GDDM

GDDM/MVS Program Directory

Version 3 Release 2

GC33-1801-01

GDDM

GDDM/MVS Program Directory

Version 3 Release 2

GC33-1801-01

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xi.

Second Edition (December 1997)

This edition applies to GDDM/MVS Version 3 Release 2, program number 5695-167.

It also applies to GDDM/MVS as an element of OS/390 Version 1 (program number 5645-001) and OS/390 Version 2 (program number 5647-A01).

Publications are not stocked at the addresses given below. Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM United Kingdom Laboratories,
Information Development, Mail Point 095,
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.

Contents

Figures

Tables

Notices

The following paragraph does not apply to 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.

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 product, program, or service in this publication is not intended to state or imply that only IBM’s product, program, or service may be used. Subject to IBM’s valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact Laboratory Counsel, Mail Point 151, IBM United Kingdom Laboratories, Hursley Park, Winchester, Hampshire SO21 2JN, England. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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 Licensing, 500 Columbus Avenue, Thornwood, New York, 10594, U.S.A.

General-use programming interfaces allow the customer to write programs that obtain the services of GDDM.

Trademarks and service marks

The following terms, used in this book, are trademarks or registered trademarks of IBM Corporation in the United States or other countries or both:

CBIPO	IBM	OS/390
CBPDO	IMS	System/370
CICS	GDDM	System/390
CustomPac	OS/2	

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

Preface

Important!

This program directory is only for planning purposes. Use the program directory shipped with the product tape when you are ready to install GDDM/MVS Version 3 Release 2.

What this book is about

This book is intended to help you plan the installation of the GDDM/MVS product and its associated products (GDDM-PGF, GDDM-IMD, GDDM-GKS, and GDDM-IVU).

The book contains guidance about planning for GDDM, and shows you the requirements to successfully complete the installation and the post-installation tasks required.

Who this book is for

This book is for the system programmer responsible for program installation and maintenance.

Latest GDDM information

For up-to-date information on GDDM products, check our Home Page on the Internet at the following URL:

`http://www.hursley.ibm.com/gddm/`

You might also like to look at the IBM Software Home Page at:

`http://www.software.ibm.com`

GDDM publications

GDDM Base	*GDDM Base Application Programming Guide*, SC33-0867 *GDDM Base Application Programming Reference*, SC33-0868 *GDDM Diagnosis*, SC33-0870 *GDDM General Information*, GC33-0866 *GDDM/MVS Program Directory*, GC33-1801 *GDDM/VM Program Directory*, GC33-1802 *GDDM/VSE Program Directory*, GC33-1803 *GDDM Messages*, SC33-0869 *GDDM Series Licensed Program Specifications*, GC33-0876 *GDDM System Customization and Administration*, SC33-0871 *GDDM User's Guide*, SC33-0875 *GDDM Using the Image Symbol Editor*, SC33-0920
GDDM-GKS	*GDDM-GKS Programming Guide and Reference*, SC33-0334
GDDM-IMD	*GDDM Interactive Map Definition*, SC33-0338
GDDM-IVU	*GDDM Image View Utility*, SC33-0479
GDDM-PGF	*GDDM-PGF Application Programming Guide*, SC33-0913 *GDDM-PGF Programming Reference*, SC33-0333 *GDDM-PGF Interactive Chart Utility*, SC33-0328 *GDDM-PGF Vector Symbol Editor*, SC33-0330 *GDDM-PGF OPS User's Guide*, SC33-1776

GDDM/MVS is an element of OS/390. GDDM-REXX/MVS and GDDM-PGF are optional features of OS/390. For a complete list of the publications associated with OS/390, see the *OS/390 Information Roadmap*, GC28-1727.

Chapter 1. Introduction

This program directory is intended for the system programmer responsible for program installation and maintenance. It contains information concerning the material and procedures associated with the installation of GDDM/MVS. You should read all of this program directory before installing the program and then keep it for future reference.

The program directory contains the following sections:

- Chapter 2, “Program materials” on page 3 identifies the basic and optional program materials and documentation for GDDM/MVS.
- Chapter 3, “Program support” on page 7 describes the IBM support available for GDDM/MVS.
- Chapter 4, “Program and service level information” on page 9 lists the APARs (program level) and PTFs (service level) incorporated into GDDM/MVS.
- Chapter 5, “Installation requirements and considerations” on page 11 identifies the resources and considerations for installing and using GDDM/MVS.
- Chapter 6, “Installation instructions” on page 13 provides detailed installation instructions for GDDM/MVS.
- Chapter 7, “Post-installation tasks” on page 33 describes what to do after installation.
- Chapter 8, “Testing GDDM installation” on page 63 describes how to test the GDDM installation.
- Chapter 9, “Servicing GDDM” on page 77 describes how to apply service updates to GDDM/MVS.
- Appendix A, “GDDM/MVS Install logic” on page 83 provides the install logic for GDDM/MVS.
- Appendix B, “JCLIN for GDDM/MVS” on page 85 gives instructions about where to find the JCLIN for GDDM/MVS.
- Appendix C, “GDDM/MVS National language support” on page 87 describes the national language support for GDDM/MVS.

Before installing GDDM/MVS, read “Preventive service planning” on page 7. This section tells you how to find any updates to the information and procedures in this program directory.

Do not use this program directory if you are installing GDDM/MVS with an MVS Custom-Built Installation Process Offering (CBIPO) (5751-CS1). Instead, use the CBIPO Related Installation Materials (RIMs) provided with the CBIPO. The CBIPO RIMs will point you to specific sections of the program directory as required.

If you are installing GDDM/MVS using the MVS Custom-Built Product Delivery Offering (CBPDO) (5751-CS3), use the softcopy program directory provided on the CBPDO tape. Your CBPDO contains a softcopy preventive service planning (PSP) upgrade for this product. All service and HOLDDATA for GDDM/MVS are included on the CBPDO tape.

Chapter 2. Program materials

An IBM program is identified by a program number and a feature code. The program number for GDDM/MVS is 5695-167. GDDM/MVS is an element of OS/390 Version 1, which is program number 5645-001, and OS/390 Version 2, which is program number 5647-A01.

The program announcement material describes the features supported by GDDM/MVS. Ask your IBM marketing representative for this information if you have not already received a copy.

The following sections identify the basic and optional program materials available with this program.

Basic machine-readable material

The distribution medium for this program is 9-track magnetic tape (written at 6250 BPI), 3480 cartridge, or 4mm DAT cartridge.

For GDDM/MVS Version 3, each of the distribution media has a group of feature numbers, each feature number corresponding to a different national language. Any feature number from one of the groups will cause an identical tape or cartridge to be shipped on the corresponding medium. Each language has its own feature number to allow any publications translated for that language to be shipped with the tape or cartridge.

The tape or cartridge contains all the programs and data needed for installation. It is installed using SMP/E. See Chapter 6, “Installation instructions” on page 13 for more information about how to install the program. Table 1 describes the tape and cartridge. Table 2 on page 4 describes the file content of the program tape and cartridge.

Table 1 (Page 1 of 2). Basic Material: Program Tape

Medium	Feature Number	Physical Volume	External Label Identification	VOLSER
6250 tape	5861	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5160	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6156	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5105	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5107	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5109	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5163	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5117	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5101	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6101	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6159	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6153	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5115	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5113	1 of 1	GDDM/MVS V3R2 BASE	GD3200

Table 1 (Page 2 of 2). Basic Material: Program Tape

Medium	Feature Number	Physical Volume	External Label Identification	VOLSER
3480	5862	1 of 1	GDDM/MVS V3R2 BASE	GD3200
cartridge	5161	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6157	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5106	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5108	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5110	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5164	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5118	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5102	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6102	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6160	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	6154	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5116	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5114	1 of 1	GDDM/MVS V3R2 BASE	GD3200
4mm DAT	5337	1 of 1	GDDM/MVS V3R2 BASE	GD3200
cartridge	5660	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5661	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5662	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5663	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5664	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5665	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5666	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5667	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5668	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5669	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5670	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5671	1 of 1	GDDM/MVS V3R2 BASE	GD3200
	5672	1 of 1	GDDM/MVS V3R2 BASE	GD3200

Table 2 (Page 1 of 2). GDDM/MVS installation tape: File Content

VOLSER	File	Data Set Name
GD3200	1	SMPMCS
	2	IBM.HGD3200.F1
	3	IBM.HGD3200.F2
	4	IBM.HGD3200.F3
	5	IBM.HGD3200.F4
	6	IBM.HGD3200.F5
	7	IBM.HGD3200.F6
	8	IBM.HGD3200.F7
	9	IBM.HGD3200.F8
	10	IBM.HGD3200.F9
	11	IBM.HGD3200.F10
	12	IBM.JGD3219.F1
	13	IBM.JGD3219.F2
	14	IBM.JGD3219.F3
	15	IBM.JGD3219.F4

Table 2 (Page 2 of 2). GDDM/MVS installation tape: File Content

VOLSER	File	Data Set Name
	16	IBM.JGD3220.F1
	17	IBM.JGD3220.F2
	18	IBM.JGD3220.F3
	19	IBM.JGD3220.F4
...	..	and so on with 4 files per national language, except for Simplified Chinese, which has an extra file (IBM.JGD3221.F5) and Japanese (Kanji), which has an extra file (IBM.JGD3227.F5).

Program publications

The following sections identify the basic and optional publications for GDDM/MVS.

Basic program publications

Table 3 identifies the basic program publications for GDDM/MVS. One copy of each of these publications is included when you order the basic materials for GDDM/MVS. For additional copies, contact your IBM representative.

Table 3. Basic Material: Unlicensed Publications

Publication Title	Form Number
GDDM Base Application Programming Guide	SC33-0867
GDDM Base Application Programming Reference	SC33-0868
GDDM Diagnosis	SC33-0870
GDDM General Information	GC33-0866
GDDM Messages	SC33-0869
GDDM Series Licensed Program Specifications	GC33-0876
GDDM System Customization and Administration	SC33-0871
GDDM User's Guide	SC33-0875
GDDM Using the Image Symbol Editor	SC33-0920

Optional program publications

Table 4 identifies the optional licensed program publications for GDDM/MVS. The first copy is available at no charge to licensees of the basic material by ordering the 7xxx Feature Number. Order additional copies using the 8xxx Feature Number. A fee is charged for additional copies.

Table 4. Optional Material: Licensed Publications

Publication Title	Form Number	Feature Number First Copy	Feature Number Additional Copy
Printing and Publishing Collection (CD-ROM)	SK2T-2921	7902	8039

Notes:

1. Unlicensed publications for GDDM are offered in softcopy form on the Printing and Publishing Collection CD-ROM. Customers outside the United States of America should order the CD-ROM by using order number 5636-PUB to obtain their first copy. The CD-ROM can also be ordered using the order number SK2T-2921.
2. Additional copies of the *GDDM/MVS Program Directory* can be obtained by using order number GC33-1801.

Publications useful during installation

The publications listed in Table 5 may be useful during the installation of GDDM/MVS. To order copies, contact your IBM representative.

Table 5. Publications Useful During Installation

Publication Title	Form Number
SMP/E User's Guide	SC28-1302
SMP/E Reference	SC28-1107
SMP/E Messages and Codes	SC28-1108

Chapter 3. Program support

This section describes the IBM support available for GDDM/MVS.

Service instructions

Contact your IBM marketing representative or systems engineer (SE) for specific information about available service instructions.

Preventive service planning

If you obtained GDDM/MVS as part of a CBPDO, there is HOLDDATA and Preventive Service Planning (PSP) information for GDDM/MVS on the CBPDO tape. Before installing GDDM/MVS, check with your IBM Support Center or use either Information/Access or SoftwareXcel Extended to see whether there is additional Preventive Service Planning (PSP) information that you should know. To obtain this information, specify the following UPGRADE and SUBSET values:

Table 6. PSP Upgrade and Subset ID

UPGRADE	SUBSET	RETAIN Release
GDDM320	HGD3200	GDDM/MVS
GDDM320	JGD3219	GDDM/MVS NLS

Statement of support procedures

Report any difficulties you have using this program to your IBM Support Center. If an APAR is required, the Support Center will provide the address to which any needed documentation can be sent.

Table 7 identifies the component IDs (COMPID) for GDDM/MVS.

Table 7. Component IDs

FMID	COMPID	Component Name	REL
HGD3200	569516701	GDDM/MVS	200
JGD3219	569516702	GDDM/MVS NLS - US English	219
JGD3220	569516702	GDDM/MVS NLS - Brazilian	220
JGD3221	569516702	GDDM/MVS NLS - Simplified Chinese	221
JGD3222	569516702	GDDM/MVS NLS - Danish	222
JGD3223	569516702	GDDM/MVS NLS - French	223
JGD3224	569516702	GDDM/MVS NLS - German	224
JGD3225	569516702	GDDM/MVS NLS - Korean (Hangeul)	225
JGD3226	569516702	GDDM/MVS NLS - Italian	226
JGD3227	569516702	GDDM/MVS NLS - Japanese (Kanji)	227
JGD3228	569516702	GDDM/MVS NLS - Norwegian	228
JGD3229	569516702	GDDM/MVS NLS - French (Canadian)	229
JGD3230	569516702	GDDM/MVS NLS - Spanish	230
JGD3231	569516702	GDDM/MVS NLS - Taiwan Chinese (Traditional)	231
JGD3232	569516702	GDDM/MVS NLS - Swedish	232

Chapter 4. Program and service level information

This section identifies the program and any relevant service levels of GDDM/MVS. The program level refers to the APAR fixes incorporated into the program. The service level refers to the PTFs integrated. Information about the cumulative service tape is also provided.

Program level information

The following APAR fixes against previous releases of GDDM/MVS have been incorporated into this release:

PN59427 PN62024
PN62733 PN63738
PN64024 PN65404
PN65498 PN65512
PN65514 PN65681
PN65937 PN65966
PN66171 PN66450
PN66458 PN66461
PN66468 PN66474
PN66507 PN66951
PN67035 PN67882
PN68256 PN68922
PN68927 PN69043
PN69302 PN69421
PN69423 PN69592
PN69719 PN69769
PN70250 PN70433
PN70441 PN70621
PN72131 PN72461
PN72998 PN73336
PN73449 PN73542
PN73588 PN73756
PN73783 PN73970
PN74028 PN74032
PN74087 PN74186
PN74248 PN74255
PN74318 PN74817
PN75807 PN76011
PN76158 PN76418
PN76534 PN76546
PN77428 PN77824
PN78575 PN79934
PN80122 PN80374
PN80720 PN82044
PN82794

Service level information

This is the initial release of GDDM/MVS 3.2.0 and no additional PTFs are included on the tape.

Cumulative service tape

A cumulative service tape, containing PTFs not incorporated into this release, might be included with this program. If you received this product as part of a CBPDO, then there is no cumulative service tape.

Chapter 5. Installation requirements and considerations

The following sections identify the system requirements for installing and activating GDDM/MVS. In most cases, you can install GDDM/MVS on a running system (target system). However, sometimes two systems may be required. If two systems are required, then the following terminology is used:

1. The system used to install the program (driving system)
2. The system on which the program is installed (target system).

Driving system requirements

This section describes the environment of the driving system required to install GDDM/MVS.

Programming requirements

SMP/E 1.7.0 plus the RFDSNPFX APAR IR23322 (PTF UR40251), or a later level of SMP/E, is required for the installation of GDDM/MVS 3.2.

DASD storage requirements

The installation process assumes that the DASD devices for the GDDM product data sets can support a block size of 32760 bytes. If you are using 3350 devices, you must change the block size to 19069 bytes (see “GDDM/MVS data set block sizes” on page 24).

Target system requirements

This section describes the environment of the target system required to install and use GDDM/MVS.

Operating system requirements

GDDM/MVS operates under the MVS/ESA operating system.

Machine requirements

GDDM/MVS runs on any IBM System/370 or System/390 processor with sufficient storage to meet the combined storage requirements of the host operating system, access methods, DB/DC, or interactive subsystem, and GDDM.

DASD storage requirements

Table 8 on page 19 and Table 9 on page 21 list the target and distribution libraries (data sets) and their attributes required to install GDDM/MVS and other GDDM licensed programs.

The installation process assumes that the DASD devices for the GDDM product data sets can support a block size of 32760 bytes. If you are using 3350 devices, you must change the block size to 19069 bytes (see “GDDM/MVS data set block sizes” on page 24).

Program considerations

There are no program, system, or special considerations for GDDM/MVS.

Chapter 6. Installation instructions

This section describes the installation method and the step-by-step procedures to install and to activate the functions of GDDM/MVS, and its associated products GDDM-PGF, GDDM-IMD, GDDM-GKS, and GDDM-IVU.

Notes:

1. You must install the national language support for at least one language. For more information about the national language support available for the GDDM licensed programs, see Appendix C, “GDDM/MVS National language support” on page 87.
2. GDDM-PCLK, GDDM-OS/2 Link, and GDDM-REXX are part of GDDM Base. To find out how to make any of them available to your users after installing GDDM/MVS, refer to the information in Chapter 7, “Post-installation tasks” on page 33.

Year 2000 support for GDDM/MVS and GDDM-PGF

GDDM/MVS is an element of OS/390; GDDM-PGF is an optional feature of OS/390. Beginning with OS/390 Version 1 Release 2, OS/390 is certified as a Year 2000-ready operating system by the Information Technology Association of America (ITAA). Follow-on releases are also Year 2000 ready. GDDM/MVS Version 3 Release 2 is Year 2000 ready.

Previous products, such as OS/390 Version 1 Release 1 and the following products are Year 2000 ready with maintenance applied.

- OS/390 V1R1 and GDDM V3R1.1

  APARs PN73783 and PN82044.
- GDDM/MVS V3R1

  APARs PN73783 and PN82044.
- GDDM/MVS V2R3

  APARs PN72852 and PN81195.
- GDDM-PGF

  All releases applicable to the above levels are Year 2000 ready.

Previous products that are not Year 2000 ready will not be marketed after December 31st, 1997.

The following changes were made to ensure Year 2000 readiness:

- Module ADMESQ0O issues the TIME macro and receives the reponse in the form 0CYYDDDF where C is the century indicator. Updates have been made to recognize the century indicator which has a value of 1 for dates from 2000 onward.
- Old dates in GDDM are held in the YYDDD format. GDDM designates dates with a year value of 80 (the lowest possible) or higher as being 20th century dates, and those with a lower value as being 21st century dates.

Migrating from earlier releases

GDDM/MVS Version 3 Release 2 Modification 0 supports the following levels of its associated Licensed Programs:

GDDM-PGF Version 2 Release 1 Modification 3
GDDM-IMD Version 2 Release 1 Modification 3
GDDM-IVU Version 1 Release 1 Modification 3
GDDM-GKS Version 1 Release 1 Modification 3

If you have installed any of these programs at an earlier level, you must order the latest level and reinstall them.

Notes:

1. GDDM/MVS and GDDM/MVS National Language support are now supplied on the same tape.
2. If you are migrating from GDDM/MVS Version 2 you should be aware of the following changes in GDDM/MVS Version 3:
   - Since Version 2 Release 3, all GDDM data sets have a target and a distribution version.
   - The names of the GDDM/MVS data sets have changed.
   - The DCB characteristics of some GDDM/MVS data sets have changed.
   - GDDM/MVS must be installed into new target and distribution libraries because of the new names and DCB characteristics.
   - Sample JCL is provided for all installation tasks.
   - A CLIST is provided to customize the installation JCL.

General considerations are described first, followed by any subsystem-specific considerations.

