

GDDM-IVU



Image View Utility

Version 1 Release 1.3

GDDM-IVU



Image View Utility

Version 1 Release 1.3

Note!

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

Third Edition (September 1996)

- | This edition applies to Version 1 Release 1 Modification 3 of the IBM licensed program GDDM Image View Utility, program number 5668-723, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable IBM system bibliography for current information on this product.

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

At the back of this publication is a page titled “Sending your comments to IBM”. If you want to make comments, but the methods described are not available to you, please address them 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.

© Copyright International Business Machines Corporation 1987, 1996. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Sample programs	vii
Programming interface information	vii
Trademarks and service marks	viii
 Preface	 ix
Who this book is for	ix
What you need to know to understand this book	ix
How to use this book	ix
Latest GDDM information	x
GDDM publications	xi
Related publications	xii
 Summary of changes	 xiii
Summary of changes for GDDM 3.1	xiii
 Chapter 1. Getting started with GDDM-IVU	 1
What is an image?	1
What is GDDM-IVU?	3
Before you start	4
Starting and ending a GDDM-IVU session	4
Ending a session	5
Special keys	6
PF keys on all types of terminal	6
Special keys on the 3193	8
What you can do with GDDM-IVU	10
Loading a GDDM image	10
Viewing an image	13
Scanning a document	16
Saving a GDDM image	18
Discarding an image	19
Printing an image	20
Nickname file for graphics printers	21
Loading an image from a non-GDDM image file (importing)	22
How to import an image	23
Saving an image in a non-GDDM image file (exporting)	25
 Chapter 2. Editing images	 27
The edit functions	27
How to edit an image	28
Projections	34
Creating a projection	35
Saving a projection	41
Loading a projection	42
Discarding a projection	43
Test page	45
 Chapter 3. Application programming	 47
An introduction to the GDDM-IVU calls	48
General application-programming guidance	49

GDDM-IVU call syntax	50
Compiling and running GDDM-IVU programs	67
Appendix A. GDDM-IVU calls	69
IUAIMG	70
IUAPRJ	71
IUDIMG	72
IUDPRJ	72
IUINIT	73
IUIVU	74
IUMLO	75
IUPFK	76
IUQIID	78
IUQINM	78
IUQLO	79
IUQMLO	80
IUQNLO	80
IUQPFK	81
IUQPID	82
IUQPNM	82
IUTERM	83
Appendix B. Sample programs	85
What the programs do	85
View an image using Personal Services/370	85
Put image from GDDM-IVU into DISOSS or Personal Services/370	86
How to install the programs	86
How the programs work	87
ADMU5DP	87
ADMU5DD	87
ADMU5DG	88
ADMU5DC	88
List of modules	88
Production environment	89
How to make changes	89
What hardware and software you need	90
Appendix C. Listing and managing images and projections in disk files	91
Under CMS	91
Under TSO	91
Under CICS	92
Appendix D. Editing GDDM-IVU panels	93
In the MVS environment	94
In the VM environment	94
In the VSE environment	94
Glossary	95
Index	111

Figures

1. A line image	2
2. Halftone image and enlarged section	3
3. Special keys on the 3193	8
4. GDDM-IVU sample image ADMU5IMG	15
5. Effect of sample projection ADMU5PRJ	35

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 is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM 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, is 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.

Sample programs

This publication contains sample programs. Permission is hereby granted to copy and store the sample programs into a data-processing machine and to use the stored copies for internal study and instruction only. No permission is granted to use the sample programs for any other purpose.

Programming interface information

This book is intended to help you use GDDM-IVU interactively. This book also documents General-use Programming Interface and Associated Guidance Information provided by GDDM-IVU.

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

General-use Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

_____ General-use programming interface _____

General-use Programming Interface and Associated Guidance Information...

_____ End of General-use programming interface _____

Trademarks and service marks

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

APL2 CICS GDDM IBM System/370

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Microsoft, Windows and the Windows 95 Logo are trademarks or registered trademarks of Microsoft Corporation.

Preface

The GDDM Image View Utility (GDDM-IVU) is an IBM licensed program that adds image-handling capabilities to IBM data-processing systems.

This book is intended to help you use GDDM-IVU interactively, and to use its application-programming interface.

Who this book is for

This book is intended for end users of GDDM-IVU and for application programmers who intend to use the GDDM-IVU application programming interface.

What you need to know to understand this book

End users need no programming or other data-processing expertise, but they do need some end-use experience on data-processing systems. They should know, for example, how to use a terminal, how to log on to the system under which GDDM-IVU runs, and be able to understand the major differences between main storage and disk files.

Application programmers may find a knowledge of GDDM Base calls useful, but the only essential requirement is programming experience in one of the supported languages (FORTRAN, COBOL, PL/I, the C part of the C/C++ product, C/370, H Assembler, High Level Assembler, REXX, APL2, or BASIC).

How to use this book

There is no need to read this book sequentially. You can use the contents list to select the chapters or sections that interest you.

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.

Related publications

DISOSS	<i>DISOSS Application Interface: Programming Guidelines</i> , GG24-1614 <i>DISOSS Application Programming</i> , SC30-3096 <i>DISOSS Interchange Architecture Reference</i> , SC30-3276
3117 Scanner	<i>IBM 3117 Scanner and IBM 3117 PC Adapter Guide to Operations</i> , GA18-2477 <i>IBM 3117 Scanner and Extension Unit Guide to Operations</i> , GA18-2478 <i>IBM 3117 Scanner Hardware Maintenance and Service</i> , SY18-2159 <i>IBM 3117 Scanner Technical Reference</i> , SC18-2105
3118 Scanner	<i>Scanner Guide to Operations</i> , GA18-2475 <i>High Speed Adaptor Guide to Operations</i> , GA18-2476 <i>IBM 3118 Scanner Hardware Maintenance and Service</i> , SY18-2158 <i>High Speed Adapter Hardware Maintenance and Service</i> , SY18-2167 <i>Scanner Technical Reference</i> , SC18-2104 <i>High Speed Adapter Technical Reference</i> , SC18-2177
3193 Display	<i>Description</i> , GA18-2364 <i>Setup Instructions</i> , GA18-2366 <i>Operator's Guide</i> , GA18-2365 <i>Problem Solving Quick Check Guide</i> , GA18-2443 <i>Problem Solving Guide</i> , GA18-2444

Summary of changes

GDDM-IVU can import and export images, under TSO, as members of a partitioned data set. See “Loading an image from a non-GDDM image file (importing)” on page 22 and “Saving an image in a non-GDDM image file (exporting)” on page 25.

Appendix D, “Editing GDDM-IVU panels” on page 93 has been moved into this book from the *GDDM System Customization and Administration* book.

Summary of changes for GDDM 3.1

Installation instructions for GDDM-IVU are provided in the program directory that accompanies the GDDM product tapes. Installation-planning information can be found in the *General Information* book for your system.

GDDM-IVU messages are described in the *GDDM Messages* book.

Diagnosis of GDDM-IVU is described in the *GDDM Diagnosis* book.

The structure of this book has been altered to reflect the removal of GDDM-IVU installation, diagnosis, and messages information. In addition, C/370 and REXX have been added to the list of supported programming languages for the GDDM-IVU programming interface. This enhancement has been incorporated as appropriate throughout this manual.

Chapter 1. Getting started with GDDM-IVU

What is an image?

Most data is entered into computers by typing in characters. Often these are words and numbers copied from documents such as application forms or invoices. The characters are called alphanumeric data. Computer output, too, is commonly presented in alphanumeric form. However, alphanumeric data entry is laborious, and is impossible for some types of information. A drawing or a signature, for example, can be neither entered into the computer nor represented in computer output as alphanumeric data. GDDM-IVU solves these problems by replacing alphanumeric data with electronic facsimiles, or *images*, of documents.

A computer image is a picture composed of an array of elements called *pixels*. The illustrations in the rest of this book contain printed examples of images. When printed, each pixel is either a small patch of ink or a small area of blank paper. When shown on a display terminal, each pixel appears as either a point of light or a blank area of screen. But the same techniques are used for printing and displaying alphanumeric data, so what is special about images?

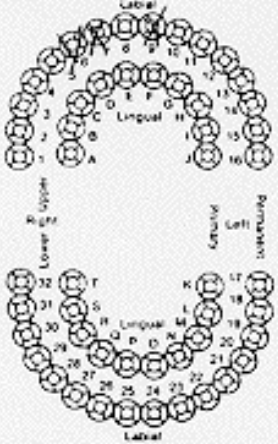
The answer lies in the way images are held in the computer's main storage and on disk files. In principle, each pixel in a GDDM image is stored as one *bit*, which is the smallest unit of electronic data. A bit value of 1 (or 'on') represents a point of light on a screen, and 0 (or 'off') a blank area of screen.

Most images are generated by passing documents through a device called a *scanner*. The scanner senses what is on the paper by measuring reflected light. Each pixel-sized area is individually measured. The measurements are converted into bits – a '1' if the intensity of light is above a certain level and a '0' if it is below. The bits are transmitted to the computer's main storage and, usually, from there to a disk file.

GDDM-IVU images are *bilevel*, meaning each pixel must be either on or off. Neither color nor varying levels of intensity (*gray-levels*, as they are known) are recorded. Gray-levels can be simulated, however, by *halftoning*: using groups of adjacent pixels to represent shades. In business, most images are of documents containing printing, typescript, handwriting, and sometimes line drawings. These types of subject – called line documents – are well suited to storage as bilevel images. Figure 1 on page 2 shows an example of such an image. Continuous-tone pictures such as photographs decline in quality when converted to images, but the results are satisfactory for many purposes. Figure 2 on page 3 shows an example of a continuous-tone picture converted to an image.

Attending Dentist's Section

Patient's Name Michael Public		If any of the treatment is for Orthodontic purposes, complete both treatment and orthodontic sections below	
Dentist's Name D.D. Teeth		Specialty Bridges	Individual Practitioner S.S. No. 999-99-9999
Dentist's Mailing Address (Street) 1010 Infinity		Dentist's Telephone No. 999-999-9999	All others - Employer ID No.
City Goodwill	State and Zip Code XT 000000	Date of first visit - in current series 2/2/83	Date of in-hospital dentist treatment
If prosthesis is this initial placement? ☐ Yes (If no, reason for replacement) ☐ No		Date of initial placement, this prosthesis	Check if estimate is required ☒



Indicate missing teeth with an 'X'

Remarks:

Tooth # Or Letter	Surfaces	Description of Service (including x-rays, prophylaxis materials used, etc.)	Date Service Performed			ADA Procedure Number	Fee by Procedure	For Ins. Co. Use Only
			Mo	Day	Yr			
6	MD	Full Service	2	2	3	2720	210	
7	DO	Gold gold Cr. Ryl	2	9	3	"	165	
9		Gold gold Cr. Ryl	3	1	3	1700	90	
		Prophylaxis	3	1	3	1110	14	

Orthodontics: (Give diagnosis, class of malocclusion and denture appliance(s) in above treatment section). Normally one completed appliance is adequate during the full course of treatment in order to process benefit payments.

Active Treatment

Date such treatment started

Date first appliance (to be) inserted

Estimated total active treatment period (no. of months)

Total fee for active treatment (including cost of appliances)

Total fee charged \$ **479**

By my signature below, I certify that the above statements and all other information provided by me pertaining to this claim are correct. I also agree that IBM, Metropolitan or their representatives may obtain or view a copy of the records pertaining to the examination, treatment, history and expenses connected with this claim.

Dentist's Signature **D.D. Teeth**

Date **3-9-83**

Note: Benefits will be paid directly to you unless the employee has indicated otherwise on the other side of this form. Please mail completed form to the address shown on the front of this form.

Figure 1. A line image

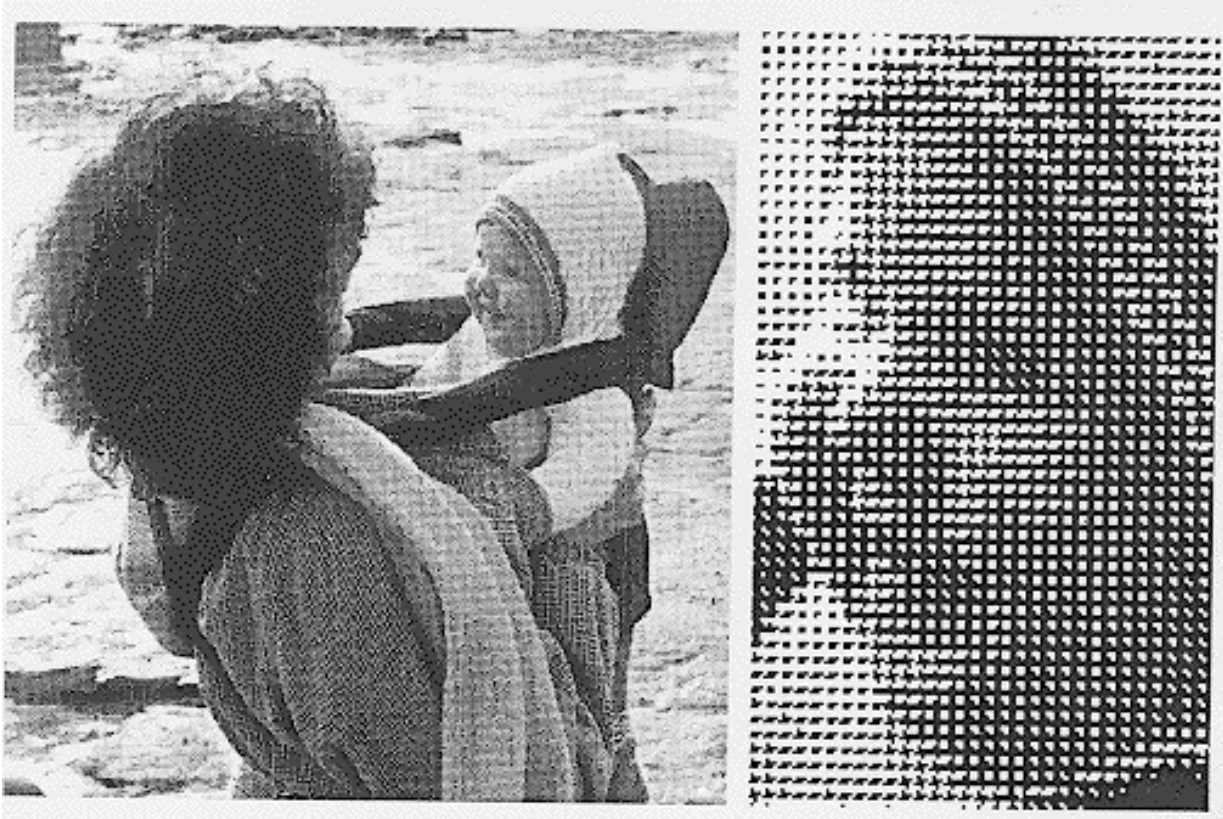


Figure 2. Halftone image and enlarged section. The enlarged section shows the image structure.

What is GDDM-IVU?

GDDM-IVU is a program that is run from a terminal to:

- Create images by scanning documents
- Save images in disk files
- View images on the terminal screen
- Edit images in various ways, such as altering their size or merging multiple images
- Create image output files for printers

GDDM-IVU is simple to use. No special knowledge of computers is required, although experience of computer terminals, personal computers, or word processors will probably help you get started quicker.

GDDM-IVU is interactive. This means that everything you tell it to do is carried out immediately. Instructions are passed to it by typing into menus, by pressing program function (PF) keys, and by indicating positions in images on the screen with a cursor.

GDDM-IVU has an application programming interface (API). Programmers can call GDDM-IVU from other programs and they can customize it, which enables them to add image capabilities to existing applications.

GDDM-IVU runs on many types of terminal. It works best on an IBM 3193 Display, because this has special usability and performance functions for images, and gives the highest-quality image display. GDDM-IVU also runs well on graphics displays, such as the IBM 3179-G and IBM 3279, and on personal-computer systems running the GDDM-OS/2 Link or GDDM-PCLK component of GDDM Base.

GDDM-IVU supports many devices in addition to display terminals. For scanning, a 3117 or 3118 scanner is attached to a 3193. For printing, output files can be created for image printers, graphics printers, and advanced-function printers designed for publications work.

Before you start

Before you can start your first GDDM-IVU session, you need to know a few things about the computer system you will be using. You may need to find them out from your computer-support personnel.

Firstly, you must know the name of the subsystem under which you will be using GDDM-IVU: will it be CMS, TSO, or CICS?

Secondly, you must know how to log on to the system.

Thirdly, you should enquire whether your computer-support personnel have provided any special way of starting a GDDM-IVU session. For TSO and CICS users, additional definition steps are required before GDDM-IVU can be run. Refer to the section on post-installation tasks in the *GDDM/MVS Program Directory*.

Finally, it is advisable to check whether any programming has been done to amend the way GDDM-IVU operates. This manual describes the standard program as supplied by IBM. If your computer-support personnel have done any programming with GDDM-IVU, they should explain any changes that affect the descriptions in this chapter.

If you do talk to computer-support personnel, you could take the opportunity to enquire about customization (also called setup). GDDM can be customized, and so can the terminal if it is an IBM 3193 Display Station. Customization is highly desirable though not absolutely essential. You can therefore leave GDDM and the terminal uncustomized, at least for your introductory sessions. Alternatively, you may be able to do it yourself.

A 3193 terminal is customized using a special key on its keyboard. GDDM is customized using instructions called *nickname statements*. The processes are described in the *GDDM System Customization and Administration* book.

When you are ready to start using GDDM-IVU, read the next section.

Starting and ending a GDDM-IVU session

Log on to your computer system, using an IBM 3193 terminal if possible. If your computer-support personnel have provided a special way of starting a GDDM-IVU session, follow their instructions. Otherwise, follow the standard method for the subsystem you are using. (On CICS and TSO, this assumes that subsystem customization has been completed as described in the *GDDM/MVS Program Directory*.)

- CMS users type ADMIVU and press ENTER
- CICS users type a transaction name of ADMU and press ENTER
- TSO users type EXEC ADMIVU and press ENTER

You should then see the GDDM-IVU Home panel, which is reproduced below.

```

----- Home -----
                        IMAGE VIEW UTILITY

Selection:              Choose a process, then press ENTER

                        1 = Scan      Operate the scanner
                        2 = View      Look at an image
                        3 = Edit      Edit an image
                        4 = Print     Print an image
                        5 = Save      Save or discard an image

GDDM Image View Utility 5668-723 (C) COPYRIGHT IBM CORP. 1987,1996
-----
PF 1 Help      2 No Menu  3 Exit    4 Input    5 Projectn  6 Output
PF 7 Backward  8 Forward  9 Clear  10         11         12

```

You will come across this panel often. It is used to select major functions of GDDM-IVU, not only when you start GDDM-IVU but throughout the session.

If you see this message across the bottom of the menu:

```
ADM4000 W CHANGING TERMINAL SETUP CAN IMPROVE GDDM-IVU PERFORMANCE
```

your terminal has not been correctly customized. GDDM-IVU will still be usable, however.

Like all GDDM-IVU menu panels, the Home panel is displayed in the lower part of the screen. The upper partition is used by GDDM-IVU to display images. Alternatively, GDDM-IVU can put help information in the upper partition if you request it. Initially it is blank.

This illustration, and others in this manual, represent what you should see on an IBM 3193. On other types of terminal, the proportions of the screen will differ but their contents will be the same.

You direct the actions of GDDM-IVU in two ways. One is by typing into menus like this one and then pressing ENTER. The other is by pressing PF (program function) keys.

The PF key actions are shown along the bottom of the screen. If GDDM-IVU needs to display a message (for example, if you make a mistake) it overwrites the dashed line above the PF key descriptions.

Ending a session

When the Home panel is on the screen, you can end the session by pressing the Exit PF key, and then pressing it a second time when prompted.

The remainder of this chapter contains instructions for displaying images in the upper partition, and for doing many other useful things with GDDM-IVU. Following them through will introduce you to most of the capabilities of GDDM-IVU. However,

you do not have to read it all – you can pick out the operations that interest you and skip the rest. GDDM-IVU is not hard to use, so you may even get by without reading any more at all.

Special keys

The information in this section is useful, but not essential. If you are in a hurry to try out GDDM-IVU, you can note the points in the box below and read the information from there through page 9 later.

PF keys on all types of terminal

The PF keys provide a set of commonly needed functions.¹ You can try out the following keys even before putting any images on the screen: Help (PF1); No menu/Menu (PF2); the scrolling keys (PF7 and PF8) – for these, first press Help (PF1) to display some scrollable help information.

On 3193 terminals, the alphanumeric cursor must be in the lower partition before a PF key function can be used.

Some things to remember about PF keys

- If you get lost during a session, you can press Home (PF12) at almost any time and get back to the Home panel to start again.
- Help information is displayed when you press Help (PF1) and removed when you press Return (PF3).
- If you press Help (PF1), you will be unable to see any images on the screen until you press Return (PF3) to remove the help information.
- The menu is removed when you press No menu (PF2) and redisplayed when you press PF2 again. On 3193 terminals, the Partition enlarge key shown in Figure 3 on page 8 has a similar effect and is quicker.

PF key	Function	Description
PF1	Help	For each menu panel, there is a help panel. This gives instructions on using the menu. Looking up the help information online is often more convenient than turning to this book. Furthermore, some help panels contain additional information about the menus. Pressing this key puts the help information into the upper partition, temporarily replacing any images on display. The help text is usually deeper than the partition, so you may need to use the scroll keys to find the information you need. Pressing Return (PF3) removes the help information. Any images originally displayed in the upper partition are restored.

¹ The PF key numbers shown in this book indicate the standard GDDM-IVU usage. You may find that your computer support personnel have customized GDDM-IVU by assigning the functions to different PF keys or removing some of functions. The PF key area along the bottom of the each menu panel indicates the PF key functions actually available to you and the keys to which they are assigned.

PF key	Function	Description
PF2	No menu or Menu	This removes the menu, allowing the upper partition to occupy the whole screen, apart from the PF key information area and the message line. This key has a toggle action – pressing it a second time restores the menu. On IBM 3193 terminals, the Partition enlarge key has a similar effect to this key and is quicker.
PF3	Return or Exit	This returns to the previous menu panel, except when the Home panel is on the screen or a help panel is being displayed. When the Home panel is on the screen, it ends the GDDM-IVU session after being pressed twice. When a help panel is being displayed, it removes the help and allows images to be displayed.
PF4	Input	This displays the panel for loading images from files and scanning documents. More information is given later in this chapter.
PF5	Projectn or Undo	This displays the panel for operations on projections, except when you are using the Edit function, in which case it undoes the previous editing action. More information about projections and editing is given later in this chapter.
PF6	Output	This displays the panel for saving images as files, printing them, and discarding them. More information is given later in this chapter.
PF7	Backward	This scrolls the upper partition backwards. It brings into view parts of the display that are conceptually off the top of the screen. On the IBM 3193, the Backward scroll key has a similar effect, and is quicker.
PF8	Forward	This scrolls the upper partition forward. It brings into view information that is conceptually off the bottom of the partition. On the IBM 3193, the Forward scroll key achieves a similar result, and is quicker.
PF9	Clear	This clears all images from the screen. On the IBM 3193, the Clear and Partition clear keys have a similar effect. The Partition clear key is the quickest.
PF10	No effect	
PF11	No effect	
PF12	Home	This displays the Home panel, so that you can either initiate some new action or end the GDDM-IVU session.

On terminals (such as the 3193) with more than twelve PF keys, PF13 through PF24 duplicate the functions of PF1 through PF12. On terminals with more than 24 PF keys, PF25 and above have no effect.

Special keys on the 3193

The 3193 terminal has several keys that are especially useful with GDDM-IVU.

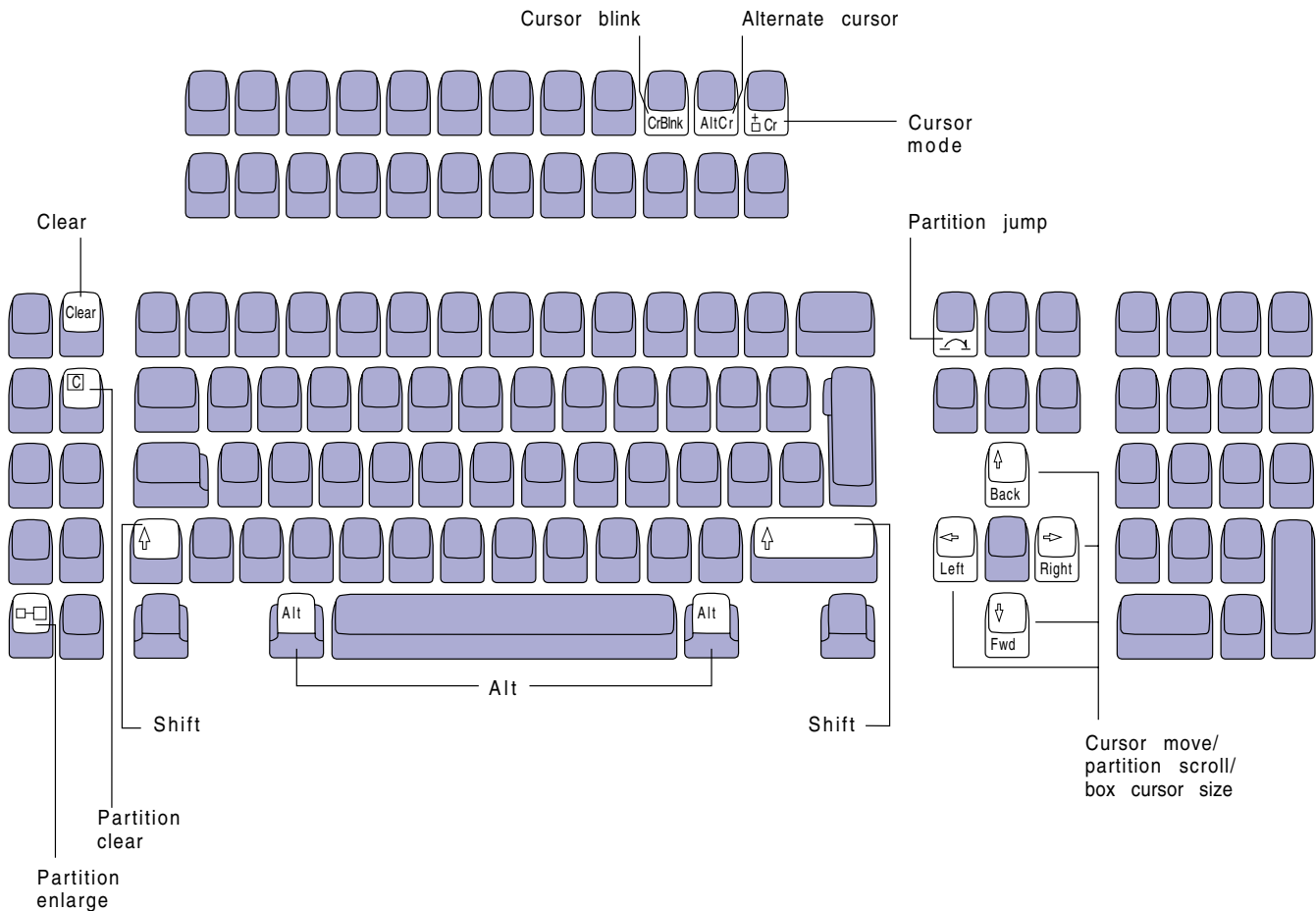


Figure 3. Special keys on the 3193

Cursor mode	This key (which is also the PF24 key) used with the Alt key activates the special image cross and box cursors. More information is given in "Cursors on 3193 terminals" on page 29.
Alternate cursor	This key (which is also the PF23 key) used with the Alt key swaps the appearance of the ordinary alphanumeric cursor between an underscore and a solid rectangle, and the cross and box cursors between dark and light.
Cursor blink	This key (which is also the PF22 key) used with the Alt key starts and stops cursors blinking. It affects the ordinary alphanumeric cursor and the cross and box cursors.

