

Invoking the GDDM Interactive Map Definition diagnostics facility

To invoke the GDDM-IMD Diagnostics facility:

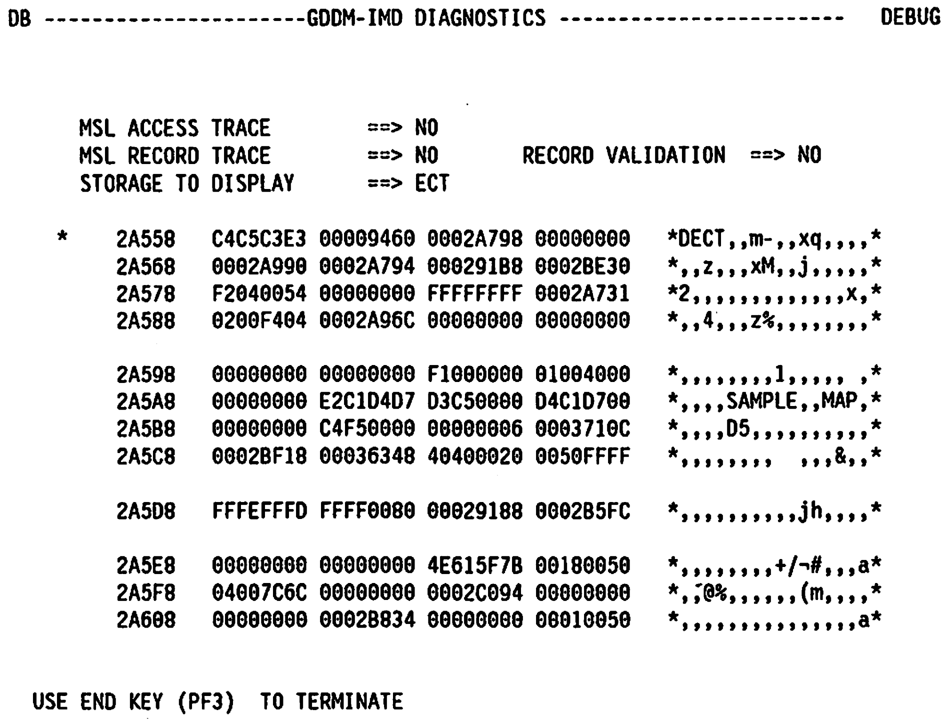
1. Press the HELP key (PF1) to enter the tutorial.
2. Enter the DEBUG command (abbreviation: D) on the command line of the tutorial frame. The GDDM-IMD Diagnostics frame is then displayed.

   Note: You can use the DEBUG command even if the tutorial is not available.

The GDDM Interactive Map Definition diagnostics frame

The example below shows the GDDM-IMD diagnostics frame. When this frame is displayed, you can select the functions of the GDDM-IMD diagnostics facility that you require. You can:

- View the contents of main storage associated with the operation you currently have running in GDDM-IMD. (This facility may be restricted to specified users in your installation.)
- Turn on and off the MSL access trace, the MSL record trace, and record validation.



```
DB ---------------------GDDM-IMD DIAGNOSTICS -----------------------   DEBUG



      MSL ACCESS TRACE       ==> NO
      MSL RECORD TRACE       ==> NO      RECORD VALIDATION  ==> NO
      STORAGE TO DISPLAY     ==> ECT

    *    2A558   C4C5C3E3 00009460 0002A798 00000000   *DECT,,m-,,xq,,,,*
         2A568   0002A990 0002A794 000291B8 0002BE30   *,,z,,,xM,,j,,,,,*
         2A578   F2040054 00000000 FFFFFFFF 0002A731   *2,,,,,,,,,,,,,x,*
         2A588   0200F404 0002A96C 00000000 00000000   *,,4,,,z%,,,,,,,,*

         2A598   00000000 00000000 F1000000 01004000   *,,,,,,,,1,,,,, ,*
         2A5A8   00000000 E2C1D4D7 D3C50000 D4C1D700   *,,,,SAMPLE,,MAP,*
         2A5B8   00000000 C4F50000 00000006 0003710C   *,,,,D5,,,,,,,,,,*
         2A5C8   0002BF18 00036348 40400020 0050FFFF   *,,,,,,,,  ,,,&,,*

         2A5D8   FFFEFFFD FFFF0080 00029188 0002B5FC   *,,,,,,,,,,jh,,,,*

         2A5E8   00000000 00000000 4E615F7B 00180050   *,,,,,,,,+/¬#,,,a*
         2A5F8   04007C6C 00000000 0002C094 00000000   *,;@%,,,,,,(m,,,,*
         2A608   00000000 0002B834 00000000 00010050   *,,,,,,,,,,,,,,,a*


   USE END KEY (PF3)  TO TERMINATE
```

Main storage display

This facility is available only if one of the 31 low-order bits of the GDDM trace control word is set to 1. See Chapter 3, "GDDM tracing" on page 37 for details of setting the GDDM trace control word.

The first column shown in each line of the storage display contains a hexadecimal address; the remainder of the display shows the contents of the 16 bytes of storage starting at that address, in hexadecimal and in characters. Characters that are not displayed are shown as periods (.). The storage address is unprotected and may be updated by typing over it so that a specified area of storage is displayed.

You can set the starting address of the storage to be displayed in the STORAGE TO DISPLAY field at the top of the frame. You specify the address as a hexadecimal expression, which may contain the following operators and special values:

+ plus
− minus
% designates that the expression to the left of the percent symbol is an address containing the fullword value to be used. (See the examples below.)
* designates the address of the storage in the current display.

Examples

```
1814%+8%
```

means "Take the fullword at address X'1814' and add 8 to it." The result is in turn an address containing a fullword. This fullword is the address of the storage to be displayed.

```
*-C8
```

means "Display the storage starting 200 bytes before the start of the currently displayed storage."

You can use PF8 and PF7 (or PF20 and PF19) to scroll the current storage display forward and backward by its current length. If you place the cursor under any of the hexadecimal fields in the storage display and press PF9 or PF21, the contents of the field are used as the starting address of a new area of storage to be displayed.

To resume normal GDDM-IMD operation, press PF3.

Note: If you try to display storage to which GDDM-IMD does not have read access, an abnormal termination occurs in module AEMDIA81 for the unacceptable address.

Tracing MSL operations

The GDDM-IMD diagnostics frame contains three input fields to control how the map specification library (MSL) is traced:

- MSL ACCESS TRACE controls the display of MSL access requests (open, get, put, close, and so on)
- MSL RECORD TRACE controls the display of MSL records
- RECORD VALIDATION controls the validation of records.

In each case, enter YES (or Y) to turn the facility on, or NO (or N) to turn it off. The initial value is NO.

Press PF3 to resume normal GDDM-IMD operation.

Using the MSL traces

If MSL traces are turned on, the GDDM-IMD screen is cleared every time there is trace data to show, and the trace data is displayed. When the screen is full, or there is no more trace data to be displayed, the following prompt appears:

```
AEM00201A PRESS ENTER TO CONTINUE, OR CANCEL
```

Pressing any interrupt key causes the next frame of trace data, or the next GDDM-IMD frame, to appear. Pressing the cancel key (PF6/18) in response to the prompt, causes trace data generated before the next GDDM-IMD frame not to be displayed, but the trace data is still directed to the GDDM trace data set if the GDDM trace facility is active.

The MSL access trace

The MSL access trace displays invocations of the MSL access instructions with the options that they specify. Here is a typical access trace:

```
HEDDBOPN  SAMPLE__ ________
  MSL=00/01 OBJ=02/02 LVL=03/03 ACC=UPD STA=A NXT=N CPY=D DIR=000F7FFC RC=0001
HEDDBPUT  SAMPLE__ ________
  MSL=01 REC=F304 AT=0004DD08 1ST=0001 LST=0001 BRK=N RTN=Y MOF=Y
  DIR=000F7FFC RC=0000
HEDDBGET  SAMPLE__ ________
  MSL=01 REC=F304 AT=000F980C 1ST=0001 LST=0001
  DIR=000F7FFC RC=0000
HEDDBOPN  SAMPLE__ MAP_____
  MSL=00/01 OBJ=01/01 LVL=03/03 ACC=INP STA=O NXT=G CPY=D DIR=000FB4FC RC=0000
HEDDBGET  SAMPLE__ MAP_____
  MSL=01 REC=F304 AT=000FC50C 1ST=0001 LST=FFFF
  DIR=000FB4FC RC=0000
HEDDBFRE  SAMPLE__ MAP_____
  MSL=01 REC=F304 DIR=000FB4FC RC=0000
HEDDBCLS  SAMPLE__ MAP_____
  MSL=01 RC=0000
HEDDBOPN  SAMPLE__ MAP_____
  MSL=00/00 OBJ=01/00 LVL=03/00 ACC=INP STA=O NXT=G CPY=D DIR=00000000 RC=0202

AEM00201A PRESS ENTER TO CONTINUE, OR CANCEL
```

The MSL record trace

The MSL record trace displays the MSL data records as they are read from the MSL and written to the MSL. Here is an example of the output produced by the MSL record trace:

```
HEDDBGET  SAMPLE__ MAP_____
  MSL=01 REC=F304 AT=000C01CC 1ST=0001 LST=FFFF
  00260000 000C01F4
             0008    01000000 00180050 00010001 00000000  *.......&........*
             0018    00004040 40404040 40408300 0000      *..        c...  *
  00140000 000C0208
             0008    02007B5F 6C614E7C 00010001           *..#¬%/+@....    *
  00330001 00000000
             0008    03000003 C1000000 000000E0 00000000  *....A...........*
             0018    00D70000 00000000 60000000 0000E400  *.P......-.....U.*
             0028    00000000 00200000 000000             *...........     *
  DIR=0009427C RC=0000
HEDDBPUT  SAMPLE__ MAP_____
  MSL=01 REC=F304 AT=000C01CC 1ST=0001 LST=0004 BRK=N RTN=Y MOF=Y
  00260000 000C01F4
               0008    01000000 00180050 00010001 00000000   *.......&........*
               0018    00004040 40404040 40408300 0000       *..        c...  *
  00140000 000C0208
               0008    02007B5F 6C614E7C 00010001            *..#¬%/+@....    *
  00330801 000C6790 S
               0008    03000003 C1000000 000000E0 00000000   *....A...........*
               0018    00D70000 00000000 60000000 0000E400   *.P......-.....U.*
               0028    00000000 00200000 000000              *...........     *
  00400501 00000000 RF
               0008    03000004 C1000000 000000E0 00000000   *....A...........*
               0018    00C20000 00000000 70000000 0000D700   *.B............P.*
               0028    00000000 00600000 000000E4 00000000   *.....-.....U....*
               0038    00002000 00000000                     *........        *
  DIR=0009427C RC=0000

AEM00201A PRESS ENTER TO CONTINUE, OR CANCEL
```

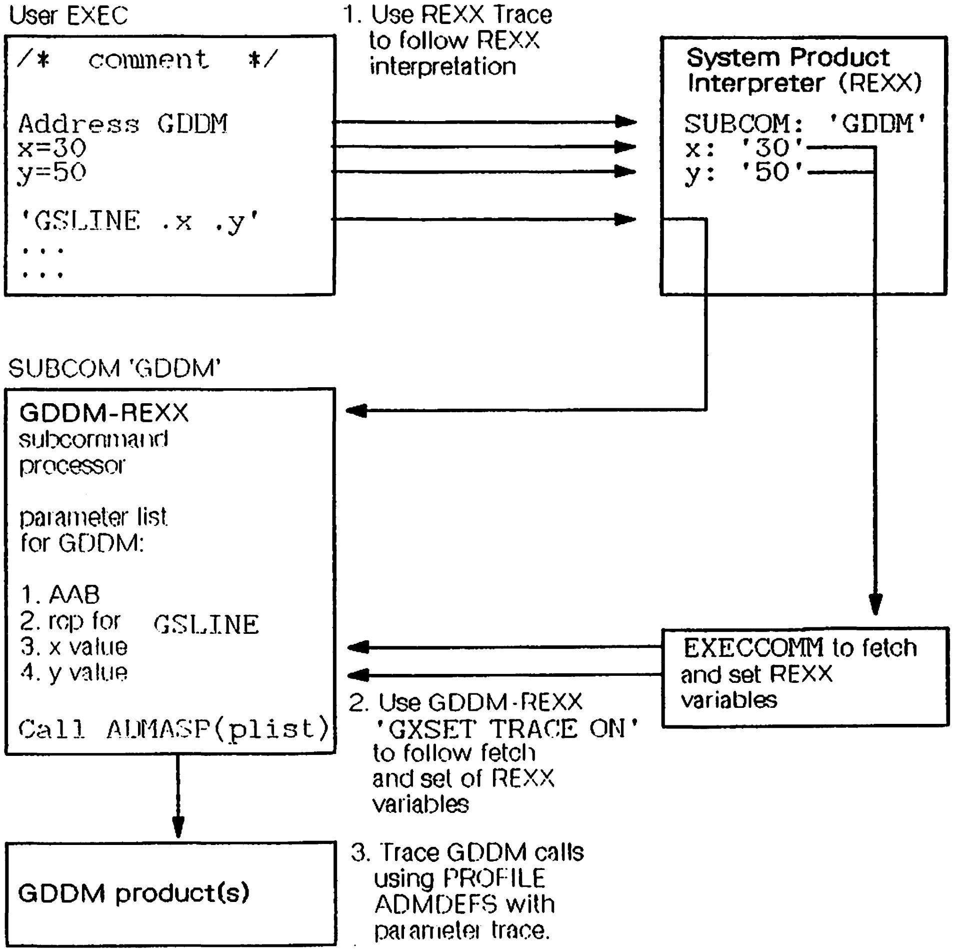
Record validation

If record validation is turned on, the diagnostics facility checks the records sent to and received from the MSL. If any check fails, trace messages are generated. If the MSL record trace is also turned on, such messages will also be displayed *before* the records are traced. The following table lists the messages that the record validation may produce:

Error message	Reason message generated
E01 INVALID OPCODE, DDBLROPC=xx	The MSL operation code is not valid
E02 LEAVES A GAP, DPRSLINE=xxxx, DPRSCOL=xxxx	Record does not follow immediately after previous record
E03 NEW LINE NOT COLUMN 1, DPRSCOL=xxxx	New line does not start in column one
E04 NON-ZERO FLID FOR CONSTANT FIELD	A constant field appears to be linked to an application structure
E05 ATTRIBUTE-BITS INVALID	The attribute field (DPRSAPSK) contains an invalid bit value
E06 JUSTIFY-BITS INVALID	Both justification bits are set
E07 INTENSITY-BITS INVALID	The intensity attribute bit setting is not valid
I02 LEADING/TRAILING BLANKS IN TEXT	There are leading or trailing blanks in DPRSTEXT
S01 MISALIGNED, ADDR=xxxxxxxx	The record must be on a fullword boundary
S02 OVERLAPS PRECEDING FIELD, DPRSLINE=xxxx, DPRSCOL=xxxx	Record overlaps end of previous record
S03 LINE LENGTHS INCONSISTENT, PREVIOUS=xxxx, MAX=xxxx	Length of the previous line of the map is not the same as the maximum found
S04 TEXT OVERFLOWS FIELD, DPRSTXTL=xx, DPRSTXTO=xx, DPRSTLEN=xx	Text offset and length are longer than field length
S05 INCONSISTENT LENGTHS, DDBLRLT=xxxx, DPRSTXTL=xx, DPRSXLEN=xx	The length in the logical record header is not consistent with the lengths of the component parts of the record
S06 TOO SHORT, DDBLRLT=xxxx	The record is shorter than the maximum
W03 CURSOR FLAG ON AGAIN	The cursor is set in more than one record
W04 RESERVED FIELD NOT NULL, DPRSTYPS=xx	A reserved bit in DPRSTYPS is not cleared to zero

Chapter 5. GDDM-REXX diagnosis

GDDM-REXX is a program that runs in the subcommand environment of REXX. Any command that is not recognized by REXX is passed to the active subcommand environment. To make GDDM-REXX the active subcommand environment, the **Address gddm** instruction is used. Then any symbolic parameters passed are resolved by GDDM-REXX. GDDM-REXX then passes calls to GDDM.


User EXEC
/* comment */
Address GDDM
x=30
y=50
'GSLINE .x .y'
. . .
. . .
1. Use REXX Trace to follow REXX interpretation
System Product Interpreter (REXX)
SUBCOM: 'GDDM'
x: '30'
y: '50'
SUBCOM 'GDDM'
GDDM-REXX subcommand processor
parameter list for GDDM:
1. AAB
2. rcp for GSLINE
3. x value
4. y value
Call ADMASP(plist)
EXECCOMM to fetch and set REXX variables
2. Use GDDM-REXX 'GXSET TRACE ON' to follow fetch and set of REXX variables
GDDM product(s)
3. Trace GDDM calls using PROFILE ADMDEFS with parameter trace.

When you are trying to find the source of an error, there are three different types of tracing that you can use: REXX, GDDM-REXX, and GDDM. The figure above shows how they relate to one another.

You may need to consult the *VM/System Product Interpreter Reference*, *VM/XA System Product Interpreter Reference*, or the *VM/SP System Programmer's Guide* for further information about subcommand concepts, the REXX Address instruction, and REXX tracing.

Here is a simple REXX EXEC and examples of trace output for it:

```
Trace r                             /* Start REXX tracing           */
Address command 'GDDMREXX INIT'
Address gddm
s='abcde'
x=70
y=60
'GXSET TRACE ON TIME'               /* Start GDDM-REXX tracing      */
'GSCHAR 50 50 5 .s'
'GSMOVE 40 60'
'GSLINE .x .y'
'GXSET TRACE OFF'                   /* Stop GDDM-REXX tracing       */
'ASREAD . . .'
Address command 'GDDMREXX TERM'
Trace off                           /* Stop REXX tracing            */
Exit
```

The REXX and GDDM-REXX traces were spooled to the console using the CP command SPOOL CONSOLE START. Here is part of the output:

```
      4 *-* Address command 'GDDMREXX INIT'
        >>>   "GDDMREXX INIT"
      5 *-* Address gddm
      6 *-* s='abcde'
        >>>   "abcde"
      7 *-* x=70
        >>>   "70"
      8 *-* y=60
        >>>   "60"
      9 *-* 'GXSET TRACE ON TIME'            /* Start GDDM-REXX tracing
 */
        >>>   "GXSET TRACE ON TIME"
 ERX0000 I TIME STAMP: 08/24/88 11:15:30.119222
 ERX0000 I "GXSET TRACE ON TIME"
     10 *-* 'GSCHAR 50 50 5 .s'
        >>>   "GSCHAR 50 50 5 .s"
 ERX0000 I Var fetch: s = "abcde"
 ERX0000 I TIME STAMP: 08/24/88 11:15:33.211771
 ERX0000 I "GSCHAR 50 50 5 .s"
     11 *-* 'GSMOVE 40 60'
        >>>   "GSMOVE 40 60"
 ERX0000 I TIME STAMP: 08/24/88 11:15:33.271971
 ERX0000 I "GSMOVE 40 60"
     12 *-* 'GSLINE .x .y'
        >>>   "GSLINE .x .y"
 ERX0000 I Var fetch: x = "70"
 ERX0000 I Var fetch: y = "60"
 ERX0000 I TIME STAMP: 08/24/88 11:15:33.284259
 ERX0000 I "GSLINE .x .y"

1    13 *-* 'GXSET TRACE OFF'                /* Stop GDDM-REXX tracing
 */
        >>>   "GXSET TRACE OFF"
     14 *-* 'ASREAD . . .'
        >>>   "ASREAD . . ."
     15 *-* Address command 'GDDMREXX TERM'
        >>>   "GDDMREXX TERM"
     16 *-* Trace off                        /* Stop REXX tracing
```

GDDM tracing was specified by a PROFILE ADMDEFS file containing this entry (note the space at the start):

```
 DEFAULT TRCESTR='IF API THEN PARMSF TIME'
```

The GDDM trace went to a file called ADM00001 ADMTRACE A1, part of which is shown below.

```
1GDDM DIAGNOSTIC TRACE - ORIGIN PATTERSN         24 AUG 1988 11:15 AM

 TRACE WORD = '00000000'X   BUILDID = 87315

/* header and defaults table sections omitted for clarity           */

 DEFAULT TRCESTR='if api then parmsf time '
 00000001 01 CPNIN  SPINIT ('00050000'X) - SPI SPECIAL INIT
 PTRACE    1 CHAR   '
 PTRACE    1 CHAR   '   '
 TIME STAMP  24 AUG 1988 11:15:30  (40530.101501 Seconds)
 00000028 01 CPNOUT SPINIT ('00050000'X) - SPI SPECIAL INIT
 PTRACE    1 CHAR   ---INPUT ONLY PARAMETER-----
 TIME STAMP  24 AUG 1988 11:15:30  (40530.118412 Seconds)

 00000029 01 CPNIN  GSCHAR ('0C0C0500'X) - CHARACTER STRING AT
 PTRACE    1 FLOAT              50
 PTRACE    2 FLOAT              50
 PTRACE    3 DIM                 5
 PTRACE    4 CHAR   'abcde'
 TIME STAMP  24 AUG 1988 11:15:30  (40530.139099 Seconds)
 00000467 01 CPNOUT GSCHAR ('0C0C0500'X) - CHARACTER STRING AT
 PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    3 DIM    ---INPUT ONLY PARAMETER-----
 PTRACE    4 CHAR   ---INPUT ONLY PARAMETER-----
 TIME STAMP  24 AUG 1988 11:15:33  (40533.211295 Seconds)

 00000468 01 CPNIN  GSMOVE ('0C0C0400'X) - MOVE TO
 PTRACE    1 FLOAT              40
 PTRACE    2 FLOAT              60
 TIME STAMP  24 AUG 1988 11:15:33  (40533.219622 Seconds)
 00000473 01 CPNOUT GSMOVE ('0C0C0400'X) - MOVE TO
 PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----
 TIME STAMP  24 AUG 1988 11:15:33  (40533.271519 Seconds)

 00000474 01 CPNIN  GSLINE ('0C0C0401'X) - LINE TO
 PTRACE    1 FLOAT              70
 PTRACE    2 FLOAT              60
 TIME STAMP  24 AUG 1988 11:15:33  (40533.281272 Seconds)
 00000483 01 CPNOUT GSLINE ('0C0C0401'X) - LINE TO
 PTRACE    1 FLOAT  ---INPUT ONLY PARAMETER-----
 PTRACE    2 FLOAT  ---INPUT ONLY PARAMETER-----
 TIME STAMP  24 AUG 1988 11:15:33  (40533.283789 Seconds)

 00000484 01 CPNIN  ASREAD ('0C100000'X) - READ
 PTRACE    1 FIXED  ---OUTPUT ONLY PARAMETER-----
 PTRACE    2 FIXED  ---OUTPUT ONLY PARAMETER-----
 PTRACE    3 FIXED  ---OUTPUT ONLY PARAMETER-----
 TIME STAMP  24 AUG 1988 11:15:33  (40533.322064 Seconds)
 00000791 01 CPNOUT ASREAD ('0C100000'X) - READ
 PTRACE    1 FIXED               0
 PTRACE    2 FIXED               0
PTRACE    3 FIXED              0
TIME STAMP  24 AUG 1988 11:15:39  (40539.016880 Seconds)
00000792 01 CPNIN  FSTERM ('0C000000'X) - TERMINATION
TIME STAMP  24 AUG 1988 11:15:39  (40539.026388 Seconds)
00001173 01 CPNOUT FSTERM ('0C000000'X) - TERMINATION
TIME STAMP  24 AUG 1988 11:15:39  (40539.251072 Seconds)

00001251 END OF GDDM DIAGNOSTIC TRACE  24 AUG 1988 11:15 AM
```

Chapter 6. GDDM-PCLK diagnosis

Reporting problems to IBM

Only system-support personnel who support the host system licensed for the PCLKF feature can be the focal point for any contact with the IBM Support Center.

GDDM-PCLK is one of a matched pair of programs composed of:

GDDM-PCLK   A PC-resident licensed program.
PCLKF   A host-computer-resident feature available with Version 2.3 of GDDM/MVS, GDDM/VMXA, or GDDM/VSE.

Service for GDDM-PCLK is always applied to the basic-license PCLKF host program feature, and is automatically downloaded from the host to GDDM-PCLK at the matching version/release level. This is nut a mechanism for delivery of upgrades.

When a PC user reports a problem with GDDM-PCLK to you, try to solve it yourself before contacting the IBM Support Center about the basic-license PCLKF program feature. When you report problems to the IBM Support Center, quote the component identifier for PCLKF. The PCLKF component identifier depends on which version of the GDDM licensed program you are using on your host computer:

566535603	GDDM/MVS with PCLKF
566632803	GDDM/VSE with PCLKF
568400703	GDDM/VMXA with PCLKF.

Quoting the component identifier helps the IBM Support Center staff to check whether the problem is known, and if so to supply the fix. Service for PCLKF is supplied by IBM on magnetic tape, and performed using:

- SMP or SMP/E in the MVS environment,
- The GDDM Service Exec ADMSERV in the VM environment, or
- MSHP in the VSE environment.

When a PC user starts GDDM-PCLK host application support (option 1 from the GDDM-PCLK Main Panel), the serviced files are automatically downloaded from the host to the PC. For details, see the *GDDM-PCLK Guide*.

If the problem is not already known, the IBM Support Center staff may ask you to submit an Authorized Program Analysis Report (APAR). See page 124 for details of this. For some problems, the IBM Support Center staff may ask you to investigate further. In this instance, you may need to use the GDDM-PCLK trace facility described in "Accessing the service functions" on page 98, or you may need to run a GDDM trace. For information on invoking GDDM trace refer to Chapter 3, "GDDM tracing" on page 37. For general points on defining problem symptoms to the IBM Support Center, refer to Chapter 8, "Reporting GDDM problems to IBM" on page 117.

This section tells you how, for the basic-license PCLKF host program feature, to:

- Find out the level of GDDM-PCLK code
- Access the GDDM-PCLK Service Functions panel.

Finding the level of GDDM-PCLK code

If a PC user has a problem running GDDM-PCLK, the PC user must report the problem to you. You can then call the IBM Support Center, quoting the component identifier for PCLKF (see "Reporting problems to IBM" on page 97).

Note: The PC user *cannot* report the problem directly to the IBM Support Center staff.

You may need to know the level of GDDM-PCLK code on the PC. Find this out by entering the command:

```
pclklev
```

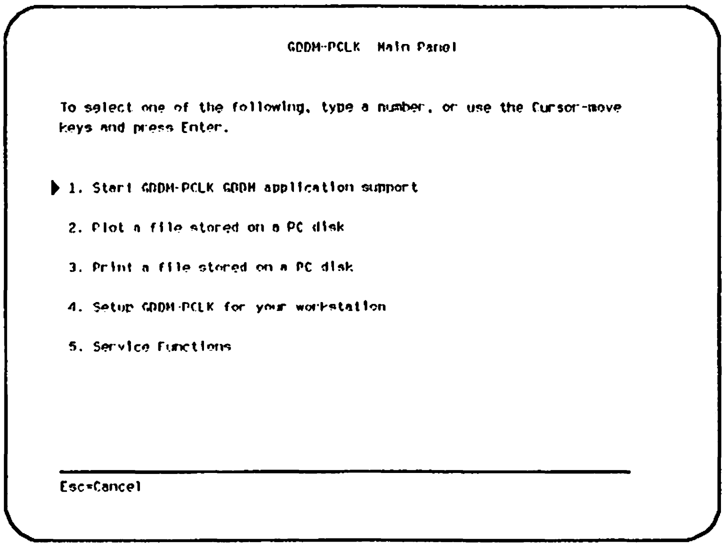
in the \PCLK11 subdirectory. GDDM-PCLK displays a list of the files that comprise GDDM-PCLK, with level numbers that you can quote to the IBM Support Center staff. The level numbers tell them if any service has been applied.

Accessing the service functions

The IBM Support Center staff may ask you to run a GDDM-PCLK trace. Access GDDM-PCLK trace from option 5 "Service functions" on the GDDM-PCLK Main Panel. To make option 5 appear, run GDDM-PCLK by entering the command:

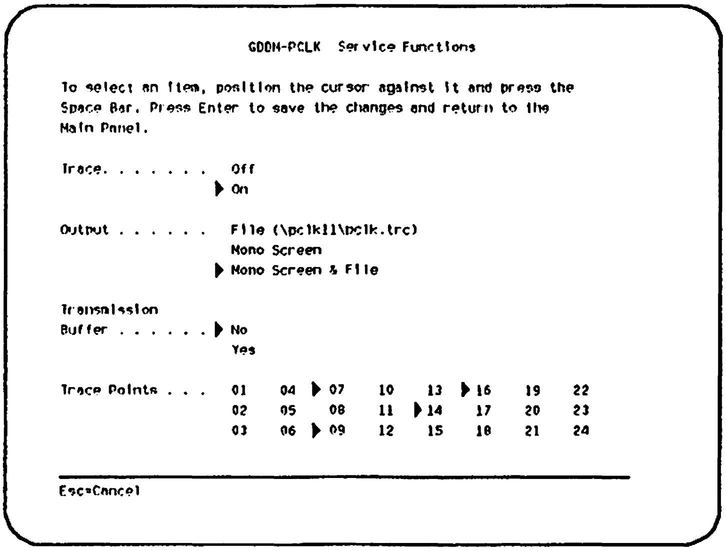
```
pclk /s
```

where /s stands for service. The Main Panel, with option 5 displayed, appears:



```
                        GDDM-PCLK  Main Panel

To select one of the following, type a number, or use the Cursor-move
keys and press Enter.

▶ 1. Start GDDM-PCLK GDDM application support

  2. Plot a file stored on a PC disk

  3. Print a file stored on a PC disk

  4. Setup GDDM-PCLK for your workstation

  5. Service Functions


________________________________________________________________________
Esc=Cancel
```

Choose option 5, and the service functions panel is displayed:



```
                  GDDM-PCLK  Service Functions

To select an item, position the cursor against it and press the
Space Bar. Press Enter to save the changes and return to the
Main Panel.

Trace. . . . . . .   Off
                   ▶ On

Output . . . . . .   File (\pclk11\pclk.trc)
                     Mono Screen
                   ▶ Mono Screen & File

Transmission
Buffer . . . . . . ▶ No
                     Yes

Trace Points . . .   01   04 ▶ 07   10   13 ▶ 16   19   22
                     02   05   08   11 ▶ 14   17   20   23
                     03   06 ▶ 09   12   15   18   21   24

_____________________________________________________________
Esc=Cancel
```

Use this panel to start the GDDM-PCLK trace facility. If you have a PC with a single screen or an IBM Personal System/2* (PS/2*), you are only given the option of sending the trace output to a file in the \PCLK11 subdirectory. If you have a dual-screen PC, you can send the trace output to a file, or to the mono screen only, or to both of these.

You can choose to trace the data stream that is sent, through the transmission buffer, to and from the host.

Trace points 01 through 24 represent the component parts of GDDM-PCLK, so you can restrict the trace to specific components. The IBM Support Center staff will tell you which trace points to select.

The trace points are:

01	XHGI calls
02	GQDE — Environmental component
03	GQDH — Hardcopy component
04	GQDK — Keystroke component
05	GQDL — Terminal emulator services component
06	GQDP — Panel services component
07	GQDS — Supervisor component
08	GQDDORD — Drawing order processor
09	GQDY — Common services component
10	GQDDBPP/GQDDPLT — Built-in procedure/plotter processor
11	GQDDCON — Query reply processor
12	GQDDFXF/GQDDPRT — File transfer/printer data stream processor
13	GQDDGDP/GQDDGDI/GQDDGDR — Graphics data processor
14	GQDDODP — Outbound data processor
15	GQDI — Installation component

* IBM trademark. For a complete list of trademarks, see page iii.

16	GQDA – Alpha merge component
17	Alpha data
18	GQDLKEY – Keystroke processing
19–24	(Spare).

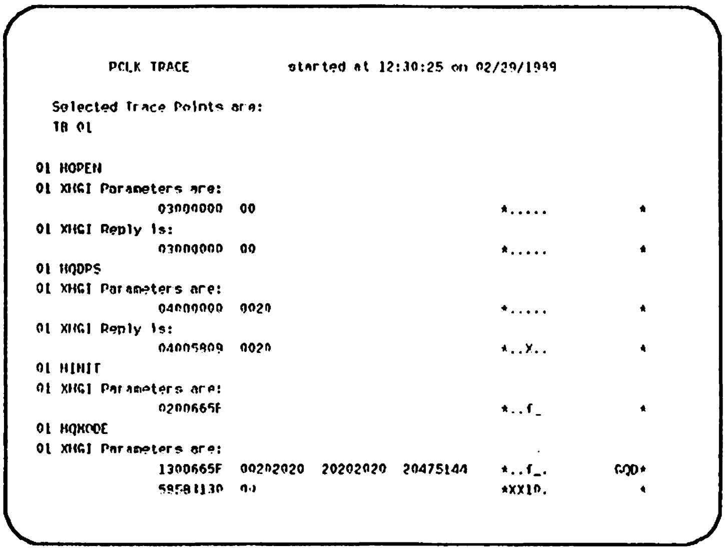
Turning on the trace point of one component activates the tracing of all modules within the component. For example, if trace point 04 is selected, all modules in the Keystroke component are traced.

The data stream component has no global trace point. To trace modules in the data stream component, you have to turn on one or more trace points (08, 10, 11, 12, 13, and 14) depending on the modules you need to trace.

More than one trace point can be active at once if there is sufficient spare memory to contain the message data for the trace points.

When you have completed the Service Functions panel, press Enter to save the changes, and return to the Main Panel. Then perform the operation that was causing the problem.

Here is a trace listing where transmission buffer and trace point 01 were chosen:



```
        PCLK TRACE              started at 12:30:25 on 02/29/1999

   Selected Trace Points are:
   TB 01

 01 HOPEN
 01 XHGI Parameters are:
              03000000  00                                *.....         *
 01 XHGI Reply is:
              03000000  00                                *.....         *
 01 HQDPS
 01 XHGI Parameters are:
              04000000  0020                              *.....         *
 01 XHGI Reply is:
              04005809  0020                              *..X..         *
 01 HINIT
 01 XHGI Parameters are:
              0200665F                                    *..f_          *
 01 HQMODE
 01 XHGI Parameters are:
              1300665F  00202020  20202020  20475144      *..f_.      GQD*
              59581130  00                                *XX10.         *
```

Trace options

When the trace output is sent to the screen, you can choose paged mode or single-line mode. In paged mode, whenever the trace output has filled the screen, GDDM-PCLK beeps and waits for you to press a key. In single-line mode, GDDM-PCLK stops and waits for a key to be pressed after every line of trace output is displayed.

When the trace output is sent to a file, you can choose whether to close and open the trace file for every trace record. If you do this, you will not lose any trace records even if the PC stops working. However, the PCLK performance will be very slow. The other option is to close and open the trace file after every

50 trace records. The PCLK performance will be less slow, but the latest trace records may be lost if the PC stops working.

You can choose whether to trace the control block and stack dumps at entry to every module you are tracing.

Trace control keys

You can change the trace options while GDDM-PCLK is in the polling loop, or while it is waiting for a key to be pressed in page mode or single-line mode. To change an option, hold down the Ctrl key and press another key. Each key combination is a toggle. Valid combinations are:

Ctrl-T	Trace
Ctrl-D	Trace to screen
Ctrl-P	Use page-mode display
Ctrl-S	Use single-line mode display
Ctrl-F	Trace to file
Ctrl-O	Close and open file after each write
Ctrl-B	Trace control block and stack dumps.

Trace output

There are four types of trace output:

- Module
- XHGI component
- Transmission buffer
- Control block.

Module

When a module is called, the entry to and exit from the module always causes a trace output if tracing is activated for that module. The trace output looks like this:

```
.
02  INTO GQDEDRV
  :
09   INTO GQDYLEV
  :
09   LEAVING GQDYLEV - RC=0000
  :
02  LEAVING GQDEDRV - RC=0000
```

The indentation of INTO and LEAVING indicates the nesting level of the call. The number on the left hand side shows the trace point the trace output belongs to.

Many modules give the requested function (why the module was called) and the return code:

```
02  INTO GQDEDRV
02   DRV - Init
  :
02  LEAVING GQDEDRV - RC=0000
```

In this example, module GQDEDRV was called for initialization (Init) and ended successfully (RC=0). To find out what module a trace output is from, use the trace point ID on the left-hand side (for example, 02 is GQDE component), and the three last letters of module name shown (for example, DRV).

All the other module trace records depend on the module.

The return code from DOS function call is often traced. Some modules trace the value assigned to an important variable with the variable name. Here is an example:

```
03  PTF - DOSFIRST RC = 00
03  PTF - file_count = 0008
```

A big module often traces the flow of process as follows:

```
07 STC - Initializing ODP
07 STC - Initializing PRT
07 STC - Setting Host Session Number
07 STC - Loading XHGI Driver
07 STC - Initializing GDP
07 STC - Initializing BPP
```

There are other types of trace output depending on the modules, however, they are mainly self-explanatory.

XHGI component

XHGI is the generic name for a set of program interfaces for display adapters. Calls to the XHGI component from the other components can be traced by activating the trace point 01. The trace output looks like this:

```
01 HQDPS
01 XHGI Parameters are:
              04000000  0020                        *.....          *
01 XHGI Reply is:
              04005809  0020                        *..X..          *
```

The trace output of each XHGI call begins with the XHGI entry point name (for example, HQDPS) followed by the parameters passed to the XHGI call both in hex and ASCII. If the XHGI call returns any parameters to the invoking routine, they are also traced in hex and ASCII.

The detail of the XHGI parameters can be found in the *IBM Personal System/2 Display Adapter 8514/A Technical Reference.*

Transmission buffer

The following information is traced by selecting the transmission buffer trace in the Service Functions panel:

- Outbound control field
- Outbound data
- Inbound control field
- Inbound data
- Attention identifier (AID) sent.

These are usually traced both in nibblised and denibblised form. Each trace record has the trace point ID of "TB" followed by the last three letters of TES-component module that activated the trace. The time of the trace is output as well.

An example of transmission buffer trace is:

```
TB  RCV - Denibblised Data is :
                                                            18:04:05
               F3000403  80000601  FF0380            *......... ..    *
```

Control block and stack

If active, the control block and stack are dumped at the entry to every module that is activated through the trace point.

The trace output shows all the GDDM-PCLK control blocks followed by the stack.

The stack is traced from the current top of the stack to the bottom of the stack area. If this is bigger than 512,bytes, only the top 512 bytes of the stack are traced.

Trace message data file

The trace message data file (GQDSTRCA.DAT) contains the text for most of the trace messages. It consists of a header table and a section for each trace point.

When tracing is initialized, the appropriate section of the trace data file is loaded into memory for each trace point selected. When fewer trace points are selected, better use is made of memory.

Trace code within GDDM-PCLK uses an index into the memory for each trace point.

Chapter 7. GDDM-OS/2 Link diagnosis

Reporting problems to IBM

Only system-support personnel who support your host GDDM program can be the focal point for any contact with the IBM Support Center. This section is addressed to system-support personnel.

GDDM-OS/2 Link is an OS/2* application that can be run with Version 2.3 of one of:

GDDM/MVS, GDDM/VMXA, or GDDM/VSE.

It can also be run with Version 2.2, if the appropriate APAR is installed. The APAR numbers are:

PL49183	MVS
PL49248	VSE
PL49249	VM/370
PL49259	VM/XA

When a PC user reports a problem with GDDM-OS/2 Link, try to solve it yourself before contacting the IBM Support Center. If you do have to report problems to the IBM Support Center, quote the component identifier for GDDM-OS/2 Link, which is 568811301; it is the same for whichever version of the GDDM licensed program you are using on your host computer. The following list identifies GDDM-OS/2 Link in relation to the host program:

GDDM/MVS with GDDM-OS/2 Link	568811301	FMID HDDM100
GDDM/VSE with GDDM-OS/2 Link	568811301	CLC A30
GDDM/VMXA with GDDM-OS/2 Link	568811301	Release level 112

Quoting the component identifier helps the IBM Support Center staff to check whether the problem is known, and if so which fix to supply for your host configuration. If the problem is not already known, the IBM Support Center staff may ask you to submit an Authorized Program Analysis Report (APAR). All problems concerning GDDM-OS/2 Link are reported through the normal GDDM support centers.

Service updates for GDDM-OS/2 Link are supplied by IBM on magnetic tape, and performed using:

- SMP or SMP/E in the MVS environment,
- The GDDM Service Exec ADMSERV in the VM environment, or
- MSHP in the VSE environment.

* IBM trademark. For a complete list of trademarks, see page iii.

Service updates for GDDM-OS/2 Link are applied to the host GDDM system. When a GDDM-OS/2 Link user starts a host GDDM application, the service files download automatically to GDDM-OS/2 Link.

For some problems, the IBM Support Center staff may ask you to investigate further. In this instance, you may need to use the GDDM-OS/2 Link trace facility described in "Running the GDDM-OS/2 Link trace program" on page 107, or you may need to run a GDDM trace.

For information on invoking GDDM trace, see "Accessing the service functions" on page 98.

For general points on defining problem symptoms to the IBM Support Center, see Chapter 8, "Reporting GDDM problems to IBM" on page 117.

System support information

For the basic-licensed GDDM-OS/2 Link program, this section tells you:

- The provisions of the license regarding the initial downloading of GDDM-OS/2 Link
- How to find the level of GDDM-OS/2 Link code
- How to run the GDDM-OS/2 Link trace program
- How to understand entries made in the Communications Manager error log by GDDM-OS/2 Link.

License control of GDDM-OS/2 Link

With GDDM-OS/2 Link running on a PC or PS/2, an application can use it as a graphics terminal in the OS/2 Extended Edition environment.