The planning process for migrating existing users from an earlier level of GDDM comprises these steps:

1. Check the level of MVS that exists at your enterprise:
   - Check in the *GDDM General Information* manual that the various software components of your computer system have been brought up to the required service levels. If they have not, you may have to apply service to the system.
   - Check the levels of the GDDM licensed programs on your system.
   - Check for prerequisites, special requirements, known errors, and so on that may affect GDDM/MVS.
   - If migrating under TSO, ensure that you are using the correct level of TSO/E.
   - If migrating under CICS, ensure that you are using CICS/ESA Version 3 Release 3.0 or later.
   - If migrating under IMS, ensure that you are using IMS Version 2 Release 2 or later.
2. Decide which GDDM licensed programs you are going to update; see the *GDDM General Information* manual.

3. Check the memory and disk space requirements of your computer system, especially if you are adding new licensed programs; see the *GDDM System Customization and Administration* book.
4. Review your existing telecommunication network arrangements. See the *GDDM System Customization and Administration* book.
5. Unless you have done so already, order new tapes for GDDM/MVS, GDDM-PGF, GDDM-IVU, GDDM-GKS, and GDDM-IMD as necessary.
6. Decide which national languages you are going to install; see Appendix C, "GDDM/MVS National language support" on page 87.

   The choice of NLS for GDDM/VM at ordering time determines which language the hard copy publications are to be shipped in if they have been translated. The same tapes are shipped, regardless of which NLS you specified. Remember, if you are planning to install a DBCS national language at your enterprise, you must also install U.S. English.

   Note: You now have to specify U.S. English in the same way as any other language; U.S. English is no longer automatically available as the default language.
7. Order new GDDM publications, as required. See Table 3 on page 5.
8. Accept delivery of the GDDM Version 3 Release 2 installation package and check the contents. Ensure that you have the GDDM installation tapes you require, and that each tape has its own Program Directory.
9. Ensure you have the necessary installation documentation to hand. For migrating users on GDDM/MVS you need:
   - A Program Directory for each GDDM licensed program you are updating
   - *SMP/E User's Guide*, SC28-1302
   - The documentation supplied with the system package if you are using that method.

   You may also need to refer to one or more of the MVS subsystem books.
10. Update the licensed programs as required. For detailed information, see the appropriate Program Directories.
11. Check with the preventive service planning (PSP) "bucket" for late-breaking information about installation. The name of the PSP bucket is given in the Program Directory for each licensed program. If you are not sure about this step, ask your IBM representative.
12. File the Program Directories for future reference.

Further considerations for migrating users

When you upgrade GDDM from an earlier release:

- You are advised to retain a copy of the old version of GDDM for use during the changeover period.
- If you have changed any of the GDDM default modules, such as the external defaults module, you need to retain a separate copy of your changed version, otherwise GDDM overwrites your changes. When GDDM Version 3 Release 2 has been installed, you must merge your changes with the new level and recreate the executable version. A list of the GDDM default modules and the tasks you need to do if you have changed them is given at "Updating GDDM defaults" on page 33.

- Similarly, you must save any GDDM objects such as symbol sets that you have changed or that have been added from other products, if you want to continue to use them. New DCB characteristics have been used for GDDM Version 3 Release 2, so they also need reloading. See “Reblocking any existing objects” on page 61.
- If you have relinked GDDM for performance reasons, or linked parts of GDDM other than the interface stubs with your own applications, you must do the link-edit again, with the new release of GDDM.
- Installation of GDDM 3.2 overwrites the GDDM-supplied CGM conversion profiles already on your system. If you have customized any of the standard GDDM CGM profiles, make backup copies of these before you begin the installation. The ADMCGM conversion profile has been updated to include comprehensive help and explanatory texts. If you wish to use these new features, remember to repeat any changes you have made to your existing version of the profile.
- The ADMDATRN table has changed. If this table has been customized following the previous installation on your system, you will need to repeat these changes after installing GDDM 3.2. Make a record of the changes you have made to the table before you begin the installation.
- The ADMUSP4 sample application program has been changed. The new version of this program overwrites the old one if you reinstall GDDM. If you wish to safeguard the old version, make a backup copy of it before installation and rename it.

Considerations with CICS: From a CICS point of view, GDDM is an application needing definitions for programs, transaction identifiers, and data sets. Therefore, if you already have installation guidelines for installing new or upgraded versions of CICS applications, follow them for GDDM.

Because many of the module names are the same as those in earlier releases, you cannot use more than one set of GDDM code in the same CICS system. You need to create copies of the GDDM VSAM data sets, because the members have the same names.

For testing, many CICS users have a second CICS system onto which they install and test new applications. This enables the production system to continue undisturbed until the new applications, in this case GDDM and its associated licensed programs, are tested and ready for use.

If a concurrent production and test CICS environment is not practical at your enterprise, create a copy of the production CICS startup job stream, giving it a different name. In the copy, refer to the new data sets. Provide System Information Table (SIT) overrides to use new versions of your tables with the new GDDM-specific entries.

During off-peak hours, the production system can be brought down and the test version started up. The GDDM-specific items can be checked out and adjustments made, until the test version is ready for use in the production environment.

Then, the production CICS startup jobstream can be adjusted as needed, using the experience gained from the test version. The original production CICS startup

should be kept as a backup until the new one has been exercised by users and confirmed to be working correctly.

Considerations with IMS: You need to reorganize your GDDM object databases by adding a secondary index. This is explained in "Loading the GDDM IMS databases" on page 58.

You have now completed planning for migrating users from an earlier level of GDDM; continue reading below.

GDDM/MVS can be installed using any of the following methods:

- As an element of OS/390. See the supplied OS/390 documentation.
- As part of a system package, such as CBIPO, CBPDO, or CustomPac
- Stand-alone using SMP/E

Do not use this program directory if you are installing GDDM/MVS with an MVS Custom-Built Installation Process Offering (CBIPO) (5751-CS1). Instead, use the CBIPO Related Installation Materials (RIMs) provided with the CBIPO. The CBIPO RIMs point you to specific sections of the program directory as required.

If you are installing GDDM/MVS using the MVS Custom-Built Product Delivery Offering (CBPDO) (5751-CS3), use the softcopy program directory provided on the CBPDO tape. Your CBPDO contains a softcopy preventive service planning (PSP) upgrade for this product. All service and HOLDDATA for GDDM/MVS are included on the CBPDO tape. Follow the instructions supplied with the CBPDO package to RECEIVE GDDM/MVS onto your system. Follow the steps in this program directory to complete the installation, substituting the names you have used to RECEIVE GDDM/MVS where appropriate, and omitting the RECEIVE step as shown in "SMP/E RECEIVE" on page 29 of this program directory.

If you are installing GDDM/MVS as part of a CustomPac, refer to the documentation supplied with the package for instructions. When you have installed GDDM/MVS, refer to "Library contents after installation" on page 31 for information on where GDDM/MVS resides, to Chapter 7, "Post-installation tasks" on page 33 for information on activating GDDM/MVS and to Chapter 8, "Testing GDDM installation" on page 63 for information on testing GDDM/MVS.

If you are installing GDDM/MVS stand-alone, follow the instructions in "Installing GDDM/MVS."

Installing GDDM/MVS

The installation process for GDDM/MVS consists of four tasks:

1. Preparation.

   Before installing GDDM/MVS, you must choose names and values for a number of parameters. These names and values depend on the standards that apply to your enterprise.

2. Reading and customizing the sample installation JCL.

   The names and values you have chosen are used to modify the sample installation JCL for your enterprise.

3. Creating the installation environment.

You must tailor the SMP/E environment for GDDM/MVS and create the new data sets.

4. Using SMP/E to install GDDM/MVS

   You use the SMP/E RECEIVE, APPLY, and ACCEPT commands to read the installation tape and place GDDM/MVS into the appropriate system libraries. If you need more information than is given in this document, refer to a full description of SMP/E in the *SMP/E User's Guide* and *SMP/E Reference* books.

   SMP/E is also used to install service on GDDM/MVS after it has been installed.

Prepare to install GDDM/MVS

The names and values you choose here when planning the installation are required later for customization of the sample installation JCL. You should note these down as you work through this section.

You must plan the following:

- Which languages you are going to install (see Appendix C, "GDDM/MVS National language support" on page 87)
- Space requirements
- High-level qualifiers
- Volume and unit names
- Data set block sizes
- SMP/E environment

GDDM/MVS space requirements

Use Table 8 on page 19 and Table 9 on page 21 to plan the space requirements for the target and distribution libraries (data sets) required to install GDDM/MVS and other GDDM licensed programs.

Note: The DASD storage requirements for all the GDDM licensed programs are given here to enable you to calculate the total DASD storage requirement of the GDDM licensed programs you intend to install. If you ensure that you have sufficient DASD storage available before you begin installation of GDDM/MVS, you will not have to increase the DASD storage when you install other GDDM licensed programs.

Table 8 (Page 1 of 2). Storage Requirements for Target Libraries

Data Set Name or Library Name	TYPE	DSORG	RECFM	LRECL	BLK SIZE	No. of BLKS	No. of DIR BLKS
SADMCDA							
GDDM-PGF	NU	PO	FB	400	8800	10	1
SADMCFO							
GDDM-PGF	NU	PO	FB	400	8800	10	1
SADMDAT							
GDDM-PGF	NU	PO	VB	255	6144	260	5
SADMGDF							
GDDM/MVS	NM	PO	FB	400	8800	10	5
+ GDDM-PGF						+60	+5
SADMIMG							
GDDM/MVS	NU	PO	FB	400	8800	10	5
SADMMAP							
GDDM-IMD	NM	PO	FB	400	8800	160	20
+ GDDM-IVU						+10	+2
+ GDDM/MVS NLS per language						+75	+2
SADMMOD							
GDDM/MVS	NM	PO	U	0	32760	170	60
+ GDDM-PGF						+50	+20
+ GDDM-IMD						+20	+10
+ GDDM-IVU						+15	+5
+ GDDM-GKS						+15	+5
+ GDDM/MVS NLS per language						+20	+15
SADMMSG							
GDDM/MVS	NU	PO	FB	80	8800	10	5
SADMOPS							
GDDM-PGF	NU	PO	VB	255	6144	65	5
SADMPCF							
GDDM/MVS	NU	PO	FB	400	8800	600	10
SADMPNL							
GDDM/MVS	NU	PO	FB	80	8800	30	10
SADMSAM							
GDDM/MVS	NM	PO	FB	80	8800	900	15
+ GDDM-PGF						+80	+2
+ GDDM-IMD						+30	+2

Table 8 (Page 2 of 2). Storage Requirements for Target Libraries

Data Set Name or Library Name	TYPE	DSORG	RECFM	LRECL	BLK SIZE	No. of BLKS	No. of DIR BLKS
+ GDDM-IVU						+65	+2
+ GDDM-GKS						+10	+2
+ GDDM/MVS NLS per language						+15	+1
SADMSYM							
GDDM/MVS	NM	PO	FB	400	8800	480	20
+ GDDM-PGF						+10	+1
+ Kanji symbol sets (part of GDDM/MVS NLS Japanese (Kanji))						+500	+5
+ Simplified Chinese symbol sets (part of GDDM/MVS NLS Simplified (PRC) Chinese)						+100	+5

Notes:

1. The values for the number of blocks and directory blocks include an extra 25% for any service and upgrades to be installed.
2. These data sets can be allocated as part of the normal installation process by the supplied job ADMUJALO.
3. The DDDEFs for these data sets can be allocated as part of the normal installation process by the supplied ADMUJDDF job.
4. The data sets can be reblocked to a larger size. Refer to "GDDM/MVS data set block sizes" on page 24 for more details.
5. Abbreviations used in this table are:
   - for the data set type (TYPE):
     NU New data set used by this program only.
     NM New data set used by more than one program.
   - for the data set organisation (DSORG):
     PO Partitioned Organization
   - for the record format (RECFM):
     U Unblocked
     FB Fixed block
     VB Variable block
   - LRECL is an abbreviation for logical record length.

Table 9 (Page 1 of 2). Storage Requirements for Distribution Libraries

Data Set Name or Library Name	TYPE	DSORG	RECFM	LRECL	BLK SIZE	No. of BLKS	No. of DIR BLKS
AADMCDA							
GDDM-PGF	NU	PO	FB	400	8800	10	1
AADMCFO							
GDDM-PGF	NU	PO	FB	400	8800	10	1
AADMDAT							
GDDM-PGF	NU	PO	VB	255	6144	260	5
AADMGDF							
GDDM/MVS	NM	PO	FB	400	8800	10	5
+ GDDM-PGF						+60	+5
AADMIMG							
GDDM/MVS	NU	PO	FB	400	8800	10	5
AADMMAP							
GDDM-IMD	NM	PO	FB	400	8800	160	20
+ GDDM-IVU						+10	+2
+ GDDM/MVS NLS per language						+75	+2
AADMMOD							
GDDM/MVS	NM	PO	U	0	32760	280	220
+ GDDM-PGF						+85	+70
+ GDDM-IMD						+30	+40
+ GDDM-IVU						+20	+10
+ GDDM-GKS						+40	+50
+ GDDM/MVS NLS per language						+30	+25
AADMMSG							
GDDM/MVS	NU	PO	FB	80	8800	10	5
AADMOPS							
GDDM-PGF	NU	PO	VB	255	6144	65	5
AADMPCF							
GDDM/MVS	NU	PO	FB	400	8800	600	10
AADMPNL							
GDDM/MVS	NU	PO	FB	80	8800	30	10
AADMSAM							
GDDM/MVS	NM	PO	FB	80	8800	900	15
+ GDDM-PGF						+80	+2
+ GDDM-IMD						+30	+2

Table 9 (Page 2 of 2). Storage Requirements for Distribution Libraries

Data Set Name or Library Name	TYPE	DSORG	RECFM	LRECL	BLK SIZE	No. of BLKS	No. of DIR BLKS
+ GDDM-IVU						+65	+2
+ GDDM-GKS						+10	+2
+ GDDM/MVS NLS per language						+15	+1
AADMSYM							
GDDM/MVS	NM	PO	FB	400	8800	480	20
+ GDDM-PGF						+10	+1
+ Kanji symbol sets (part of GDDM/MVS NLS Japanese (Kanji))						+500	+5
+ Simplied Chinese symbol sets (part of GDDM/MVS NLS Simplified (PRC) Chinese)						+100	+5

Notes:

1. The values for the number of blocks and directory blocks include an extra 25% for any service and upgrades to be installed.
2. These data sets can be allocated as part of the normal installation process by the supplied job ADMUJALO.
3. The DDDEFs for these data sets can be allocated as part of the normal installation process by the supplied ADMUJDDF job.
4. The data sets can be reblocked to a larger size. Refer to “GDDM/MVS data set block sizes” on page 24 for more details.
5. Space for the GDDM SMP/E temporary data sets, equivalent to the space for the GDDM distribution data sets, is also required during installation.
6. Abbreviations used in this table are:
   - for the data set type (TYPE):
     NU New data set used by this program only.
     NM New data set used by more than one program.
   - for the data set organisation (DSORG):
     PO Partitioned Organization
   - for the record format (RECFM):
     U Unblocked
     FB Fixed block
     VB Variable block
   - LRECL is an abbreviation for logical record length.

High-level qualifiers

Use Table 10 to plan the high-level qualifiers for the GDDM/MVS data sets.

Table 10. High-level qualifiers for data set names

Parameter Name in the Sample JCL Provided	Default Value	Description
SHLQUAL	GDDM.SMPE	High-level qualifiers for the new SMP/E data sets that you are going to use to install GDDM/MVS. If you are going to install into existing data sets, this name is already defined.
THLQUAL	GDDM	High-level qualifiers for the GDDM/MVS target data sets.
DHLQUAL	GDDM.DLIB	High-level qualifiers for the GDDM/MVS distribution data sets.

It is recommended that GDDM data sets be protected with RACF (or another security product) and be cataloged. If you plan to add the SADMMOD library to the link list, it must be cataloged. For more information about adding libraries to the link list, see the *OS/390 MVS Initialization and Tuning Reference* manual.

Note: The TSO user ID used to install or service GDDM/MVS requires ALTER access to these high-level qualifiers. Using GDDM after installation requires READ access to the THLQUAL data sets, and READ access to the `DHLQUAL.AADMMOD` data set if you wish to link edit applications.

If you are going to use the Spool Display and Search Facility (SDSF) to look at the output from the installation jobs, you must have authorization to use the prefix `ADM*` to view them, or you must change the names of the jobs to use a prefix to which you are authorized after you have customized them.

Volume and unit names

Plan the volume and unit names shown in Table 11 on page 24. These are used by the supplied sample JCL during the installation of GDDM/MVS. You must decide where you are going to install the product, taking into account the space requirements (see Table 9 on page 21 and Table 8 on page 19) and how frequently the data sets are going to be accessed.

Table 11. Volume and Unit Names		
Parameter Name in the Sample Jobs	**Values Required**	**Description**
SMPVOL	VOLSER and unit type	SMP/E data set volume if installing into new SMP/E data sets. Also used for temporary libraries.
TARVOL	VOLSER and unit type	GDDM/MVS target data set volume.
DISVOL	VOLSER and unit type	GDDM/MVS distribution data set volume.
TAPEUNIT	Tape unit type	Unit type of the tape drive used to read the GDDM/MVS tapes.

Note: You can use esoteric values such as SYSDA or SYSALLDA for the unit type of TARVOL and DISVOL by substituting a period (.) for the VOLSER in the customization step in section “Customize the sample installation JCL” on page 26. SMPVOL must have both the VOLSER and the unit type specified.

GDDM/MVS data set block sizes

Plan the block sizes of the GDDM data sets. The block sizes of GDDM/MVS Version 3 data sets can be changed from the default values shown in Table 12 to any valid number allowed by the operating system and hardware being used. These default values are designed for installation on most types of DASD but can be changed to meet local requirements. If you create large GDDM objects such as GDF files, you can reduce load times by increasing BLKFB400. If you are installing onto 3350 DASD, the maximum block size you can specify is 19069 bytes.

If you have Storage Management Subsystem (SMS) active on your system, it may override the values you choose when you create the data sets.

Note: Any existing BLKSIZE=400 objects that you may have defined, such as user symbol sets or GDDM-IMD maps, should be re-blocked to match the value of BLKFB400 that you have chosen. If you do not change all of your object data sets to the same block size, you may have problems concatenating the data sets. This procedure is described in “Reblocking any existing objects” on page 61.

Table 12. Data set blocksizes		
Parameter Name in the Sample Jobs	**Default Value**	**Description**
BLKFB80	8800	LRECL=80 data sets such as JCL samples
BLKFB400	8800	LRECL=400 data sets such as symbol sets
BLKU	32760	Unblocked data sets such as load libraries
BLKV	6144	Variable blocked data sets, such as ADMOPS presentations

SMP/E environment

You must decide whether to install GDDM/MVS into new or existing SMP/E data sets.

Using new data sets is simpler because GDDM/MVS supplies sample jobs to allocate and initialize a complete SMP/E environment. If you choose to use new data sets, use the default values for the parameters shown in the table here. Specifying NEW for `GZONECSI` means that the customization step adds the name `SHLQUAL.GLOBAL.CSI`.

If you use existing SMP/E data sets, you need to know the names of your target zone and distribution zone, and the data set name of your global CSI. These will be used to replace the default values shown in Table 13.

Table 13. Target and Distribution Zone Names

Parameter Name in the Sample Jobs	Default Value	Description
GZONECSI	NEW	The data set name of your global zone CSI
TZNAME	TZONE	The name of your target zone
DZNAME	DZONE	The name of your distribution zone

Remember that installing a new release of GDDM into the same SMP/E zones as an existing release will delete the earlier level from both the SMP/E zones and the existing release data sets.

If you are going to install into the same SMP/E zones, then to preserve your current release while you install and test the new release you should copy the existing release data sets to new data sets with different names. Do not rename or delete the current data sets because they need to be available to SMP/E for delete processing during the installation of the new release. If they are not available, the APPLY and ACCEPT steps will fail with a return code of 12. After running a successful ACCEPT of the GDDM products, the previous release data sets are no longer required and can be deleted.