Cursor move	The keys inscribed with arrows move the cursor. Two of them also scroll the upper partition. If you put the cursor in the upper partition and then hold down the Alt key, the Back and Fwd keys scroll the image or help text. These keys are faster than Backward (PF7) and Forward (PF8). However, GDDM-IVU is not informed of the scrolling, and this may produce unexpected results. To avoid such results, always scroll back again before using any GDDM-IVU functions, especially the PF scrolling keys, and the editing and projection creation functions (described later in this chapter). The Cursor move keys also control the special image box cursor, as described in “Cursors on 3193 terminals” on page 29.
Partition jump	This makes the ordinary alphanumeric cursor jump between the lower and upper partitions. It does not work on the box and cross cursors.
Partition enlarge	This key enlarges either partition – whichever contains the cursor – to fill the screen. There is, however, no benefit in enlarging the lower partition. Enlarging the upper partition makes the image or help information fill the screen. It is similar in effect to the Menu/No menu (PF2) key, but quicker. It differs in that it removes the PF key information and any message, as well as the menu, from the screen. Pressing the partition enlarge key a second time restores the menu, the PF key information, and any message to the screen. You are advised to do this as soon as you have finished with the full-screen image; otherwise you may have difficulty keeping track of GDDM-IVU's actions. If you enlarge the upper partition and then select another function by pressing a PF key, the image or help information disappears, and the menu for the newly selected function is displayed in the middle of the screen. To restore the normal two-partition layout, press the Partition enlarge key again.
Clear	This removes images from the upper partition, leaving it blank. The lower partition remains unchanged. It is similar to Clear (PF9) PF clear.
Partition clear	If the cursor is positioned in the upper partition, this key removes images from that partition, leaving it blank, like the Clear (PF9) PF key. The partition clear key is the quickest way of clearing images from the screen. It leaves the screen unchanged when the cursor is in the lower partition.

What you can do with GDDM-IVU

The following sections give step-by-step instructions for using GDDM-IVU. Each explains how to perform one major operation with an image.

- “Loading a GDDM image” explains how to copy an image from a GDDM image file on disk storage into GDDM-IVU storage.
- “Viewing an image” on page 13 explains how to display an image on the screen of your terminal.
- “Scanning a document” on page 16 explains how to use a scanner to create an image of a document.
- “Saving a GDDM image” on page 18 explains how to copy an image from GDDM-IVU storage to a GDDM image file on disk.
- “Discarding an image” on page 19 explains how to remove an image from GDDM-IVU storage.
- “Printing an image” on page 20 explains how to print an image held in GDDM-IVU storage.

Loading a GDDM image

Before you can do anything with an image, it must be held in GDDM-IVU storage. If the image you require is stored on disk in a GDDM image file, it must first be loaded into GDDM-IVU storage in the following way. If it is stored on disk but not in a GDDM image file, it must be imported, as described in “Loading an image from a non-GDDM image file (importing)” on page 22.

Under CICS, only one image can be held in GDDM-IVU storage at a time. So if your subsystem is CICS, and you have already loaded or scanned an image, you will normally need to save it and discard it (see page 19) before you can load another.²

1. Press Input (PF4). You should then see the Input Selection panel:

```

----- Input Selection -----

Input image:          Name of the image to be obtained

Selection: 1          Choose an input method, then press ENTER
                        1 = Load image in GDDM format
                        2 = Scan with an image scanner
                        3 = Import image from a non-GDDM format file

Load options: 0        0 = Use default options
                        1 = Use current options
                        2 = Set options

-----
PF 1 Help    2 No Menu  3 Return  4          5 Projectn  6 Output
PF 7 Backward 8 Forward  9 Clear  10         11         12 Home

```

2. On the first line, against Input image, type the name of a GDDM image stored in a disk file. The sample image supplied with GDDM-IVU is suitable for trial

² It is possible to write a CICS transaction that sets the image limit higher than “1” using the IUMLO call. If your system-support personnel have provided such a transaction, you are not subject to this limitation.

purposes. Its name is ADMU5IMG (unless it has been renamed by your system-support personnel). It is illustrated in Figure 4 on page 15.

3. Leave the value 1 in the Selection field. Values 2 and 3 are for scanning documents, and for loading images from files other than GDDM image files. More information about these selections is given in later sections.

4. You can leave the initial value of 0 in the Load options field.

A value of 2 causes the Load Image panel to be displayed for specifying more about the loading process. This is described on page 12. A value of 1 reuses the parameters previously entered on that panel.

5. Press ENTER. GDDM-IVU loads the image. However, some images may cause a further panel, the Load Resolution panel, to appear. This panel is described below.

6. GDDM-IVU displays the image as soon as it is loaded, unless you have specified otherwise using a Load Options value of 1 or 2 in the Input Selection panel, and a value of 1 in the Echo field of the Load Image panel. In that case, you can display it by following the instructions in “Viewing an image” on page 13.

Remember that help information obscures the image; to remove it, press Return (PF3).

GDDM images stored on disk: They are sometimes called GDDM image objects. Under CMS, they have a filetype of ADMIMG. Under TSO, they reside in a partitioned data set that will be allocated with the DD name ADMIMG. Under CICS, they reside in either data set ADMF or in a private VSAM data set. Instructions for listing the names of GDDM images stored on disk are given on page 91. GDDM images in disk files have some information stored with them in addition to their bit patterns. This information usually includes the resolution – the number of pixels per inch or meter. Some GDDM images (though not those saved by GDDM-IVU) can be stored without this information. If you try to load an image that was saved without a resolution, GDDM-IVU will display this panel and message to enable you to specify the resolution.

----- Load Resolution -----

Resolution Horizontal: 0 Vertical: 0	Choose the load resolution 0 = Same resolution as the display 1 = 120 pixels per inch 2 = 200 pixels per inch 3 = 240 pixels per inch
---	--

ADM4068 W LOADED IMAGE HAS NO RESOLUTION. CHOOSE A RESOLUTION FOR IT

PF 1 Help	2 No Menu	3 Return	4
PF 7 Backward	8 Forward	9 Clear	10

	5	6	
	11	12 Home	

1. Select a horizontal and vertical resolution by leaving or overtyping the initial values in the Horizontal and Vertical fields. Normally, the two should be the same.

loading an image

The appropriate values to select depend on the purpose for which you are loading the image. There are two considerations: the device on which it will be presented and the required size.

For best visual quality, the resolution should be the same as that of the screen or printer on which you intend to present the image. If you intend just to view it, you can optimize quality by entering 0 values.

On the other hand, the resolution affects the size of the image – 120 pixels per inch, for instance, will give an image twice as big as 240 pixels per inch. If the image was created by scanning a document and you want to present it at same size as the original, the selected resolution must be the same as the scanning resolution. This can be either 120 or 240 pixels per inch (normal or fine resolution) in the case of the 3117 and 3118 scanners, or 200 pixels per inch in the case of the IBM 8815 Scanmaster.

2. Press ENTER. GDDM-IVU loads the image and redisplay the Input Selection panel. The Load options field in the Input Selection panel shown on page 10 is for specifying more about the loading process. If you type a value of 2 and press ENTER (after typing a name against Input image), the image is not loaded: instead, you will see the Load Image panel:

```
----- Load Image -----
Load image: ADMU5IMG      Name of the image to be loaded
File name:                External name of image (if different from above)
Projection:               Name of a projection to be applied (optional)

Protect: 0                0 = Existing image must not be overwritten
                        1 = Existing image can be overwritten
Echo: 0                   0 = Display the loaded image
                        1 = Do not display the loaded image
Description:

-----
PF 1 Help      2 No Menu  3 Return  4          5 Projectn  6 Output
PF 7 Backward  8 Forward  9 Clear  10         11          12 Home
```

1. Notice that the Load image field is already initialized with the name you entered on the Input Selection panel. GDDM-IVU usually initializes name fields with the name most recently entered.
2. You can leave the File name field blank. Information about when to fill out this field is given in “Names of images” on page 13.
3. You can leave the Projection field blank. More information is given in Chapter 2, “Editing images” on page 27.
4. If an image with the name you have chosen is already present in GDDM-IVU storage, you can replace it with the new image by setting the value of the Protect parameter to 1. A value of 0 gives protection against accidentally destroying an existing image if its name is specified by mistake.
5. A 0 in the Echo field causes the image to appear on the screen as soon as it is loaded. If you do not want this to happen, set the value to 1.
6. You cannot enter anything against Description. After loading the image, GDDM-IVU may display a description of it here. This would have been

supplied when the image was saved on disk. If no description was supplied, GDDM-IVU leaves this field blank.

7. Press ENTER to load the image. If echoing was selected, the image appears on the screen.

Names of images: All images held in GDDM-IVU storage must have names. The names must follow the GDDM-IVU image-naming convention. A safe standard naming convention is to always start with a letter and follow it with up to seven letters or numbers.

When loading an image, it is simplest to keep the internal GDDM-IVU name the same as the external name of the image on disk. Entering this external name in the Load image field and leaving the File name field blank ensures this. However, there may be some circumstances when you want the names to be different. In this case, enter your chosen internal GDDM-IVU name in the Load image field, and the name of the image on disk in the File name field.

The GDDM-IVU internal naming rules are actually the same as the rules that the subsystem (TSO, CMS, or CICS) has for naming disk files (or partitioned data set members in the case of TSO and CICS). Whether a particular name is valid therefore depends on which subsystem is in use. However, a letter followed by up to seven letters or numbers is valid on all subsystems.

The same naming rules apply to projections (described in Chapter 2, “Editing images” on page 27).

Viewing an image

1. On the Home panel (remember Home (PF12) always takes you to the Home panel) type 2 against Selection, and press ENTER. You should then see the View Image panel:

```

----- View Image -----
View image:      Name of the image to be viewed
Projection:      Name of a projection to be applied (optional)
Retain: 0        0 = Clear the image display area
                  1 = Preserve the image display area

-----
PF 1 Help      2 No Menu    3 Return    4 Input    5 Projectn  6 Output
PF 7 Backward  8 Forward    9 Clear    10         11         12 Home
  
```

2. Ensure that the name of the image you want to view is entered in the View image field. The GDDM-IVU sample image, standard name ADMU5IMG, is suitable for a test of the viewing function. If viewing is not the first thing you are doing in this GDDM-IVU session – if, for example, you have just loaded an image – this field will probably already have an image name in it. In that case, you can leave it or overwrite it with the name of another image.
3. You can leave the Retain field set to 0. Its purpose is to help you deal with several images at once. The value 0 clears the screen of images before the

next one is displayed. The value 1 causes the next image to overwrite the required area of the screen without first clearing.

4. You can leave the Projection field blank. More information is given in Chapter 2, "Editing images" on page 27.
5. Press ENTER. Remember that help information obscures the image; to remove it, press Return (PF3).

The image is displayed in the upper partition if it is present in GDDM-IVU storage. If it is not present, the Input Selection panel shown on page 10 is displayed. The Input image field will have been initialized with the image name. Assuming that the image exists in a GDDM image file, all you need do is press ENTER again. GDDM-IVU loads and displays the image, replacing the Input Selection panel with the View Image panel.

Some notes about image sizes: The largest image you can view on a screen using GDDM-IVU is 209 × 298 millimeters (8.25 × 11.75 inches). This is also the standard size for any image that you create using the editing and projection function. The size of an image created by scanning is defined by the data you enter on the panel shown on page 17.

You can load images that are larger than the viewable maximum, but you can display only the top left 209 × 298-millimeter (8.25 × 11.75-inch) section.

On the 3193 terminal, images are displayed approximately 10% smaller than their actual size. On other devices, they are presented at their actual size.

JOB APPLICATION FORM	
1. Position applied for	Computer Programmer
2. Name	Geraldine Moore
3. Date of Birth	28 August 1966
4. Place of Birth	Hallfield, England
5. Home Address	"Chilans" Ringsham Way Winchampton
6. Telephone Number	W. 4433
7. ☐ Male ☒ Female	 Current photograph
8. ☐ Married ☒ Single	
9. Do you possess a valid driving licence? ☒ Yes ☐ No	
10. Educational Qualifications First Class degree in Image Manipulation from Oxbridge University, 1986.	
11. Present Employer Winchampton Software Consultants	
12. Recent Job Experience Working as an applications programmer in the computer department.	
13. Do you need relocation assistance? ☐ Yes ☒ No	
14. Can you work overtime and/or shift? ☒ Yes ☐ No	
15. Signature	Date
G. Moore	12 January 1987

Figure 4. GDDM-IVU sample image ADMU5IMG

Scanning a document

To scan a document, you need a 3193 terminal with a scanner attached. Make sure that the scanner is switched on. If it is not, end the GDDM-IVU session, switch on the scanner, and start a new GDDM-IVU session.

Scanning creates an image in GDDM-IVU storage, but does not store it on disk. To keep the image, you need to save it after scanning. This is described in “Saving a GDDM image” on page 18.

Under CICS, only one image can be held in GDDM-IVU storage at a time, (unless your system-support personnel have provided a locally written transaction that uses the IUMLO call to increase this limit before calling GDDM-IVU). If your subsystem is CICS, and you have already loaded or scanned an image, you will normally need to save it before discarding it (see page 19) so that you can scan another.

1. On the Home panel, select option 1. Alternatively, if you happen to have the Input Selection panel on the screen, choose a name for the image you are going to create and type it in the Input image field, then select option 2. In either case, when you press ENTER, you see the Scan Image panel:

----- Scan Image -----					
Scan image:		Name to be given to the scanned image			
Projection:		Name of a projection to be applied (optional)			
Scan control: 0		0 = Use default scan parameters 1 = Use current scan parameters 2 = Set scan parameters			
Protect: 0		0 = Existing image must not be overwritten 1 = Existing image can be overwritten			
Echo: 0		0 = Display the scanned image 1 = Do not display the scanned image			

PF 1	Help	2 No Menu	3 Return	4	5 Projectn 6 Output
PF 7	Backward	8 Forward	9 Clear	10	11 12 Home

2. The name you have chosen to give the image must be entered in the Scan image field. If GDDM-IVU has put a name into this field, you can leave it or overwrite it with a different one. It must be a valid GDDM-IVU image name: a letter followed by up to seven letters or numbers is valid on all systems.
3. You can leave the Projection field blank. More information is given in Chapter 2, “Editing images” on page 27.
4. You can leave the initial value of 0 in the Scan control field.
A value of 2 enables you to specify more about the loading process using the Scan Control panel. This is described on page 17. A value of 1 reuses the parameters previously entered on that panel.
5. If an image with the name you have chosen is already present in GDDM-IVU storage, you can replace it with the new image by setting the value of the Protect parameter to 1. A value of 0 will give protection against accidentally destroying an existing image if its name is specified by mistake.
6. If you leave a value of 0 in the Echo field, an image of the document appears on the screen as it is scanned. This process is known as “echoing.” Changing the value to 1 prevents echoing.

7. Press ENTER. GDDM-IVU displays a message telling you to put the document in the scanner. When you are ready for the scan to begin, press ENTER again. Alternatively, you can end the scan by pressing Return (PF3). To cancel the scan after it has started, press Clear on the terminal keyboard. GDDM-IVU displays a message when scanning is complete.
8. If you want the image to be stored on disk, save it as described on page 18. If the image is not saved, it is held in GDDM-IVU storage only and is discarded when you end the GDDM-IVU session. The Scan control field shown in the Scan Image panel on page 16 is for specifying more about the scanning process. If you type a value of 2 and press ENTER (after typing a name against Scan image), scanning is not started: instead, the Scan Control panel is displayed so that you can supply additional parameter values:

```

----- Scan Control -----

      Resolution          Choose the scanner resolution
      Horizontal: 1      1 = Normal
      Vertical: 1        2 = Fine

      Paper width: 8.5
      Paper height: 11.7
      Units: 1           1 = Inches
                        2 = Millimeters

-----

PF 1 Help      2 No Menu  3 Return  4          5 Projectn  6 Output
PF 7 Backward  8 Forward  9 Clear  10         11         12 Home
  
```

1. Choose the resolution of the image (that is, the density of the pixels) by entering 1 or 2 against Horizontal and Vertical. You can select fine resolution for the horizontal direction and normal resolution for the vertical (although not the reverse). Usually, the horizontal and vertical resolutions are set to the same value.

The scanner's normal resolution of 120 pixels per inch is suitable for most purposes. The fine resolution of 240 pixels per inch might be required for halftone images, or for images of line drawings. If the images are to be printed on an IBM 38xx page printer, the best quality is obtained if the scanning resolution is set to the same value as the printing resolution. On some of these printers, the resolution can be set to 120 or 240 pixels per inch.

The disadvantages of fine resolution are that scanning takes longer and the image is likely to occupy considerably more storage. Therefore, it is advisable to discover by experiment whether normal resolution will suffice.

2. The maximum size of document that you can scan depends on the scanner you are using. For the 3117 it is 8.5 × 11.7 inches (215 × 297 millimeters), and for the 3118, 8.5 × 14 inches (215 × 355 millimeters). If a document is actually smaller than the maximum, you can reduce the scan time and the storage required by specifying a smaller size on this panel. Enter the dimensions against Paper width and Paper height, and either a 1 or 2 against Units to indicate whether they are in inches or millimeters.
3. Press ENTER to return to the Scan Image panel.

Saving a GDDM image

Images held in GDDM-IVU storage can be copied into GDDM image files on disk, as follows.

1. Press Output (PF6). You should then see the Output Selection panel:

```

----- Output Selection -----

Output image:          Name of the image to be output

Selection: 1          Choose an output method, then press ENTER
                      1 = Save image in GDDM format
                      2 = Print the image
                      3 = Export image to a non-GDDM format file
                      4 = Discard the image

-----

PF 1 Help      2 No Menu    3 Return    4 Input    5 Projectn  6
PF 7 Backward  8 Forward    9 Clear    10         11         12 Home
  
```

2. On the first line, against Output image, type the name of an image currently held in GDDM-IVU storage. It can be an image that you have loaded, imported, created by scanning, or created from other images by editing.
3. Make sure that the value 1 is typed against Selection; the other values are explained later.
4. Press ENTER. You should see the Save Image panel:

```

----- Save Image -----

Save image: NEWIMG    Name of the image to be saved
File name:           External name of image (if different from above)
Projection:          Name of a projection to be applied (optional)

Protect: 0           0 = Existing file must not be overwritten
                   1 = Existing file can be overwritten

Description:

-----

PF 1 Help      2 No Menu    3 Return    4 Input    5 Projectn  6
PF 7 Backward  8 Forward    9 Clear    10         11         12 Home
  
```

5. The Save image field is initialized with the name you entered on the Output Selection panel. If you make a mistake, or if you change your mind about which image to save, overwrite this name with a new one.
6. You can leave the File name field blank. If you do, the name of the image on disk is the same as the internal GDDM-IVU name that you specify in the Save image field. Alternatively, you can specify a different name in the File name field. It must be a valid name for GDDM objects on disk: a letter followed by up to seven letters or numbers is valid on all systems. If you are unsure whether to specify a different name, the information in “Names of images” on page 13 may help.
7. You can leave the Projection field blank. More information is given in Chapter 2, “Editing images” on page 27.

8. To protect all existing images on disk, you can leave the value 0 in the Protect field. If there is already an image on disk with the same name as the one selected for the image now being saved, and if you want to overwrite the old image, set the value in the Protect field to 1. GDDM-IVU deletes the old image from disk storage before saving the new one and giving it the same name.
9. If you want to store a description with the image, you can enter up to 200 alphanumeric characters in the Description field. This text is saved in the image object and displayed on the Load Image panel whenever the object is loaded.
10. Press ENTER again to save the image. GDDM-IVU displays a message to confirm that the image has been saved.

Discarding an image

This operation erases a specified image from GDDM-IVU storage. The storage thus released becomes available for another image to use.³

1. Press Output (PF6). You should then see the Output Selection panel shown on page 18.
2. Type the name of the image you intend to discard in the Output image field. To discard all images, enter an asterisk (*).
3. Type a value of 4 in the Selection field.
4. Press ENTER. You should then see the Discard Image panel:

----- Discard Image -----

Discard image: OLDIMG Name of the image to be discarded

Press ENTER to confirm discard

Press RETURN or HOME to retain the image

PF 1 Help	2 No Menu	3 Return	4 Input	5 Projectn	6 Output
PF 7 Backward	8 Forward	9 Clear	10	11	12 Home

5. Check the Discard image field. If it contains the correct name, press ENTER. If it does not, or if you decide not to discard this image after all, press Return (PF3) to go back to the Output Selection panel – into which you can enter a different image name – or Home (PF12) to go to the Home panel.

³ Under CMS, however, if you have already reached the insufficient-storage condition (indicated by a CMS error message), discarding images does not release storage. You need to end the GDDM-IVU session and start again.

Printing an image

Background information about printing with GDDM is given in the *GDDM User's Guide*. You may find it helpful to glance through that information before carrying out the procedures described in this section. It may also help if you run into any difficulties.

You can print an image on a graphics or graphics/image printer such as an IBM 3287, 3816, or 4224. Under TSO and CMS systems only, you can also create image output suitable for advanced-function printers such as the IBM 3800 Model 3, 3820, or 3825. However, the Export function (see page 25) creates this type of output, and is generally simpler to use.

1. To use a graphics or graphics/image printer, you need some information from your system-support personnel:
 - If you are running under CICS or TSO, find out the name of the printer. Under CICS, this is a terminal identifier, and under TSO an LU (logical unit) name.
 - If you are running under CMS, GDDM-IVU creates a print file. This file is sent to a printer. Find out whether this happens automatically, or whether you have to take any action to cause it to happen. Find out, as well, whether there are any special naming requirements for the print file.
2. Select option 4 on the Home panel, or option 2 on the Output Selection panel, described on page 18, which you can display from other panels by pressing Output (PF6). You should then see the Print Image panel:

----- Print Image -----

Print image: Name of the image to be printed

Projection: Name of a projection to be applied (optional)

Printer name: Name of the printer to be used

PF 1 Help	2 No Menu	3 Return	4 Input	5 Projectn	6
PF 7 Backward	8 Forward	9 Clear	10	11	12 Home

3. Ensure that the name of the image to be printed appears in the Print image field.
4. The Projection field is described in Chapter 2, "Editing images" on page 27. You can leave it blank.
5. Type the name of the printer in the Printer name field. If your system-support personnel have not specified any particular requirements, choose a name yourself. It must follow the conventions for image names, as described in "Names of images" on page 13. A letter followed by up to seven letters or numbers is always a valid name.

On systems in which GDDM-IVU puts the image output into a disk file rather than sending it direct to a printer, GDDM-IVU normally gives the output the name you specify as the printer name. The output is normally of type

ADMPRINT if intended for a graphics (or graphics/image) printer. Ask your computer support personnel how to print these files on your system.

6. Press ENTER.

7. If any further actions are required to print GDDM print files, end the GDDM-IVU session and do them.

Nickname file for graphics printers

Nicknames for printers other than the IBM 3268 or 3287 need to be defined before you can use those printers. This is a system-setup task and is described in the *GDDM System Customization and Administration* book. Brief instructions are also provided here.

If a suitable nickname has not already been defined, and your subsystem is either CMS or TSO, follow the basic instructions for creating a nickname file given here. For CICS subsystems, only system-support personnel can create nicknames.

Under TSO systems, you must allocate a file with a ddname of ADMDEFS. Under CMS systems, the file must be called PROFILE ADMDEFS and be on a currently accessed disk (usually the A disk). The file must contain a nickname statement specifying a suitable device token for the printer. If the printer does not match the device token specified by the nickname, the printed output may be incorrect, some parts of the image may be missing, or the output may be all black.

Here is an example nickname statement (the first position in the line must be blank):

```
ADMMNICK FAM=2,NAME=PRT123,DEVTOK=X4224QE
```

This supplies a device token for a 4224 printer called PRT123 set up with quarto paper.

For devices other than 3287 and 3268 printers, if you do not use a device token, the results are unpredictable; for example, the printed page may be completely black.

A nickname file set up for printing output from the GDDM-PGF Interactive Chart Utility would also be suitable for printing GDDM-IVU images.

Advanced-function printers: The only difference between using the Print function as described here and the Export function as described on page 25 is in the filenames and filetypes of the output files under CMS. The Export function assigns a filename and filetype specified by you. The Print function, if used as described here, assigns a filename that is the same as the printer name, and a filetype of PSEG38PP or PSEG4250 for page segments in 38xx or 4250 formats respectively, or LIST38PP or LIST4250 for documents in these formats.

To use the Print function to create output for an advanced-function printer you will need to create a nickname file. Setting up a nickname file is outlined above. It will need to contain a statement like this:

```
NICKNAME FAM=2,TOFAM=4,DEVTOK=IMG240X,
        PROCOPT=( (OFDSTYPE,PSEG))
```

This creates a page segment for a 38xx type printer. If the image is to be printed on a 4250, change IMG240X to IMG600X. If a document is required rather than a page segment, change PSEG to DOC.

GDDM supplies a number of processing options (procopts) that can be specified for an advanced-function printer on the nickname statement. However, the only procopts that are honored by GDDM-IVU are OFDSTYPE (or its earlier equivalent, CDPFTYPE) and HRIFORMT (which is used to specify unformatted output). The HRISPILL, HRISWATH, HRIPSIZE, HRIDOCNM, and COLORMAS processing options have no effect.

Plotting images: If you run GDDM-IVU from a distributed-function terminal such as a 3179-G or a 3472-G, you can direct the output to an attached plotter instead of a printer. You need a nickname file with a statement such as this in it:

```
NICKNAME FAM=2,NAME=PLOTTER,TOFAM=1,TONAME=(*,ADMPLLOT)
```

If you enter PLOTTER in the Printer name field, the image is plotted (on the first plotter, if several are attached).

A nickname file set up for plotting output from the GDDM-PGF Interactive Chart Utility would be suitable for plotting GDDM-IVU images.

Plotting can be canceled after it has started by pressing the Clear key on the terminal keyboard.

Loading an image from a non-GDDM image file (importing)

This function is available on TSO and CMS systems, but not on CICS systems.

You can import any type of file. GDDM-IVU inspects the contents of all files being imported, and recognizes several different formats. If a file is in one of these formats, GDDM-IVU interprets the image data it contains. If a file is not in a recognizable format, GDDM-IVU assumes it to be unformatted. Unformatted files are also called *bit maps*. However, if the file is an AFPDS file, it must contain an image; if it contains more than one image, only the first is processed.