After the initial download of the program for which the license is payable, the PC can access any other host that has either GDDM 2.2 (with the appropriate APAR), or a later version installed.

Finding the level of GDDM-OS/2 Link code

If PC users have a problem running GDDM-OS/2 Link, they should report the problem to you, and you can then call the IBM Support Center if necessary, quoting the component identifier for GDDM-OS/2 Link; see "Reporting problems to IBM" on page 105.

Note: The PC user *cannot* report the problem directly to the IBM Support Center staff.

You may need to know the level of GDDM-OS/2 Link installed at the PC. Find this out by changing to the `c:\CMLIB` subdirectory and entering the command:

```
GQFLEVEL
```

in this subdirectory. GDDM-OS/2 Link displays a list of its files and their level numbers. These level numbers indicate to the IBM Support Center staff whether any service has been applied.

Trace support

Two types of trace support are available to GDDM-OS/2 Link:

- GQFTRACE.EXE, the internal trace program of GDDM-OS/2 Link and
- The trace component of OS/2 Extended Edition Common Services.

Running the GDDM-OS/2 Link trace program

The IBM Support Center staff may ask you to run a GDDM-OS/2 Link trace. Before starting the trace program, you must stop the OS/2 Extended Edition Common Services Communications Manager. Then start the trace program by issuing the command:

```
C:\CMLIB\GQFTRACE.EXE
```

This creates a message queue and then produces the trace window.

Note: The trace program cannot be started in the DOS box.

The default trace output file name of c:\GQFTRACE.DAT can be changed by specifying a different name as a parameter to GQFTRACE; for example:

```
C:\CMLIB\GQFTRACE.DAT A:\CMLIB\PROBLEM.TRC
```

Note: No trace records are written to this file until `Output to file` has been selected from the menu bar of the trace window.

The TRACE program can produce several error messages:

GQF0900 Insufficient memory is available.

GQF0910 TRACE is already running in another window. The TRACE program running in this window will be terminated.

GQF0920 TRACE was unable to start.

GQF0930 TRACE was unable to start because a program previously traced is still running.

GQF0940 TRACE was unable to open the output file; tracing to file has been disabled.

GQF0950 A DOS file error occurred; the reason code is given.

After you start the trace program, restart the Communications Manager.

As each traceable process of GDDM-OS/2 Link starts, a trace point dialog is displayed to enable you to select different trace points to trace for that process.

Trace point dialogs

Trace points 01 through 16 represent the different component parts of the process, so you can restrict the trace to specific parts.

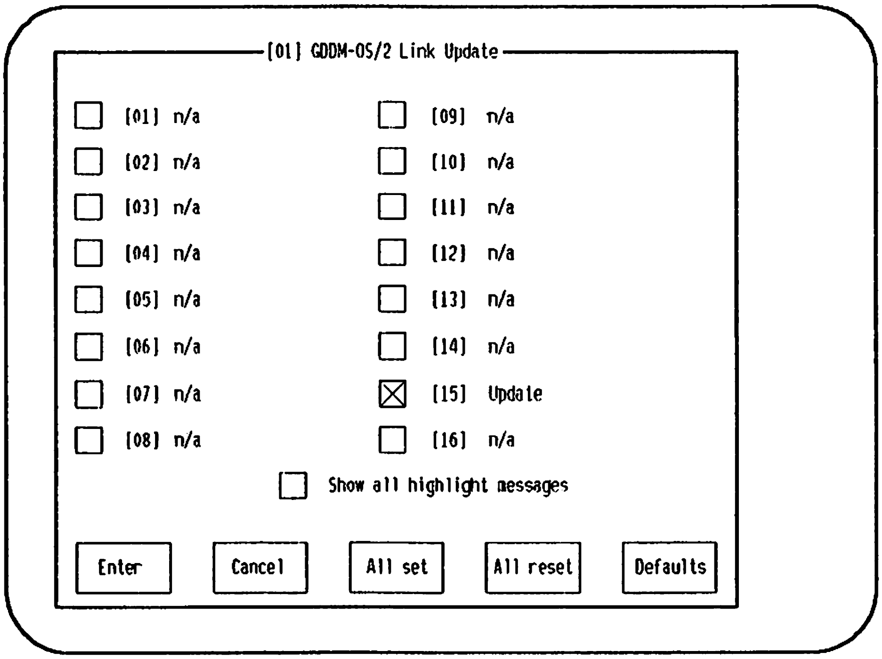
You do this by making one or more selections from the trace point dialogs as shown following. "n/a" means that the trace point is not applicable to that particular dialog.

Turning on the trace point of one component activates the tracing of all modules within the component.

The IBM Support Center staff will tell you which trace points to select.

Note: The screens shown here are typical only; they may vary in size and format according to the type of display in use.

Update dialog



[01] GDDM-OS/2 Link Update

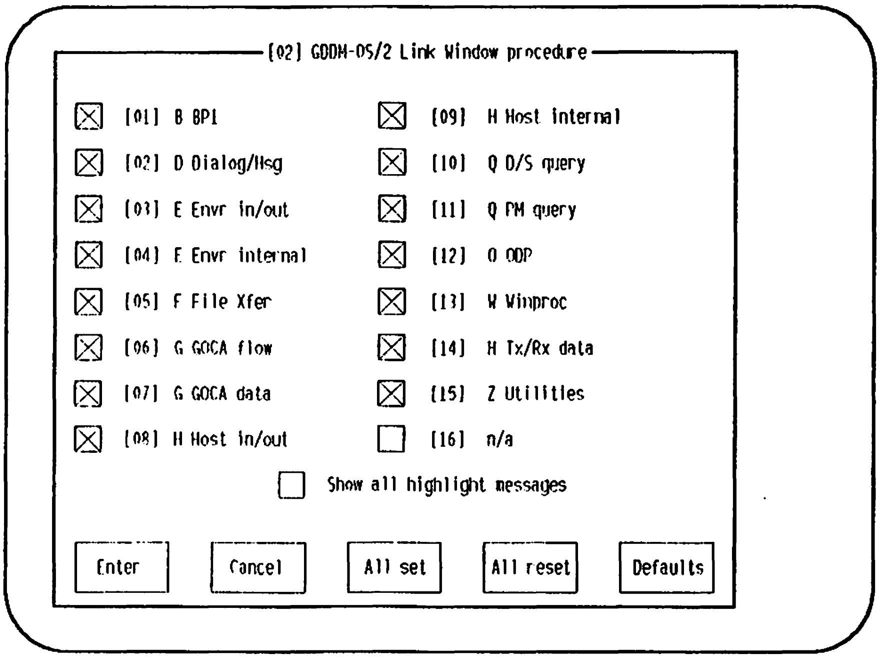
□ [01] n/a
□ [02] n/a
□ [03] n/a
□ [04] n/a
□ [05] n/a
□ [06] n/a
□ [07] n/a
□ [08] n/a
□ [09] n/a
□ [10] n/a
□ [11] n/a
□ [12] n/a
□ [13] n/a
□ [14] n/a
☒ [15] Update
□ [16] n/a

□ Show all highlight messages

Figure 1. GDDM-OS/2 Link: update dialog

Trace point 15: Update
Used by the Service update component.

Window procedure



[02] GDDM-OS/2 Link Window procedure

☒ [01] B BPI
☒ [02] D Dialog/Msg
☒ [03] E Envr in/out
☒ [04] E Envr internal
☒ [05] F File Xfer
☒ [06] G GOCA flow
☒ [07] G GOCA data
☒ [08] H Host in/out
☒ [09] H Host internal
☒ [10] Q O/S query
☒ [11] Q PM query
☒ [12] O OOP
☒ [13] W Winproc
☒ [14] H Tx/Rx data
☒ [15] Z Utilities
□ [16] n/a

□ Show all highlight messages

Figure 2. GDDM-OS/2 Link: window procedure dialog

Trace point 02: D Dialog/Msg
Used by the dialogs, message, and help components.

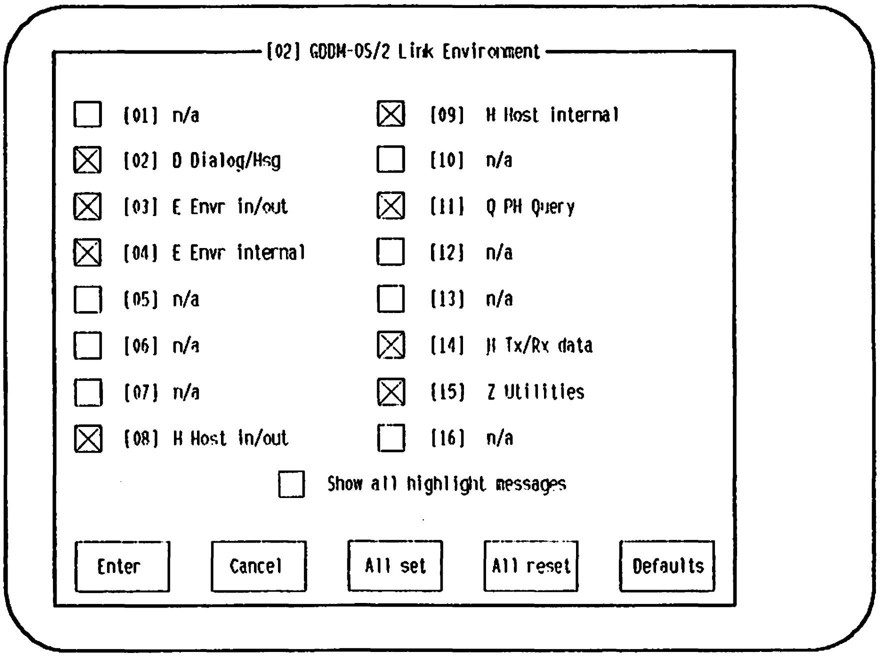
Trace point 03: E Envr in/out
Used by the high-level Environment component to trace entry and exit of each function.

Trace point 04: E Envr internal
Used by the high-level Environment component to trace internal processing.

Trace point 13: W Winproc
Used by the GDDM-OS/2 Link window procedure.

Trace point 15: Z Utilities
Used by the general-purpose utilities component.

Environment



[02] GDDM-OS/2 Link Environment

□ [01] n/a
☒ [02] D Dialog/Hsg
☒ [03] E Envr in/out
☒ [04] E Envr internal
□ [05] n/a
□ [06] n/a
□ [07] n/a
☒ [08] H Host in/out
☒ [09] H Host internal
□ [10] n/a
☒ [11] Q PH Query
□ [12] n/a
□ [13] n/a
☒ [14] H Tx/Rx data
☒ [15] Z Utilities
□ [16] n/a

□ Show all highlight messages

Figure 3. GDDM-OS/2 Link: environment dialog

Trace point 02: D Dialog/Msg
Used by the dialogs, message, and help components.

Trace point 03: E Envr in/out
Used by the high-level Environment component to trace entry and exit of each function.

Trace point 04: E Envr internal
Used by the high-level Environment component to trace internal processing.

Trace point 08: H Host in/out
Used by the Host Communication component to trace entry and exit from each function.

Trace point 09: H Host internal
Used by the Host Communication component to trace internal processing.

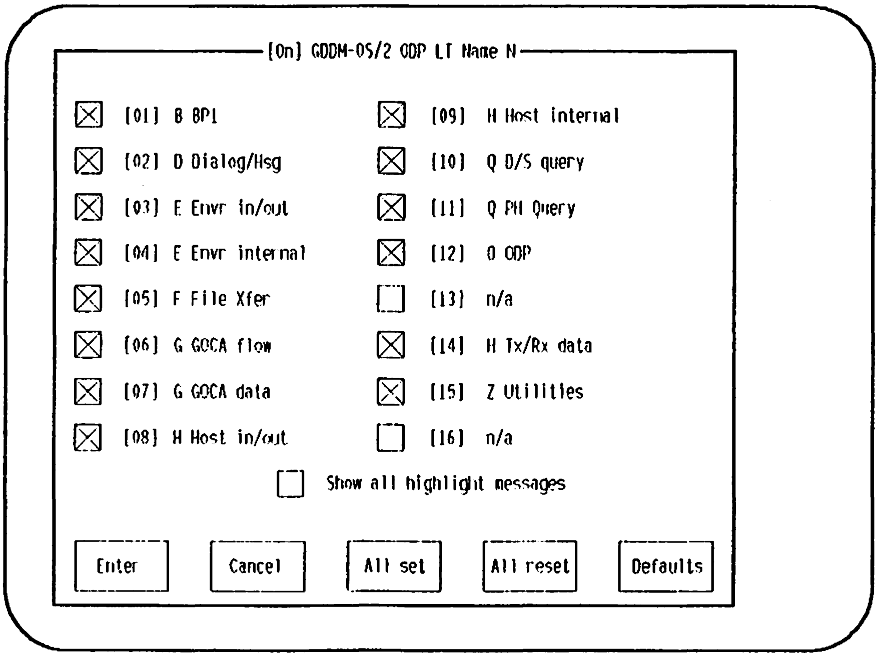
Trace point 11: Q PM Query
Used by the Query component to trace Presentation Manager Query processing.

Trace point 14: H Tx/Rx data
Used to trace the transmission buffer data sent and received from the host.

Trace point 15: Z Utilities
Used by the general-purpose utilities component.

Note: Any trace output resulting from one of these trace points being selected must be allowed to finish before you start your GDDM application.

LT Name N



[On] GDDM-OS/2 ODP LT Name N

☒ [01] B BP1
☒ [02] D Dialog/Msg
☒ [03] E Envr in/out
☒ [04] E Envr internal
☒ [05] F File Xfer
☒ [06] G GOCA flow
☒ [07] G GOCA data
☒ [08] H Host in/out
☒ [09] H Host internal
☒ [10] Q D/S query
☒ [11] Q PM Query
☒ [12] O ODP
□ [13] n/a
☒ [14] H Tx/Rx data
☒ [15] Z Utilities
□ [16] n/a

□ Show all highlight messages

Figure 4. GDDM-OS/2 Link: LT name dialog

Note: In the above screen heading, the identifier "On" and the LT name "a" varies according to which host session Logical Terminal (LT) the trace process is for.

Trace point 01: B BP1
Used by the graphic input device component.

Trace point 02: D Dialog/Msg
Used by the dialogs, message, and help components.

Trace point 03: E Envr in/out
Used by the high-level Environment component to trace entry and exit of each function.

Trace point 04: E Envr internal
Used by the high-level Environment component to trace internal processing.

Trace point 05: F File Xfer
Used by the File Transfer component when service updates are applied, or picture interchange files are transferred from the host.

Trace point 06: G GOCA flow
Used by the graphics data stream processing component to trace the flow of control.

Trace point 07: G GOCA data
Used by the graphics data stream processing component to trace the data being processed.

Trace point 08: H Host in/out
Used by the Host Communication component to trace entry and exit from each function.

Trace point 09: H Host internal
Used by the Host Communication component to trace internal processing.

Trace point 10: Q D/S query
Used by the Query component to trace internal processing.

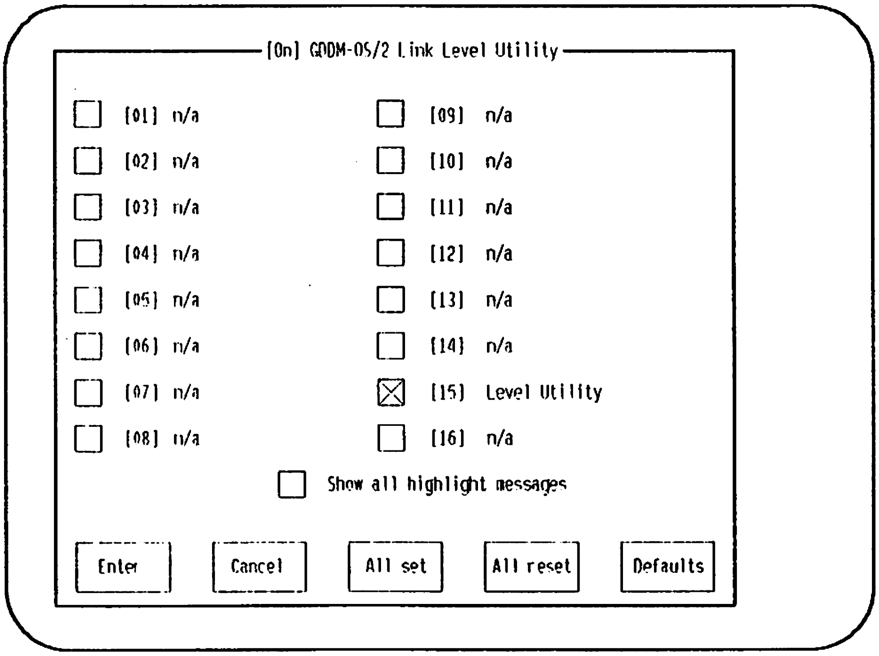
Trace point 11: Q PM Query
Used by the Query component to trace Presentation Manager Query processing.

Trace point 14: H Tx/Rx data
Used to trace the transmission buffer data sent and received from the host.

Trace point 15: Z Utilities
Used by the general-purpose utilities component.

Note: Before stopping the GDDM-OS/2 Link trace program, you must stop the OS/2 Extended Edition Common Services Communications Manager.

Level Utility



[On] GDDM-OS/2 Link Level Utility

□ [01] n/a
□ [02] n/a
□ [03] n/a
□ [04] n/a
□ [05] n/a
□ [06] n/a
□ [07] n/a
□ [08] n/a
□ [09] n/a
□ [10] n/a
□ [11] n/a
□ [12] n/a
□ [13] n/a
□ [14] n/a
☒ [15] Level Utility
□ [16] n/a

□ Show all highlight messages

Figure 5. GDDM-OS/2 Link: Level Utility dialog

Trace point 15: Level Utility
Used by the Level Utility GQFLEVEL.EXE.

Trace options

By selecting TRACE POINTS on the menu bar, you can change the trace points selected for any process at any time.

If OUTPUT is selected on the menu bar, you can select where the trace output is to be sent. This can be to the screen (which it is by default), or to the file specified in the optional parameter to GQFTRACE.EXE, or to both of these; see page 107. If no output file has been specified, the default output file GQFTRACE.DAT is created in the root directory of the C drive.

When the trace output is sent to a file, you can choose whether to close and open the trace file for every trace record. If you do this, you do not lose any trace records even if the PC stops working. The GDDM-OS/2 Link performance, however, will be very slow. The other option is to close and open the trace file after every 50 trace records. In this case, the GDDM-OS/2 Link performance is faster, but the most recent trace records can be lost if the PC stops working.

Tracing through modules

When a module is called, the entry and exit from the module always causes a trace output if tracing is activated for that module. The trace output looks like this:

```
[01] (03) INTO GqfeProcDQ @19d7c324
 [01] (04) ...
   .   .
   .   .
 [01] (08) INTO GqfhPcReceiveData @19d7e5bc
  [01] (09) ...
    .   .
    .   .
 [01] (08) LEAVING GqfhPcReceiveData 010d202d
 [01] (04) ...
   .   .
   .   .
[01] (03) LEAVING GqfeProcDQ 00000000
```

The indentation between INTO and LEAVING indicates the nesting level of the call. The number in parentheses on the left-hand side, for example (08), shows the trace point the trace output belongs to.

The number in square brackets on the left-hand side for example [01], shows the process the trace output belongs to.

The return code from an OS/2 system call is often traced. Some modules trace the value assigned to an important variable with the variable name. Here is an example:

```
[02] (13) hwnd=00a864c0, Msg 0070, Parms 00ed012a 00000000
```

There are other types of trace output depending on the modules; however, they are mainly self-explanatory.

Trace as part of OS/2 Extended Edition Common Services

GDDM-OS/2 Link uses OS/2 Extended Edition Common Services for trace. This assists problem isolation among the different products. The common services use the same test points as GDDM-OS/2 Link, but put the output in a trace buffer. There are two trace controls; code 27 for data stream, and code 28 for the parameters and communication flow.

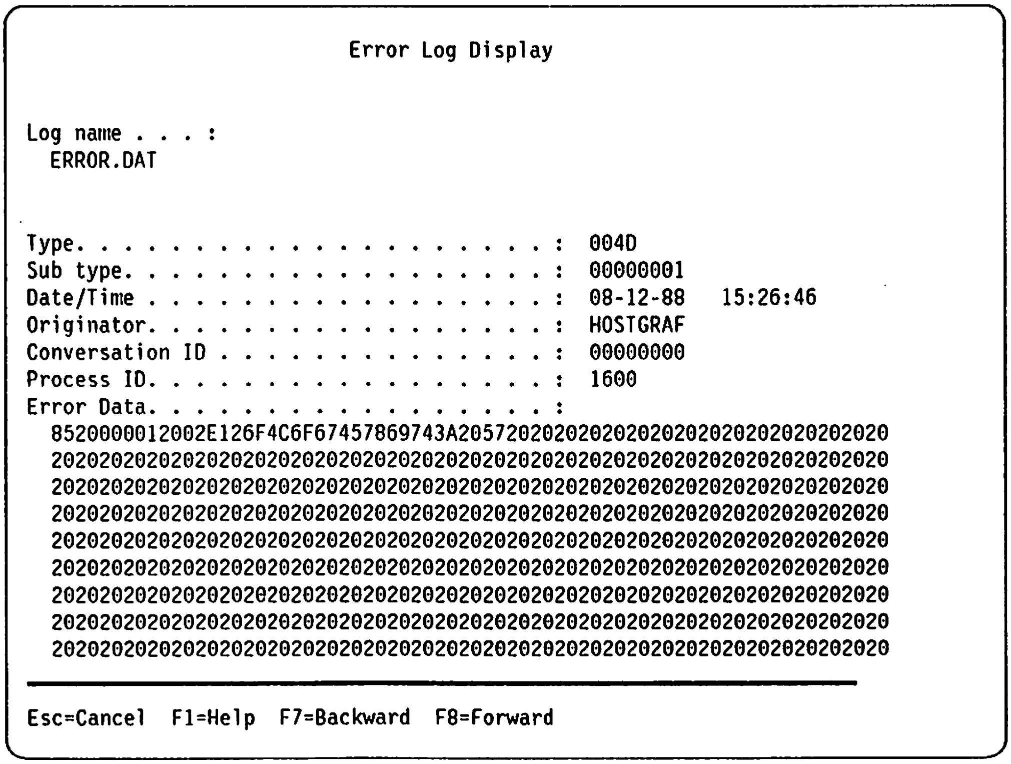
Sequential tracing, when data flows from one device to another, can be done through this trace. For information on how to run this trace, see *IBM Operating System/2 Extended Edition Version 1.2 Problem Determination Guide for the Service Coordinator.*

Using OS/2 Extended Edition Common Services Error Log entries

If GDDM-OS/2 Link detects an error, it enters these details into the Communications Manager error log:

Type
Subtype
Originator
Conversation ID
Process ID
Error Data.

The error data field contains this error-dependent information:



```
                           Error Log Display


Log name . . . :
  ERROR.DAT


Type. . . . . . . . . . . . . . . . . . . . . . :  004D
Sub type. . . . . . . . . . . . . . . . . . . . :  00000001
Date/Time . . . . . . . . . . . . . . . . . . . :  08-12-88   15:26:46
Originator. . . . . . . . . . . . . . . . . . . :  HOSTGRAF
Conversation ID . . . . . . . . . . . . . . . . :  00000000
Process ID. . . . . . . . . . . . . . . . . . . :  1600
Error Data. . . . . . . . . . . . . . . . . . . :
  8520000012002E126F4C6F67457869743A205720202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020
  2020202020202020202020202020202020202020202020202020202020202020202020

Esc=Cancel  F1=Help  F7=Backward  F8=Forward
```

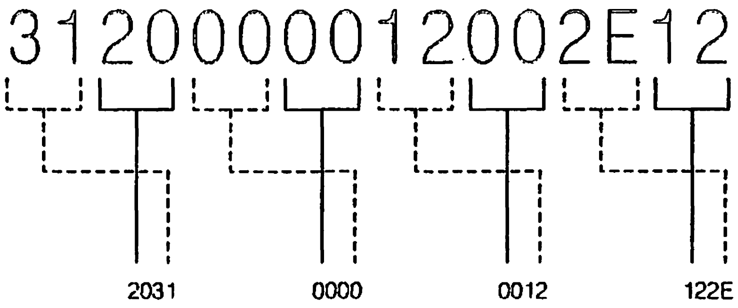
Figure 6. Example of common services error log entries

Values that can be taken for figure 6 are shown in this table:

Error log entry	Valid values	Explanation of error
Type	004D	GDDM-OS/2 Link
Subtype	00000000	GDDM-OS/2 Link
	00000001	Presentation Manager
	00000002	DOS
	00000003	Communications manager queue
	00000004	Install program
Originator	HOSTGRAF	GDDM-OS/2 Link
Conversation ID	00000000	GDDM-OS/2 Link
Process ID	various	DOS Program ID for application generating the error
Error Data	See page 115.	

How to interpret the error data in log messages

For error log entries with subtype = 00000001, the error data field contains a Presentation Manager error code. The error code is the first two bytes, which must be byte-reversed for correct interpretation. The Presentation Manager errors are listed in Appendix A of the *OS/2 Programming Tools and Information: Programming Reference.* They are also listed in the various Presentation Manager header files available in the OS/2 Extended Edition Toolkit.



For example, if the error data is 3120 (see the illustration above), this represents error code 2031, which is PMERR_HARDWARE_INIT_FAILURE.

For error log entries with subtype = 0000002, the error data field contains a DOS error code. The error code is the first two bytes, which must be byte-reversed for correct interpretation. DOS errors can be identified using the OS/2 HELP command, by converting the error code from hexadecimal to decimal, and prefixing it with "SYSO". For example, if the error data is 1500, this represents error code 0015, which is 0021 in decimal; therefore you must type `HELP SYS00021`.

For error log entries with other subtype values, the error data field contains information on an error generated by GDDM-OS/2 Link, which should be specified if you report the error to IBM; see Chapter 8, "Reporting GDDM problems to IBM" on page 117. IBM service personnel can find more detailed information in the appropriate service documentation.

Chapter 8. Reporting GDDM problems to IBM

If you find a problem in one of the licensed programs covered by this book, you should report the problem to the IBM Support Center. To do that, fill in a problem report form (see page 118), then pass the form to the Support Center. The Support Center representative will search the Software Support Facility (SSF) database to determine whether your problem has already been reported, and if a fix is available.

For some problems, the IBM Support Center may ask you to investigate the problem further. You may need to run a GDDM trace or use the GDDM-IMD, GDDM-REXX, GDDM-PCLK, or GDDM-OS/2 Link diagnostics facility. These aids are described earlier in this book.

Note: In some instances, you might be asked to submit an Authorized Program Analysis Report (APAR). The section "Submitting an APAR" on page 124 gives more information on this.

How to use a problem report form

Complete a problem report form whenever you need to define a GDDM problem, even if the information you record is not to be later passed on to the IBM Support Center. You can use completed forms to create and maintain a local register of problems and solutions. Such a register could help you solve some problems without the need to call the IBM Support Center.

If you need further guidance about the information you must supply, see "Defining the symptom of the problem" on page 119, "Defining the operating environment" on page 122, or "Defining the devices in use" on page 123.

We recommend that you prepare several copies of the master problem report form (on page 118), and distribute those copies among your company's system programmers and administrative personnel.

Note: Do not write on the master problem report form.

IBM PROBLEM REPORT FORM

1. GDDM version, release, and modification: ______________________________

2. GDDM Program Update Tape (PUT) level: ______________________________

3. Operating system/subsystem: ______________________________ ___

4. Operating system/subsystem release and PUT levels: ______________________________

5. Controller type and model: ______________________________

6. Controller configuration and code level: ______________________________ ___

7. Device type and model: (including printers): ______________________________ ___

8. If VTAM, VCNA, or VSCS, show PSERVIC: ______________________________

9. Device tokens in use: ______________________________

10. If CICS, DFHPCT SCRNSZE, DFHTCT feature: ______________________________

11. If ABEND, give ABEND code, CSECT name, and offset: ____________________

    ____

12. Brief description of the problem including full message text:

    ____

    ____

    ____

    ____

    ____

    ____

13. Applicable entries in ADMDEFS: ______________________________

    ____

14. If ABEND, give registers and PSW: ______________________________

    ____

15. If a trace is available, state type of trace: ____________________

Defining the symptom of the problem

The symptom is whatever you first notice that leads you to think that there is a problem. Symptoms can be grouped into the following categories, each of which corresponds to an APAR type:

- Unexpected, wrong, or no output
- Device program check
- GDDM message
- GDDM abend
- Loop
- Degraded performance
- Wait
- Error in the GDDM documentation.

The following sections of this chapter describe these categories in more detail. Select the category that best describes your problem.

Unexpected, wrong, or no output

This category is APAR type 'INCORROUT'.

This includes output to displays, printers, plotters, and GDDM files (such as GDF files). When you report the problem, describe what output you expected and what you received that led you to suspect an error.

Note: Unexpected output from the GDDM/TSO Print Utility may have some GDDM messages associated with it. These appear on the system console log.

Device program check

This is APAR type 'PROGCKxxx'.

This is an error detected by a terminal, printer, or plotter. On a terminal, the error usually appears as 'PROGxxx' in the operator information area. On a printer, a code may be displayed on a panel at the front of the device. Further device-specific problem analysis may be needed; the IBM Support Center will advise you.

GDDM message

This is APAR type 'MSGxxxxxxx'.

Some GDDM messages contain variable fields, for example:

```
ADM0411 E DEVICE EXCEPTION a1.  SENSE=n2
```

It is important with this type of message that you tell the IBM Support Center the text of the message in full.

Note: Messages produced by the GDDM/TSO Print Utility and GDDM applications running in MVS batch or VSE batch appear on the system console log.

GDDM abend

This is APAR type 'ABENDUxxxx'.

A GDDM abend is issued when GDDM determines that it cannot continue. Some abend codes are produced from more than one module, and further problem analysis may be needed; the IBM Support Center will advise you.

A list of GDDM abends and the modules that issue them are listed in Appendix B, "Abend codes" on page 155.

If you are using GDDM-IVU, an abend subcode is contained in register 0; you should report this with the abend code.

System abend

This is APAR type 'ABENDSxxxx'.

A system abend is issued when the operating system determines that execution cannot continue. Report the module (CSECT) and the offset into that module at which the abend occurred.

If the subsystem produces a diagnostic message, that message normally incorporates the abend code. An exception is TSO where, *after* receiving the message:

```
PROGRAM TERMINATED DUE TO ERROR+
```

you need to enter a question mark (?). TSO then produces a second-level message incorporating the abend code, if applicable. If you then press ENTER, TSO produces a dump. Note that a dump can only be produced if a SYSABEND or SYSUDUMP file has been previously allocated. You may have to run the application again to reproduce the problem.

Loop

This is APAR type 'LOOP'.

Standard host system and subsystem techniques or the GDDM trace function can be used to diagnose loop and wait states. If in doubt, use the trace facilities described in Chapter 3, "GDDM tracing" on page 37 to determine whether there is a loop.

Degraded performance

This is APAR type 'PERFM'.

Apparent poor performance by GDDM may be caused by the host system or subsystem. Report this as a GDDM problem only if you determine that GDDM is responsible, or if the performance problem occurs as the result of the installation of a PTF or a new release.

Wait

This is APAR type 'WAIT'.

GDDM contains a small number of calls to system or subsystem wait services. It also issues requests for system services. These may produce wait states. Report this as a GDDM problem only if you determine that GDDM causes the wait.

GDDM documentation error

This is APAR type 'DOC'.

This covers wrong, missing, or unclear information in the GDDM library.

Note: For missing messages, report a message problem. For errors in the utility panels and the GDDM-IMD tutorial, report an unexpected output problem.

If you have a documentation problem, report the correct title and order number for the manual in error. The GDDM library is shown on page ix.

Add the numbers of any TNLs (Technical Newsletters) that have been applied to the manuals.

Defining the operating environment

Report the following four facts about the operating environment at the time that the problem occurred.

1. Program Identifier

   GDDM is made up of licensed programs, as shown below:

Program	Contents
GDDM/MVS, GDDM/VSE, GDDM/VMXA	a. Graphics and alphanumerics API b. Image Symbol Editor c. Run-time mapping.
GDDM-PGF	a. Presentation Graphics routines b. Interactive Chart Utility c. Vector Symbol Editor.
GDDM-IMD	Interactive creation of screen and printer maps at program development time.
GDDM-IVU	Viewing, creating, modifying, storing, and printing images.
GDDM-REXX	Use of GDDM from EXECs written for the VM/System Product Interpreter.
GDDM-GKS	Use of GDDM with Graphical Kernel System functions.
GDDM-PCLK	Use of GDDM on a PC linked to a host computer.
GDDM-OS/2 Link	Use of GDDM with GDDM-OS/2 Link linked to a host computer.
GDDM-CSPF	Background plotting; generating slides and background slide production.

   In addition, the GDDM base programs and GDDM-PGF, GDDM-IVU, and GDDM-GKS each have a National Language feature that provides translation of panels and messages into several languages. If you find a problem with the translation of a panel or a message, report the problem to your IBM representative, but do not submit an APAR.

   Note: National Language support for the other GDDM programs is provided on the same tape as the programs themselves.

2. Release Level

   This is GDDM Version 2, Release 3. You can identify it as 'R23'.

3. Application

   Report the application being run. Identify it as one of these types:

   - A program written by the user
   - A program written by a third party
   - A program written by IBM, but not a GDDM utility
   - A GDDM utility (ICU, Vector Symbol Editor, Image Symbol Editor, Print Utility, IVU, GDDM-CSPF, GDDM-PCLK, and GDDM-OS/2 Link)

4. When running MVS or MVS/XA, specify whether you are using IMS/VS, CICS, or TSO.

Defining the devices in use

GDDM supports a wide range of devices. It is important that you report the device environment accurately. You should report:

1. The device type and model
2. The controller type and model to which the device is attached
3. The device tokens, if used
4. The device definitions.

An application always has a primary device opened. This may be a real device, or a dummy device. If it is a real device, report its device type and model and those of the controller that it is attached to. If it is a dummy device, report the device token. The application has a dummy device if it is running in a batch environment or on a disconnected VM machine.

Some applications may use device tokens even though they have a real primary device. Report this.

The application may also have an alternate device open. If there is an alternate device and it has a device token associated with it, report this token.

If you are using a device token supplied by GDDM, report the token name. If you have created your own token, you may be asked to give the definitions you used to create it.

If the symptom is associated with a GDDM Family 1 printer, report the LU type of the printer.

On some subsystems, the device definitions to the subsystem or telecommunications access method are important. If you are using:

- TSO, report the PSERVIC.
- GDDM/TSO Print Utility, report the PSERVIC.
- IMS, report the device token being used by the system definition database.
- CICS, report the feature operands in the DFHTCT macro.
- PCLK, report the emulator in use and the definitions on the PCLK setup panels.

Reporting the problem

When you have completed your problem report form, report the problem to the IBM Support Center staff. The action taken by the Support Center depends on whether a fix exists for the problem you report.

If there is no fix, they will give you guidance on what to do next.

Submitting an APAR

The IBM Support Center staff may ask you to submit an APAR on your GDDM problem. If so, they will give you an *APAR number* and *severity code*, and if necessary will help you to complete the form.

Appendix A. Data areas

This section lists the major GDDM control blocks that are likely to be produced in trace output. They are presented in alphabetic order of mnemonic. The description of each has three parts:

1. A reference list that precedes the detailed portion of the control block. The format of the reference list is:

Control Block mnemonic and title	For example, ADMTAAB — application anchor block (AAB)
Function	A brief description of what the control block does.

2. An alphabetic index for all items contained in the control block.
3. A tabular description of the storage layout of the control block. The following information is provided:

   Offset Field addresses in hexadecimal relative to the beginning of the control block.

   Type The type of item, which will be one of:

A	Address
B	Bit string
C	Character string
F	Fixed binary signed
U	Unsigned binary.

   Bit positions and values in bit strings are shown in the following examples:

   ```
   1... .... (a reference to bit 0)
   .... ..11 (a reference to bits 6 and 7).
   ```

   Length The length of the field in bytes, as a decimal value.

   Name The name of the field.

   Description A brief description of the meaning of the field.

The formats of GDDM data records are described in the *GDDM Base Programming Reference* manual.

ADMTAAB — application anchor block (AAB)

Function

The application anchor block (AAB), contains information which is passed between the application and GDDM when the application is using the reentrant application call interface.