Create the sample installation JCL

You must read in the sample installation JCL and customize it for your enterprise.

Read in the sample installation JCL

To read the sample installation JCL, you can RECEIVE GDDM/MVS and copy the sample JCL from the `SHLQUAL.HGD3200.F4` SMP/E temporary data set, or, if you are installing GDDM/MVS from a standalone tape, you can read the sample JCL from the tape. To read the sample JCL from a CBPDO tape, check the CBPDO *Memo to Users* which contains cross-references between the GDDM-supplied file numbers and the CBPDO-supplied file numbers. You cannot read the samples from a CBIPO tape using this JCL because the file number for the installation samples is not correct. Decide which method you are going to use and follow this procedure.

1. Allocate the data set `DHLQUAL.AADMINS` using JCL or ISPF/PDF panel 3.2 where `DHLQUAL` is the name you have chosen in the planning section. The data set should have the following attributes:

```
SPACE UNITS        ===> TRACKS
PRIMARY QUANTITY   ===> 15
SECONDARY QUANTITY ===> 3
DIRECTORY BLOCKS   ===> 20
RECORD FORMAT      ===> FB
RECORD LENGTH      ===> 80
BLOCK SIZE         ===> 8800
```

If the data set exists, you do not need to reallocate it as long as there is enough space remaining in it for the sample jobs.

2. Type in the following JCL as member ADMSETUP in DHLQUAL.AADMINS, substituting the values you have chosen for SHLQUAL, DHLQUAL and TAPEUNIT into the JCL, and adding job card information if required:

```
//ADMSETUP JOB
//AALLOC   EXEC PGM=IEBCOPY,REGION=1024K
//SYSPRINT DD SYSOUT=*
//*
//* Use the next two lines if you are reading the samples
//* from the installation tape
//IN       DD DSN=IBM.HGD3200.F4,DISP=OLD,LABEL=(5,SL),
//            UNIT=tapeunit,VOL=SER=GD3200
//*
//* Use the next line if you are reading the samples
//* from the SMP/E temporary data set
//IN       DD DSN=shlqual.HGD3200.F4,DISP=OLD
//*
//OUT      DD DSN=dhlqual.AADMINS,DISP=SHR
//SYSUT3   DD SPACE=(TRK,(5)),UNIT=SYSDA
//SYSUT4   DD SPACE=(TRK,(5)),UNIT=SYSDA
//SYSIN    DD *
    COPY INDD=((IN,R)),OUTDD=OUT
/*
```

3. If you are reading the samples from the installation tape, mount the tape.
4. Run the above job to load the samples. The job should finish with a return code of 0.

   If the return code is not 0, check the job output carefully to ensure that the sample jobs have been copied correctly. If the sample jobs have not been copied correctly, correct any errors found, and re-submit the job. If you have to re-submit this job several times, it may be necessary for you to compress the AADMINS data set.

You have now read all the sample jobs into the data set DHLQUAL.AADMINS. The next step is to customize the sample jobs for your system.

Customize the sample installation JCL

The sample jobs are customized by adding the names you chose in section “Prepare to install GDDM/MVS” on page 18 to the job called ADMINST in DHLQUAL.AADMINS and running this job. ADMINST calls a CLIST to copy and edit the sample jobs from DHLQUAL.AADMINS to THLQUAL.SADMINS. If THLQUAL.SADMINS does not exist, it is created by ADMINST.

The following rules apply when updating ADMINST:

- The keywords in the JCL can appear in any order and starting in any column as long as they are the first nonblank characters on a line and are within columns 1 through 72.

- If you place a nonblank character before a keyword, the line is treated as a comment. Use an asterisk as the nonblank character to avoid accidentally forming a valid keyword.
- Do not delete lines in ADMINST, as you may need them in the future. If they do not apply, comment them out by placing an asterisk at the beginning of the unwanted lines.

Use your usual editor to update ADMINST in `DHLQUAL.AADMINS` as follows:

1. Change the job statement at the top of ADMINST to match the requirements of your enterprise. If you need to code a 'TIME=' parameter on your JCL, a value of 10 minutes CPU time is enough to run ADMINST.
2. Customize the JCL statements at the top of ADMINST with the names you have chosen for `THLQUAL` and `DHLQUAL` to replace the variables THQ and DHQ for the SYSPROC, GINPUT, and GOUTPUT lines.
3. Choose which GDDM licensed programs and which national languages you are going to install. There is a line in ADMINST for GDDM/MVS and for each of its associated licensed programs. Select the lines for the programs you are going to install by overtyping the asterisk at the beginning of the appropriate lines with a blank.

   It is important for the data set allocation sample JCL that you include all of the programs you intend to install, because the GDDM data set sizes are calculated based on this product set.

   If you plan to install further GDDM licensed programs now or in the future, you should select them in the list of programs now. When you have completed customizing the sample installation JCL, you must manually edit the ADMUJAPP and ADMUJACC SMP/E APPLY and ACCEPT jobs in the `THLQUAL.SADMINS` data set to remove the FMIDs of the products you are not installing now. The FMIDs are given in Table 14 on page 30.
4. Choose which sample jobs you want to customize. ADMINST is supplied with no jobs selected, so it will customize all jobs. This is correct for a new installation but, if you re-run ADMINST for any reason, you may want to select specific jobs by overtyping the asterisk at the beginning of the appropriate lines with a blank.
5. Type job card information for the sample jobs. You must leave the characters 'JOB //XXXXXXXX JOB' intact on the first line but should add any other parameters to this line as though it were the real job card.

   You can enter more than one line for your JOB statement. Begin each line with the keyword JOB, followed by at least one blank. Type the required parameters, including all punctuation such as start of line '//' and continuation commas. You can also add lines such as /*ROUTE or /*JOBPARM by using the JOB keyword in ADMINST.

   If your enterprise needs a 'TIME=' parameter, you must specify a value large enough to run the SMP/E apply step. This can take up to 120 minutes for GDDM and all of its associated licensed programs on a small processor.
6. Type the values you have chosen for your high-level qualifiers, block sizes, volume serial numbers, and unit types.
7. Type the SMP/E values you have chosen.

If you use your existing SMP/E data sets, you must customize the names of your target zone, distribution zone, and global CSI data set. If you intend to let GDDM create a new SMP/E environment, leave the SMP/E information as it is.

8. Run ADMINST. It should finish with a return code of 0.

   If the return code is not 0, inspect the output to discover what caused the problem. ADMINST can be run again when the error has been corrected, but running it many times may cause the output data set `THLQUAL.SADMINS` to need compressing.

Now carry out the installation steps in the following sections, using the customized versions of the sample jobs from the `THLQUAL.SADMINS` data set.

Create the installation environment

You must create the environment in which to run SMP/E to install GDDM/MVS.

Create or review your SMP/E environment

Either run the job ADMUJSMP from `THLQUAL.SADMINS` to create a new SMP/E environment, or review your existing environment.

Run ADMUJSMP: All steps of ADMUJSMP should finish with a return code of 0. If any step finishes with a return code other than 0, inspect the job output listing to determine what caused the problem. When you have corrected the problem, either rerun the job from the step that failed, or delete all the data sets created with the high-level qualifier `SHLQUAL`, and run the whole job again.

Note: If you wish to run ADMUJSMP under JES3, you must first uncomment the DUMMYCSI step in the JCL. If you do not do this, the job will fail with a JCL error because the VSAM data sets are defined and used in the same job stream.

Review your existing SMP/E environment: Ensure that the following SMP/E global zone values are available:

DSSPACE (100,50,300) or greater
PEMAX 2500 or greater.

In addition to the above values you are strongly recommended to change the UTILITY linkage product entry (IEWL for the Linkage editor, IEWBLINK for Binder) so that the RC subentry has a value of 4 instead of the default of 8. When this value is 8, SMP/E can permit link failures which are severe enough to prevent GDDM/MVS from working and yet still give a return code of 0 or 4 for the APPLY job.

The sample installation JCL supplied with GDDM/MVS uses DDDEFs to allocate all the SMP/E data sets except SMPCSI and SMPPTFIN. This is to allow users of existing SMP/E environments to use the sample installation JCL, but allow them to use their own names for the SMP/E data sets using DDDEFs or a JCL procedure. If you do not have these values available, you can copy the ADMUJSMP PRIMESMP step and edit it to include any values that are not defined.

Update SMP/E DDDEF information

SMP/E needs DDDEF information for the GDDM/MVS data sets, whether you are installing into new or old SMP/E data sets.

Run the job ADMUJDDF from THLQUAL.SADMINS to build your SMP/E DDDEF statements. It should finish with a return code of 0 or 4, and is designed to replace the DDDEFs if they already exist.

If the return code is not 0 or 4, check the job output carefully to ensure that the DDDEFs have been allocated correctly. ADMUJDDF can be rerun if required.

Allocate distribution and target libraries for GDDM/MVS

Before using SMP/E to install GDDM/MVS, you must allocate two groups of data sets, the target and distribution libraries, for GDDM/MVS. If you have previously installed GDDM/MVS Version 3 these data sets may already exist, in which case you may need to increase their size. Do not delete the previous release data sets; they are required by SMP/E.

Run the job ADMUJALO from THLQUAL.SADMINS to allocate the libraries. The job should complete with a return code of 0.

If the return code is not 0, check the job output carefully to ensure that the data sets have been allocated correctly. If you need to rerun the job, you must delete any data sets that were created by the unsuccessful attempt. These all begin with the THLQUAL or DHLQUAL high-level qualifiers. (Be careful not to delete any SMP/E data sets.)

ADMUJALO always creates the data sets required for GDDM-PGF. If you are not installing GDDM-PGF, these data sets are of minimal size and can be ignored or deleted.

Use SMP/E to install GDDM/MVS

SMP/E uses three steps to install each part of GDDM/MVS:

1. RECEIVE
2. APPLY
3. ACCEPT

Note: You must run the steps in this order.

SMP/E RECEIVE

Receive each GDDM/MVS product using SMP/E by mounting the tape and running the sample job for each product you install from data set THLQUAL.SADMINS. (See Table 14 on page 30 for the names of the receive jobs and the tape volume serial numbers.)

Notes:

1. You must receive GDDM/MVS before you can receive any of the associated products, and you must receive all the products that you selected at JCL customization before moving on to the APPLY step.
2. If you are installing GDDM/MVS as part of a CBPDO, you do not need to run this step. Instead, follow the instructions supplied with the CBPDO package to receive GDDM/MVS.

All the RECEIVE jobs should complete with a return code of 0.

If the return code is not 0, check the job output listing to find out why the job failed, correct the error, and then rerun the job.

Table 14. SMP/E RECEIVE job data

Product	RECEIVE Job Name	Tape Volume Serial Number	FMID
GDDM/MVS	ADMUJRB	GD3200	HGD3200
GDDM-PGF	ADMUJRP	GD3201	HGD3201
GDDM-IMD	ADMUJRI	GD3202	HGD3202
GDDM-GKS	ADMUJRJ	GD3203	HGD3203
GDDM-IVU	ADMUJR5	GD3204	HGD3204
GDDM/MVS NLS - US English	ADMUJRB	GD3200	JGD3219
GDDM/MVS NLS - Brazilian	ADMUJRB	GD3200	JGD3220
GDDM/MVS NLS - Simplified Chinese	ADMUJRB	GD3200	JGD3221
GDDM/MVS NLS - Danish	ADMUJRB	GD3200	JGD3222
GDDM/MVS NLS - French	ADMUJRB	GD3200	JGD3223
GDDM/MVS NLS - German	ADMUJRB	GD3200	JGD3224
GDDM/MVS NLS - Korean (Hangeul)	ADMUJRB	GD3200	JGD3225
GDDM/MVS NLS - Italian	ADMUJRB	GD3200	JGD3226
GDDM/MVS NLS - Japanese (Kanji)	ADMUJRB	GD3200	JGD3227
GDDM/MVS NLS - Norwegian	ADMUJRB	GD3200	JGD3228
GDDM/MVS NLS - French (Canadian)	ADMUJRB	GD3200	JGD3229
GDDM/MVS NLS - Spanish	ADMUJRB	GD3200	JGD3230
GDDM/MVS NLS - Taiwan Chinese (Traditional)	ADMUJRB	GD3200	JGD3231
GDDM/MVS NLS - Swedish	ADMUJRB	GD3200	JGD3232

SMP/E APPLY

Run the sample job ADMUJAPP from data set `THLQUAL.SADMINS` to apply all the products selected when you customized the JCL. This job can take a long time to run, depending on the capacity of your system, and on what other jobs are running.

The APPLY step should complete with a return code of 0 or 4. Message GIM61903 for the ++DELETE statements can be ignored.

If the return code is greater than expected, check the output listing to find out why, and correct the error.

The job can be rerun, but if some of the products were successfully applied, you must first delete the lines from ADMUJAPP that refer to these product FMIDs. Products successfully applied result in the following message in the output listing of ADMUJAPP:

```
GIM22701I APPLY PROCESSING WAS SUCCESSFUL FOR SYSMOD xxxxxxx
```

(where xxxxxxx is the FMID).

If required, GDDM can be tested after the APPLY step has completed successfully.

If you are installing any of the GDDM associated products (such as GDDM-PGF), you must either APPLY them at the same time as GDDM/MVS or you must ACCEPT GDDM/MVS before you run the APPLY step for the associated product(s).

SMP/E ACCEPT

Run the sample job ADMUJACC from data set `THLQUAL.SADMINS` to accept all the products selected when you customized the JCL. This job takes a few minutes to run, depending on the capacity of your system, and on what other jobs are running.

The ACCEPT step should complete with a return code of 0 or 4. Message GIMxxxxx for the ++DELETE statements can be ignored.

If the return code is not 0, check the job output listing to find out why the job failed and correct the error. The job can be rerun, but if some of the products were successfully accepted, you must first delete the lines from ADMUJACC that refer to these product FMIDs. Products successfully accepted result in the following message in the output listing of ADMUJACC:

```
GIM22701I ACCEPT PROCESSING WAS SUCCESSFUL FOR SYSMOD xxxxxxx
```

(where xxxxxxx is the FMID).

What SMP/E has done

You have now taken the information off the distribution tape that IBM supplied and installed it onto your system. You now need to test the installed system before making it available to users. Before GDDM can be used with CICS or IMS, it must be defined to the subsystem, and storage areas must be allocated. This is described in Chapter 7, “Post-installation tasks” on page 33.

Library contents after installation

After the successful completion of SMP/E processing, your target data sets have the names and contents shown in Table 15. They all have a corresponding distribution data set with a name prefix of AADM. The GDDM/MVS Version 2 data set GDDMLIB is now called AADMMOD.

Table 15 (Page 1 of 2). Library contents after installation

Data Set Name	Former Name	Description
SADMCDA	none	Sample GDDM-PGF ICU chart data.
SADMCFO	none	Sample GDDM-PGF ICU chart format.
SADMDAT	none	Help material and other supporting files for the GDDM-PGF OPS utility.
SADMGDF	none	Sample GDDM ADMGDFs.
SADMIMG	SGDDMSAM	Sample GDDM ADMIMGs.

Table 15 (Page 2 of 2). Library contents after installation		
Data Set Name	**Former Name**	**Description**
SADMINS	none	Working library containing the customized version of the installation samples if you used the sample GDDM JCL customization process.
SADMMAP	SGDDMMAP	GDDM-IMD and GDDM-IVU panels and tutorials, supplied as part of GDDM/MVS NLS.
SADMMOD	GDDMLOAD	GDDM executable code.
SADMMSG	none	ISPF messages for the GDDM Print Queue Manager (PQM).
SADMOPS	none	Sample presentations for the GDDM-PGF OPS utility.
SADMPCF	SGDDMSAM	GDDM-PCLK and GDDM-OS/2 Link system files.
SADMPNL	none	ISPF panels for the GDDM Print Queue Manager (PQM).
SADMSAM	SGDDMSAM	Sample application macros, CLISTs, EXECs, and job streams for later use.
SADMSYM	SGDDMSYM	GDDM/MVS sample symbol sets. These are listed in the *GDDM Base Application Programming Reference* book.

Note: In Table 15 on page 31, the column headed 'Former Name' gives the name of each data set prior to GDDM/MVS Version 3.

Chapter 7. Post-installation tasks

This chapter provides supporting information for any tasks that may need to be done after installation but before formal testing. These tasks comprise:

- Activating NLS
- Updating GDDM default modules (see “Updating GDDM defaults”)
- Defining GDDM to TSO (see “Defining GDDM to TSO” on page 34)
- Defining GDDM to CICS (see “Defining GDDM to CICS” on page 40)
- Defining GDDM to IMS (see “Defining GDDM to IMS” on page 49)
- Making GDDM-REXX available (not under CICS or IMS) (see “Making GDDM-REXX available” on page 61)
- Reblocking any existing objects to the new block size (see “Reblocking any existing objects” on page 61)

Activating NLS

This task applies only if you have installed a national language other than U.S. English.

If you have installed NLS for languages other than U.S. English, you use the NATLANG external default to define these languages to GDDM to enable your users to use them. You can define the GDDM NATLANG external default in one of two places, and these are accessed in the following order:

1. The external defaults file. This applies to TSO and CICS only.
2. The external defaults module: ADMADFT for TSO, ADMADFC for CICS, and ADMADFI for IMS.

To specify a national language other than U.S. English for your enterprise, add the following line to the external defaults file or the external defaults module:

```
 ADMMDFT NATLANG=N
```

where `N` defines the new language selected, in this instance Norwegian. The line must start with a space in column 1.

For more information about GDDM user-default specifications, see the *GDDM System Customization and Administration* book.

Updating GDDM defaults

During its operation, GDDM uses many values that you can change to customize GDDM for your installation. These values are known either as user default specifications (UDSs) or are held as entries in the GDDM default modules.

For information about how to update GDDM default modules or user default specifications, see the *GDDM System Customization and Administration* book.

Defining GDDM to TSO

This section describes the tasks you have to do to define GDDM to the TSO subsystem.

Data sets required for GDDM under TSO

To be able to run under TSO, GDDM needs some data sets to be allocated. In addition, to be able to save output from GDDM applications, other data sets are needed. To avoid users creating GDDM objects and then not being able to save them, you should set up an environment where all of the required data sets are available and allocated before calling GDDM applications.

There are two types of data set needed for running GDDM. These are:

- The installed GDDM target data sets.
- User data sets for saving or accessing user generated objects.

Below is a list of the GDDM target data sets that you may need to allocate before you can run GDDM.

ddname	Data set	Needed for
ADMPC	GDDM.SADMPCF	GDDM-PCLK or GDDM-OS/2 Link
ADMCGM	GDDM.SADMSAM	Conversion profiles for use with the computer graphics metafile (CGM) function.
ADMGGMAP	GDDM.SADMMAP	GDDM-IVU panels
ADMPROJ	GDDM.SADMMAP	GDDM-IVU sample projection
ADMIMG	GDDM.SADMIMG	GDDM-IVU sample image
ADMGIMP	GDDM.SADMMAP	GDDM-IMD panels
ADMGDF	GDDM.SADMGDF	The GDDM/MVS or GDDM-REXX verification program GDFs
ADMSYMBL	GDDM.SADMSYM	GDDM utilities or user applications that access the supplied symbol sets

Below is a list of the GDDM user data sets and their record lengths. They are all RECFM=FB, and are all partitioned except for ADMDECK.

ddname	LRECL	Used for
GDDM/MVS		
ADMDECK	80	Symbol sets in deck format. It is required for saving deck format symbol sets in the symbol set editors.
ADMGDF	400	GDF files. It is for GDDM/MVS and the ICU.
ADMGGMAP	400	Generated mapgroups that are produced by GDDM-IMD, and used by GDDM/MVS run-time mapping.
ADMIMG	400	Image data files. Used by GDDM/MVS and GDDM-IVU.
ADMPRINT	400	Files that are to be processed by the GDDM/MVS print utility. This data set is normally created dynamically by an application when print output is requested and deleted after printing by the print utility.

ddname	LRECL	Used for
ADMPROJ	400	Image projection files. Used by GDDM/MVS and GDDM-IVU.
ADMSAVE	400	GDDM pictures held in device-dependent data-stream format.
ADMSYMBL	400	Symbol sets. Required for GDDM/MVS and GDDM-PGF.
GDDM-PGF only		
ADMCDATA	400	The ICU.
ADMCDEF	400	The ICU.
ADMCFORM	400	The ICU.
GDDM-IMD only		
ADMGNADS	80	Generated ADSs produced by GDDM-IMD.
ADMIFMT	256	Export files from the GDDM-IMD export utility.
GDDM-GKS only		
ADMGKSM	400	GDDM-GKS metafiles.