GDDM-IVU interprets each bit of an unformatted file as representing a pixel, and each record as representing one row of the image. Unformatted files are correctly interpreted if they are either uncompressed, or compressed using the MMR (modified modified read) algorithm.

The formats that GDDM-IVU recognizes are:

- IBM 38xx page printer:
 - complete document, or
 - page segment, or
 - image segment
- IBM 4250 page printer:
 - complete document, or
 - page segment, or
 - image segment

- IBM 8815 Scanmaster data stream
- IBM 3193 data stream:
 - compressed, or
 - uncompressed

Note: An image imported from an IBM 38xx or 4250 page printer file may be smaller than it was when the image was created. This is because the width and depth are rounded down on output to whole multiples of 32 pixels. For example, if a page size of 8 inches by 11 inches is used when a file is created for a 240-pixels per inch device, the size in pixels is 1920 by 2640. This is rounded down to 1920 by 2624 pixels.

How to import an image

1. Starting with the Home panel (or any other panel that has Input assigned to a PF key), press Input (PF4). You should then see the Input Selection panel. This is shown on page 10.
2. Choose an internal GDDM-IVU name for the image and type it into the Input field.
3. Type 3 into the Selection field.
4. Press ENTER.

You should then see the Import Image panel, with the Import image field initialized with your chosen internal name:

----- Import Image -----					
Import image: IMPIMAG		Name of the image to be imported			
File id:					
Projection:		Name of a projection to be applied (optional)			
Protect: 0		0 = Existing image must not be overwritten 1 = Existing image can be overwritten			
Echo: 0		0 = Display the imported image 1 = Do not display the imported image			

PF 1	Help	2 No Menu	3 Return	4 Input	5 Projectn 6 Output
PF 7	Backward	8 Forward	9 Clear	10	11 12 Home

5. Type the name of the file containing the image to be imported in the File id field.

Under CMS, the name must include the filename and filetype, and can include the filemode. The two or three elements of the name must be separated by blanks, like this:

IM01 PSEG38PP B

If the filemode is omitted or specified as an asterisk (*), all accessed disks are searched in the usual CMS order.

importing an image

Under TSO, it must be allocated to a sequential data set, or a single specified member of a partitioned data set, such as:

IMFILE

or a sequential data set name, such as:

'PICTURES.IMAGES.NUM123'

or a member of a partitioned data set, such as:

'PICTURES.IMAGES.PSEGS(IMAGE1)'

If the quotation marks are omitted, the PROFILE PREFIX high-level qualifier is added to the front of the name in the usual TSO way.

6. You can leave the Projection field blank. More information is given in Chapter 2, "Editing images" on page 27.
7. If an image with the name you have chosen is already present in GDDM-IVU storage, you can replace it with the imported image by setting the value of the Protect parameter to 1. Leaving a value of 0 prevents you from accidentally destroying an existing image if you specify its name by mistake.
8. A 0 in the Echo field causes the image to appear on the screen as soon as it is imported. If you do not want this to happen, set the value to 1.
9. Press ENTER. The image is loaded into GDDM-IVU storage, unless it was unformatted, in which case, the Import Unformatted panel (shown below) is displayed first.

The imported image is held in the same internal form as an image loaded from a GDDM image file or created by scanning. Any GDDM-IVU function can be used on it.

If the image is unformatted, follows steps 10 through 12 with the Import Unformatted panel:

----- Import Unformatted -----					
Resolution		Choose the import resolution			
Horizontal: 0		0 = Same resolution as the display			
Vertical: 0		1 = 120 pixels per inch			
		2 = 200 pixels per inch			
		3 = 240 pixels per inch			
Compression: 0		Choose the import compression			
		0 = Uncompressed			
		1 = Compressed			

PF 1 Help	2 No Menu	3 Return	4	5	6
PF 7 Backward	8 Forward	9 Clear	10	11	12 Home

10. Specify a horizontal and vertical resolution for the image. The value 2 in the resolution fields on this panel is intended for importing IBM 8815 Scanmaster images, and causes GDDM-IVU to assign resolutions of 203 pixels per inch horizontally and 196 vertically. Otherwise, similar considerations to those described on page 11 for GDDM images without resolution apply.
11. If the image is compressed using the MMR algorithm, change the Compression field to 1. Otherwise leave it as 0.
12. Press ENTER. The image is imported, and the Import Image panel is redisplayed.

Saving an image in a non-GDDM image file (exporting)

This function is available under TSO and CMS, but not under CICS.

You can create any of the types of file mentioned on page 22 as being recognized on import, and compressed or uncompressed unformatted images.

38xx and 4250 format output may not be printable if the resolution of the image does not match that of the printer. To obtain printable output in such cases, use the Print function and a nickname statement, as described in “Advanced-function printers” on page 21.

- 1. Starting with the Home panel (or any other panel that has Output assigned to a PF key), press Output (PF6). You should then see the Output Selection panel. This is shown on page 18.
- 2. Type the internal GDDM-IVU name of the image to be exported in the Output image field.
- 3. Type 3 into the Selection field
- 4. Press ENTER.

You should then see the Export Image panel, with the Export image field initialized:

Export image: EXPIMG

File id:

Projection:

Output format: 1

Output type: 1

Protect: 0

----- Export Image -----

Name of the image to be exported

Name of a projection to be applied (optional)

1 = 38xx Printer 2 = 4250 Printer

3 = 8815 Scanner 4 = 3193 Uncompressed

5 = 3193 Compressed 6 = Unformatted

1 = Page segment 2 = Document

3 = Image segment

0 = Existing file must not be overwritten

1 = Existing file can be overwritten

PF 1 Help 2 No Menu 3 Return 4 Input 5 Projectn 6

PF 7 Backward 8 Forward 9 Clear 10 11 12 Home

- 5. Type the name of the file to which the image is to be exported in the File id field.

Under CMS, this name must include the filename and filetype, and can include the filemode. The two or three elements of the name must be separated by blanks, like this:

IM99 IMDATA C

If the filemode is omitted, A1 is the default.

Under TSO, it must be allocated to a sequential data set, or a single specified member of a partitioned data set, such as:

IMAGFIL

or a sequential data set name, such as:

'FORMS.IMAGES.SEQ321'

or a member of a partitioned data set, such as:

'PICTURES.IMAGES.PSEGS(IMAGE1)'

If the quotation marks are omitted, a data-set qualifier is added to the front of the name in the usual TSO way.

Under TSO, the standard DCB characteristics for files containing exported images are a RECFM of V, LRECL of 2004, and BLKSIZE of 2008.

6. You can leave the Projection field blank. More information is given in Chapter 2, “Editing images” on page 27.
7. Enter a number in the range 1 through 6, corresponding to the output format you require, in the Output format field.
8. If you specify an output format of 1, 2, or 3, type a number in the range 1 through 3, corresponding to the output type you require, in the Output type field. This field is ignored for other output formats.
9. If a file with the name you have specified already exists, you can replace it with the exported image by changing the value of the Protect parameter to 1. Leaving a value of 0 prevents you from accidentally destroying an existing file if you specify its name by mistake. This also applies to a member of a partitioned data set.

If a TSO ddname was specified in the File id field, the Protect field has no effect. Specifying the ddname of an existing file causes that file to be overwritten, irrespective of the value in the Protect field.

10. Press ENTER. The image is copied to the specified file in the specified format (unless it is empty – contains no pixels – in which case no output is produced).

Chapter 2. Editing images

This chapter describes two different ways of making changes to images. You can *edit* an image from GDDM-IVU's Edit Image panel, in which case any changes you make are immediate and apply to a single image only. Alternatively, you can use a *projection*. A projection is a stored set of editing instructions that you can apply repeatedly to any number of images. If the changes you intend to make are likely to apply to more than one image, you are recommended to define a projection. You should use a projection if, for example, you intend scanning many completed copies of a form and want to extract and display the same information from each one. Projections are described on page 35.

At the end of this chapter (on page 45) there is a test page. As soon as you have gained enough knowledge, you can scan it and try the manipulations suggested.

The edit functions

To change an image, you can:

- Copy a section from one part of the image to another.
- Move a section from one part of the image to another. The “move” operation differs from the “copy” operation in that copying leaves the original section unchanged, whereas moving leaves a blank space. Moving is essentially a “cut-and-paste” type of operation, similar to cutting a section out of a document and pasting it into a new position.
- Erase a section, either inside or outside a specified rectangle.
- Fetch another image to be overlaid on a section of the one currently being edited. This function is not available on CICS systems unless your system-support personnel have provided a transaction that sets the image limit higher than “1” (using the IUMLO call) and invokes GDDM-IVU. IUMLO is described in Appendix A, “GDDM-IVU calls” on page 69. An example of its use is shown on page 50.
- Transform *a copy* of any section of the image by:

Scaling (making the image bigger or smaller and possibly altering its proportions)

Reflecting (reversing, as if the image were viewed in a mirror)

Negating (making the light pixels dark and the dark pixels light)

Rotating (turning the image through multiples of 90 degrees).

GDDM-IVU allows you to indicate where in the image the transformed section is to be positioned.

- Trim the edges of an image to make it smaller. This differs from erasing, in that erasing blanks out part of the image without altering its overall size.

Any operations that can be carried out on a section of the image can also be applied to the whole image.

Several editing operations involve taking a section of an image and placing it in a different position. In GDDM terminology, the original section is called the *source* or *extract*, and the area into which it is placed, the *target*.

How to edit an image

1. Press Home (PF12) to go to the Home panel (shown on page 5).
2. Enter 3 in the Selection field of the Home panel, and press ENTER. You should then see the Edit Image panel:

```

----- Edit Image -----
Edit image:      Name of the image to be edited

Selection:      Choose an edit function, then press ENTER
                1 = Copy      2 = Move
                3 = Erase     4 = Fetch
                5 = Transform 6 = Trim

Edit defaults: 0      0 = Use current defaults
                  1 = Set defaults

-----
PF 1 Help      2 No Menu  3 Return  4 Input   5 Projectn  6 Output
PF 7 Backward  8 Forward  9 Clear  10       11       12 Home
  
```

3. Decide whether you want to (a) alter an image already held in GDDM-IVU storage, or (b) create a completely new image from one or more existing ones. In case (b), the existing image or images can be held in either GDDM-IVU storage or in a disk file.

If you decide on (a), type the name of the image you intend to alter in the Edit image field. If you decide upon (b), choose a name for the new image and type that.

Option (b) is not normally available when the subsystem is CICS.

4. Indicate the editing function you require by typing a number from 1 through 6 in the Selection field.

If you decided upon option (b) and so typed a new name in the Edit image field, you should specify the fetch function, 4, to obtain an image to edit.

5. You can leave a value of 0 in the Edit defaults field.

A 1 in this field lets you change specific parameters that affect the way editing operations are carried out. You can control, for example:

- Whether blank areas are to appear dark or light on the screen
- Whether you can change the size and proportions of the source image during copying, moving, and fetching
- How the source image is to overpaint or merge with the target area.

Instructions are given on page 33.

6. Press ENTER. GDDM-IVU creates an image for you to edit. More information about this is given in the section “New image created for editing” on page 29.
7. What happens next depends on which edit function you selected. In the cases of the following three functions, further menus are displayed, the use of which is described in later pages:

Erase (which is described on page 30)

Fetch (which is described on page 31)

Transform (which is described on page 32)

In the cases of the copy, move, and trim functions, there are no further menus; instead, GDDM-IVU issues step-by-step instruction messages.

All the editing operations require you to indicate areas or positions within the image. You make these indications by adjusting a cursor and pressing ENTER. On the 3193 terminal, you indicate an area with a box cursor and a position with a cross cursor; more information is given in Cursors on 3193 terminals. On other types of terminal, you use the ordinary alphanumeric cursor for both.

You can cancel any operation at any time before it is completed by pressing Return (PF3).

8. When you complete the selected operation, the Edit Image panel is redisplayed. You can then select another editing operation, either for the same image or for another image, which you name in the Edit image field.

If you do not like the results of the last operation, you can reverse its effects at this stage by pressing the Undo (PF5) key. You cannot undo any operation other than the most recent.

9. The final operation is frequently a trim to reduce the edited image to the required size. Untrimmed, the image is 8.25 by 11.75 inches.

Cursors on 3193 terminals: The box and cross cursors appear on the screen only when you need to use them. The keys that control them are shown in Figure 3 on page 8.

To use one of these cursors, you must first activate it by holding down an Alt key and pressing the Cursor mode key (which is also the PF24 key). You move the active cursor with the same Cursor move keys that you use for the ordinary alphanumeric cursor. Pressing Alt and Cursor mode again reactivates the alphanumeric cursor.

The operator information area at the bottom of the screen indicates when a box or cross cursor is active by displaying Cr preceded by a box or a cross symbol.

You can alter the size of the box cursor by holding down a Shift key and pressing one of the Cursor move keys. The Left and Right keys move the right-hand side of the box in and out, and the Up and Down keys move the bottom of the box up and down.

If the cursors are hard to see, adjusting the contrast and, perhaps, the brightness of the screen may help. You can also change the cursors to dark or make them blink. The Alternate cursor and Cursor blink keys (which are also the PF23 and PF22 keys), when pressed with the Alt key held down, respectively change the cross and box cursors between light and dark, and start or stop them blinking.

Sometimes the cross cursor is hard to recognize because it is at the edge of the image area and one or two of its arms are not visible. If GDDM-IVU tells you to indicate a position but you cannot find the cross, check the edges, particularly at the corners.

New image created for editing: When you press ENTER after filling out the required fields on the Edit Image panel, the first thing GDDM-IVU does is to create a new, blank image.

erasing part of an image

On a 3193 terminal, the size of the new image is 8.25 by 11.75 inches (209 by 298 millimeters), which is the maximum that can be displayed. On other types of terminal, the image may be larger than the screen size, and you can scroll to edit different parts of the image.

The name of the new image is the one you entered in the Edit image field. If this is the name of an image already present in GDDM-IVU storage, GDDM-IVU copies that existing image into the top left of the new image before discarding the existing image. The apparent effect is to make the named image 8.25 by 11.75 inches in size, extending it to the right and bottom with blank areas if necessary.

If the name you enter is a new one, the new image remains entirely blank, which is why your first operation is normally a fetch to obtain an image to work on. The resolution of this new blank image is the same as that of the screen you are using, so any images you fetch have a resolution no higher than your screen.

By default, blank areas appear dark on the screen, so you may not be able to see the extent of the image. You can change this by setting the Clear mode field on the Edit Defaults panel to 1 (see page 33), making blank areas light and thereby visible. If you specify an erase operation on the Edit Image panel, you should see the Edit Erase panel when you press ENTER:

----- Edit Erase -----

Erase mode: 0 0 = Erase inside the area
 1 = Erase outside the area

Press ENTER to perform the erase

Press RETURN to cancel the erase

PF 1 Help	2 No Menu	3 Return	4 Input	5 Undo	6 Output
PF 7 Backward	8 Forward	9 Clear	10	11	12 Home

Erasing a section of the image

1. Enter a 0 in the Erase mode field if you want to erase the part of the image lying inside a specific rectangle (you indicate the rectangle in the next step), or a 1 if you want to erase the part outside.
2. Press ENTER. GDDM-IVU displays messages instructing you to define the rectangle on the screen with a cursor.

The erased area appears either dark or light, depending on the setting of the Clear mode field of the Edit Defaults panel (see page 33).

3. After the erase is complete, press Return (PF3) to return to the Edit Image panel. If you specify a fetch operation on the Edit Image panel, you should see the Edit Fetch panel when you press ENTER:

----- Edit Fetch -----						
Fetch image:		Name of the image to be fetched				

PF 1	Help	2 No Menu	3 Return	4 Input	5 Undo	6 Output
PF 7	Backward	8 Forward	9 Clear	10	11	12 Home

The fetch function is not normally available when the subsystem is CICS, because fetching requires GDDM-IVU storage to hold more than one image. Unless your computer support personnel have supplied a locally written transaction that uses IUMLO to increase the in-store image limit, only one image can be held at a time under CICS.

Fetching an image

1. Enter the name of the image to be fetched in the Fetch image field.
2. Press ENTER.

If the named image is already held in GDDM-IVU storage, GDDM-IVU displays messages instructing you to indicate the position for the image with a cursor. If the image is not held in GDDM-IVU storage, GDDM-IVU displays the Input Selection panel (shown on page 10). If you press ENTER again, GDDM-IVU loads the named image, returns to the Fetch Image panel, and issues the positioning instructions.

The instructions tell you to indicate either a position for the top left corner of the fetched image, or an area to be filled by it, depending on the value in the Extract position field of the Edit Defaults panel (see page 33). In the first case, the image is placed in position at its actual size, whereas in the second, it is scaled to fit the area.

If you are editing one image and you fetch another image of higher resolution, the definition of the second image is reduced.

- After the fetch is complete, press Return (PF3) to return to the Edit Image panel. If you specified a transform operation on the Edit Image panel, you should see the Edit Transform panel when you press ENTER:

```

----- Edit Transform -----
Reflect: 0      0 = None      1 = Left - right  2 = Top - bottom
                        3 = Top - left   4 = Top - right

Negate: 0      0 = No       1 = Yes

Rotate: 0      0 = 0        1 = 90
                        2 = 180       3 = 270

Horizontal scale: 1.0
Vertical scale: 1.0
-----
PF 1 Help      2 No Menu    3 Return    4 Input     5 Undo      6 Output
PF 7 Backward  8 Forward    9 Clear    10          11          12 Home

```

Transforms

Transforms (transformations) are carried out on a section of the image that you select using a cursor, as follows.

- The four transformations you can make are: reflection, negation, rotation, and scaling. GDDM-IVU always does them in that order. Scaling can be different in the horizontal and vertical directions, allowing you to alter the proportions of the source. This gives a total of five transformation parameters.

Specify the transformations you require by entering one of the listed parameter values into one or more of the first three fields, and the required scale factors in the last two fields.

- Press ENTER. GDDM-IVU issues instruction messages telling you to indicate the source area upon which the transformations are to be performed, and where the image is to be positioned after transformation.

GDDM-IVU then extracts the source image, transforms it, and places the result at the indicated location.

- When the process is complete, press Return (PF3) to return to the Edit Image panel. If you set the value in Edit defaults field on the Edit Image panel to 1, you should see the Edit Defaults panel when you press ENTER:

```

----- Edit Defaults -----
Extract position: 0      0 = Scale to fit defined area
                        1 = Place top left at defined point

Clear mode: 0      0 = Set move/erase area to black
                        1 = Set move/erase area to white

Mix mode: 0      0 = Default      1 = Overpaint      2 = Merge
                        3 = Difference  4 = 'AND'          5 = Subtract

Scale mode: 0      0 = Normal      1 = Priority to black
                        2 = Priority to white

-----
PF 1 Help      2 No Menu    3 Return    4 Input     5 Undo      6 Output
PF 7 Backward  8 Forward    9 Clear    10          11          12 Home

```

This panel allows you to specify parameters that affect several different editing operations. To use it, enter the required values in one or more of the four input fields and press Return (PF3) to return to the Edit Image panel. There must be a value in each field; if you do not wish to change a parameter, leave the default value of 0 in the field.

The defaults you select apply until you either change them or end the GDDM-IVU session.

The Extract position field affects the copy, move, and fetch operations. A value of 0 means that GDDM-IVU tells you to indicate the size and position of the target area for these operations, and scales the source to fit that area. A value of 1 means that GDDM-IVU tells you to indicate a position only, and puts the top left-hand corner of the source at that position without altering its size.

The Clear mode field affects the move, erase, and transform operations. It also affects the creation of the new image at the start of editing (see “New image created for editing” on page 29). A value of 0 means that the pixels corresponding to blank areas of the image are set to ‘Off’, making the areas appear dark on the screen, and a value of 1 means that the pixels are set to ‘On’, making the areas appear light on the screen.

The Mix mode field affects the copy, move, fetch, and transform operations. In all these operations, a source image is overlaid on a target area. The mix mode determines how the pixels of the source are combined with those of the target. The default is overpaint, which makes the source completely obscure the target. The results of all five modes of mixing are shown below.

Source and target	0 or 1 Overpaint	2 Merge	3 Difference	4 AND	5 Subtract
W on W	W	W	B	W	W
W on B	W	W	W	B	B
B on W	B	W	W	B	W
B on B	B	B	B	B	B

W = ‘On’ pixel – white on 3193 screen

B = ‘Off’ pixel – black on 3193 screen

The Scale mode field affects the copy, move, fetch, and transform operations. All these operations can involve a source image changing size, and thereby cause an increase or decrease in the number of pixels in the image. The scale mode affects the way pixels are deleted to make an image smaller. A value of 0 means that the proportion of ‘Off’ and ‘On’ pixels remains as nearly as possible the same. 1 means that ‘On’ pixels are deleted in preference to ‘Off’, giving better retention of dark lines on a light background. 2 means that ‘Off’ pixels are deleted in preference to ‘On’, giving better retention of light lines on a dark background.

Projections

A *projection* is a set of stored cut-and-paste-type editing instructions. A projection tells GDDM-IVU to take one or more rectangular sections, called *extracts*, from a source image and use them to build a target image. It defines:

- Where in the source image each extract is to be taken from
- Transformations, such as rotating and negating, to be performed on each extract
- Where in the target image each extract is to be placed

There are no exact equivalents to the edit functions, but similar end results can usually be achieved. However, all the extracts in any target image must come from a single source image, so there is no way of doing what the Edit fetch function does (that is, you cannot take several different source images and combine them in a single target).

Projections can be created in GDDM-IVU storage, and saved in and loaded from disk files. Details are given in subsequent sections.

You can apply a projection when you do one of the following image transfers:

- Viewing (transferring from GDDM-IVU storage to the screen)
- Loading (transferring from a GDDM image file to GDDM-IVU storage)
- Scanning (transferring from a scanner to GDDM-IVU storage)
- Saving (transferring from GDDM-IVU storage to a GDDM image file)
- Printing (transferring from GDDM-IVU storage to a printer)
- Importing (transferring from a non-GDDM disk file to GDDM-IVU storage)
- Exporting (transferring from GDDM-IVU storage to a non-GDDM disk file)

You apply a projection by entering its name in the `Projection` field of the relevant panel. If the projection is not present in GDDM-IVU storage, GDDM-IVU searches for it on disk and loads it. The name is retained in the `Projection` field of the panel on which you specify it, even when you leave that panel and return to it. Unlike image names, however, a projection name is not propagated to any other panel: the different functions on other panels are likely to require different projections.

When you are creating a projection, GDDM-IVU demonstrates its effects using the image currently displayed on the screen. Before starting, therefore, you normally need to select and display an image that is typical of those to which the projection will be applied.

The effects are demonstrated by applying the projection during the transfer of the image from GDDM-IVU storage to the screen. Although you appear to be editing the image, you are, in fact, not. Your view of it changes, but the original image itself remains unchanged. To create an image containing what you see on the screen at the end of projection creation, you must apply the projection to the original image in a save operation.

To see the effects of a projection, you can view the GDDM-IVU sample image `ADMU5IMG` with no projection and then with the sample projection `ADMU5PRJ`.

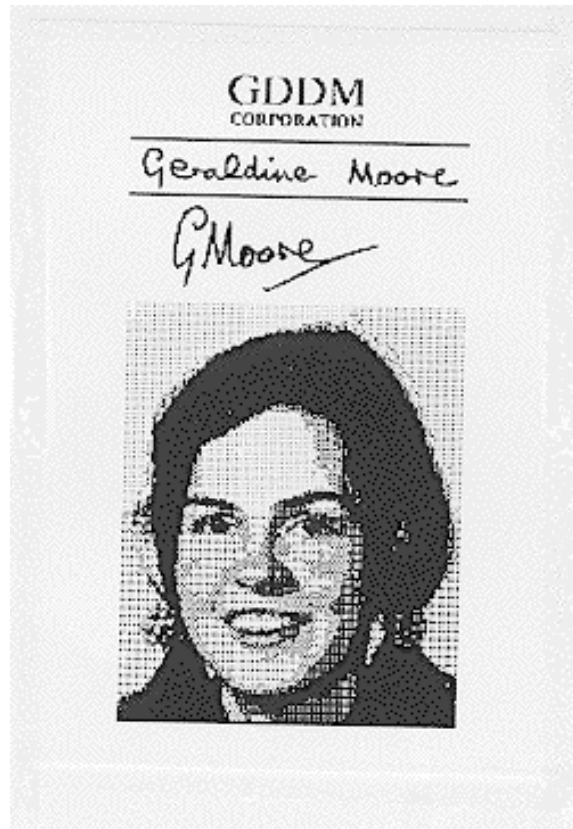


Figure 5. Effect of sample projection ADMU5PRJ. This shows the result of applying the GDDM-IVU sample projection to the GDDM-IVU sample image ADMU5IMG (illustrated in Figure 4 on page 15). Selected parts of the sample image have been extracted and rearranged in the form of an identity badge.

Creating a projection

1. Make sure you have a suitable image on the screen. If not, use the view function (see page 13) to put one there.
2. Starting with any panel except those edit panels that have an Undo (PF5) key, press Projectn (PF5). You should then see the Projection panel:

```

----- Projection -----
Projection:      Name of the projection

Selection: 1     Choose a projection function, then press ENTER
                  1 = Create a new projection
                  2 = Load projection in GDDM format
                  3 = Save projection in GDDM format
                  4 = Discard projection

Protect: 0       0 = Existing projection must not be overwritten
                  1 = Existing projection can be overwritten
-----
PF 1 Help      2 No Menu  3 Return  4 Input   5          6 Output
PF 7 Backward  8 Forward  9 Clear  10         11         12 Home
  
```

3. Choose a name for the projection and enter it in the Projection field. It can be the name of an existing projection. GDDM-IVU allows you to discard an

existing projection and create a new one with the same name, though not to alter it. When it has been created, a projection cannot be altered.

The name must be a valid one. The validity rules depend on the subsystem, but a letter followed by up to seven letters or numbers is valid on all subsystems. The information about image names in “Names of images” on page 13 applies equally to projections.

4. Enter 1 in the Selection field.
5. If a projection with the name you have chosen is already present in GDDM-IVU storage, you can replace it with the new projection by setting the value of the Protect parameter to 1. A value of 0 gives protection against accidentally destroying an existing projection if its name is specified by mistake.
6. Press ENTER.

The Create Projection panel is displayed. The Create Projection panel lets you define up to eight extracts. It has one column per extract, with one line each for defining the source, the transform, and the target. At the top of each column there is also a command line. Its use is described later, in “The command line” on page 38.

If you have previously defined a projection during the current GDDM-IVU session, the projection panels are initialized with the parameter values retained from their previous use, as explained in “Altering a projection” on page 38.