Specifically, it contains an anchor used by GDDM for all its cross-invocation storage, feedback areas for returning error codes back to the application, and an optional application-dependent extension.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
AABAP	4	AABFC	0	AABXNEPA	8
AABEC	2	AABSC	0	AABXTN	8
AABEXTD	0	AABXFEPA	C	ADMTAAB	0

Storage layout

Offset	Type	Length	Name	Description
0	C	8	ADMTAAB	APPLICATION ANCHOR BLOCK
0	F	4	AABFC	GDDM FULLWORD FEEDBACK CODE
0	F	2	AABSC	GDDM SEVERITY CODE
2	F	2	AABEC	GDDM ERROR CODE
4	A	4	AABAP	GDDM ANCHOR POINTER
8	C	0	*	APPLICATION DEPENDENT EXTENSION
0	C	16	AABEXTD	EXTENDED AAB USED ONLY FOR THE GDDM ALLOCATED, STATIC AAB USED FOR THE NACI
0	C	8	*	NORMAL AAB
8	C	8	AABXTN	AAB EXTENSION
8	A	4	AABXNEPA	NORMAL ENTRY POINT ADDRESS
C	A	4	AABXFEPA	FAST ENTRY POINT ADDRESS

ADMTACA – application interface control area

Function

This is the main anchor block of the application interface component (AIC). It contains the general AIC status, and pointers to other AIC data. There is one instance per GDDM representation. It is organized in two sections. The first section contains those fields that are initialized by GDDM stub linkage code. The second section contains those fields that are initialized by the subsystem-independent AIC controller code.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ACAAABP	3C	ACAIFCBP	54	ACASTXTP	D8
ACAACPX	A0	ACAIFTYP	4A	ACATRACC	86
ACAAPLTP	64	ACAINIT	3A	ACATRCEF	38
ACAATRNP	9C	ACAINSMN	D0	ACATRCEP	6C
ACABLANG	87	ACAMEP0P	94	ACATRCFL	70
ACACFDTP	60	ACAMSTGI	D2	ACATRCLC	7E
ACADSAOP	58	ACAMSTGL	C8	ACATRCLP	88
ACADSAP	8	ACAMSTGP	E4	ACATRCNN	78
ACADSASP	4	ACAMSX0P	C	ACATRCPC	7C
ACADTRNP	98	ACANLCPX	AC	ACATRCQM	F0
ACAESSF	B0	ACAOBCPL	A4	ACATRCSI	84
ACAEXITH	90	ACAOBCPS	A8	ACATRCSP	80
ACAEXITP	8C	ACAPINSP	DC	ACATRCTB	68
ACAEXPBP	C0	ACAPLPTR	44	ACATRC13	74
ACAEXTRP	E8	ACAPRDCT	CC	ACATR58L	F4
ACAEXT31	8C	ACAPRMNO	48	ACATXTMN	CE
ACAFLAGS	38	ACAPTXTP	D4	ACAUCXAF	38
ACAFLG1	38	ACAPVTAP	34	ACAUETPS	BC
ACAFORCE	86	ACAPVTEP	10	ACAULANG	93
ACAGKEH	92	ACAQERB	100	ACAUR1	4C
ACAGKSEP	14	ACAQERBP	C4	ACAUR14	50
ACAGKSSP	EC	ACARCPP	40	ADMTACA	0
ACAICBP	B4	ACASINSP	E0		
ACAID	0	ACASOSSP	5C		

Storage layout

Offset	Type	Length	Name	Description
0	C	*	ADMTACA	AIC CONTROL AREA
0	C	4	ACAID	EYE CATCHER 'ACA '
STUB-DEPENDENT SECTION				
4	A	4	ACADSASP	A(INITIAL DSA STACK)
8	A	4	ACADSAP	A(INITIAL DSA)
C	A	4	ACAMSX0P	A(DEFAULT ERROR EXIT)
1	A	4	ACAPVTEP	A(ESI PRODUCT VECTOR TABLE)
1	A	4	ACAGKSEP	A(GKS ERROR HANDLER)
1	A	4	* (7)	SPARE FIELDS
STUB-INDEPENDENT SECTION				
34	A	4	ACAPVTAP	A(AIC PRODUCT VECTOR TABLE)
GENERAL FLAGS AND INDICATORS				
38	C	2	ACAFLAGS	GENERAL FLAGS

Offset	Type	Length	Name	Description
38	C	1	ACAFLG1	FLAG BYTE 1
	1...		*	SPARE
	.1..		ACATRCEF	TRACE PROCESSING IS OPEN
	..1.		ACAUCXAF	USER CALL EXIT IS ACTIVE
39	C	1	*	RESERVED
3A	F	2	ACAINIT	INITIALIZATION INDICATOR
APPLICATION CALL RELATED FIELDS				
3C	A	4	ACAAABP	A(AAB)
40	A	4	ACARCPP	A(REQUEST CONTROL PARAMETER)
44	A	4	ACAPLPTR	A(USER PARMS, LESS AAB AND RCP)
48	F	2	ACAPRMNO	NO. OF USER PARMS
4A	F	2	ACAIFTYP	INVOCATION TYPE 0 NACI, 1 RACI, 2 SPI
4C	F	4	ACAUR1	INVOKER'S REGISTER 1
50	F	4	ACAUR14	INVOKER'S REGISTER 14
HOUSEKEEPING RELATED FIELDS				
54	A	4	ACAIFCBP	A(INTERFACE CONTROL BLOCK)
58	A	4	ACADSAOP	A(OVERFLOW DSA STACK)
5C	A	4	ACASOSSP	A(SOS STACK)
SPARE FIELDS				
60	A	4	ACACFDTP	A(CALL FORMAT DESCR TABLE)
64	A	4	ACAAPLTP	A(APL EQUIVALENCE TABLE)
TRACE-RELATED FIELDS				
68	A	4	ACATRCTB	A(IN CORE TRACE TABLE)
6C	A	4	ACATRCEP	A(TRACE PROCESSOR)
70	A	4	ACATRCFL	A(TRACE FILE REQUEST BLOCK)
74	A	4	ACATRC13	A('CURRENT DSA')
78	F	4	ACATRCNN	CURRENT TRACE SEQUENCE NO.
7C	F	2	ACATRCPC	CURRENT PAGE COUNT
7E	F	2	ACATRCLC	CURRENT LINE COUNT
80	A	4	ACATRCSP	A(TRACE STG USE TABLES)
84	F	2	ACATRCSI	INDEX OF CUR STG USE ENTRY
	1...		ACAFORCE	FORCE SPECIFIED = 1
	.1..		ACATRACC	TRACE STRING ACCEPTED = 1
	..11 1111		*	
87	C	1	ACABLANG	BACKUP LANGUAGE

88	A	4	ACATRCLP	A(TRACE LOOP DETECT TABLE)
				SEE EXTENSION FIELDS FOR
				A(EX TRACE CONTROL BLOCK)

ERROR PROCESSING RELATED FIELDS				
8C	A	4	ACAEXITP	A(USER ERROR EXIT)
	1...		ACAEXT31	AMODE OF USER EXIT
90	F	2	ACAEXITH	CURRENT ERROR EXIT THRESHOLD
	1...		ACAGKEH	GKS ERR HANDLING IN PROGRESS
	.111 1111		*	SPARE FIELDS
93	C	1	ACAULANG	CURRENTLY USED LANGUAGE
94	A	4	ACAMEP0P	A(ERROR PROCESSOR)
COUNTRY EXTENDED CODE PAGE (CECP) RELATED FIELDS				
98	A	4	ACADTRNP	A(GDDM ADMDATRN MODULE)
9C	A	4	ACAATRNP	A(CURRENT DATRN MODULE)
A0	F	4	ACAACPX	APPLICATION CODE PAGE INDEX

Offset	Type	Length	Name	Description
A4	F	4	ACAOBCPL	OBJECT CODE PAGE LOAD
A8	F	4	ACAOBCPS	OBJECT CODE PAGE SAVE
AC	F	4	ACANLCPX	NATLANG CODE PAGE INDEX
B0	C	1		
	1...		ACAESSF	ESSCPG API CALL FLAG
	.111 1111		*	SPARE
B1	C	3	*	SPARE
B4	A	4	ACAICBP	A(ADMTICB)
B8	A	4	* (1)	SPARE
FURTHER ERROR PROCESSING RELATED FIELDS				
BC	A	4	ACAUETPS	A(USER ERROR TEXT TABLES)
C0	A	4	ACAEXPBP	A(FSEXIT PARAMETER BLOCK)
C4	A	4	ACAQERBP	A(QUERY ERROR BLOCK)
FURTHER EXTENSION FIELDS				
C8	F	4	ACAMSTGL	L(MERGE STORAGE BLOCK)
CC	F	2	ACAPRDCT	GDDM FAMILY PRODUCT COUNT
CE	F	2	ACATXTMN	MESSAGE TEXT MODULE COUNT
D0	F	2	ACAINSMN	MESSAGE INSERT MODULE COUNT
D2	F	2	ACAMSTGI	MERGE STORAGE BLOCK INDEX
D4	A	4	ACAPTXTP	A(PRIM MSG TEXT MODULE TAB)
D8	A	4	ACASTXTP	A(SEC MSG TEXT MODULE TAB)
DC	A	4	ACAPINSP	A(PRIM MSG INSERT MODULE TAB)
E0	A	4	ACASINSP	A(SEC MSG INSERT MODULE TAB)
E4	A	4	ACAMSTGP	A(MERGE STORAGE BLOCK)
E8	A	4	ACAEXTRP	A(EX TRACE CONTROL BLOCK
EC	A	4	ACAGKSSP	A(GKS DSA STACK)
EXTRA FIELDS FOR ENHANCED TRACE				
F0	A	4	ACATRCQM	A(QUAD MAX. LAST QUAD)
F4	F	4	ACATR58L	LENGTH FOR 5080 PARM TRACE
POINTERS FOR FURTHER FUNCTIONS NOT YET IMAGINED				
F8	A	4	* (2)	SPARE FIELDS
FOR EASE OF CONTROL BLOCK EXTENSION, THE FOLLOWING FIELD IS ALWAYS LAST, AND NORMALLY ADDRESSED VIA THE FIELD ACAQERBP				
100	C	*	ACAQERB	QUERY ERROR BLOCK (SEE ADMTQERB)

ADMTCQYV – VM/CMS display communications block

Function

This control block contains the control information that is used by the VM/CMS input/output routine.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTCQYV	0	CQYDVSTT	22	CQYPSCSW	50
CQYDADSP	2A	CQYDVTYP	21	CQYPSDTA	60
CQYDARMT	2A	CQYD8CL	34	CQYPSLCC	50
CQYDATTN	2B	CQYD8CP	38	CQYPSNSB	60
CQYDATTR	2A	CQYHDLEN	14	CQYPSNSC	60
CQYDAVCN	2A	CQYHEAD	10	CQYPSNSD	60
CQYDDISC	2B	CQYHPLEN	10	CQYPSNSE	60
CQYDEV	18	CQYPATH	40	CQYPSNSI	60
CQYDLLEN	28	CQYPATTN	58	CQYPSNSN	60
CQYDNUMB	1C	CQYPBUSY	58	CQYPSNSO	60
CQYDQR	2C	CQYPCC	50	CQYPSNSU	60
CQYDQRCL	2E	CQYPCCCK	59	CQYPSTMD	58
CQYDQREC	2C	CQYPCCW	54	CQYPSUB	51
CQYDQREH	2C	CQYPCDCK	59	CQYPUNCK	58
CQYDQRFL	2C	CQYPCHCK	59	CQYPUNEX	58
CQYDQRPN	2D	CQYPCHEN	58	CQYPUST	58
CQYDQRPS	2C	CQYPCST	59	CQYPXWRD	44
CQYDQRRW	30	CQYPCUE	58	CQYSECT	10
CQYDQR14	2C	CQYPDVEN	58	DISPASYF	2
CQYDQYCD	32	CQYPEXIT	40	DISPBUFF	9
CQYDRCLS	24	CQYPFLG	48	DISPCLR	2
CQYDREAL	24	CQYPICCK	59	DISPCMD	8
CQYDRFTR	27	CQYPICL	59	DISPDEVA	0
CQYDRMDL	26	CQYPKSL	50	DISPFLAG	2
CQYDRTYP	25	CQYPLIO	48	DISPNORD	4
CQYDSTAT	2B	CQYPLOG	50	DISPOPCC	C
CQYDTMCD	29	CQYPORB	51	DISPOPER	8
CQYDUSCT	18	CQYPPCI	59	DISPPA1	2
CQYDVCLS	20	CQYPPGCK	59	DISPRCDE	3
CQYDVCNS	28	CQYPPRCK	59	DISPRW	8
CQYDVFLG	23	CQYPRCT	5A	DISPSIZE	E
CQYDVIRT	20	CQYPSCNT	5C	DISPWAIT	2

Storage layout

Offset	Type	Length	Name	Description
0			ADMTCQYV	DISPLAY COMMUNICATIONS BLOCK
0	F	2	DISPDEVA	DEVICE ADDRESS (-1 = CONSOLE)
2	B	1	DISPFLAG	OPTION BYTE
	1...		DISPCLR	CLEAR SCREEN BEFORE WRITE
	.1..		DISPWAIT	WAIT AFTER WRITE
	..1.		DISPPA1	CALLER WANTS TO SEE PA1
	...1		DISPASYF	ASYNCH I/O RECEIVED
	 1111		*	
3	F	1	DISPRCDE	RETURN CODE FOR SNAP TRACING
4	F	2	DISPNORD	NO. OF BYTES READ/WRITTEN
6	F	2	*	NOT USED
8	C	8	DISPOPER	START OF OPERATION LIST
8	B	1	DISPCMD	3270 LOCAL COMMAND CODE
	1111 111.		*	
	1		DISPRW	1 IF WRITE COMMAND
9	A	3	DISPBUFF	ADDRESS OF READ/WRIRE BUFFER
C	F	2	DISPOPCC	CCW FLAGS AND CTL

Offset	Type	Length	Name	Description
E	F	2	DISPSIZE	LENGTH OF READ/WRITE BUFFER

Mapping of CQYSECT for Console Query function

10			CQYSECT	QUERY REPLY BUFFER
10			CQYHEAD	REPLY BUFFER HEADER
10	F	4	CQYHPLEN	LENGTH OF PATH SECTION
14	F	4	CQYHDLEN	LENGTH OF DEVICE SECTION
18			CQYDEV	DEVICE SECTION
18	F	4	CQYDUSCT	NO. OF PATHS OPENED TO THIS DEVICE
1C	F	4	CQYDNUMB	VIRTUAL DEVICE NUMBER
20			CQYDVIRT	VIRTUAL DEVICE INFO
20	C	1	CQYDVCLS	VIRTUAL DEVICE TYPE CLASS
21	C	1	CQYDVTYP	VIRTUAL DEVICE TYPE
22	C	1	CQYDVSTT	VIRTUAL DEVICE STATUS
23	C	1	CQYDVFLG	VIRTUAL DEVICE FLAGS
24			CQYDREAL	REAL DEVICE INFO
24	C	1	CQYDRCLS	REAL DEVICE TYPE CLASS
25	C	1	CQYDRTYP	REAL DEVICE TYPE
26	C	1	CQYDRMDL	REAL MODEL NUMBER
27	C	1	REAL FEATURE CODE	
28			CQYDVCNS	MORE DIAG X'24' INFO
28	C	1	CQYDLLEN	LINE LENGTH
29	C	1	CQYDTMCD	TERMINAL CODE
2A	B	1	CQYDATTR	DEVICE ATTRIBUTE FLAG
	1111 1...		*	RESERVED
	.... .1..		CQYDARMT	DEVICE IS A REMOTE 3270
	.... ..1.		CQYDADSP	DEVICE IS A DISPLAY
	1		CQYDAVCN	DEVICE IS A VIRTUAL CONSOLE
2B	B	1	CQYDSTAT	DEVICE STATUS FLAG 1
	1...		CQYDATTN	ATTENTION PENDING
	.1..		CQYDDISC	DEVICE IS DISCONNECTED
	..11 1111		*	RESERVED
2C			CQYDQR	FIRST 6 BYTES 8C INFO
2C	B		CQYDQRFL	FLAGS
	1...		CQYDQREC	EXTENDED COLOR
	.1..		CQYDQREH	EXTENDED HIGHLIGHTING
	..1.		CQYDQRPS	PSS
	...1 111.		*	NOT USED
	1		CQYDQR14	14-BIT ADDRESSING
2D	C	1	CQYDQRPN	NUMBER OF PARTITIONS
2E	F	2	CQYDQRCL	NUMBER OF COLUMNS
30	F	2	CQYDQRRW	NUMBER OF ROWS
32	C	1	CQYDQYCD	DEVICE QUERY CODE
33	C	1	*	RESERVED
34	F	4	CQYD8CL	LENGTH OF REMAINING 8C INFO
38	A	4	CQYD8CP	PTR TO WSF INFO AFTER FIRST 6 BYTES OF 8C INFO
3C	F	4	*	RESERVED
40			CQYPATH	PATH DATA SECTION
40	A	4	CQYPEXIT	USER EXIT ADDRESS
44	F	4	CQYPXWRD	USER WORD
48	B	1	CQYPFLG	PATH FLAG
	1...		CQYPLIO	PATH DID LAST I/O
	.111 1111		*	RESERVED
49	C	3	*	RESERVED
4C	F	4	*	RESERVED
50			CQYPSCSW	CHANNEL STATUS WORD
50	B	1	CQYPSLCC	LOGOUT PEND/COND CODES
	1111 1...		CQYPKSL	KEY/SUSPEND/LOG BITS
	.... .1..		CQYPLOG	LOGOUT PENDING
	11		CQYPCC	DEFERRED CONDITION CODE
51	1111 1...		CQYPORB	FLAGS FROM ORB
	.... .111	19 bits	CQYPSUB	SUBCHANNEL CONTROL BITS
	1111 1111			

Offset	Type	Length	Name	Description
	1111 1111			
54	A	4	CQYPCCW	LAST CCW EXECUTED
58	B	1	CQYPUST	UNIT STATUS BYTE
	1...		CQYPATTN	ATTENTION
	.1..		CQYPSTMD	STATUS MODIFIER
	..1.		CQYPCUE	CONTROL UNIT END
	...1		CQYPBUSY	BUSY
	.... 1...		CQYPCHEN	CHANNEL END
	.... .1..		CQYPDVEN	DEVICE END
	.... ..1.		CQYPUNCK	UNIT CHECK
	1		CQYPUNEX	UNIT EXCEPTION
59	B	1	CQYPCST	CHANNEL STATUS BYTE
	1...		CQYPPCI	PROGRAM-CONTROLLED INT.
	.1..		CQYPICL	INCORRECT LENGTH
	..1.		CQYPPGCK	PROGRAM CHECK
	...1		CQYPPRCK	PROTECTION CHECK
	.... 1...		CQYPCDCK	CHANNEL DATA CHECK
	.... .1..		CQYPCCCK	CHANNEL CONTROL CHECK
	.... ..1.		CQYPICCK	INTERFACE CONTROL CHECK
	1		CQYPCHCK	CHAINING CHECK
5A	F	2	CQYPRCT	RESIDUAL COUNT
5C	F	4	CQYPSCNT	SENSE COUNT
60	C	32	CQYPSDTA	SENSE DATA
	1...		CQYPSNSC	COMMAND REJECT
	.1..		CQYPSNSI	INTERVENTION REQUIRED
	..1.		CQYPSNSB	BUS-OUT CHECK
	...1		CQYPSNSE	EQUIPMENT CHECK
	.... 1...		CQYPSNSD	DATA CHECK
	.... .1..		CQYPSNSU	UNIT SPECIFY
	.... ..1.		CQYPSNSN	CONTROL CHECK
	1		CQYPSNSO	OPERATION CHECK
61	C	31	*	REMAINDER OF SENSE DATA

ADMTDFT — general defaults table

Function

This control block contains the current settings of all GDDM defaults and user exits. It also contains a pointer to the internal list of nickname entries; see "ADMTNICK — nickname list entry format" on page 147.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDFT - General defaults table					
ADMTDFT	0	DFTOERP	44	DFTONSNF	32
DFTH	0	DFTOERTH	3C	DFTOOBJO	24
DFTHHEAD	0	DFTOERTY	40	DFTOOPNF	32
DFTHLENG	4	DFTOESIF	30	DFTOPA2F	32
DFTHUDS	8	DFTOFFEF	31	DFTOPVCF	31
DFTHUDSL	8	DFTOFFSF	31	DFTOREMF	32
DFTHUDSP	C	DFTOFRCV	D8	DFTOSNAF	32
DFTXTNA	E0	DFTOFSXK	1C	DFTOSOSF	1C
DFTO	30	DFTOFSXP	18	DFTOSVBZ	5C
DFTOABRF	31	DFTOGKWT	90	DFTOSYNF	31
DFTOAPCP	EC	DFTOGSXK	14	DFTOSYSO	2C
DFTOAUKB	31	DFTOGSXP	10	DFTOTFMT	36
DFTOCIBL	64	DFTOICUF	DA	DFTOTRBZ	58
DFTOCIBP	68	DFTOICUI	3A	DFTOTRCE	4C
DFTOCMPF	31	DFTOICUP	D9	DFTOTRCN	50
DFTOCPG4	88	DFTOICUS	DB	DFTOTRLI	D4
DFTOCTLS	D4	DFTOINCP	E8	DFTOTRNP	DC
DFTODBDG	D4	DFTOLANG	34	DFTOTRWD	D0
DFTODBDY	D4	DFTOLCLF	32	DFTOTSHR	D4
DFTODBLN	39	DFTOMGST	60	DFTOUXBO	28
DFTODBMX	78	DFTOMSOC	38	DFTOXFBF	32
DFTODFMT	35	DFTOMSOF	31	DFTOXFBL	70
DFTODFSF	30	DFTONFMT	37	DFTOXFBP	74
DFTOEINP	F0	DFTONICK	6C	DFTO0	10
DFTOERL	48	DFTONLSF	30		
ADMTDFTO - Object type defaults (Offset defined by DFTOOBJO)					
ADMTDFTO	0	DFTOOBJT	0		
ADMTDFTU - User exit blocks (Offset defined by DFTOUXBO)					
ADMTDFTU	0	DFTUXCIK	10	DFTUXTSC	0
DFTUXCI	C	DFTUXCIP	14	DFTUXTSK	4
DFTUXCIC	C	DFTUXTS	0	DFTUXTSP	8
ADMTDFTC - CICS-specific defaults (Offset defined by DFTOSYSO)					
ADMTDFTC	0	DFTCIFMT	30	DFTCSYSP	14
DFTCDECK	8	DFTCPRAP	20	DFTCTISF	0
DFTCDFPX	24	DFTCPRNT	10	DFTCTQRY	40
DFTCIADS	18	DFTCSRAP	1C	DFTCTRTD	4
DFTCIESL	28	DFTCSTGF	38	DFTCTSPX	C
ADMTDFTI - IMS-specific defaults (Offset defined by DFTOSYSO)					
ADMTDFTI	0	DFTIOBDK	98	DFTISHUT	60
DFTICHRT	48	DFTIOBDS	88	DFTISYSP	70
DFTIDBXX	80	DFTIOBRK	90	DFTITRDD	20

Name	Offset	Name	Offset	Name	Offset
DFTIDECK	28	DFTIOBRS	80	DFTIUIOL	10
DFTIEXIT	58	DFTIPRNT	30	DFTIUMAX	C
DFTIISSE	38	DFTISDSD	50	DFTIVSSE	40
DFTIMAST	68	DFTISDSF	A8	DFTIWTOD	18
DFTIMODN	78	DFTISDSS	A0	DFTIWTOR	14

ADMTDFTT - TSO-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTT	0	DFTTEMUL	0	DFTTPRNT	28
DFTTADVF	0	DFTTIADS	38	DFTTSYSP	30
DFTTCOLM	50	DFTTICBP	4	DFTTS99S	8
DFTTCPT	68	DFTTIESL	20	DFTTS99U	60
DFTTDECK	18	DFTTIFMT	40	DFTTTRDD	10
DFTTDFDD	58	DFTTMONO	48		

ADMTDFTV - CMS-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTV	0	DFTVDFFT	68	DFTVPRNF	18
DFTVADVF	0	DFTVIADS	28	DFTVSYSP	20
DFTVCOLM	48	DFTVIFMT	30	DFTVTMFT	8
DFTVCPT	70	DFTVMONO	40	DFTVTRFN	50
DFTVDECK	10	DFTVMSLT	38	DFTVTRFT	58
DFTVDFFN	60				

ADMTDFTD - VSE-specific defaults (Offset defined by DFT0SYSO)

ADMTDFTD	0	DFTDDFDD	18	DFTDTRDD	0
DFTDCOLM	10	DFTDMONO	8		

Storage layout

Offset	Type	Length	Name	Description
ADMTDFT - General defaults table				
0	C	248	ADMTDFT	GENERAL DEFAULTS TABLE
0	C	16	DFTH	HEADER
0	C	4	DFTHHEAD	'DFT 'EYE CATCHER
4	F	4	DFTHLENG	LENGTH OF DEFAULTS TABLE
8	C	8	DFTHUDS	DUMMY UDS INFORMATION
8	F	4	DFTHUDSL	DUMMY LENGTH (NON ZERO)
C	A	4	DFTHUDSP	DUMMY ADDRESS(ZERO)
10	C	32	DFT00	STORAGE EXIT DEFAULTS
10	A	4	DFT0GSXP	ADDR(GET STG APPL EXIT)
14	F	4	DFT0GSXK	APPL DEFINED GET STORAGE TOKEN
18	A	4	DFT0FSXP	ADDR(FREE STG APPL EXIT)
1C	F	4	DFT0FSXK	APPL DEFINED FREE STORAGE TOKEN
20	1...		DFT0SOSF	SHORT ON STORAGE PROCESSING
	.1..		*	
	..1.		*	
	...1		*	
	.... 1...		*	
	.... .1..		*	
	.... ..1.		*	
	1		*	
21	C	1	*	
22	C	1	*	
23	C	1	*	
24	F	4	DFT0OBJO	OFFSET TO OBJECT TYPE DEFAULTS
28	F	4	DFT0UXBO	OFFSET TO USER EXIT BLOCKS
2C	F	4	DFT0SYSO	OFFSET TO SUBSYSTEM SPECIFIC DTF
30	C	200	DFT0	SUBSYSTEM COMMON DEFAULTS
30	1...		DFT0ESIF	INITIALIZE ESI ONLY

Offset	Type	Length	Name	Description
	.1..		DFTODFSF	DFT FILE DFTS ON SPIB
	..1.		DFTONLSF	NATLANG DFTS ON SPIB
	...1		*	
	.... 1...		*	
	.... .1..		*	
	.... ..1.		*	
	1		*	
31	1...		DFTOPVCF	SPI PARAMETER VERIFICATION
	.1..		DFTOABRF	RETURN ON ABEND
	..1.		DFTOSYNF	SYNCHRONIZED I/O
	...1		DFTOCMPF	PS COMPRESSION ALLOWED
	.... 1...		DFTOAUKB	ALWAYS UNLOCK KEYBOARD
	.... .1..		DFTOFFSF	FORM FEED AT START
	.... ..1.		DFTOFFEF	FORM FEED AT END
	1		DFTOMSOF	USE SO/SI IN MIXED FIELDS
32	1...		DFTOLCLF	ALL DEVICES ARE LOCAL
	.1..		DFTOREMF	ALL DEVICES ARE REMOTE
	..1.		DFTOSNAF	ALL DEVICES ARE SNA
	...1		DFTONSNF	ALL DEVICES ARE NON SNA
	.... 1...		*	
	.... .1..		DFTOOPNF	MUST OPEN DEVICE ON FSINIT
	.... ..1.		DFTOPA2F	RETURN PA2 TO ASREAD(VM)
	1		DFTOXFBF	ATTN FBACK BLOCK SET(VM)
33	C	1	*	
34	C	1	DFTOLANG	NATIONAL LANGUAGE CHARACTER
35	F	1	DFTODFMT	DATE CONVENTION
36	F	1	DFTOTFMT	TIME CONVENTION
37	F	1	DFTONFMT	NUMBER CONVENTION
38	C	1	DFTOMSOC	SO/SI EMULATION CHARACTER
39	C	1	DFTODBLN	DBCS SYMBOL SET LANGUAGE CHAR
3A	F	1	DFTOICUI	ICU ISOLATE VALUE
3B	C	1	*	
3C	F	4	DFTOERTH	ERROR THRESHOLD VALUE
40	F	4	DFTOERTY	TYPE OF ERROR EXIT ACTION
44	A	4	DFTOERP	A(F/BACK BLOCK)
48	F	4	DFTOERL	L(F/BACK BLOCK)
4C	F	4	DFTOTRCE	TRACE CONTROL VALUE
50	F	4	DFTOTRCN	INCORE TRACE TABLE SIZE
54	C	4	*	
58	F	4	DFTOTRBZ	TRANSMISSION BUFFER SIZE
5C	F	4	DFTOSVBZ	FSSAVE BUFFER SIZE
60	F	4	DFTOMGST	MAPGROUP STORAGE THRESHOLD
64	F	4	DFTOCIBL	L(CALL INFO FEEDBACK BLOCK)
68	A	4	DFTOCIBP	A(CALL INFO FEEDBACK BLOCK)
6C	A	4	DFTONICK	NICKNAME LIST CHAIN ANCHOR
70	F	4	DFTOXFBL	ATTENTION FBACK BLOCK LENGTH
74	A	4	DFTOXFBP	ATTENTION FBACK BLOCK ADDRESS
78	F	4	DFTODBMX	DBCS COMPONENT IN CORE THRESHOLD
7C	C	4	*	
80	C	8	*	
88	C	8	DFTOCPG4	DEFAULT CODE PAGE NAME FOR 4250
90	C	8	DFTOGKWT	(8) GKS WKSTATION DFLT DVC TOKENS
D0	F	4	DFTOTRWD	TRACE OUTPUT WIDTH
D4	A	4	DFTOTRLI	TRACE LIST CHAIN ANCHOR
D8	1...		DFTOTSHR	TRACE FILE SHARING
	.1..		DFTODBDG	GDDM DETERMINED DBCS SUPPORT
	..1.		DFTODBDY	DBCS SUPPORTED
	...1		DFTOCTLS	DEFAULT CTLSAVE VALUE
	.... 1...		DFTOFRCV	FORCE HPA VALIDATION
	.... .1..		*	
	.... ..1.		*	
	1		*	
D9	F	1	DFTOICUP	ICU PANEL COLOR
DA	F	1	DFTOICUF	ICU FORMAT DEFAULTS

Offset	Type	Length	Name	Description
DB	F	1	DFTOICUS	ICU SYMBOL SET USE
DC	A	4	DFTOTRNP	TRANSLATE TABLE ADDRESS
E0	A	4	DFTXTNA	ADDRESS OF DEFAULTS EXTENSION BLOCK
E4	C	4	*	
E8	F	4	DFTOINCP	INSTALLATION CODE PAGE
EC	F	4	DFTOAPCP	APPLICATION CODE PAGE
F0	1...		DFTOEINP	ENABLE CECP INPUT
	.111 1111		*	
F1	C	7	*	

ADMTDFTO - Object type defaults (Offset defined by DFTOOBJO)

0	C	96	DMTDFTO	OBJECT TYPE DEFAULTS
0	C	8	DFTOOBJT	(12) OBJECT TYPE TABLE

ADMTDFTU - User exit blocks (Offset defined by DFTOUXBO)

0	C	24	DMTDFTU	USER EXIT BLOCKS
0	C	12	DFTUXTS	TASK SWITCH EXIT UXBLOCK
0	F	4	DFTUXTSC	USER EXIT CODE
4	F	4	DFTUXTSK	USER EXIT TOKEN
8	A	4	DFTUXTSP	USER EXIT ADDRESS
C	C	12	DFTUXCI	CALL INTERCEPT EXIT UXBLOCK
C	F	4	DFTUXCIC	USER EXIT CODE
10	F	4	DFTUXCIK	USER EXIT TOKEN
14	A	4	DFTUXCIP	USER EXIT ADDRESS

ADMTDFTC - CICS-specific defaults (Offset defined by DFTOSYSO)

0	C	68	DMTDFTC	CICS SPECIFIC DEFAULTS
0	1...		DFTCTISF	USE TRANS INDEPENDENT SERVICES
	.111 1111		*	
1	C	1	*	
2	C	1	*	
3	C	1	*	
4	C	4	DFTCTRTD	CICS TRACE TD NAME
8	C	4	DFTCDECK	CICS DECK O/P TD NAME
C	C	4	DFTCTSPX	CICS TEMP STORAGE PREFIX
10	C	4	DFTCPRNT	CICS GDDM PRINT UTILITY NAME
14	C	4	DFTCSYSP	CICS SYSTEM PRINTER TD NAME
18	C	4	DFTCIADS	CICS IMD ADS O/P TD NAME
1C	A	4	DFTCSRAP	CICS STG AUDIT TRAIL ANCHOR
20	A	4	DFTCPRAP	CICS PGM AUDIT TRAIL ANCHOR
24	C	4	DFTCDFPX	CICS DEFAULTS TS PREFIX
28	C	8	DFTCIESL	CICS IMD ADMGIMP FCT NAME
30	C	8	DFTCIFMT	CICS IMD STAGED DATA FILETYPE
38	C	8	DFTCSTGF	CICS IMD STAGED FILE FCT NAME
40	C	4	DFTCTQRY	CICS TEMP STORAGE PREFIX FOR QR

ADMTDFTI - IMS-specific defaults (Offset defined by DFTOSYSO)

0	C	176	ADMTDFTI	IMS-SPECIFIC DEFAULTS
0	1111 1111		*	
1	C	3	*	
4	A	4	*	
8	A	4	*	
C	F	4	DFTIUMAX	IMS GDDM UTIL MAX NO OF USERS
10	F	4	DFTIUIOL	IMS GDDM INPUT AREA SIZE
14	C	4	DFTIWTOR	IMS WTO ROUTING CODES
18	C	4	DFTIWTOD	IMS WTO DESCRIPTOR CODES
1C	C	4	*	
20	C	8	DFTITRDD	IMS TRACE DD NAME
28	C	8	DFTIDECK	IMS DECK O/P LTERM NAME

Offset	Type	Length	Name	Description
30	C	8	DFTIPRNT	IMS GDDM PRINT UTILITY NAME
38	C	8	DFTIISSE	IMS ISE TRANSACTION NAME
40	C	8	DFTIVSSE	IMS VSE TRANSACTION NAME
48	C	8	DFTICHRT	IMS ICU TRANSACTION NAME
50	C	8	DFTISDSD	IMS GDDM SYSDEF DBD NAME
58	C	8	DFTIEXIT	IMS UTIL EXIT STRING
60	C	8	DFTISHUT	IMS UTIL SHUTDOWN STRING
68	C	8	DFTIMAST	IMS UTIL SHUTDOWN LTERM NAME
70	C	8	DFTISYSP	IMS SYSTEM PRINTER DEST NAME
78	C	8	DFTIMODN	IMS GDDM MFS MOD NAME
80	C	48	DFTIDBXX	IMS SEGMENT/KEY FIELD NAMES
80	C	8	DFTIOBRS	OBJECT DB ROOT SEGMENT NAME
88	C	8	DFTIOBDS	OBJECT DB DEP SEGMENT NAME
90	C	8	DFTIOBRK	OBJECT DB ROOT KEY FIELD NAME
98	C	8	DFTIOBDK	OBJECT DB DEP SEGMENT KEY FIELD NAME
A0	C	8	DFTISDSS	SYSDEF DB SEGMENT NAME
A8	C	8	DFTISDSF	SYSDEF DB KEY FIELD NAME

ADMTDFTT - TSO-specific defaults (Offset defined by DFT0SYSO)

0	C	112	ADMTDFTT	TSO SPECIFIC DEFAULTS
0	1...		DFTTADVF	NON QUER 327X 2 IS 3278/9
	.1..		DFTTEMUL	TSO EMULATION
	..11 1111		*	
1	C	1	*	
2	C	1	*	
3	C	1	*	
4	A	4	DFTTICBP	TSO PRINT UTILITY ICB
8	F	4	DFTTS99S	TSO SVC99 ALLOCATION SIZE
C	C	4	*	
10	C	8	DFTTTRDD	TSO TRACE DD NAME
18	C	8	DFTTDECK	TSO DECK O/P DD NAME
20	C	8	DFTTIESL	TSO IMD ADMGIMP DD NAME
28	C	8	DFTTPRNT	TSO PRINT DATASET QUALIFIER
30	C	8	DFTTSYSP	TSO SYSTEM PRINTER DD NAME
38	C	8	DFTTIADS	TSO IMD ADS DD NAME
40	C	8	DFTTIFMT	TSO IMD EXPORT DD NAME
48	C	8	DFTTMONO	TSO HRIG MONO DD OR DSNAME HLQ
50	C	8	DFTTCOLM	TSO HRIG COL DD OR DSNAME HLQ
58	C	8	DFTTDFDD	TSO DEFAULTS FILE DDNAME
60	C	8	DFTTS99U	TSO SVC99 UNIT SPECIFICATION
68	C	8	DFTTCPT	TSO CGM PROFILE DATASET QUALIFIER

ADMTDFTV - CMS-specific defaults (Offset defined by DFT0SYSO)

0	C	120	ADMTDFTV	CMS SPECIFIC DEFAULTS
	1...		DFTVADVF	NON QUER 328X IS 3287 TYPE
	.111 1111		*	
1	C	1	*	
2	C	1	*	
3	C	1	*	
4	C	4	*	
8	C	8	DFTVTMFT	CMS WORK FILE FILE-TYPE
10	C	8	DFTVDECK	CMS DECK O/P FILE-TYPE
18	C	8	DFTVPRNF	CMS PRINT FILE-TYPE
20	C	8	DFTVSYSP	CMS SYSTEM PRINTER FILE-TYPE
28	C	8	DFTVIADS	CMS IMD ADS FILE-TYPE
30	C	8	DFTVIFMT	CMS IMD EXPORT FILE-TYPE
38	C	8	DFTVMSLT	CMS IMD MSL FILE-TYPE
40	C	8	DFTVMONO	CMS HRIG MONO FILE-TYPE
48	C	8	DFTVCOLM	CMS HRIG COL FILE-TYPE
50	C	8	DFTVTRFN	CMS TRACE FILE-NAME
58	C	8	DFTVTRFT	CMS TRACE FILE-TYPE

Offset	Type	Length	Name	Description
60	C	8	DFTVDFFN	CMS DEFAULTS FILE-NAME
68	C	8	DFTVDFFT	CMS DEFAULTS FILE-TYPE
70	C	8	DFTVCPT	CMS CGM PROFILE DATASET QUALIFIER

ADMTDFTD - VSE-specific defaults (Offset defined by DFTOSYSO)

0	C	32	ADMTDFTD	VSE SPECIFIC DEFAULTS
0	C	8	DFTDTRDD	VSE TRACE D LABEL
8	C	8	DFTDMONO	VSE HRIG MONO D LABEL
10	C	8	DFTDCOLM	VSE HRIG COLOR D LABEL
18	C	8	DFTDDFDD	VSE DEFAULTS FILE D LABEL

ADMTDSA — dynamic save area (DSA)

Function

This control block contains a GDDM dynamic save area (DSA).

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDSA	0	DSALEN	2	DSASOSP	4C
DSAEPTR	50	DSANEXT	8	DSASTATE	48
DSAFLAGS	0	DSARSA	C	DSATRCE	48
DSAHDR	48	DSASOSEF	4C	DSAWORD1	0
DSALAST	4				

In the DSA extension (addressed from DSAEPTR):

Name	Offset	Name	Offset	Name	Offset
ADMTDSAE	0	DSAEEND	18	DSANEST	4
DSACNTRL	4	DSAERRPL	C	DSATRCSV	8
DSAEDSAE	0	DSAERRPP	10		

Storage layout

Offset	Type	Length	Name	Description
0		88	ADMTDSA	STANDARD DSA DECLARATION
0	C	4	DSAWORD1	
0	C	2	DSAFLAGS	RESERVED SET TO X'0000' FIRST BYTE IS COMMITTED TO ALWAYS REMAIN ZERO
2	F	2	DSALEN	ALLOCATED DSA LENGTH
4	A	4	DSALAST	A(LAST DSA)
8	A	4	DSANEXT	A(NEXT DSA)
C	A	60	DSARSA	REGISTER SAVE AREA
8	C	8	DSASTATE	STATE CONTROLS PROPAGATED DOWN VIA ADMMBEG MACRO
48	A	4	DSAHDR	A(DSA STACK HEADER)
	1...		DSATRCE	TRACE OPTIMIZATION BIT
	.111 1111		*	
4C	A	4	DSASOSP	A(SHORT ON STORAGE ELEMENT)
	1...		DSASOSEF	SOS EVENT IN PROGRESS
	.111 1111		*	
50	A	4	DSAEPTR	POINTER TO DSA EXTENSION
54	A	4	*	RESERVED
58	C	*	*	USER AREA

In the DSA extension (addressed from DSAEPTR):

Offset	Type	Length	Name	Description
0	C	*	ADMTDSAE	EXTENSION BLOCK
0	C	4	DSAEDSAE	ROOM FOR AN EYE CATCHER LATER
4	C	4	DSACNTRL	STANDARD DSA CONTROLS
4	F	1	DSANEST	DSA NESTING LEVEL
5	C	3	*	RESERVED
8	A	4	DSATRCSV	R14/R15 TRACE SAVE SLOT
C	A	4	DSAERRPL	ERROR STACK POINTER FOR CALLS
10	A	4	DSAERRPP	ERROR STACK POINTER FOR PROCESS
14	C	4	*	RESERVED
18	C	*	DSAEEND	END OF EXTENSION STACK

ADMTDSAO — dynamic save area overflow stack (DSAO)

Function

This control block contains the header of the GDDM dynamic save area overflow stack. This stack is used to supply "emergency" save areas to the DSA overflow processing code.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDSAO	0	DSASOENT	8	DSASOMN	10
DSASOACA	C	DSASOHDR	0	DSASOW1	0
DSASOBOS	4				

Storage layout

Offset	Type	Length	Name	Description
0	C	16	ADMTDSAO	DSA OVERFLOW STACK
0	C	16	DSASOHDR	DSA STACK HEADER
0	A	4	DSASOW1	RESERVED
4	A	4	DSASOBOS	A(BOTTOM OF STACK)
8	A	4	DSASOENT	A(OUT OF LINE PROLOG)
C	A	4	DSASOACA	A(AIC CONTROL AREA)
10	C	*	DSASOMN	ALLOCATION AREA

ADMTDSAS — dynamic save area stack (DSAS)

Function

This control block contains the header of a GDDM dynamic save area stack.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTDSAS	0	DSASENTE	8	DSASMAIN	10
DSASACAP	C	DSASHDR	0	DSASNXT	0
DSASBOS	4				

Storage layout

Offset	Type	Length	Name	Description
0	C	16	ADMTDSAS	DSA STACK
0	C	16	DSASHDR	DSA STACK HEADER
0	A	4	DSASNXT	CHAIN TO NEXT DSA STACK (0 IF LAST)
4	A	4	DSASBOS	A(BOTTOM OF STACK)
8	A	4	DSASENTE	A(OUT OF LINE PROLOG)
C	A	4	DSASACAP	A(AIC CONTROL AREA)
10	C	*	DSASMAIN	ALLOCATION AREA

ADMTIFCB — interface control block (IFCB)

Function

This is the interface control block (IFCB) which contains control information used throughout GDDM. Specifically, it contains the IFC response field, the interface control table (IFCT), and a subsystem-dependent subsystem save area.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTIFCB	0	IFCTGKSE	E0	IFCTPLT0	98
IFCBFC	0	IFCTGKSS	E4	IFCTPREV	5C
IFCBSSA	110	IFCTGKS0	E0	IFCTPRI	15
IFCTACOP	1C	IFCTGRA	15	IFCTPRVT	6C
IFCTAICE	70	IFCTHEAD	4	IFCTPTN	15
IFCTAICS	74	IFCTIMCE	E8	IFCTPTS	15
IFCTAIC0	70	IFCTIMCS	EC	IFCTPTXT	6C
IFCTA31C	14	IFCTIMC0	E8	IFCTPVSE	6D
IFCTA31F	14	IFCTIMDE	D0	IFCTROUT	15
IFCTBGSE	90	IFCTIMDS	D4	IFCTSCBP	8
IFCTBGSS	94	IFCTIMD0	D0	IFCTSOSE	20
IFCTBGS0	90	IFCTIMSE	C0	IFCTSPIP	24
IFCTCDUE	F0	IFCTIMSS	C4	IFCTSSMA	40
IFCTCDUS	F4	IFCTIMS0	C0	IFCTSSMT	3C
IFCTCDU0	F0	IFCTISEE	A0	IFCTSTOP	16
IFCTCOMP	44	IFCTISES	A4	IFCTTDF	10
IFCTCSIE	A8	IFCTISE0	A0	IFCTTDFA	10
IFCTCSIS	AC	IFCTIVUE	F8	IFCTTDFB	10
IFCTCSI0	A8	IFCTIVUS	FC	IFCTTDFD	10
IFCTDB22	16	IFCTIVU0	F8	IFCTTDFE	10
IFCTDDBG	48	IFCTMAXX	18	IFCTTDFI	10
IFCTDFDB	16	IFCTMODE	0	IFCTTDFK	11
IFCTDFTP	4C	IFCTMODS	4	IFCTTDFL	10
IFCTEDFP	50	IFCTMODX	0	IFCTTDFN	11
IFCTEFBP	28	IFCTNEXT	58	IFCTTDFO	11
IFCTERRA	2C	IFCTNL22	17	IFCTTDFP	10
IFCTERRL	34	IFCTNONN	14	IFCTTDFV	11
IFCTERRN	38	IFCTNSRE	C8	IFCTTDFY	11
IFCTERRP	30	IFCTNSRS	CC	IFCTTDF1	11
IFCTESIE	80	IFCTINSR0	C8	IFCTTDF3	11
IFCTESIS	84	IFCTOUTE	B8	IFCTTFRC	10
IFCTESI0	80	IFCTOUTS	BC	IFCTTLVL	13
IFCTFAPG	68	IFCTOUT0	B8	IFCTTQUL	12
IFCTFIGE	D8	IFCTPAG	15	IFCTTRCB	6C
IFCTFIGS	DC	IFCTPAPG	64	IFCTTRCE	10
IFCTFIG0	D8	IFCTPCLK	6D	IFCTTSID	78
IFCTFINT	14	IFCTPCGM	6D	IFCTTSIS	7C
IFCTFLGS	14	IFCTPCSF	6C	IFCTTSI0	78
IFCTFLG1	16	IFCTPDS	6C	IFCTUDFP	54
IFCTFPA	15	IFCTPDS0	6C	IFCTVERS	C
IFCTFSME	88	IFCTPGTP	60	IFCTVER1	C
IFCTFSMS	8C	IFCTPGDP	6C	IFCTVER2	E
IFCTFSM0	88	IFCTPHRG	6D	IFCTVSEE	B0
IFCTFTRM	14	IFCTPID	6C	IFCTVSES	B4
IFCTFWAP	14	IFCTPIM	6C	IFCTVSE0	B0
IFCTGEN	78	IFCTPLTE	98	IFCTWIN	15
IFCTGKER	14	IFCTPLTS	9C	IFCT31LE	14
				IFCT5080	6C