All of these data sets can be created by running the CLIST ADMUCNEW from the SADMSAM data set. This creates the GDDM TSO user data sets with names formed by concatenating the TSO profile data set name prefix to the ddname of the data set. The data sets are created with default sizes. Use the information in the *GDDM System Customization and Administration* book to estimate the actual sizes you require. You should copy the complete CLIST and then delete the lines for the licensed programs you do not plan to install.

Other data sets you may need to allocate are ADMPRNTQ if you plan to use the TSO print utility, and ADMDEFS if you plan to use an external defaults file. For more information about these items, see the *GDDM System Customization and Administration* book.

To use the data sets, they also need to be allocated. This can be done by:

- Putting the relevant DD statements in the logon procedure.
- Running a user logon CLIST from the logon procedure.
- Allocating them using an application CLIST.

If you have added GDDM to the LPA list, you must move the two non-reentrant modules, ADMECOMT and ADMOPUT, into a separate loadlib that you must add to the STEPLIB definition for the SADMMOD data set in the logon procedure.

Managing the VSAM data sets for use with GDDM-IMD under TSO

Migrating from earlier releases

You do not need to do this task if these data sets for GDDM-IMD already exist.

ADMX — GDDM-IMD staging data set

This data set is necessary if you want to move GDDM-IMD maps to or from systems outside TSO. Do not create this data set unless you have GDDM-IMD.

Space for ADMX — GDDM-IMD export data set

The suggested size for this is 500 records. This value assumes an average size per exported map of 5 records of 400 bytes, and allows for a maximum of 100 exported maps at any one time.

User Map Specification Libraries — maps produced on GDDM-IMD

This data set is used to hold the maps that users produce with GDDM-IMD. Map Specification Libraries (MSLs) contain maps in an editable format. You can create one of these per user, per group of users, or per system, depending on your security and performance requirements. Do not create this data set unless you have GDDM-IMD.

Space for user MSLs — data sets for holding maps in editable format

Estimate how many records to allocate using the information on GDDM objects in the *GDDM System Customization and Administration* book.

Creating the VSAM data sets

Review the appropriate supplied JCL in the SADMSAM data set, customize it for your enterprise, and run it to create the required user MSL.

Tailor the SHAREOPTIONS parameters in the supplied job streams to the requirements of the users at your enterprise. For more information, see the *Access Method Services Reference* book that applies to your operating system.

Table 16. JCL for creating VSAM data sets

Cluster	Supplied JCL
ADMX	ADMUJC75
User MSL	ADMUJC74

Run ADMUJCIP if you want to edit the GDDM-IVU panel source using GDDM-IMD. If you do not have GDDM-IVU and GDDM-IMD, do not run this job. You must edit ADMUJCIP to match the national language panels you want to load, and run it once for each language you need. For more information, see the *GDDM Image View Utility* book.

CLISTs for TSO users

Migrating from earlier releases

Provide your users with CLISTs and job streams. You must modify any existing ones, changing the name qualifiers of the sample, symbol set, and mapgroup data sets to SADMSAM, SADMSYM, and SADMMAP respectively.

You may want to provide your users with suitable CLISTs and job streams so that they can more easily perform particular GDDM-related tasks. Some suggested CLISTs are given on the following pages, for such tasks as:

- Using the Image and Vector Symbol Editors (Figure 1 and Figure 2)
- Using the ICU (Figure 3)
- Using GDDM-IMD (Figure 4)
- Using GDDM-IVU (Figure 5)

GDDM provides other CLISTs to enable users to generate image files for page printers, and to help users transfer files to and from the auxiliary storage of a workstation. You may need to change these to suit the requirements of the users at your enterprise.

You may also like to create and catalog JCL for link-editing GDDM application programs. Examples are given in the *GDDM Base Application Programming Reference* book.

Allocating data sets before users start work

It is possible for users to invoke the symbol editors, the ICU, GDDM-IMD, GDDM-IVU, and GDDM-GKS without having first created data sets on which to store the symbols, charts, or generated mapgroups they produce. However, having done considerable work, users may then find they cannot save it. Providing CLISTs based on those shown on the following pages can help avoid this situation.

The data sets need to be created before you can allocate them in the CLISTs. GDDM supplies a sample CLIST ADMUCNEW in the SADMSAM data set to create the data sets.

More information about the data sets and the required data-set characteristics—the data set control blocks (DCBs)—can be found in “Data sets required for GDDM under TSO” on page 34.

If you are using an ISPF installation, you may prefer to set up suitable ISPF panels that give your users access to the various functions they need.

A copy of the suggested user CLISTs shown here is supplied in one member ADMUCRUN in the SADMSAM data set. To use this sample, copy it to a new data set and delete the lines you do not want. Then, modify the samples as required.

```
Suggested name: IMSYM

/* IMAGE SYMBOL EDITOR */
 ALLOC F(ADMDECK)  DA(ADMDECK)  MOD
 ALLOC F(ADMSYMBL) DA(ADMSYMBL) SHR REU
 CALL 'GDDM.SADMMOD(ADMISSE)'
 FREE F(ADMDECK)
 FREE F(ADMSYMBL)
```

The ADMSYMBL allocation must not have concatenated data sets
if you want to save symbol sets.
Concatenated partitioned data sets cannot be accessed read/write.

If you concatenate data sets of different block sizes,
the data set of the highest block size must appear first.

Figure 1. Suggested CLIST for using the Image Symbol Editor

```
Suggested name: VECSYM

/* VECTOR SYMBOL EDITOR      */
 ALLOC F(ADMDECK)  DA(ADMDECK)  MOD
 ALLOC F(ADMSYMBL) DA(ADMSYMBL) SHR REU
 CALL 'GDDM.SADMMOD(ADMVSSE)'
 FREE F(ADMDECK)
 FREE F(ADMSYMBL)
```

The ADMSYMBL allocation must not have concatenated data sets
if you want to save symbol sets.
Concatenated partitioned data sets cannot be accessed read/write.

If you concatenate data sets of different block sizes,
the data set of the highest block size must appear first.

Figure 2. Suggested CLIST for using the Vector Symbol Editor

```
Suggested name: CHART

/* INTERACTIVE CHART UTILITY */
 ALLOC F(ADMCDATA) DA(ADMCDATA) SHR REU
 ALLOC F(ADMCDEF)  DA(ADMCDEF)  SHR REU
 ALLOC F(ADMCFORM) DA(ADMCFORM) SHR REU
 ALLOC F(ADMGDF)   DA(ADMGDF)   SHR REU
 ALLOC F(ADMSYMBL) DA(ADMSYMBL 'GDDM.SADMSYM') SHR REU
 CALL 'GDDM.SADMMOD(ADMCHART)'
 FREE F(ADMCDATA)
 FREE F(ADMCDEF)
 FREE F(ADMCFORM)
 FREE F(ADMGDF)
 FREE F(ADMSYMBL)
```

The ADMSYMBL allocation must not have concatenated data sets
if you want to save symbol sets.
Concatenated partitioned data sets cannot be accessed read/write.

You are advised to make both the system and the user's own symbol
sets available to users of the ICU.
This gives the users a wide choice of type faces from the system sets,
and special symbols from their own.

If you concatenate data sets of different block sizes,
the data set of the highest block size must appear first.

Figure 3. Suggested CLIST for using the ICU

```
Suggested name: GDDMIMD

/* INTERACTIVE MAP DEFINITION */
 ALLOC F(ADMGGMAP) DA(ADMGGMAP) SHR REU
 ALLOC F(ADMGIMP)  DA('GDDM.SADMMAP') SHR REU
 ALLOC F(ADMGNADS) DA(ADMGNADS) SHR REU
 ALLOC F(ADMIFMT)  DA(ADMIFMT)   SHR REU
 ALLOC F(ADMLIST)  SYSOUT(A)
 ALLOC F(MSL1)     DA(YOUR-CLUSTER-NAME) SHR REU
 CALL 'GDDM.SADMMOD(ADMIMD)' 'MSL1'
 FREE F(ADMGGMAP)
 FREE F(ADMGIMP)
 FREE F(ADMGNADS)
 FREE F(ADMIFMT)
 FREE F(ADMLIST)
 FREE F(MSL1)
```

Figure 4. Suggested CLIST for using GDDM-IMD

```
Suggested name: GDDMIVU

/* IMAGE VIEW UTILITY       */
 ALLOC F(ADMGGMAP) DA(ADMGGMAP 'GDDM.SADMMAP') SHR REU
 ALLOC F(ADMIMG)   DA(ADMIMG)   SHR REU
 ALLOC F(ADMPROJ)  DA(ADMPROJ)  SHR REU
 CALL 'GDDM.SADMMOD(ADMIVU)'
 FREE F(ADMGGMAP)
 FREE F(ADMIMG)
 FREE F(ADMPROJ)
```

If you plan to use your own panels in addition to IBM-supplied panels, you must concatenate your own data set with the supplied one:

```
ALLOC F(ADMGGMAP) DA(ADMGGMAP 'GDDM.SADMMAP') SHR
```

If you want to load the supplied sample image ADMU5IMG and projection ADMU5PRJ, you must concatenate the SADMMAP and SADMIMG data sets to your own ADMIMG and ADMPROJ data set as follows:

```
ALLOC F(ADMIMG) DA(ADMIMG 'GDDM.SADMIMG') SHR REU
ALLOC F(ADMPROJ) DA(ADMPROJ 'GDDM.SADMMAP') SHR REU
```

The ADMIMG and ADMPROJ allocations must not have concatenated data sets if you want to save images or projections.
Concatenated partitioned data sets cannot be accessed read/write.

If you concatenate data sets of different block sizes, the data set of the highest block size must appear first.

Figure 5. Suggested CLIST for using GDDM-IVU

When you have finished

You should now have GDDM defined to the TSO subsystem.

Defining GDDM to CICS

This section describes the tasks you have to do to define GDDM to the CICS subsystem. The tasks are:

- Telling CICS where to find GDDM
- Managing the CICS VSAM data sets required by GDDM
- Updating CICS resource definitions (see “Updating CICS resource definitions” on page 45)

Telling CICS where to find GDDM

The GDDM code can be accessed by CICS using one of the following methods:

- Add the SADMMOD data set to the DFHRPL DD statement in your CICS startup JCL.
- Put the GDDM code into the LPA. If you do this, you must also change the PROGRAM entries in your CSD to tell CICS that you want to use the LPA to find GDDM. Two modules, ADMECOMT and ADMOPUT, are non-reentrant

and, therefore, are not LPA eligible; you must move these modules from SADMMOD into a STEPLIB load library.

Managing the CICS VSAM data sets required by GDDM

Migrating from earlier releases

You must reload the new GDDM objects into the ADMF and (if you are installing GDDM-IMD), the ADMGIMP data sets.

If you have already created the VSAM data sets, you do not need to recreate them. However, you may need to increase the size of your VSAM ADMF data set. If you do not need to increase its size, you can use the supplied JCL to overwrite your existing data set.

Note that, if you have modified the symbol sets from a previous release of GDDM, and you want to retain the modifications in the new release, you must take a backup copy and retain it outside the GDDM library because the new release overwrites the previous symbol sets.

If you need to enlarge your existing ADMF data set, you must:

1. Make a copy of the VSAM data set for ADMF from the previous release of GDDM.
2. Define the ADMF VSAM data set for the new release, using job stream ADMUJC02.
3. When the data set is allocated, copy all members from your old data set to your new data set. You can use a job based on job stream ADMUJCBS to do this, replacing the input label with a label corresponding to that of your backed-up data set. Find the job stream in the SADMSAM data set.
4. Load the GDDM objects for this release into your new ADMF VSAM data set, using the supplied JCL.

GDDM requires several VSAM data sets to be available before it can function under CICS. The data set names and contents are defined here.

ADMF — GDDM object data set

ADMF can have two uses. It holds the following GDDM-supplied data sets:

- Base symbol sets
- ADMPC objects
- Simplified Chinese symbol sets
- Japanese (Kanji) symbol sets
- GDDM-IVU panels and samples
- GDF objects

ADMF can also be used to hold the following user-created objects:

- User-created symbol sets
- Chart format, data, and definition data sets
- Image data and projection data sets
- GDF objects
- Pictures
- Generated map groups

If many of your users are going to create GDDM objects, you may prefer to create one or more private VSAM data sets to hold these objects. This is explained in "Private VSAM data sets" on page 42 below.

Private VSAM data sets

It is possible to customize GDDM so that it uses private VSAM data sets for the different types of user-created objects, using the OBJFILE external default, as described in the *GDDM System Customization and Administration* book.

Private data sets for GDDM objects may be needed for security reasons, either because their contents are confidential or to prevent access to the system data sets. Alternatively, if your enterprise is a heavy user of GDDM, using private data sets may help to improve system performance.

However, you should be aware that the GDDM symbol editors operate on the default VSAM data set for symbol sets, which is normally the VSAM data set ADMF into which you have installed the GDDM-supplied symbol sets. The symbol editors can use private data sets only if you write an access program that includes the GDDM ESLIB call to point to the private data sets. A simpler way of creating symbol sets in private data sets is to put them in the ADMF data set from the symbol editor, and then use the advanced directory function of the ICU to copy them to a private data set.

ADMGIMP — GDDM-IMD frames

This data set contains GDDM-IMD frames. It is of a fixed size. Do not create this data set unless you have GDDM-IMD.

ADMX — GDDM-IMD staging data set

This data set is necessary if you want to move GDDM-IMD maps to or from systems outside CICS. Do not create this data set unless you have GDDM-IMD.

User Map Specification Libraries — maps produced on GDDM-IMD

This data set is used to hold the maps that users produce with GDDM-IMD. Map Specification Libraries (MSLs) contain maps in an editable format. You can create one of these per user, per group of users, or per system, depending on your security and performance requirements. Do not create this data set unless you have GDDM-IMD.

DFHTEMP — CICS temporary storage data set

DFHTEMP is a CICS data set, and you probably have it already on your computer system. It is used by GDDM to retain print data streams until they can be printed. Because print data streams contain many bytes of data (typically 3KB to 5KB per GDDM page, but sometimes much more), you may need to enlarge DFHTEMP.

If you are installing GDDM-IVU for the first time, you may also need to enlarge DFHTEMP because of the larger size of many image data streams.

Calculating the CICS VSAM space requirements

The data sets ADMF, ADMX, user MSLs, and DFHTEMP, can be used to hold GDDM objects that are created by users of GDDM. Therefore, if you are going to use them, you have to make a judgment on the likely space requirements of the users at your enterprise before you allocate the data sets. The data set ADMGIMP contains only items supplied by GDDM, and is of a fixed size.

Space for ADMF — GDDM object data set

Table 17 shows how much space is needed for GDDM-supplied objects. If you are using ADMF to store user-supplied objects as well, you must estimate how many extra records to allocate using the information on GDDM objects in the *GDDM System Customization and Administration* book.

Table 17. Suggested space requirements for VSAM data sets

GDDM-supplied object	Component	Number of 400-byte records
GDDM symbol sets, Base	GDDM/MVS	7500
GDDM-PCLK ADMPC data sets	GDDM/MVS	2100
GDDM-OS/2 Link ADMPC data sets	GDDM/MVS	3900
GDDM symbol sets, Chinese	GDDM/MVS Simplified Chinese NLS	1500
GDDM symbol sets, Kanji	GDDM/MVS Japanese NLS	2200
GDDM symbol sets, high-quality Kanji	GDDM/MVS Japanese NLS	5700
Panels and samples	GDDM-IVU	600 per national language

Space for private VSAM data sets

Estimate how many records to allocate, based on how many data sets you are creating and how many people are to use them. For guidance, use the information on GDDM objects in the *GDDM System Customization and Administration* book.

Space for ADMGIMP — GDDM-IMD frames

This data set contains only GDDM-supplied data. The suggested value of 2100 records of 400 bytes is enough to hold the GDDM-IMD frames.

Space for ADMX — GDDM-IMD export data set

The suggested size for this is 500 records. This value assumes an average size per exported map of 5 records of 400 bytes, and allows for a maximum of 100 exported maps at any one time.

Space for user MSLs — data sets for holding maps in editable format

Estimate how many records to allocate, using the information on GDDM objects in the *GDDM System Customization and Administration* book.

Creating the VSAM data sets

Review the appropriate supplied JCL in the SADMSAM data set, customize it for your enterprise, and then run it to create the required VSAM data sets.

Tailor the SHAREOPTIONS parameters in the supplied job streams to the requirements of the users at your enterprise. For more information, see the *Access Method Services Reference* book that applies to your operating system.

Table 18. JCL for creating VSAM data sets

Cluster	Supplied JCL
ADMF	ADMUJC02
Private VSAM data sets	ADMUJC04
ADMGIMP	ADMUJC72
ADMX	ADMUJC75
User MSL	ADMUJC74

Loading the VSAM data sets

Table 19. JCL for loading VSAM data sets

Objects being loaded	Component	Data set	Supplied JCL
GDDM symbol sets	GDDM/MVS	ADMF	ADMUJCBS
GDDM-PCLK ADMPC data sets	GDDM/MVS	ADMF	ADMUJCPC
GDDM-OS/2 Link ADMPC data sets	GDDM/MVS	ADMF	ADMUJCOS
Japanese symbols	GDDM/MVS Japanese NLS	ADMF	ADMUJCJS
Chinese symbols	GDDM/MVS Simplified Chinese NLS	ADMF	ADMUJCCS
Maps and tutorials	GDDM-IMD	ADMGIMP	ADMUJCIM
Panels and samples	GDDM-IVU	ADMF	ADMUJCIV
Panel source	GDDM-IVU	ADMX	ADMUJCIP

Review the appropriate supplied JCL in the SADMSAM data set, customize it for your enterprise, and then run it to load the GDDM-supplied objects into your data sets.

If you want to install multiple-language NLS for GDDM-IVU, you must edit ADMUJCIV to include all of the languages, or edit it and run it again, once for each language.

You need only run ADMUJCIP if you want to edit the GDDM-IVU panel source using GDDM-IMD. If you do not have GDDM-IVU and GDDM-IMD, do not run this job. You must edit ADMUJCIP to match the national language panels you want to load, and run it once for each language you need. For more information, see the *GDDM-IVU User's Guide*.

If you have decided to enlarge or create DFHTEMP (see “DFHTEMP — CICS temporary storage data set” on page 42), do so now. The way to do this is described in the *CICS/ESA System Definition Guide*.

Updating CICS resource definitions

Migrating from earlier releases

If you are migrating from an earlier level of GDDM/MVS, you need to update your CICS CSD GDDM resource definitions.

If you are going to use GDDM on any new devices, you need to review the entries in the CICS terminal control table (see “Reviewing the CICS terminal definitions” on page 48).