Otherwise, GDDM-IVU initially defines one extract using default parameter values. The panel shows a * wherever default values are in force. Wherever you have defined parameters explicitly, in the ways described below, the panel displays the word Set. Unused columns are filled with dots.

```

----- Create Projection -----
To define Extract 1 source, put cursor in Source line column 1 and press ENTER

Extract -->  1      2      3      4      5      6      7      8
Commands -->

Source      *      .      .      .      .      .      .      .
Transform   *      .      .      .      .      .      .      .
Target      *      .      .      .      .      .      .      .

Commands: I Insert  M Move   C Copy  D Delete  R Repeat  A After  B Before
-----
PF 1 Help      2 No Menu  3 Return  4 Input    5          6 Output
PF 7 Backward  8 Forward  9          10         11         12 Home
  
```

This is how you define the source, transform, and target for each extract.

1. The default extract is the whole of the source image, expanded with ‘Off’ pixels to 8.25 x 11.75 inches if necessary. If you want the default, go to step 4 on page 37. Otherwise, to define less than the whole image as the extract, put the cursor (the ordinary alphanumeric cursor) in the Source line of column 1.
2. Press ENTER.

The menu disappears and GDDM-IVU issues step-by-step instruction messages for indicating the source area with a cursor. On 3193 terminals, you use the box cursor; more information about this is given in “Cursors on 3193

terminals” on page 29. On other terminals, use the ordinary alphanumeric cursor.

As an alternative, you can press Menu (PF2) to display a panel for defining the source area. It is the Extract Source panel, which is described on page 39. Use it when you want to define the extract size and position in inches or millimeters, rather than by pointing with a cursor.

3. When you have defined the extract, either with the cursor or by entering parameters on the Extract Source panel, press Return (PF3). GDDM-IVU returns to the Create Projection panel, and displays the extract only.
4. To change the extract’s appearance by transforming it, or to control the quality of images created by scanning, put the cursor on the Transform line of column 1. If you do not want to do either of these things, go to step 7.
5. Press ENTER.
6. The Transform panel, which is described on page 39, is displayed. When you have entered all the necessary transform parameters on this panel, press Return (PF3). GDDM-IVU returns to the Create Projection panel, displaying the transformed extract.
7. The default target area is the whole of the 8.25 x 11.75 inch target image. If this is satisfactory, go to step 10. To define less than the whole target image as the target area, put the cursor in the Target line of column 1.
8. Press ENTER.

The menu disappears and GDDM-IVU issues step-by-step instruction messages for indicating the target area with the cursor. The extract is scaled to fit the target area.

On 3193 terminals, you define the area with the box cursor, which GDDM-IVU initializes to be the same size as the extract. If you do not alter the size of the box, the target area is the same size as the extract, so the extract does not change size.

Alternatively, or in addition to using the cursor, you can press Menu (PF2) to display a panel for defining the target area. It is the Extract Target panel, which is described on page 40. Use this panel when you want to define the target’s size and position in inches or millimeters. Also use it if you want to change the scale mode or mix mode. These modes are explained in the section on edit defaults on page 33.

The Extract Target panel offers the best way of keeping the source and target areas the same size when creating a projection on a terminal other than the 3193. After the second cursor pointing, which defines the size of the target area, press Menu (PF2) to display the menu, overwrite the values in the two scale fields with the value 1, and press ENTER.

9. When you have defined the target area, press Return (PF3). GDDM-IVU returns to the Create Projection panel, and displays the effects of the projection in full.
10. If you require two or more extracts, repeat the whole procedure from step 1 on page 36, using columns 2, 3, and so on in place of column 1. Some further information is given in “Multiple extracts” on page 38.
11. When the source, transform, and target for all extracts are satisfactorily defined, press Return (PF3) (or Home (PF12)).

GDDM-IVU creates the projection and redisplay the Projection panel (or the Home panel). The projection is held in GDDM-IVU storage, ready for use. It can be specified in any Projection field. It can also be saved in a disk file (see “Saving a projection” on page 41).

Altering a projection: Strictly speaking, you cannot alter a projection after you have pressed Return (PF3) with the Create Projection panel on the screen. Up to this point, though, you can repeat any step and change any parameter in the projection you are creating, whether that parameter was defined by a cursor or by entering data on a panel.

However, after a projection has been created, you can return to the Create Projection function. If you do, you will find that GDDM-IVU has retained all the parameter values from the previous projection. You can use these as the basis of a new projection, altering them as required. In this way, you can, in effect, alter the last projection you created, though only the last one: no values are retained for any earlier projections. So you can repeatedly create a projection, try it out, and recreate it with amendments, as long as you do not end the GDDM-IVU session. If you want to reuse the same projection name, remember to set the Protect field on the Projection panel to 1.

If you load a projection from disk, you can neither change it nor display its values in the projection panels.

Multiple extracts: Multiple extracts are drawn in the order in which they appear on the Create Projection panel – the extract defined in column 1 first, the one defined in column 2 next, and so on. In the default mix mode of overpaint, later drawn extracts obscure earlier drawn ones where they overlap. If an extract disappears from the screen when you define another one, this could be the reason. You can alter the drawing order by altering the order of the columns using the M (Move) command described in “The command line.”

Sometimes you may want to keep the whole of the source image and overlay an extract on top of it. In this case, you need a projection in column 1 that has the whole image as both the source and the target. Since these are the defaults, the simplest way to achieve the required result is to leave the default values in column 1 and define the overlay extract in column 2. This assumes, of course, that you have not previously set non-default values in column 1 during the current GDDM-IVU session.

Setting default values: This is how to put a set of default values into any column on the Create Projection panel.

If the column contains any values already, delete it using the D (delete) command and then reinstate it (unless there was an unused column to the right) with no initial values using the I (Insert) command. Put the cursor on the Transform line and press ENTER. Press ENTER again to select the default transform values, and Return (PF3) to return to the Create Projection panel.

The command line: The commands operate on the columns of the Create Projection panel by reordering, copying, or deleting them.

To copy a column, enter C (Copy) on its Commands line, and A (After) or B (Before) on the Commands line of the column after or before which you want it copied. Moving a column is similar, except that you enter M (Move) instead of C.

To create an adjacent copy of a column, enter R (Repeat) on its Commands line. GDDM-IVU copies it immediately to the right.

To create an entirely new column, enter I (Insert) on the Commands line of any column except the last. GDDM-IVU inserts a new column to its left.

To delete a column, enter D (Delete) on its Commands line.

When a column is deleted using the D command, the contents of all columns to the right are moved one position to the left, and a new column 8 is created. When a column is created using the C, R, or I command, the contents of all columns to the right of the new one are moved along one position, and the content of column 8 is discarded.

```

----- Extract 1 Source -----
      Measure extract edges from the top and left of the image

                Top: 0.0
      Left: 0.0                Right: 8.25
                Bottom: 11.75

                Units: 1          1 = Inches      2 = Millimeters
-----
PF 1 Help      2 No Menu    3 Return      4          5          6
PF 7 Backward  8 Forward   9          10         11         12 Home
  
```

1. Decide whether you want to specify the extract in inches or millimeters, and enter 1 or 2 accordingly in the Units field.
2. Type the values you require in the Top, Bottom, Left, and Right fields.
3. Press ENTER.
4. Press Return (PF3) to return to the Create Projection panel and a display of the extract.

```

----- Extract 1 Transform -----
      Brightness: 0      0 = Normal      1 = Decreased      2 = Increased
      Contrast: 0       0 = Normal      1 = Decreased      2 = Increased
      Tone: 0          0 = Normal      1 = Decreased      2 = Increased
                                   3 = Intricate      4 = Graylevel

      Reflect: 0        0 = None        1 = Left - right   2 = Top - bottom
                                   3 = Top - left     4 = Top - right
      Negate: 0         0 = No          1 = Yes

      Rotate: 0         0 = 0           1 = 90
                                   2 = 180          3 = 270
-----
PF 1 Help      2 No Menu    3 Return      4          5          6
PF 7 Backward  8 Forward   9          10         11         12 Home
  
```

using projection target panel

1. The Brightness, Contrast, and Tone fields are for controlling image creation by a scanner. They have no effect if the projection is applied to a process other than scanning.

A value of 0, 1, or 2 in the Brightness or Tone field determines how light or dark the image will be. In either field, 1 makes the image darker and so is for lighter-than-normal originals, and 2 makes it lighter and is for darker-than-normal originals.

Values of 3 and 4 can be used only in the Tone field; they cause the scanner to simulate gray-levels by using half-toning. (These terms are explained in Chapter 1, "Getting started with GDDM-IVU" on page 1.) They are mainly for use with continuous-tone originals such as photographs. 4 gives the best representation of a gray-levels, but with some loss of detail. 3 gives better retention of detail and is recommended for intricate continuous-tone originals.

The contrast field also is mainly for use with continuous-tone originals, though changing the parameter value can sometimes give an improved image of a line original. A value of 1 gives the lowest contrast, and 2 the highest. The lower the contrast, the more overall gray is the image. The higher the contrast, the more areas of extreme black and white the image will contain. High contrast tends to cause loss of detail in the lightest and darkest areas, but gives the sharpest distinction between areas of slightly different tone.

2. Enter the required values in the Reflect, Negate, and Rotate fields.
3. Press ENTER to see a display of the transformed extract.
4. Press Return (PF3) to return to the Create Projection panel.

```
----- Extract 1 Target -----
Horizontal offset: 0.0
Vertical offset: 0.0
Units: 1          1 = Inches    2 = Millimeters

Horizontal scale: 1
Vertical scale: 1
Scale mode: 0     0 = Normal    1 = Priority to black
                                   2 = Priority to white

Mix mode: 0       0 = Default    1 = Overpaint    2 = Merge
                                   3 = Difference    4 = 'AND'        5 = Subtract

-----
PF 1 Help      2 No Menu    3 Return    4          5          6
PF 7 Backward  8 Forward    9          10         11         12 Home
```

1. Decide whether you want to specify the position of the target area in inches or millimeters and enter 1 or 2 accordingly in the Units field.
2. Enter, in the Horizontal offset and Vertical offset fields, the position for the top left corner of the extract in relation to the left and top edges of the overall target image.
3. Enter the values for the required modes in the Scale mode and Mix mode fields. Their meanings are explained on page 33.
4. Press ENTER to see the extract in the target area.
5. Press Return (PF3) to return to the Create Projection panel.

Saving a projection

Projections created by GDDM-IVU can be copied from GDDM-IVU storage to GDDM projection files on disk, as follows.

1. Starting with any panel except those edit panels that have an Undo (PF5) key, and those panels concerned with projections, press Projectn (PF5). You should then see the Projection panel shown on page 35.
2. Type the name of the projection you intend to save in the Projection field.
3. Type a value of 3 in the Selection field.
4. Leave the Protect field unchanged: it affects only the creation of a new projection.
5. Press ENTER. You should then see the Save Projection panel:

```

----- Save Projection -----
Save projection: PR123      Name of the projection to be saved

File name:                 External name of projection
                           (if different from above)

Protect: 0                 0 = Existing file must not be overwritten
                           1 = Existing file can be overwritten

Description:

-----
PF 1 Help      2 No Menu  3 Return  4 Input    5          6 Output
PF 7 Backward  8 Forward  9 Clear  10         11         12 Home
  
```

6. The Save projection field is initialized with the name you entered on the Projection panel. If you make a mistake or change your mind about which projection to save, overtype this name with a new one.
7. You can leave the File name field blank. If you do, the name of the projection on disk is the same as the internal GDDM-IVU name that you specify in the Save projection field. Alternatively, you can specify a different name in the File name field. It must be a valid name for GDDM objects on disk: a letter followed by up to seven letters or numbers is valid on all systems and subsystems. The information about image names in “Names of images” on page 13 applies equally to projections.
8. To protect all existing projections on disk, you can leave the value 0 in the Protect field. If there is already a projection on disk with the same name as the projection now being saved, and if you want to discard the old projection, set the value in the Protect field to 1. GDDM-IVU deletes the old projection from disk storage before saving the new one and giving it the same name.
9. If you want to store a description with the projection, enter up to 200 alphanumeric characters in the Description field. This text is saved in the projection object, and displayed on the Load Projection panel whenever the object is loaded.
10. Press ENTER to save the projection. GDDM-IVU displays a message to confirm that the save was successful.

Loading a projection

Projections in disk files must be loaded into GDDM-IVU storage before they can be used. If you specify a name in the Projection field on any transfer operation (such as viewing or scanning – see on page 27 for a full list), and no projection of that name is held in GDDM-IVU storage, GDDM-IVU automatically searches for it on disk and loads it. Alternatively, you can load projections explicitly, as follows.

1. Starting with any panel except those edit panels that have an Undo (PF5) key, and those panels concerned with projections, press Projectn (PF5). You should then see the Projection panel shown on page 35.
2. Type the name of the projection you intend to load in the Projection field.
3. Type a value of 2 in the Selection field.
4. Leave the Protect field unchanged: it affects only the creation of a new projection.
5. Press ENTER. You should then see the Load Projection panel:

----- Load Projection -----					
Load projection: PR456		Name of the projection to be loaded			
File name:		External name of projection (if different from above)			
Protect: 0		0 = Existing projection must not be overwritten 1 = Existing projection can be overwritten			
Description:					

PF 1 Help	2 No Menu	3 Return	4 Input	5	6 Output
PF 7 Backward	8 Forward	9 Clear	10	11	12 Home

6. The Load projection field is initialized with the name you entered on the Projection panel. If you make a mistake or change your mind about which projection to load, overwrite this name with a new one.
7. Normally, you should leave the File name field blank. This makes the internal GDDM-IVU name for the projection the same as the name on disk. If you want the internal and external names to be different, enter your chosen internal GDDM-IVU name in the Load projection field, leaving the projection object name in the File name field.
8. If a projection with the internal name you have chosen is already present in GDDM-IVU storage, you can replace it with the loaded projection by setting the value of the Protect parameter to 1. Leaving a value of 0 gives protection against accidentally destroying an existing projection if its name is specified by mistake.
9. You cannot enter anything against Description. After loading the projection, GDDM-IVU may display a description of it here. This would have been supplied when the projection was saved on disk. If no description was supplied, GDDM-IVU leaves this field blank.
10. Press ENTER to load the projection. GDDM-IVU displays a message to confirm that the load was successful.

Discarding a projection

1. Starting with any panel except those edit panels that have an Undo (PF5) key, and those panels concerned with projections, press Projectn (PF5). You should then see the Projection panel shown on page 35.
2. Type the name of the projection you intend to discard in the Projection field. To discard all projections, enter a *.
3. Type a value of 4 in the Selection field.
4. Leave the Protect field unchanged: it affects only the creation of a new projection.
5. Press ENTER. You should then see the Discard Projection panel:

----- Discard Projection -----

Discard projection: PR789 Name of the projection to be discarded

Press ENTER to confirm discard

Press RETURN or HOME to retain the projection

PF 1 Help	2 No Menu	3 Return	4 Input	5	6 Output
PF 7 Backward	8 Forward	9 Clear	10	11	12 Home

6. Check the Discard projection field. If it contains the correct name of the projection that you wish to discard, press ENTER. If it does not, or if you decide not to discard this projection after all, press Return (PF3) to go back to the Projection panel – into which you can type a different projection name – or Home (PF12) to go to the Home panel.

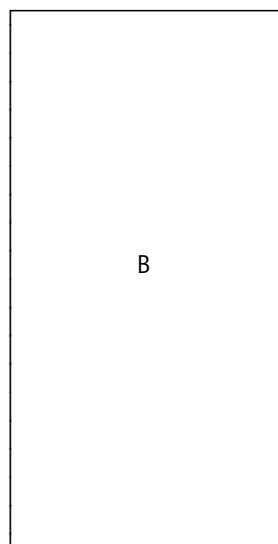
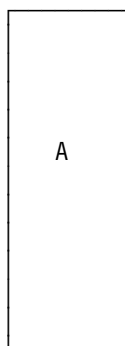
discarding a projection

Test page

Scan this page using the default scan parameters. Then use the Edit panel to perform the tasks written above the boxes. Before starting to edit, you should save the image using the Save Image panel. This will allow you to recover from editing mistakes by loading from disk, which is usually quicker than rescanning the page.

Take this text, scale it so that it completely fills box B, and negate it.

Take this text, rotate it, and then position it in Box A.



As an alternative to editing, all the necessary steps can be done with just one projection. You will need to create and manipulate one extract for each piece of text using the Create Projection, Extract Source, Extract Transform, and Extract Target panels.

To test the projection, clear the screen and then select the View Image panel. Enter the name of the image of this page and the name of the projection you have just created.

Check whether the scanned page has been manipulated according to the instructions above. If the result is not quite what you had expected, perhaps you made a mistake in your projection definition. Try again, if necessary referring to the relevant sections in this manual and to help panels.

When you have created the required projection, you could print your results using the Print Image panel. If you want to use the projection in a later GDDM-IVU session, make sure you save it using the Save Projection panel.

Chapter 3. Application programming

General-use programming interface

This chapter is intended to help you write application programs. It defines a general-use programming interface that allows you to write programs that use the services of GDDM-IVU.

Like all GDDM application-programming interfaces, the API to GDDM-IVU is provided by subroutine calls. The calls invoke GDDM-IVU, pass images and projections to it, and control the way it operates.

The applications can be run under TSO, CMS, or CICS. They can be coded in any of the languages generally allowed for GDDM programs. These are, principally, PL/I, FORTRAN, COBOL, the C part of the C/C++ product, C/370, REXX, H Assembler, and High Level Assembler. There are also special interfaces to APL2 and BASIC, using software provided by these languages.

A program that calls GDDM-IVU must itself be invoked from a terminal: batch-mode execution is not supported. When called by an application, GDDM-IVU can operate exactly as if invoked directly from the terminal. Unless the application specifies otherwise, the Home panel appears on the screen first, and the full range of GDDM-IVU functions is available from the menu panels and PF (program function) keys. When the user ends GDDM-IVU by redisplaying the Home panel and pressing PF3 twice, control returns to the application.

These standard GDDM-IVU functions can be tailored by the application in several ways. For example:

- A panel other than the Home panel can be displayed first.
- The PF key settings can be altered or disabled, which can be used to limit the function available to the user.
- Images and projections can be passed between the application and GDDM-IVU.
- A limit can be set on the number of images and projections held by GDDM-IVU in main storage.
- The definitions of the menu and help panels can be tailored.

The appropriate GDDM Base product (GDDM/MVS, GDDM/VM, or GDDM/VSE) is a prerequisite for GDDM-IVU. You do not need to understand the Base API to write programs that call and tailor GDDM-IVU. For advanced applications, however, you might need to use some GDDM Base calls, in particular those that provide image-handling functions. All GDDM Base calls are described in the *GDDM Base Application Programming Reference* book. Guidance on how to use them, including a large number of examples, is given in the *GDDM Base Application Programming Guide*.

An introduction to the GDDM-IVU calls

There are five groups of GDDM-IVU calls, each of which is discussed briefly in this section. The syntax of the GDDM-IVU calls is provided in Appendix A, “GDDM-IVU calls” on page 69.

Initialization and termination of GDDM-IVU:

IUINIT Initialize GDDM-IVU
IUTERM Terminate GDDM-IVU

These calls initialize, or reinitialize, and release GDDM-IVU storage. IUINIT sets all fields on all menu panels to their default values. IUTERM deletes all images and projections known to GDDM-IVU. If GDDM-IVU is called once only, neither of these calls is essential.

Invocation of GDDM-IVU:

IUIVU Invoke GDDM-IVU

Image-name-table handling: The following calls enable applications to pass images to, and receive them from, GDDM-IVU:

IUAIMG Add image to the name table
IUDIMG Delete image from the name table
IUQIID Query image identifier
IUQINM Query image name
IUMLO Set the maximum number of loaded objects
IUQMLO Query the maximum number of loaded objects
IUQLO Query the currently loaded object identifiers and names
IUQNLO Query the number of loaded objects

The last four calls in the list apply to both the projection table and the image table.

GDDM-IVU processes images in GDDM Base image format. While in main storage, images have fullword-integer identifiers. However, GDDM-IVU users identify images at the terminal by eight-character names. The image table is thus a list of all images available to GDDM-IVU, containing both the GDDM identifier and the GDDM-IVU name of each image.

There are calls for adding images to the table and deleting them from it, limiting the size of the table, and hence the number of images that the user is allowed to handle at one time, and querying the table's contents.

Projection-name-table handling: The following calls enable applications to pass projections to, and receive them from, GDDM-IVU:

IUAPRJ Add projection to the name table
IUDPRJ Delete projection from the name table
IUQPID Query projection identifier
IUQPNM Query projection name

Like images, projections are known to GDDM by fullword-integer identifiers and to GDDM-IVU users at the terminal by eight-character names. The projection table has a similar purpose and contents to the image table.

PF (program function) key setting:

IUPFK Set PF key definitions

IUQPFK Query PF key definitions

IUPFK allows programs to assign any GDDM-IVU PF key function to a different PF key from the default, or to disable any PF key.

General application-programming guidance

This section describes the principles of programming with GDDM-IVU calls, illustrated with example code. Some concepts and principles applying to all GDDM application programming are also summarized. If you need more information about these, refer to the *GDDM Base Application Programming Reference* book. or the *GDDM Base Application Programming Guide*.

Calling GDDM-IVU

Here is the simplest possible GDDM-IVU application program, coded in PL/I:

```
IVUCALL: PROC OPTIONS(MAIN);

CALL FSINIT;
CALL IUIVU(0,' ',' ');
CALL FSTERM;

%INCLUDE ADMUPINF;
%INCLUDE ADMUPIN5;
END;
```

An FSINIT call must always be the first GDDM call. It is a GDDM Base call that initializes GDDM.

The FSTERM call terminates GDDM. It is not essential, but is recommended as the last GDDM call issued to ensure that all main storage allocated by GDDM is released.

The single GDDM-IVU call, IUIVU, invokes GDDM-IVU to run in the standard way.

In PL/I and C/370, each GDDM call name used by a program should be declared. This is the reason for the %INCLUDE statements in the PL/I example above. A set of files containing the required PL/I and C/370 entry declarations is supplied for inclusion in application programs. For PL/I, each file has a name of the form ADMUPIN_x. For C/370, each file has a name of the form ADMUCIN_x⁴. For both PL/I and C/370, the “x” value stands for the first letter of the names of the calls declared within it. ADMUCINA, for example, includes C/370 declarations of all GDDM Base calls beginning with the letter “A.”

⁴ For reentrant entry points, the PL/I files have names of the form ADMUPIR_x. The C/370 files have names of the form ADMUCIR_x.

The GDDM-IVU PL/I call declarations are contained in ADMUPIN5 (ADMUPIR5 for reentrant entry points). The GDDM-IVU C/370 call declarations are contained in ADMUCIN5 (ADMUCIR5 for reentrant entry points). These files are subject to change and rearrangement as they are dependent on interfaces with other IBM products.

Here is a program that does rather more than the previous example:

```
IVUCALL: PROC OPTIONS(MAIN);

CALL FSINIT;
CALL IUIVU(12,'NEWIMAGE',' ');
CALL FSTERM;

%INCLUDE ADMUPINF;
%INCLUDE ADMUPIN5;
END;
```

This time, the IUIVU call specifies a panel other than the Home panel as the first one to be displayed. The parameter value 12 selects the Load Image panel as the first.

The second IUIVU parameter, 'NEWIMAGE', is the name with which the image-name field on this panel is to be initialized.

The last, blank, parameter on the IUIVU call is for specifying a name with which to initialize the Projection field on the panel.

When this program is run, GDDM-IVU begins by displaying the Load Image panel with NEWIMAGE in the image-name field. If the user presses Enter, GDDM-IVU loads the image. If an image of that name is not found in a disk file, GDDM-IVU issues an error message.

Notice that the image is loaded when the terminal user interacts with the Load Image panel by pressing Enter, rather than when the IUIVU call is executed. GDDM-IVU calls never act directly upon images: user action through a GDDM-IVU panel is always required.

GDDM-IVU call syntax

GDDM-IVU supports application programs written in APL2, System/370 Assembler, BASIC, COBOL, C/370, FORTRAN, PL/I, and REXX.

The following examples shows how a GDDM-IVU call is made in each of the supported languages. The APL code used in APL2 and the RCP code used in BASIC are given in the header box for each call in Appendix A, "GDDM-IVU calls" on page 69.

The call used in this example is the IUMLO call, for which the syntax is:

IUMLO(type,count)

APL2 IUMLO GDMX 1 3

Assembler CALL IUMLO,(TYPE,COUNT),VL

 TYPE DC F'1'

```

COUNT DC F'3'

BASIC      100    IUMLO=1141374976
           110    CALL GDDM (IUMLO,1,3)

COBOL      MOVE 1 TO TYPE.
           MOVE 3 TO COUNT.
           CALL 'IUMLO' USING TYPE COUNT.

FORTRAN    CALL IUMLO(1,3)

PL/I       CALL IUMLO(1,3);

C/370      iumlo(1,3);

REXX       'IUMLO 1 3'
```

Setting PF keys

An application program can disable PF keys or reassign their functions. For example, the call below, if inserted before the IUIVU call in the previous example, would prevent access to the image-output functions using PF6 or PF18, and to the Home panel using PF12 or PF24.

```

DECLARE
  KEYS(24)      FIXED BINARY(31) INIT(1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,
                                     13,14,15,16,17,18,19,20,21,22,23,24),
  FUNCTIONS(24) FIXED BINARY(31) INIT(0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,-1,
                                     0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,-1);
CALL IUPFK(24,KEYS,FUNCTIONS);
```

The IUPFK call requires two arrays as its last two parameters, and a count of the number of elements in them as the first parameter. The first array is a set of PF key numbers, the second a set of codes for the function to be assigned to each key. A code of 0 means the default function. Omitting a key from the arrays, as in the next example, has the same effect. A value of -1 means no-operation, and here has been assigned to the keys that would provide the output and “Home” panel functions by default.

Other codes can be used in the second array to specify particular functions. For example, this puts the Menu function on PF10 as well as PF2, and the Return function on PF12 and PF24 as well as PF3, thereby disabling the Home panel:

```

DECLARE
  KEYS(3)      FIXED BINARY(31) INIT(10,12,24),
  FUNCTIONS(3) FIXED BINARY(31) INIT( 2, 3, 3);
CALL IUPFK(3,KEYS,FUNCTIONS);
```

Using the image and projections tables

This code passes three images and a projection to GDDM-IVU:

```
CALL LDIMAG;                      /* Call to internal subroutine */

CALL IUAIMG(101,'IMAGE01');
CALL IUAIMG(102,'IMAGE02');
CALL IUAIMG(103,'IMAGE03');

CALL CREATEPROJ;                  /* Call to internal subroutine */

CALL IUAPRJ(11,'APROJ');

CALL IUIVU(5,'IMAGE01','APROJ');
```

The images are loaded from disk by the program in a subroutine called LDIMAG. It gives them identifiers of 101, 102, 103. They are then added to the image table and named IMAGE01, IMAGE02, and IMAGE03 in three IUAIMG calls. The projection is created by a subroutine called CREATEPROJ, with the identifier 11. It is added to the projection table and named APROJ in an IUAPRJ call. The subroutines contain no GDDM-IVU calls, so their source code is not shown.

GDDM-IVU is invoked with an IUIVU call specifying the View Image panel, with the image-name field set to the name of the first image, and the projection field set to the name of the projection. If the terminal user presses Enter when the view panel is displayed, GDDM-IVU applies APROJ to IMAGE01 and displays the result.