Storage layout

Offset	Type	Length	Name	Description
0	C	*	ADMTIFCB	INTERFACE CONTROL BLOCK.
0	F	2	IFCBFC	FEEDBACK CODE
2	F	2	*	RESERVED INTERFACE CONTROL TABLE
4	C	4	IFCTHEAD	'IFCT' EYE CATCHER
8	A	4	IFCTSCBP	A(STUB COMMUNICATIONS BLOCK
C	C	4	IFCTVERS	VERSION FIELDS
C	C	2	IFCTVER1	'RESIDENT CODE' VERSION
E	C	2	IFCTVER2	'TRANSIENT CODE' VERSION
10	F	4	IFCTTRCE	TRACE WORD
10	B	2	IFCTTDF	TRACE FLAGS
	1...		IFCTTFRC	1 = SUPPRESS ANY FURTHER TRACE WORD CHANGES COMPONENT DISABLEMENT FLAGS
	.1..		IFCTTDFA	1 = DISABLE AIC TRACE
	..1.		IFCTTDFL	1 = DISABLE TSI TRACE
	...1		IFCTTDFE	1 = DISABLE ESI TRACE
	.... 1...		IFCTTDFD	1 = DISABLE FSM TRACE
	.... .1..		IFCTTDFB	1 = DISABLE BGS TRACE
	.... ..1.		IFCTTDFP	1 = DISABLE PLT TRACE
	1		IFCTTDFI	1 = DISABLE ISE TRACE
	1...		IFCTTDFY	1 = DISABLE CSI TRACE
	.1..		IFCTTDFV	1 = DISABLE VSE TRACE
	..1.		IFCTTDFO	1 = DISABLE OUT TRACE
	...1		IFCTTDFK	1 = DISABLE IMS TRACE
	.... 1...		IFCTTDFN	1 = DISABLE NSR TRACE
	.... .1..		IFCTTDF1	1 = DISABLE IMD TRACE
	.... ..1.		*	RESERVED
	1		IFCTTDF3	1 = DISABLE IMC TRACE
12	F	1	IFCTTQUL	TRACE QUALIFIER
13	F	1	IFCTTLVL	TRACE LEVEL
14	B	1	IFCTFLGS	GENERAL PURPOSE FLAGS
	1...		IFCTFWAP	1 => WITHIN APPLICATION
	.1..		IFCTA31F	1 => GDDM INITIALIZED IN 31 BIT MODE
	..1.		IFCTA31C	1 => GDDM CALLED IN 31 BIT MODE
	...1		IFCT31LE	1 => 31 BIT LOAD ERROR HAS OCCURRED (ATTEMPT USE MODULE ABOVE 16MEG BUT GDDM INITIALIZED IN 24BIT MODE)
	.... 1...		IFCTNONN	1 => BYPASS NICKNAME PROCESSING
	.... .1..		IFCTFTRM	IMPLICIT FSTERM REQUIRED
	.... ..1.		IFCTFINT	IMPLICIT FSINIT PERFORMED
	1		IFCTGKER	GKS ERROR STATE
15	B	1	IFCTROUT	FSM ROUTING FLAGS:
	1...		IFCTPRI	PRIMARY DEVICE EXISTS
	.1..		IFCTPTS	PARTITION SET EXISTS
	..1.		IFCTPTN	PARTITION EXISTS
	...1		IFCTPAG	PAGE EXISTS
	.... 1...		IFCTWIN	WINDOW EXISTS
	.... .1..		*	RESERVED
	.... ..1.		IFCTGRA	GRAPHICS MAY FAST
	1		IFCTFPA	FAST PATH IS ACTIVE
16	B	1	IFCTFLG1	GENERAL PURPOSE FLAGS
	1...		IFCTDFDB	1 => GDDM HAS DETERMINED THAT THE 'DFT' TERMINAL CAN SUPPORT LINE O/P DBCS THROUGH SUBSYSTEM
	.1..		IFCTDB22	BASE NATIONAL LANGUAGE IS DBCS
	..1.		IFCTSTOP	STOP RECURSIVE ABENDS
	...1 1111		*	SPARE
17	C	1	IFCTNL22	BASE NATIONAL LANGUAGE
18	F	4	IFCTMAXX	CMS/XA STORE REQ. LIMIT
1C	A	4	IFCTACOP	A(AIC CONTROLLER)
20	A	4	IFCTSOSE	A(AIC SOS PROCESSOR)
24	A	4	IFCTSPIP	A(SPIB COPY)
28	A	4	IFCTEFBP	A(ERROR FEEDBACK BLOCK)

Offset	Type	Length	Name	Description
2C	A	4	IFCTERRA	A(ERROR LIST ANCHOR)
30	A	4	IFCTERRP	A(END OF ERROR LIST)
34	A	4	IFCTERRL	A(CODE OF LAST ERROR ADDED)
38	F	4	IFCTERRN	NUMBER OF ERRORS IN LIST
3C	F	4	IFCTSSMT	SMALL STORAGE SIZE THRESHOLD
40	A	4	IFCTSSMA	SMALL STORAGE POOL TABLE
44	A	4	IFCTCOMP	A(X INSTANCE COMM TABLE)
48	A	4	IFCTDDBG	DEVELOPMENT DEBUG ANCHOR
4C	A	4	IFCTDFTP	A(GENERAL DEFAULTS TABLE)
50	A	4	IFCTEDFP	A(DEFAULT DEFINITION MODULE)
54	A	4	IFCTUDFP	A(USER DEFAULTS MODULE)
58	A	4	IFCTNEXT	NEXT IFCB IN APG CHAIN
5C	A	4	IFCTPREV	PREVIOUS IFCB IN APG CHAIN
60	A	4	IFCTPDTP	INCORE PRODUCT DESCR TAB
64	A	4	IFCTPAPG	PARENT APPLICATION GROUP
68	A	4	IFCTFAPG	FIRST APPLICATION GROUP
6C	F	4	IFCTTRCB	PRIVATE TRACE BITS
	1...		IFCT5080	0 5080IO for PHIGS/5080
6C	B	31 bits	IFCTPRVT	TESTED BY TRACE
	.1..		IFCTPIM	1 IMAGE MANIPULATION
	..1.		IFCTPID	2 IMAGE DEVICES
	...1		IFCTPDS	3 DATASTREAM SYMBOL SETS
	.... 1...		IFCTPDSO	4 DSOPEN PARAMETER TRACING
	.... .1..		IFCTPGDP	5 FSM/GDP TRACING
	.... ..1.		IFCTPTXT	6 FSM/TEXT TRACING
	1		IFCTPCSF	7 COMPOSITE DOCUMENT STRUCTURED FIELDS
6D	1...		IFCTPHRG	8 HRIG DATA STREAM
	.1..		IFCTPVSE	9 VSE/BATCH TRACING
	..1.		IFCTPCLK	10 PCLINK
	...1		IFCTPCGM	11 CGM REPORT
	 1111		*	
6E	1111 1111		*	
6F	1111 1111		*	
70	C	8	IFCTAIC0	AIC RCPCPNT = 00
70	A	4	IFCTAICE	AIC FUNC PROCESSOR ENTRY POINT
74	A	4	IFCTAICS	AIC STORAGE ANCHOR
78	C	152	IFCTGEN	GENERAL MODULE DECLARATIONS
78	C	8	IFCTTSI0	TSI RCPCPNT = 04
78	C	1	IFCTTSID	TSI ENVIRONMENT IDENTIFIER
79	C	3	*	RESERVED
7C	A	4	IFCTTSIS	TSI STORAGE ANCHOR
80	C	8	IFCTESI0	ESI RCPCPNT = 08
80	A	4	IFCTESIE	ESI ENTRY POINT
84	A	4	IFCTESIS	ESI STORAGE ANCHOR
88	C	8	IFCTFSM0	FSM (GDDM) RCPCPNT = 0C
88	A	4	IFCTFSME	FSM ENTRY POINT
8C	A	4	IFCTFSMS	FSM STORAGE ANCHOR
90	C	8	IFCTBGS0	BGS (BUS/GRPHCS) RCPCPNT = 10
90	A	4	IFCTBGSE	ASC ENTRY POINT
94	A	4	IFCTBGSS	ASC STORAGE ANCHOR
98	C	8	IFCTPLT0	PLT (PLOT UTIL) RCPCPNT = 14
98	A	4	IFCTPLTE	PLT ENTRY POINT
9C	A	4	IFCTPLTS	PLT STORAGE ANCHOR
A0	C	8	IFCTISE0	ISE (IMAGE EDITOR) RCPCPNT = 18
A0	A	4	IFCTISEE	ISE ENTRY POINT
A4	A	4	IFCTISES	ISE STORAGE ANCHOR
A8	C	8	IFCTCSI0	CSI (MISC SERVICES) RCPCPNT = 1C
A8	A	4	IFCTCSIE	CSI ENTRY POINT
AC	A	4	IFCTCSIS	CSI STORAGE ANCHOR
B0	C	8	IFCTVSE0	VSE (VECTOR EDITOR) RCPCPNT = 20
B0	A	4	IFCTVSEE	VSE ENTRY POINT
B4	A	4	IFCTVSES	VSE STORAGE ANCHOR
B8	C	8	IFCTOUT0	OUT (PRINT) RCPCPNT = 24
B8	A	4	IFCTOUTE	OUT ENTRY POINT

Offset	Type	Length	Name	Description
BC	A	4	IFCTOUTS	OUT STORAGE ANCHOR
C0	C	8	IFCTIMS0	IMS (SUPERVISOR) RCPCPNT = 28
C0	A	4	IFCTIMSE	IMS ENTRY POINT
C4	A	4	IFCTIMSS	IMS STORAGE ANCHOR
C8	C	8	IFCTNSR0	NSR (NUM SUPPORT) RCPCPNT = 2C
C8	A	4	IFCTNSRE	NSR ENTRY POINT
CC	A	4	IFCTNSRS	NSR STORAGE ANCHOR
D0	C	8	IFCTIMD0	IMD (MAP EDITOR) RCPCPNT = 30
D0	A	4	IFCTIMDE	IMD ENTRY POINT
D4	A	4	IFCTIMDS	IMD STORAGE ANCHOR
D8	C	8	IFCTFIG0	PHIGS MAPPER RCPCPNT = 34
D8	A	4	IFCTFIGE	PHIGS MAPPER ENTRY POINT
DC	A	4	IFCTFIGS	PHIGS MAPPER STORAGE ANCHOR
E0	C	8	IFCTGKS0	GKS RCPCPNT = 38
E0	A	4	IFCTGKSE	GKS ENTRY POINT
E4	A	4	IFCTGKSS	GKS STORAGE ANCHOR
E8	C	8	IFCTIMC0	IMC RCPCPNT = 3C
E8	A	4	IFCTIMCE	IMC ENTRY POINT
EC	A	4	IFCTIMCS	IMC STORAGE ANCHOR
F0	C	8	IFCTCDU0	COMPOSITE DOCUMENT PRINT UTILITY RCPCPNT = 40
F0	A	4	IFCTCDUE	CDPU ENTRY POINT
F4	A	4	IFCTCDUS	CDPU STORAGE ANCHOR
F8	C	8	IFCTIVU0	IMAGE VIEW UTILITY RCPCPNT = 44
F8	A	4	IFCTIVUE	IVU ENTRY POINT
FC	A	4	IFCTIVUS	IVU STORAGE ANCHOR
100	C	8	*	RESERVED RCPCPNT = 48
100	A	4	*	
104	A	4	*	
108	C	8	*	RESERVED RCPCPNT = 4C
108	A	4	*	
10C	A	4	*	
110	C	*	IFCBSSA	SUBSYSTEM SAVE AREA BASED ENTRY DECLARATIONS
0	C	8	IFCTMODX	(15)
0	A	4	IFCTMODE	GENERAL MODULE ENTRY POINT
4	A	4	IFCTMODS	GENERAL MODULE STORAGE ANCHOR

ADMTIFCE — GDDM interface control error feedback

Function

This control block contains error feedback fields, providing ancillary information relating to the response code in the Interface Control Block (IFCB). Refer to "ADMTIFCB — interface control block (IFCB)" on page 142. The IFCE is normally addressed via a pointer in the Interface Control Table (IFCT).

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTIFCE	0	IFCEERRK	68	IFCES1	1C
IFCEA1	C	IFCEEXTN	4	IFCES2	26
IFCEA2	10	IFCEFC	4	IFCES3	30
IFCEA3	14	IFCEFLGS	8	IFCES4	3A
IFCEA4	18	IFCEL2	1C	IFCETXT	C
IFCEBLDF	8	IFCEPREF	8	IFCET1	14
IFCEC1	1C	IFCEPRET	9	IFCET2	18
IFCEC2	30	IFCEQQ	0	IFCEUSRF	8

Storage layout

Offset	Type	Length	Name	Description
0		108	ADMTIFCE	INTERFACE ERROR FEEDBACK BLOCK
0	B	4	IFCEQQ	REQUEST CODE
4		8	IFCEEXTN	EXTENDED ERROR INDICATORS (The following fields are meaningful only if IFCBFC MSG NO.=X'FFF'.)
4	F	4	IFCEFC	EXTENDED ERROR MSG NO.
8	B	1	IFCEFLGS	FLAG BYTE
	1...		IFCEBLDF	TEXT ALREADY BUILT
	.1..		IFCEPREF	NON-GDDM PREFIX REQUIRED (1 = > NON-GDDM)
	..1.		IFCEUSRF	NON-GDDM/IMD MSG REQUIRED (1 = > NON-GDDM and NON-IMD)
	...1 1111			SPARE
9	C	3	IFCEPRET	REQUIRED NON-GDDM PREFIX
C	C	68	IFCETXT	MSG TEXT EXCLUDING PREFIX, MSG NUMBER, AND SEVERITY, OR, ENCODED MESSAGE INSERTS
C	F	4	IFCEA1	ARITH INSERT FIELD 1
10	F	4	IFCEA2	ARITH INSERT FIELD 2
14	F	4	IFCEA3	ARITH INSERT FIELD 3
14	F	4	IFCET1	TEXT INSERT FIELD 1
18	F	4	IFCEA4	ARITH INSERT FIELD 4
18	F	4	IFCET2	TEXT INSERT FIELD 2
1C	C	40	IFCEL2	CHARACTER INSERT LONG FIELD
1C	C	20	IFCEC1	CHARACTER INSERT FIELD
1C	C	10	IFCES1	SHORT CHARACTER INSERT FIELD 1
26	C	10	IFCES2	SHORT CHARACTER INSERT FIELD 2
30	C	20	IFCEC2	CHARACTER INSERT FIELD 2
30	C	10	IFCES3	SHORT CHARACTER INSERT FIELD 3
3A	C	10	IFCES4	SHORT CHARACTER INSERT FIELD 4
44	C	12	*	SPARE
50	C	24	*	SPARE
68	F	4	IFCEERRK	KEY FOR ERROR MESSAGE SERVICE

ADMTNICK — nickname list entry format

Function

This control block defines the structure of a nickname list entry. The nickname lists are chained from field DFTONICK in the general defaults table; see page 133.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTNICK	0	NICKHEAD	0	NICKSNMC	24
NICKAPPF	10	NICKLENG	4	NICKSNMP	14
NICKBPTR	C	NICKPRCC	38	NICKTFAM	28
NICKDVTK	30	NICKPRCP	1C	NICKTNMC	2C
NICKFLGS	10	NICKSCNF	10	NICKTNMP	18
NICKFPTR	8	NICKSFAM	20		

Storage layout

Offset	Type	Length	Name	Description
0	C	*	ADMTNICK	NICKNAME LIST ENTRY
0	C	4	NICKHEAD	'NICK' EYE CATCHER
4	F	4	NICKLENG	LENGTH OF ENTRY
8	C	8	*	
8	A	4	NICKFPTR	FORWARD CHAIN POINTER
C	A	4	NICKBPTR	BACK CHAIN POINTER
0	C	4	NICKFLGS	FLAGS
	1...		NICKSCNF	USED BY FSM FOR SCAN CONTROL
	.1..		NICKAPPF	0 -> REPLACE OPTION SPECIFIED 1 -> APPEND OPTION (DEFAULT)
14	C	12	*	
14	A	4	NICKSNMP	ADDR (SOURCE NAME DATA)
18	A	4	NICKTNMP	ADDR (TARGET NAME DATA)
1C	A	4	NICKPRCP	ADDR (PROCOPT DATA)
20	C	28	*	
20	F	4	NICKSFAM	SOURCE FAMILY
24	F	4	NICKSNMC	COUNT OF SOURCE NAME DWORDS
28	F	4	NICKTFAM	TARGET FAMILY
2C	F	4	NICKTNMC	COUNT OF TARGET NAME DWORDS
30	C	8	NICKDVTK	DEVICE TOKEN
38	F	4	NICKPRCC	COUNT OF PROCOPT FWORDS
3C	C	*	*	VARIABLE LENGTH DATA

ADMTTCA — Terminal services interface control area (TCA)

Function

This control block contains the addresses, data areas,and flags to be retained by the terminal service interface area (TSIA) across invocations. One instance of this control block occurs in each instance of the TSIA.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTTCA	0	TCALASTP	C	TCASTFIP	14
TCAACTVF	10	TCANEXTP	8	TCASTFOP	14
TCADUMLF	10	TCANOTF	10	TCASTFRL	14
TCADVTOK	1C	TCAQRYBF	10	TCASTFSL	14
TCAERRFC	16	TCAROPNF	10	TCASTFTL	14
TCAFLGCS	10	TCARROUP	18	TCASTGL	4
TCAFLGST	14	TCARRSVF	10	TCASYSL	28
TCAF2NCC	10	TCASSIA	0	TCASYSP	24
TCAID	0	TCASTFBS	14		

Storage layout

Offset	Type	Length	Name	Description
0	C	164	ADMTTCA	TSIA CONTROL AREA
0	C	84	TCASSIA	HEADER
0	C	4	TCAID	EYE-CATCHER TCA
4	F	4	TCASTGL	LENGTH OF ACQUIRED STORAGE
8	A	4	TCANEXTP	ADDRESS (NEXT TCA IN THE RING)
C	A	4	TCALASTP	ADDRESS (LAST TCA IN THE RING)
10	C	4	TCAFLGCS	CURRENT STATUS FLAGS
	1...		TCANOTF	1 => NOT FIRST NOTABLE I/O
	.1..		TCARRSVF	1 => RESOURCE RESERVED VIA CSI
	..1.		TCAACTVF	1 => TSIA ACTIVE
	...1		TCAROPNF	1 => RESOURCE IS OPEN
	.... 1...		TCADUMLF	1 => DUMMY ADAPTER IS LOADED
	.... .1..		TCAQRYBF	1 => I/P BUFFER IS QUERY REPLY
	.... ..1.		TCAF2NCC	1 => FAMILY 2 I/P WITH NOSPAN, NOCC
	1		*	RESERVED
	1111 1111		*	RESERVED
	1111 1111		*	RESERVED
	1111 1111		*	RESERVED
14	C	2	TCAFLGST	STATE FLAGS - VALID NEXT OPS
	1...		TCASTFBS	BASE FUNCTIONS, INCLUDES TQRY, RNIT, TERM, ACQ, O/P BUF
	.1..		TCASTFOP	OUTPUT OPS (SEND/CONVERSE)
	..1.		TCASTFIP	INPUT OPS (RECEIVE)
	...1		TCASTFRL	RELEASE INPUT BUFFER
	.... 1...		TCASTFTL	NOT FIRST/ONLY TMIT
	.... .1..		TCASTFSL	NOT FIRST/ONLY SEGMENT
	11		*	RESERVED
	1111 1111		*	RESERVED
16	F	2	TCAERRFC	INTERNAL IFCBFC-TYPE ERROR CODE
18	A	4	TCARROUP	IF DUMMY DEVICE, A(REAL ROUTER)
1C	C	8	TCADVTOK	DEVICE TOKEN
24	A	4	TCASYSP	ADDRRESS(TCA SYSOPT-LIST)
28	F	4	TCASYSL	LENGTH(TCA SYSOPT-LIST)
2C	C	20	*	SPARE FIELDS
40	C	56	ADMTTRB	TERMINAL REQUEST BLOCK
78	C	44	ADMTTDB	TERMINAL DESCRIPTOR BLOCK

ADMTTDB — terminal descriptor block (TDB)

Function

This control block contains fields and flags that describe the characteristics of the terminal. It includes the terminal type, its modes of operation, screen sizes, and link protocol.

It is a part of the terminal services interface control area (TCA). All offsets defined below are from the start of the TCA.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTRDBE	A4	TDBGAKAT	78	TDBREST	7B
ADMTTDB	78	TDBGAMON	78	TDBSCSBL	7A
TDBAPKB	7A	TDBHWS	7E	TDBSCSFB	7A
TDBAPLF	7A	TDBINPT	7C	TDBSDLBS	7D
TDBAPTU	7A	TDBINPUT	7C	TDBSDLC	7D
TDBAPTX	7A	TDBIO	7C	TDBSVBSZ	8A
TDBASCDR	7C	TDBIPDS	98	TDBSZ	7E
TDBASZ	82	TDBIRPI	9A	TBDSZAL	79
TDBASZC	84	TDBISTM	98	TDBSZF	79
TDBASZR	82	TDBLCLRM	7D	TDBSZMD	79
TDBAUXON	78	TDBLINKF	7D	TDBTEKGL	7C
TDBBSC	7D	TDBLOCAL	7D	TDBTEKPX	90
TDBCOLMP	94	TDBLU1P	78	TDBTEKPY	92
TDBCURBO	7B	TDBLU3P	78	TDBTRBSZ	8C
TDBDECGL	7C	TDBMISCF	7B	TDBTXKB	7A
TDBDEVAD	9C	TDBMOVEF	7B	TDBTXPR	7A
TDBDFDB	78	TDBMPAGE	86	TDBTYPE	78
TDBDUMMY	7C	TDBMPCOL	88	TDBUNF	78
TDBFLAGS	78	TDBMPROW	86	TDBUSESF	7B
TDBFSSAV	8A	TDBMSEG	7B	TDBWSF	7B
TDBF4DEP	86	TDBMXMT	7B	TDB3270E	7B
TDBF4LNW	8A	TDBOUTPT	7C	TDB3277	78
TDBF4PIX	90	TDBPSZ	7E	TDB3278	78
TDBF4PIY	92	TDBPSZC	80	TDB38XX	78
TDBF4UNI	8E	TDBPSZR	7E	TDB3800	78
TDBF4WID	88	TDBREMOT	7D	TDB4250	78

Storage layout

Offset	Type	Length	Name	Description
78	C	44	ADMTTDB	TERMINAL DESCRIPTOR BLOCK
78	B	6	TDBFLAGS	
78	B	1	TDBTYPE	TERMINAL TYPE FLAGS
	1...		TDB3277	OLD TYPE (3271,72,77,84,88)
	1...		TDB3800	3800 (SONORA) PRINTER TYPE
	.1..		TDB3278	NEW TYPE (3276,78,87,89)
	.1..		TDB4250	4250 (BREG) PRINTER TYPE
	..1.		TDBLU1P	DEVICE IS LU1 (SCS) PRINTER
	..1.		TDB38XX	38XX ( ) PRINTER TYPE
	...1		TDBLU3P	DEVICE IS LU3 PRINTER
	...1		TDBUNF	UNFORMATTED HRIG OUTPUT
	.... 1...		TDBGAMON	GRAPHICS ATTACHMENT PRINTER
	.... .1..		TDBGAKAT	GRAPHICS ATTACHMENT (KATAKANA) (SET IN ADDITION TO TDBGAMON)
	.... ..1.		TDBAUXON	AUXILIARY ONLY DEVICE
	1		TDBDFDB	1 = > THIS IS 'DFT' TERMINAL ON SUBSYSTEM THAT SUPPORTS LINE O/P DBCS
79	B	1	TDBSZF	TERMINAL SIZE FLAGS
	1...		TDBSZMD	SCREEN SIZE IS MODIFIABLE
	.1..		TDBSZAL	CURRENT SCREEN IF TDBSZMD = 0 1 = > ALTERNATE, 0 = > PRIMARY
	..11 1111		*	RESERVED
7A	B	1	TDBAPLF	APL/TEXT FLAGS
	1...		TDBAPTU	APL TEXT FEA STATUS UNKNOWN
	.1..		TDBAPTX	APL TEXT FEA PRESENT
	..1.		TDBTXKB	TEXT KEYBOARD KNOWN
	...1		TDBAPKB	APL KEYBOARD KNOWN
	.... 1...		TDBTXPR	TEXTPRINT FEATURE ON
	.... .1..		TDBSCSFB	0 = SCS BASE CODES (NL, FF) SUPPORTED (LU 1) 1 = SCS FULL BASE CODES (NL, FF, BS, CR, LF) SUPPORTED (LU 1)
	.... ..1.		TDBSCSBL	1 = SCS BEL CODE SUPPORTED (LU 1)
	1		*	RESERVED
7B	B	1	TDBMISCF	MISCELLANEOUS FLAGS
	1...		TDBCURBO	CURSOR FIELD RETURN TYPE
	.1..		TDBMSEG	0=SINGLE SEGM,1=MULTI SEGM XMIT
	..1.		TDBMXMT	0=SINGLE XMIT,1=MULTI XMIT XMIT
	...1		TDBREST	1=PERISHABLE PS STORES
	.... 1...		TDB3270E	0=DO NOT USE 3270E OUTGOING
	.... .1..		TDBWSF	1=THIS BOX SUPPORTS WSF (LU2/LU3) OR FM1 FOR LU 1
	.... ..1.		TDBUSESF	1=ALWAYS USE STRUCTURED FIELDS IF SUPPORTED BY BOX
	1		TDBMOVEF	MODE OF ADAPTER OPERATION
7C	B	1	TDBIO	I/O FLAGS
	1...		TDBINPT	ADAPTER WILL RETURN INPUT
	.1..		TDBOUTPT	ADAPTER WILL ACCEPT OUTPUT
	..1.		TDBDUMMY	DEVICE IS A DUMMY DEVICE
	...1		TDBINPUT	INPUT PATH TO DEV AVAILABLE
	.... 1...		TDBDECGL	ASCII GRAPHICS DEC DEVICE
	.... .1..		TDBTEKGL	ASCII GRAPHICS TEK DEVICE
	11		TDBASCDR	ASCII GRAPHICS DEVICE LEVEL
7D	B	1	TDBLINKF	TYPE OF COMMUNICATIONS LINK
	11..		TDBLCLRM	THESE BITS HAVE VALUES: (11 = INVALID 10 = DEVICE IS LOCAL 01 = DEVICE IS REMOTE 00 = TYPE IS UNKNOWN)
	1...		TDBLOCAL	DEVICE IS LOCALLY ATT

Offset	Type	Length	Name	Description
	.1..		TDBREMOT	DEVICE IS REMOTELY AT
	..11		TDBSDLBS	THESE BITS HAVE VALUES: (11 = INVALID 10 = LINK IS SDLC 01 = LINK IS BSC 00 = TYPE IS UNKNOWN)
	..1.		TDBSDLC	LINK IS SDLC
	...1		TDBBSC	LINK IS BI SYNC
	 1111		*	RESERVED
7E	C	22	TDBHWS	
7E	C	8	TDBSZ	SCREEN SIZES
7E	C	4	TDBPSZ	PRIMARY SCREEN SIZES
7E	F	2	TDBPSZR	PRIMARY SCREEN SIZE (ROW)
80	F	2	TDBPSZC	PRIMARY SCREEN SIZE (COLS)
82	C	4	TDBASZ	ALTERNATE SCREEN SIZES
82	F	2	TDBASZR	ALTERNATE SCREEN SIZE (ROWS)
84	F	2	TDBASZC	ALTERNATE SCREEN SIZE (COLS)
86	C	4	TDBMPAGE	MAXIMUM PAGE SIZE
86	F	2	TDBMPROW	MAXIMUM NO. OF ROWS
86	F	2	TDBF4DEP	FAMILY 4 MEDIUM DEPTH
88	F	2	TDBMPCOL	MAXIMUM NO. OF COLUMNS
88	F	2	TDBF4WID	FAMILY 4 MEDIUM WIDTH
8A	C	2	TDBFSSAV	
8A	F	2	TDBSVBSZ	FSSAVE BUFFER SIZE
8A	F	2	TDBF4LNW	FAMILY 4 PELS/UNIT LINEWIDTH
8C	F	2	TDBTRBSZ	TRANSMISSION BUFFER SIZE
8E	F	2	TDBF4UNI	FAMILY 4 MEDIUM UNITS
90	F	2	TDBF4PIX	FAMILY 4 PELS PER INCH X
90	F	2	TDBTEKPX	ASCII GRAPHICS SCREEN WIDTH
92	F	2	TDBF4PIY	FAMILY 4 PELS PER INCH Y
92	F	2	TDBTEKPY	ASCII GRAPHICS SCREEN DEPTH
94	A	4	TDBCOLMP	TO COLOR MASTER ENTRY
98	C	4	TDBIPDS	
98	F	2	TDBISTM	LENGTH OF STM QUERY REPLY
9A	F	2	TDBIRPI	LENGTH OF RPI QUERY REPLY
9C	F	4	TDBDEVAD	DEVICE ADDRESS
A0	F	4	*	SPARE FIELDS

ADMTTRB — terminal request block (TRB)

Function

This control block contains fields and flags for handling I/O operations with the device. It contains various buffer addresses and lengths, and fields that show the state of the I/O.

It is a part of the Terminal Services Interface Control Area (TCA). All offsets defined below are from the start of the TCA.

Alphabetic index to field names

Name	Offset	Name	Offset	Name	Offset
ADMTTRB	40	TRBIMIDS	5C	TRBOQURY	5A
ADMTTRBE	78	TRBIMIDX	5C	TRBORDIM	59
TRBCMPNT	60	TRBINL	54	TRBORQNN	5B
TRBEFLG1	64	TRBINLH	56	TRBORQPS	5B
TRBEFLG2	65	TRBINP	50	TRBORQRS	5B
TRBEFLG3	66	TRBLROUP	40	TRBORQSC	5B
TRBEFLG4	67	TRBOCD	59	TRBOSTNN	5B
TRBERROR	64	TRBOCDOB	59	TRBOSTPS	5B
TRBERS	64	TRBODEV	5A	TRBOSTRS	5B
TRBERSIO	64	TRBODR	59	TRBOSTSC	5B
TRBERSPS	64	TRBOEB	59	TRBOSYS	5A
TRBERSSC	64	TRBOFLG1	58	TRBOTDB	5A
TRBERS1	64	TRBOFLG2	59	TRBOUTL	4C
TRBICD	5D	TRBOFLG3	5A	TRBOUTLH	4E
TRBIEB	5D	TRBOFLG4	5B	TRBOUTP	48
TRBIFLG1	5C	TRBOFMH	59	TRBRQEST	58
TRBIFLG2	5D	TRBOFMHI	59	TRBRSPND	5C
TRBIFLG3	5E	TRBOFSTS	58	TRBSESS1	68
TRBIFLG4	5F	TRBOFSTX	58	TRBSRCV1	68
TRBIFMH	5D	TRBOLSTS	58	TRBSRFL1	68
TRBIFSTS	5C	TRBOLSTX	58	TRBTCAP	44
TRBIFSTX	5C	TRBOMIDS	58	TRBUACK	6C
TRBIFSTS	5C	TRBOMIDX	58	TRBUACKL	6C
TRBILSTX	5C	TRBOPART	59	TRBUACKP	70

Storage layout

Offset	Type	Length	Name	Description
40	C	56	ADMTTRB	TERMINAL REQUEST BLOCK
40	A	4	TRBLROUP	ADDRESS OF TSI ROUTER MODULE
44	A	4	TRBTCAP	ADDRESS OF TSI ANCHOR BLOCK
48	A	4	TRBOUTP	ADDRESS OF OUTPUT BUFFER
4C	F	4	TRBOUTL	LENGTH OF OUTPUT SEGMENT
4C	F	2	*	FIRST H/WORD
4E	F	2	TRBOUTLH	LENGTH AS H/WORD (FOR CICS)
50	A	4	TRBINP	ADDRESS OF INPUT SEGMENT
54	F	4	TRBINL	LENGTH OF INPUT SEGMENT
54	F	2	*	FIRST H/WORD
56	F	2	TRBINLH	LENGTH AS H/WORD (FOR CICS)
58	C	4	TRBRQEST	TRB REQUEST FLAGS FOR CALL OPTIONS
58	B	1	TRBOFLG1	SEGMENT AND TRANSMIT SEQUENCE
	1...		TRBOFSTX	TRANSMIT(FIRST)
	.1..		TRBOMIDX	TRANSMIT(MIDDLE)
	..1.		TRBOLSTX	TRANSMIT(LAST)
	...1		*	RESERVED
	.... 1...		TRBOFSTS	SEGMENT(FIRST)
	.... .1..		TRBOMIDS	SEGMENT(MIDDLE)

Offset	Type	Length	Name	Description
	.... ..1.		TRBOLSTS	SEGMENT(LAST)
	1		*	RESERVED
59	B	1	TRBOFLG2	SESSION CONTROL, FMH AND WAIT CODE
	1...		TRBOFMH	FMH REQUIRED
	.1..		TRBOCD	RETURN FLOW WITH CHANGE DIRECTION (I.E. TRANSMISSION IS PART OF A CONVERSE CALL)
	..1.		TRBOEB	RETURN FLOW WITH END BRACKET
	...1		TRBORDIM	RETURN IMMEDIATELY IF NO IN
	.... 1...		TRBOCDOB	RETURN FLOW WITH CHANGE DIRECTION ON OUTBOUND TRANSMISSION (AS ARCHITECTURALLY REQUIRED BY OUTBOUND DATA STREAMS CONTAINING "READ" OR "QUERY" COMMANDS)
	.... .1..		TRBOPART	SCREEN IS BEING OPERATED USING PARTITION(S)
	.... ..1.		TRBOFMHI	FMH ALREADY INCLUDED
	1		TRBODR	SNA DEFINITE RESPONSE
5A	B	1	TRBOFLG3	ADMMTQRY OPTION FLAGS
	1...		TRBOQURY	RETURN QUERY REPLY
	.1..		TRBOTDB	RETURN TERMINAL DESCRIPTOR BLOCK
	..1.		TRBODEV	RETURN DEVICE OPTIONS
	...1		TRBOSYS	RETURN SUBSYSTEM OPTIONS
	 1111		*	RESERVED
5B	B	1	TRBOFLG4	RESOURCE USAGE FLAGS
	1111		TRBORQRS	RESOURCES REQUIRED BY CURRENT TRANSMISSION. '0000'B IMPLIES DEFAULT OF (SCREEN + PS)
	1...		TRBORQSC	SCREEN BUFFER REQUIRED
	.1..		TRBORQPS	PS BUFFER REQUIRED
	..1.		*	RESERVED
	...1		TRBORQNN	INHIBITS DFT INFERENCE
	 1111		TRBOSTRS	RESOURCES CHANGED BY CURRENT TRANSMISSION. '0000'B IMPLIES DEFAULT AS FOLLOWS, EW/EWA/W ALL EXCEPT PS WSF ALL
	.... 1...		TRBOSTSC	SCREEN BUFFER CHANGED
	.... .1..		TRBOSTPS	PS BUFFER CHANGED
	.... ..1.		*	RESERVED
	1		TRBOSTNN	INHIBITS DFT INFERENCE
5C	C	4	TRBRSPND	TRB RESPONSE FLAGS FROM ADAPTER
5C	B	1	TRBIFLG1	SEGMENT AND TRANSMIT SEQUENCE
	1...		TRBIFSTX	TRANSMIT(FIRST)
	.1..		TRBIMIDX	TRANSMIT(MIDDLE)
	..1.		TRBILSTX	TRANSMIT(LAST)
	...1		*	RESERVED
	.... 1...		TRBIFSTS	SEGMENT(FIRST)
	.... .1..		TRBIMIDS	SEGMENT(MIDDLE)
	.... ..1.		TRBILSTS	SEGMENT(LAST)
	1		*	RESERVED
5D	B	1	TRBIFLG2	SESSION STATE AND FMH PRESENT
	1...		TRBIFMH	FMH PRESENT IN DATA
	.1..		TRBICD	RETURNED FLOW WITH CHANGE DIRECTION
	..1.		TRBIEB	RETURNED FLOW WITH END BRACKET
	...1 1111		*	RESERVED
5E	B	1	TRBIFLG3	RESERVED
5F	B	1	TRBIFLG4	RESERVED
60	F	4	TRBCMPNT	COMPONENT NDX TO FILE SET
64	C	4	TRBERROR	TRB ERROR FLAGS FROM ADAPTER
64	B	1	TRBEFLG1	REQUIRED UNAVAILABLE RESOURCES FLAGS. IF THERE IS A RESHOW ERROR, THESE FLAGS IDENTIFY WHICH RESOURCES NEED RESHOW.
	1111		TRBERS	
	1...		TRBERSSC	SCREEN BUFFER
	.1..		TRBERSPS	PS BUFFER
	..1.		*	RESERVED
	...1		TRBERSIO	LAST TRANSMISSION CONTENTS
	 1111		TRBERS1	RESERVED
65	B	1	TRBEFLG2	RESERVED

Offset	Type	Length	Name	Description
66	B	1	TRBEFLG3	RESERVED
67	B	1	TRBEFLG4	RESERVED
68	C	1	TRBSESS1	FURTHER SESSION CONTROL FLAG
68	B	1	TRBSRFL1	RECEIVE PROCESSING FLAGS
	1...		TRBSRCV1	FIRST RECEIVE PROCESSED (CURRENTLY USED ONLY FOR CICS PSEUDO CONV)
	.111 1111		*	RESERVED
69	C	3	*	SPARE
6C	C	8	TRBUACK	UNEXPECTED ACKNOWLEDGEMENT
6C	F	2	TRBUACKL	ACK BUFFER LENGTH
6E	C	2	*	RESERVED
70	A	4	TRBUACKP	-> FSM ACK BUFFER
74	C	4	*	FURTHER SPARE FIELDS
78	C	0	ADMTTRBE	END OF REQUEST BLOCK

Appendix B. Abend codes

This appendix lists in numeric order the abend codes produced by GDDM Base, GDDM-PGF, GDDM Interactive Map Definition (GDDM-IMD), GDDM Image View Utility (GDDM-IVU), GDDM-GKS, GDDM-REXX, GDDM-PCLK, and GDDM-CSPF.

Each abend code is listed with information you may find helpful when looking for the cause of the abend. A code for the component from which the abend was issued, is given in the "Component" column; these codes are explained on page 55. GDDM-CSPF is the exception to this. You will sometimes find "CSPF" in the "Component" column, but it is not a code like those listed on page 55.

In some instances, the abend is preceded by the following message:

```
ADM0309 U GDDM ABEND n AT OFFSET X'xxxx' IN 'a'
```

where:

n is the abend code.
X'xxxx' is the offset of the location in the failing module where the abend was raised.
a is the module eye-catcher, which consists of the module name, compile date, GDDM version, and (possibly) service level.

For all GDDM abends, general-purpose register 2 (GPR02) is set to the address of the interface control block (IFCB). IFCTEFBP in the IFCB is set to the address of the error feedback block (IFCE). Within the IFCE fields, IFCEA1 is set to the abend code, IFCEA2 is set to the offset, and IFCEL2 is set to the eye-catcher.

On CICS, the first digit of the abend code is replaced by a "G" to identify the abend as originating from GDDM.

On IMS/VS, the codes listed have 1000 added before issuing an abend, to avoid conflict with standard IMS abend codes.

Thus, listed abend code 1163 will appear as "G163" on CICS, and as 2163 on IMS/VS.

On VM/CMS, the abend code may appear in hexadecimal if the issuing module is in a discontiguous shared segment.

On VSE, the abend code appears in register 5 when a dump has been requested. It also appears as the return code from the job step.

GDDM-IVU abends appear in register 0 and result from the following main causes:

- An error record with severity greater than zero returned from GDDM Base.
- Bad return codes from the GDDM-IVU input data converter (ADM5CV).
- Bad return codes from the GDDM-IVU default value inserter (ADM5DF).
- Bad return codes from the GDDM-IVU name handler (ADM5NM).
- Bad return codes from the GDDM-IVU screen handler (ADM5SH).
- Errors detected during GDDM-IVU storage acquisition and release.

Other causes of GDDM-IVU abends are listed by the relevant module.