The CICS subsystem needs to know about its *resources*. These include software resources such as programs and data, and hardware resources such as terminals, printers, and telecommunication links. This information, collectively called *resource definitions*, is held internally by CICS.

Updating the CICS CSD

The CICS CSD must contain FILE, TRANSACTION, and PROGRAM entries for GDDM. Do the following to add these entries to your CSD:

1. Copy the menber ADMUBCSD from the SADMSAM data set.
2. Review the CICS FILE definitions in ADMUBCSD. ADMF is defined and the entry does not need any further customization. Example definitions are also given for the ADMGIMP, ADMX, GDDM private VSAM files, and GDDM-IMD user-MSLs. These entries are supplied commented out so the comment character must be removed if you need to specify these FILE definitions. You need to copy the GDDM private VSAM file or GDDM-IMD user-MSL definitions if you have more than one of these data sets.

   Include corresponding DD statements for the VSAM data sets to be used in the CICS startup job stream. Remember to include DD statements for any private VSAM data sets you have defined. For example:

   ```
   //ADMF     DD DSN=VSAM-data-set-name,DISP=OLD
   //msl-name DD DSN=VSAM-data-set-name,DISP=OLD
   //ADMGIMP  DD DSN=VSAM-data-set-name,DISP=OLD
   //ADMX     DD DSN=VSAM-data-set-name,DISP=OLD
   ```

   You must **not** use RECOVERY=ALL or RECOVERY=BACKOUTONLY with these data sets because they may cause problems with generic key processing.
3. Review the CICS TRANSACTION definitions in ADMUBCSD. (The following table shows the transactions that are defined.) Definitions are given for all the GDDM/MVS transactions, as well as the associated products such as GDDM-PGF.

   GDDM supplies a number of sample programs written in PL/1, COBOL, and C/370. Some example TRANSACTION entries are given for the GDDM sample programs; these are commented out. If you want to use any of the sample programs (such as ADMUSP4), locate these entries by searching for ‘ADMUSB1’. Repeat and change these entries to match the GDDM sample

programs you want to use. Ensure that you set the LANGUAGE parameter correctly and remove the comment character.

Table 20. Transaction definitions

Transaction names	Component	Description
ADMA	GDDM/MVS	Installation verification.
ADMI	GDDM/MVS	Image Symbol Editor.
ADMP	GDDM/MVS	Print utility.
ADM4	GDDM/MVS	Composite Document Print Utility.
GQFI	GDDM/MVS	OS/2 download.
ADMC	GDDM-PGF	ICU.
ADMV	GDDM-PGF	Vector Symbol Editor.
ADMM	GDDM-IMD	GDDM-IMD.
ADMU	GDDM-IVU	GDDM-IVU.

Notes:

a. The TRANSACTION type entries in your job stream are coded with SPURGE(NO). The SPURGE definition decides whether the transaction is eligible for purging by CICS if main storage is constrained. If your GDDM transactions are not considered to be important, change these definitions to SPURGE(YES).

b. SCRNSIZE (ALTERNATE) must be specified in the PROFILE definition for all GDDM transactions.

4. Review the CICS PROGRAM definitions in ADMUBCSD. ADMUBCSD contains entries for all of GDDM/MVS, including GDDM-OS/2 Link and the sample programs, plus GDDM-PGF, GDDM-IMD, GDDM-IVU, and GDDM/MVS US English National Language support.

   If you want to use other national languages, you need to add the definitions in members ADMUH*CS from the SADMSAM data set, where * is the national language letter defined in Table 24 on page 88.

   Notes:

   a. If you have any application programs that call GDDM and that have the attribute RESIDENT(YES), you can make the parts of GDDM used by these programs resident as well.

   b. If you want to use GDDM from the LPA, you must change the PROGRAM definitions from USELPACOPY(NO) to USELPACOPY(YES).

5. Use the CICS RDO offline facility or the macro utility DFHCSDUP to load your GDDM resource definitions into the CSD.

6. Add your CSD definitions to your CSD list. The GROUP names you need to add are:

Table 21. GROUP names for GDDM products	
GROUP name	**Product**
GDDMBASE	GDDM/MVS + NLS
GDDMPGF	GDDM-PGF
GDDMIMD	GDDM-IMD
GDDMIVU	GDDM-IVU
GDDMGKS	GDDM-GKS

Updating the CICS destination control table (DFHDCT)

The following types of output need to be included in the destination control table (DCT), which must also define both their transient destination and their extra-partition destination:

ADML Error log records
ADMS System printer output
ADMD Object decks that can be produced by the symbol editors
ADMT Trace output
ADMG Application Data Structures (ADSs) generated by GDDM-IMD.

Note: With the exception of ADML, the transient destinations all have external names that you can change in the external defaults module, if the defaults are unsuitable for your enterprise. For more information on changing the names in the external default module, see the *GDDM System Customization and Administration* book.

1. Decide which DCT definitions you are going to need. If you do not intend to install GDDM-IMD, you do not need ADMG. If you can be sure that you are not going generate any of the other types of output, you can omit the definition.
2. Decide to which destination you are going to route the output. Printer output is usually routed to a system printer using SYSOUT. Symbol object decks and Application Data Structures are normally routed to a data set. Trace output and error log records can be routed either to a system printer using SYSOUT, or to a data set, depending on the requirements of the users at your enterprise. If you are routing the output to a data set, you must define it first. For information about the DCB characteristics, see the sample DCT entries in the supplied sample ADMUBDCT in the SADMSAM data set.
3. Copy the supplied sample ADMUBDCT from the SADMSAM data set to your own data set, and edit it to suit the requirements of your users. Then add these entries to your current DCT entries. ADMUBDCT is listed in Figure 6 on page 48.
4. Reassemble the DCT and link-edit it. The *CICS/ESA System Definition Guide* describes how to assemble and link-edit the table. A job-stream for this purpose, DFHAUPLE, is supplied with CICS.
5. Include DD statements in the CICS startup job stream for the entries you have defined. These must match the DSCNAME in the DCT. For example:

```
//ADMLDDX DD SYSOUT=A
//ADMSDDX DD SYSOUT=A
//ADMTDDX DD DSN=your dataset1, DISP=SHR
//ADMDDDX DD SYSOUT=A
//ADMGDDX DD DSN=your dataset2, DISP=SHR
```

```
* ERROR LOG RECORDS
         DFHDCT TYPE=SDSCI,DSCNAME=ADMLDDX,                          C
               RECFORM=VARUNB,                                       C
               RECSIZE=120,                                          C
               BLKSIZE=124,                                          C
               TYPEFLE=OUTPUT
* SYSTEM PRINTER OUTPUT
         DFHDCT TYPE=SDSCI,DSCNAME=ADMSDDX,                          C
               RECFORM=VARBLKA, (OR AS DEFINED IF DIFFERENT)         C
               RECSIZE=142, (OR AS DEFINED IF DIFFERENT)             C
               BLKSIZE=146, (OR AS DEFINED IF DIFFERENT)             C
               TYPEFLE=OUTPUT
* SYMBOL EDITOR OBJECT DECK OUTPUT
         DFHDCT TYPE=SDSCI,DSCNAME=ADMDDDX,                          C
               RECFORM=FIXUNB,                                       C
               RECSIZE=80,                                           C
               BLKSIZE=80,                                           C
               TYPEFLE=OUTPUT
* TRACE OUTPUT
         DFHDCT TYPE=SDSCI,DSCNAME=ADMTDDX,                          C
               RECFORM=VARBLKA,                                      C
               RECSIZE=137,                                          C
               BLKSIZE=141,                                          C
               TYPEFLE=OUTPUT
* GDDM-IMD APPLICATION DATA STRUCTURE (ADS) OUTPUT
         DFHDCT TYPE=SDSCI,DSCNAME=ADMGDDX,                          C
               RECFORM=FIXUNB,                                       C
               RECSIZE=80,                                           C
               BLKSIZE=80,                                           C
               TYPEFLE=OUTPUT
* ERROR LOG RECORDS
         DFHDCT TYPE=EXTRA,DESTID=ADML,                              C
               DSCNAME=ADMLDDX
* SYSTEM PRINTER OUTPUT
         DFHDCT TYPE=EXTRA,DESTID=ADMS,                              C
               DSCNAME=ADMSDDX
* SYMBOL EDITOR OBJECT DECK OUTPUT
         DFHDCT TYPE=EXTRA,DESTID=ADMD,                              C
               DSCNAME=ADMDDDX
* TRACE OUTPUT
         DFHDCT TYPE=EXTRA,DESTID=ADMT,                              C
               DSCNAME=ADMTDDX
* GDDM-IMD APPLICATION DATA STRUCTURE (ADS) OUTPUT
         DFHDCT TYPE=EXTRA,DESTID=ADMG,                              C
               DSCNAME=ADMGDDX
```

Figure 6. Listing of supplied sample ADMUBDCT

Reviewing the CICS terminal definitions

The terminals you are going to use to run GDDM must be defined to CICS in your CSD. GDDM may need some attributes to be specified that are not set by default. Here is a list of things to check:

- GDDM needs particular attributes set for each terminal that is going to display graphics. These can be set automatically using the QUERY function in the CSD entry for the terminal. For more information about querying the terminals,

see the *CICS Resource Definition* books. If you are not querying the terminals, check the following attributes:

- EXTENDEDDS(YES) indicates that the device supports extended 3270 datastream, which includes a number of features. Some of these features are described under the information on queriable terminals and printers in the *GDDM System Customization and Administration* book. If this option is set in the terminal definition, GDDM assumes that it can query the device.
- All queriable printers must have ATI(YES) and TTI(YES) in their TYPETERM definition. (A queriable printer is a printer that can accept a Read Partition (QUERY) structured field. See the information on queriable terminals and printers in the *GDDM System Customization and Administration* book. These values are needed because each time GDDM accesses a new printer or terminal, it issues a “query device”.
- IPDS printers must have DEVICE=SCSPRINT in their TYPETERM definition.
- VTAM SNA terminals and printers must have a SENDSIZE value of at least 1536 in their TYPETERM definition.

When you have finished

You should now have GDDM defined to the CICS subsystem.

Defining GDDM to IMS

This section describes the tasks you have to do to define GDDM to the IMS subsystem. The tasks are:

- Accessing the GDDM utilities
- Allocating and cataloging IMS system data sets
- Checking and updating the IMS system-definition values
- Preparing the operating system for IMS
- Generating the IMS database description
- Generating IMS program specification blocks
- Generating IMS application control blocks
- Loading the GDDM IMS databases
- Defining the IMS system
- Establishing IMS system security

Accessing the GDDM utilities

Migrating from earlier releases

If you are upgrading from GDDM Version 1 Release 4 or later, you must do this task to reestablish any defaults you need.

Version 2 Release 2 of GDDM introduced support for *country-extended code pages*, which associate GDDM objects (symbol sets, charts, and so on) with the environment in which they are used. An object that is tagged with a code page does not need any special action to make it usable in a different linguistic environment, if GDDM is suitably set up. For more information, see the *GDDM System Customization and Administration* book. Consider the advantages of using this function at your enterprise if you have not yet done so.

You should be aware of the method of operation of the GDDM utilities. Read the following and take any action you think necessary.

Under IMS, the GDDM interactive utilities are run under the control of a single transaction that emulates the expected environment. In the following notes, the transaction code for the utility is assumed to be ADM, but this may have been changed by the installation.

- The transaction is a “wait for input” conversational transaction. While the transaction is scheduled, the message-processing region is unable to service other transactions. This normally implies dedicating a message-processing region to the transaction.
- The transaction can support only a predefined number of concurrent transactions. Any attempt to start a new session with a utility that would cause the limit to be exceeded is rejected with GDDM message ADM0772 being displayed.

  The number of concurrent transactions allowed may be changed by modifying the external defaults module.
- The transaction is unable to continue conversations if, for any reason, it is rescheduled during the lifetime of a conversation. Such conversations are terminated with GDDM message ADM0774 being displayed.
- A particular scheduling of the transaction usually ends when it has no record of any existing conversations. Because it is possible for a conversation to be terminated without the transaction being aware of the fact (for example, because of some error situations), the transaction may not be completed even though the user has terminated the conversation. In such an instance, the user should enter the request:

  ```
  ADM EXIT
  ```

  which causes the utility to note that all conversations against the LTERM from which the request originates have been terminated.
- To force a return to the region controller by the transaction, irrespective of the current state of any active conversations, the request:

  ```
  ADM SHUTDOWN
  ```

  can be entered from an authorized terminal. By default, this authorized terminal has an LTERM name of MASTER.

The keywords EXIT and SHUTDOWN, and the LTERM name of the terminal authorized to issue the latter request are GDDM defaults and can be changed in the external defaults module.

- If, during a session with a utility, the current screen format is destroyed (for example, by a high priority error message), it can be restored by entering two blank characters as the next input message.
- On some terminals, IMS reserves PF12 for use as a print request key and does not pass this as a valid interrupt to the utility transaction. If the terminal has 24 rather than 12 PF keys, the use of PF key 12 may be avoided because PF key 24 usually has the same function.

  If only 12 PF keys are available, the IMS system definition for a terminal should specify NOCOPY if the GDDM utilities are to be accessed from that terminal.

Allocating and cataloging IMS system data sets

Migrating from earlier releases

You do not need to do this task.

1. Allocate an IMSVS.SPA data set if not already in use and if the GDDM interactive utility is defined to operate with a DASD SPA.
2. Reappraise the space requirements for the data sets IMSVS.QBLKS, IMSVS.SHMSG, and IMSVS.LGMSG.

   The output from a GDDM transaction is a data stream that is to be sent to the device. This output is held in a message queue before being sent to the device. The size of the data stream depends on the complexity of the picture and the device to which it is to be sent. Each picture can vary between 3KB and 30KB or more, depending on its complexity. For display terminals, the message can contain only one picture. For printers, any number of pictures may be placed in a single message.
3. Allocate space for SYSOUT data sets if output from GDDM is to be directed to a line printer.
4. Allocate a data set to contain the TEXT decks produced from the interactive symbol editors if this is required. The data set should have these characteristics:

   ```
   LRECL=80
   BLKSIZE=800
   RECFM=FB
   ```

Checking and updating the IMS system-definition values

Migrating from earlier releases

You need to add statements for a secondary index to the object database.

1. Include the SPAREA macro in the system definition if the interactive GDDM utilities are to be used. In Figure 7 on page 53, the example TRANSACT statement for the interactive utility program, ADM, gives the required size.
2. Large amounts of communications I/O pool (CIOP) space may be used if non-SNA devices are defined within the network. Therefore, you may need to review and, if necessary, change the allocation of space for and within the CIOP. See the description of the COMM and BUFPOOLS macros in the *IMS/VS Installation Guide*.
3. Specify the option TRANCMD=YES on the SECURITY macro because some of the supplied GDDM transactions use the /BROADCAST function to notify errors when no other means is available.
4. Include DATABASE macros for the GDDM databases. Under IMS, GDDM uses two types of database:
   - The first contains information linking LTERM names to device characteristics and is called the *system-definition database*.
   - The second contains “objects” accessed by GDDM (for example, symbol sets and chart descriptors), and is called the *object database*.

The object database requires a secondary index to be set up, which enables GDDM objects to be managed from within the ICU.

These databases are required when installing GDDM:

```
         DATABASE DBD=ADMSYSDF
*
         DATABASE DBD=ADMOBJ1
         DATABASE DBD=ADMIDX1
```

It is usually necessary only to have the system-definition database. The number of GDDM object databases depends upon the requirements of the users at your enterprise. By default, GDDM assumes that ADMOBJ1 is the only database and places all GDDM objects in it. ADMIDX1 is the default name of the secondary index for the GDDM object database.

Include DD statements for these databases in the IMS control region JCL.

5. Include the APPLCTN and TRANSACT macros for the various GDDM utilities and the GDDM sample programs for IMS.

   The APPLCTN and TRANSACT macros are supplied as a member, ADMUJI01 (GDDM/MVS) and ADMUJI51 (GDDM-PGF) in the SADMSAM data set and are shown in Figure 7 on page 53.

```
*
* GDDM PRINT UTILITY
* (Note that PGMTYPE=BATCH is allowed)
*
          APPLCTN PSB=ADMOPUI,PGMTYPE=TP
          TRANSACT CODE=ADMPRINT,
                   MSGTYPE=(MULTSEG,NONRESPONSE,1),
                   INQUIRY=(YES,RECOVER),
                   MODE=SNGL,
                   EDIT=(ULC)

*
* GDDM INTERACTIVE UTILITY
*
          APPLCTN PSB=ADMUTIL,PGMTYPE=TP
          TRANSACT WFI,
                   CODE=ADM,
                   MSGTYPE=(MULTSEG,RESPONSE,1),
                   INQUIRY=(NO,RECOVER),
                   MODE=SNGL,
                   EDIT=(ULC),
                   SPA=(50,CORE,FIXED)

*
* GDDM UTILITY TO MAINTAIN SYSTEM DEFINITION DATABASE
* (Note that this can be omitted if it is only run as a BPP)
*
          APPLCTN PSB=ADMFSDU,PGMTYPE=BATCH

*
* GDDM UTILITY TO MAINTAIN OBJECT DATABASE
* (Note that this can be omitted if it is only run as a BPP)
*
          APPLCTN PSB=ADMFOU,PGMTYPE=BATCH

*
* GDDM SAMPLE PROGRAM: ADMUSP1
*
          APPLCTN PSB=ADMUSP1,PGMTYPE=TP
          TRANSACT CODE=ADMUSP1,
          MSGTYPE=(SNGLSEG,NONRESPONSE,1),
                   INQUIRY=(NO,RECOVER),
                   MODE=SNGL,
                   EDIT=(ULC)
*
* GDDM SAMPLE PROGRAM: ADMUSP2
*
          APPLCTN PSB=ADMUSP2,PGMTYPE=TP
          TRANSACT CODE=ADMUSP2,
          MSGTYPE=(SNGLSEG,NONRESPONSE,1),
                   INQUIRY=(YES,RECOVER),
                   MODE=SNGL,
                   EDIT=(ULC)
```

Figure 7. Supplied macro definitions in member ADMUJI01 for utilities and sample programs

```
*
* GDDM-PGF SAMPLE PROGRAM: ADMUSP5
*
          APPLCTN PSB=ADMUSP5,PGMTYPE=TP
          TRANSACT CODE=ADMUSP5,
                   MSGTYPE=(SNGLSEG,NONRESPONSE,1),
                   INQUIRY=(YES,RECOVER),
                   MODE=SNGL,
                   EDIT=(ULC)
```

Figure 8. Supplied macro definition in member ADMUJI51 for utilities and sample programs

6. Review the Data Communication Macro Statements:

   - For all terminals from which the interactive utilities may be invoked, specify OPTIONS=TRANRESP.

     Also consider specifying OPTIONS=LOCK, to prevent IMS unlocking keyboards prematurely during conversations, and to prevent the terminal errors that might otherwise arise. (The LOCK option normally takes effect only when one of the GDDM interactive utilities is running on a terminal.)

   - For remote BTAM-attached displays or printers that are to be used by GDDM, specify OPTIONS=XPAR to request *data transparency* (which is needed to handle the program-symbol definitions sent in the data streams).

   - For advanced function display devices, which are neither defined as SLUTYPE2, nor as remote BTAM-attached devices for which the XPAR option has been specified, specify a value of OUTBUF equal to the length of the largest output message that will be sent to them from a GDDM application.

     This size depends on the mode of attachment of the terminal (remote or local), and the complexity of the picture. Simple pictures may be expected to use about 3 to 5KB, whereas complex pictures typically use about 15 to 32KB. Data streams generated for controllers that support the compressed data format for symbol definitions (for more information, see the *IBM 3174 Data Stream Programmer's Reference* or the *IBM 3274 Control Unit Description and Programmer's Guide*) are usually shorter by a factor of between 2 and 3 than similar data streams using uncompressed formats.