The images IMAGE02 or IMAGE03 are also available to the user without going through the GDDM-IVU load process. To view these images, the user has only to change the image name on the “View Image” panel and press Enter.

The user can view any of the three images without the projection being applied by erasing its name from the projection field. Alternatively, APROJ can be applied to any other image that the user may load or create using GDDM-IVU.

Relation to other GDDM output

GDDM-IVU calls can be freely mixed with other GDDM calls. However, an application cannot put anything on the screen (neither alphanumerics, graphics, text, nor another image) at the same time as GDDM-IVU. The same restriction applies to printed and plotted output.

The reason is that GDDM-IVU creates its own partition set, and therefore has exclusive use of all objects in the GDDM hierarchy, including GDDM pages, below that level. The GDDM hierarchy is explained in the *GDDM Base Application Programming Guide*.

Multiple calls to GDDM-IVU

An application program can issue more than one IUIVU call. If no IUNIT, IUTERM, FSINIT, or FSTERM call is issued in between, the state of GDDM-IVU remains unchanged from one IUIVU call to the next, apart from changes made by the application, such as updating the name tables. To the terminal user, the repeated IUIVU calls appear as a single GDDM-IVU session.

There are GDDM Base calls that alter the definition of the terminal with which GDDM-IVU communicates (the primary device, in GDDM Base terms). If the

definition is altered between calls to GDDM-IVU, the results are unpredictable. The Base calls in question are DSOPEN, DSCLS, DSUSE, and DSDROP. They should not be executed between IUIVU calls, unless an IUINIT or IUTERM call is also executed.

A dialog with the user

Here is a program that uses GDDM Base and GDDM-IVU calls to hold a dialog with the terminal user.

```

IMAG11: PROC OPTIONS(MAIN);
/*****
/*      Establish PF keys 1,3,and 12 only as usable.          */
/*      Handle user input of a document name, then invoke GDDM-IVU  */
/*      to allow scanning and saving of multiple pages until    */
/*      the user indicates document end.                        */
/*      The program generates a sequence of image names based   */
/*      on the document name chosen, and uses only one image    */
/*      at a time in the GDDM-IVU name table.                  */
*****/
DCL (ATTTYPE,ATTVAL,COUNT) FIXED BIN(31);
DCL DOC_NAME CHAR(7);          /* document name          */
DCL PAGE PIC'9';              /* for image naming       */
DCL IM_NAME CHAR(8);          /* image name             */
DCL IMGID FIXED BIN(31);      /* image identifier       */
DCL DOC_END BIT(1);           /* on for end of document */
DCL KEY_ARRAY(12) FIXED BIN(31) /* PF key numbers        */
    INIT(1,2,3,4,5,6,7,8,9,10,11,12);
DCL FUN_ARRAY(12) FIXED BIN(31) /* PF key functions      */
    INIT(0,-1,0,-1,-1,-1,-1,-1,-1,-1,-1,0);
                                /* 0 default  -1 nop    */
                                /* use only keys 1,3,and 12 */
CALL FSINIT;
CALL ASDFLD(1,1,1,1,60,2);      /* define output (prompt) field */
CALL ASDFLD(2,1,65,1,7,0);      /* doct.name input field        */
CALL ASCPUT(1,37,'Specify document name (7 characters):');
CALL ASFCUR(2,1,1);             /* position cursor              */
CALL ASREAD(ATTTYPE,ATTVAL,COUNT); /* do the alphanumeric i/o      */
CALL ASCGET(2,7,DOC_NAME);      /* get document name            */
CALL IUPFK(12,KEY_ARRAY,FUN_ARRAY); /* set up PF keys              */
CALL IUML0(1,1);                /* permit 1 image only in the  */
                                /* image name table             */
PAGE=0;
CALL ASCPUT(2,7,'          '); /* clear input field            */
DOC_END='0'B;
DO UNTIL (DOC_END);             /* image process loop          */
    PAGE=PAGE+1;
    IM_NAME=DOC_NAME||PAGE;      /* generate image name          */
    CALL IUIVU(4,IM_NAME,'      '); /* display scan panel          */
    CALL IUIVU(13,'=          '); /* display save panel           */
    CALL IUQIID(IM_NAME,IMGID);  /* get image identifier         */
    CALL IUDIMG(IMGID,IM_NAME);  /* delete image from            */
                                /* name table                   */
CALL ASCPUT(1,55,
    'Press ENTER to continue scanning or PF3 to end document');
CALL ASREAD(ATTTYPE,ATTVAL,COUNT);
IF ATTTYPE=0 THEN;              /* continue                    */

```

```

ELSE IF ATTYPE=1 THEN
  IF ATTVAL=3 THEN DOC_END='1'B; /* end of document          */
    ELSE;                /* (continue)          */
    ELSE;                /* (continue)          */
END;                      /* image process loop  */
CALL ASCPUT(1,27,'Document scanning completed');
CALL ASREAD(ATTYPE,ATTVAL,COUNT);
CALL FSTERM;
%INCLUDE ADMUPINA;
%INCLUDE ADMUPINF;
%INCLUDE ADMUPIN5;

END IMAG11;

```

Image-indexing example programs

This is a suite of four programs, written in PL/I, that demonstrate image indexing, which is the combining, for disk storage and retrieval, of alphanumeric and image data. The alphanumeric data is a sequence of reference numbers and names. This might in reality be part of a personnel or parts database, in which a person or part name is associated with a unique reference number. The image data is created by scanning. Both types of data are stored together in a single file.

The first of the four programs is IMAG12F, which creates a small file of numbers and names, from compiled-in data.

The second program is IMAG12P, which creates a projection used by the principal program IMAG12.

The third program IMAG12 reads the file created by IMAG12F, and enables the user to scan an image, such as a passport photograph on a form, to be associated with a particular number and name. Scanning is done by calling GDDM-IVU. The combined alphanumeric and image data is then written out to another sequential file. Alternatively, a given number or name may be skipped, or the program halted.

The fourth program IMAG13 does the reverse of IMAG12. It reads back the file written by IMAG12, displays the alphanumeric data, and gives the user the options of viewing the image (by GDDM-IVU call), or skipping to the next, or halting.

The programs use the following files:

IMAG12F: One output file called OUTPUT.

IMAG12: One output file called OUTPUT, and the output file created by IMAG12F, now an input file called INPUT

IMAG13: One output file called OUTPUT

No files are required for program IMAG12P.

Program IMAG12F

```

IMAG12F: PROC OPTIONS(MAIN);
/*****
/*      FUNCTION                                */
/*      Create a file of numbers and names for use by IMAG12          */
/*      FILES                                                                */
/*      The output file is named OUTPUT                                */
*****/

```

```

/* DCLs */

DCL OUTPUT FILE OUTPUT RECORD ENV(F RECSIZE(30));
DCL 1 OUTREC,
      2 REFNO PIC'999999',          /* reference number */
      2 NAME CHAR(24);              /* name */

DCL ARRAY1(5) PIC'999999' STATIC
      INIT ('012345',
            '055555',
            '066666',
            '077888',
            '098765');

DCL ARRAY2(5) CHAR(24) STATIC
      INIT ('TCMAVERICK', /* Note */
            'PEPENGELLY', /* These names must */
            'RWPENNEY',   /* contain no */
            'HCSMITH',    /* embedded blanks */
            'BJTINDALL',  /* (for IMAG12 use) */

/* Process */
OPEN FILE(OUTPUT);
DO I=1 TO 5;
  OUTREC.REFNO=ARRAY1(I);
  OUTREC.NAME =ARRAY2(I);
  WRITE FILE(OUTPUT) FROM (OUTREC);
END;
CLOSE FILE(OUTPUT);

END IMAG12F;

```

Program IMAG12P

```

IMAG12P: PROC OPTIONS (MAIN);
/*****
/*
/*      FUNCTION
/*      Creates a 1.5 inch by 2.0 inch extract projection
/*      named PHOTOEX, for use by IMAG12.
/*
/*      FILES
/*      None
/*
*****/

```

```

/* DCLs                                                                    */
DCL PROJID FIXED BIN(31);          /* projection identifier */
DCL (LEFT,RIGHT,TOP,BOTTOM) FLOAT DEC(6);

/* Process                                                                    */
CALL FSINIT;
CALL IMPGID(PROJID);
CALL IMPCRT(PROJID);
LEFT=0.0;                          /* corresponds to a passport- */
RIGHT=1.5;                          /* sized photo in the top */
TOP=0.0;                            /* left corner of the scanned*/
BOTTOM=2.0;                         /* form */
CALL IMREXR(PROJID,0,LEFT,RIGHT,TOP,BOTTOM);
CALL IMRPLR(PROJID,0,0.0,0.0,0);
CALL IMPSAV(PROJID,'PHOTOEX ',25,'1.5 by 2.0 in. extraction',0);
CALL FSTERM;

/* GDDM entry point DCLs                                                    */
%INCLUDE ADMUPINF;
%INCLUDE ADMUPINI;
END IMAG12P;

```

Program IMAG12

```

IMAG12: PROC OPTIONS(MAIN);
/*****
/*
/* FUNCTION
/* This is a simple image 'indexing' application.
/* An existing file of (e.g. personnel) reference numbers and
/* names is read, and the user prompted to initiate scanning
/* (by the IVU scan process)
/* of a photograph corresponding to the reference number and
/* name.
/* For each satisfactorily scanned image, a group of records
/* are written to an output file, comprising a number/name
/* record, followed by one or more records containing the
/* image plus the reference number.
/* The above is a simple representation of a customer's
/* existing alphanumeric database, and an alpha plus image
/* database.
/*
/* The input file must consist of fixed length records each of
/* 30 bytes.
/* The first 6 bytes contain a 6-digit reference number and
/* the remaining 24 bytes contain a name or part description.
/* Such a file may be created using the program IMAG12F.
/*
/* This file is read sequentially. For each input record, a
/* screen panel is displayed, showing the reference number and
/* name as in the input file.
/* The end user interacts with this panel to select either:
/* (1) scan the image (by invoking the IVU scan process),
/* or
/* (2) skip to next input record,
/* or

```

```

/*      (3) stop.                                     */
/*
/*      When scan is selected, the IVU scan process is invoked,
/*      with a restriction of PF key use to PF3 only.
/*      On return from the IVU, the image is assumed to have been
/*      successfully scanned, and is obtained from GDDM using the
/*      Base API IMAGTx calls.
/*      It is written to an output file preceded by an 'image
/*      header' record, containing the reference number and name.
/*      Also, the image data record(s) contain the reference number.
/*
/*      The output file contains 3 record types:
/*      Type 01 - image header record. The reference number and
/*      name fields are as in the input file (IMAG12F output).
/*      Type 02 - image data record, containing 6-byte reference
/*      number, followed by image data length field (2 bytes),
/*      followed by image data.
/*      CPDS format, 3800 compression, has arbitrarily been chosen
/*      for storing the image data. This format and compression
/*      are assumed by the complementary program IMAG13.
/*      Type 03 - as type 02, but signifying the last image data
/*      record, for a given image.
/*      Image data is variable length, up to 400 bytes.
/*
/*      Depending on image size and complexity, one or more type 02
/*      records may be written; one and only one type 03 record
/*      will be written, all per image scanned, and for which there
/*      will also be (preceding these image data records) a single
/*      type 01 header record.
/*
/*      In a customer application, different data structures will
/*      probably be chosen, to suit direct access, possibly by use
/*      of the reference number as a record key.
/*
/*      In order to minimize the image data volume for testing,
/*      an image extraction projection named 'PHOTOEX' is invoked
/*      (actually, supplied as the IVU default projection name,)
/*      at the IVU scan stage.
/*      This projection may be created by use of the IMAG12P program.
/*
/*      No input data validation is provided.
/*
/*      FILES
/*      The input file is named INPUT
/*      The output file is named OUTPUT
/*
/*****

```

```

/*****
/*      DCLs                                     */
/*****

```

DCL INPUT FILE INPUT RECORD;

DCL OUTPUT FILE OUTPUT RECORD ENV(V RECSIZE(416));

```

DCL 1 INREC,                                     /* input file record format */
      2 REFNO PIC'999999',                       /* reference number field   */

```

```

        2 NAME CHAR(24);           /* name field */
DCL 1 OUTBUF1,                     /* output data buffer 1 */
        2 RECTYPE CHAR(2),        /* record type field */
        2 REFNO PIC'999999',      /* reference number field */
        2 DLEN FIXED BIN(15),    /* image data length */
        2 DATA CHAR(400) VARYING; /* depends on record type */

DCL 1 OUTBUF2,                     /* output data buffer 2 */
        2 RECTYPE CHAR(2),        /* record type field */
        2 REFNO PIC'999999',      /* reference number field */
        2 DLEN FIXED BIN(15),    /* image data length */
        2 DATA CHAR(400) VARYING; /* depends on record type */

DCL 1 OUTREC BASED(P),            /* output record format */
        2 RECTYPE CHAR(2),        /* record type field */
                                   /* '01','02', or '03' */
        2 REFNO PIC'999999',      /* reference number field */
        2 DLEN FIXED BIN(15),    /* image data length */
                                   /* 0 for type 01 record */
        2 DATASUB CHAR(400) VARYING; /* name or image data */
                                   /* type 01: name field */
                                   /* type 02: image data */
                                   /* type 03: image data, final */

DCL ADDR BUILTIN;                 /* address function */
DCL SUBSTR BUILTIN;              /* substring function */

DCL STOP BIT(1);                 /* file process switch */
                                   /* '0'B to continue, */
                                   /* '1'B to stop */

DCL (ATTTYPE,ATTVAL,COUNT) FIXED BIN(31); /* for ASREAD */
DCL INSEL CHAR(1);               /* should be 1,2, or 3 */
DCL IMGNAME CHAR(8);             /* shortened image name */
DCL IMGID FIXED BIN(31);         /* image identifier */
DCL DATALEN FIXED BIN(31);      /* length of data in databuf */
DCL BUFLen FIXED BIN(31) INIT(400); /* length of databuf */
DCL DATABUF CHAR(400);           /* for IMAGT (not varying) */
DCL (P,P1,P2) POINTER;          /* output buffer pointers */
DCL KEY_ARRAY(12) FIXED BIN(31) /* PF key numbers */
    INIT( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);
DCL FUN_ARRAY(12) FIXED BIN(31) /* PF key functions */
    INIT(-1,-1, 0,-1,-1,-1,-1,-1,-1, -1, -1, -1);
                                   /* 0 default -1 nop */
DCL INREC_HOLD CHAR(6);          /* Temporary field */

/*****
/*  INITIALIZATION */
*****/

/* Establish on ENDFILE unit for input file */
ON ENDFILE(INPUT) BEGIN;         /* Endfile on-unit */
    STOP='1'B;
                                END; /* Endfile on-unit */

/* Open files and establish output buffer pointers */
OPEN FILE(INPUT);
OPEN FILE(OUTPUT);
P1=ADDR(OUTBUF1);

```

```

P2=ADDR(OUTBUF2);

/* Initialize GDDM */
CALL FSINIT;

/* Establish display panel */
CALL ASDFLD(2,2,2,1,15,2); /* Output Refno descriptor */
CALL ASCPUT(2,15,'Reference No.: ');
CALL ASDFLD(3,2,20,1,6,2); /* Output Refno data */
CALL ASDFLD(4,3,2,1,15,2); /* Output Name descriptor */
CALL ASCPUT(4,15,'Name : ');
CALL ASDFLD(5,3,20,1,24,2); /* Output Name data */

CALL ASDFLD(6,5,2,5,60,2); /* Output prompt field */
CALL ASDFLD(7,11,10,1,1,0); /* Input selection field */

/* Permit use of IVU PF key 3 (return) only */
CALL IUPFK(12,KEY_ARRAY,FUN_ARRAY);

/*****
/* MAIN PROCESSING */
*****/

/* Do file process until endfile or stop request */
STOP='0'B;
DO WHILE(~STOP); /* file process loop */

/* Read input file */
READ FILE(INPUT) INTO(INREC); /* read an input record */

/* If not stop (endfile), do name process */
IF (~STOP) THEN
DO; /* name process */
/* Display the refno. and name just read in */
INREC_HOLD=INREC.REFNO;
CALL ASCPUT(3,6,INREC_HOLD);
CALL ASCPUT(5,24,INREC.NAME);
/* Get user to select 'Scan', 'skip', or 'stop' */
CALL ASCPUT(6,300,
'Type in one of the following and press ENTER:
'      1   to scan image for this name
'      2   to skip to next name
'      3   to stop
');
CALL ASFCUR(7,1,1); /* position cursor */
CALL ASREAD(ATTTYPE,ATTVAL,COUNT); /* do the i/o */
CALL ASCGET(7,1,INSEL); /* get value typed in */

/* Select on result */
SELECT (INSEL);

/* When 'stop', set stop flag */
WHEN ('3') STOP='1'B;

/* When 'skip', skip scan-write process */
WHEN ('2') ;

/* When 'scan', do scan-write process */

```

```

WHEN ('1')
DO;                                /* scan-write process */
/* Write output file name/number record (type 01) */
P=P1;                              /* use buffer 1 */
OUTREC.RECTYPE='01'; /* record type 01 */
OUTREC.REFNO=INREC.REFNO; /* P1->OUTREC */
OUTREC.DLEN =0; /* not used in type 01 record */
OUTREC.DATASUB=INREC.NAME;
WRITE FILE(OUTPUT) FROM(OUTREC);

/* Call for scan image till ok */
IMGNAME=SUBSTR(INREC.NAME,1,8);
CALL IUIVU(4,IMGNAME,'PHOTOEX ');

/* Get id of scanned image */
CALL IUQIID(IMGNAME,IMGID);

/* initialize image 'get' */
CALL IMAGTS(IMGID,0,3,4);/* get the image with */
/* identity projection, */
/* CPDS format, */
/* 3800 compression */

/* get first image buffer */
CALL IMAGT(IMGID,BUFLEN,DATABUF,DATALEN);
OUTREC.DLEN=DATALEN; /* length of image data */
OUTREC.DATASUB=SUBSTR(DATABUF,1,DATALEN);

P=P2; /* switch output buffers */
OUTREC.REFNO=INREC.REFNO; /* P2->OUTREC */
/* Do until end of image data */
DO UNTIL(DATALEN=0); /* fetch till end of data */

/* get next image buffer */
CALL IMAGT(IMGID,BUFLEN,DATABUF,DATALEN);
IF DATALEN>0 THEN /* image data is there */
DO; /* image data is there */
OUTREC.DLEN=DATALEN; /* length of image data */
OUTREC.DATASUB=SUBSTR(DATABUF,1,DATALEN);
END; /* image data is there */
ELSE; /* end of image data */

/* write image file record (type 02 or 03) */
/* from the other output buffer */
IF P=P2 THEN P=P1; /* switch from 2 to 1 */
ELSE P=P2; /* switch from 1 to 2 */
IF DATALEN=0
THEN OUTREC.RECTYPE='03';/* last image buffer */
ELSE OUTREC.RECTYPE='02';/* not last buffer */
WRITE FILE(OUTPUT) FROM(OUTREC);
END; /* fetch till end of data */

/* Terminate image 'get' */
CALL IMAGTE(IMGID);

/* Delete image from name table */
CALL IUDIMG(IMGID,IMGNAME);

```

```

        END;                                /* scan-write process      */
    OTHERWISE ;
    END;                                /* select group                */

    /* End name process                                */
    END;                                /* name process                */
    /* Else stop (endfile)                            */
ELSE;

END;                                /* file process loop          */

/*****
/*  TERMINATION                                     */
*****/

/* Close files                                         */
CLOSE FILE(INPUT);
CLOSE FILE(OUTPUT);

/* terminate GDDM                                     */
CALL FSTERM;

/*****
/*  GDDM ENTRY POINT DCLS                           */
*****/

%INCLUDE ADMUPINA;
%INCLUDE ADMUPINF;
%INCLUDE ADMUPINI;
%INCLUDE ADMUPIN5;

END IMAG12;

```

Program IMAG13

```

IMAG13: PROC OPTIONS(MAIN);
/*****
/*
/*  FUNCTION                                         */
/*
/*  This program complements IMAG12, and requires as input
/*  the output file from that program.
/*  (See IMAG12 Prologue for a description of the file record
/*  types and formats.)
/*  This program reads back the IMAG12 output file which
/*  contains alphanumeric and image data. It uses the alpha
/*  data in a displayed panel, like that of IMAG12, to permit
/*  the user to choose either:
/*  (1) to view the image (using the IVU View process),
/*  or
/*  (2) to skip to the next image header,
/*  or
/*  (3) to stop.
/*

```

```

/* The input file is read sequentially - a real application */
/* might use direct access methods to retrieve the desired */
/* image by use of the reference number. */
/*
/* For each image selected to be viewed, the IVU view process */
/* is called after retrieving and passing to GDDM the complete */
/* image, which is required to be in CPDS format, 3800 */
/* compression, as provided by IMAG12. */
/* For IVU purposes, the image name as in the file is truncated */
/* to 8 bytes. No projection is invoked in the view process. */
/* The user is restricted to use of PF 3 only (but must first */
/* press ENTER to update the image). */
/* When the end user returns from the IVU by pressing PF3, */
/* the next input file header record is processed, and so on. */
/*
/* This program depends on correct record type 01/02/03 */
/* sequence in the input file. Incorrect sequence will be */
/* detected, causing an error message and the program */
/* terminates. */
/* No other input data validation is provided. */
/*
/* FILES */
/* The input file is named INPUT. */
/*
/*****

```

```

/*****
/* DCLs */
/*****

```

DCL INPUT FILE INPUT RECORD;

```

DCL 1 INBUF, /* input data buffer */
    2 RECTYPE CHAR(2), /* record type field */
    /* '01','02', or '03' */
    2 REFNO PIC'999999', /* reference number field */
    2 DLEN FIXED BIN(15), /* image data length field */
    /* 0 for type 01 record */
    2 DATA CHAR(400) VARYING; /* depends on record type */
    /* type 01: name field */
    /* type 02: image data */
    /* type 03: image data, final */

DCL SUBSTR BUILTIN; /* substring function */

DCL STOP BIT(1); /* file process switch */
    /* '0'B to continue, */
    /* '1'B to stop */

DCL (ATTTYPE,ATTVAL,COUNT) FIXED BIN(31); /* for ASREAD */
DCL INSEL CHAR(1); /* should be 1,2, or 3 */
DCL INAME CHAR(24); /* image name as in file */
DCL IMGNAME CHAR(8); /* shortened image name */
DCL IMGID FIXED BIN(31); /* image identifier */
DCL DATALEN FIXED BIN(31); /* length of data in databuf */
DCL BUFLen FIXED BIN(31) INIT(400); /* length of databuf */
DCL DATABUF CHAR(400); /* for IMAPT (not varying) */
DCL KEY_ARRAY(12) FIXED BIN(31) /* PF key numbers */

```

```

        INIT( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);
DCL FUN_ARRAY(12) FIXED BIN(31) /* PF key functions */
        INIT(-1,-1, 0,-1,-1,-1,-1,-1,-1, -1, -1, -1);
                                /* 0 default -1 nop */
DCL INBUF_HOLD CHAR(6);        /* Temporary field */
/*****
/*  INITIALIZATION */
*****/

/* Establish on ENDFILE unit for input file */
ON ENDFILE(INPUT) BEGIN;      /* Endfile on-unit */
    STOP='1'B;
                                END;          /* Endfile on-unit */

/* Open file */
OPEN FILE(INPUT);

/* Initialize GDDM */
CALL FSINIT;

/* Establish display panel */
CALL ASDFLD(1,1,2,1,79,2);    /* Message field */
CALL ASDFLD(2,2,2,1,15,2);    /* Output Refno descriptor */
CALL ASCPUT(2,15,'Reference No.: ');
CALL ASDFLD(3,2,20,1,6,2);    /* Output Refno data */
CALL ASDFLD(4,3,2,1,15,2);    /* Output Name descriptor */
CALL ASCPUT(4,15,'Name : ');
CALL ASDFLD(5,3,20,1,24,2);   /* Output Name data */

CALL ASDFLD(6,5,2,5,60,2);    /* Output prompt field */
CALL ASDFLD(7,11,10,1,1,0);   /* Input selection field */

/* Permit use of IVU PF key 3 (return) only */
CALL IUPFK(12,KEY_ARRAY,FUN_ARRAY);

/*****
/*  MAIN PROCESSING */
*****/

/* Do file process until endfile or stop request */
STOP='0'B;
LOOP: DO WHILE(~STOP);        /* file process loop */

    /* Read input file */
    READ FILE(INPUT) INTO(INBUF); /* read an input record */

    /* If not stop (endfile), do name process */
    IF (~STOP) THEN
    DO;                        /* name process */
        /* Check that record type is 01 */
        IF INBUF.RECTYPE ='01'
        THEN
        DO;                    /* get user to choose action */
            /* Display the refno. and name just read in */
            INBUF_HOLD=INBUF.REFNO;
            CALL ASCPUT(3,6,INBUF_HOLD);
            INAME=INBUF.DATA;    /* image name in full */
            CALL ASCPUT(5,24,INAME); /* show name */

```

```

/* Get user to select 'View', 'skip', or 'stop' */
CALL ASCPUT(6,300,
'Type in one of the following and press ENTER:
'      1      to view image for this name
'      2      to skip to next name
'      3      to stop
');

CALL ASFCUR(7,1,1);
CALL ASREAD(ATTTYPE,ATTVAL,COUNT); /* do the i/o */
CALL ASCGET(7,1,INSEL);

END; /* get user to choose action */
ELSE
DO; /* error exit */
/* display error message and exit from file process loop*/
CALL ASCPUT(1,27,'Input file sequence error 1');
CALL ASREAD(ATTTYPE,ATTVAL,COUNT);
LEAVE LOOP;
END; /* error exit */

/* Select on result */
SELECT (INSEL);

/* When 'stop', set stop flag */
WHEN ('3') STOP='1'B;

/* When 'skip', skip to end of current image data */
WHEN ('2')
DO UNTIL(INBUF.RECTYPE='03'|STOP='1'B);
READ FILE(INPUT) INTO(INBUF);
END;

/* When 'view', do read-view process */
WHEN ('1')
DO; /* read-view process */

/* initialize image 'put' */
CALL IMAGID(IMGID); /* get an unused image identifier*/
CALL IMAPTS(IMGID,0,3,4);/* put the image with
/* identity projection,
/* CPDS format,
/* 3800 compression

/* Do until end of image data */
DO UNTIL /* fetch till end of data */
(INBUF.RECTYPE='03');
/* read image file record (type 02 or 03) */
READ FILE(INPUT) INTO(INBUF);
/* if endfile is hit here this is a file sequence error */
/* error */
IF STOP THEN DO; /* endfile - sequence error */
CALL ASCPUT
(1,27,'Input file sequence error 2');
CALL ASREAD(ATTTYPE,ATTVAL,COUNT);
LEAVE LOOP; /* exit from file process loop*/
END; /* endfile - sequence error */

/* 'Put' image buffer */
DATALEN=INBUF.DLEN; /* image data length

```

```

        DATABUF=INBUF.DATA; /* not varying for IMAPT */
        CALL IMAPT(IMGID,DATALEN,DATABUF);
    END; /* fetch till end of data */

    /* Terminate image 'put' */
    CALL IMAPTE(IMGID);

    /* add image to the name table */
    IMGNAME=SUBSTR(INAME,1,8); /* shortened image name */
    CALL IUAIMG(IMGID,IMGNAME);

    /* call view image */
    CALL IUIVU(5,IMGNAME,' ');

    /* Delete image from name table */
    CALL IUDIMG(IMGID,IMGNAME);

    END; /* read-view process */

    OTHERWISE /* (should trap input data */
              /* errors - this code merely */
              /* skips to next image) */
    DO UNTIL(INBUF.RECTYPE='03'|STOP='1'B);
        READ FILE(INPUT) INTO(INBUF);
    END;
    END; /* select group */

    /* End name process */
    END; /* name process */
    /* Else stop (endfile) */
ELSE;

END; /* file process loop */

/*****
/* TERMINATION */
*****/

/* Close file */
CLOSE FILE(INPUT);

/* terminate GDDM */
CALL FSTERM;

/*****
/* GDDM ENTRY POINT DCLS */
*****/

%INCLUDE ADMUPINA;
%INCLUDE ADMUPINF;
%INCLUDE ADMUPINI;
%INCLUDE ADMUPIN5;

END IMAG13;

```

Data types for parameters on GDDM calls

Three types of data are used for the parameters on GDDM calls, although only two (fullword integer and character) are used in GDDM-IVU calls. This table shows their meanings in terms of the four principal high-level languages.