Abend Dec	Abend Hex	Component	Issuing Module	Description
1000	3E8	ESI	ADM000C	Error log records present
1050	41A	AIC	ADMACFP ADMACUP	Invalid request code
1051	41B	AIC	ADMACLP	Abend on FSTERM requested
1053	41D	AIC	ADME000I	AIC-detected FSINIT error
1054	41E	AIC	ADME000I	AIC-detected non-FSINIT error
1058	422	AIC	ADMACSSP	Invalid call to ADMACSSP
1059	423	AIC	ADMACDP ADMACSSP	SOS stack overflow
1060	424	AIC	ADMACDP	Overflow stack overflow
1062	426	AIC	ADMACUPS	Local EUDS area too small
1063	427	AIC	ADMACUP ADMACUPS ADMATPE	ADMEODFnn entry wrong
1064	428	AIC	ADMACLP	Product information unavailable
1100	44C	FSM	ADMDSRO ADMD3SXF	Invalid request code
1101	44D	FSM	ADMDSRO	Invalid FCA pointer
1102	44E	FSM	ADMDSRO	Not FSM component code
1103	44F	FSM	ADMDSLD	Invalid module group
1104	450	FSM	ADMDSLD	Invalid subcomponent code for module group
1105	451	FSM	ADMDSLD	No DST address in DCA
1106	452	FSM	ADMDSDS	Invalid PROCOPT list
1107	453	FSM	ADMDSDS	All device IDs in use
1108	454	FSM	ADMDSF1 ADMDSOO	Invalid group length
1109	455	FSM	ADMDSOO	Table size exceeded
1110	456	FSM	ADMDCDS	Unsupported RCP code
1111	457	FSM	ADMDCDS	Zero pointer detected
1112	458	FSM	ADMDCPC	No current page
1114	45A	FSM	ADMDCGS	No SST entry for pattern or marker set (GSLSS or GSDSS)
1115	45B	FSM	ADMDCCD	Invalid conversion type
1120	460	FSM	ADMDSOO	Invalid default POPCODE
1121	461	FSM	ADMDSOO	Invalid code parameter
1122	462	FSM	ADMDSOO	Invalid system options
1123	463	FSM	ADMDSOO	Input option length > default
1124	464	FSM	ADMDSMC	Invalid request code
1125	465	FSM	ADMDSMC	Invalid control blocks
1126	466	FSM	ADMDSOO	Defaults definition error
1127	467	FSM	ADMDSF2 ADMDSF3 ADMDSF4	PROCOPT code error
1128	468	FSM	ADMDSA1 ADMDSF1 ADMDSF2 ADMDSF3 ADMDSF4 ADMDSTQ	PROCOPT number error
1129	469	FSM	ADMDOOC ADMDSPQ	No print control option
1130	46A	FSM	ADMDOAL	ASQFLD error on primary device
1133	46D	FSM	ADMDOCO	No primary device
1139	473	FSM	ADMDOBS	Invalid code passed to ADMDOBS
1150	47E	FSM	ADMDGAR ADMDTSHA	Invalid system pattern
1151	47F	FSM	ADMDGS1 ADMDGS3 ADMDGS4 ADMDGS6	Temporary buffer overflow
1153	481	FSM	ADMDGREG	Error during regenerate
1154	482	FSM	ADMDGI0 ADMDGI1 ADMDGI3 ADMDGI4 ADMDGI5 ADMDGI6	Invalid system symbol set
1155	483	FSM	ADMDGSE	Invalid pointer SEG delete

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1156	484	FSM	ADMDGI1 ADMDGI4 ADMDGI5 ADMDGI6	Invalid family 1 symbol set
1157	485	FSM	ADMDSQP ADMDSYMT ADMDWUFL	Error in symbol set load
1158	486	FSM	ADMDGI1 ADMDGI3 ADMDGI4 ADMDGI5 ADMDGI6	Invalid 'function' code
1159	487	FSM	ADMDGPCX	Default symbol set not found
1160	488	FSM	ADMDTCON ADMDTPGD	IEEE488 or BP1 error response to transmission of plotter data stream
1161	489	FSM	ADMDDIMF ADMDDIMG	Invalid parameter
1162	48A	FSM	ADMDDIMG	Invalid projection (Place_Pos Mix Mode) for conversion to IOCA
1163	48B	FSM	ADMDDUFL	Invalid load type
1164	48C	FSM	ADMDDUFA ADMDWUFL ADMDXUFI ADMDXUFM	Invalid (that is, 0) pointer
1165	48D	FSM	ADMDXUFM	Invalid function used
1169	491	FSM	ADMDDSRD	AFT does not belong to any current page
1170	492	FSM	ADMDBFN1	Invalid RCP code
1171	493	FSM	ADMDBFN2	Invalid RCP code
1180	49C	FSM	ADMDEFN1	Invalid RCP code
1181	49D	FSM	ADMDEFN2	Invalid RCP code
1190	4A6	FSM	ADMD3DIF	Invalid device id
1191	4A7	FSM	ADMD3DQC ADMD3DQF	Array count mismatch
1192	4A8	FSM	ADMD3DIF ADMD3DPG	Unsupported RCP code
1193	4A9	FSM	ADMD3DIS ADMD3DPL ADMD3EMU	Unsupported function code
1194	4AA	FSM	ADMD3FPI	Invalid projection for conversion to IOCA
1195	4AB	FSM	ADMD3DIF	ISPUT invalid span_state
1200	4B0	FSM	ADMDSSST	An invalid value of TRNDEFRT has been found in ADMDATRN. Change ADMDATRN
1201	4B1	FSM	ADMDSSST	None of the 'types' match the device characteristics, so a default 'type' cannot be selected. Change ADMDATRN
1203	4B3	FSM	ADMDSDS ADMDSRO	TRNUCTP is 0. No upper case translate table defined
1204	4B4	FSM	ADMDSSST	TRNPTP is 0. No protection translate table defined
1205	4B5	FSM	ADMDSSST	TRNTYPEP(N) is 0. There is no address for a type descriptor block
1206	4B6	FSM	ADMDSSST	Number of types defined (TRNNTYPE) is zero or negative
1207	4B7	FSM	ADMDSII	Invalid request parameter
1208	4B8	FSM	ADMDSUB	No alternate device QIB
1209	4B9	FSM	ADMDSIK	Invalid request parameter
1211	4BB	FSM	ADMDASLC	Invalid parameter
1230	4CE	FSM	ADMDHCO	Invalid order code detected
1231	4CF	FSM	ADMDHIN	Invalid device cell size
1232	4D0	FSM	ADMDHIN	Not enough syslinetypes
1237	4D5	FSM	ADMDHAL ADMDHPA	Invalid system linetype
1238	4D6	FSM	ADMDHUW	One-cell buffer control error
1239	4D7	FSM	ADMDHUA ADMDHUW	Area shading routing error
1260	4EC	FSM	ADMDIHD	Error from FSPCRT for create default page
1261	4ED	FSM	ADMDIHD	Error from FSPQRY for query default page
1262	4EE	FSM	ADMDIHD	Error from ASDFLD for define header line 1
1264	4F0	FSM	ADMDIHD	Error from ASCPUT for header data line 1
1265	4F1	FSM	ADMDIOC	Invalid PROCOPT list
1267	4F3	FSM	ADMDIPA	Error from FSPQRY for print page

Abend Dec	 Hex	Comp- onent	Issuing Module	Description
1268	4F4	FSM	ADMDIPA	Error from FSPDEL for print page
1269	4F5	FSM	ADMDIGR	Old format print file — not supported
1276	4FC	FSM	ADMDICO	No primary device
1290	50A	FSM	ADMDFPI	Invalid PHIGS return
1293	50D	FSM	ADMDGGP ADMUPCT ADMUPCV ADMUPGT ADMUPGV	Area greater than 60k
1300	514	ESI	ADMEIN0C ADMEIN0D ADMEIN0O ADMEIN0V ADMEROUI ADMERO0C ADMERO0D ADMERO0O ADMERO0V	Invalid ECA
1301	515	ESI	ADMEROUI ADMERO0C ADMERO0D ADMERO0O ADMERO0V	Invalid RCP subcomponent code
1302	516	ESI	ADMEGFAC ADMEGFBC ADMEGFCO ADMEGFDO ADMEGFEV ADMEGFFC ADMEGFGO ADMEGFII ADMEGFJI ADMEGFKV ADMEGFLO ADMEGFOD ADMEGFPD ADMEGFSO ADMEGF00 ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF40 ADMEGF60 ADMEGF70 ADMEGF80 ADMEGF90 ADMEOSD0 ADMEOSLC ADMEOSLD ADMEOSLO ADMEPQ0O ADMEROUI ADMERO0C ADMERO0D ADMERO0O ADMERO0V	Invalid RCP function code
1303	517	ESI	ADMECM0C ADMECM0D ADMECM0I ADMECM0O ADMECM0V ADMEDSSV ADMEGFQD ADMEPCBI ADMEPRGI ADMEPR0C ADMEPR0D ADMEPR0O ADMEPR0P ADMEPR0V ADMEROUI	Function not supported
1308	51C	ESI	ADMEPRGI ADMEPR0C ADMEPR0D ADMEPR0O ADMEPR0V	Program load failure — only if unconditional request
1309	51D	ESI	ADMECM0C ADMECM0D ADMECM0I ADMECM0O ADMECM0V	X-INST table already loaded or unloaded

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1310	51E	ESI	ADMECM0V	Bad return code from CMS SUBCOM
1311	51F	ESI	ADMECM0V	Bad return code from CMS DMSFRT
1312	520	ESI	ADMEROUI ADMEROOD ADMEROOO ADMEROOV ADMKSRVC	Unconditional GETMAIN failure
1313	521	ESI	ADMEROOD ADMEROOO ADMEROOV	FREEMAIN failure
1314	522	ESI	ADMEROUI ADMEROOC ADMEROOD ADMEROOO ADMEROOV	Request exceeds maximum storage
1315	523	ESI	ADMEGF00	Invalid family code
1316	524	ESI	ADMEGFAC ADMEGFBC ADMEGFCO ADMEGFDO ADMEGFEV ADMEGFFC ADMEGFGO ADMEGFII ADMEGFJI ADMEGFKV ADMEGFLO ADMEGFOD ADMEGFPD ADMEGFQD ADMEGFSO ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF40 ADMEGF50 ADMEGF60 ADMEGF70 ADMEGF80 ADMEGF90	Invalid FRB or FRB parameters
1317	525	ESI	ADMEGFAC ADMEGFBC ADMEGFCO ADMEGFDO ADMEGFEV ADMEGFFC ADMEGFGO ADMEGFII ADMEGFJI ADMEGFKV ADMEGFLO ADMEGFOD ADMEGFPD ADMEGFQD ADMEGFSO ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF40 ADMEGF50 ADMEGF60 ADMEGF70 ADMEGF80 ADMEGF90	Invalid ORB or ORB parameters
1318	526	ESI	ADMEGF10 ADMEGF11 ADMEGF12 ADMEGF13 ADMEGF14 ADMEGF20 ADMEGF30 ADMEGF70 ADMEGF80 ADMEGF90	Invalid IFCT TSID field
1319	527	ESI	ADMEDCKI	Invalid address or length parameters
1321	529	ESI	ADMEGF40	Invalid piece-long parameter
1322	52A	ESI	ADMEGFLO ADMEGFOD	Error from VSAM GENCB or MODCB
1323	52B	ESI	ADMEPQ0O	Print queue not open
1324	52C	ESI	ADMEPQ0O	Queue identifier changed
1326	52E	ESI	ADMEOSLC ADMEOSLD ADMEOSLO ADMEOSQI	Invalid object type
1327	52F	ESI	ADMEOSD0	Invalid name count type
1328	530	ESI	ADMEOSD0	Object not on queue
1329	531	ESI	ADMEGF30	Invalid object type

Abend Dec	Abend Hex	Comp-onent	Issuing Module	Description
1330	532	ESI	ADMESSM0	Invalid SSM request
1331	533	ESI	ADMESSM0	Overlap found on FAQE
1334	536	ESI	ADMEGFCO ADMEGFGO ADMEPQ0O ADMEPR0O ADMERO0O	Invalid feedback or return from task switch exit
1350	546	ESI	ADMEINTI	ASMTDLI not link-edited
1353	549	ESI	ADMEPCBI	Invalid ordinal
1354	54A	ESI	ADMEPCBI	Invalid PCB type
1355	54B	ESI	ADMEDLII	Probable GDDM logic error in DL/I operation
1356	54C	ESI	ADMEABNI ADMKSRVC	DL/I terminating condition found in scheduler subtask
1380	564	ESI	ADMEPRGI ADMEPR0C ADMEPR0D ADMEPR0O ADMEPR0V	Wrong program load form
1381	565	ESI	ADMEPRGI ADMEPR0C ADMEPR0D ADMEPR0O	Module loaded above 16 megabytes when GDDM initialized in 24-bit mode - second occurrence - possible recursion
1400	578	TSI	ADMLRO00	TSIA already initialized
1401	579	TSI	ADMLRO00 ADMLRO1C ADMLRO1I ADMLRO1O ADMLRO1T ADMLRO1V ADMLRO2C ADMLRO2I ADMLRO2O ADMLRO2T ADMLRO2V ADMLRO3C ADMLRO3I ADMLRO3T ADMLRO3V ADMLRO4D ADMLRO4T ADMLRO4V	TSIA not initialized or bad TCA
1402	57A	TSI	ADMLRO00 ADMLRO1C ADMLRO1I ADMLRO1O ADMLRO1T ADMLRO1V ADMLRO2C ADMLRO2I ADMLRO2O ADMLRO2T ADMLRO2V ADMLRO3C ADMLRO3I ADMLRO3T ADMLRO3V ADMLRO4D ADMLRO4T ADMLRO4V	Invalid RCP code
1403	57B	TSI	ADMLCN1C ADMLRO00 ADMLSN1C ADMLSN1I ADMLSN1O ADMLSN1T ADMLSN1V ADMLSN20 ADMLSN30 ADMLSN40	Invalid send parameters
1405	57D	TSI	ADMLQU1C ADMLQU1I ADMLQU1O ADMLQU1T ADMLQU1V	Abend after QDV requested via trace qualifier
1406	57E	TSI	ADMLBC1V ADMLSN1V	Invalid command code

Abend Dec	Abend Hex	Comp-onent	Issuing Module	Description
1407	57F	TSI	ADMLAC1C ADMLAC1I ADMLAC1O ADMLAC1T ADMLAC1V ADMLAC30 ADMLAC40 ADMLCN1C ADMLCN1O ADMLCN1T ADMLCN1V ADMLRC1C ADMLRC1I ADMLRC1O ADMLRC1T ADMLRC1V ADMLRC20 ADMLRL1C ADMLRL1I ADMLRL1O ADMLRL1T ADMLRL1V ADMLRL30 ADMLRL40 ADMLRN1I ADMLRN2I ADMLRO00 ADMLSN1C ADMLSN1I ADMLSN1O ADMLSN1T ADMLSN1V ADMLSN20 ADMLSN30 ADMLSN40 ADMLTM1I	Invalid TSIA state
1408	580	TSI	ADMLAC1I ADMLRL1I ADMLRL1O ADMLRL1T ADMLRL1V	Invalid buffer parameters
1410	582	TSI	ADMLCN1C ADMLCN1I ADMLCN1O ADMLCN1T ADMLRC1O ADMLRO00	Invalid CNVS parameters (TRB)
1411	583	TSI	ADMLIN1O ADMLIN2O	Invalid initialization parameters
1412	584	TSI	ADMLAC40 ADMLRN2O	Invalid request parameters
1413	585	TSI	ADMLIN1C ADMLIN1I ADMLIN1O ADMLIN1T ADMLIN1V ADMLIN2C ADMLIN2I ADMLIN2T ADMLIN2V ADMLIN3C ADMLIN3I ADMLIN3T ADMLIN3V ADMLIN4D ADMLIN4T ADMLIN4V	Invalid syslist parameters
1420	58C	TSI	ADMLIN1T	Failure in initialize, TCLEARQ
1422	58E	TSI	ADMLIN1T	Failure in initialize, STTMPMD
1423	58F	TSI	ADMLRC1T	Failure in receive, TGET
1425	591	TSI	ADMLRC1T	Failure in receive, TCLEARQ 2
1427	593	TSI	ADMLSN1T	Failure in send, STFSMODE
1428	594	TSI	ADMLRC1T	Failure in send, STFSMODE
1429	595	TSI	ADMLSN1T	Failure in send, TPUT
1430	596	TSI	ADMLSN1T	Failure in send, TGET
1431	597	TSI	ADMLSN1T	Failure in send, TCLEARQ
1432	598	TSI	ADMLSN1T	Failure in send, TCLEARQ 2
1437	59D	TSI	ADMLQU1T	Failure in query, GTTERM
1438	59E	TSI	ADMLRN1T ADMLTM1T	Failure in reinitialize or terminate, TCLEARQ
1439	59F	TSI	ADMLTM1T	Failure in terminate, STFSMODE
1440	5A0	TSI	ADMLTM1T	Failure in terminate, STLINENO
1441	5A1	TSI	ADMLTM1T	Failure in terminate, STTMPMD
1442	5A2	TSI	ADMLTM1T	Failure in terminate, TPUT
1443	5A3	TSI	ADMLRL1T	Failure in release buffer, TCLEARQ
1450	5AA	TSI	ADMLIN1O	ICB invalid or not defined

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1451	5AB	TSI	ADMLER1O ADMLIN1O ADMLQU1O ADMLSN1O ADMLXR1O	Failure in CB-type macro
1452	5AC	TSI	ADMLTM1O	Session in use at termination
1461	5B5	TSI	ADMLER1V	Unexpected ADMLIO1V return code detected by ADMLER1V
1462	5B6	TSI	ADMLTM2V	CSI resource list not empty after INVKOPUV
1470	5BE	TSI	ADMLEN1C	TERMEPR return from CICS
1500	5DC	PGF	ADMBBGS	Invalid component code
1501	5DD	PGF	ADMBBGS	Invalid BGDA pointer or overwritten (after initializing)
1502	5DE	PGF	ADMBBGS	Invalid BGCB pointer or overwritten (after initializing)
1503	5DF	PGF	ADMBBGS	Invalid subcomponent code (before searching branch table)
1504	5E0	PGF	ADMBBGS ADMBGCHR ADMBGFIX ADMBGFLT ADMBGINT ADMBSET	Invalid function
1505	5E1	PGF	ADMBDRAX	Zero axis range
1506	5E2	PGF	ADMBDRAX	Zero tick interval
1507	5E3	PGF	ADMBDRAX	Autotick leading digit calculation error
1520	5F0	IVU	ADM5CUR	Invalid input parameters
1521	5F1	IVU	ADM5CV	
1522	5F2	IVU	ADM5DC	
1523	5F3	IVU	ADM5DF	
1524	5F4	IVU	ADM5ED	
1525	5F5	IVU	ADM5EDC	
1526	5F6	IVU	ADM5EDD	
1527	5F7	IVÚ	ADM5EDE	
1528	5F8	IVU	ADM5EDF	
1529	5F9	IVU	ADM5EDT	
1530	5FA	IVU	ADM5ERR	
1531	5FB	IVU	ADM5EX	File closure error
1532	5FC	IVU	ADM5EXP	
1533	5FD	IVU	ADM5IM	
1535	5FF	IVU	ADM5IM1	
1536	600	IVU	ADM5INIT	
1537	601	IVU	ADM5IP	
1538	602	IVU	ADM5IV	
1539	603	IVU	ADM5IVP	
1540	604	IVU	ADM5LD	
1541	605	IVU	ADM5LDP	
1542	606	IVU	ADM5LDR	
1543	607	IVU	ADM5MAIN	
1544	608	IVU	ADM5NM	
1545	609	IVU	ADM5NMA	
1546	60A	IVU	ADM5OP	
1547	60B	IVU	ADM5PFK	
1548	60C	IVU	ADM5PFS	
1549	60D	IVU	ADM5PJ	
1550	60E	IVU	ADM5PJA	
1551	60F	IVU	ADM5PJC	
1552	610	IVU	ADM5PJL	
1553	611	IVU	ADM5PJS	
1554	612	IVU	ADM5PJ1	

Abend Dec	 Hex	Comp- onent	Issuing Module	Description
1555	613	IVU	ADM5PJ2	
1556	614	IVU	ADM5PJ3	
1557	615	IVU	ADM5PR	
1558	616	IVU	ADM5PR4	File closure error
1559	617	IVU	ADM5SC	
1560	618	IVU	ADM5SCP	
1561	619	IVU	ADM5SH	
1562	61A	IVU	ADM5SHN	
1563	61B	IVU	ADM5SI	
1564	61C	IVU	ADM5TERM	
1565	61D	IVU	ADM5VW	
1566	61E	IVU	ADM5PJD	
1600	640	VSE	ADMVHELP	Invalid interrupt type
1601	641	VSE	ADMVFSM	Invalid GDDM request
1602	642	VSE	ADMVEDRW ADMVFSM	Bad feedback code from FSM
1603	643	VSE	ADMVHELP	Invalid order in help text
1604	644	VSE	ADMVHELP	Invalid PFKEY interrupt
1620	654	CSPF	EAKABVSS	Symbol set contains invalid data (a dump is produced)
1621	655	CSPF	EAKTPCAD	The EAKLNAME and EAKPNAME parameters in EAKDEFSB are the same
1622	656	CSPF	EAKTPCDS	The DSPRINT request queue is full
1623	657	CSPF	EAKTPCDS EAKTPCJE	SVC 99 error for output data set (DSPRINT) SVC 99 error for sysout file (JES/328X) (and a dump is produced)
1686	696	ICU	ADMPIGUS	I/G SUBCHT error (abend subcode in register 0)
1687	697	ICU	ADMPDRAS	SUBCHT error (abend subcode in register 0)
1688	698	ICU	ADMPHWSH	Bad hierarchy — too many nodes
1689	699	ICU	ADMPSUB0	Function error (abend subcode in register 0)
1690	69A	ICU	ADMPDRSS	Symbol set error (abend subcode in register 0)
1691	69B	ICU	ADMPHWSH	Bad hierarchy — chart type unknown
1692	69C	ICU	ADMPDRAW	Chart-draw error (abend subcode in register 0)
1693	69D	ICU	ADMPMAIN	No CHARTIDs
1694	69E	ICU	ADMPRMLT	Screen formatting error
1695	69F	ICU	ADMPWMLT	Screen formatting error (abend subcode in register 0)
1696	6A0	ICU	ADMPMENW	Invalid call (including word not found)
1697	6A1	ICU	ADMPBCHC ADMPBIFC	Invalid values table
1698	6A2	ICU	ADMPBCHC ADMPBDAT ADMPMAIN	Invalid RCP
1699	6A3	ICU	ADMPIGDA	I/G drawing aid error (abend subcode in register 0)
1700	6A4	ICU	ADMPMENU	Menu index entry not found
1701	6A5	ICU	ADMPFUNC ADMPIDIF	Invalid function request
1702	6A6	ICU	ADMPINFO	Not enough room in PF key or command information field to contain display information
1703	6A7	ICU	ADMPHELP	Help panel index entry not found
1704	6A8	ICU	ADMPGET	Invalid op-code
1705	6A9	ICU	ADMPSAVE	Invalid op-code
1706	6AA	ICU	ADMPSAVE	Invalid save method
1707	6AB	ICU	ADMPCHEK ADMPRIV	Input field too long. Maximum is 132
1708	6AC	ICU	ADMPAREC	Expected record not found (that is, no create and no return code)
1709	6AD	ICU	ADMPAREC	Invalid record type
1710	6AE	ICU	ADMPCONV	Number conversion error found and no return code requested
1711	6AF	ICU	ADMPINFO	Function text not found

Abend Dec	 Hex	Comp- onent	Issuing Module	Description
1712	6B0	ICU	ADMPWTXT	Record type not suitable for text display
1714	6B2	ICU	ADMPMENU	Conversion type invalid for display of stored value
1716	6B4	ICU	ADMPRCMD	Invalid command specified
1717	6B5	ICU	ADMPERR	Field in error does not exist
1720	6B8	ICU	ADMPTXTL	Invalid record type for insert/delete
1723	6BB	ICU	ADMPTERM	Termination error (abend subcode in register 0)
1724	6BC	ICU	ADMPFUNC	No help panel to unload
1725	6BD	ICU	ADMPERR	Length of error text exceeds maximum (as shown by QA0ERRTL)
1726	6BE	ICU	ADMPRECX	Invalid record type for insert/delete
1727	6BF	ICU	ADMPDRAN	Chart-note error (abend subcode in register 0)
1728	6C0	ICU	ADMPINIT	Invalid DSQDEV call for alternate device
1729	6C1	ICU	ADMPINIT	Invalid DSQDEV call for primary device
1731	6C3	ICU	ADMPBNUM ADMPWDIR	Unexpected type code in object descriptor
1732	6C4	ICU	ADMPWDIR	Internal ICU type code not found
1733	6C5	ICU	ADMPWDIR	ASQFLD error (column heading field ID not known)
1734	6C6	ICU	ADMPWDIR	ASRFMT error
1735	6C7	ICU	ADMPWDIR	ASCPUT error
1736	6C8	ICU	ADMPWATT	ASQFLD error (column heading field ID not known)
1740	6CC	ICU	ADMPWCMD	ASRATT debug
1742	6CE	ICU	ADMPWCMD	ASRFMT debug
1750	6D6	OPU	ADMOQPU	Invalid request code
1751	6D7	OPU	ADMOQPU	Error issuing DSQDEV
1753	6D9	OPU	ADMOQPU	An invalid PROCOPT or PROCOPT value is contained within the header of the ADMPRINT file that was processed.
1757	6DD	OPU	ADMOPMT ADMOPUJ	Failure in SPINIT call
1759	6DF	OPU	ADMOPUI	Unable to initialize GDDM
1760	6E0	CSI	ADMYRO00	Invalid RCP subcomponent code
1762	6E2	CSI	ADMYGQC	Invalid QUICKSELL request
1763	6E3	CSI	ADMYGQC	Zero cellsize
1764	6E4	CSI	ADMYASRT	Invalid argument on call to ADMTASRT
1765	6E5	CSI	ADMYGQC	Unconditional allocate
1770	6EA	IMS	ADMKSCHD	Error from GU to I/O PCB
1771	6EB	IMS	ADMKEXST	Input message exceeds input area size
1772	6EC	IMS	ADMKTSWI	No CWA in SSAI
1773	6ED	IMS	ADMKTSWI	Called routine address 0
1780	6F4	NMR	ADMNUMER	Invalid parameter
1781	6F5	CSI	ADMYERRM	Invalid call arguments
1782	6F6	CSI	ADMYERRM	Invalid key
1783	6F7	CSI	ADMYERRM	Duplicate use of retrieve
1784	6F8	CSI	ADMYERRM	Duplicate use of release
1785	6F9	CSI	ADMYERRM	Invalid call arguments
1786	6FA	CSI	ADMYERRM	Invalid key
1790	6FE	IMG	ADM3ARO ADM3RO	Unsupported RCP code
1791	6FF	IMG	ADM3ARO ADM3IRO ADM3PRO ADM3PROJ ADM3RO	Invalid RCP code (component or subcomponent)
1792	700	IMG	ADM3ARO ADM3RO	Invalid control block

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1793	701	IMG	ADM3AASL ADM3AMCP ADM3AMCV ADM3AMEX ADM3AMRH ADM3AMRV ADM3AMSC ADM3AQQL ADM3AQQR ADM3AQQS ADM3AQQW ADM3AQRQ ADM3ATGI ADM3ATGS ADM3ATGT ADM3ATIM ADM3ATPI ADM3ATPS ADM3ATPT ADM3ATRF	IMC NIF logic error
1795	703	IMG	ADM3AAAR ADM3AAEP ADM3AARL ADM3AARS ADM3AASL ADM3AASP ADM3AAWN ADM3AMCP ADM3AMEX ADM3AMRP ADM3AQQA ADM3ATFI	Illegal call from presentation service
1800	708	ISE	ADMIHELP ADMISSCH ADMISSPR ADMISTEP ADMISYMR	Invalid interrupt type from ASREAD
1801	709	ISE	ADMISSCH ADMISTEP ADMISYMR	Unknown alphanumeric field number retrieved by ASQMOD
1802	70A	ISE	ADMIFSM	Unknown component type for ADMIFSM
1803	70B	ISE	ADMISYME	Invalid component code
1810	712	FSM IMC	ADMDDIMG ADM3DCAN ADM3IAAR ADM3IACL ADM3IACR ADM3IADE ADM3IAG ADM3IAGE ADM3IAGI ADM3IAGS ADM3IAII ADM3IAP ADM3IAPE ADM3IAPS ADM3IAQR ADM3IARE ADM3IASV ADM3IATR ADM3IAXF ADM3ICLR ADM3IDEL ADM3IGT ADM3IGTE ADM3IGTS ADM3INEG ADM3INIM ADM3INIT ADM3IORN ADM3IPE2 ADM3IPLC ADM3IPS2 ADM3IPT ADM3IPTE ADM3IPTS ADM3IQRY ADM3IREF ADM3IRES ADM3IRF ADM3ISCL ADM3ISUB ADM3ITFR ADM3ITGT ADM3ITRM ADM3PAPF ADM3PAPT ADM3PEND ADM3PROJ ADM3PRST ADM3PSAV ADM3PSMP	Control block eye-catcher is invalid

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1811	713	FSM	ADMDDIMF	Internal Consistency Check Failed
		FSM	ADMDDIMG	
		IMC	ADMDXSCD	
			ADMD3DIF	
			ADMD3DQR	
			ADMD3EMU	
			ADMD3FPI	
			ADMD3SQR	
			ADMD3SRE	
			ADMM3VER	
			ADM3DCAN	
			ADM3IAAR	
			ADM3IACL	
			ADM3IACR	
			ADM3IADE	
			ADM3IAG	
			ADM3IAGE	
			ADM3IAGI	
			ADM3IAGS	
			ADM3IAII	
			ADM3IAP	
			ADM3IAPE	
			ADM3IAPS	
			ADM3IAQR	
			ADM3IARE	
			ADM3IASV	
			ADM3IATR	
			ADM3IAXF	
			ADM3ICLR	
			ADM3IDEL	
			ADM3IGT	
			ADM3IGTE	
			ADM3IGTS	
			ADM3INEG	
			ADM3INIT	
			ADM3IORN	
			ADM3IPLC	
			ADM3IPS2	
			ADM3IPT	
			ADM3IPTS	
			ADM3IQRY	
			ADM3IREF	
			ADM3IRES	
			ADM3IRF	
			ADM3ISCL	
			ADM3ISUB	
			ADM3ITFR	
			ADM3ITGT	
			ADM3ITRM	
			ADM3PAPF	
			ADM3PAPT	
			ADM3PBEG	
			ADM3PCRT	
			ADM3PDEL	
			ADM3PEND	
			ADM3PROJ	
			ADM3PRST	
			ADM3PSAV	
			ADM3PSMP	
			ADM3TREC	
			ADM3TREL	

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1812	714	FSM	ADMDDIMF	Internal consistency check failed
		FSM	ADMDDIMG	
		IMC	ADMDXSCD	
			ADMD3DIF	
			ADMD3DQR	
			ADMD3EMU	
			ADMD3FPI	
			ADMD3SQR	
			ADMD3SRE	
			ADMM3VER	
			ADM3DCAN	
			ADM3IAAR	
			ADM3IACL	
			ADM3IACR	
			ADM3IADE	
			ADM3IAG	
			ADM3IAGE	
			ADM3IAGI	
			ADM3IAGS	
			ADM3IAII	
			ADM3IAP	
			ADM3IAPE	
			ADM3IAPS	
			ADM3IAQR	
			ADM3IARE	
			ADM3IASV	
			ADM3IATR	
			ADM3IAXF	
			ADM3ICLR	
			ADM3IDEL	
			ADM3IGT	
			ADM3IGTE	
			ADM3IGTS	
			ADM3INEG	
			ADM3INIT	
			ADM3IORN	
			ADM3IPLC	
			ADM3IPS2	
			ADM3IPT	
			ADM3IPTS	
			ADM3IQRY	
			ADM3IREF	
			ADM3IRES	
			ADM3IRF	
			ADM3ISCL	
			ADM3ISUB	
			ADM3ITFR	
			ADM3ITGT	
			ADM3ITRM	
			ADM3PAPF	
			ADM3PAPT	
			ADM3PBEG	
			ADM3PCRT	
			ADM3PDEL	
			ADM3PEND	
			ADM3PROJ	
			ADM3PRST	
			ADM3PSAV	
			ADM3PSMP	
			ADM3TREC	
			ADM3TREL	

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1813	715	FSM	ADMDDIMF	Internal consistency check failed
		FSM	ADMDDIMG	
		IMC	ADMDXSCD	
			ADMD3DIF	
			ADMD3DQR	
			ADMD3EMU	
			ADMD3FPI	
			ADMD3SQR	
			ADMD3SRE	
			ADMM3VER	
			ADM3DCAN	
			ADM3IAAR	
			ADM3IACL	
			ADM3IACR	
			ADM3IADE	
			ADM3IAG	
			ADM3IAGE	
			ADM3IAGI	
			ADM3IAGS	
			ADM3IAII	
			ADM3IAP	
			ADM3IAPE	
			ADM3IAPS	
			ADM3IAQR	
			ADM3IARE	
			ADM3IASV	
			ADM3IATR	
			ADM3IAXF	
			ADM3ICLR	
			ADM3IDEL	
			ADM3IGT	
			ADM3IGTE	
			ADM3IGTS	
			ADM3INEG	
			ADM3INIT	
			ADM3IORN	
			ADM3IPLC	
			ADM3IPS2	
			ADM3IPT	
			ADM3IPTS	
			ADM3IQRY	
			ADM3IREF	
			ADM3IRES	
			ADM3IRF	
			ADM3ISCL	
			ADM3ISUB	
			ADM3ITFR	
			ADM3ITGT	
			ADM3ITRM	
			ADM3PAPF	
			ADM3PAPT	
			ADM3PBEG	
			ADM3PCRT	
			ADM3PDEL	
			ADM3PEND	
			ADM3PROJ	
			ADM3PRST	
			ADM3PSAV	
			ADM3PSMP	
			ADM3TREC	
			ADM3TREL	

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1814	716	FSM	ADMDDIMF	Internal consistency check failed
		FSM	ADMDDIMG	
		IMC	DMDXSCD	
			ADMD3DIF	
			ADMD3DQR	
			ADMD3EMU	
			ADMD3FPI	
			ADMD3SQR	
			ADMD3SRE	
			ADMM3VER	
			ADM3DCAN	
			ADM3IAAR	
			ADM3IACL	
			ADM3IACR	
			ADM3IADE	
			ADM3IAG	
			ADM3IAGE	
			ADM3IAGI	
			ADM3IAGS	
			ADM3IAII	
			ADM3IAP	
			ADM3IAPE	
			ADM3IAPS	
			ADM3IAQR	
			ADM3IARE	
			ADM3IASV	
			ADM3IATR	
			ADM3IAXF	
			ADM3ICLR	
			ADM3IDEL	
			ADM3IGT	
			ADM3IGTE	
			ADM3IGTS	
			ADM3INEG	
			ADM3INIT	
			ADM3IORN	
			ADM3IPLC	
			ADM3IPS2	
			ADM3IPT	
			ADM3IPTS	
			ADM3IQRY	
			ADM3IREF	
			ADM3IRES	
			ADM3IRF	
			ADM3ISCL	
			ADM3ISUB	
			ADM3ITFR	
			ADM3ITGT	
			ADM3ITRM	
			ADM3PAPF	
			ADM3PAPT	
			ADM3PBEG	
			ADM3PCRT	
			ADM3PDEL	
			ADM3PEND	
			ADM3PROJ	
			ADM3PRST	
			ADM3PSAV	
			ADM3PSMP	
			ADM3TREC	
			ADM3TREL	

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1815	717	FSM FSM IMC	ADMDDIMF ADMDDIMG ADMDXSCD ADMD3DIF ADMD3DQR ADMD3EMU ADMD3FPI ADMD3SQR ADMD3SRE ADMM3VER ADM3DCAN ADM3IAAR ADM3IACL ADM3IACR ADM3IADE ADM3IAG ADM3IAGE ADM3IAGI ADM3IAGS ADM3IAII ADM3IAP ADM3IAPE ADM3IAPS ADM3IAQR ADM3IARE ADM3IASV ADM3IATR ADM3IAXF ADM3ICLR ADM3IDEL ADM3IGT ADM3IGTE ADM3IGTS ADM3INEG ADM3INIT ADM3IORN ADM3IPLC ADM3IPS2 ADM3IPT ADM3IPTS ADM3IQRY ADM3IREF ADM3IRES ADM3IRF ADM3ISCL ADM3ISUB ADM3ITFR ADM3ITGT ADM3ITRM ADM3PAPF ADM3PAPT ADM3PBEG ADM3PCRT ADM3PDEL ADM3PEND ADM3PROJ ADM3PRST ADM3PSAV ADM3PSMP ADM3TREC ADM3TREL	Internal consistency check failed
1870	74E	IMS	ADMGSTBI	Invalid utility ID

Abend		Comp-	Issuing	Description
Dec	Hex	onent	Module	
1901	76D	IMD	AEMIOS03	Receive request not recognized
1902	76E	IMD	AEMIOS03	Screen send request not recognized
1903	76F	IMD	AEMIOS03	Send request not recognized
1904	770	IMD	AEMIOS03	Request was not send or receive
1908	774	IMD	AEMIOS00	Sequential request type not recognized
1910	776	IMD	AEMIOS01	Open request not recognized
1915	77B	IMD	AEMSRV02	Function requested not recognized
1940	794	GKS	ADMJGSM	Invalid segment update
1950	79E	FSM	ADMDMOMD	Module called with invalid parameters
1952	7A0	FSM	ADMDMOQ	Module called with invalid parameters
1953	7A1	FSM	ADMDMOR	Module called with invalid parameters
1955	7A3	FSM	ADMDMOPD	Module called with invalid parameters
1956	7A4	FSM	ADMDMDMD	Module called with invalid parameters
1958	7A6	FSM	ADMDMOC	Module called with invalid parameters
1959	7A7	FSM	ADMDMSLG	Module called with invalid parameters
1960	7A8	FSM	ADMDMODG	Module called with invalid parameters
1961	7A9	FSM	ADMDMR	Module called with invalid parameters
1962	7AA	FSM	ADMDMCT	Module called with invalid parameters
1970	7B2	FSM	ADMDMI	Module called at wrong time
1971	7B3	FSM	ADMDMOMD	Module called at wrong time
1972	7B4	FSM	ADMDMODG	Module called at wrong time
1974	7B6	FSM	ADMDMOGO	Module called at wrong time
1991	7C7	FSM	ADMDMSMV	Output buffer too small
1992	7C8	FSM	ADMDMOFM	Unable to find Symbol Set Table entry after load

Abend Dec	Hex	Comp- onent	Issuing Module	Description
2nnn				On IMS/VS, abend codes 2nnn correspond to codes 1nnn listed above. Thus, to find the meaning of, for example, code 2163, look up code 1163.
Gnnn				On CICS, abend codes Gnnn correspond to codes 1nnn listed above. Thus, to find the meaning of, for example, code G163, look up code 1163.

Appendix C. Message-to-module cross-reference

This appendix lists the messages issued by GDDM Base, GDDM-PGF, GDDM-IVU, GDDM-GKS, GDDM-CSPF, and GDDM Interactive Map Definition (GDDM-IMD) in numeric order with the modules that issue the messages. The full text of the messages and an explanation of each is given in the *GDDM Messages* manual. The GDDM-IMD messages start on page 188. The GDDM-CSPF messages start on page 191.

GDDM, GDDM-PGF, GDDM-IVU, and GDDM-GKS messages start with the letters **ADM** followed by four digits. All GDDM-IMD messages start with the letters **AEM** followed by five digits. The three letters with which each of these messages starts are omitted from this list to aid clarity.

GDDM-CSPF messages start with the letters **EAK** followed by **B** if they are background messages, **F** if they are foreground messages, or **V** if they are view utility messages. GDDM-CSPF messages are shown in full.

GDDM-REXX and GDDM-PCLK messages are not shown in the cross-reference.