   - For remote BTAM-attached displays, specify an OUTBUF value of 3KB or less.

   - For BTAM-attached or non-SNA advanced function printers, specify an OUTBUF parameter greater than or equal to the transmission buffer size (IOBFSZ) specified in the GDDM external defaults module. The choice of value for IOBFSZ and OUTBUF are described in the *GDDM System Customization and Administration* book.

7. If you want to make use of the TEXT output option of the Image Symbol Editor and the Vector Symbol Editor, define the destination ADMDECK to which the fixed 80-byte TXT records may be output. The name, ADMDECK, can be specified in the external defaults module to suit any existing standards at your enterprise.

   Two suggested forms of definition are shown below. The first directs output to a punch, the second to a named data set. You should choose a value for the ADDR parameter of the LINE macro.

- Output to punch:

```
LINEGRP  DDNAME=GDDMPUN,UNITYPE=PUNCH
LINE     ADDR=xxx
TERMINAL MSGDEL=SYSINFO
NAME     ADMDECK
```

- Output to data set:

```
LINEGRP  DDNAME=GDDMDISK,UNITYPE=DISK
LINE     ADDR=xxx,BUFSIZE=820
TERMINAL MSGDEL=SYSINFO
NAME     ADMDECK
```

Note: The BUFSIZE parameter must be at least 20 bytes larger than the BLKSIZE parameter of the data set to which the output is to be sent.

Preparing the operating system for IMS

Migrating from earlier releases

You do not need to do this task.

1. Consider the size of the dependent regions in which GDDM applications are to be run. The online storage requirements of such applications are given in the *GDDM System Customization and Administration* book.
2. Consider the possibility of making frequently used GDDM modules resident.
3. Include DD statements for the GDDM databases in the IMS startup procedure. Base them on those shown below.

```
//ADMSYSDF   DD DSN=ADMSYSDF,DISP=SHR
//ADMOBJ1    DD DSN=ADMOBJ1,DISP=SHR
//ADMIDX1    DD DSN=ADMIDX1,DISP=SHR
```

4. If any applications make use of the trace files produced by the GDDM FSTRCE functions, add DD statements for these files. The DD statements should be added to the procedures for executing online message or batch-message processing regions and for executing offline batch-processing regions. Choose the data set type and DCB characteristics for the DD statements from the values in Table 22.

Table 22. GDDM trace data set characteristics for IMS

Type of data	GDDM default file name	Data set type	RECFM format	LRECL	BLKSIZE
Trace records	ADMTRACE	Sequential data sets or SYSOUT classes	VA	125 or greater	LRECL
			VBA	125 or greater	LRECL+4 or greater

5. If punch or disk destinations were defined during IMS system definition for TEXT output from the interactive utilities, change the procedures containing the JCL to execute the IMS control region as follows:

- For punch output

```
//GDDMPUN DD SYSOUT=B
```

The output will be punched when the IMS control region terminates.

- For disk output

  ```
  //GDDMDISK DD DSN=xxxxxx,DISP=yyy
  ```

  where the DSN parameter is the data-set name of the data set to contain the output TEXT records. DISP may be OLD (in which case no other application may access the data set while the control region is executing) or SHR.

Generating the IMS database description

Migrating from earlier releases

You do not need to do this task.

Database descriptions are required for the system-definition database, for an object database, and for an index database.

A suggested form for DBDGEN statements is supplied as a member ADMUJI02, in the SADMSAM data set, and is shown in Figure 9. Customize the statements as necessary and use them to generate the database definitions.

```
DBDGEN statements for system-definition database

        DBD     NAME=ADMSYSDF,ACCESS=HDAM,RMNAME=(DFSHDC40,7,100)
        DATASET DD1=ADMSYSDF,DEVICE=3380,SIZE=512
        SEGM    NAME=ADMSDSGM,PARENT=0,BYTES=16
        FIELD   NAME=(ADMSDKEY,SEQ,U),START=1,BYTES=8,TYPE=C
        DBDGEN
        FINISH
        END

DBDGEN statements for object database

        DBD     NAME=ADMOBJ1,ACCESS=HDAM,RMNAME=(DFSHDC40,4,100)
        DATASET DD1=ADMOBJ1,DEVICE=3380,SIZE=8192
        SEGM    NAME=ADMOBROO,PARENT=0,BYTES=400
        FIELD   NAME=(ADMOBRKY,SEQ,U),START=1,BYTES=16,TYPE=C
        LCHILD  NAME=(ADMIXSEG,ADMIDX1),PTR=INDX
        XDFLD   NAME=ADMIXFLD,SRCH=ADMOBRKY
        SEGM    NAME=ADMOBDEP,PARENT=ADMOBROO,BYTES=400
        FIELD   NAME=(ADMOBDKY,SEQ,U),START=17,BYTES=4,TYPE=C
        DBDGEN
        FINISH
        END

DBDGEN statements for index database

        DBD     NAME=ADMIDX1,ACCESS=INDEX
        DATASET DD1=ADMIDX1,DEVICE=3380,SIZE=8192
        SEGM    NAME=ADMIXSEG,PARENT=0,BYTES=16
        FIELD   NAME=(ADMFLD,SEQ,U),START=1,BYTES=16,TYPE=C
        LCHILD  NAME=(ADMOBROO,ADMOBJ1),INDEX=ADMIXFLD,PTR=SNGL
        DBDGEN
        FINISH
        END
```

Figure 9. Supplied DBDGEN statements in member ADMUJI02

Generating IMS program specification blocks

Migrating from earlier releases

You need to do this task.

Sample program specification block (PSB) statements are provided for the GDDM utility programs and for the sample programs.

The job streams are supplied as members ADMUJI03 and ADMUJI52, in the SADMSAM data set.

1. Find the PSB statements, ADMUJI03, in the SADMSAM data set, change if necessary, and run the job.
2. If you have installed GDDM-PGF, repeat the operation using the PSB statements in ADMUJI52.

Generating IMS application control blocks

Migrating from earlier releases

You do not need to do this task.

Application control blocks (ACBs) must be constructed for the GDDM utility programs and sample programs.

The job streams are supplied as members ADMUJI04 and ADMUJI53, in the SADMSAM data set.

1. Find the ACBGEN statements (ADMUJI04) in the SADMSAM data set, change them if necessary, and run the job.
2. If you have installed GDDM-PGF, repeat the operation for the PGF ACBGEN statements in ADMUJI53, change them if necessary, and run the job.

Loading the GDDM IMS databases

Migrating from earlier releases

You do not need to set up the system-definition database if it already exists.

You should read in the new symbol sets, even if you already have a GDDM object database.

Review the size of your GDDM object database and index database.

Details of procedures to reorganize databases are given in the *IMS Utilities Reference Manual*. The necessary control information can be found in the DBDGEN for ADMIDX1.

How to update the GDDM object database:

1. Create a VSAM file for the secondary index using the job stream ADMUJI08 supplied in the SADMSAM data set.
2. Ensure that the index database has been added to the IMS system definition as detailed in "Checking and updating the IMS system-definition values" on page 51.
3. Run the IMS utilities and ensure that they perform correctly.
4. Find the job stream ADMUJI07 in the SADMSAM data set and change the PARM value in the EXEC statement to:

   ```
   'DLI,ADMFOU,ADMFOU,,,,,,,,,,N,N'
   ```

   Make any other necessary changes. You may need to increase the region size.
5. Run the job stream and ensure that it performs correctly.

GDDM requires two IMS databases (one of which includes an index database) before it can be used. These must be created as VSAM data sets and loaded with the required information. For each database, GDDM supplies a utility that is used for loading it. The databases and their associated utilities are:

- The system-definition database, ADMSYSDF, and the system-definition utility, ADMFSDU.
- The GDDM object database, ADMOBJ1, and the GDDM object import/export utility, ADMFOU.

A full description of the databases and the respective utilities is given in the *GDDM System Customization and Administration* book. This section gives you enough information to perform the installation steps using the utilities.

System-definition database

GDDM uses the system-definition database to determine the type of device in use, and in so doing determines the format of the data stream that is sent to that device. The system-definition database contains the relationship between the LTERMs defined for the system and the characteristics of the devices to which they are normally assigned.

You must take great care to associate the correct device definition with an LTERM. If the wrong device characteristics are used, GDDM will almost certainly not be able to show graphics and even may not show alphanumerics.

GDDM object database and index database

The GDDM object database contains the sample objects distributed with GDDM. These are symbol sets that are essential to the running of GDDM.

The installation process creates a partitioned data set containing the objects. This data set is used as the input to the application program ADMFOU, which moves the objects into the database. GDDM cannot work without this database.

Instructions for loading the databases

This section describes how to set up and load the GDDM system definition and GDDM object databases.

Setting up the system-definition database

1. Set up the system-definition database using a job based on the job stream ADMUJI09 in the SADMSAM data set.
2. Create a file for input to the system-definition utility as follows:
   a. Create a list of all the LTERMs on which you intend to use GDDM.
   b. For each LTERM, find a device token that has the correct device characteristics associated with it.

      GDDM supplies tables of device tokens in the modules ADMLSYS1, ADMLSYS3, ADMLSYS4, and ADMLSYSA. The device tokens supplied are shown in the *GDDM System Customization and Administration* book.
   c. If you have a device for which there is no suitable device token, you must create one as described in the *GDDM System Customization and Administration* book.
   d. Create a SYSIN file (LRECL 80, RECFM F) of the format:

      ```
      LTERM     DEVICE-TOKEN      optional comments
      ```

      For example, if you had a terminal and a printer to include in the data, the SYSIN entries would look like this:

      ```
      TERM1     L3472G      3472 graphics terminal
      PRT1      L87S9       3287 printer
      ```
3. Find the supplied job stream ADMUJI06 in the SADMSAM data set, and make any changes necessary. Note particularly the PARM values. If the database is being created for the first time, you should use the PSB ADMFSDUL. If the database already exists and is being updated, you should use the PSB ADMFSDU.
4. Place the input file as the SYSIN file to ADMUJI06.
5. Run the job stream ADMUJI06 and ensure that it performs correctly.

The GDDM system-definition database has now been set up.

Setting up the GDDM object database: The GDDM object database is used to hold GDDM objects such as:

- Symbol sets

- Chart format files
- Chart data files
- Saved pictures (ADMSAVE files)
- Graphics Data Format (GDF) files
- Maps imported from TSO.

A number of symbol sets are supplied with GDDM. These are essential to its operation and are used for such things as shading and coloring areas in the graphics field, and writing text in the graphics field. Kanji symbol sets are supplied with GDDM/MVS Japanese NLS, and you can exclude them if they are not needed.

Set up a new database as follows:

1. Create a VSAM file for the database and for its secondary index using the supplied job stream ADMUJI08.

   This database needs to contain all GDDM objects that may be created by GDDM users, as well as those supplied with GDDM. Therefore, you should think carefully about the amount of space required, to ensure that there will enough. See the *GDDM System Customization and Administration* book.

2. Find the supplied job stream ADMUJI07, which reads in the sample symbol sets.

3. Make any changes necessary.

   Remove the IMPORT statements for the Kanji symbol sets if they are not required. (The Kanji symbol sets are the ones with names that begin ADMIK, ADMVK, and ADMVQ).

   The job stream ADMUJI07 is the user-provided load program to be run with the IMS utilities to load the database, as detailed in the *IMS Utilities Reference* manual. If you are not familiar with IMS, you should also read the *IMS System Programming Reference Manual* for a description of the interfaces to IMS utilities.

4. Run the IMS utilities and the job stream, and ensure that they perform correctly.

   The new database has now been created.

5. Ensure that the index database has been added to the IMS system definition as detailed in “Checking and updating the IMS system-definition values” on page 51.

Defining the IMS system

Migrating from earlier releases

You need to define the IMS system to incorporate the index database.

Before GDDM can be tested, and before the system security values can be changed (see below), the IMS system must be defined. This should be a nucleus definition, but you may prefer to do it in some other manner. More information can be found in the *IMS Installation Guide*.

Establishing IMS system security

Migrating from earlier releases

You do not need to do this task.

Security under IMS can be handled either internally via the IMS Security Maintenance Utility (SMU) and the BROADCAST command, or externally via RACF.

If GDDM transactions or utilities detect errors, and the error message cannot be sent over a normal message queue, they try to inform the user by means of a BROADCAST command. However, these transactions and utilities need authorization to perform the BROADCAST command.

You should consider authorizing other transactions that use the GDDM default error-handling mechanism.

The transactions that need to be authorized either via the BROADCAST command or via RACF are:

ADM
ADMPRINT
ADMUSP1
ADMUSP2
ADMUSP5

When you have finished

You should now have finished defining GDDM to the IMS subsystem.

Making GDDM-REXX available

The GDDM-REXX code is shipped with GDDM/MVS Base.

If you ordered GDDM/MVS as part of OS/390, you should follow the instructions in the OS/390 documentation to enable GDDM-REXX.

If you ordered GDDM/MVS as a standalone product, you should follow the instructions for enabling GDDM-REXX in the document *How to activate GDDM-REXX*, LY33-6080, that is shipped when you order the GDDM-REXX/MVS feature.

Reblocking any existing objects

You need only do this task if you have created any user object data sets for levels of GDDM before Version 3 Release 1 with BLKSIZE=400. For a list of user objects, see the information on GDDM objects in the *GDDM System Customization and Administration* book.

To reblock a data set, do the following:

1. Determine the block size for your new GDDM data sets SADMPCF, SADMMAP, or SADMSYM. The block size of these should match the

BLKFB400 parameter chosen during installation, although if you have SMS installed on your system, it may have changed this.

2. Create new user object data sets with these DCB characteristics:

   ```
   DCB = (LRECL=400, BLKSIZE=n, RECFM=FB)
   ```

   where n is the number you found in Step 1 on page 61.

3. Copy this JCL and edit it to match the requirements of your users:

   ```
   //COPYOBJ  JOB
   //IEBCOPY  EXEC    PGM=IEBCOPY,REGION=2M
   //SYSPRINT DD      SYSOUT=*
   //SYSUT3   DD      UNIT=SYSDA,DISP=NEW,SPACE=(CYL,(1,1))
   //SYSUT4   DD      UNIT=SYSDA,DISP=NEW,SPACE=(CYL,(1,1))
   //INPUT    DD      DISP=SHR,DSN=your-old-dataset,
   //  DCB=(LRECL=400,BLKSIZE=400,RECFM=FB)
   //OUTPUT   DD      DISP=SHR,DSN=your-new-dataset
   //SYSIN    DD      *
     COPY INDD=((INPUT,R)),OUTDD=OUTPUT
   /*
   ```

4. Run the job to copy the objects. The job should finish with return code zero. If the return code is not zero, inspect the job output to determine the cause of the problem, and run the job again if necessary. If you do have to resubmit this job, you may need to compress the OUTPUT data set.

Chapter 8. Testing GDDM installation

This chapter describes how to test that GDDM/MVS Base and the other GDDM licensed programs have been successfully installed.

This chapter provides separate sections on testing GDDM under TSO, CICS, and IMS subsystems.

This chapter also describes what to do if any of the tests fail, and what to do when you have finished testing.

Before you begin

Make sure that the terminal you are going to use for these tests can show graphics, and that it is configured correctly. If, during a test, a GDDM message is displayed that says `ADM0275 W GRAPHICS CANNOT BE SHOWN`, your terminal does not support graphics. You should consider repeating the test on a terminal that does.

Instructions for testing

Before you can test GDDM/MVS and any of the GDDM licensed programs, you must have installed support for at least one national language from GDDM/MVS NLS. If you have not installed NLS for U.S. English, use the NATLANG external default to specify the language you want to use.

To specify a national language other than U.S. English for your enterprise, add the following line to the external defaults file, or the external defaults module:

```
 ADMMDFT NATLANG=N
```

where `N` identifies the new language selected, in this instance Norwegian. The line must start with a space in column 1.

You must also have defined GDDM to your CICS system, and loaded the GDDM VSAM data sets using the instructions in Chapter 7, “Post-installation tasks” on page 33.

This section is divided into three parts, depending on which subsystem of MVS you are using:

- If you are using TSO, see “Testing GDDM under TSO”
- If you are using CICS, see “Testing GDDM under CICS” on page 69
- If you are using IMS, see “Testing GDDM under IMS” on page 72

Testing GDDM under TSO

This section describes how to test the GDDM licensed programs you have installed under the TSO subsystem of MVS.

To use GDDM under TSO, you need to allocate some specific data sets. For more information, see “Defining GDDM to TSO” on page 34. If you have already allocated these, you can ignore the ALLOC statements in the following steps.

Testing GDDM/MVS Base

This is how to test that GDDM/MVS has been installed successfully for use under TSO:

1. Allocate the GDDM symbol and map data sets on your TSO session by typing:

   ```
   ALLOC F(ADMSYMBL) DA('GDDM.SADMSYM') SHR REU
   ALLOC F(ADMGDF)   DA('GDDM.SADMGDF') SHR REU
   ```

   If you also intend to use this step to test the GDDM TSO Print Utility, you must allocate the Master Print Queue data set:

   ```
   ALLOC F(ADMPRNTQ) DA('your-master-print-queue') SHR REU
   ```

2. Run the GDDM installation verification program:

   ```
   CALL 'GDDM.SADMMOD(ADMUGC)' 'FROM(ADMTEST)'
   ```

   A picture should be displayed that shows the words “Welcome to GDDM Version 3.2”. This picture is the ADMTEST ADMGDF from the SADMGDF data set being displayed by the ADMUGC conversion utility.

3. Go into GDDM User Control by pressing the User Control key.

   The default User Control key is PA3. If your keyboard does not have a PA3 key, or the PA3 key is already used by another application, set the User Control key to another value. To do this, add the following nickname user default specification to the external defaults file, or external defaults module:

   ```
    ADMMNICK FAM=1,PROCOPT=((CTLKEY,type,value))
   ```

   where `type` defines the type of key selected for entering User Control (1=PF key, 4=PA key), and `value` defines the number of the PF or PA key to be used. The line must start with a space in column 1.

   For more information about nicknames and supplying user-default specifications, see the *GDDM System Customization and Administration* book.

   For more information about User Control, see the *GDDM User's Guide*.

   The User Control panel should now be superimposed on the bottom of the display, using the language specified by the NATLANG external default.

   While you are in this panel, you can also test the GDDM print utility if you have already customized it. For information about the print utility, see the *GDDM System Customization and Administration* book. To create a file for testing the GDDM print utility using GDDM User Control:

   a. Press PF4 for the User Control output panel.

   b. Type the terminal ID of the printer to which you want to send the output, as defined to VTAM or in your external defaults module.

   c. Press PF4 to send the file to be printed. A highlighted message is displayed when the print has completed successfully.

4. Press PF3 until you are back in TSO.

This completes the test of the graphic and alphanumeric functions of GDDM/MVS.

Testing GDDM-PGF

This is how to test that GDDM-PGF has been installed successfully.

Use the commands suggested for a CLIST in “CLISTs for TSO users” on page 36 to call the ICU:

1. Run the CLIST.

   When you start, the Home Panel of the ICU is displayed in the language specified in the NATLANG external default.

   For using the ICU, there is a built-in Help facility available, which you can view by pressing the PF1 key; you can also find more information in the *GDDM-PGF Interactive Chart Utility* book.

2. This is what you do:

   a. Type 0 to move to the Chart by Example panel.

   b. Type 1 on the Chart by Example panel to get data headings and titles for your chart.

   c. Type any number from 1 through 7 on the Chart by Example - Data panel to get the sample data supplied with the ICU.

   d. Display the data by pressing PF5, the Display key.

   e. Return to the Home panel by pressing PF12.

   f. Exit the ICU by pressing PF9 twice.