	FORTRAN	COBOL	PL/I	C/370
Fullword integer	INTEGER*4	PICTURE S9(8)	FIXED BINARY(31)	INT
Short floating point	REAL*4	COMPUTATIONAL-1	FLOAT DECIMAL(6)	FLOAT
Character n bytes long, where n is defined in descriptions of calls	String literal or numeric data array initialized with string literals	PICTURE X(n)	CHARACTER(n)	char[n]

Reentrant and system programmer interfaces

The examples in this section use the GDDM nonreentrant interface. Two other interfaces are available: the reentrant and the system programmer. The instructions in the *GDDM Base Application Programming Reference* book on how to access these interfaces apply equally to the GDDM-IVU API.

In summary, the reentrant interface uses the same calls as the nonreentrant, except that an extra parameter comprising a control block in program storage is always passed. The system programmer interface uses a single call, with codes called request-control parameters (RCPs) replacing the individual calls. The RCP codes are given in the call descriptions.

Managing GDDM files

Appendix C, "Listing and managing images and projections in disk files" on page 91 explains to terminal users how they can list, copy, rename, and delete images. You may need to limit their actions by using normal system functions. "Restricting access to disk storage" on page 67 explains how to limit user actions when GDDM-IVU is called from an application program.

Under CICS systems, you may need to write transactions for managing images and projections stored on disk. GDDM images and projections, together with other GDDM objects, are held in a key-sequenced VSAM data set normally called ADMF. The records are normally 400-bytes long, with the key embedded in the first 20 bytes. The format of the records is:

Byte number																			
1 1 1 1 1 1 1 1 1 1 2 2 2 . . .																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Name of image or projection										A D M I M G A D M P R O J					Seqnce. number				

The first eight bytes contain the name of the image or projection, the next eight, the type (normally ADMIMG for images and ADMPROJ for projections), the next four, a binary sequence number, and the rest of the record, the image or projection data.

To delete an image or projection, for example, you can code a CICS file-control statement similar to this one in a program that you execute as a CICS transaction:

```
EXEC CICS DELETE DATASET (ADMF)
      RIDFLD (1st-16-bytes-of-key)
      KEYLENGTH(16)
      GENERIC
```

Restricting access to disk storage

This is how to prevent users from writing to disk:

1. Disable the Output (PF6) key, Projectn (PF5) key, and the Home (PF12) key, using the IUPFK call.
2. Use the first parameter of the IUIVU call to enter GDDM-IVU at the required function, which must not be any of the following: Home, Output Selection, Projection, Save Projection, Save Image, or Export. The Print function also writes output to disk, so you might want to avoid specifying this function on the IUIVU call. It may write temporary print files only, depending on how your system is set up.

Compiling and running GDDM-IVU programs

GDDM programs have no special compilation requirements, unless they are written in PL/I or C/370 and include the IBM-supplied declarations. In that case, the declarations must be made available to the compiler.

At run time, the only special requirement is that the GDDM subroutines must be available to the program. This requirement is satisfied by including one or more interface modules when the program is link-edited.

The precise methods depend on the subsystem being used. Information about compiling and running GDDM programs under TSO, CICS, and CMS is given in the *GDDM Base Application Programming Guide*. The information applies equally to programs using the GDDM-IVU API.

End of General-use programming interface

Appendix A. GDDM-IVU calls

General-use programming interface

This appendix contains the definitive description of the GDDM-IVU call interface. The “principal errors” lists are neither exhaustive nor guaranteed. Although part of the interface, the messages and their texts are subject to change from release to release or as a result of maintenance.

The GDDM-IVU calls that comprise the API (application programming interface) are listed here in alphabetic order of call name. Each call description has the following format:

1. The call name
2. The parameter string (if any)
3. APL code, in decimal
4. RCP code, in hexadecimal and decimal
5. Description of call function
6. Definition of parameters (if any):
 - a. Source of parameter contents (user or GDDM)
 - b. Data type
 - c. Description of parameter function
 - d. Description of contents, if array
 - e. Definition of limits (if any)
7. Principal errors (if any).

IUAIMG

Function: To add image to the name table.

IUAIMG (**imgid, imgnm**)

APL code 1715
GDDM-IVU RCP code X'44050000' (1141178368)

Parameters

imgid (*specified by user*) (*fullword integer*)

The GDDM image identifier.

imgnm (*specified by user*) (*8-byte character string*)

The GDDM-IVU name to be associated with the image.

Description

Adds the specified image identifier and image name to the GDDM-IVU name table. The maximum number of image entries in the name table is controlled by the IUMLO call; see “IUMLO – Set the maximum number of loaded objects” on page 75.

Checks are performed to ensure that the name and identifier are not already in the GDDM-IVU name table, and that the specified image exists.

Image identifiers of less than 1 are not valid.

The image name is checked for adherence to the GDDM-IVU convention for names as follows:

- Folded to uppercase
- Left justified
- Padded with blanks
- Contents validated for the current subsystem conventions.

Principal errors

```
ADM4013 E IMAGE WITH THE SPECIFIED NAME ALREADY EXISTS
ADM4016 E IMAGE LIMIT HAS BEEN REACHED. THE IMAGE CANNOT BE LOADED
ADM4037 E INVALID IMAGE NAME
ADM4038 E INVALID IMAGE IDENTIFIER
ADM4047 E IMAGE WITH THE SPECIFIED IDENTIFIER ALREADY EXISTS
ADM4048 E IMAGE WITH THE SPECIFIED IDENTIFIER DOES NOT EXIST
ADM4077 E IMAGE WITH THE SPECIFIED NAME DOES NOT EXIST
```

IUAPRJ

Function: To add projection to the name table.

IUAPRJ (prjid, prjnm)

APL code 1705
GDDM-IVU RCP code X'44050200' (1141178880)

Parameters

prjid (*specified by user*) (*fullword integer*)

The GDDM projection identifier.

prjnm (*specified by user*) (*8-byte character string*)

The GDDM-IVU name to be associated with the projection.

Description

Adds the specified projection identifier and projection name to the GDDM-IVU name table. The maximum number of projection entries in the name table is controlled by the IUMLO call; see “IUMLO – Set the maximum number of loaded objects” on page 75.

Checks are performed to ensure that the name and identifier are not already in the GDDM-IVU name table, and that the specified projection exists.

Projection identifiers of less than 1 are not valid.

The projection name is checked for adherence to the GDDM-IVU convention for names as follows:

- Folded to uppercase
- Left justified
- Padded with blanks
- Contents validated for the current system conventions.

Principal errors

```
ADM4015 E PROJECTION WITH THE SPECIFIED NAME DOES NOT EXIST
ADM4017 E PROJECTION LIMIT HAS BEEN REACHED. PROJECTION CANNOT BE LOADED
ADM4039 E INVALID PROJECTION NAME
ADM4040 E INVALID PROJECTION IDENTIFIER
ADM4049 E PROJECTION WITH THE SPECIFIED NAME ALREADY EXISTS
ADM4050 E PROJECTION WITH THE SPECIFIED IDENTIFIER ALREADY EXISTS
ADM4051 E PROJECTION WITH THE SPECIFIED IDENTIFIER DOES NOT EXIST
```

IUDIMG

Function: To delete image from the name table.

IUDIMG (imgid, imgnm)

APL code 1716
GDDM-IVU RCP code X'44050100' (1141178624)

Parameters

imgid (*specified by user*) (*fullword integer*)

The identifier of the image whose entry is to be deleted.

imgnm (*specified by user*) (*8-byte character string*)

The name of the image whose entry is to be deleted.

Description

Removes the specified image entry from the GDDM-IVU name table.

The name table is searched for an entry that matches both the identifier and name specified. If found, the entry in the name table is deleted.

Principal errors

ADM4037 E INVALID IMAGE NAME
ADM4038 E INVALID IMAGE IDENTIFIER
ADM4046 W NO MATCH FOUND FOR THE SPECIFIED NAME AND IDENTIFIER

IUDPRJ

Function: To delete projection from the name table.

IUDPRJ (prjid, prjnm)

APL code 1706
GDDM-IVU RCP code X'44050300' (1141179136)

Parameters

prjid (*specified by user*) (*fullword integer*)

The identifier of the projection whose entry is to be deleted.

prjnm (*specified by user*) (*8-byte character string*)

The name of the projection whose entry is to be deleted.

Description

Removes the specified projection entry from the GDDM-IVU name table.

The name table is searched for an entry that matches both the identifier and name specified. If found, the entry in the name table is deleted.

Principal errors

ADM4039 E INVALID PROJECTION NAME
ADM4040 E INVALID PROJECTION IDENTIFIER
ADM4046 W NO MATCH FOUND FOR THE SPECIFIED NAME AND IDENTIFIER

IUINIT

Function: To initialize GDDM-IVU.

IUINIT	(mapgrp-prefix)
APL code	1702
IVU RCP code	X'44000000' (1140850688)

Parameters

mapgrp-prefix (*specified by user*) (4-byte character string)
The map-group prefix to be used in forming the map-group names.

Specifying four blanks as the prefix indicates that the GDDM-IVU default prefix of “ADM5” is to be used.

Description

Initializes or re-initializes GDDM-IVU. The GDDM-IVU storage locations are initialized. All options and parameters are set to their default values.

A four-character map-group name prefix may be supplied. This prefix is later applied to the variable suffixes held within GDDM-IVU in order to derive the full names of the map groups to be used.

Principal errors

ADM4004 E SUITABLE MAP GROUPS COULD NOT BE FOUND

IUIVU

Function: To invoke GDDM-IVU.

IUIVU (control, image-name, projn-name)

APL code 1704
IVU RCP code X'44030000' (1141047296)

Parameters

control (*specified by user*) (*fullword integer*)

Selects how GDDM-IVU is invoked:

- 0 Invoke GDDM-IVU as a stand-alone utility
- 1 Invoke the Home panel
- 2 Invoke the Input Selection panel
- 3 Invoke the Output Selection panel
- 4 Invoke the Scan process
- 5 Invoke the View process
- 6 Invoke the Edit process
- 7 Invoke the Print process
- 8 Invoke the Projection Panel
- 9 Invoke the Create Projection process
- 10 Invoke the Load Projection process
- 11 Invoke the Save Projection process
- 12 Invoke the Load Image process
- 13 Invoke the Save Image process
- 14 Invoke the Import process
- 15 Invoke the Export process
- 16 Invoke the Discard Image process
- 17 Invoke the Discard Projection process

image-name (*specified by user*) (*8-byte character string*)

Set the current image name. This is the name that is copied into the image name field of all panels that contain such a field.

When it is blank, the current image name is set to blank. When the first character is an equals sign (=) and the remaining characters are blanks, the current image name is left unchanged.

projn-name (*specified by user*) (*8-byte character string*)

Set the current projection name. This is the name that is copied into the projection name field of all panels that contain such a field.

When it is blank, the current projection name is set to blank. When the first character is an equals sign (=) and the remaining characters are blanks, the current projection name is left unchanged.

Description

Invokes GDDM-IVU.

The IUIVU call can be used to invoke GDDM-IVU as a stand-alone utility or it can select a specific GDDM-IVU process (such as Scan or View) and invoke it as an independent function. It also allows the current image and projection names to be set by the application program.

When GDDM-IVU is invoked as a stand-alone utility, its storage locations are initialized to their default values. GDDM-IVU calls made before the IUIVU call have no effect on the operation of the utility. The **image-name** and **projn-name** parameters can be used to supply default image and projection names, if required.

When a specific process is invoked, GDDM-IVU's storage locations remain unchanged. GDDM Base image calls and GDDM-IVU calls can be used to preload images and projections and the **image-name** and **projn-name** parameters can be used to prime the image and projection name fields of the selected menu panel.

In the case of the Create Projection process, selected by a value of **9** in the **control** parameter, there is no image-name field on the menu. The only way of specifying an image is by means of the **image-name** parameter. For this process, therefore, a nonblank image name is normally a requirement.

The default (or previously specified) map group prefix is applied to the variable suffixes held within GDDM-IVU in order to derive the full names of the map groups to be used. The existence of these map groups is checked before processing is allowed to continue. If map groups for the correct national language cannot be found, the default language map groups are substituted.

Principal errors

```
ADM4000 W CHANGING TERMINAL SETUP CAN IMPROVE GDDM-IVU PERFORMANCE
ADM4001 E TERMINAL IS NOT SUITABLE FOR RUNNING GDDM-IVU. REASON CODE n
ADM4004 E SUITABLE MAP GROUPS COULD NOT BE FOUND
ADM4031 W THE DEFAULT LANGUAGE MAP GROUPS HAVE BEEN SELECTED
ADM4036 E INVALID CONTROL VALUE
ADM4037 E INVALID IMAGE NAME
ADM4039 E INVALID PROJECTION NAME
```

IUMLO

Function: To set the maximum number of loaded objects.

IUMLO (type, count)

APL code	1717
IVU RCP code	X'44080000' (1141374976)

Parameters

type (*specified by user*) (*fullword integer*)

Indicates the type of object for which the limit is to be set.

- 1** Images
- 2** Projections

count (*specified by user*) (*fullword integer*)

The maximum number of objects (specified by **type**) that can be loaded in GDDM-IVU.

The smallest value that can be specified for **count** is the larger of 1 and the number of objects already loaded in the specified name table.

The largest value that can be specified for **count** is 32.

Description

Sets the maximum number of images or projections that can have entries in the GDDM-IVU name table.

Principal errors

ADM4043 E INVALID COUNT

ADM4044 E INVALID TYPE

IUPFK

Function: To set PF key definitions.

IUPFK (count, key-array, func-array)

APL code 1713

IVU RCP code X'44040000' (1141112832)

Parameters

count (*specified by user*) (*fullword integer*)

The number of elements in the **key-array** and **func-array** parameters.

key-array (*specified by user*) (*an array of fullword integers*)

An array of PF key numbers. Key numbers 1 through 32 are valid.

func-array (*specified by user*) (*an array of fullword integers*)

An array of PF key definition values that correspond to the key numbers specified in **key-array**. Valid values are:

-1 No operation (the PF key does nothing)

0 Set to the default definitions:

PF 1 and 13	Help
PF 2 and 14	No Menu / Menu
PF 3 and 15	Return / Exit
PF 4 and 16	Input
PF 5 and 17	Projectn / Undo
PF 6 and 18	Output
PF 7 and 19	Backward
PF 8 and 20	Forward
PF 9 and 21	Clear
PF 10 and 22	No operation
PF 11 and 23	No operation
PF 12 and 24	Home
PF 25 through PF 32	No operation

1 Help

Displays the help panel associated with the current menu panel.

2 No Menu / Menu

Removes the current menu panel so that more of the image or help information can be seen. Restores the current menu panel when pressed again.

- 3** Return / Exit
When pressed from the first panel invoked, an exit from GDDM-IVU is performed. Otherwise the previous menu panel is displayed.
- 4** Input
Displays the Input Selection menu panel.
- 5** Projectn / Undo
Projectn displays the Projection menu panel.
Undo removes the effect of the previous image editing operation.
- 6** Output
Displays the Output Selection menu panel.
- 7** Backward
Scrolls towards the top of the image or help information.
- 8** Forward
Scrolls towards the bottom of the image or help information.
- 9** Clear
Erases the image from the screen.
- 10** No operation
- 11** No operation
- 12** Home
Displays the Home menu panel.

Description

Assigns functions to the keyboard PF (program function) keys. The only requirement is that one PF key in the range 1 through 12 must be set to the Return function.

Principal errors

```
ADM4041 E INVALID PF KEY NUMBER
ADM4042 E INVALID PF KEY DEFINITION VALUE
ADM4043 E INVALID COUNT
ADM4074 E INVALID ATTEMPT TO SET THE PF KEYS
```

IUQIID

Function: To query image identifier.

IUQIID (imgnm, imgid)

APL code 1707
GDDM-IVU RCP code X'44060000' (1141243904)

Parameters

imgnm (specified by user) (8-byte character string)

The name of the image to search for.

imgid (returned by GDDM) (fullword integer)

The returned image identifier.

Description

Returns the GDDM image identifier corresponding to the GDDM-IVU image name specified.

The GDDM-IVU name table is searched for the image name specified.

If it is found, the associated image identifier is returned.

If it is not found, the returned image identifier is set to zero.

Principal errors

ADM4037 E INVALID IMAGE NAME
ADM4077 E IMAGE WITH THE SPECIFIED NAME DOES NOT EXIST

IUQINM

Function: To query image name.

IUQINM (imgid, imgnm)

APL code 1708
GDDM-IVU RCP code X'44060100' (1141244160)

Parameters

imgid (specified by user) (fullword integer)

The identifier of the image to search for.

imgnm (returned by GDDM) (8-byte character string)

The returned image name.

Description

Returns the GDDM-IVU image name corresponding to the GDDM image identifier specified.

The GDDM-IVU name table is searched for the image identifier specified.

If it is found, the associated image name is returned.

If it is not found, the returned image name is set to blanks.

Principal errors

ADM4038 E INVALID IMAGE IDENTIFIER

ADM4048 E IMAGE WITH THE SPECIFIED IDENTIFIER DOES NOT EXIST

IUQLO

Function: To query the currently loaded object identifiers and names.

IUQLO (type, count, id-array, name-array, number)

APL code 1712

GDDM-IVU RCP code X'44070100' (1141309696)

Parameters

type (specified by user) (fullword integer)

Indicates the type of object to return information about:

- 1 Return information about loaded images
- 2 Return information about loaded projections

count (specified by user) (fullword integer)

The number of objects to return information about. The IUQNLO call (see “IUQNLO – Query the number of loaded objects” on page 80) can be used to find how many objects of the type required exist in the GDDM-IVU name table.

id-array (returned by GDDM) (an array of fullword integers)

The GDDM identifiers of the first **count** objects, of specified **type**, found in the GDDM-IVU name table. The array must have at least **count** elements.

name-array (returned by GDDM) (array of 8-byte character tokens)

The GDDM-IVU names of the first **count** objects, of specified **type**, found in the GDDM-IVU name table. Each element contains the name associated with the GDDM identifier in the corresponding element in **id-array**. The array must have at least **count** elements.

number (returned by GDDM) (fullword integer)

The number of identifiers returned in **id-array** and **name-array**. This may be smaller than **count**. When **number** is smaller than **count**, the unused elements of **id-array** and **name-array** are left unchanged.

Description

Returns the GDDM identifiers and the corresponding GDDM-IVU names of images or projections, by searching the GDDM-IVU name table.

Principal errors

ADM4043 E INVALID COUNT
ADM4044 E INVALID TYPE

IUQMLO

Function: To query the maximum number of loaded objects.

IUQMLO (type, count)

APL code 1718
IVU RCP code X'44080100' (1141375232)

Parameters

type (*specified by user*) (*fullword integer*)

Indicates the type of object for which the current maximum is required.

- 1 Images
- 2 Projections

count (*returned by GDDM*) (*fullword integer*)

The maximum number of objects (specified by **type**) that can be loaded in GDDM-IVU.

Description

Returns the maximum number of images or projections that can have entries in the GDDM-IVU name table.

Principal errors

ADM4044 E INVALID TYPE

IUQNLO

Function: To query the number of loaded objects.

IUQNLO (type, count)

APL code 1711
GDDM-IVU RCP code X'44070000' (1141309440)

Parameters

type (*specified by user*) (*fullword integer*)

Indicates the type of object to return information about:

- 1 Return the number of loaded images
- 2 Return the number of loaded projections

count (*returned by GDDM*) (*fullword integer*)

The number of objects (specified by **type**) that are currently loaded in GDDM-IVU.

Description

Returns the number of images or projections that have an entry in the GDDM-IVU name table.

Principal errors

ADM4044 E INVALID TYPE

IUQPFK

Function: To query PF key definitions.

IUQPFK (count, key-array, func-array)

APL code 1714

GDDM-IVU RCP code X'44040100' (1141113088)

Parameters

count (*specified by user*) (*fullword integer*)

The number of elements in the **key-array** and **func-array** parameters.

key-array (*specified by user*) (*an array of fullword integers*)

An array of PF key numbers. Key numbers 1 through 32 are valid.

func-array (*returned by GDDM*) (*an array of fullword integers*)

An array of current PF key definition values that correspond to the key numbers specified in **key-array**. For a list of possible values, see “IUQPFK – Set PF key definitions” on page 76.

Description

Returns the current functions assigned to the keyboard PF (program function) keys.

Principal errors

ADM4041 E INVALID PF KEY NUMBER

ADM4043 E INVALID COUNT

IUQPID

Function: To query projection identifier.

IUQPID (prjnm, prjid)

APL code 1709
GDDM-IVU RCP code X'44060200' (1141244416)

Parameters

prjnm (specified by user) (8-byte character string)

The name of the projection to search for.

prjid (returned by GDDM) (fullword integer)

The returned projection identifier.

Description

Returns the GDDM projection identifier corresponding to the GDDM-IVU projection name specified.

The GDDM-IVU name table is searched for the projection name specified.

If it is found, the associated projection identifier is returned.

If it is not found, the returned projection identifier is set to zero.

Principal errors

ADM4015 E PROJECTION WITH THE SPECIFIED NAME DOES NOT EXIST
ADM4039 E INVALID PROJECTION NAME

IUQPNM

Function: To query projection name.

IUQPNM (prjid, prjnm)

APL code 1710
GDDM-IVU RCP code X'44060300' (1141244672)

Parameters

prjid (specified by user) (fullword integer)

The identifier of the projection to search for.

prjnm (returned by GDDM) (8-byte character string)

The returned projection name.

Description

Returns the GDDM-IVU projection name corresponding to the GDDM projection identifier specified.

The GDDM-IVU name table is searched for the projection identifier specified.

If it is found, the associated projection name is returned.

If it is not found, the returned projection name is set to blanks.

Principal errors

ADM4040 E INVALID PROJECTION IDENTIFIER
ADM4051 E PROJECTION WITH THE SPECIFIED IDENTIFIER DOES NOT EXIST

IUTERM

Function: To terminate GDDM-IVU.

IUTERM

APL Code	1703
GDDM-IVU RCP code	X'44010000' (1140916224)

Parameters

None

Description

Terminates GDDM-IVU.

All images and projections that have an entry in the GDDM-IVU name table are deleted.

All storage acquired by previous GDDM-IVU calls is released.

The caller's partition set, partition, and page are restored. If GDDM-IVU opened the primary device, it is closed.

Principal errors

None

_____ End of General-use programming interface _____

Appendix B. Sample programs

General-use programming interface

This appendix describes the sample programs that are supplied with GDDM-IVU.

As these programs are samples only, they may require changes before they can be used on your system. Although they have been subject to some testing by IBM, they are not guaranteed to operate on all configurations and levels of system, subsystem, and related software products. Nor do the programs contain extensive error checking and recovery logic. If you intend using these programs on your system, you must consider whether extensions and other changes may be needed.

These programs demonstrate the use of the programming interface provided by GDDM-IVU. You can use them as the basis of programs providing an interface from GDDM-IVU to Distributed Office Support System/370 (DISOSS, program numbers 5665-290 (MVS) and 5666-270 (VSE)) via Personal Services/370 (program numbers 5665-330 (MVS) and 5666-318 (VSE)). As well as providing an interface to DISOSS, the programs allow users of the Image View Facility program offering (IVF, program number 5785-ECX) to migrate to GDDM-IVU.

The programs are written in either PL/I or Assembler language. Both source files and object (compiler and assembler output) files are supplied with GDDM-IVU. Listings of the programs are not included in this manual. You can obtain listings by printing the source files that were placed on your system as part of the GDDM-IVU installation process.

What the programs do

These programs allow terminal users to:

- Use Personal Services/370 to view an image-type DISOSS document. The document can be either retrieved by the user from the DISOSS document library or received from another user.
- File an image in the DISOSS document library, or distribute an image to a Personal Services/370 user. The image can be one which has been scanned or created by GDDM-IVU (or another GDDM application program) or one which is on view from Personal Services/370.

These functions are described in more detail in the next two sections.

View an image using Personal Services/370

With Release 1 of Personal Services/370, a user can simply request the **View** option for a document with type IMAGE. In Release 2, users must ask for option **85** on the in-basket panels IB21 or IB22, although that option may not actually be described on those panels. In both cases, users must have set their Personal Services/370 User Defaults to allow the handling of image documents.

When a user chooses an image document for viewing, Personal Services/370 calls the program ADMU5DG, which calls the GDDM-IVU view function. Although the main use of this function is to view an existing image, users can access other features of GDDM-IVU, including the output functions, and even scanning and

loading. GDDM-IVU is called in such a way that no more than one image may be held by GDDM-IVU at one time: the discard function must be used to remove one image before loading another.

When users leave GDDM-IVU, the option of filing or distributing a retained image is offered, as described in “Put image from GDDM-IVU into DISOSS or Personal Services/370.” Whatever they choose, the original image document is not changed or deleted.

Put image from GDDM-IVU into DISOSS or Personal Services/370

This function is provided by a CICS transaction named **IVDP**.

This transaction invokes the program ADMU5DP, which calls GDDM-IVU. Users can call upon any of the functions of GDDM-IVU, including scanning and loading. GDDM-IVU is called in such a way that no more than one image may be held by GDDM-IVU at one time: the discard function must be used to change images.

If an image is held by GDDM-IVU (that is, scanned or loaded and not discarded) when a user terminates GDDM-IVU, the program offers (by means of a panel to fill in) the options of filing the image in the DISOSS document library or of distributing it to a Personal Services/370 user. Both these options may be chosen as many times as required.

How to install the programs

When GDDM-IVU itself has been installed, the source of the programs is available in the SADMSAM library (for MVS) or the PRD2.PROD library (for VSE), while the object code for the modules is also available in the SADMMOD library (for MVS) or the PRD2.PROD library (for VSE).