Msg	Modules that issue the message			
0001	ADMAC0	ADMASEPB	ADMASP	ADME000C
0002	ADME000C	ADME000I	ADMe000O	ADME000V
0003	ADMASEPB			
0004	ADMASP			
0005	ADMASP			
0006	ADMASP			
0007	ADME000C			
0008	ADME000C			
0013	ADMASL			
0014	ADMAC0			
0040	ADMACFP			
0041	ADMACFP			
0042	ADMADQP			
0043	ADMACPG	ADMUOTT	ADMUOTV	
0048	ADMACUP			
0049	ADMACUP			
0050	ADMACPV	ADMACUPS	ADMATPE	
0051	ADMAC0	ADMUOTT	ADMUOTV	
0052	ADMAC0			
0053	ADMACUPS			
0054	ADMAC0	ADMUOTT	ADMUOTV	
0055	ADME0EXC	ADME0EXD	ADME0EXO	ADME0EXV
0056	ADMAC0			
0057	ADMACUP			
0058	ADMACUP			
0059	ADMACUP			
0060	ADMACUP			
0061	ADMACUP	ADMACUPS		
0062	ADMACUP	ADMACUPS		
0063	ADMACUPS			
0064	ADMACUP			
0065	ADMACUP			
0066	ADMACUP	ADMACUPS		
0067	ADMAC0			
0068	ADMACFP			
0069	ADMACUPS			

Msg	Modules that issue the message			
0070	ADMDSRO			
0071	ADMDSDO			
0072	ADMDSDO			
0073	ADMDSDO			
0074	ADMDSDS			
0075	ADMDSDS			
0076	ADMDSDS			
0077	ADMDSDS			
0078	ADMDSDS			
0079	ADMDSDS	ADMLIN1I	ADMLIN2I	ADMLIN3I
0080	ADMDSF1			
0081	ADMDSDS			
0082	ADMDSDS	ADMDSRO		
0084	ADMDSDS			
0085	ADMLIN1C	ADMLIN1I	ADMLIN1O	ADMLIN1T
	ADMLIN1V	ADMLIN1X	ADMLIN2C	ADMLIN2I
	ADMLIN2T	ADMLIN2V	ADMLIN3C	ADMLIN3I
	ADMLIN3T	ADMLIN3V	ADMLIN4D	ADMLIN4T
	ADMLIN4V			
0086	ADMDSF1	ADMDSF2	ADMDSF4	ADMDSTQ
	ADMLIN2I			
0087	ADMDSDQ	ADMDSQL	ADMDSQP	
0088	ADMDSDQ			
0089	ADMDSDS	ADMDSF1	ADMDSF2	ADMDSF3
	ADMDSF4			
0090	ADMDSDQ			
0091	ADMDSDS	ADMDSF1		
0092	ADMDSF1			
0093	ADMDSF4			
0094	ADMDSTQ			
0095	ADMDSF1			
0096	ADMDSTQ	ADMPRINT		
0097	ADMDSDQ	ADMPRINT		
0098	ADMDSRO			
0099	ADMDSRT			
0100	ADMDSF1	ADMDSF2		
0101	ADMDSDQ			
0102	ADMDSOO			

Msg	Modules that issue the message			
0103	ADMDSTQ	ADMPRINT		
0104	ADMDTCON	ADMPRINT		
0105	ADMDSA1			
0106	ADMDSF1			
0107	ADMDSF1			
0108	ADMDSDS			
0109	ADMDSDS			
0110	ADMDCPC			
0111	ADMDCOS			
0113	ADMDCQS			
0114	ADMDCRS			
0115	ADMDCGS	ADMDCPS	ADMDCRS	
0116	ADMDCQS			
0117	ADMDCDS	ADMDCGS	ADMDCPG	ADMDCPS
0118	ADMDCDS	ADMDCGS	ADMDCPG	ADMDGSQ
0119	ADMDCGS	ADMDCVS		
0120	ADMDCDS	ADMDCPG	ADMDGSQ	
0121	ADMDCAS	ADMDCPS		
0122	ADMDCPS			
0123	ADMDCVS	ADMIFSM		
0124	ADMDCGS	ADMDCPS	ADMDCQS	ADMDCVS
0125	ADMDCVS			
0126	ADMDCPS			
0127	ADMDCRS	ADMDGMD		
0128	ADMDCVS			
0129	ADMDCQD	ADMDGSQ		
0130	ADMDCPC			
0131	ADMDCPC			
0132	ADMDCPC	ADMIFSM	ADMVFSM	
0133	ADMDCPC			
0134	ADMDCPC			
0135	ADMDCGS			
0136	ADMDCQD			
0137	ADMDCPC			
0138	ADMDCPC			
0139	ADMDCQD			
0140	ADMBCHSG	ADMBNOTE	ADMDGIN	ADMDGSC
	ADMDGSE	ADMDGSV	ADMDGTR	ADMDGMSV
0141	ADMDGWI			
0142	ADMDGWI			
0143	ADMBCHSG	ADMBNOTE	ADMDGSE	ADMDGSV
	ADMDGMSV			
0144	ADMDGWI			
0145	ADMDGPO	ADMDGSC	ADMDGSE	ADMDGTR
0146	ADMDGGI	ADMDGIN	ADMDGLD	ADMDGPAD
	ADMDGPC	ADMDGPEE	ADMDGPMK	ADMDGPO
	ADMDGPR	ADMDGSV	ADMDGTR	ADMDGXCR
	ADMDGMLD	ADMDGMSV		
0147	ADMDGPA			
0148	ADMDGPM			
0149	ADMDGSC	ADMDGSE	ADMDGTR	
0150	ADMDGLD	ADMDGPAD	ADMDGPO	ADMDGSC
	ADMDGSE	ADMDGSQ	ADMDGSV	ADMDGWI
	ADMDGMLD	ADMDGMSV		
0151	ADMBCHVU	ADMDGWI		
0152	ADMDGPA	ADMDGPAC		
0153	ADMDAPRS	ADMDGEN	ADMDGLD	ADMDGPAD
	ADMDGPI	ADMDGPO	ADMDGPR	ADMDGSQ
	ADMDGSV	ADMDGWI		
0154	ADMDGEN	ADMDGFL	ADMDGFP	ADMDGGDD
	ADMDGIN	ADMDGLD	ADMDGPA	ADMDGPAC
	ADMDGPAX	ADMDGPC	ADMDGPE	ADMDGPEE
	ADMDGPI	ADMDGPM	ADMDGPMK	ADMDGPO
	ADMDGPR	ADMDGQI	ADMDGRQ	ADMDGSE
	ADMDGSQ	ADMDGTR	ADMDGWI	ADMDGXC
	ADMDGXCR	ADMDGXE	ADMDOGR	ADMDWPIP
	ADMTGWI			
0155	ADMDGPAC			
0156	ADMBNOTE	ADMDGL4	ADMDGP1	ADMDGP3
	ADMDGP4	ADMDGT4	ADMDGPD6	ADMDGPT6
0157	ADMDGPAC			
0158	ADMDGPC	ADMDGPM	ADMDGPMK	
0159	ADMDGPR			
0160	ADMDGPR			
0161	ADMDGGDD	ADMDGPO	ADMDGSC	ADMDGSPP
	ADMDGSV	ADMDGMSV		
0162	ADMDGWI			
0163	ADMDGWI			
0164	ADMDGWI			
0165	ADMDGWI			
0166	ADMDGWI			
0167	ADMDGSE			
0168	ADMDGPAC			
0169	ADMDGPC			
0170	ADMDGPM			
0171	ADMDGPM			
0172	ADMDGPI	ADMDGPM		
0173	ADMDGPI	ADMDGPO	ADMDGMLD	ADMDGMSV
0174	ADMDGPI	ADMUPCT	ADMUPCV	ADMUPGT
	ADMUPGV			
0175	ADMDGPI	ADMDGPM		
0176	ADMDGPEE			
0177	ADMBPIE	ADMDGPEE		
0178	ADMDGPO			
0179	ADMDGGP	ADMDGLD	ADMDGPAD	ADMDGPO
	ADMDGSC	ADMDGSE	ADMDGSQ	ADMDGSV
	ADMDGTR	ADMDGWI	ADMDOCO	ADMDGMLD
	ADMDGMSV			
0180	ADMDGPO			
0181	ADMDGPI	ADMDGPM		
0182	ADMDGGNS	ADMDGQTB		
0183	ADMDGPM			
0184	ADMDGSE			
0185	ADMDGSE			
0187	ADMDGPA			
0188	ADMDGREG			
0189	ADMDGTR			
0190	ADMDGTR			
0191	ADMDGPI			
0192	ADMDGPI			
0193	ADMDGPI			
0194	ADMDGTR			
0195	ADMDGXCR			
0196	ADMDGXCR			
0197	ADMUPCT	ADMUPCV	ADMUPCX	ADMUPGT
	ADMUPGV	ADMUPGX		
0198	ADMDGPA	ADMDGPC	ADMDGPE	ADMDGPEE
	ADMDGPM	ADMDGPR		
0199	ADMDGSE			
0200	ADMDAPDT			
0201	ADMDAPFA	ADMDAPFC	ADMDAPFD	ADMDAPQC
	ADMDAPQL			
0203	ADMDASFA			
0204	ADMDASFA			
0205	ADMDASCR			
0206	ADMDASCR			
0207	ADMDAPFM	ADMDAPQF		
0208	ADMDAPDF	ADMDAPFM	ADMDAPQF	
0209	ADMDAPCU	ADMDAPFM	ADMDAPMF	ADMDAPQF
0211	ADMDAPMF	ADMDASCA		

Msg	Modules that issue the message			
0212	ADMDAPCU	ADMDAPFA	ADMDAPFC	ADMDAPFM
	ADMDAPMF	ADMDAPQC	ADMDAPQL	
0213	ADMDAPQF			
0214	ADMDAPFC	ADMDAPQC		
0215	ADMDAPFC			
0216	ADMDATYP			
0217	ADMDAPFC	ADMDAPQC		
0218	ADMDAPER	ADMDAPQF	ADMDAPRS	ADMDAQCU
0219	ADMDAPFC	ADMPMENU	ADMPWTXT	ADMPWXY
0220	ADMDAPQM			
0221	ADMDAPCU			
0222	ADMDAPRM	ADMDQDEF	ADMDQMOD	
0223	ADMDAPFC			
0224	ADMDASFA			
0225	ADMDAPFM			
0226	ADMDASFA			
0227	ADMDASFA			
0228	ADMDAPFC	ADMPMENU	ADMPWTXT	ADMPWXY
0229	ADMDAPFC	ADMPMENU	ADMPWTXT	ADMPWXY
0230	ADMACPT			
0231	ADMACPT			
0232	ADMACPT			
0233	ADMDGPCX	ADMDJCS	ADMUPCT	ADMUPCV
	ADMUPCX	ADMUPGT	ADMUPGV	ADMUPGX
0234	ADMDHPC	ADMDIAL	ADMDICO	ADMDIGR
	ADMDIMP	ADMDIOC	ADMDIPA	ADMDISS
0244	ADMDIAL	ADMDICO	ADMDIGR	ADMDIIM
	ADMDIMP	ADMDIOC	ADMDIPA	ADMDISS
	ADMDITX			
0248	ADMDIOC	ADMEGF40		
0249	ADMDICO			
0250	ADMDIAL			
0257	ADMDKDGI	ADMDPUAO		
0258	ADMDIHD	ADMDSUB	ADMDSOI	
0259	ADMDKAP	ADMDPUAO		
0260	ADMDIPA			
0261	ADMDIPA			
0264	ADMDISS			
0270	ADMDDSRD			
0272	ADMDDPWR	ADMDKSH		
0273	ADMDPC	ADMDPSCD	ADMDWC	
0274	ADMDDIMF	ADMDDIMG	ADMDDIMX	ADMDDUBC
	ADMDKTX	ADMDSEH	ADMDXSCD	
0275	ADMDLC	ADMDPC	ADMDWC	ADMD3DIF
	ADMD3EMU	ADMD3SDS	ADMD3SXF	
0276	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDTC	ADMDWC	ADMD3SDE
0277	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDSLD	ADMDTC	ADMDWC
0279	ADMDWC	AEMIOS03		
0281	ADMDOOC	ADMDSPQ		
0282	ADMDICO	ADMDOLO		
0283	ADMDOCO			
0284	ADMDOCO			
0285	ADMDOIM			
0300	ADMACSSP	ADMDJCX	ADMECM0V	ADMECMOX
	ADMEGFLO	ADMEGFOD	ADMEROUI	ADMERO0C
	ADMERO0D	ADMERO0O	ADMERO0V	ADMERO0X
	ADMIESI	ADMLIN1I	ADMLIN2I	ADMLIN3I
	ADMLRN1I	ADMYERRM	ADM3AAAR	ADM3AMRM
	ADM3ATBF	ADM3ATPT	ADM3ATTF	ADM3IASV
	ADM3IGT	ADM5ERR	ADM5INIT	
0301	ADMEOSLC	ADMEOSLD	ADMEOSLO	
0302	ADMEGFCO	ADMEGFGO	ADMEGLGO	ADMEPRGI
	ADMEPR0D	ADMEPR0O	ADMEPR0V	ADMEPR0X
0303	ADMEGF30	ADMEGF80		
0304	ADMEGFAC	ADMEGFBC	ADMEGFCO	ADMEGFDO
	ADMEGFEV	ADMEGFEX	ADMEGFFC	ADMEGFGO
	ADMEGFLO	ADMEGFOD	ADMEGFPD	ADMEGF12
	ADMEGF14	ADMEGF30	ADMEGF60	ADMEGF80
	ADMEGLEV	ADMEGLEX	ADMEGLFC	ADMEGLGO
	ADMEGLII	ADMEOSD0		
0305	ADMEGFCO	ADMEGFGO	ADMEGLGO	
0306	ADMEGFFC	ADMEGFOD	ADMEGLFC	ADMUPCT
	ADMUPCV	ADMUPCX	ADMUPGT	ADMUPGV
	ADMUPGX			
0307	ADMACUP	ADMDCOS	ADMDMSLG	ADMEGFBC
	ADMEGFDO	ADMEGFEV	ADMEGFEX	ADMEGFFC
	ADMEGFGO	ADMEGFII	ADMEGFQD	ADMEGF30
	ADMIESI	ADMIFSM	ADMVFSM	AEMIOS00
0309	ADMERO0V	ADMERO0X		
0310	ADMEGFCO	ADMEGFEV	ADMEGFEX	ADMEGFGO
	ADMEGFPD	ADMEGFQD	ADMEGLEV	ADMEGLEX
	ADMEGLGO	ADMEPQ0O	ADMPCHEK	ADM1MSLV
0311	ADMEGFOD	ADMEGFPD	ADMERO0C	ADMERO0D
	ADMERO0O	ADMERO0V	ADMERO0X	
0312	ADMEGFAC	ADMEGFBC	ADMEGFCO	ADMEGFEV
	ADMEGFEX	ADMEGFFC	ADMEGFGO	ADMEGFII
	ADMEGFLO	ADMEGFOD	ADMEGFPD	
0313	ADMDCFW	ADMDMSLG	ADMDSCC	ADMEGFOD
	ADMEGF30	ADMEGF40	ADMEGF70	ADMEGF80
	ADMEOSD0	ADMPGET	ADMPOBJS	ADM3IASV
	ADM3PRST			
0314	ADMACUP	ADMEGFCO	ADMEGFOD	
0315	ADMEGFKV	ADMEGFKX	ADMLTM1V	ADMLTM1X
0316	ADMEGFKV	ADMEGFKX		
0317	ADMEGFAC	ADMEGFCO	ADMLTM30	ADMLTM40
0318	ADMEGFCO	ADMEGFGO	ADMEGLGO	
0319	ADMEGF10	ADMEPR0C		
0320	ADMEGFCO	ADMEGFGO	ADMEGFLO	ADMEGFOD
	ADMEGLGO	ADMEPQ0O		
0321	ADMEPQ0O			
0322	ADMEGFEV	ADMEGFEX	ADMEGF70	ADMEGLFC
0323	ADMEGFEV	ADMEGFEX	ADMEGFFC	ADMEGFLO
	ADMEGFOD	ADMEGFQD		
0324	ADMEGFBC	ADMEGFDO	ADMEGFEV	ADMEGFEX
	ADMEGFFC	ADMEGFGO	ADMEGFII	ADMEGFOD
	ADMEGFQD	ADMPSAVE	AEMIOS00	
0325	ADMACPG	ADMACUP	ADMDCFW	ADMDDPWR
	ADMDGLD	ADMDJUT	ADMDKSH	ADMDMSLG
	ADMDSCC	ADMDSQP	ADMEGFBC	ADMEGFCO
	ADMEGFEV	ADMEGFEX	ADMEGFFC	ADMEGFGO
	ADMEGFII	ADMEGFJI	ADMEGFLO	ADMEGFOD
	ADMEGFPD	ADMEGFQD	ADMEGF30	ADMEGF60
	ADMEGF80	ADMEGLEV	ADMEGLEX	ADMEGLFC
	ADMEGLGO	ADMEGLII	ADMEOSD0	ADMFOU
	ADMLRC20	ADMOPRT	ADMPGET	ADMPIMP
	ADMPISEQ	ADMPOBJS	ADMUPCT	ADMUPCV
	ADMUPCX	ADMUPGT	ADMUPGV	ADMUPGX
	ADM3IASV	ADM3PRST	ADM4GSF	ADM5IM
	AEMIOS03			
0326	ADMEDCKI	ADMEDK0C	ADMEDK0O	ADMEDK0V
0327	ADMEGFAC	ADMEGFBC	ADMEGFFC	ADMEGF10
	ADMEGF13	ADMEGLFC	ADMLRN2C	ADMLTM2C
0328	ADMEGFDO	ADMEGFSO	ADM5IM	
0329	ADMEPQ0O			
0330	ADMEPQ0O			
0334	ADMEPCBI	ADMLIN2I	ADMLIN3I	

Msg	Modules that issue the message			
0335	ADMEDPCI			
0336	ADMEDPCI			
0338	ADMEDLII			
0339	ADMEGFJI			
0340	ADMEGFJI			
0341	ADMESDSI			
0342	ADMESDSI			
0343	ADMESDSI			
0344	ADMEGFII	ADMEOSLI		
0346	ADMEMSGI			
0347	ADMEDLII			
0348	ADMEGFII	ADMEGFJI		
0360	ADMEGFGO	ADMEGFLO	ADMEGFOD	
0361	ADMEGF10			
0370	ADMACPG	ADMEOSLC	ADMEOSLD	ADMEOSLI
	ADMEOSLO	ADMUOTT	ADMUOTV	ADMEOSQ0
0371	ADMEOSLC	ADMEOSLD	ADMEOSLI	ADMEOSLO
0372	ADMEGF60			
0373	ADMEPRGI	ADMEPR0C	ADMEPR0D	ADMEPR0O
0374	ADMEGF60			
0400	ADMLIN2I	ADMLQU1C	ADMLQU1I	ADMLQU1O
	ADMLQU1T	ADMLQU1V	ADMLQU1X	
0401	ADMLIN1T	ADMLTM1T		
0402	ADMLSN1T			
0403	ADMLIN1T	ADMLRC1T	ADMLSN1T	ADMLTM1T
0404	ADMDDPWR	ADMDDUBC	ADMDKRX	ADMDSQA
	ADMDSQL	ADMDSQP	ADMDSXP	ADMDXRIO
	ADMLBC1V	ADMLBC1X	ADMLQU1C	ADMLQU1O
	ADMLQU1T	ADMLQU1V	ADMLQU1X	
0405	ADMLRC1T	ADMLSN1T		
0406	ADMDTCON	ADMDTPGD	ADMLRC1I	ADMLRC1V
	ADMLRC1X	ADMLRO00		
0407	ADMLQU1C	ADMLQU1I	ADMLQU1O	ADMLQU1T
	ADMLQU1V	ADMLQU1X		
0408	ADMLAC1I	ADMLRN2C	ADMLRN2I	ADMLRN2V
	ADMLRN2X	ADMLSN20		
0409	ADMLIN4D	ADMLIN4T	ADMLIN4V	
0410	ADMLER1O	ADMLIN1O	ADMLQU1O	ADMLSN1O
	ADMLXR1O			
0411	ADMLER1O			
0412	ADMLER1O			
0413	ADMLER1O			
0414	ADMLER1O			
0415	ADMLER1O			
0416	ADMLER1O			
0417	ADMLER1O			
0418	ADMLER1O			
0419	ADMLER1O			
0420	ADMLER1O			
0421	ADMLER1O			
0422	ADMLER1O	ADMLIN1O		
0423	ADMLER1O			
0424	ADMLER1O			
0425	ADMLER1O	ADMOQPU		
0426	ADMI ER1O			
0427	ADMLER1O			
0428	ADMLXR1C	ADMLXR1O		
0435	ADMLIN1T			
0436	ADMLIN1T			
0437	ADMLSN1T			
0438	ADMLIN2T			
0440	ADMLRC1V	ADMLRC1X		
0441	ADMLER1V	ADMLER1X		
0442	ADMLER1V			
0443	ADMLER1V	ADMLER1X	ADMLQU1V	ADMLQU1X
0444	ADMLIN1V	ADMLIN1X		
0445	ADMLIN1V	ADMLIN1X		
0446	ADMLRN2V	ADMLRN2X	ADMLTM2V	ADMLTM2X
0447	ADMLIN1V	ADMLIN1X		
0448	ADMLIN1V	ADMLIN1X		
0449	ADMLIN1V	ADMLIN1X		
0460	ADMLRO1I			
0461	ADMLIN1I			
0462	ADMLIN1I	ADMLIN2I	ADMLIN3I	
0463	ADMLSN1I			
0470	ADMLRN3C			
0481	ADMLIN1C	ADMLIN1I	ADMLIN1O	ADMLIN1T
	ADMLIN1V	ADMLIN1X	ADMLIN2C	ADMLIN2I
	ADMLIN2T	ADMLIN2V	ADMLIN3C	ADMLIN3I
	ADMLIN3T	ADMLIN3V	ADMLIN4D	ADMLIN4T
	ADMLIN4V			
0482	ADMLIN1T	ADMLIN1V	ADMLIN1X	ADMLIN2C
	ADMLIN2I	ADMLIN2V	ADMLIN3I	ADMLIN3V
	ADMLIN4D	ADMLIN4T	ADMLIN4V	
0483	ADMLQU40			
0484	ADMLIN1C	ADMLIN1I	ADMLIN1T	ADMLIN1V
	ADMLIN1X			
0485	ADMLIN1C	ADMLIN1T	ADMLIN1V	ADMLIN1X
	ADMPINIT			
0486	ADMLIN1C	ADMLIN1O		
0488	ADMDSDS	ADMLIN1C	ADMLIN1T	ADMLIN1V
	ADMLIN1X			
0489	ADMLIN4T	ADMLIN4V		
0490	ADMLER1O	ADMLQU1O	ADMLSN1O	ADMLXR1O
0491	ADMLIN1O	ADMLQU1O	ADMLSN1O	ADMLXR1O
	ADMOPST			
0495	ADMLXR1C	ADMLXR1O	ADMOQPU	
0496	ADMDDPWR	ADMDDUBC	ADMDKTX	ADMDSQA
	ADMDSQL	ADMDSQP	ADMDSXP	ADMDWTRM
	ADMDXRIO	ADMLER1O	ADMLQU1C	ADMLQU1O
	ADMLQU1T	ADMLQU1V	ADMLQU1X	ADMLSN1T
	ADMLXR1C	ADMMTSI		
0497	ADMDKTX	ADMLBC1V	ADMLBC1X	ADMLXR1O
	ADMLXR1C			
0498	ADMDKRX	ADMDKUT		
0499	ADMDDPWR	ADMDDUBC	ADMDKTX	ADMDSQA
	ADMDSQL	ADMDSQP	ADMDSXP	ADMDWTRM
	ADMDXRIO	ADMLBC1V	ADMLBC1X	ADMLQU1C
	ADMLQU1T	ADMLQU1V	ADMLQU1X	ADMLSN1T
	ADMMTSI			
0501	ADMBBGS	ADMBCHRT	ADMBGCHR	ADMBGFIX
	ADMBGFLT			
0502	ADMBDRAW			
0503	ADMBBGS	ADMBGCHR	ADMBGFIX	ADMBGFLT
0504	ADMBDRAW			
0505	ADMBSET			
0506	ADMBASEL			
0507	ADMBSET			
0508	ADMBGCHR	ADMBGFIX	ADMBGFLT	ADMBMPRJ
	ADMBSET			
0509	ADMBGCHR	ADMBNOTE	ADMBSET	
0510	ADMBDKEY			
0511	ADMBDKEY			
0512	ADMBDRAW			
0513	ADMBSET			
0514	ADMBLABL			

Msg	Modules that issue the message			
0515	ADMBSET			
0516	ADMBGCHR	ADMBGFIX	ADMBGFLT	ADMBMPRJ
	ADMBSET			
0517	ADMBSET			
0518	ADMBSET			
0519	ADMBSET			
0520	ADMBSET			
0521	ADMBSET			
0522	ADMBSET			
0523	ADMBCHCV	ADMBGFIX	ADMBGFLT	ADMBMPRJ
	ADMBQPOS	ADMBSET		
0524	ADMBGFIX	ADMBGFLT	ADMBMPRJ	ADMBSET
0525	ADMBCHSG			
0526	ADMBDRAX			
0527	ADMBDRAX			
0528	ADMBDRAX			
0529	ADMBARS			
0530	ADMBADTM	ADMBDSAX		
0531	ADMBADTM	ADMBDOAX	ADMBNOTE	
0532	ADMBCHRT			
0533	ADMBCHRT			
0534	ADMBARS	ADMBDTAB	ADMBHIST	ADMBMANH
	ADMBPLOT	ADMBPOLR		
0535	ADMBARS	ADMBCHRT	ADMBDTAB	ADMBHIST
	ADMBPLOT			
0536	ADMBARS			
0537	ADMBPIE			
0538	ADMBPIE			
0539	ADMBPIE			
0540	ADMBPIE			
0541	ADMBVENN			
0542	ADMBVENN			
0543	ADMBCHRT			
0544	ADMBCHRT	ADMBGFTX	ADMBMISS	ADMBPOLR
0545	ADMBDRAX			
0546	ADMBCHVU	ADMBDRAX	ADMBDTAB	
0547	ADMBARS			
0548	ADMBCHSG	ADMBNOTE		
0549	ADMBSET			
0550	ADMBARS	ADMBCHRT	ADMBDKEY	ADMBDTAB
	ADMBMISS	ADMBSET		
0551	ADMBNOTE			
0552	ADMBNOTE			
0553	ADMBNOTE			
0554	ADMBSET			
0555	ADMBSET			
0556	ADMBNOTE			
0557	ADMBDTTL			
0558	ADMBDTTL			
0559	ADMBVENN			
0560	ADMBDRAX			
0561	ADMBPIE			
0562	ADMBPIE	ADMBVENN		
0563	ADMBNOTE			
0564	ADMBNOTE			
0565	ADMBNOTE			
0566	ADMBDSAX			
0567	ADMBNOTE			
0568	ADMBCHCV	ADMBQPOS		
0569	ADMBSET			
0570	ADMBMPRJ			
0571	ADMBPIE			
0572	ADMBPIE			
0573	ADMBMANH			

Msg	Modules that issue the message			
0574	ADMBMANH			
0575	ADMBCHCV	ADMBQPOS		
0576	ADMBCHCV			
0577	ADMBCHCV			
0578	ADMBCHCV			
0579	ADMBCHCV	ADMBNOTE		
0580	ADMBCHCV			
0581	ADMBMANH			
0582	ADMBMANH			
0583	ADMBDOAX	ADMBDSDO		
0584	ADMBDRAX			
0585	ADMBDRAW			
0586	ADMBNOTE			
0587	ADMBDTAB			
0588	ADMBDTAB			
0589	ADMBLABL			
0601	ADMVSELP	ADMVSYMB		
0603	ADMVASPE	ADMVCC	ADMVERAS	ADMVREF
	ADMVREN	ADMVSELP	ADMVSWIT	
0604	ADMVSELP			
0605	ADMVSELP			
0606	ADMVVSSE			
0607	ADMVSELP			
0608	ADMVSELP			
0609	ADMVSELP			
0610	ADMVSELP			
0611	ADMVBREA	ADMVJOIN		
0612	ADMVBREA	ADMVJOIN		
0613	ADMVJOIN			
0614	ADMVBREA			
0615	ADMVREN			
0616	ADMVREN			
0617	ADMVREN			
0618	ADMVSYMB			
0619	ADMVSYMB			
0620	ADMVSYMB			
0621	ADMVSYNT			
0622	ADMVVSSE			
0623	ADMVSYNT			
0624	ADMVSAVE			
0625	ADMVSYNT			
0626	ADMVSYNT			
0627	ADMVSYNT			
0628	ADMVSYNT			
0629	ADMVJOIN			
0630	ADMVSYNT			
0632	ADMVSYNT			
0633	ADMVSYNT			
0634	ADMVSYMB	ADMVSYNT		
0635	ADMVSYNT			
0636	ADMVSYNT			
0637	ADMVSYNT			
0638	ADMVSGET			
0639	ADMVSAVE			
0640	ADMVFIND			
0644	ADMVSHAD			
0645	ADMVSHAD			
0646	ADMVSSEC	ADMVSSET	ADMVSSEV	
0647	ADMVVSSE			
0649	ADMVCC			
0650	ADMVBACK			
0651	ADMVBACK	ADMVCHAN	ADMVCOPY	ADMVCOP1

Msg	Modules that issue the message			
	ADMVDEL	ADMVSHIF	ADMVSTRE	
0652	ADMVBOUN			
0653	ADMVBOUN	ADMVGRIN	ADMVSHIF	
0654	ADMVUPDC			
0657	ADMVCC			
0660	ADMVCC			
0661	ADMVEXIT			
0662	ADMVFORW			
0663	ADMVNEXT			
0664	ADMVPREV			
0666	ADMVDEL			
0671	ADMVHELP			
0672	ADMVHELP			
0673	ADMVHELP			
0674	ADMVHELP			
0675	ADMVHELP			
0676	ADMVHELP			
0677	ADMVGRIN	ADMVHELP	ADMVSELP	ADMVSYMB
0678	ADMVHELP			
0679	ADMVHELP			
0680	ADMVHELP			
0685	ADMVCC	ADMVSAVE	ADMVSETR	
0688	ADMVSAVE			
0689	ADMVSAVE			
0690	ADMVSAVE			
0691	ADMVSAVE			
0697	ADMVSETR			
0698	ADMVCANC			
0699	ADMVCANC			
0700	ADMPINIT			
0701	ADMPDRAP			
0702	ADMPINIT			
0703	ADMPGXY			
0704	ADMPINIT			
0705	ADMPGXY			
0706	ADMPGXY	ADMPINIT		
0707	ADMPINIT			
0708	ADMPGXY	ADMPRXY		
0709	ADMPGXY			
0711	ADMPGET			
0712	ADMPGET			
0713	ADMPSAVE			
0714	ADMPDRAM	ADMPDRAP	ADMPDRAW	ADMPSAVE
0716	ADMPINIT			
0717	ADMPINIT			
0718	ADMPGXY			
0719	ADMPINIT			
0720	ADMPMAIN			
0721	ADMPMAIN			
0722	ADMPMAIN			
0723	ADMPMAIN			
0724	ADMPMAIN			
0725	ADMPBCHC	ADMPBCSQ	ADMPBDIR	ADMPBIFC
	ADMPBNUM	ADMPMAIN		
0726	ADMPBCSQ	ADMPMAIN		
0727	ADMPBCSQ	ADMPBDAT	ADMPBDIR	ADMPMAIN
0728	ADMPBIFC			
0729	ADMPBCHC	ADMPBIFC		
0730	ADMPBIFC			
0731	ADMPBIFC			
0732	ADMPBCHC			
0733	ADMPBDIR			
0734	ADMPBDAT			
0735	ADMPBDAT			
0736	ADMPBCHC	ADMPBIFC	ADMPBNUM	
0737	ADMPBCSQ			
0738	ADMPBNUM			
0739	ADMPBDAT			
0740	ADMPBCHC	ADMPBIFC		
0741	ADMPBCHC	ADMPBIFC		
0742	ADMPBDAT			
0743	ADMPBCHC	ADMPBIFC		
0750	ADMOQPU			
0751	ADMOQPU			
0752	ADMOQPU			
0753	ADMOQPU			
0754	ADMOQPU			
0755	ADMOQPU			
0756	ADMOQPU			
0757	ADMOQPU			
0758	ADMDICO			
0759	ADMOQPU			
0760	ADMYSSTB			
0761	ADMYRSRL			
0762	ADMYRSRL			
0767	ADMYDTIM			
0770	ADMKSCHD			
0771	ADMKNEWM			
0772	ADMKNEWM			
0773	ADMKNEWM			
0774	ADMKOLDM			
0775	ADMKSRVC			
0776	ADMKNEWM			
0777	ADMKEXIT			
0801	ADMISHFT	ADMISTEP		
0802	ADMISSCH	ADMISSPR	ADMISTEP	ADMISYMR
0803	ADMISTEP	ADMISYME	ADMISYMR	
0804	ADMISYNT			
0805	ADMISSCH	ADMISTEP	ADMISYMR	ADMISYNT
0806	ADMISTEP			
0807	ADMIHELP	ADMISSCH	ADMISSPR	ADMISTEP
	ADMISYMR			
0808	ADMISTEP			
0809	ADMISTEP			
0810	ADMISTEP	ADMISYNT		
0811	ADMISSCH			
0812	ADMIBNSS			
0813	ADMISSCH			
0814	ADMISSCH			
0815	ADMISSCH			
0816	ADMISSCH			
0817	ADMISAVE			
0818	ADMIBNSS			
0819	ADMIBNSS			
0820	ADMISYNT			
0821	ADMISAVE			
0822	ADMIDMCH	ADMISHFT		
0823	ADMISSPB	ADMISSPR		
0824	ADMISSPB	ADMISSPR		
0825	ADMISYMB	ADMISYMR		
0826	ADMIHELP			
0827	ADMIBHED			
0828	ADMISYNT			

Msg	Modules that issue the message			
0829	ADMISYNT			
0830	ADMISYNT			
0831	ADMICOMP	ADMISYNT		
0832	ADMIGLOB	ADMISYMR	ADMISYNT	
0833	ADMIBNSS			
0834	ADMIBNSS			
0835	ADMIHELP	ADMISSCH	ADMISSPR	ADMISTEP
	ADMISYMR			
0836	ADMIBNSS			
0837	ADMISSCH			
0838	ADMISSPR	ADMISYMR		
0839	ADMIREF			
0840	ADMISSCH	ADMISYMR		
0841	ADMISCRL			
0842	ADMICURS			
0843	ADMICURS			
0844	ADMICHNG	ADMISYMR		
0845	ADMISYMR			
0846	ADMISTEP			
0848	ADMIBHED			
0849	ADMIBHED			
0850	ADMISYMR			
0851	ADMIBNSS			
0852	ADMIFILL			
0853	ADMIFILL			
0854	ADMIFILL	ADMISYMR		
0855	ADMISYMR			
0857	ADMIFILL	ADMISYMR		
0858	ADMIFILL	ADMISYMR		
0859	ADMIFILL	ADMISYMR		
0860	ADMISYME			
0861	ADMISYMR			
0862	ADMISTEP			
0872	ADMDSF1			
0873	ADMDSQP			
0874	ADMDSQP			
0875	ADMDSXP	ADMDXCOU		
0876	ADMDDPWR	ADMDDUBC		
0877	ADMDDUBC	ADMDTCON	ADMDTPGD	
0878	ADMDDUBC			
0879	ADMDSXP			
0900	ADMDJIN			
0901	ADMDJIN			
0904	ADMDCDS	ADMDCPS	ADMDCQS	ADMDJDOC
	ADMDOCO	ADMDSLD		
0906	ADMDGS4	ADMDJUT		
0909	ADMDJIN			
0911	ADMDJCH			
0912	ADMDCGS			
0913	ADMDJDOC			
0914	ADMDJDOC			
0915	ADMDJCX			
0920	ADMDTCON	ADMDTPGD	ADMPRINT	
0921	ADMDTCON	ADMDTPGD	ADMPRINT	
0922	ADMDTCON	ADMDTPGD	ADMPRINT	
0923	ADMDTCON	ADMDTPGD	ADMPRINT	
0950	ADMDMOR			
0960	ADMDMOPC			
0962	ADMDMSLG			
0963	ADMDMODG	ADMDMSLG		
0964	ADMDMIDE	ADMDMISR		
0966	ADMDMOC			
0967	ADMDMSLG			
0968	ADMDASCR	ADMDMORC		
0969	ADMDMOQF			
0970	ADMDMOQ	ADMDMORC		
0971	ADMDMOR			
0972	ADMDMORC			
0973	ADMDMORC			
0974	ADMDMOQF			
0975	ADMDMOQF			
0976	ADMDMOR			
0977	ADMDMOR			
0978	ADMDMOR			
0979	ADMDMOR			
0980	ADMDMOR			
0981	ADMDMORP			
0982	ADMDMORP			
0983	ADMDMORP			
0984	ADMDMOC			
0985	ADMDMORP			
0990	ADMDMORA			
0991	ADMDMORC			
0992	ADMDMORC			
0999	ADMDMD	ADMDRD		

GDDM messages — 1000 +

Msg	Modules that issue the message			
1000	ADMPANEL	ADMPIGIO		
1001	ADMPCHEK	ADMPRXY		
1002	ADMPCHEK	ADMPRATT	ADMPRMLT	ADMPRNOT
1003	ADMPCIDR			
1004	ADMPCHEK			
1005	ADMPCHEK	ADMPRMLT	ADMPRNOT	
1006	ADMPCHEK			
1007	ADMPCHEK	ADMPRMLT		
1008	ADMPRCBE	ADMPRCMD		
1009	ADMPANEL	ADMPIGIO		
1010	ADMPRCMD			
1011	ADMPRATT			
1012	ADMPRATT			
1013	ADMPRATT			
1014	ADMPRATT			
1015	ADMPRCMD			
1017	ADMPDRAM			
1018	ADMPFUNC			
1019	ADMPCHEK			
1020	ADMPCHEK			
1021	ADMPCHEK			
1022	ADMPCHEK	ADMPRCBE	ADMPRCMD	
1023	ADMPRECT			
1024	ADMPLFIT	ADMPTXTL		
1025	ADMPDRAX			
1026	ADMPRCMD			
1027	ADMPRCMD			
1028	ADMPRCMD			
1029	ADMPRCMD			
1030	ADMPCHEK			
1031	ADMPFUNC			
1032	ADMPRCMD			
1033	ADMPRCMD			
1034	ADMPRCMD			
1035	ADMPRCMD			
1036	ADMPRCMD			
1037	ADMPRCMD			
1038	ADMPRCMD			
1039	ADMPRCMD			
1040	ADMPRCMD			
1041	ADMPRCMD			
1042	ADMPCHEK	ADMPMAIN	ADMPRCMD	
1043	ADMPFUNC			
1044	ADMPRDIR			
1045	ADMPCIDR			
1046	ADMPRCMD			
1047	ADMPRATT			
1049	ADMPRCMD			
1050	ADMPCHEK			
1051	ADMPCHEK			
1052	ADMPCHEK	ADMPRCMD		
1053	ADMPRINT			
1054	ADMPRINT			
1055	ADMPRINT			
1056	ADMPRINT			
1057	ADMPRINT			
1058	ADMPRINT			
1059	ADMPCHEK			
1060	ADMPRINT			
1061	ADMPCHEK			
1062	ADMPDRAS			
1063	ADMPDRAS			
1064	ADMPRMLT			
1065	ADMPCABS	ADMPCEXC	ADMPCIDR	ADMPCSEL
	ADMPCSRT	ADMPCSUM	ADMPRCMD	
1066	ADMPINDX			
1067	ADMPINDX			
1068	ADMPINDX			
1069	ADMPRATT			
1070	ADMPRATT			
1071	ADMPLFIT			
1072	ADMPLFIT			
1073	ADMPLCMD			
1074	ADMPRCMD			
1075	ADMPLCMD	ADMPLFIT		
1076	ADMPLCMD			
1077	ADMPINDX			
1078	ADMPINDX			
1079	ADMPRINT			
1080	ADMPRINT			
1081	ADMPRINT			
1082	ADMPIGUN			
1083	ADMPIGUN			
1084	ADMPIGUN			
1085	ADMPIGUN			
1086	ADMPIGUN			
1087	ADMPIGUN			
1088	ADMPCHEK			
1089	ADMPGDFL			
1090	ADMPFUNC			
1091	ADMPDRAM			
1092	ADMPIGIO			
1093	ADMPRATT			
1094	ADMPDRAX			
1096	ADMPDRAS			
1098	ADMPMENU	ADMPWTXT	ADMPWXY	
1099	ADMPRCMD			
1100	ADMPCHEK	ADMPFUNC	ADMPGOTO	
1101	ADMPGOTO			
1102	ADMPGOTO			
1103	ADMPGOTO			
1104	ADMPGOTO	ADMPRCBE		
1105	ADMPRCMD			
1106	ADMPFUNC			
1107	ADMPRXY			
1108	ADMPRATT	ADMPWATT		
1109	ADMPCHEK	ADMPGET		
1110	ADMPDRAM			
1118	ADMPIGIO			
1119	ADMPIGIO			
1120	ADMPIGDA			
1121	ADMPIGIO			
1122	ADMPIGDA			
1123	ADMPIGDA			
1124	ADMPIGDA			
1125	ADMPIGDA			
1126	ADMPIGDA			
1127	ADMPIGDA			
1128	ADMPIGDA			
1129	ADMPIGDA			
1130	ADMPIGDA			
1131	ADMPIGDA			
1132	ADMPIGDA			
1133	ADMPIGDA			
1134	ADMPIGDA			
1135	ADMPIGDA			
1136	ADMPIGDA			

Msg	Modules that issue the message	
1137	ADMPIGDA	
1138	ADMPIGDA	
1139	ADMPIGDA	
1140	ADMPIGDA	
1141	ADMPIGDA	
1142	ADMPIGDA	
1143	ADMPIGIO	
1144	ADMPIGDA	
1145	ADMPIGDA	
1146	ADMPIGDA	
1147	ADMPIGDA	
1149	ADMPIMP	
1150	ADMPIDIF	ADMPISEQ
1154	ADMPIMP	
1155	ADMPIMP	
1156	ADMPRIV	
1157	ADMPRIV	
1158	ADMPRIV	
1159	ADMPRIV	
1160	ADMPWIV	
1161	ADMPRIV	
1162	ADMPIMP	
1163	ADMPIMP	ADMPRIV
1164	ADMPWIV	
1165	ADMPRCMD	
1166	ADMPRIV	
1167	ADMPIMP	
1168	ADMPRIV	
1169	ADMPWIV	
1170	ADMPWIV	
1171	ADMPRIV	ADMPWIV
1172	ADMPWIV	
1173	ADMPCHEK	
1174	ADMPFUNC	
1175	ADMPIMP	
1176	ADMPRIV	
1177	ADMPWIV	
1178	ADMPWIV	
1179	ADMPRIV	
1180	ADMPIDIF	
1182	ADMPIMP	
1183	ADMPIMP	
1184	ADMPIMP	
1185	ADMPIMP	
1186	ADMPIMP	
1187	ADMPIMP	
1188	ADMPIMP	
1189	ADMPIMP	
1190	ADMPIMP	
1191	ADMPIMP	
1192	ADMPCHEK	
1199	ADMPGET	
1201	ADMPRCBE	
1202	ADMPRCBE	
1203	ADMPRCBE	
1204	ADMPRCBE	
1205	ADMPRCBE	
1206	ADMPRCBE	
1207	ADMPFUNC	ADMPRCBE
1208	ADMPFUNC	ADMPRCBE
1209	ADMPRCBE	
1210	ADMPRWSH	
1220	ADMPRWSH	
1221	ADMPDRSS	
1222	ADMPDRSS	
1223	ADMPDRSS	
1224	ADMPIGUS	
1225	ADMPIGUS	