This tests the graphic and alphanumeric functions of GDDM-PGF.

Testing GDDM-IMD

This is how to test that GDDM-IMD has been installed successfully. Use the commands suggested for a CLIST in “CLISTs for TSO users” on page 36 to call GDDM-IMD:

1. Create an MSL if you have not already done so. For more information, see “Managing the VSAM data sets for use with GDDM-IMD under TSO” on page 35.

2. Substitute the name of your MSL into your IMD CLIST, and run the CLIST.

   If you have just created your MSL, the first GDDM-IMD panel to be displayed is the Initial Selection panel, numbered 0.0. You get the message:

   ```
   AEM00204I  NO OBJECTS IN MSL. PRESS PF1 IF YOU NEED HELP
   ```

   Otherwise, the Directory List panel, 0.1, is displayed.

3. Press PF1, the Help key.

   A tutorial panel is displayed.

4. Press PF3, the End key, until you return to GDDM-IMD panel 0.0.

5. Try out all the options in panel 0.0, returning to panel 0.0 each time by pressing PF3.

6. Exit GDDM-IMD from panel 0.0 by pressing PF3.

Testing GDDM-IVU

This is how to test that GDDM-IVU has been installed successfully. A sample image file, ADMU5IMG, and a sample projection file, ADMU5PRJ, are provided with GDDM to help you with this task. Use the commands suggested for a CLIST in "CLISTs for TSO users" on page 36.

1. Run the CLIST.

   The Home panel of the Image View Utility is displayed in the language specified in the NATLANG external default.

2. Press PF4, Input.

   The Input Selection panel is displayed.

3. Type ADMU5IMG in the "Input image" field.

4. Press ENTER.

   The View Image panel is displayed, with an image showing part of a job application form. The full image is shown in the *GDDM Image View Utility* book.

5. As an additional optional test:

   a. Press PF3
   b. Type 2
   c. Press ENTER
      The View Image panel is displayed.
   d. Type ADMU5PRJ in the "Projection" field.
   e. Press ENTER
      You then see selected parts of the sample image arranged as a badge.
   f. Press PF2, the No Menu key, to view the complete image.

6. Exit GDDM-IVU by pressing PF12 once, then pressing PF3 twice.

Setting up the GDDM-IVU CLIST for your users: When testing is complete, you are advised to copy the CLIST you have generated into a generally available library so that your TSO users can access it. The *GDDM Image View Utility* book assumes that you are going to call the CLIST GDDMIVU. If you decide not to do this, tell your users how they can access GDDM-IVU.

Testing GDDM-GKS

This is how to test that GDDM-GKS has been installed successfully. You need:

- A terminal that can display graphics
- One of these FORTRAN compilers:

  VS FORTRAN
  FORTRAN IV G
  FORTRAN IV H.

If FORTRAN is not available on your computer system, you cannot test the installation of GDDM-GKS now.

Testing consists of compiling, linking, and running the ADMJROOM sample program. The program has been written so that it can be compiled and run using one of the above FORTRAN compilers.

Compiling the ADMJROOM program: Compile ADMJROOM using the JCL based on the following:

```
//ADMJFORT JOB
//COMPILE EXEC PGM=FORTVS2,REGION=3M
//STEPLIB  DD DSN=your.VSF2COMP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSIN    DD DSN=GDDM.SADMSAM(ADMJROOM),DISP=SHR
//SYSLIN   DD DSN=your.obj.dataset,DISP=OLD
//SYSLIB   DD DSN=GDDM.SADMMOD,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,DISP=NEW,SPACE=(CYL,(5,5))
/*
```

Notes:

1. The names of the compiler program and the STEPLIB data set are correct for VS FORTRAN Version 2. If you are using FORTRAN IV or VS FORTRAN Version 1, you must change these names.
2. If you do not have a suitable SYSLIN data set that you can use for the object deck, you must create one with DCB characteristics LRECL=80, RECFM=FB. SYSLIN can be a sequential or a partitioned data set.

Linking the ADMJROOM program: Link ADMJROOM using JCL based on the following:

```
//ADMGLINK JOB
//LINK   EXEC PGM=IEWL,PARM='AMODE=31,RMODE=ANY,MAP,LIST',REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSLMOD   DD DSN=your.load.dataset,DISP=OLD
//INPUT     DD DSN=your.obj.dataset,DISP=SHR
//SYSLIB    DD DSN=your.VSF2FORT,DISP=SHR
//           DD DSN=GDDM.SADMMOD,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,DISP=NEW,SPACE=(CYL,(5,5))
//SYSLIN   DD *
 INCLUDE INPUT(ADMJROOM)
 INCLUDE SYSLIB(ADMASNT)
 INCLUDE SYSLIB(ADMJB77)
 NAME ADMJROOM(R)
/*
```

Notes:

1. If you do not have a suitable SYSLMOD data set that you can use for the ADMJROOM program, you must create one with DCB characteristics LRECL=0, RECFM=U. SYSLMOD can be a sequential or a partitioned data set.
2. ADMJB77 is the GDDM GKS FORTRAN call module for use with FORTRAN 77 compilers such as VS FORTRAN. If you are using a FORTRAN IV compiler, replace the INCLUDE for ADMJB77 with ADMJBIV.

Running the ADMJROOM program: To load and run the program, set up and run the following CLIST:

```
LISTDSI GKS.ERRORLOG
IF &LASTCC = 0 THEN DELETE GKS.ERRORLOG
ALLOC F(ADMSYMBL) DA('GDDM.SADMSYM') SHR REU
ALLOC F(FT06F001) DA(*) SHR REU
ALLOC F(ADMERLOG) DA(GKS.ERRORLOG) REU NEW
    SPACE(5,1) UNIT(SYSDA) LRECL(80) RECFM(F) BLKSIZE(6160)
CALL 'GDDM.SADMMOD(ADMJROOM)'
FREE  F(ADMSYMBL)
FREE  F(FT06F001)
FREE  F(ADMERLOG)
```

When ADMJROOM runs, the first panel of a simple interactive program is displayed. More information about running this program can be found in the *GDDM-GKS Programming Guide and Reference* book.

When you have finished with the program, press PF5 to exit GDDM-GKS.

Setting up ADMJROOM as a demonstration program: When testing is complete, you are recommended to copy the CLIST you generated in the previous section into a generally available library, so that your TSO users can use ADMJROOM as a demonstration program. The *GDDM-GKS Programming Guide and Reference* book assumes that you are going to do this, and that you are going to call the CLIST “GKSROOM”. If you decide not to do this, tell your users how they can use the ADMJROOM program.

Testing GDDM-REXX

Before you can test GDDM-REXX, it must first be enabled. For information about how to do this, refer to the document *How to activate GDDM-REXX*, LY33-6080, that was shipped with the feature.

This is how to test that GDDM-REXX has been installed successfully:

1. Allocate the GDDM map and symbol set data sets to your session by typing:

   ```
   ALLOC F(ADMGDF) DA('GDDM.SADMGDF') SHR REU
   ALLOC F(ADMSYMBL) DA('GDDM.SADMSYM') SHR REU
   ```

2. Type the command `EXEC 'GDDM.SADMSAM(ERXMODEL)'` from a terminal that can display graphics.
3. Press ENTER.

   A picture is displayed, with the large characters “GDDM-REXX” near the top of the screen.

4. Press ENTER again or press the PF3 key to return to TSO.

Testing the print utility ADMOPUT and ADMOPUJ

To test the GDDM print utility ADMOPUT and ADMUPUJ, use the tests in “Testing GDDM/MVS Base” on page 64.

For information about setting up the print utility and how to select particular printers, see the *GDDM System Customization and Administration* book.

Testing the telecommunication network

This is how to test that you have established the telecommunication network successfully:

1. Try running GDDM on all the types of terminals and different types of connections that you intend using with GDDM at your enterprise.
2. Ensure that terminals you do not test are set up in the same way as terminals of the same type that have been tested successfully.

You have now completed the tests for the GDDM licensed programs you have installed under the TSO subsystem. Continue reading at "What to do if any of the installation tests fail" on page 74.

Testing GDDM under CICS

This section describes how to test the GDDM licensed programs you have installed under the CICS subsystem of MVS.

To test GDDM under CICS, you must have allocated the required GDDM VSAM data sets, defined them to CICS, and included them in the CICS startup JCL. You must also have defined GDDM to CICS in your CICS tables or CSD. For more information, see "Managing the CICS VSAM data sets required by GDDM" on page 41, and "Defining GDDM to CICS" on page 40.

Testing GDDM/MVS Base

This is how to test that GDDM/MVS has been installed successfully for use under the CICS subsystem:

1. Type the transaction name for the GDDM/MVS installation verification program: ADMA.
2. Press ENTER

   A picture should be displayed that shows the words "Welcome to GDDM Version 3.2". This picture is the ADMTEST ADMGDF, which is stored in the ADMF data set.
3. Go into GDDM User Control by pressing the User Control key.

   The default User Control key is PA3. If your keyboard does not have a PA3 key, or the PA3 key is already used by another application, set the User Control key to another value. To do this, add the following nickname user default specification to the external defaults file, or external defaults module:

   ```
   ADMMNICK FAM=1,PROCOPT=((CTLKEY,type,value))
   ```

   where `type` defines the type of key selected for entering User Control (1=PF key, 4=PA key), and `value` defines the number of the PF or PA key to be used. The line must start with a space in column 1.

   For more information about nicknames and supplying user-default specifications, see the *GDDM System Customization and Administration* book.

   For more information about User Control, see the *GDDM User's Guide*.

   The User Control panel should now be superimposed on the bottom of the display, using the language specified by the NATLANG external default.

   While you are in this panel, you can also test the GDDM print utility if you have already customized it. For information about the print utility, see the *GDDM*

System Customization and Administration book. To create a print file for testing the GDDM print utility using GDDM User Control:

a. Press PF4 for the User Control output panel.

b. Type the terminal ID of the printer to which you want to send the output, as defined in your TCT.

c. Press PF4 to send the file to be printed.

A highlighted message is displayed when the print has completed successfully.

4. Press PF3 until you have left the transaction.

This completes the test of the graphic and alphanumeric functions of GDDM/MVS.

Testing GDDM-PGF

This is how to test that GDDM-PGF has been installed successfully:

1. Type the transaction name for the ICU: ADMC
2. Press ENTER.

   The Home Panel of the ICU is displayed in the language specified in the NATLANG external default.

   For using the ICU, there is a built-in Help facility available, which you can view by pressing the PF1 key; you can also find more information in the *GDDM-PGF Interactive Chart Utility* book.

3. This is what you do:

   a. Type 0 to move to the Chart by Example panel.

   b. Type 1 on the Chart by Example panel to get data headings and titles for your chart.

   c. Type any number from 1 through 7 on the Chart by Example - Data panel to get the sample data supplied with the ICU.

   d. Display the data by pressing PF5, the Display key.

   e. Return to the Home panel by pressing PF12

   f. Exit the ICU by pressing PF9 twice.

This tests the graphic and alphanumeric functions of GDDM-PGF.

Testing GDDM-IMD

This is how to test that GDDM-IMD has been installed successfully:

1. Type the GDDM-IMD transaction code ADMM, followed by the File Control Table (FCT) name of a user MSL VSAM data set that you included in your FCT or CSD when you updated the CICS tables.
2. Press ENTER.

   If you have just created your MSL, the first GDDM-IMD panel to be displayed is the Initial Selection panel numbered 0.0. You get the message:

   ```
   AEM00204I  NO OBJECTS IN MSL. PRESS PF1 IF YOU NEED HELP
   ```

   Otherwise, the Directory List panel, 0.1, is displayed.

3. Press PF1, the Help key.

A tutorial panel is displayed.

4. Press PF3, the End key, until you return to GDDM-IMD panel 0.0.
5. Try out all the options in panel 0.0, returning to panel 0.0 each time by pressing PF3.
6. Exit GDDM-IMD from panel 0.0 by pressing PF3.

Testing GDDM-IVU

This is how to test that GDDM-IVU has been installed successfully. A sample image file, ADMU5IMG, and a sample projection file, ADMU5PRJ, are provided with GDDM to help you with this task.

1. Type the command to run GDDM-IVU: ADMU
2. Press ENTER.

   The Home panel of the GDDM Image View Utility is displayed in the language specified by the NATLANG external default.
3. Press PF4, Input.

   The Input Selection panel is displayed.
4. Type ADMU5IMG in the “Input image” field.
5. Press ENTER.

   The Input Selection panel is displayed, with an image showing part of a job application form. The full image is shown in the *GDDM Image View Utility* book.
6. As an additional optional test:
   a. Press PF3
   b. Type 2
   c. Press ENTER.
      The View image panel is displayed.
   d. Type ADMU5PRJ in the “Projection” field.
   e. Press ENTER.
      You then see selected parts of the sample image arranged as a badge.
   f. Press PF2, the No Menu key, to see the complete image.
7. Exit GDDM-IVU by pressing PF12 once, then pressing PF3 twice.

Testing the print utility ADMOPUC

To test the GDDM print utility ADMOPUC, use the tests in “Testing GDDM/MVS Base” on page 69.

For information about setting up the print utility and how to select particular printers, see the *GDDM System Customization and Administration* book.

Testing the telecommunication network

This is how to test that you have established the telecommunication network successfully:

1. Try running GDDM on all the types of terminals and different types of connections that you intend using with GDDM at your enterprise.
2. Ensure that terminals you do not test are set up in the same way as terminals that have been tested successfully.

You have now completed the tests for the GDDM licensed programs you have installed under the CICS subsystem. Continue reading at “What to do if any of the installation tests fail” on page 74.

Testing GDDM under IMS

This section describes how to test the GDDM licensed programs you have installed under the IMS subsystem of MVS.

Testing GDDM/MVS Base

1. Type the transaction name for the Image Symbol Editor: ADM ISSE
2. Press ENTER.

   If you are not familiar with the Image Symbol Editor, there is a help facility that you can view by pressing the PF1 key. You can also find more information in the *GDDM Using the Image Symbol Editor* book.

   When you start the transaction, the first panel of the Image Symbol Editor is displayed. It is called Step Selection.

   This what you do:

   a. Type the symbol set name ADMDHII. (note the final period), and choose option 2, Edit Symbol Set.

   b. Press ENTER.

      The next panel, Symbol Selection, is displayed.

   c. Press PF6.

      A different set of characters should be displayed on the same panel. (If GDDM message ADM0824 or ADM0825 is displayed, and some of the characters are displayed as “?”, this does not invalidate the test).

   d. Move the cursor to a nonblank character in the set of characters (not one in reverse-video).

   e. Press ENTER

      The display should change to the Symbol Definition panel, and the pixel pattern of the chosen symbol should be displayed at the left of the screen. (If message ADM0824 or ADM0825 was displayed earlier, the symbol is the one that could not be displayed and not the “?” chosen).

   f. Leave the cursor where it is, and type the command TEST ON

   g. Press ENTER

      If your device supports programmed symbols (PS), a small copy of the character should be displayed below and to the right of the pixel pattern. Otherwise, GDDM message ADM0861 is displayed, but this does not invalidate the test.

   h. Exit the Image Symbol Editor by pressing PF3 three times.

This tests both the graphic and alphanumeric functions of GDDM.

If your workstation is capable of showing graphics, but does not have PS support, you can test the graphic functions of GDDM/MVS by running one of the sample programs, described in the *GDDM Base Application Programming Reference* book.

Testing GDDM-PGF

This is how to test that GDDM-PGF has been installed successfully:

1. Type the transaction name for the ICU: ADM CHART
2. Press ENTER.

   The Home Panel of the ICU is displayed in the language specified by the NATLANG external default.

   For using the ICU, there is a built-in Help facility available, which you can view by pressing the PF1 key; you can also find more information in the *GDDM-PGF Interactive Chart Utility* book.
3. This is what you do:
   a. Type 0 to move to the Chart by Example panel.
   b. Type 1 on the Chart by Example panel to get data headings and titles for your chart.
   c. Type any number from 1 through 7 on the Chart by Example - Data panel to get the sample data supplied with the ICU.
   d. Display the data by pressing PF5, the Display key.
   e. If you have a printer, create a print file at this point so that you can later test the GDDM print utility (see "Testing the print utility ADMOPUI"):
      1) Press PF4, the Print key.
      2) Type the LTERM name of the printer.
      3) Press ENTER.

         The message `CHART SUCCESSFULLY OUTPUT` is displayed.
   f. Return to the Home panel by pressing PF12.
   g. Exit the ICU by pressing PF9 twice.

This tests the graphic and alphanumeric functions of GDDM-PGF.

Testing the print utility ADMOPUI

This is how to test that the print utility ADMOPUI has been installed successfully. However, before you can start, you must have a GDDM file for it to print. If GDDM-PGF has been installed, you can use the ICU to produce one; see "Testing GDDM-PGF."

If GDDM-PGF has not been installed, you can create a print file using the IMS version of the sample program ADMUSP1. The source for this program is called ADMUSP1I.

Find the program in the sample library SADMSAM, and compile and link-edit it as described in the *GDDM Base Application Programming Guide* book.

1. Give the LTERM name on the transaction invocation.
2. Assign the ADMPRINT transaction to a suitable class, and start it.

   After you have created a print file, the print utility prints it asynchronously.

Testing the telecommunication network

This is how to test that you have established the telecommunication network successfully:

1. Try running GDDM on all the types of terminals and different types of connections that you intend using with GDDM at your enterprise.
2. Ensure that terminals you do not test are set up in the same way as terminals that have been tested successfully.

You have now completed the tests for the GDDM licensed programs you have installed under the IMS subsystem. Continue reading below.

What to do if any of the installation tests fail

If any of the tests fail, the first thing you may see is an error message displayed on your screen. On the other hand, you may find that graphics are not displayed on your screen.

Error messages

If you receive an error message, look it up in the *GDDM Messages* book. If it is a GDDM-OS/2 Link message, you can also use the online help.

Check the three-character prefix of the message; this tells you from what part of GDDM the message originates.

Table 23. GDDM error messages

GDDM component	Message prefix
GDDM Base	ADM...
GDDM-PGF	ADM...
GDDM-IVU	ADM...
GDDM-GKS	ADM...
GDDM-IMD	AEM...
GDDM/REXX	ERX...
GDDM-PCLK	GQD...
GDDM-OS/2 Link	GQF...

Inability to show graphics

If graphics cannot be shown on one or more of the terminals when you test GDDM or the telecommunication network, see the *GDDM Diagnosis* book.

Diagnosing errors

If you cannot quickly identify the cause of the error, you may find it useful to read the *GDDM Diagnosis* book, which contains detailed information about diagnosing problems with GDDM and its components. However, if after reading this book you still cannot solve the problem, you should report it to the IBM Support Center for further investigation. Information about detailed diagnosis and the procedure for reporting errors can be found in the *GDDM Diagnosis* book.

After you have finished testing

Normally, the first task to be performed on a successfully installed system is customization; this includes tuning the operating system-related values and defining input and output devices to GDDM. For information about how to do these tasks, see the *GDDM System Customization and Administration* book.

Telling your users about GDDM

Migrating from earlier releases

Tell your users about the benefits they can expect by using GDDM Version 3 Release 2.

When you have successfully installed GDDM, you need to tell your users that GDDM Version 3 Release 2 is now available on the system.

Details about what is available can be found in the *GDDM General Information* manual.

after testing is finished

Chapter 9. Servicing GDDM

This chapter describes how to install service for GDDM and its associated licensed programs under MVS.

GDDM service is installed under TSO using SMP/E.

If you are using GDDM under CICS or IMS, there may be additional steps you need to do to make the service available.

Most of the information is in the section “Servicing GDDM under TSO,” but subsystem-specific information for CICS and IMS is given in later sections.