You must link-edit again the programs using the object code to resolve the various CICS and DISOSS references. You may put the resulting load modules in any library you choose. With Personal Services/370 Release 2, you must also link-edit a user exit into Personal Services/370 itself.

You must provide information to CICS, DISOSS, and Personal Services/370 about the programs:

- The transient data queue must be defined in the CICS destination control table (DCT).
- The file service requests must be defined in the CICS file control table (FCT).
- The transaction names must be defined in the CICS program control table (PCT).
- The program names must be defined in the CICS processing program table (PPT).
- Userids must be defined in the DISOSS host user profile (HUP).
- With Personal Services/370 Release 2, the user exit must be defined in the dialog processor branch and leaf table (DPBLT).

Since images involve large amounts of data, it may be desirable to increase the size of certain DISOSS and Personal Services/370 data sets, such as the API request queue, the delivery queue, and the document pool.

Sample statements for the link editing and for the definitions in the various tables are supplied in the source module ADMU5DZ, which also includes fuller installation instructions.

How the programs work

There are four programs in all, comprising 11 separate source modules, written in either PL/I or Assembler language. The programs work only in a CICS environment, and they use the reentrant programming interface to GDDM-IVU and GDDM/MVS or GDDM/VSE. An overview of each program follows, together with a summary list of the modules. For detailed information, you should refer to the module commentaries.

ADMU5DP

This puts images from GDDM-IVU into DISOSS. It is invoked by the CICS transaction IVDP.

After calling GDDM-IVU, if there is an image held in GDDM-IVU storage, a panel is presented from which filing or distribution requests for the image can be made. Each request is recorded by writing a transfer record to the CICS intrapartition transient data queue IVDX, with the associated image data on a CICS temporary storage queue; these queues are subsequently processed by ADMU5DD.

ADMU5DD

This transfers images to DISOSS for filing or distribution. It is invoked automatically by CICS (as transaction IVDD) when a transfer request is put on the queue IVDX, and is reinvoked by DISOSS between each of its four phases of operation.

The phases are:

1. Sign-on to DISOSS with userid IVDDIS. CICS temporary storage queue IVDQDDON records the sign-on status.
2. Read the transfer request from the transient data queue IVDX, read the associated image data, and send it to DISOSS.
3. If the request came from a Personal Services/370 view operation, send a message to DISOSS for the originating user.
4. Return to phase 2 for the next request on the queue. If there are no more requests, sign-off from DISOSS and terminate.

ADMU5DD works asynchronously from the users' requests and works in its four phases because of the essentially asynchronous nature of the application programming interface (API) to DISOSS; however, it does not attempt to overlap separate image transfers. Linkage to DISOSS is provided by routines based on those described in *DISOSS Application Interface: Programming Guidelines*, which also, together with *DISOSS Application Programming* and *DISOSS Interchange Architecture Reference*, provides full information on the DISOSS API.

ADMU5DG

This provides the function for viewing images from Personal Services/370. It is invoked directly by Release 1 of Personal Services/370, using the program name ECXVMEN; a communication area describes the image document. With Release 2, when option 85 is requested by the user, Personal Services/370 invokes user exit 6 (which is ADMU5DX) by the name DMDFOX06; this emulates the processing which is internal in Release 1, constructing a similar communication area describing the image document and calling ADMU5DG.

After transferring the image document from the Personal Services/370 document pool data set into GDDM-IVU storage, ADMU5DG invokes GDDM-IVU and proceeds exactly like ADMU5DP.

ADMU5DC

This cleans up requests to DISOSS, which may be necessary after unexpected errors occur. It is invoked by the CICS transaction IVDC.

List of modules

ADMU5DA	Assembler subroutine called by ADMU5DD to analyze output from DISOSS.
ADMU5DC	PL/I main routine of program ADMU5DC, which cleans up requests to DISOSS.
ADMU5DD	PL/I main routine of program ADMU5DD, which handles the transfer of images to DISOSS.
ADMU5DE	PL/I declarations of the parameter and communication areas for the DISOSS linkage subroutines in ADMU5DA and ADMU5DL.
ADMU5DG	PL/I main routine of program ADMU5DG, which provides the function for viewing images from Personal Services/370.
ADMU5DI	Assembler equivalent of ADMU5DE.
ADMU5DL	Assembler subroutines called by ADMU5DC and ADMU5DD to link to the DISOSS API.
ADMU5DP	PL/I main routine of program ADMU5DP, which puts images from GDDM-IVU into DISOSS.
ADMU5DT	PL/I declarations of the transfer request record, built by ADMU5DU for use by ADMU5DD.
ADMU5DU	PL/I subroutine called by ADMU5DG and ADMU5DP, which handles the user request panel.
ADMU5DX	Assembler routine comprising program ADMU5DX, the user exit for Personal Services/370 Release 2, into which it is link-edited.
ADMU5DZ	Sample installation information.

Production environment

Apart from being samples only, there are several reasons why these programs might be unsuitable for a production environment.

Names: The modules and the programs all have names beginning ADMU5D. The CICS names and DISOSS userids used all begin with IVD. These names may be unsuitable for your system. The source module ADMU5DZ tabulates where each name is used.

Message and request panel wording: The message identifiers all begin with ADMU5D. The wording of the messages and of the request panel may not be suitable, or may be required in a different national language. The programs use only U.S. English, and are not supplied in any other language.

Error handling and recovery: The error handling provided by the programs is basic, with limited diagnosis, while recovery from errors is minimal. Generally, the programs simply issue a message and stop. This may need to be more extensive and provide for easier diagnosis after errors occur.

Journaling: Messages and error reports are written to the PL/I SYSPRINT file provided by CICS. Another destination might be more suitable.

Personal Services/370 exit: For Personal Services/370 Release 2, the number of the exit, the number of the option, or the choice of panels on which it is allowed might be unsuitable.

Other improvements: Some examples of improvements you may want to consider are: handling more than one image at a time (ADMU5DU), overlapping transfer requests (ADMU5DD), handling other image formats (ADMU5DG), better data checking (ADMU5DU), different request panel protocols (ADMU5DU).

Other things to fit your system: There may be other aspects of the conventions applying to your system or of the levels of the related programs you are using which would require different programming.

The overviews in “How the programs work” on page 87 together with the detailed module commentaries will help you understand how to prepare programs that meet your own particular requirements.

How to make changes

If you change any modules in the sample programs, they must undergo the usual processes for CICS transactions before they can be executed. You must process them with the CICS command language translator. Then you must compile them (using PL/I) or assemble them as you would any other GDDM application program which uses the reentrant interface to GDDM in a CICS environment. Finally, you must link-edit again the programs as described in “How to install the programs” on page 86. Information about these processes is given in the *GDDM Base Application Programming Guide*.

What hardware and software you need

You require no special hardware to run the programs, beyond that required by any of the following software products.

You require GDDM-IVU Version 1 and GDDM/MVS or GDDM/VSE Version 3, together with whatever levels of CICS and MVS or VSE and PL/I Libraries they, in their turn, require. You require Personal Services/370 Release 1 or Release 2 together with whatever level of DISOSS it requires. If you want to change the programs, you will need a PL/I compiler.

_____ End of General-use programming interface _____

Appendix C. Listing and managing images and projections in disk files

If the GDDM-PGF ICU (Interactive Chart Utility) is available to you, you can use it to list GDDM images and projections stored in disk files.

After invoking the ICU, go first to the menu-control panel and specify advanced mode for the directory panel. Then press PF2 to display the directory panel. Finally, type L (list) under Commands, type IMAGE under Type to list all available images, or IMPROJ to list all available projections, and press Enter.

Other commands are available on the directory panel for copying and deleting images and projections.

If the ICU is not available, there are other methods, which depend on the subsystem.

Under CMS

1. If you are in a GDDM-IVU session, press PA2. This puts you into CMS subset mode (unless your computer support personnel have assigned some other action to PA2).
2. GDDM images on disk files normally have a filetype of ADMIMG, and projections a filetype of ADMPROJ. There are several ways of listing them. This is a good way for images:
`FILELIST * ADMIMG *`
 or this for projections:
`FILELIST * ADMPROJ *`
 Your computer support personnel may have assigned different filetypes.
3. If you were in a GDDM-IVU session, enter the command
`RETURN`
 to return to it.

Each file has three parts to its name. The name that you need to specify on GDDM-IVU panels is the filename, which is the first part.

The CMS COPY, RENAME, and ERASE commands can also be used on images and projections.

Under TSO

If you have ISPF on your system, you can use it to list images and projections. Ask your computer-support personnel about the names of the data sets in which they are stored, or use the LISTALC command shown below.

If you do not have ISPF, you can use TSO commands. First you need to find out the names of the data sets holding the images and projections. Enter this command:

```
LISTALC STATUS
```

Look through the resulting screen output for a data set with the ddname ADMIMG (for images) or ADMPROJ (for projections)⁵.

Then, to list the images or projections, enter the following command:

```
LISTDS 'data-set-name' MEMBERS
```

where data-set-name is the fully qualified name you found in the LISTALC output.

The names that you specify on GDDM panels are the member names listed by this command.

Alternatively, use the ISPF LISTDS Utility panel, ISPF Option 3.4.

The TSO COPY, RENAME, and DELETE commands can also be used on images and projections.

Under CICS

Special programming is needed to manage disk files under CICS. Information for programmers is given on page 66. You should ask your computer-support personnel about the facilities available and procedures to follow.

⁵ ADMIMG and ADMPROJ are the recommended ddnames for the data sets that hold GDDM images and projections. Your computer support personnel may have chosen other names. Ask them what these are if you cannot find the recommended ddnames.

Appendix D. Editing GDDM-IVU panels

If you have installed both GDDM Interactive Map Definition (GDDM-IMD) and GDDM-IVU, you might want to use GDDM-IMD to edit the panels supplied with GDDM-IVU. You can change any of the existing fields, or add further information for your users. For information about GDDM-IMD, see the *GDDM Interactive Map Definition* book.

Tailoring GDDM-IVU panels

The standard name for the MSL (map specification library) containing the source of the panels is ADM5IUx, where x is one of the following national-language identifiers:

B	Brazilian (Portuguese)
D	Danish
F	French
Q	French (Canadian)
G	German
H	Hangeul (Korean)
I	Italian
K	Kanji (Japanese)
N	Norwegian
C	Simplified Chinese (PRC)
S	Spanish
V	Swedish
T	Traditional Chinese (Taiwan)
A	U.S. English

The MSL contains five mapgroups called ADM51x through ADM55x. These hold all the menu and help panels.

Most of the maps have names of the form Mgxxynn, where:

g is 1 for menu maps and 2 for help maps
 xx indicates the process that the map is for (for example, VW for View)
 y is D for menu maps and H for help maps
 nn is a sequence number starting at 01

Maps with names not of this form are concerned with the PF key lines at the bottom of the screen.

GDDM-IVU accepts changed maps, subject to the following restrictions:

- Mapgroup characteristics must not be changed.
- Mapgroups must not be deleted.
- Map characteristics must not be changed.
- Maps must not be moved between mapgroups.
- Maps must not be deleted.
- Maps must not be renamed.
- The ADSs (application data structures) required by the maps must not be changed.
- Field adjuncts must not be changed.
- AID tables must not be changed.

The last three items apply to menu panels only.

If you intend to tailor any maps, take a copy of the MSL and edit the copy. Do not edit the original MSL. This is not only a precaution against mistakes. It could also be required if you report a problem in GDDM-IVU to IBM, because if the problem occurred while using your own versions of maps, you may be asked if it recurs using the standard maps. It is also advisable, for reasons explained below, to generate standard ADSs using the mapgroup generator function, without generating the mapgroup.

After editing the maps, you must create new generated mapgroups using the mapgroup generator function in the normal way. This process can also create new application data structures. It is not essential to do this, but it provides a way of checking that you have not inadvertently changed the ADSs required by the maps. If you specify a nondefault ADS filetype (under CMS), library ddname (under TSO), or transient data queue (under CICS), you can check that the generated ADS files are the same as those generated from the standard MSL.

In the MVS environment

Under CICS, you need to use the ADMX staging file (described in the *GDDM/MVS Program Directory*. using the supplied job stream ADMUJCIP to move the map source into ADMX. You must edit ADMUJCIP to match the national language panels you want to load, and run it once for each language you need. You then invoke GDDM-IMD to import the files into a map specification library (MSL).

Under TSO, you use the ADMFPSU or ADMFCSU sample program (supplied with GDDM-IMD — see the *GDDM Interactive Map Definition* book) to move the maps from ADMX into a sequential file. You can then import the files into an MSL using GDDM-IMD.

Edit the maps and generate your required mapgroups.

In the VM environment

You use GDDM-IMD to import the panels from the ADM5IUx ADMIFMT files (x is the national language character). Then edit the maps, and generate your required mapgroups.

In the VSE environment

Under CICS, you need to use the ADMX staging file (described in the *GDDM/VSE Program Directory*. using the supplied job stream ADMUJDIP to move the map source into ADMX. You must edit ADMUJCIP to match the national language panels you want to load, and run it once for each language you need. You then invoke GDDM-IMD to import the files into a map specification library (MSL).

Edit the maps and generate your required mapgroups.

Glossary

This glossary defines technical terms used in GDDM documentation. If you do not find the term you are looking for, refer to the index of the appropriate GDDM manual or view the *IBM Dictionary of Computing*, located on the Internet at:

<http://www.networking.ibm.com/nsg/nsgmain.htm>

A

AAB. Application anchor block.

ACB. Application control block.

active operator window. In GDDM, the operator window with the highest priority in the viewing order.

active partition. The partition containing the cursor. Contrast with *current partition*.

advanced function printing. The ability of licensed programs to use the all-points-addressable concept to print text and illustrations.

adjunct. In mapped alphanumerics, one of a set of optional subfields in an application data structure that specifies some attribute of a data field; for example, that it is highlighted. An adjunct enables the attribute to be varied at run time.

ADMGDF. See *graphics data format (GDF)*.

ADS. Application data structure.

AFPDS. Advanced-function presentation data stream.

AIC. Application interface component.

alphanumeric character attributes. In GDDM, the highlighting, color, and symbol set to be used for individual characters.

alphanumeric cursor. A physical indicator on a display. It can be moved from one hardware cell to another.

alphanumeric field. A field (area of a screen or printer page) that can contain alphabetic, numeric, or special characters. In GDDM, contrast with *graphics field*.

alphanumeric field attributes. In GDDM, the intensity, highlighting, color, and symbol set to be used for field type, field end, output conversion, input conversion, translate table assignment, transparency, field outlining, and mixed-string fields.

alphanumerics. Pertaining to alphanumeric fields. In GDDM there are three types of alphanumerics:

- Procedural alphanumerics
- Mapped alphanumerics
- High performance alphanumerics (HPA)

alternate device. In GDDM, a device to which copies of the primary device's output are sent. Usually the alternate device is a printer or plotter. See also *primary device*.

annotation. An added descriptive comment or explanatory note.

APA. All points addressable.

aperture. See *pick aperture*.

API. Application programming interface.

APL. One of the programming languages supported by GDDM.

application data structure (ADS). A structure created by GDDM-IMD that contains an entry for each variable field within a *map*. The data to be displayed in a mapped field is placed into the application data structure by the user's program.

application image. In GDDM, an image contained in GDDM main storage, and independent of any device or GDDM page. Contrast with *device image*.

application programming interface (API). The formally defined interface used by an application programmer to pass commands to, and get responses from, an IBM system control program or licensed program.

area. In GDDM, a shaded shape, such as a solid rectangle. It is created by opening the area, defining its outline, and closing the area.

aspect ratio. The width-to-height ratio of an area, symbol, or shape.

attention identifier. A number indicating which button the operator pressed to satisfy a read operation. For example, 0 (returned from GDDM to the application program) means that the operator pressed the Enter key.

attribute byte. The screen position that precedes an alphanumeric field on a 3270-family device and holds the attribute information. See also *trailing attribute byte*.

attributes. Characteristics or properties that can be controlled, usually to obtain a required appearance; for example, the color of a line. See also *alphanumeric character attributes*, *alphanumeric field attributes*, and *graphics attributes*.

axis. In a chart, a line that is drawn to indicate units of measurement against which items in the chart can be viewed.

A3. A paper size, more common in Europe than in the U.S. It measures 297mm by 420mm, and is twice the size of A4. See also A4.

A4. A paper size, more common in Europe than in the U.S. It measures 210mm by 297mm, and is half the size of A3. Compare with *quarto*. See also A3.

B

background color. Black on a display, white on a printer. The initial color of the display medium. Contrast with *neutral color*.

bar code. A code representing characters by sets of vertical parallel bars of varying thickness and separation that are read optically by transverse scanning.

BASIC. One of the programming languages supported by GDDM.

BDAM. Basic Direct Access Method.

bi-level image. An image in which each pixel is either black or white (value 0 or 1). Contrast with *gray-scale image* and *halftone image*.

BMS. Basic Mapping Support (CICS).

BPAM. Basic Partitioned Access Method.

business graphics. The methods and techniques for presenting commercial and administrative information in chart form; for example, the creation and display of a sales bar chart. Contrast with *general graphics*.

C

CALS. Continuous Acquisition and Life-Cycle Support.

CDPDS. Composite Document Presentation Data Stream.

CDPF. Composed Document Print Facility.

CDPU. Composite Document Print Utility.

CECP. Country-extended code page.

cell. See *character cell*.

CGM. Computer Graphics Metafile. A file that contains information about the content of a picture, and conforms to the International Standard, ISO 8632, or is of a similar format.

channel-attached. Pertaining to devices that are attached directly to a computer by means of data (I/O) channels. Synonymous with *local*. Contrast with *link-attached*.

character. A letter, digit, or other symbol.

character attributes. See *alphanumeric character attributes*. See also *graphics text attributes*.

character box. In GDDM, the rectangle or (for sheared characters) the parallelogram boundaries that govern the size, orientation, spacing, and italicizing of individual symbols or characters to be shown on a display screen or printer page.

The box width, height, and, if required, shear are specified in world coordinates and can be program-controlled. See also *character mode*. Contrast with *character cell*.

character cell. The physical, rectangular space in which any single character or symbol is displayed on a screen or printer device. The size and position of a character cell are fixed. Size is usually specified in pixels on a given device; for example, 9 by 12 on an IBM 3279 Model 3 display. Position is addressed by row and column coordinates. Synonymous with *hardware cell* and *symbol cell*. Contrast with *character box*.

character code. The means of addressing a symbol in a symbol set, sometimes called *code point*.

The particular form and range of codes depends on the GDDM context. For example:

- For the Image Symbol Editor, a hexadecimal constant in the range X'41' through X'FE', or its EBCDIC character equivalent
- For the Vector Symbol Editor, a hexadecimal constant in the range X'00' through X'FF', or its EBCDIC character equivalent
- For the GDDM API, a decimal constant in the range 0 through 239, or subsets of this range (for example, a marker symbol code range of 1 through 8)

character grid. A notional grid that covers the *graphics field*. The size of the grid determines the basic size of the characters in all text constructed by presentation graphics routines. It is the fundamental measurement in chart layout, governing the spacing of mode-2 characters and the size of mode-3 characters. It also governs the size of the chart margins and thus the plotting area.

character matrix. Synonym for *dot matrix*.

character mode. In GDDM, the type of characters to be used. There are three modes:

- Mode-1 characters are loadable into PS and are of device-dependent fixed size, spacing, and orientation, as are hardware characters.
- Mode-2 characters are image (ISS) characters. Size and orientation are fixed. Spacing is variable by program.
- Mode-3 characters are vector (VSS) characters. Box size, position, spacing, orientation, and shear of individual characters are variable by program.

chart. In GDDM, usually means business chart; for example, a *bar chart*.

choice device. A logical input device that enables the application program to identify keys pressed by the terminal operator.

CICS. Customer Information Control System. A subsystem of MVS or VSE under which GDDM can be used.

clipping. In computer graphics, removing parts of a display image that lie outside a viewport. Synonymous with *scissoring*.

CMS. Conversational Monitor System. A time-sharing subsystem that runs under VM/SP.

COBOL. One of the programming languages supported by GDDM.

code page. Defines the relationship between a set of code points and graphic characters. This relationship covers both the standard alphanumeric characters and the national language variations. GDDM supports a set of code pages used with typographic fonts for the IBM 4250 page printer.

code point. Synonym for *character code*.

Composite Document Presentation Data Stream (CDPDS). A data stream containing graphics, image, and text that is the input to the GDDM Composite Document Print Utility (CDPU).

Composed Document Print Facility (CDPF). An IBM licensed program for processing documents destined for the IBM 4250 page printer.

composed-page image file. An intermediate form, residing on disk, of a picture destined for a page printer.

composed-page printer. See *page printer*.

composed-page printer format. A general term describing the format of print data destined for output by using either *CDPF* or *PSF*.

composite document. A document that contains both formatted text, such as that produced by the DCF program, and graphic or image data, such as that produced by GDDM. It is a combination of text and pictures on a page or set of pages. The pictures can be computer graphics or images created by scanning paper originals.

Composite Document Print Utility (CDPU). A utility that can print or display composite documents

compressed data stream. A data stream that has been made more compact by use of a data-compression algorithm.

constant data. In GDDM, data that is defined in a map and need not be known to the application program.

correlation. The translation (by GDDM) of a screen position into a part of the user's picture. This follows a *pick* operation.

country-extended code page (CECP). An extension of a normal EBCDIC code page that includes definitions of all code points in the range X'41' through X'FE'. Each code page contains the same 190 characters, but the mapping between code points and graphics characters depends on the country for which the code page is defined. This is a method of marking a GDDM object so that the environment in which it was created can be identified. It enables automatic translation to a different environment.

CSD. (1) Under MVS or VSE, CICS system definition. (2) In personal computer systems, Corrective Service Diskette; the means by which service is applied to the personal computer system.

current partition. The partition selected for processing by the application program. Contrast with *active partition*.

current position. In GDDM, the end of the previously drawn primitive. Unless a "move" is performed, this position is also the start of the next primitive.

cursor. A physical indicator that can be moved around a display screen. See *alphanumeric cursor* and *graphics cursor*.

CUT. Control unit terminal.

D

DASD. Direct access storage device.

data stream compatibility (DSC). In IBM 8100 systems, the facility that provides access to System/370 applications that communicate with IBM 3270 Information Display System terminals.

data stream compression. The shortening of an I/O data stream for the purpose of more efficient transmission between link-attached units.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DBCS. Double-byte character set.

DCF. Document Composition Facility.

DCSS. Discontiguous saved segment (VM/SP).

DCT. Destination control table (CICS).

default value. The value of an attribute chosen by GDDM when no value is explicitly specified by the user. For example, the default line type is a solid line. The default value is sometimes device-dependent. See also *drawing default* and *standard default*.

denibblized data. The decoded data stream used between the GDDM DOS Support feature in the host and GDDM-PCLK on the workstation.

designator character. The first byte of a light-pen-detectable field that indicates whether or not the field has been selected.

device echo. A visual identification of the position of the graphics cursor. The form of the device echo is defined by the application program.

device family. In GDDM, a device classification that governs the general way in which I/O will be processed. See also *processing option*. For example:

- Family 1: 3270 display or printer
- Family 2: queued printer
- Family 3: system printer (alphanumerics only)
- Family 4: page printer

device image. In GDDM, an image contained in a device or GDDM page. Contrast with *application image*.

device suffix. In GDDM-IMD, a suffix to a mapgroup name that indicates the device class.

device token. In GDDM, an 8-byte code giving entry to a table of pre-established device hardware characteristics that are required when the device is opened (initialized).

DIF. In GDDM terms, data interchange format.

digital image. A two-dimensional array of picture elements (pixels) representing a picture. A digital image can be stored and processed by a computer, using bits to represent pixels. In GDDM, pixels have the value black or white. Often called simply *image*.

direct transmission. In GDDM image processing, the transfer of image data direct from a source outside GDDM to an image device, including manipulation by a projection in the device, and without GDDM maintaining a copy or buffer of the data.

display device. Any output unit that gives a visual representation of data; for example, a screen or printer. More commonly, the term is used to mean a screen and not a printer.

display point. Synonym for *pixel*.

display-point matrix. Synonym for *dot matrix*.

display terminal. An input/output unit by which a user communicates with a data processing system or subsystem. It usually includes a keyboard and always provides a visual presentation of data. For example, an IBM 3179 display.

DL/1. Data language 1. A language for database processing operations.

dot matrix. In computer graphics, a two-dimensional pattern of dots used for constructing a display image. This type of matrix can be used to represent characters by dots. Synonymous with *character matrix* and *display-point matrix*.

double-byte characters. See *double-byte character set (DBCS)*.

double-byte character set (DBCS). A set of characters in which each character occupies two byte positions in internal storage and in display buffers. Used for oriental languages; for example, *Kanji* or *Hangeul*. Contrast with *single-byte character set*.

DPCX. Distributed Processing Control Executive. An IBM 8100 system control program.

DPPX. Distributed Processing Programming Executive. An IBM 8100 system control program.

drawing default. The value of a graphics attribute chosen by GDDM when no value is explicitly specified

by the user. The drawing default may be altered by the user.

DSC. Data stream compatibility.

dual characters. See *double-byte characters*.

dummy device. An output destination for which GDDM does all the normal processing but for which no actual output is generated. Used, for example, to test programming for an unavailable output device.

E

EBCDIC. Extended binary coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

echo. In interactive graphics, the visible form of the locator or other logical input device.

ECSA. Extended character set adapter.

edit. To enter, modify, or delete data.

editing grid. In the GDDM Image and Vector Symbol Editors, a grid used as a guide for editing a symbol. In the Image Symbol Editor, it is a dot matrix. In the Vector Symbol Editor, it is a grid of lines.

enterprise. An organization or company that undertakes local, national, or international business ventures.

extended data stream. For IBM 3179, 3192, 3278, 3279, and 3287 devices, input/output data formatted and encoded in support of color, programmed symbols, and extended highlighting. These features extend the IBM 3270 data stream architecture.

extended highlighting. The emphasizing of a displayed character's appearance by blinking, underscore, or reverse video.

external defaults. GDDM-supplied values that users can change to suit their own needs.

extracted image. In GDDM, an image on which transform element calls operate. It may imply the whole source image or just a part of it, depending on whether a define sub-image transform element has been applied in its derivation.

F

FCT. File control table (CICS).

field. An area on the screen or the printed or plotted page. See *alphanumeric field*, *graphics field*, and *mapped field*.

field attributes. See *alphanumeric field attributes*.

field list. The high performance alphanumerics data structure used to define alphanumeric fields.

fillet. A curve that is tangential to the end points of two adjoining lines.

flat file. A file that contains only data; that is, a file that is not part of a hierarchical data structure. A flat file can contain fixed-length or variable-length records.

floating area. The part of a page reserved for *floating maps*.

floating map. A map whose absolute position on the GDDM page is not fixed. During execution, a floating map takes the next available space that satisfies its specification.

floating-point feature. A processing unit feature that provides four 64-bit floating-point registers to perform floating-point arithmetic calculations.

foil. A transparency for overhead projection.

font. A particular style of typeface (for example, Gothic English). In GDDM, a font can exist as a programmed symbol set.

formatted document. A type of file containing text, images, and graphics.