Msg	Modules that issue the message			
2000	ADMOPMT			
2001	ADMOPUT			
2002	ADMOPUT			
2003	ADMOPUJ	ADMOPUT		
2004	ADMOPUT			
2005	ADMOPUT			
2006	ADMOPUT			
2007	ADMOPUT			
2008	ADMOPUT			
2009	ADMOPUT			
2010	ADMOPUJ	ADMOPUT		
2011	ADMOPUT			
2012	ADMOPUJ	ADMOPUT		
2013	ADMOPUJ	ADMOPUT		
2014	ADMOPUJ	ADMOPUT		
2015	ADMOPUJ	ADMOPUT		
2017	ADMOPUT			
2018	ADMOPUT			
2019	ADMOPUT			
2020	ADMOPUJ			
2040	ADMOPRT			
2041	ADMOPRT			
2042	ADMOPRT			
2043	ADMOPRT			
2104	ADMOPUV	ADMOPUX		
2105	ADMOPUV	ADMOPUX		
2106	ADMOPUV	ADMOPUX		
2110	ADMOPUI			
2111	ADMOPUI			
2120	ADMOPUC			
2121	ADMUPRTC			
2122	ADMUPRTC			
2123	ADMUPRTC			
2124	ADMUPRTC			
2125	ADMUPRTC			
2126	ADMUPRTC			
2127	ADMUPRTC			
2128	ADMUPRTC			
2129	ADMUPRTC			
2130	ADMUPRTC			
2131	ADMUPRTC			
2132	ADMUPRTC			
2133	ADMUPRTC			
2134	ADMUPRTC			
2135	ADMUPRTC			
2136	ADMUPRTC			
2137	ADMUPRTC			
2138	ADMUPRTC			
2139	ADMUPRTC			
2140	ADMUPRTC			
2150	ADMUCDSD			
2151	ADMUCDSD			
2152	ADMUCDSD			
2153	ADMUCDSD			
2154	ADMUCDSD			
2155	ADMUCDSD			
2156	ADMUCDSD			
2200	ADMFOU	ADMFSDU		
2201	ADMFOU	ADMFSDU		
2202	ADMFOU	ADMFSDU		
2203	ADMFSDU			
2204	ADMFSDU			
2205	ADMFSDU			
2206	ADMFSDU			
2207	ADMFOU	ADMFSDU		
2208	ADMFSDU			
2209	ADMFSDU			
2210	ADMFOU	ADMFSDU		
2211	ADMFOU			
2212	ADMFOU			
2213	ADMFOU			
2214	ADMFOU			
2215	ADMFOU			
2400	ADMFOU	ADMFSDU		
2401	ADMFOU	ADMFSDU		
2402	ADMFOU			
2403	ADMFOU			
2700	ADM1MSLV			
2701	ADM1MSLV			
2702	ADM1MSLV			
2703	ADM1MSLV			
2704	ADM1MSLV			
2705	ADM1MSLV			
2706	ADM1IMDC	ADM1IMDT		
2750	ADMD4PUT			
2751	ADMD4PUT			
2752	ADMD4AEG	ADMD4MEG	ADMD4PUT	
2753	ADMD4PUT			
2754	ADMD4PUT			
2755	ADMD4AEG			
2756	ADMD4AEG			
2757	ADMD4MEG			
2758	ADMDLC ADMDWC	ADMDOCO ADM4CDU	ADMDPC	ADMDTC
2759	ADMD4QPG			
2760	ADM4CONV			
2775	ADM4CDU			
2776	ADM4CDU			
2777	ADM4CDU			
2778	ADM4CDU			
2779	ADM4BCM	ADM4CDU	ADM4RER	
2780	ADM4CDU			
2781	ADM4CDI			
2782	ADM4CDG			
2783	ADM4GSF			
2784	ADM4GSF			
2785	ADM4GSF			
2786	ADM4GSF			
2787	ADM4BCM	ADM4RER		
2788	ADM4CDU			
2789	ADM4BCM	ADM4RER		
2790	ADM4BCM	ADM4RER		
2791	ADM4GSF			
2792	ADM4CDI	ADM4CDU		
2793	ADM4BCM	ADM4RER		
2794	ADM4BCM	ADM4RER		
2799	ADM4RER			
2800	ADMFUD02			
2801	ADMFUD02			
2802	ADMFUD02			
2803	ADMFUD02			

Msg	Modules that issue the message
2804	ADMFUD02
2805	ADMFUD02
2806	ADMFUD02
2807	ADMFUD02
2808	ADMFUD02
2850	ADMDDSRD
2864	ADMDFCG
2865	ADMDWC
2866	ADMDWAGD
2900	ADMDXMD
2901	ADMDXMD
2902	ADMDXMD
2903	ADMDXMD
2904	ADMDXMD
2905	ADMDXMD
2906	ADMDXMD
2907	ADMDXMD
2908	ADMDXMD
2909	ADMDXMD
2910	ADMDXMD
2911	ADMDXMD
2912	ADMDXMD
2913	ADMDXMD
2914	ADMDXMD
2915	ADMDXMD
2916	ADMDXMD
2917	ADMDXMD
2918	ADMDXMD
2919	ADMDXMD
2920	ADMDXCMM
2921	ADMDXMD
2922	ADMDXMD
2923	ADMDXMD
2924	ADMDXMD
2925	ADMDXMD
2927	ADMDXMD
2928	ADMDXMD
2929	ADMDXMD
2930	ADMDXMD
2931	ADMDXMD
2932	ADMDXCMM
2933	ADMDXCMM
2934	ADMDXMD
2935	ADMDXMD
2936	ADMDXMD
2937	ADMDXMD
2938	ADMDXMD
2939	ADMDXMD
2950	ADMDXMD
2951	ADMDXMD
2952	ADMDXMD
2953	ADMDXMD
2954	ADMDXMD
2955	ADMDXMD
2956	ADMDXMD
2957	ADMDXMD
2958	ADMDXMD
2959	ADMDXMD
2960	ADMDXMD
2961	ADMDXMD
2962	ADMDXMD
2963	ADMDXMD
2964	ADMDXMD
2965	ADMDXMD
2966	ADMDXMD
2967	ADMDXMD
2968	ADMDXWIN
2969	ADMDXCMM
2970	ADMDXWIN
2971	ADMDXWIN
2972	ADMDXWIN
2973	ADMDXWIN
2974	ADMDXCPZ
2975	ADMDXCPZ
2976	ADMDXCPZ
2977	ADMDXCPZ
2978	ADMDXCOU
2979	ADMDXCOU
2980	ADMDXCOU
2981	ADMDXCOU
2982	ADMDXCOU
2983	ADMDXCOU
2984	ADMDXCOU
2988	ADMDXCMM ADMDXCMN
2989	ADMDXCMN
2990	ADMDXCMN
2991	ADMDXCPZ
2992	ADMDXWIN
2993	ADMDXWIN
2994	ADMDXCMC
2995	ADMDXCMD
2996	ADMDXCOU
2997	ADMDAPRS
2998	ADMDXCMC
2999	ADMDXCMM

GDDM messages — 3000 +

Msg	Modules that issue the message			
3000	ADMDQDEF			
3001	ADMDQDEF			
3002	ADMDQDEF	ADMDQDEL	ADMDQMOD	ADMDQQRY
	ADMDQQSZ			
3003	ADMDQDEF	ADMDQMOD	ADMDQQRY	
3004	ADMDQVFL	ADMDQVPG		
3005	ADMDQVDB			
3006	ADMDQDEF	ADMDQMOD		
3007	ADMDQMOD			
3008	ADMDQDEL	ADMDQMOD	ADMDQQRY	ADMDQQSZ
3009	ADMDQDEF	ADMDQMOD	ADMDQQRY	
3010	ADMDQVBL			
3012	ADMDASCR			
3013	ADMDQDEF	ADMDQMOD		
3080	ADMDMOFM			
3081	ADMDMOFM			
3082	ADMDMOFM			
3083	ADMDMDMR	ADMDMP		
3090	ADMDMOQ			
3091	ADMDMOQ			
3092	ADMDMOQ			
3093	ADMDMOQ			
3095	ADMDMR			
3100	ADMDBFN1			
3101	ADMDBFN2			
3102	ADMDBFN1	ADMDBFN2		
3103	ADMDBCRT			
3104	ADMDBCRT			
3105	ADMDBFN1			
3106	ADMDBFN1			
3115	ADMDBPN2			
3116	ADMDBPN1			
3117	ADMDQQID			
3118	ADMDBFN1	ADMDBPN1	ADMDBPN2	ADMDEFN1
	ADMDNCRT	ADMDNMOD	ADMDNQRY	ADMDNQWI
	ADMDNQWN	ADMDNQWP	ADMDNSWP	ADMDQQID
	ADMDQQNO			
3119	ADMDBFN1	ADMDBPN2	ADMDEFN1	ADMDNMOD
	ADMDNQRY	ADMDNQWN	ADMDQQNO	
3120	ADMDEFN1			
3121	ADMDBPN1	ADMDEFN2		
3122	ADMDBPN1	ADMDEFN1	ADMDEFN2	
3123	ADMDEFN1			
3124	ADMDEFN1			
3125	ADMDEFN1			
3126	ADMDEFN1			
3127	ADMDEFN1			
3128	ADMDEFN1			
3129	ADMDEFN1			
3130	ADMDEFN1			
3131	ADMDEFN1			
3132	ADMDEFN1			
3133	ADMDEFN1			
3134	ADMDEFN1			
3135	ADMDEFN1			
3150	ADMDCWIN			
3151	ADMDCWIN			
3152	ADMDCWIN			
3153	ADMDCWIN			
3154	ADMDCWIN			
3155	ADMDCPC			
3156	ADMDCWIN	ADMDEFN1		

Msg	Modules that issue the message			
3157	ADMDCGS	ADMDCPS		
3158	ADMDCFW			
3159	ADMDCENA			
3160	ADMDCENA			
3170	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDTC	ADMDWC	
3172	ADMDTC	ADMDWC		
3173	ADMDPC	ADMDWC		
3174	ADMDTPSR			
3175	ADMDWPRL	ADMDXPRD		
3176	ADMDWPIP			
3177	ADMDWPIP			
3178	ADMDKGP	ADMDWUFL		
3179	ADMDDIEM	ADMDJDOC	ADMDLC	ADMD3EMU
3180	ADMDNCRT	ADMDNDEL	ADMDNMOD	ADMDNQRY
	ADMDNQUN	ADMDNQWI	ADMDNQWN	ADMDNQWP
	ADMDNSEL	ADMDNSWP	ADMDWC	
3181	ADMDNCRT			
3182	ADMDNCRT	ADMDNDEL	ADMDNQWP	ADMDNSEL
	ADMDNSWP			
3183	ADMDNCRT	ADMDNMOD		
3184	ADMDNCRT	ADMDNMOD		
3185	ADMDNCRT	ADMDNMOD		
3186	ADMDNCRT	ADMDNMOD		
3187	ADMDNCRT	ADMDNMOD		
3188	ADMDNCRT	ADMDNMOD		
3189	ADMDNCRT			
3190	ADMDNCRT			
3191	ADMDNCRT	ADMDNMOD	ADMDNQRY	
3192	ADMDNDEL	ADMDNQWP	ADMDNSEL	ADMDNSWP
3193	ADMDBPN1	ADMDNQWP	ADMDNSWP	
3194	ADMDNSWP			
3195	ADMDNDEL			
3196	ADMDNQWI			
3200	ADMDGIN			
3201	ADMDGEN	ADMDGIN	ADMDGRQ	
3202	ADMDGEN	ADMDGIN		
3203	ADMDGIN			
3204	ADMUPCT	ADMUPCV	ADMUPCX	ADMUPGT
	ADMUPGV	ADMUPGX		
3205	ADMDGEN			
3206	ADMDGEN	ADMDGIN		
3207	ADMDGPA			
3208	ADMDGEN	ADMDGIN		
3209	ADMDGEN	ADMDGIN		
3210	ADMDGGI			
3211	ADMDGEN			
3212	ADMDGEN	ADMDGIN		
3213	ADMDGEN			
3214	ADMDFPIP	ADMDWPIP		
3215	ADMDWPIP			
3216	ADMDGEN	ADMDGIN		
3217	ADMDGIN			
3218	ADMDGIN			
3219	ADMDGIN			
3220	ADMDGIN			
3221	ADMDGSC			
3222	ADMDGSC			
3223	ADMDGPAX			
3224	ADMDGPC	ADMDGPE	ADMDGPEE	ADMDGPM
	ADMDGPR			
3225	ADMDGPAX			
3226	ADMDGSE			

Msg	Modules that issue the message			
3227	ADMDGSE			
3228	ADMDGSE			
3229	ADMDGSE			
3230	ADMDGPAC			
3231	ADMDGPO			
3232	ADMDGLD			
3233	ADMDGLD			
3234	ADMDGLD			
3235	ADMDGLD			
3236	ADMDGLD			
3237	ADMDGLD			
3238	ADMUPCT	ADMUPCV	ADMUPCX	ADMUPGT
	ADMUPGV	ADMUPGX		
3239	ADMUPC			
3240	ADMUPC			
3241	ADMUPC			
3242	ADMUPC			
3243	ADMUPC			
3244	ADMUPC			
3245	ADMUPC			
3246	ADMUPC			
3247	ADMUPC			
3248	ADMUPC			
3249	ADMUPC			
3250	ADMDGXCR			
3251	ADMDGIN			
3252	ADMDGGNS			
3253	ADMDGGNS			
3254	ADMDGPA			
3255	ADMDGWI			
3256	ADMDGWI			
3257	ADMDGWI			
3258	ADMDGPA			
3259	ADMDGPA			
3260	ADMDGGP	ADMDGLD	ADMDGPA	ADMDGPAC
	ADMDGPAD	ADMDGPO	ADMDGSC	ADMDGSE
	ADMDGSV	ADMDGTR	ADMDGMLD	ADMDGMSV
3261	ADMDGPAD			
3262	ADMDGSC			
3263	ADMDGSE			
3264	ADMDGGNS			
3265	ADMDGLD	ADMDGSC		
3266	ADMDGPA			
3267	ADMDGSEN			
3268	ADMDGSC			
3269	ADMUPGT	ADMUPGV	ADMUPGX	
3271	ADMDGCFU			
3272	ADMDGWI			
3273	ADMDGSC			
3274	ADMDGPD6	ADMDGPT6		
3277	ADMDGPD6			
3278	ADMDGI6			
3280	ADMDKRX			
3281	ADMDKDGG			
3282	ADMDKDGI	ADMDKGP	ADMDKTP	
3283	ADMDKDGI	ADMDKGP		
3284	ADMDKUT			
3285	ADMDKUT	ADMOQPU		
3286	ADMDKAP			
3291	ADMDGMLD	ADMDGMSV		
3292	ADMDGMLD	ADMDGMSV		
3293	ADMDGMA			

Msg	Modules that issue the message			
3294	ADMDGMLD	ADMDGMSV		
3300	ADM3AAAR	ADM3AARC	ADM3AARL	ADM3AASL
	ADM3AAWN	ADM3AMCP	ADM3AMEX	ADM3AMRP
	ADM3AQQA	ADM3AQQR	ADM3AQQW	
3301	ADM3AAAR	ADM3AMRP		
3302	ADM3AAAR			
3303	ADM3AALC	ADM3AARS	ADM3AMCP	ADM3AMCV
	ADM3AMER	ADM3AMEX	ADM3AMFR	ADM3AMIV
	ADM3AMMI	ADM3AMOR	ADM3AMRP	ADM3AMSC
	ADM3AMTM	ADM3AQQL	ADM3AQQS	ADM3AQRQ
	ADM3ATGS	ADM3ATPS		
3304	ADM3AARS			
3305	ADM3AARS			
3306	ADM3AARC	ADM3AARL	ADM3AASL	ADM3AAWN
	ADM3AMCP	ADM3AMEX	ADM3AQQA	ADM3AQQR
	ADM3AQQW			
3307	ADM3AARC			
3308	ADM3AARC			
3309	ADM3AALC			
3310	ADM3AAMX			
3311	ADM3AAWN			
3312	ADM3AAWN			
3315	ADM3AMCV	ADM3AMSC		
3316	ADM3AMCV	ADM3AMSC		
3317	ADM3AMCV	ADM3AMSC	ADM3ATGS	ADM3ISCL
3318	ADM3AMCV	ADM3AMSC	ADM3ISCL	
3319	ADM3AMOR			
3320	ADM3AMMI			
3321	ADM3ATGS	ADM3ATPS		
3322	ADM3AEEH	ADM3ATGS	ADM3ATPS	
3323	ADM3ATGT			
3324	ADM3ATPE			
3325	ADM3ATGT	ADM3ATPT		
3331	ADM3ATGE	ADM3ATGT		
3333	ADM3AALC	ADM3AARC	ADM3AAWN	ADM3AMCP
	ADM3AMCV	ADM3AMER	ADM3AMEX	ADM3AMFR
	ADM3AMIV	ADM3AMMI	ADM3AMOR	ADM3AMRP
	ADM3AMSC	ADM3AMTM	ADM3ATGS	
3334	ADM3AARC	ADM3AAWN	ADM3AMCP	ADM3AMEX
3335	ADM3AARS			
3336	ADM3AMTM			
3337	ADM3AMCP	ADM3AMEX		
3338	ADM3ATGS	ADM3ATPI	ADM3ATPS	
3340	ADM3ATPS			
3341	ADM3ARO			
3342	ADM3AMCP	ADM3AMCV	ADM3AMEX	ADM3AQRQ
	ADM3ATGS			
3343	ADM3AMCP	ADM3AMEX		
3344	ADM3AASC			
3350	ADMD3SDE	ADM3IACR		
3351	ADMD3SLD	ADM3IAAR	ADM3IACL	ADM3IACR
	ADM3IADE	ADM3IAG	ADM3IAGE	ADM3IAQR
	ADM3IARE	ADM3IATR		
3352	ADMD3SDE	ADM3IACR	ADM3INIT	
3353	ADMD3SDE	ADM3IACR		
3354	ADMD3SDE	ADM3IACR	ADM3INIT	
3355	ADMD3SDE	ADMD3SRF	ADM3IAAR	ADM3IACR
	ADM3IRF			
3356	ADMD3DQR	ADMD3SDE	ADMD3SQR	ADMD3SRE
	ADM3IACR	ADM3IAQR	ADM3IARE	ADM3IQRY
	ADM3IRES	ADMM3IRS	ADMM3XRS	
3357	ADMD3SDE	ADMD3SRE	ADM3IACR	ADM3INIT
	ADM3IRES			
3358	ADMD3DIF	ADMD3SDL	ADMD3SLD	ADMD3SQR

Msg	Modules that issue the message			
	ADMD3SRE	ADMD3SRF	ADMD3STR	ADMD3SXF
	ADM3AMCP	ADM3IAAR	ADM3IACL	ADM3IADE
	ADM3IAQR	ADM3IARE	ADM3IATR	
3359	ADMD3SRF	ADM3IRF		
3360	ADM3IARE	ADM3IPLC	ADM3ISCL	ADM3PROJ
3361	ADM3IACL	ADM3INIT		
3362	ADMD3CIB	ADMD3DIF	ADMD3STR	ADM3IACL
	ADM3IATR	ADM3ICLR	ADM3ISUB	ADM3ITRM
	ADM3PROJ			
3363	ADMD3DIF	ADMD3STR	ADM3ICLR	ADM3ITRM
3364	ADM3IASV	ADM3PRST	ADM3PSAV	
3365	ADM3IASV	ADM3PSAV		
3366	ADM3IAGS	ADM3IAXF		
3367	ADM3IAGS	ADM3IAXF		
3368	ADMD3DIF	ADM3IGT	ADM3IGTE	ADM3IGTS
3369	ADMD3SSD	ADMD3SXF	ADM3ITGT	ADM3PAPF
	ADM3PAPT	ADM3PSMP		
3370	ADM3IAP	ADM3IAPE	ADM3IAPS	ADM3IAXF
3371	ADM3IAPS	ADM3IAXF	ADM3IPE2	ADM3IPS2
	ADM3IPT	ADM3IPTE		
3372	ADMD3SSD	ADMD3SXF	ADM3ITGT	
3373	ADM3IAGS	ADM3IAPS	ADM3IGTS	ADM3IPTS
3374	ADM3IAGS	ADM3IAPS	ADM3IGTS	ADM3IPTS
3375	ADM3IAGS	ADM3IAPS		
3376	ADMD3DIF	ADM3IAPS		
3377	ADM3IAPS			
3378	ADMD3DIF	ADMD3SXF	ADM3IAGS	ADM3IAPS
	ADM3IGTS	ADM3IPS2		
3379	ADMD3DIF	ADM3ATFT	ADM3IAG	ADM3IAP
	ADM3IGT	ADM3IPT		
3380	ADMD3DIF	ADM3AEEH	ADM3ATBF	ADM3ATFM
	ADM3ATIF	ADM3ATIM	ADM3ATMF	ADM3ATPI
	ADM3ATPT	ADM3ATTF		
3381	ADMD3DIF	ADM3IAP	ADM3IAPE	ADM3IPE2
	ADM3IPT	ADM3IPTE		
3382	ADMD3DIF	ADM3ATPT	ADM3IPT	
3383	ADM3ATPE			
3384	ADMD3DIF	ADMD3SXF	ADM3IAG	ADM3IAGE
	ADM3IGT	ADM3IGTE		
3385	ADM3IAG			
3386	ADMDJC	ADMDKC	ADMDLC	ADMDOCO
	ADMDPC	ADMDTC	ADMDWC	ADMD3DIF
	ADMD3DPG	ADMD3SXF	ADM3IACL	ADM3IATR
	ADM3IAXF	ADM3ICLR	ADM3IDEL	ADM3INEG
	ADM3IORN	ADM3IPLC	ADM3IQRY	ADM3IREF
	ADM3IRES	ADM3ISCL	ADM3ISUB	ADM3ITRM
3387	ADMD3SSD	ADMD3SXF	ADM3ITFR	ADM3PSMP
3388	ADM3IAGS	ADM3IGTS		
3389	ADMD3DIF	ADM3ATPE		
3390	ADM3IAGS			
3391	ADM3DCAN			
3392	ADM3DCAN			
3393	ADM3DCAN			
3394	ADM3DCAN			
3400	ADM3PCRT			
3401	ADM3IAGS	ADM3IAPS	ADM3IAXF	ADM3PCRT
	ADM3PDEL	ADM3PRST	ADM3PSAV	ADM3TREL
3402	ADMD3SBX	ADM3PDEL	ADM3PSAV	ADM3TREL
3403	ADMD3DIF	ADMD3SBX	ADM3XFER	
3404	ADM3PROJ			
3405	ADM3PROJ			
3406	ADM3PROJ			
3407	ADM3PROJ			
3408	ADM3PROJ			

Msg	Modules that issue the message			
3409	ADM3IPLC	ADM3PROJ		
3410	ADM3ISCL	ADM3PROJ		
3411	ADM3IORN	ADM3PROJ		
3412	ADM3IREF	ADM3PROJ		
3413	ADM3PROJ			
3414	ADM3PROJ			
3415	ADM3PROJ			
3450	ADMD3DPG			
3451	ADMD3DPG			
3452	ADMD3DPG			
3453	ADMD3CEN	ADMD3DPG	ADMD3DQR	ADMD3SES
3454	ADMD3DPL			
3455	ADMD3DPG	ADMD3DQC	ADMD3DQF	ADMD3DQR
3456	ADMD3DPG			
3457	ADMD3DPG			
3458	ADMD3DPG			
3459	ADMD3DQC	ADMD3DQF		
3461	ADMDXRIO	ADMD3DIF		
3462	ADMD3DIS			
3463	ADMD3DPL	ADMOQPU		
3470	ADMD3DQR	ADMD3SDE	ADMD3SDL	ADMD3SES
	ADMD3SLD	ADMD3SQR	ADMD3SQS	ADMD3SRE
	ADMD3SRF	ADMD3STR	ADMD3SXF	
3471	ADMD3SRF			
3472	ADMD3SRF			
3473	ADMD3SQS			
3474	ADMD3SDE	ADMD3SRE		
3475	ADMD3SDE			
3476	ADMD3SDE			
3477	ADMD3SDS			
3478	ADMD3SLD	ADMD3SPX	ADMD3SXF	
3479	ADMD3SDS			
3480	ADMD3SDS			
3481	ADMD3SDS			
3482	ADMD3SGS			
3483	ADMD3SLD	ADMD3SPX	ADMD3SXF	
3484	ADMD3SSD	ADMD3SXF		
3490	ADMD3CEN			
3491	ADMD3CIB	ADMD3CIL		
3492	ADMD3CIB			
3493	ADMD3CIB			
3494	ADMD3CIB			
3495	ADMD3CEN			
3497	ADMD3CIL			
3498	ADMD3CEN	ADMD3CIL	ADMD3CQL	
3499	ADMD3CEN	ADMD3CIB	ADMD3CQB	
3500	ADMJCPF1			
3501	ADMJCP00			
3502	ADMJCP01			
3503	ADMJCP05	ADMJCP38		
3504	ADMJCPA6	ADMJCP39		
3505	ADMJCP0C	ADMJCP0D	ADMJCP0E	ADMJCP0F
	ADMJCP10	ADMJCP11	ADMJCP3F	ADMJCP65
3506	ADMJCP04	ADMJCP06	ADMJCP3D	ADMJCP3E
3507	ADMJCPAA	ADMJCPAB	ADMJCPAC	ADMJCPAD
	ADMJCPAE	ADMJCPAF	ADMJCPA7	ADMJCPA8
	ADMJCPA9	ADMJCPBA	ADMJCPBB	ADMJCPBC
	ADMJCPBD	ADMJCPBE	ADMJCPBF	ADMJCPB0
	ADMJCPB1	ADMJCPB2	ADMJCPB3	ADMJCPB4
	ADMJCPB5	ADMJCPB6	ADMJCPB7	ADMJCPB8
	ADMJCPB9	ADMJCPC0	ADMJCPDE	ADMJCPDF
	ADMJCP0A	ADMJCP03	ADMJCP07	ADMJCP08
	ADMJCP09	ADMJCP2B	ADMJCP2C	ADMJCP2D

Msg	Modules that issue the message			
	ADMJCP2E	ADMJCP2F	ADMJCP3A	ADMJCP3B
	ADMJCP3C	ADMJCP30	ADMJCP36	ADMJCP37
	ADMJCP4A	ADMJCP4B	ADMJCP4C	ADMJCP4D
	ADMJCP4E	ADMJCP4F	ADMJCP40	ADMJCP41
	ADMJCP42	ADMJCP43	ADMJCP44	ADMJCP45
	ADMJCP46	ADMJCP47	ADMJCP48	ADMJCP49
	ADMJCP5A	ADMJCP5B	ADMJCP5C	ADMJCP5D
	ADMJCP5E	ADMJCP5F	ADMJCP50	ADMJCP51
	ADMJCP52	ADMJCP53	ADMJCP54	ADMJCP55
	ADMJCP56	ADMJCP57	ADMJCP58	ADMJCP59
	ADMJCP60	ADMJCP61	ADMJCP62	ADMJCP63
	ADMJCP64	ADMJCP66	ADMJCP67	
3508	ADMJCPA0	ADMJCPA1	ADMJCPA2	ADMJCPA3
	ADMJCPA4	ADMJCPA5	ADMJCPCA	ADMJCPCB
	ADMJCPCC	ADMJCPCD	ADMJCPCE	ADMJCPCF
	ADMJCPC1	ADMJCPC2	ADMJCPC3	ADMJCPC4
	ADMJCPC5	ADMJCPC6	ADMJCPC7	ADMJCPC8
	ADMJCPC9	ADMJCPDA	ADMJCPDB	ADMJCPDC
	ADMJCPDD	ADMJCPD0	ADMJCPD1	ADMJCPD2
	ADMJCPD3	ADMJCPD4	ADMJCPD5	ADMJCPD6
	ADMJCPD7	ADMJCPD8	ADMJCPD9	ADMJCPUB
	ADMJCPUC	ADMJCP0B	ADMJCP02	ADMJCP1A
	ADMJCP1B	ADMJCP1C	ADMJCP1D	ADMJCP1E
	ADMJCP1F	ADMJCP12	ADMJCP13	ADMJCP14
	ADMJCP15	ADMJCP16	ADMJCP17	ADMJCP18
	ADMJCP19	ADMJCP2A	ADMJCP20	ADMJCP21
	ADMJCP22	ADMJCP23	ADMJCP24	ADMJCP25
	ADMJCP26	ADMJCP27	ADMJCP28	ADMJCP29
	ADMJCP31	ADMJCP32	ADMJCP33	ADMJCP34
	ADMJCP35	ADMJCP6A	ADMJCP68	ADMJCP69
	ADMJCP8A	ADMJCP8B	ADMJCP8C	ADMJCP8D
	ADMJCP8E	ADMJCP8F	ADMJCP81	ADMJCP82
	ADMJCP83	ADMJCP84	ADMJCP85	ADMJCP86
	ADMJCP87	ADMJCP88	ADMJCP89	ADMJCP9A
	ADMJCP9B	ADMJCP9C	ADMJCP9D	ADMJCP9E
	ADMJCP9F	ADMJCP90	ADMJCP91	ADMJCP92
	ADMJCP93	ADMJCP94	ADMJCP95	ADMJCP96
	ADMJCP97	ADMJCP98	ADMJCP99	
3520	ADMJCPAA	ADMJCPAB	ADMJCPAC	ADMJCPAD
	ADMJCPAE	ADMJCPAF	ADMJCPA9	ADMJCPBA
	ADMJCPBB	ADMJCPBC	ADMJCPBD	ADMJCPBE
	ADMJCPBF	ADMJCPB0	ADMJCPB1	ADMJCPB2
	ADMJCPB3	ADMJCPB4	ADMJCPB5	ADMJCPB6
	ADMJCPB7	ADMJCPB8	ADMJCPB9	ADMJCPC0
	ADMJCP0A	ADMJCP02	ADMJCP03	ADMJCP04
	ADMJCP05	ADMJCP06	ADMJCP07	ADMJCP08
	ADMJCP09	ADMJCP2B	ADMJCP2C	ADMJCP2D
	ADMJCP2E	ADMJCP2F	ADMJCP3C	ADMJCP3D
	ADMJCP3E	ADMJCP30	ADMJCP36	ADMJCP37
	ADMJCP4A	ADMJCP4B	ADMJCP4C	ADMJCP4D
	ADMJCP4E	ADMJCP4F	ADMJCP45	ADMJCP46
	ADMJCP47	ADMJCP48	ADMJCP49	ADMJCP50
	ADMJCP51	ADMJCP52	ADMJCP53	ADMJCP54
	ADMJCP55	ADMJCP56	ADMJCP65	ADMJCP66
	ADMJCP67			
3521	ADMJCP02	ADMJMIEC	ADMJMOEC	
3522	ADMJCPCA	ADMJCPCB	ADMJCPCC	ADMJCPCD
	ADMJCPCE	ADMJCPCF	ADMJCPC1	ADMJCPC2
	ADMJCPC3	ADMJCPC4	ADMJCPC5	ADMJCPC6
	ADMJCPC7	ADMJCPC8	ADMJCPC9	ADMJCPDA
	ADMJCPDB	ADMJCPDC	ADMJCPDD	ADMJCPD0
	ADMJCPD1	ADMJCPD2	ADMJCPD3	ADMJCPD4
	ADMJCPD5	ADMJCPD6	ADMJCPD7	ADMJCPD8
	ADMJCPD9	ADMJCP02		
3523	ADMJCPCA	ADMJCPCB	ADMJCPCC	ADMJCPCD
	ADMJCPCE	ADMJCPCF	ADMJCPC1	ADMJCPC2
	ADMJCPC3	ADMJCPC4	ADMJCPC5	ADMJCPC6
	ADMJCPC7	ADMJCPC8	ADMJCPC9	ADMJCPDA
	ADMJCPDB	ADMJCPDC	ADMJCPDD	ADMJCPD0
	ADMJCPD1	ADMJCPD2	ADMJCPD3	ADMJCPD4
	ADMJCPD5	ADMJCPD6	ADMJCPD7	ADMJCPD8
	ADMJCPD9	ADMJCP02		
3524	ADMJCP02			
3525	ADMJCPAA	ADMJCPAB	ADMJCPAC	ADMJCPAD
	ADMJCPAE	ADMJCPAF	ADMJCPA9	ADMJCPBA
	ADMJCPBB	ADMJCPBC	ADMJCPBD	ADMJCPBE
	ADMJCPBF	ADMJCPB0	ADMJCPB1	ADMJCPB2
	ADMJCPB3	ADMJCPB4	ADMJCPB5	ADMJCPB6
	ADMJCPB7	ADMJCPB8	ADMJCPB9	ADMJCPC0
	ADMJCP0A	ADMJCP03	ADMJCP04	ADMJCP06
	ADMJCP07	ADMJCP08	ADMJCP09	ADMJCP2B
	ADMJCP2C	ADMJCP2D	ADMJCP2E	ADMJCP2F
	ADMJCP3C	ADMJCP3D	ADMJCP3E	ADMJCP30
	ADMJCP36	ADMJCP37	ADMJCP4A	ADMJCP4B
	ADMJCP4C	ADMJCP4D	ADMJCP4E	ADMJCP4F
	ADMJCP45	ADMJCP46	ADMJCP47	ADMJCP48
	ADMJCP49	ADMJCP50	ADMJCP51	ADMJCP52
	ADMJCP53	ADMJCP54	ADMJCP55	ADMJCP56
	ADMJCP65	ADMJCP66	ADMJCP67	
3526	ADMJCP02	ADMJD1EC	ADMJMIBP	ADMJMIEC
	ADMJMOBP	ADMJMOEC		
3527	ADMJCP3D	ADMJCP3E	ADMJCP3F	
3528	ADMJCP02			
3529	ADMJCP03	ADMJCP04		
3530	ADMJCP05			
3531	ADMJCPC3			
3532	ADMJCP65			
3533	ADMJCPAA	ADMJCPAB	ADMJCPAC	ADMJCPAD
	ADMJCPAE	ADMJCPAF	ADMJCPBA	ADMJCPB0
	ADMJCPB1	ADMJCPB3	ADMJCPB4	ADMJCPB5
	ADMJCPB6	ADMJCPB7	ADMJCPB8	ADMJCPB9
	ADMJCPC3	ADMJCP04	ADMJCP05	ADMJCP06
	ADMJCP07	ADMJCP08	ADMJCP09	ADMJCP2B
	ADMJCP2C	ADMJCP2D	ADMJCP2E	ADMJCP2F
	ADMJCP3C	ADMJCP3D	ADMJCP3E	ADMJCP30
	ADMJCP36	ADMJCP37		
3534	ADMJCP66	ADMJCP67		
3535	ADMJCPAA	ADMJCPAB	ADMJCPAC	ADMJCPAD
	ADMJCPAE	ADMJCPAF	ADMJCPBA	ADMJCPB0
	ADMJCPB1	ADMJCPB3	ADMJCPB4	ADMJCPB5
	ADMJCPB6	ADMJCPB7	ADMJCPB8	ADMJCP04
	ADMJCP05	ADMJCP06	ADMJCP07	ADMJCP08
	ADMJCP09	ADMJCP2B	ADMJCP2C	ADMJCP2D
	ADMJCP2E	ADMJCP2F	ADMJCP3C	ADMJCP3D
	ADMJCP3E	ADMJCP30		
3536	ADMJCPAB	ADMJCPAC	ADMJCPAD	ADMJCPAE
	ADMJCPAF	ADMJCPB0	ADMJCPB1	ADMJCPB3
	ADMJCPB4	ADMJCPB5	ADMJCPB6	ADMJCPB7
	ADMJCPB8	ADMJCPB9	ADMJCPC3	ADMJCP0A
	ADMJCP07	ADMJCP08	ADMJCP09	ADMJCP2B
	ADMJCP2C	ADMJCP2D	ADMJCP2E	ADMJCP2F
	ADMJCP3E	ADMJCP30	ADMJCP36	ADMJCP37
3537	ADMJCPBF	ADMJCP4F	ADMJCP49	ADMJCP55
3538	ADMJCPBB	ADMJCPBC	ADMJCPBD	ADMJCPBE
	ADMJCPC0	ADMJCPDA	ADMJCPDB	ADMJCPDC
	ADMJCPDD	ADMJCPD7	ADMJCPD8	ADMJCPD9
	ADMJCP4A	ADMJCP4B	ADMJCP4C	ADMJCP4D
	ADMJCP4E	ADMJCP45	ADMJCP46	ADMJCP47
	ADMJCP48	ADMJCP50	ADMJCP51	ADMJCP52
	ADMJCP53	ADMJCP54	ADMJCP56	
3539	ADMJCPB2	ADMJCPCA	ADMJCPCB	ADMJCPCC
	ADMJCPCD	ADMJCPCE	ADMJCPCF	ADMJCPC2
	ADMJCPC4	ADMJCPC5	ADMJCPC6	ADMJCPC7
	ADMJCPC8	ADMJCPC9	ADMJCPD0	ADMJCPD1
	ADMJCPD2	ADMJCPD3	ADMJCPD4	ADMJCPD5
	ADMJCPD6			

Msg	Modules that issue the message			
3540	ADMJCPE0	ADMJCPE1	ADMJCPE2	
3541	ADMJCPD3			
3542	ADMJCP02			
3550	ADMJCPA4	ADMJCP31	ADMJCP32	ADMJCP33
	ADMJCP34			
3551	ADMJCP31	ADMJCP32	ADMJCP36	ADMJCP37
	ADMJCP4A	ADMJCP45	ADMJCP46	ADMJCP47
	ADMJCP48	ADMJCP49		
3552	ADMJCP32			
3553	ADMJCP36			
3554	ADMJCP37			
3560	ADMJCPAD	ADMJCPC7	ADMJCP12	ADMJCP2B
	ADMJD1I1	ADMJD1I2		
3561	ADMJCPAD			
3562	ADMJCPC7			
3563	ADMJCP13	ADMJCP2B	ADMJD1I1	ADMJD1I2
3564	ADMJCP2B			
3565	ADMJCP14	ADMJCP2B	ADMJD1I1	ADMJD1I2
3566	ADMJCPAF	ADMJCPC9	ADMJCP16	ADMJCP2C
	ADMJD1I2			
3567	ADMJCPAF			
3568	ADMJCPC9			
3569	ADMJCP17	ADMJCP2C	ADMJD1I2	
3570	ADMJCP2C			
3571	ADMJCP18	ADMJCP2C	ADMJD1I2	
3572	ADMJCPB1	ADMJCPCB	ADMJCP1A	ADMJCP2D
3573	ADMJCPB1			
3574	ADMJCPCB			
3575	ADMJCP1B	ADMJCP2D		
3576	ADMJCP2D			
3577	ADMJCP1C	ADMJCP2D		
3578	ADMJCP1F			
3579	ADMJCP20			
3580	ADMJCPB4	ADMJCPCD	ADMJCP2E	ADMJCP23
	ADMJD1I1			
3581	ADMJCPB4			
3582	ADMJCPCD			
3583	ADMJCP2E			
3584	ADMJCP25	ADMJD1I1		
3585	ADMJCPB6	ADMJCPCF	ADMJCP2E	ADMJCP2F
3586	ADMJCP2E			
3587	ADMJCP27			
3588	ADMJCPB6			
3589	ADMJCPCF			
3590	ADMJCPB6	ADMJCPCF	ADMJCP2F	
3591	ADMJCP10	ADMJCP2F	ADMJCP68	
3592	ADMJCP1E	ADMJCP15	ADMJCP19	ADMJCP26
	ADMJD1I1	ADMJD1I2		
3593	ADMJCPB8	ADMJCPD1	ADMJCP2B	ADMJCP2C
	ADMJCP2D	ADMJCP2E	ADMJCP2F	ADMJCP30
3594	ADMJCPB8			
3595	ADMJCPD1			
3596	ADMJCP30			
3597	ADMJCP2A			
3600	ADMJCP0C	ADMJCP0D	ADMJCP0F	ADMJCP68
3601	ADMJCP0E	ADMJD1P8		
3604	ADMJCP11			
3620	ADMJCPDE	ADMJCPDF	ADMJCP3A	ADMJCP3B
	ADMJCP3C	ADMJCP3D	ADMJCP3E	ADMJCP3F
	ADMJCP38	ADMJCP40	ADMJCP41	ADMJCP42
	ADMJCP43	ADMJCP44	ADMJD1I1	
3621	ADMJCP3A	ADMJCP38		