Because GDDM-PCLK, GDDM-OS/2 Link, and GDDM-REXX are integral parts of GDDM/MVS Base for GDDM Version 3 Release 2, there are no special service requirements over and above those for GDDM/MVS.

All GDDM licensed programs, whose installation is described in this book and the associated Program Directories, have central service including IBM Support Center, and are serviced in the normal manner for IBM licensed programs.

Servicing GDDM under TSO

You must decide how you are going to install GDDM service, what testing is required, and how you are going to make it available to your users.

How service affects GDDM

Installing GDDM service can have different effects, depending on what parts of GDDM are updated. This can be a factor in deciding whether to test the service and if so, how you are going to test it. The different groups of parts of GDDM are as follows:

- Load modules. These are all shipped as ++MOD elements. These directly affect the operation of GDDM and are the most important category you might want to test. You may need to relink user-written applications, or reapply user mods if these are affected. If you have relinked GDDM for performance reasons, you must do the linkedit again.
- GDDM-PCLK and GDDM-OS/2 Link objects. These are all shipped as ++DATA elements, and have names beginning GQD and GQF respectively. They affect users of workstations who call GDDM applications but do not directly affect the operation of GDDM. They are automatically downloaded to any workstation that accesses them after service has been applied.
- User-modifiable source, such as sample JCL or the source of the default modules. Applying service to these does not affect the operation of GDDM, but you could inadvertently overwrite changes you have made. You should always keep copies of modified source outside of the GDDM data sets or change them using SMP/E usermods, to ensure that they do not get overwritten.

  You should merge any changes made to these samples into your modified copy and recreate the executable version if applicable. For information about recreating the executable version of the GDDM default modules, see the

GDDM System Customization and Administration book. Whether the change justifies formal testing depends on what is updated.

- Other GDDM objects, such as symbol sets or GDDM-IMD panels. These are shipped as ++DATA elements and have names beginning ADM, AEM, or ERX. These do not affect the general use of GDDM, but they could cause problems to particular applications that use them.
- Source files, such as macros or samples. These are shipped as ++MAC elements. These have no immediate affect on the operation of GDDM, but may mean that you need to change user-written applications if they use any of these elements.

Testing GDDM service

How extensively you need to test GDDM depends on how important GDDM is to your enterprise, what use you make of GDDM, and what components of GDDM are updated. You must decide what you want to test before you can plan how you are going to test it.

If you have already have an established procedure for testing service to products, you should follow it for GDDM. Otherwise, consider a method based on the following. These suggested steps refer to testing for TSO; more information is provided later for CICS and IMS, but the principle is the same.

1. Create new GDDM target data sets to hold the test level of GDDM.

   Whether you create all of them depends on what you want to test. For example, you may choose to create only the SADMMOD data set, and if you are using GDDM-PCLK or GDDM-OS/2 Link, the SADMPCF data set. (Changes to the other data sets should be transparent to users, although you must not change them while they are being used). You can either give the data sets new names, such as GDDM.TEST.SADMMOD, or create uncataloged versions with the same names or using a user catalog.

2. Copy your existing data sets to the new data sets.
3. Change your SMP/E DDDEFs to point to the new data sets.
4. Use SMP/E to receive and apply the GDDM service.
5. Test the service.

   If your production system uses GDDM from the linklist or LPA, you must use STEPLIB to override it, or if you have sole access to the system, you could change it to point to your test data set and re-IPL or refresh the linklist. For general information about testing GDDM, see Chapter 8, “Testing GDDM installation” on page 63.

6. When you are satisfied that your applications work with the new level of GDDM, put the new level of GDDM into production.

   Depending on how you use GDDM, you can rename the data sets, or copy their contents, or change the SMP/E DDDEFs back and use APPLY REDO. Further considerations about making the new level of GDDM available are discussed in the section “Using SMP/E to install GDDM service” on page 79.

7. Use SMP/E to accept the GDDM service.
8. Delete any test levels of GDDM.

Using SMP/E to install GDDM service

Before using the instructions here, ensure that you understand the implications of installing GDDM service. These are discussed in the previous two sections “How service affects GDDM” on page 77, and “Testing GDDM service” on page 78.

If your SMP/E DDDEFs point to your production GDDM data sets, using SMP/E to apply GDDM service updates them directly. You should only update your production GDDM data sets when you are sure that noone else is using them, otherwise your user's applications could fail.

If you have an established procedure for installing service using SMP/E, then use it for GDDM. Alternatively, you could use one of these methods:

- The SMP/E dialogs
- Modified copies of related installation materials (RIMs) shipped with a system package
- Modified copies of the sample installation JCL shipped with GDDM. These jobs are called ADMUJRB (receive), ADMUJAPP (apply), and ADMUJACC (accept). If you used these samples at installation time, they will be in the SADMINS data set. Otherwise, use a copy of the version in the SADMSAM data set.

Point SMPPTFIN to the tape or data set containing the service. PUT service is shipped on an unlabeled tape, so to receive from PUT you can use a JCL statement as follows:

```
//SMPPTF1N DD UNIT=XXXX,VOL=SER=DUMMYL,
//   LABEL=(,NL),DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)
```

Use SMP/E to receive, apply, and accept the service. GDDM can be tested after the apply step.

When installing GDDM service using SMP/E, the following are acceptable return codes:

Receive 0
Apply 0
Accept 0 or 4

If you get a return code higher than that expected, check the job output carefully to find out why. Correct any errors, and rerun each step as necessary. For more information about SMP/E return codes and messages, and help in correcting any errors, see the *SMP/E Messages and Codes*, and *SMP/E Reference* books.

If you have moved (rather than copied) any GDDM load modules from the SADMMOD data set, the SMP/E APPLY step fails because SMP/E cannot find the module. Use UCLIN to change the SYSLIB information on LMOD entry for the moved load module.

If you have copied any GDDM load modules from the SADMMOD data set to another data set, the copy will not be updated when you install service using SMP/E. You must ensure that you refresh the copied versions of the load modules after you have applied any service that affects them. This also applies to GDDM load modules such as the application interface stubs that you may have linked with your application outside SMP/E.

Any SMP/E usermods you have applied may be superseded by GDDM service. In this case, restore the usermod, apply the GDDM service, then rework and reapply the usermod. If you have used usermods to update the GDDM default modules, and need more information about reworking the usermods, see the *GDDM System Customization and Administration* book.

If you have put the SADMMOD data set in the linklist, any updates you make to it will not be available until you refresh the linklist. Changing linklist data sets while they are in use is not recommended, because user applications may fail until you refresh the linklist. If you have moved a data set in the linklist, you must re-IPL to refresh the linklist.

If you have put the SADMMOD data set in the LPA list, you must do a Clear Link Pack Area (CLPA) IPL of MVS before any changes become available. Remember:

- GDDM load modules ADMECOMT and ADMOPUT are non-reentrant and cannot run from the LPA. You must move these modules to a separate loadlib that you must add to the STEPLIB.
- LPA data sets need to be APF-authorized and can be required to be on a particular volume. If you have moved the data set and not updated the APF entry, you will exclude the data set from the LPA.

If you have relinked GDDM for performance reasons, or linked parts of GDDM other than the interface stubs with your applications, you must do the link-edit again with the updated level of GDDM.

Servicing GDDM under CICS

Service for GDDM under CICS is installed on TSO using SMP/E. This is described in the previous section “Servicing GDDM under TSO” on page 77. Installing service for GDDM for use with CICS differs from TSO in two respects:

- There are additional tasks that need to be done to make service to GDDM objects available. The objects are serviced in the GDDM targets data sets, but accessed from VSAM data sets. You must refresh the copy in the VSAM data set to make the new level available for use with CICS. See “Loading the VSAM data sets” on page 44.
- Testing new levels of GDDM with CICS is not as easy to do as under TSO. This is described in the next section.

Testing new levels of GDDM under CICS

If you have an established procedure for testing updated versions of CICS transactions, follow it for GDDM. You should create copies of the VSAM data sets, particularly if you want to test updated GDDM objects. For general information about testing, see Chapter 8, “Testing GDDM installation” on page 63.

If you have a test CICS system on which you prove new application levels, you can test GDDM without affecting your production system. If you have put GDDM in the LPA, change your CICS resource definitions to point to your test level of GDDM. Otherwise, update your DFHRPL list.

If a concurrent production and test CICS environment is not practical at your enterprise, create a copy of the production CICS startup job stream, giving it a

different name. In the copy, refer to the new libraries and data sets. Provide System Information Table (SIT) overrides to use new versions of your tables with the new GDDM-specific entries.

During off-peak hours, the production system can be brought down and the test version started up. The GDDM-specific items can be checked out and adjustments made, until the test version is ready for use in the production environment.

Then, the production CICS startup jobstream can be adjusted as needed, using the experience gained from the test version. The original production CICS startup should be kept as a backup until the new one has been exercised by users and confirmed to be working correctly.

Servicing GDDM under IMS

Service for GDDM under IMS is installed on TSO using SMP/E. This is described in the section “Servicing GDDM under TSO” on page 77.

To make GDDM service available under IMS, you need to:

- Include any changed GDDM utilities in your IMS.PGMLIB. See “Defining GDDM to IMS” on page 49.
- Reload any changed objects into the object database. See “Setting up the GDDM object database” on page 59.

Reinstalling GDDM after system generation for another product

The GDDM series of programs are not eligible for system generation, because they include inline JCLIN. If you perform system generation activities for another product that is installed in the same SMP/E target zone, you may need to reallocate the target zone. If this is the case, the target library information for GDDM and its features will have been lost.

You do not need to reinstall GDDM.

Instead, to re-create this information, see the discussion about system generation in the *SMP/E User's Guide* and the *SMP/E Reference*, for instructions about using the SMP/E GENERATE command to create a job stream for regenerating the target zone.

Appendix A. GDDM/MVS Install logic

This is the System Modification Program install logic for GDDM/MVS:

```
++FUNCTION(HGD3200) FILES(10) FESN(0565485)
                    REWORK(1996114) RFDSNPFX(IBM)
/* COPYRIGHT;
         5695-167 (C) COPYRIGHT IBM CORP. 1979, 1996.
         ALL RIGHTS RESERVED
         LICENSED MATERIALS - PROGRAM PROPERTY OF IBM

         US Government Users Restricted Rights -
         Use, duplication or disclosure restricted
         by GSA ADP Schedule Contract with IBM Corp.
 */.
++VER(C150) DELETE(HGD2100,HGD2104,HGD2202,HGD2300,HGD3100,
            HDDM100,HGD3110,HGD3101,HGD3102,HGD3103,HGD3104)
            SUP(HGD3110).
++VER(Z038) DELETE(HGD2100,HGD2104,HGD2202,HGD2300,HGD3100,
            HDDM100,HGD3110,HGD3101,HGD3102,HGD3103,HGD3104)
            SUP(HGD3110).
++VER(P115) DELETE(HGD2100,HGD2104,HGD2202,HGD2300,HGD3100,
            HDDM100,HGD3110,HGD3101,HGD3102,HGD3103,HGD3104)
            SUP(HGD3110).
++DELETE (ADMACIN)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASNC)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASNI)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASNJ)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASNT)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASPC)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASPI)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASPJ)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASPP)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASPT)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASPLC) SYSLIB(GDDMLOAD) .
++DELETE (ADMASPLI) SYSLIB(GDDMLOAD) .
++DELETE (ADMASPLJ) SYSLIB(GDDMLOAD) .
++DELETE (ADMASPLO) SYSLIB(GDDMLOAD) .
++DELETE (ADMASPLT) SYSLIB(GDDMLOAD) .
++DELETE (ADMASRC)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASRI)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASRJ)  SYSLIB(GDDMLOAD) .
++DELETE (ADMASRT)  SYSLIB(GDDMLOAD) .
++DELETE (ADMDSA1)  SYSLIB(GDDMLOAD) .
++DELETE (ADMEGFAC) SYSLIB(GDDMLOAD) .
++DELETE (ADMEGFBC) SYSLIB(GDDMLOAD) .
++DELETE (ADMEGFFC) SYSLIB(GDDMLOAD) .
++DELETE (ADME000C) SYSLIB(GDDMLOAD) .
++DELETE (ADMFOU)   SYSLIB(GDDMLOAD) .
++DELETE (ADMFSDU)  SYSLIB(GDDMLOAD) .
++DELETE (ADMISSEC) SYSLIB(GDDMLOAD) .
++DELETE (ADMISSET) SYSLIB(GDDMLOAD) .
++DELETE (ADMKSCHI) SYSLIB(GDDMLOAD) .
++DELETE (ADMLRO1C) SYSLIB(GDDMLOAD) .
++DELETE (ADMLRO2C) SYSLIB(GDDMLOAD) .
++DELETE (ADMLRO3C) SYSLIB(GDDMLOAD) .
++DELETE (ADMOPRT)  SYSLIB(GDDMLOAD) .
```

```
++DELETE (ADMOPST)  SYSLIB(GDDMLOAD) .
++DELETE (ADMOPUC)  SYSLIB(GDDMLOAD) .
++DELETE (ADMOPUI)  SYSLIB(GDDMLOAD) .
++DELETE (ADMOPUJ)  SYSLIB(GDDMLOAD) .
++DELETE (ADMOPUT)  SYSLIB(GDDMLOAD) .
++DELETE (ADMUCG)   SYSLIB(GDDMLOAD) .
++DELETE (ADMUGC)   SYSLIB(GDDMLOAD) .
++DELETE (ADMUIMPT) SYSLIB(GDDMLOAD) .
++DELETE (ADMUOTT)  SYSLIB(GDDMLOAD) .
++DELETE (ADMUPCT)  SYSLIB(GDDMLOAD) .
++DELETE (ADMUPGT)  SYSLIB(GDDMLOAD) .
++DELETE (ADM4CDUC) SYSLIB(GDDMLOAD) .
++DELETE (ADM4CDUT) SYSLIB(GDDMLOAD) .
++DELETE (AEMPMS00) SYSLIB(GDDMLOAD) .
++DELETE (GQFINSTC) SYSLIB(GDDMLOAD) .
++DELETE (GQFINSTT) SYSLIB(GDDMLOAD) .
++DELETE (ADMPSTBT) SYSLIB(GDDMLOAD) .
++DELETE (ADMUCDSO) SYSLIB(GDDMLOAD) .
++DELETE (ADMVSSET) SYSLIB(GDDMLOAD) .
++DELETE (ADMPSTBC) SYSLIB(GDDMLOAD) .
++DELETE (ADMVSSEC) SYSLIB(GDDMLOAD) .
++DELETE (ADM1IMDC) SYSLIB(GDDMLOAD) .
++DELETE (ADM1IMDT) SYSLIB(GDDMLOAD) .
++DELETE (ADM5IVUC) SYSLIB(GDDMLOAD) .
++DELETE (ADM5IVUT) SYSLIB(GDDMLOAD) .
++DELETE (ADMU5DC)  SYSLIB(GDDMLOAD) .
++DELETE (ADMU5DD)  SYSLIB(GDDMLOAD) .
++DELETE (ADMU5DG)  SYSLIB(GDDMLOAD) .
++DELETE (ADMU5DP)  SYSLIB(GDDMLOAD) .
++JCLIN RELFILE(1).
```

This is the System Modification Program install logic for GDDM/MVS NLS US English. The other languages are identical except for the ++FUNCTION FMID. Japanese (Kanji) and Simplified Chinese have a FILES value of 5.

```
++FUNCTION(JGD3219) FILES(4) FESN(0565486)
                    REWORK(1996103) RFDSNPFX(IBM)
/* COPYRIGHT;
         5695-167 (C) COPYRIGHT IBM CORP. 1979, 1996.
         ALL RIGHTS RESERVED
         LICENSED MATERIALS - PROGRAM PROPERTY OF IBM

         US Government Users Restricted Rights -
         Use, duplication or disclosure restricted
         by GSA ADP Schedule Contract with IBM Corp.
 */.
++VER(C150) FMID(HGD3200).
++VER(Z038) FMID(HGD3200).
++VER(P115) FMID(HGD3200).
++JCLIN RELFILE(1).
```

Appendix B. JCLIN for GDDM/MVS

The JCLIN for GDDM/MVS is supplied as member ADM3200 in the SADMSAM data set.

The JCLIN for each GDDM/MVS NLS language is also supplied in the SADMSAM data set with a member name ADMxxxx, where xxxx are the figures of the corresponding FMID. The GDDM/MVS NLS JCLIN is only available for the languages you have installed.

Appendix C. GDDM/MVS National language support

The GDDM/MVS national language support lets you use one or more national languages for GDDM/MVS, GDDM-PGF, GDDM-IVU, and GDDM-GKS. National language support for all these GDDM licensed programs is shipped on the GDDM/MVS tape.

You must install the national language support for at least one language. You are recommended to do this after you have installed GDDM/MVS and before you install any other GDDM licensed programs, or the verification steps after installation will fail.

National language support needs to be installed once only. It establishes any or all of the following:

- The language of messages for GDDM/MVS
- The language of messages for GDDM-REXX/MVS
- The language of the ICU messages and panels for GDDM-PGF
- The language of messages and panels for GDDM-IVU
- The language of messages for GDDM-GKS

The languages in Table 24 on page 88 are available and can be selected when installing GDDM/MVS NLS support.

Table 24. National language support for GDDM

Language	Letter	GDDM	REXX	PCLK	OS/2 Link
U.S. English	A	Y	Y	Y	Y
Brazilian (Portuguese)	B	Y	Y	Y	Y
Simplified Chinese (PRC, DBCS)	C	Y	Y	—	—
Danish	D	Y	Y	—	Y
French	F	Y	X	—	Y
German	G	Y	Y	Y	Y
Hangeul (Korean, DBCS)	H	Y	Y	—	—
Italian	I	Y	Y	Y	Y
Kanji (Japanese, DBCS)	K	Y	Y	—	Y
Norwegian	N	Y	Y	—	Y
French (Canadian)	Q	Y	X	—	Y
Spanish	S	Y	Y	Y	Y
Traditional Chinese (Taiwan, DBCS)	T	X	X	—	—
Swedish	V	Y	Y	—	Y

Notes:

Y means language file is available
X means file exists, but text is not translated
— means file does not exist
DBCS means double-byte character set.
PRC means People's Republic of China.

Notes:

1. Japanese language support includes the Kanji symbol sets.
2. PRC Chinese (Simplified) language support includes the Simplified Chinese vector symbol sets.
3. If you are planning to install a DBCS national language at your enterprise, you must also install U.S. English.
4. National language support is available for GDDM-PCLK and GDDM-OS/2 Link on personal computer systems at your enterprise. You must have at least one national language installed on the host, but you do not need to have the same languages installed on the host as you have selected for GDDM-PCLK and GDDM-OS/2 Link.
5. The GDDM-IMD licensed program does not have national language support. All messages are in U.S. English only, and are available without installing GDDM/MVS NLS for U.S. English.

6. National Language Support for the GDDM-PGF OPS Utility is supplied as part of GDDM-PGF. See the *GDDM-PGF OPS User's Guide* for information about OPS National Language Support.

Sending your comments to IBM

GDDM

GDDM/MVS Program Directory

GC33-1801-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

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.

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.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form
- By fax:
  - From outside the U.K., after your international access code use 44 1962 870229
  - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Readers' Comments

GDDM

GDDM/MVS Program Directory

GC33-1801-01

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	
Telephone	Email

You can send your comments POST FREE on this form from any one of these countries:

Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

1 Cut along this line

2 Fold along this line

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO

NE PAS AFFRANCHIR

NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____

Company or Organization _____

Address _____

_____

EMAIL _____

Telephone _____

1 Cut along this line

4 Fasten here with adhesive tape

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC33-1801-01