Msg	Modules that issue the message			
3622	ADMJCPDE	ADMJCPDF	ADMJCP3A	ADMJCP3B
	ADMJCP40	ADMJCP41	ADMJCP42	ADMJCP43
	ADMJCP44			
3623	ADMJCP3C	ADMJD1R1		
3624	ADMJCP3D	ADMJCP3E	ADMJCP3F	
3625	ADMJCP3B	ADMJCP3C	ADMJCP3F	
3626	ADMJCP43			
3640	ADMJCPBB	ADMJCPBC	ADMJCPBD	ADMJCPBE
	ADMJCPBF	ADMJCPC0	ADMJCPDA	ADMJCPDB
	ADMJCPDC	ADMJCPDD	ADMJCPD8	ADMJCPD9
	ADMJCP4A	ADMJCP4B	ADMJCP4C	ADMJCP4D
	ADMJCP4E	ADMJCP4F	ADMJCP45	ADMJCP46
	ADMJCP47	ADMJCP48	ADMJCP49	ADMJCP50
	ADMJCP51	ADMJCP52	ADMJCP53	ADMJCP54
	ADMJCP55	ADMJCP56		
3641	ADMJCP4A	ADMJCP45	ADMJCP46	ADMJCP47
	ADMJCP48	ADMJCP49	ADMJCP51	ADMJCP52
	ADMJCP53	ADMJCP54	ADMJCP55	ADMJCP56
3643	ADMJCPA8	ADMJCPE3	ADMJCP4B	ADMJCP4C
	ADMJCP4D	ADMJCP4E	ADMJCP4F	ADMJCP5A
	ADMJCP5B	ADMJCP5C	ADMJCP5D	ADMJCP5E
	ADMJCP5F	ADMJCP50	ADMJCP57	ADMJCP58
	ADMJCP59	ADMJCP60	ADMJCP61	ADMJCP62
	ADMJCP63	ADMJCP64		
3644	ADMJCP4A	ADMJCP45	ADMJCP46	ADMJCP47
	ADMJCP48	ADMJCP49		
3645	ADMJCP4A	ADMJCP45	ADMJCP46	ADMJCP47
	ADMJCP48	ADMJCP49		
3646	ADMJCP47	ADMJD1I1	ADMJD1I2	
3652	ADMJCP4A	ADMJCP45	ADMJCP46	ADMJCP47
	ADMJCP48	ADMJCP49		
3653	ADMJCP46			
3654	ADMJCP4A			
3660	ADMJCP65			
3661	ADMJCP65	ADMJCP68		
3662	ADMJCP66	ADMJCP67		
3663	ADMJCP66	ADMJCP67	ADMJCP68	
3664	ADMJCP68			
3665	ADMJCPUA	ADMJCP68		
3666	ADMJCP67			
3667	ADMJCP68			
3680	ADMJCP0B			
3700	ADMJCP00			
3800	ADMJCPB6	ADMJCPUA	ADMJCPUB	ADMJCPUC
	ADMJCPU1	ADMJCPU4	ADMJCP0C	ADMJCP0D
	ADMJCP0E	ADMJCP0F	ADMJCP00	ADMJCP02
	ADMJCP10	ADMJCP2F	ADMJCP4A	ADMJCP46
	ADMJCP52	ADMJCP65	ADMJD1EC	ADMJD1S1
	ADMJMAIN	ADMJMIBP	ADMJMIEC	ADMJMOBP
	ADMJMOEC	ADMJWIBP		
3801	ADMJCPUA	ADMJCPUB	ADMJCPUC	ADMJCP0C
	ADMJCP0D	ADMJCP0E	ADMJCP0F	ADMJCP1A
	ADMJCP1B	ADMJCP1C	ADMJCP1D	ADMJCP1E
	ADMJCP1F	ADMJCP10	ADMJCP11	ADMJCP12
	ADMJCP13	ADMJCP14	ADMJCP15	ADMJCP16
	ADMJCP17	ADMJCP18	ADMJCP19	ADMJCP2A
	ADMJCP20	ADMJCP21	ADMJCP22	ADMJCP23
	ADMJCP24	ADMJCP25	ADMJCP26	ADMJCP27
	ADMJCP28	ADMJCP29	ADMJCP3F	ADMJCP31
	ADMJCP32	ADMJCP34	ADMJCP35	ADMJCP38
	ADMJCP65	ADMJD1P6	ADMJD1U4	ADMJWIBP
3802	ADMJMIBP			
3803	ADMJCP00	ADMJMOBP		
3804	ADMJD1BP	ADMJD1P2	ADMJD1R3	ADMJD1R6

Msg	Modules that issue the message			
	ADMJD1S1			
3805	ADMJD1R1	ADMJD1R2	ADMJD1R3	ADMJD1R4
	ADMJD1R5	ADMJD1R6		
3806	ADMJCP00	ADMJD1U3		
3807	ADMJCP00			
3808	ADMJCONV	ADMJCP0C	ADMJCP0D	ADMJCP0E
	ADMJCP0F	ADMJCP10	ADMJCP11	ADMJCP18
	ADMJCP2C	ADMJCP68		
3900	ADMJCPAD	ADMJCPAF	ADMJCPBB	ADMJCPBC
	ADMJCPBF	ADMJCPB1	ADMJCPB4	ADMJCPB6
	ADMJCPB8	ADMJCP06	ADMJCP08	ADMJCP09
	ADMJCP1B	ADMJCP2E	ADMJCP21	ADMJCP22
	ADMJCP24	ADMJCP29	ADMJCP33	ADMJCP35
	ADMJCP4B	ADMJCP4C	ADMJCP4D	ADMJCP4E
	ADMJCP4F	ADMJCP41	ADMJCP42	ADMJCP44
	ADMJCP48	ADMJCP49	ADMJCP50	ADMJCP6A
	ADMJCP69	ADMJD1I1	ADMJD1I2	
3901	ADMJCP0A	ADMJCP0E	ADMJCP6B	ADMJCP6C
	ADMJCP67			
3902	ADMJCPAC	ADMJCPAE	ADMJCPA3	ADMJCPA7
	ADMJCPBA	ADMJCPB0	ADMJCPB3	ADMJCPB5
	ADMJCPB7	ADMJCPCA	ADMJCPCC	ADMJCPC6
	ADMJCPC8	ADMJCPDA	ADMJCPDB	ADMJCPDC
	ADMJCPDD	ADMJCPDE	ADMJCPD2	ADMJCPD8
	ADMJCPD9	ADMJCP82	ADMJCP85	ADMJCP86
3903	ADMJCP6B	ADMJCP6C	ADMJCP68	
3999	ADMJCPU1	ADMJCP06	ADMJCP2E	ADMJCP3A
	ADMJCP4A	ADMJCP40	ADMJCP41	ADMJCP42
	ADMJCP43	ADMJCP44	ADMJCP46	ADMJD1BP
	ADMJD1EC	ADMJD1I1	ADMJD1I2	ADMJD1P1
	ADMJD1P2	ADMJD1P3	ADMJD1P6	ADMJD1P7
	ADMJD1P8	ADMJD1R2	ADMJD1R4	ADMJD1R5
	ADMJD1S1	ADMJD1S3	ADMJD1U2	ADMJD1U3
	ADMJD1U4	ADMJERQM	ADMJWIBP	
4000	ADM5IV			
4001	ADM5CUR	ADM5IV	ADM5SHN	
4002	ADM5PFK			
4003	ADM5IV			
4004	ADM5SHN			
4005	ADM5IV			
4006	ADM5DC	ADM5ED	ADM5EDD	ADM5EDE
	ADM5EDF	ADM5EDT	ADM5EX	ADM5IM
	ADM5IM1	ADM5IP	ADM5LD	ADM5LDR
	ADM5OP	ADM5PJ	ADM5PJD	ADM5PJL
	ADM5PJS	ADM5PJ1	ADM5PJ2	ADM5PJ3
	ADM5PR	ADM5SC	ADM5SCP	ADM5SI
	ADM5VW			
4007	ADM5EDT	ADM5PJ3		
4008	ADM5PJ1			
4009	ADM5PJ1			
4010	ADM5ERR	ADMPJ3		
4011	ADM5EDC	ADM5EDF	ADM5PJ1	
4012	ADM5EDC	ADM5EDF	ADM5PJ1	
4013	ADM5IM	ADM5IP	ADM5LD	ADM5NMA
	ADM5SC			
4014	ADM5EDF	ADM5PR	ADM5VW	
4015	ADM5NMA	ADM5PJD	ADM5PJS	
4016	ADM5ED	ADM5IM	ADM5IP	ADM5LD
	ADM5NMA	ADM5SC		
4017	ADM5EX	ADM5IM	ADM5IP	ADM5LD
	ADM5NMA	ADM5PJL	ADM5PJ	ADM5PR
	ADM5SC	ADM5SI	ADM5VW	
4018	ADM5ED	ADM5EX	ADM5IM	ADM5IP
	ADM5LD	ADM5PJ	ADM5PJC	ADM5PJL
	ADM5PR	ADM5SC	ADM5SI	ADM5VW
4019	ADM5DC	ADM5EX	ADM5SI	ADM5ED
4020	ADM5ERR	ADM5IM		
4021	ADM5ERR	ADM5PR4		
4022	ADM5SC			
4023	ADM5SC			
4024	ADM5SC			
4025	ADM5IP	ADM5LD		
4026	ADM5IM			
4027	ADM5SI			
4028	ADM5PR			
4029	ADM5EX			
4030	ADM5DC			
4031	ADM5IV			
4032	ADM5SC			
4033	ADM5ERR	ADM5SC		
4034	ADM5SCP			
4035	ADM5MAIN	ADM5NMA		
4036	ADM5IV			
4037	ADM5ED	ADM5IV	ADM5IVP	ADM5NMA
4038	ADM5NMA			
4039	ADM5IV	ADM5IVP	ADM5NMA	
4040	ADM5NMA			
4041	ADM5PFS			
4042	ADM5PFS			
4043	ADM5NMA	ADM5PFS		
4044	ADM5NMA			
4045	ADM5PJC			
4046	ADM5NMA			
4047	ADM5NMA			
4048	ADM5NMA			
4049	ADM5NMA	ADM5PJ	ADM5PJL	
4050	ADM5NMA			
4051	ADM5NMA			
4052	ADM5PFK			

Msg	Modules that issue the message			
4053	ADM5SCP			
4054	ADM5SCP			
4055	ADM5PJL			
4056	ADM5PJS			
4057	ADM5CUR			
4058	ADM5CUR			
4059	ADM5CUR			
4060	ADM5CUR			
4061	ADM5EDC	ADM5EDE	ADM5EDF	ADM5EDT
4062	ADM5CUR			
4063	ADM5PJC			
4064	ADM5IP	ADM5IV		
4065	ADM5IV	ADM5OP		
4066	ADM5MAIN	ADM5SHN		
4067	ADM5PFK			
4068	ADM5LDR			
4069	ADM5ED	ADM5EDF	ADM5LD	ADM5NMA
	ADM5PJC	ADM5PR	ADM5SI	ADM5VW
4070	ADM5PJ3			
4071	ADM5ED	ADM5EDD	ADM5EDE	ADM5EDF
	ADM5EDT			
4072	ADM5CUR	ADM5EDC	ADM5EDE	ADM5EDF
	ADM5EDT			
4073	ADM5EX	ADM5IM		
4074	ADM5PFS			
4075	ADM5ERR			
4076	ADM5PJD			
4077	ADM5NMA			
4078	ADM5PR4			
4079	ADM5PFK			
4080	ADM5PJ3			
4081	ADM5PJ	ADM5PJC		
4082	ADM5CUR			
4083	ADM5ERR			
4084	ADM5PJC	ADM5SH		
4085	ADM5SH			
4086	ADM5PFK			
4087	ADM5PJC			
4088	ADM5ED	ADM5EDD	ADM5EDE	ADM5EDF
	ADM5EDT			
4089	ADM5IM			
4090	ADM5ED	ADM5EDD	ADM5EDE	ADM5EDF
	ADM5EDT			
4091	ADM5PFK			
4092	ADM5ED			
4093	ADM5PR			
4094	ADM5CUR			

GDDM-IMD messages

Msg	Modules that issue the message			
00011	AEMIOS01			
00012	AEMGRP01	AEMGRP02	AEMGRP08	AEMGRP09
	AEMINT00	AEMLIB07	AEMMAP01	AEMMAP02
	AEMMAP08	AEMMAP09	AEMPSG01	AEMUTY00
00014	AEMGRP00	AEMGRP03	AEMGRP06	AEMLIB03
	AEMLIB04	AEMLIB07	AEMMAP00	AEMMAP03
	AEMPSG00	AEMSRV70	AEMTBL00	AEMUTY01
	AEMUTY02	AEMUTY03		
00019	AEMPSG03			
00022	AEMDIA80	AEMGRP03	AEMGRP05	AEMMAP03
	AEMMAP10	AEMTBL01	AEMUTY03	AEMMAP67
	AEMPSG00	AEMPSG05		
00023	AEMGRP00	AEMGRP05	AEMLIB03	AEMLIB04
	AEMLIB07	AEMMAP00	AEMMAP67	AEMPSG00
	AEMTBL00	AEMUTY01	AEMUTY02	
00025	AEMSTG00			
00029	AEMLIB04	AEMLIB07	AEMLIB82	AEMLIB84
	AEMPSG00	AEMUTY01	AEMUTY02	
00030	AEMINT02			
00031	AEMUTY03			
00032	AEMMAP03	AEMPSG00		
00033	AEMINT02			
00034	AEMINT02			
00035	AEMPSG01	AEMPSG05		
00037	AEMMAP67			
00038	AEMPSG95			
00039	AEMMAP90			
00040	AEMUTY01			
00041	AEMTUT80			
00042	AEMTUT82			
00043	AEMTUT81			
00044	AEMIOS05			
00048	AEMUTY01			
00049	AEMMAP28			
00050	AEMMAP28			
00052	AEMGRP03	AEMMAP03	AEMMAP10	AEMMAP56
	AEMMAP67	AEMPSG05		
00053	AEMLIB07			
00055	AEMMAP45			
00057	AEMMAP81			
00058	AEMMAP29	AEMMAP45		
00059	AEMMAP29			
00060	AEMGRP04	AEMGRP08	AEMGRP82	
00063	AEMMAP77	AEMMAP82	AEMMAP90	AEMMAP92
	AEMMAP93			
00065	AEMMAP73			
00066	AEMMAP02	AEMMAP80	AEMPSG00	
00068	AEMMAP02			
00069	AEMGRP09	AEMMAP09		
00079	AEMGRP04			
00080	AEMMSL00			
00081	AEMMSL00			
00082	AEMMSL00	AEMMSL01		
00083	AEMMSL00	AEMMSL02		
00084	AEMMSL00	AEMMSL03	AEMMSL04	AEMMSL05
	AEMMSL06			
00085	AEMMSL03	AEMMSL04	AEMMSL05	AEMMSL06
00086	AEMMSL00	AEMMSL02	AEMMSL83	
00087	AEMGRP00	AEMGRP01	AEMGRP05	AEMGRP82
	AEMGRP83	AEMGRP85	AEMGRP87	AEMMAP02
	AEMMAP04	AEMMAP27	AEMMAP28	AEMMAP29

Msg	Modules that issue the message			
	AEMMAP80	AEMMAP81	AEMMAP84	AEMMAP87
	AEMMSL77	AEMMSL78	AEMMSL79	AEMMSL87
	AEMPSG00	AEMPSG04	AEMPSG88	AEMPSG99
	AEMTBL01	AEMTBL87	AEMTBL88	
00088	AEMMSL86			
00089	AEMUTY03			
00090	AEMDSP00			
00091	AEMDSP05			
00095	AEMMSL00			
00097	AEMIOS01	AEMMSL00	AEMMSL01	AEMMSL03
	AEMMSL04	AEMMSL05	AEMMSL06	AEMMSL07
	AEMMSL77	AEMMSL78	AEMMSL79	AEMMSL82
	AEMMSL84	AEMMSL85	AEMMSL86	AEMMSL87
	AEMPAR81			
00098	AEMMAP10	AEMMAP43	AEMMAP59	AEMMAP60
	AEMMAP66	AEMMAP69		
00099	AEMGRP00	AEMGRP01	AEMGRP02	AEMGRP81
	AEMMAP00	AEMMAP01	AEMMAP65	AEMMSL77
	AEMMSL78	AEMPSG01	AEMSRV80	AEMSRV83
	AEMUTY01	AEMUTY86		
00102	AEMGRP00	AEMMAP00	AEMPSG00	
00103	AEMGRP00	AEMMAP00		
00104	AEMMAP00			
00105	AEMGRP83	AEMMAP00	AEMTBL00	
00106	AEMGRP00	AEMMAP00	AEMTBL00	
00107	AEMGRP03	AEMGRP05	AEMGRP09	AEMMAP03
	AEMMAP04	AEMMAP05	AEMMAP07	AEMMAP09
	AEMMAP10			
00109	AEMPMS80			
00110	AEMMAP03	AEMMAP83		
00111	AEMSRV64			
00113	AEMPSG73	AEMPSG81	AEMUTY01	
00115	AEMGRP08	AEMMAP08	AEMTBL01	
00116	AEMGRP01	AEMMAP01		
00117	AEMMAP61			
00118	AEMMAP41	AEMMAP45		
00119	AEMMAP41			
00120	AEMMAP05			
00121	AEMMAP05			
00122	AEMMAP05			
00123	AEMMAP59			
00124	AEMMAP81			
00125	AEMMAP81			
00126	AEMMAP42			
00127	AEMMAP42			
00128	AEMMAP78			
00130	AEMPSG99			
00131	AEMSRV64			
00132	AEMSRV64			
00133	AEMSRV64			
00134	AEMSRV64			
00135	AEMMAP67	AEMSRV64		
00136	AEMSRV63			
00137	AEMSRV63			
00138	AEMUTY03			
00139	AEMUTY03			
00141	AEMMAP29	AEMPAR80	AEMSRV90	
00142	AEMMAP77	AEMMAP78	AEMMAP82	AEMMAP92
	AEMMAP94	AEMMAP98	AEMPAR80	
00143	AEMPSG84			
00144	AEMMAP29	AEMPAR80		
00145	AEMMAP07	AEMMAP55	AEMMAP72	AEMMAP74
	AEMMAP79	AEMMAP95		
00146	AEMMAP95	AEMMAP96		

Msg	Modules that issue the message			
00147	AEMMAP78	AEMPAR80	AEMPAR90	
00148	AEMMAP07	AEMMAP97		
00149	AEMMAP04			
00150	AEMMAP77	AEMMAP94		
00151	AEMMAP77	AEMMAP94		
00152	AEMMAP71	AEMPAR80		
00153	AEMMAP92			
00154	AEMMAP92			
00155	AEMGRP00	AEMMAP00	AEMTBL00	
00156	AEMMAP02			
00157	AEMMAP62			
00158	AEMMAP62			
00159	AEMMAP77			
00160	AEMMAP04	AEMMAP77		
00161	AEMMAP76			
00162	AEMPSG84			
00163	AEMMAP76			
00164	AEMMAP78			
00165	AEMMAP75			
00167	AEMPSG00	AEMPSG05	AEMPSG50	
00168	AEMPSG88			
00170	AEMMAP98			
00171	AEMMAP83			
00172	AEMMAP07	AEMMAP10	AEMPAR80	
00173	AEMMAP83			
00174	AEMMAP83			
00175	AEMMAP55	AEMMAP74		
00176	AEMMAP74			
00177	AEMINT80			
00178	AEMGRP02	AEMGRP83	AEMMAP02	
00179	AEMSRV64			
00180	AEMSRV64			
00183	AEMMAP08			
00184	AEMMAP72			
00186	AEMMAP72			
00187	AEMMAP04			
00188	AEMMAP04			
00189	AEMMAP78			
00190	AEMGRP00	AEMMAP00		
00191	AEMMAP83			
00192	AEMGRP03	AEMMAP83		
00193	AEMMAP67			
00194	AEMGRP03	AEMMAP83		
00195	AEMMAP78			
00196	AEMSRV64			
00199	AEMGRP82			
00200	AEMGRP82			
00201	AEMDIA92			
00202	AEMLIB82	AEMPSG00	AEMUTY02	
00203	AEMLIB82			
00204	AEMINT80			
00205	AEMINT80			
00206	AEMINT01			
00207	AEMMAP90			
00208	AEMLIB03			
00210	AEMLIB04			
00211	AEMLIB03			
00212	AEMLIB03			
00213	AEMLIB04	AEMLIB82		
00214	AEMDIA80	AEMGRP05	AEMMAP90	AEMPAR90
	AEMTBL01			
00215	AEMDIA80			
00216	AEMDIA80			

GDDM-IMD messages

Msg	Modules that issue the message			
00217	AEMDIA80			
00218	AEMDIA80			
00219	AEMDIA80			
00220	AEMPSG01			
00224	AEMPSG00			
00243	AEMMAP56			
00244	AEMMAP57			
00253	AEMPSG00	AEMUTY01		
00255	AEMGRP05			
00256	AEMMAP90			
00258	AEMGRP05			
00259	AEMPAR90			
00260	AEMPAR90			
00263	AEMGRP02	AEMMAP02		
00271	AEMMAP72			
00272	AEMMAP98			
00275	AEMPSG85			
00276	AEMMAP72			
00277	AEMUTY03			
00279	AEMGRP02	AEMGRP03		
00280	AEMGRP02	AEMGRP03		
00281	AEMGRP02	AEMGRP03		
00282	AEMMAP90			
00285	AEMMAP83			
00286	AEMMAP83			
00287	AEMGRP82			
00288	AEMGRP82			
00289	AEMGRP82			
00290	AEMGRP82			
00291	AEMGRP82			
00292	AEMGRP82			
00295	AEMPSG03			
00296	AEMTBL01			
00297	AEMTBL01			
00299	AEMMAP67			
00300	AEMGRP02			
00302	AEMGRP09			
00303	AEMMAP02			
00304	AEMGRP03			
00305	AEMMAP02			
00306	AEMMAP02			
00307	AEMMAP83			
00308	AEMMAP83			
00309	AEMGRP03			
00315	AEMPSG81			
00316	AEMMAP62			
00319	AEMSRV90			
00320	AEMSRV90			
00321	AEMSRV90			
00322	AEMSRV90			
00323	AEMSRV90			
00325	AEMMAP47			
00326	AEMMAP44			
00327	AEMMAP44	AEMMAP90		
00328	AEMMAP44	AEMMAP90		
00330	AEMPSG99			
00332	AEMSRV64			
00333	AEMINT01			
00335	AEMPSG73			
00336	AEMPSG73			
00337	AEMINT01			
00338	AEMMAP10	AEMMAP67		
00339	AEMMAP10	AEMMAP67		
00342	AEMGRP05			
00343	AEMPSG04			
00344	AEMPSG04			
00345	AEMGRP05			
00346	AEMGRP05			
00347	AEMPSG81			
00348	AEMGRP00	AEMMAP00	AEMMAP03	AEMMAP80
	AEMPSG00			
00351	AEMMAP56			
00352	AEMMAP56			
00353	AEMMAP56	AEMMAP67	AEMSRV64	
00354	AEMMAP57			
00355	AEMMAP55	AEMMAP57		
00356	AEMINT80			
00358	AEMMAP67			
00359	AEMMAP54			
00360	AEMMAP54			
00361	AEMMAP54			
00362	AEMMAP54			
00363	AEMMAP54			
00364	AEMMAP54			
00365	AEMMAP54			
00366	AEMMAP54			
00367	AEMMAP54			
00368	AEMMAP07			
00369	AEMSRV75			
00370	AEMPSG04			
00377	AEMPSG01			
00378	AEMPSG82			
00379	AEMPSG81			
00381	AEMMAP62			
00382	AEMMAP02	AEMMAP03	AEMMAP83	
00383	AEMGRP02	AEMGRP03		
00384	AEMMAP62	AEMMAP78		
00385	AEMMAP05	AEMMAP62		
00386	AEMINT01			
00387	AEMMAP55			
00388	AEMMAP55			
00389	AEMMAP55			
00390	AEMMAP55			
00391	AEMMAP55			
00393	AEMPSG04			
00395	AEMPSG84			
00400	AEMMSL00			
00401	AEMMSL00	AEMMSL88		
00402	AEMMSL00			
00403	AEMMSL00	AEMMSL03	AEMMSL04	AEMMSL05
	AEMMSL06	AEMMSL86		
00404	AEMMSL88			
00410	AEMPSG05			
00411	AEMPSG05			
00412	AEMPSG50			
00414	AEMMAP29			
00415	AEMMAP29			
00416	AEMMAP08	AEMMAP76		

GDDM-CSPF messages

Message	Background modules that issue the message			
EAKB001	EAKTINIT	EAKVINIT		
EAKB003	EAKTINIT	EAKTPCAD	EAKTPCDS	EAKTPCJE
EAKB004	EAKTINIT			
EAKB005	EAKTINIT	EAKVINIT		
EAKB007	EAKTINIT	EAKVINIT		
EAKB008	EAKVINIT			
EAKB009	EAKTINIT	EAKVINIT		
EAKB010	EAKVINIT			
EAKB013	EAKTINIT			
EAKB014	EAKTLVSS			
EAKB015	EAKTLVSS	EAKVLVSS		
EAKB017	EAKTLVSS	EAKVLVSS		
EAKB018	EAKTPCDS			
EAKB023	EAKTINIT			
EAKB024	EAKTINIT			
EAKB025	EAKAINTP	EAKTINTP		
EAKB027	EAKAINTP	EAKTINTP		
EAKB029	EAKTINIT			
EAKB030	EAKTLVSS	EAKVLVSS		
EAKB031	EAKAELLI			
EAKB032	EAKAHATC			
EAKB033	EAKAINTP			
EAKB034	EAKTPCJE			
EAKB100	EAKTINIT	EAKVINIT		

Message	Foreground modules that issue the message			
EAKF000	EAKERR			
EAKF001	EAKP05			
EAKF002	EAKP05			
EAKF003	EAKP05			
EAKF004	EAKP05			
EAKF005	EAKP05			
EAKF006	EAKCOP			
EAKF008	EAKCOP	EAKHELP	EAKP00	EAKP01
	EAKP02	EAKP03	EAKP04	EAKP05
	EAKP06			
EAKF009	EAKP00	EAKP05		
EAKF010	EAKP00			
EAKF011	EAKP01			
EAKF012	EAKP01	EAKP02	EAKP03	EAKP04
EAKF013	EAKP01			
EAKF014	EAKP01	EAKP02	EAKP03	
EAKF015	EAKP01			
EAKF016	EAKP01	EAKP02		
EAKF017	EAKP04			
EAKF018	EAKP04			
EAKF019	EAKP05			
EAKF020	EAKP05			
EAKF021	EAKP03	EAKP05	EAKP06	
EAKF022	EAKP03	EAKP05	EAKP06	
EAKF023	EAKP03	EAKP05	EAKP06	
EAKF024	EAKP03	EAKP05	EAKP06	
EAKF025	EAKP03	EAKP05	EAKP06	
EAKF026	EAKP03	EAKP05	EAKP06	
EAKF027	EAKP03	EAKP05	EAKP06	
EAKF028	EAKP05			
EAKF029	EAKP03			
EAKF030	EAKP03			
EAKF032	EAKP03			
EAKF033	EAKP03			
EAKF034	EAKP03			
EAKF035	EAKP03			
EAKF036	EAKP01			
EAKF037	EAKP06			
EAKF038	EAKP06			
EAKF039	EAKP06			
EAKF040	EAKP06			
EAKF042	EAKP06			
EAKF043	EAKP06			
EAKF044	EAKP00			
EAKF045	EAKP06			
EAKF046	EAKP06			
EAKF047	EAKP05			
EAKF048	EAKP00			
EAKF049	EAKP06			
EAKF050	EAKP06			
EAKF051	EAKP03			

Message	View utility modules that issue the message
EAKV001	EAKVIEW
EAKV002	EAKVIEW
EAKV003	EAKVIEW
EAKV004	EAKVIEW
EAKV005	EAKVIEW

Appendix D. Trace-string grammar

Product-sensitive programming interface

The grammar of a set of TRCESTR statements is shown, in Backus Naur form, in the table below:

program	→	program_body
	or	CLEAR program_body
	or	program_body FORCE
	or	CLEAR program_body FORCE
program_body	→	statement
	or	program_body statement
statement	→	compound_statement
	or	simple_statement
	or	if_statement
if_statement	→	IF relational_expression THEN statement
	or	IF relational_expression THEN statement ELSE statement
compound_statement	→	DO statement_list END
statement_list	→	statement
	or	statement_list ; statement
simple_statement	→	function
	or	simple_statement function
relational_expression	→	expression
	or	expression RELOP expression
expression	→	term
	or	expression ADDOP term
term	→	factor
	or	term MULOP factor
factor	→	id
	or	(relational_expression)
	or	¬ factor
	or	SIGN factor
id	→	constant
	or	address
	or	function
	or	CHARACTER_STRING
constant	→	BINARY_NUMBER
	or	DECIMAL_NUMBER
	or	HEXADECIMAL_NUMBER

address	→	expression %
	or	expression GR
	or	expression FR
function	→	VARIABLE_NAME
	or	VARIABLE_NAME (parameter-list)
parameter-list	→	cprameter-item
	or	cprameter-list , parameter-item
parameter-item	→	relational_expression
SIGN	→	+ or -
ADDOP	→	+ or - or OR
MULOP	→	* or / or AND
RELOP	→	EQ or NE or LT or GT or LE or GE

Recognized tokens

The text patterns that match the lexical tokens used by the grammar are defined below using the operators defined in the following table:

& A & B indicates the character A followed immediately by the character B. For simplicity, this operator is assumed:

1. Between characters that are not operators.
2. After the operator) and before the operator (except that only one & is assumed between each) and (.

| A|B indicates either the character A or the character B.

( ) Brackets change the relative order of priority of operators.

{ } Operators enclosed in these brackets are treated as ordinary characters.

+ One or more occurrences of the immediately preceding character or group of characters if they are enclosed in brackets.

- Indicates a range, which may only be defined within:

- Uppercase letters
- Lowercase letters
- Digits 0 through 9.

Note: Blanks are allowed between the template characters.

Here are some examples of the text patterns that may be described using these operators:

a+	matches a aa aaaaa
abc+	matches abc abcccccc abcc
(abc)+	matches abc abcabc abcabcabc
(a|b)	matches a b
(a|b)+	matches a b aba
(a|b)+c	matches ac aaaac abac.

The patterns that correspond to the allowed lexical tokens and their alternative forms are given in the following table (uppercase or lowercase characters may be used):

Token	**Pattern**				
+	{+}				
-	{-}				
OR	{|}	|	(OR)		
*	{*}				
/	/				
AND	{&}	|	(AND)		
EQ	=	|	(EQ)		
NE	(¬=)	|	(NE)		
LT	<	|	(LT)		
GT	>	|	(GT)		
LE	(<=)	|	(=<)	|	(LE)
GE	(>=)	|	(=>)	|	(GE)
(	{(}				
)	{)}				
¬	¬				
DO	(DO)				
END	(END)				
IF	(IF)				
THEN	(THEN)				
ELSE	(ELSE)				
;	;				
,	,				
GR	(GR)				
FR	(FR)				
%	%				
CLEAR	(CLEAR)				
FORCE	(FORCE)				
BINARY_NUMBER	B'(0|1)+'				
DECIMAL_NUMBER	(0-9)+				
HEXADECIMAL_NUMBER	X'(0-9|A-F)+'				
CHARACTER_STRING	'(A-Z|0-9|{*}|{+}|_|%|.|$|¢|{|}|:|;|?|,)+'				
VARIABLE_NAME	(A-Z|0-9)+				

The pattern matching to find tokens proceeds according to the following rules:

1. Any character that does not occur in the above table is invalid.
2. A space is automatically added to the end of each line.
3. Matching proceeds from a given start position until:
    a. An invalid character is reached
    b. A space is reached
    c. No further match is possible.
4. The longest match is chosen.
5. If there are two candidates of equal length, the token that appears higher in the above list is chosen:
6. After a successful match, the next start position is the next nonblank character.
7. If there is no match, the next start position is the first nonblank character *after the next blank*.

Rule 2 means tokens must end at the end of a line.

Rule 3 means that the variable GOTIF will not be interpreted as the variable GOT followed by the token IF.

Rule 4 means that <= will not be mistaken as < and = and that FR will not be interpreted as the hexadecimal number F followed by the variable name R.

Rule 5 ensures that THEN will not be interpreted as a variable name.

Rule 6 means that tokens that can be distinguished need not be separated by a blank. Thus (1+2) will be interpreted as 5 tokens.

Rule 7 means that tokens will not be picked out of an invalid string; thus IF will not be found in the string xxxxxIFxxx.

Blanks are required to separate tokens that could form a valid single token, otherwise they are ignored.

Note: A consequence of rule 3.b is that blanks may *not* appear within a character string. An underline character within a character string will be replaced with a blank after lexical analysis has taken place.

Grammatical constructs that might appear peculiar, for example:

```
1 fr gr
```

are considered to be syntactically correct but semantically in error.

End of Product-sensitive programming interface

Glossary

This glossary defines various terms used in the documentation of GDDM.

This glossary includes terms and definitions from the *IBM Dictionary of Computing*, SC20-1699.

A

APAR. Authorized program analysis report. A report of a problem caused by a suspected defect in a current unaltered release of a program.

API. Application program interface.

application program interface (API). The formally defined programming-language interface between an IBM system control program or licensed program and its user.

ASCII. American National Standard Code for Information Interchange.

C

CDPDS. Composite-document presentation data stream.

CDPU. Composite-document print utility.

CGM. Computer graphics metafile.

CICS. Customer Information Control System. A subsystem of MVS or VSE under which GDDM can be used.

CMS. Conversational Monitor System. A time-sharing subsystem that runs under VM/SP.

D

ddname. Data definition name.

default value. A value chosen by GDDM when no value is explicitly specified by the user. For example, the default line type is a solid line.

denibblised data. The decoded data stream used between GDDM in the host and GDDM-PCLK in the PC.

device family. In GDDM, a device classification that governs the general way I/O will be processed. For example:

- Family 1: 3270 display or printer
- Family 2: queued printer
- Family 3: system printer (alphanumerics only)
- Family 4: high-resolution (page) printer.

display device. Any output unit that gives a visual representation of data. For example, a screen or printer. More commonly, the term is used to mean a screen as opposed to a printer.

display terminal. An input/output unit by which a user communicates with a data-processing system or subsystem. Usually includes a keyboard and always provides a visual presentation of data. For example, an IBM 3179 display.

E

extended data stream. For 3179, 3192G, 3268, 3278, 3279, 3287, and 3472G devices, input/output data formatted and encoded in support of color, programmed symbols, and extended highlighting. These features extend the 3270 data-stream architecture.

external defaults. GDDM-supplied values that users can change to suit their own needs.

G

GDDM. Graphical Data Display Manager.

GDDM-CSPF. GDDM Central Slide and Plot Facility

GDDM Central Slide and Plot Facility. A member of the GDDM family of licensed programs. It allows plotting to take place on a remote auto-feed plotter. Also it allows enhanced color slides to be produced.

GDDM-GKS. A member of the GDDM family of licensed programs. It allows the use of GKS functions with GDDM.

GDDM Image View Utility. A member of the GDDM family of licensed programs. It allows users to view, create, modify, store, and print images.

GDDM-IMD. GDDM Interactive Map Definition.

GDDM Interactive Map Definition. A member of the GDDM family of licensed programs. It allows users to create alphanumeric layouts at the terminal. The operator defines the position of each field within the layout and may assign attributes, default data, and associated variable names to each field. The resultant map may be tested from within the utility.

GDDM-IVU. GDDM Image View Utility.

GDDM-PCLK. A personal computer licensed program that allows GDDM graphics to be displayed to a personal computer.

GDDM-REXX. A member of the GDDM family of licensed programs. It allows the use of GDDM from programs written for the VM/System Product Interpreter.

GDDM storage. The portion of host computer main storage used by GDDM.

GKS. Graphical Kernel System.

graphics. A picture defined by graphics primitives and graphics attributes.

H

hardware characters. Synonym for *hardware symbols.*

hardware symbols. The characters that are supplied with the device. The term is loosely used also for GDDM mode-1 symbols that are loaded into a PS store for later display. Synonymous with *hardware characters.*

help panel. A panel presenting tutorial text to assist the terminal user. All the GDDM interactive utilities possess comprehensive help panels.

I

ICU. Interactive Chart Utility.

| **IFCB.** Interface control block.

Image Symbol Editor (ISE). A GDDM-supplied interactive editor that lets users create or modify their own image symbol sets (ISS).

image symbol set (ISS). A set of symbols each of which was created as a pattern of dots. Contrast with *vector symbol set (VSS).*

IMS/VS. Information Management System/Virtual Storage. A subsystem of MVS under which GDDM can be used.

integer. A whole number (for example, −2, 3, 457).

Interactive Chart Utility (ICU). A GDDM-PGF menu-driven program that allows business charts to be created interactively by non-programmers.

| **IPDS.** Intelligent printer data stream (for 3812, 3816, 4224, and 4234 printers).

ISE. Image Symbol Editor.

ISS. Image symbol set.

J

JCL. Job Control Language.

K

kanji. A character set of symbols used in Japanese ideographic alphabets.

L

link edit. To create a loadable computer program with a linkage editor.

load module. A program unit that is suitable for loading into main storage for execution; it is usually the output of a linkage editor.

M

map. A set of values having defined correspondence with the quantities or values of another set.

map specification library (MSL). The data set where maps are held in their source form.

MSL. Map specification library.

N

National Language (NL) feature. The translations of the ICU panels and some of the GDDM messages into a variety of languages other than U.S. English.

nibblised data. The encoded data stream used between GDDM in the host and GDDM-PCLK in the PC.

nickname. In GDDM, a quick and easy means of referring to a device, the characteristics and identity of which have been predefined.

O

object code. Output from a compiler or assembler that is in itself executable machine code or is suitable for processing to produce executable machine code.

object deck. Synonym for *object module*.

object libraries. An area on a direct access storage device used to store object programs and routines.

object module. A module that is the output of an assembler or a compiler and is input to a linkage editor.

operator reply mode. In GDDM, the mode of interaction available to the operator (display terminal user) with respect to the modification (or not) of alphanumeric character attributes for an input field.

outbound structured field. An element in 3270 data streams from host to terminal with formatting that permits variable-length and multiple-field data to be sequentially translated by the receiver into its component fields without having to examine every byte.

P

page printer. A printer, such as the 4250 or 3800-3, that has a high density of pixels to the inch and therefore produces output of good quality (also known as a composed-page printer).

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Synonymous with program library.

PCB. Program communication block (IMS/VS).

PCT. Program control table (CICS).

PDS. In MVS/TSO, a partitioned data set.

PL/I. One of the programming languages supported by GDDM.

plotter. An output device that uses pens to draw its output on paper or transparency foils.

PPT. Processing program control table (CICS).

PGF. A member of the GDDM family of licensed programs. It is concerned with business graphics, as opposed to general graphics.

print utility. A subsystem-dependent utility that sends print files from various origins to a queued printer.

PS. Programmed symbols.

PS overflow. A condition where the graphics cannot be displayed in its entirety because the picture is too complex to be contained in the device's PS stores.

PSB. In IMS/VS, a program specification block.

Q

QSAM. Queued sequential access method.

QTAM. Queued telecommunications access method.

queued printer. A printer belonging to the subsystem under which GDDM runs, to which output is sent indirectly by the GDDM Print Utility program. In some subsystems, this may allow the printer to be shared between multiple users. Contrast with **system printer.**

R

RCP. Request control parameter.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

reply mode. See **operator reply mode.**

S

scanner. A device that produces a digital image from a document.

SPI. System programmer interface.

SPIB. System programmer interface block.

symbol set. A collection of symbols, usually but not necessarily forming a font. GDDM applications may use the hardware device's own symbol set. Alternatively, they can use image or vector symbol sets, which the user may have created.

symbol set identifier. In GDDM, an integer (or the equivalent EBCDIC character) by which the programmer refers to a loaded symbol set.

system printer. A printer belonging to the subsystem under which GDDM runs, to which output is sent indirectly by system spooling facilities. Contrast with **queued printer.**

T

TCA. Terminal services interface control area.

TCT. Terminal control table (CICS).

terminal. A device, usually equipped with a keyboard and a display unit, capable of sending and receiving information over a link. See also **display terminal.**

TNL. Technical newsletter.

toggle. The switch between two modes. In GDDM-PLCK, the user can toggle between the host session and the DOS session by pressing one or more keys.

TSO. Time sharing option. A subsystem of OS/VS under which GDDM can be used.

U

UDS. User default specification.

UDSL. A list of user default specifications (UDSs).

unformatted data. In GDDM image processing, compressed or uncompressed binary image data that has no headers, trailers, or embedded control fields other than any defined by the compression algorithm, if applicable. The data is in row major order, beginning with the top left of the picture.

user default specification (UDS). The means of changing a GDDM default value. The default values that a UDS can change are those of the GDDM or subsystem environment, GDDM user exits, and device definitions.

user exit. A point in GDDM execution where a user routine will gain control if such has been requested.

V

VCNA. VTAM communications network application.

VM/SP CMS. IBM Virtual Machine/System Product Conversational Monitor System. A system under which GDDM can be used.

VM/XA. IBM Virtual Machine/Extended Architecture. A system under which GDDM can be used.

VSE. Virtual storage extended. An operating system consisting of VSE/Advanced Functions and other IBM programs. In GDDM, the abbreviation VSE has sometimes been used to refer to the Vector Symbol Editor, but to avoid confusion, this usage is deprecated.

VSS. Vector symbol set.

VTAM. Virtual Telecommunications Access Method.

W

| **WACK.**
| Wait-before-transmitting-positive-acknowlegment
| character.

window. (1) In GDDM, a defined section of world coordinates. The window can be regarded as a set of coordinates that are overlaid on the viewport. (2) In GDDM, the "graphics window" is the set of coordinates used for defining the primitives that make up a graphics display. By default, both x and y coordinates run from 0 through 100. (3) In GDDM, an "operator window" is an independent rectangular subdivision of the screen. Several can exist at the same time, and each can receive output from, and send input to, either a separate GDDM program or a separate function of a single GDDM program. (4) In GDDM, the "page window" defines which part of a deep or wide page should currently be displayed.

workstation. A display screen together with attachments such as a local copy device or a tablet.

Index

Reader's Comments

GDDM
Diagnosis and Problem Determination Guide
Version 2 Release 3

Publication No. SC33-0326-3

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

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.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Phone No.

Reader's Comments
SC33-0326-3

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

International Business Machines Corporation
Department 6R1H
180 KOST ROAD
MECHANICSBURG PA 17055-0786

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

Cut or Fold
Along Line

SC33-0326-3
Version 2 Release 3

GDDM Diagnosis and Problem Determination Guide Printed in U.S.A. SC33-0326-3

SC33-0326-03

IBM