FORTRAN. One of the programming languages supported by GDDM.

four-button cursor. A hand-held device, with cross-hair sight, used on the surface of a *tablet* to indicate position on a screen. Synonymous with *puck*.

frame. In GDDM-IMD, a synonym for *panel*.

full-screen alphanumeric operation. Full-screen processing operations on alphanumeric fields.

full-screen mode. A form of screen presentation in which the contents of an entire terminal screen can be displayed at once. Full-screen mode is often used for fill-the-blanks prompting, and is an alternative to line-by-line I/O.

full-screen processor. A host software component that, together with display terminal functions, supports display terminal input/output in full-screen mode.

G

GDDM. Graphical Data Display Manager. A series of IBM licensed programs, running in a host computer, that manage communications between application programs and display devices, printers, plotters, and scanners for graphics applications.

GDDM-GKS. GDDM Graphical Kernel System. A member of the GDDM family that runs under TSO and CMS and provides an alternative graphics programming interface to that of the GDDM base product. It is an implementation of the Graphical Kernel Standard, ISO 7942, of the International Organization for Standardization.

GDDM/graPHIGS. A member of the GDDM family used for creating hierarchical three-dimensional structures on the IBM 5080 Graphics System. It is based on the proposed ANSI standard for the Programmer's Hierarchical Interactive Graphics System (PHIGS).

GDDM Interactive Map Definition. GDDM-IMD. A member of the GDDM family of licensed programs. It enables users to create alphanumeric layouts at the terminal. The user defines the position of each field within the layout and may assign attributes, default data, and associated variable names to each field. The resultant map can be tested from within the utility.

GDDM-IVU. GDDM Interactive View Utility. A member of the GDDM family of licensed programs. It enables users to view, create, modify, store, and print images.

GDDM-OS/2. A licensed program that enables IBM PS/2 and other personal-computer systems with OS/2 installed to run GDDM application programs in the host computer.

GDDM-PCLK. A licensed program that enables IBM PS/2 and other personal computers with graphics-display adapters, and IBM 3270 terminal emulators to run GDDM application programs in the host computer.

GDDM-PGF. GDDM-Presentation Graphics Facility. A member of the GDDM family of licensed programs. It is concerned with business graphics, rather than general graphics.

GDDM storage. The portion of host computer main storage used by GDDM.

GDF. Graphics data format.

general graphics. The methods and techniques for converting data to or from graphics display in mathematical, scientific, or engineering applications; that is, in any application other than business graphics. See also *business graphics*.

generated mapgroup. The output produced when a source GDDM-IMD mapgroup is generated. It contains the information needed by GDDM at execution to position the mapped fields on the GDDM page.

GIF. Graphics Interchange Format.

GKS. Graphical Kernel System. See *GDDM-GKS*.

GL. Graphical Language.

Graphical Data Display Manager. See *GDDM*.

graphics. A picture defined in terms of *graphics primitives* and *graphics attributes*.

graphics area. Part of a mapped field that is reserved for later insertion of graphics.

graphics attributes. In GDDM, color selection, color mix, line type, line width, graphics text attributes, marker symbol, and shading pattern definition.

graphics cursor. A physical indicator that can be moved (often with a joystick, mouse, or stylus) to any position on the screen.

graphics data format (GDF). A picture definition in an encoded order format used internally by GDDM and, optionally, providing the user with a lower-level programming interface than the GDDM API.

graphics data stream. The data stream that produces graphics on the screen, printer, or plotter.

graphics field. A rectangular area of a screen or printer page, used for graphics. Contrast with *alphanumeric field*.

graphics input queue. A queue associated with the graphics field onto which elements arrive from logical input devices. The program can remove elements from the queue by issuing a graphics read.

graphics primitive. A single item of drawn graphics, such as a line, arc, or graphics text string. See also *graphics segment*.

graphics read. A form of read that solicits graphics input or removes existing elements from the graphics input queue.

graphics segment. A group of graphics primitives (lines, arcs, and text) that have a common window and a common viewport and associated attributes. Graphics

segments allow a group of primitives to be subject to various operations. See also *graphics primitive*.

graphics text attributes. In GDDM, the symbol (character) set to be used, character box size, character angle, character mode, character shear angle, and character direction.

graPHIGS. See *GDDM/graPHIGS*.

gray-level. A digitally encoded shade of gray, normally (and always in GDDM) in the range 0 through 255. See also *gray-scale image*.

gray-scale image. An image in which the gradations between black and white are represented by discrete gray-levels. Contrast with *bi-level image* and *halftone image*.

green lightning. The name given to the flashing streaks on an IBM 3270 screen while a programmable symbol set is being loaded.

H

halftone image. A bi-level image in which intermediate shades of gray are simulated by patterns of adjacent black and white pixels. Contrast with *gray-scale image*.

Hangeul. A character set of symbols used in Korean ideographic alphabets.

hardware cell. Synonym for *character cell*.

hardware characters. Synonym for *hardware symbols*.

hardware symbols. The characters that are supplied with the device. The term is loosely used also for GDDM mode-1 symbols that are loaded into a PS store for subsequent display. Synonymous with *hardware characters*.

hexadecimal. Pertaining to a numbering system with base sixteen.

host. See *host computer*.

high performance alphanumerics. The creation of alphanumeric displays using field list data structures. Contrast with *procedural* and *mapped* alphanumerics.

host computer. The primary or controlling computer in a multiple-computer installation.

I

ICU. Interactive Chart Utility.

identity projection. In GDDM image processing, a projection that is transferred from source image to target image without any processing being performed on it.

image. Synonym for *digital image*.

image data stream. The internal form of the GDDM data in an image environment.

image field. A rectangular area of a screen or printer page, used for image. Contrast with *alphanumeric field* and *graphics field*.

Image Object Content Architecture (IOCA). An architected collection of constructs used to interchange and present images.

image symbol. A character or symbol defined as a dot pattern.

Image Symbol Editor (ISE). A GDDM-supplied interactive editor that enables users to create or modify their own image symbol sets (ISS).

image symbol set (ISS). A set of symbols each of which was created as a pattern of dots. Contrast with *vector symbol set (VSS)*.

IMD. See *GDDM Interactive Map definition*.

IMS/VS. Information Management System/Virtual Storage. A subsystem of MVS under which GDDM can be used.

include member. A collection of source statements stored as a library member for later inclusion in a compilation.

input queue. See *graphics input queue*.

integer. A whole number (for example, -2, 3, 457).

Intelligent Printer Data Stream (IPDS). A structured-field data stream for managing and controlling printer processes, allowing both data and controls to be sent to the printer. GDDM uses IPDS to communicate with the IBM 4224 printer.

Interactive Chart Utility (ICU). A GDDM-PGF menu-driven program that allows business charts to be created interactively by nonprogrammers.

interactive graphics. In GDDM, those graphics that can be moved or manipulated by a user at a terminal.

Interactive Map definition. A member of the GDDM family of licensed programs. It enables users to create alphanumeric layouts at the terminal. The operator defines the position of each field within the layout and may assign attributes, default data, and associated variable names to each field. The resultant map can be tested from within the utility.

interactive mode. A mode of application operation in which each entry receives a response from a system or program, as in an inquiry system or an airline reservation system. An interactive system can also be conversational, implying a continuous dialog between the user and the system.

interactive subsystem. (1) One or more terminals, printers, and any associated local controllers capable of operation in interactive mode. (2) One or more system programs or program products that enable user applications to operate in interactive mode; for example, CICS.

intercept. In a chart, a method of describing the position of one axis relative to another. For example, the x axis can be specified so that it intercepts (crosses) the y axis at the bottom, middle, or top of the plotting area of a chart.

inter-device copy. The ability to copy a page or the graphics field from the current primary device to another device. The target device is known as the alternate device.

IOCA. See *Image Object content Architecture*.

IPDS. See *Intelligent Printer Data Stream*.

ISE. Image Symbol Editor.

ISO. International Organization for Standardization.

ISPF. Interactive System Productivity Facility.

ISS. Image symbol set.

IVU. Image View Utility. See *GDDM-IVU*.

J

joystick. A lever that can pivot in all directions in a horizontal plane, used as a *locator* device.

K

Kanji. A character set of symbols used in Japanese ideographic alphabets.

Katakana. A character set of symbols used in one of the two common Japanese phonetic alphabets; Katakana is used primarily to write foreign words phonetically. See also *Kanji*.

key. In a legend, a symbol and an associated data group name. A key might, for example, indicate that the blue line on a graph represents "Predicted Profit." See also *legend*.

key symbol. A small part of a line (from a line graph) or an area (from a shaded chart) used in a legend to identify one of the various data groups.

L

Latin. Of or pertaining to the Western alphabet. In GDDM, a synonym for *single-byte character set*.

legend. A set of symbolic keys used to identify the data groups in a business chart.

line attributes. In GDDM, color, line type, and line width.

link pack area. An MVS term that describes an area of shared storage.

link-attached. Pertaining to devices that are connected to a controlling unit by a data link. Synonymous with *remote*. Contrast with *channel-attached*.

local. Synonym for *channel-attached*.

local character set identifier. A hexadecimal value stored with a GDDM symbol set, which can be used by symbol-set-loading means other than GDDM in the context of local copy on a printer.

locator. A logical input device used to indicate a position on the screen. Its physical form may be the alphanumeric cursor or a graphics cursor moved by a joystick.

logical input device. A concept that allows application programs to be written in a device-independent manner. The logical input devices to which the program refers may be subsequently associated with different physical parts of a terminal, depending on which device is used at run time.

LPA. Link pack area.

LTERM. In IMS/VS, logical terminal.

M

map. A predefined format of alphanumeric fields on a screen. Usually constructed outside of the application program.

map specification library (MSL). The data set in which maps are held in their source form.

mapgroup. A data item that contains a number of maps and information about the device on which those maps are to be used. All maps on a GDDM page must come from the same mapgroup.

mapped alphanumerics. The creation of alphanumeric displays using predefined maps. Contrast with *procedural alphanumerics* and *high performance alphanumerics*.

mapped field. An area of a page whose layout is defined by a map.

mapped graphics. Graphics placed in a graphics area within a mapped field.

mapped page. A GDDM page whose content is defined by maps in a mapgroup.

mapping. The use of a map to produce a panel from an output record, or an input record from a panel.

marker. In GDDM, a symbol centered on a point. Line graphs and polar charts can use markers to indicate the plotted points.

MDT. Modified data tag.

menu. A displayed list of logically grouped functions from which the user can make a selection. Sometimes called a menu panel.

menu-driven. Describes a program that is driven by user response to one or more displayed menus.

MFS. Message format service.

MICR. Magnetic ink character recognition.

mixed character string. A string containing a mixture of *Latin* (one-byte) and *Kanji* or *Hangeul* (two-byte) characters.

Mixed Object Document Content Architecture (MO:DCA). An architected, device-independent data stream for interchanging documents.

mode-1/-2/-3 characters. See *character mode*.

mountain shading. A method of shading surface charts where each component is shaded separately from the base line, instead of being shaded from the data line of the previous component.

mouse. A device that a user moves on a flat surface to position a pointer on a screen.

MSHP. Maintain System History Program. A software process for installing licensed programs on VSE systems.

MSL. Map specification library.

MVS. IBM Multiple Virtual Storage. A system under which GDDM can be used.

MVS/XA. Multiple Virtual Storage/Extended Architecture. A subsystem under which GDDM can be used.

N

name-list. A means of identifying which physical device is to be opened by a GDDM program. It can be used as a parameter of the DSOPEN call, or in a *nickname*.

National Language Support (NLS). A special feature that provides translations of the ICU panels and some of the GDDM messages into a variety of languages, including US English.

negate. In bi-level image data, setting zero bits to one and one bits to zero.

neutral color. White on a display, black on a printer. Contrast with *background color*.

nibblized data. The encoded data stream used between the GDDM DOS Support feature in the host and GDDM-PCLK on the workstation.

nickname. In GDDM, a means of referring to a device, the characteristics and identity of which have been already defined.

NLS. National Language Support.

nonqueriable printer. A printer about which GDDM cannot obtain any information.

NSS. Named saved system (VM/XA and VM/ESA).

null character. An empty character represented by X'00' in the EBCDIC code. Such a character does not occupy a screen position.

O

operator reply mode. In GDDM, the mode of interaction available to the operator (display terminal user) with respect to the modification (or not) of alphanumeric character attributes for an input field.

operator window. Part of the display screen's surface on which the GDDM output of an application program can be shown. An operator window is controlled by the end user; contrast with *partition*. A *task manager* may create a window for each application program it is running.

outbound structured field. An element in IBM 3270 data streams from host to terminal with formatting that allows variable-length and multiple-field data to be sequentially translated by the receiver into its component fields without the receiver having to examine every byte.

P

page. In GDDM, the main unit of output and input. All specified alphanumerics and graphics are added to the current page. An output statement always sends the current page to the device, and an input statement always receives the current page from the device.

page printer. A printer, such as the IBM 3820 or IBM 4250, to which the host computer sends data in the form of a succession of formatted pages. Such devices can print pictorial data and text, and can position all output to pixel accuracy. The pixel density and the general print quality both often suffice as camera-ready copy for publications. Also known as *composed-page printer*.

page segment. A picture file in a form that can be printed. It can only be printed if it is embedded in a primary document. Also known as a *PSEGo* file.

panel. A predefined display that defines the locations and characteristics of alphanumeric fields on a display terminal. When the panel offers the operator a selection of alternatives it may be called a menu panel. Synonymous with *frame*.

partition. Part of the display screen's surface on which a page, or part of a page, of GDDM output can be shown. Two or more partitions can be created, each displaying a page, or part of a page, of output. A partition is controlled by the GDDM application; contrast with *operator window*.

partition set. A grouping of partitions that are intended for simultaneous display on a screen.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

PCB. In GDDM, program communication block (IMS/VS).

PCLK. See *GDDM-PCLK*.

PDS. Partitioned data set (MVS).

pel. Picture element. See *pixel*.

PGF. Presentation Graphics Facility. A member of the GDDM family of licensed programs. It is concerned with business graphics, rather than general graphics.

PHIGS. Programmer's Hierarchical Interactive Graphics System.

pick. The action of the operator in selecting part of a graphics display by placing the graphics cursor over it.

pick aperture. A rectangular or square box that is moved across the screen by the graphics cursor. An item must lie at least partially within the pick aperture before it can be picked.

pick device. A logical input device that allows the application to determine which part of the picture was selected (or picked) by the operator.

picture interchange format (PIF) file. In graphics systems, the type of file, containing picture data, that can be transferred between GDDM and an IBM 3270-PC/G, /GX, or /AT workstation.

picture space. In GDDM, an area of specified aspect ratio that lies within the graphics field. It is centered on the graphics field and defines the part of the graphics field in which graphics will be drawn.

PIF. Picture interchange format.

pixel. The smallest area of a display screen capable of being addressed and switched between visible and invisible states. Synonymous with *display point*, *pel*, and *picture element*.

PL/I. One of the programming languages supported by GDDM.

plotter. An output device that uses pens to draw its output on paper or transparency foils.

pointings. Pairs of x-y coordinates produced by an operator defining positions on a screen with a locator device, such as a *mouse*.

polar chart. A form of business chart where the x axis is circular and the y axis is radial.

polyfillet. In GDDM, a curve based on a sequence of lines. It is tangential to the end points of the first and last lines, and tangential also to the midpoints of all other lines.

polyline. A sequence of adjoining lines.

popping. A method of ordering data whereby each item in a list or sequence takes the value of the previous item in the list or sequence, and is then removed from the list; when this happens, the list or sequence of data is said to be "popped."

ppi. Pixels per inch.

PQE. Printer queue element.

presentation graphics. Computer graphics products or systems, the functions of which are primarily concerned with graphics output presentation. For example, the display of business planning bar charts.

preview chart. A small version of the current chart that can be displayed on ICU menu panels.

primary device. In GDDM, the main destination device for the application program's output, usually a display terminal. The default primary device is the user console. See also *alternate device*.

primitive. See *graphics primitive*.

primitive attribute. A specifiable characteristic of a graphics primitive. See *graphics attributes* and *graphics text attributes*.

Print Services Facility (PSF). An IBM licensed program for processing documents destined for the IBM 3800 Model 3 page printer.

print utility. A subsystem-dependent utility that sends print files from various origins to a queued printer.

procedural alphanumerics. The creation of alphanumeric displays using the GDDM alphanumeric API. Contrast with *mapped alphanumerics* and *high performance alphanumerics*.

processing option. Describes how a device's I/O is to be processed. It is a device-family-dependent and subsystem-dependent option that is specified when the device is opened (initialized). An example is the choice between CMS attention-handling protocols.

procopt. Processing option.

profile. In GDDM, a file that contains information about how GDDM is to process requests for services to devices or other functions.

program library. (1) A collection of available computer programs and routines. (2) An organized collection of computer programs.

programmed symbols (PS). Dot patterns loaded by GDDM into the PS stores of an output device.

projection. In GDDM image processing, an application-defined function that specifies operations to be performed on data extracted from a source image. Consists of one or more *transforms*. See also *transform element*.

PS. Programmed symbols.

PS overflow. A condition where the graphics cannot be displayed in its entirety because the picture is too complex to be contained in the device's PS stores.

PSB. Program specification block (IMS).

PSEG. See *page segment*.

PSF. Print Services Facility.

PSP bucket. A database containing descriptions of faults found in programs. Used by Service personnel.

PS/2. Personal System/2.

puck. Synonym for *four-button cursor*.

PUT. Program update tape.

Q

quarto. A paper size, more common in the U.S. than in Europe. It measures 8.5 inches by 11.0 inches. Also known as A size. Compare with A4.

queued printer. A printer belonging to the subsystem under which GDDM runs, to which output is sent indirectly by means of the GDDM Print Utility program. In some subsystems, this may allow the printer to be shared between multiple users. Contrast with *system printer*.

R

raster device. A device with a display area consisting of dots. Contrast with *vector device*.

rastering. The transforming of graphics primitives into a dot pattern for line-by-line sequential use. In GDDM PS devices, this is done by transforming the primitives into a series of programmed symbols (PS).

glossary

real device. A GDDM device that is not being windowed by means of operator window functions. Contrast with *virtual device*.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

remote. Synonym for *link-attached*.

reply mode. See *operator reply mode*.

resolution. In graphics and image processing, the number of pixels per unit of measure (inch or meter).

reverse clipping. Where one graphics primitive overlaps another, removing any parts of the underlying primitive that are overpainted by the overlying primitive.

reverse video. A form of alphanumeric highlighting for a character, field, or cursor, in which its color is exchanged with that of its background. For example, changing a red character on a black background to a black character on a red background.

REXX. Restructured Extended Executor Language. One of the programming languages supported by GDDM.

Roman. Relating to the *Latin* type style, with upright characters.

S

SBCS. Single-byte character set.

scanner. A device that produces a digital image from a document.

scissoring. Synonym for *clipping*.

scrolling. In computer graphics, moving a display image vertically or horizontally in a manner such that new data appears at one edge as existing data disappears at the opposite edge.

SCS. SNA character string.

segment. See *graphics segment*.

segment attributes. Attributes that apply to the segment as an entity, rather than to the individual primitives within the segment. For example, the visibility, transformability, or detectability of a segment.

segment library. The portion of auxiliary storage where segment definitions are held. These definitions are GDDM objects in graphics data format (GDF) and are managed by GDDM API calls. GDDM handles the file accesses to and from auxiliary storage.

segment priority. The order in which segments are drawn; also the order in which they are detected.

segment transform. The means to rotate, scale, and reposition segments without re-creating them.

selector adjunct. A subfield of an application data structure that qualifies a data field.

shear. The action of tilting graphics text so that each character leans to the left or right while retaining a horizontal baseline.

single-byte character set (SBCS). A set of characters in which each character occupies one byte position in internal storage and in display buffers. Used for example, in most non-Oriental symbols. Contrast with *double-byte character set*.

SMP/E. System Modification Program/Extended. A software process for installing licensed programs on MVS systems.

SNA. System Network Architecture.

source image. An image that is the data input to image processing or transfer.

spill file. A means of reducing storage requirements at the cost of processing time, when creating high-resolution output files for page printers, for example.

stand-alone (mode). Operation that is independent of another device, program, or system.

standard default. The value of a graphics attribute chosen by GDDM when no value is explicitly specified by the user. The standard default cannot be altered by the user, although it may be overridden by the user.

string device. A logical input device that enables an application program to process character data entered by the terminal operator.

stroke device. A logical input device that enables an application program to process a sequence of x,y coordinate data entered by the terminal operator.

stylus. A pen-like pointer used on the surface of a tablet to indicate position on a screen.

surface chart. A chart similar to a line graph, except that no markers appear and the areas between successive lines are shaded.

swathe. A horizontal slice of printer output, forming part of a complete picture. Page printer images are often constructed in swathes to reduce the amount of storage required.

symbol. Synonymous with *character*. For example, the following terms all have the same meaning: vector symbols, vector characters, vector text.

symbol cell. Synonym for *character cell*.

symbol matrix. Synonym for *dot matrix*.

symbol set. A collection of symbols, usually but not necessarily forming a font. GDDM applications may use the hardware device's own symbol set. Alternatively, they can use image or vector symbol sets that the user has created.

symbol set identifier. In GDDM, an integer (or the equivalent EBCDIC character) by which the programmer refers to a loaded symbol set.

system printer. A printer belonging to the subsystem under which GDDM runs, to which output is sent indirectly by use of system spooling facilities. Contrast with *queued printer*.

T

tablet. (1) A locator device with a flat surface and a mechanism that converts indicated positions on the surface into coordinate data. (2) The IBM 5083 Tablet Model 2, which, with a four-button cursor or stylus, allows positions on the screen to be addressed and the graphics cursor to be moved without use of the keyboard.

tag. In interactive graphics, an identifier associated with one or more primitives that is returned to the program if such primitives are subsequently picked.

target image. An image that is the destination of processed or transferred data.

target position. In the GDDM Vector Symbol Editor, the grid coordinates of a point on the editing grid to which a vector is to be drawn.

task manager. A program that supervises the concurrent running of other programs.

temporary graphics. Graphics created outside a segment.

terminal. A device, usually equipped with a keyboard and a display unit, capable of sending and receiving information over a link. See also *display terminal*.

terminal emulator. A program that enables a device such as a personal computer system to enter and receive data from a host computer system as if it were a particular type of attached terminal.

test symbol. In the GDDM Image and Vector Symbol Editors, an area on the Symbol Edit panel in which the currently chosen symbol is displayed.

text. Characters or symbols sent to the device. GDDM provides alphanumeric text and graphics text.

text attributes. See *graphics text attributes*.

tilted pie chart. A pie chart drawn in three dimensions, which has been tilted away from full face to reveal its three-dimensional properties.

trailing attribute byte. The screen position following an alphanumeric field. This attribute byte can specify, for example, that the cursor should auto-skip to the next field when the current field is filled.

transfer operation. In GDDM image processing, an operation in which a projection is applied to a source image, and the result placed in a target image. The source and target images can be device or application images in any combination, or one or other of them (but not both) can be image data within the application program.

transform. (1) The action of modifying a picture for display; for example, by scaling, rotating, or displacing. (2) The object that performs or defines such a modification; also referred to as a *transformation*. (3) In GDDM image processing, a definition of three aspects of the data manipulation to be done by a projection:

1. A transform element or sequence of transform elements
2. A resolution conversion or scaling algorithm
3. A location within the target image for the result

Only the third item is mandatory.

See also *projection* and *transform element*.

transform element. In GDDM image processing, a specific function in a transform, which can be one of the following: define sub-image, scale, orient, reflect, negate, define place in target image.

A given transform element can be used only once in a *transform*.

transformable. A segment must be defined as transformable if it will subsequently be moved, scaled, or rotated.

transparency. (1) A document on transparent material suitable for overhead projection. (2) An alphanumeric attribute that allows underlying graphics or image to show.

TSO. Time Sharing Option. A subsystem of MVS under which GDDM can be used.

TWA. Transaction work area.

U

UDS. User default specification.

UDSL. A list of user default specifications (UDSs).

unformatted data. In GDDM image processing, compressed or uncompressed binary image data that has no headers, trailers, or embedded control fields other than any defined by the compression algorithm, if applicable. The data is in row major order, beginning with the top left of the picture.

User Control. A GDDM function that enables the terminal or workstation to perform some functions without the need for application programming. The actions include: moving and zooming graphics; manipulating windows; printing, plotting, and saving pictures.

user default specification (UDS). The means of changing a GDDM external default value. The external default values that a UDS can change are those of the GDDM or subsystem environment, GDDM user exits, and device definitions.

user exit. A point in GDDM execution where a user routine will gain control if such has been requested.

V

variable cell size. In most devices, the hardware cell size is fixed, but the IBM 3290 Information Panel has a cell size that can be varied. This, in turn, causes the number of rows or columns on the device to alter.

vector. (1) In computer graphics, a directed line segment. (2) In the GDDM-PGF Vector Symbol Editor, a straight line between two points.

vector device. A device capable of displaying lines and curves directly. Contrast with *raster device*.

vector symbol. A character or shape composed of a series of lines or curves.

Vector Symbol Editor. A program supplied with GDDM-PGF, the function of which is to create and edit vector symbol sets (VSS).

vector symbol set (VSS). A set of symbols, each of which was originally created as a series of lines and curves.

Venetian blind effect. The name given to the appearance of bars across shaded patterns on an IBM 3270-PC when GDDM tries to match the image symbol sets.

Venn diagram. A form of business chart in which, in GDDM, two or more populations and their intersection are represented by overlapping circles.

viewport. A subdivision of the picture space, most often used when two separate pictures are to be displayed together.

virtual device. A GDDM device that is being windowed by use of operator window functions. Contrast with *real device*.

virtual screen. The presentation space viewed through an *operator window*.

VM/ESA. IBM Virtual Machine Enterprise Systems Architecture.

VM/SP CMS. IBM Virtual Machine/System Product Conversational Monitor System; a system under which GDDM can be used.

VMXA. IBM Virtual Machine Extended Architecture; a system under which GDDM can be used.

VSE. Virtual storage extended; an operating system consisting of VSE/Advanced Functions and other IBM programs.

Note: In GDDM, the abbreviation VSE has sometimes been used to refer to the Vector Symbol Editor, but to avoid confusion, this usage is deprecated.

VSS. Vector symbol set.

W

Ward. One of the 190 matrices used to contain the symbols of a double-byte character set. The value in the first byte of each double-byte character code refers to the ward in which the character is contained. The value in the second byte denotes the character's position in the matrix.

window. In GDDM, the term window has three distinct meanings:

1. The "graphics window" is the coordinate space used for defining the primitives that make up a graphics picture. By default, both x and y coordinates run from 0 through 100. The graphics window can be regarded as a set of coordinates that are overlaid on the viewport.
2. An "operator window" is an independent rectangular subdivision of the screen. Several can exist at the same time, and each can receive output from, and send input to, either a separate GDDM program or a separate function of a single GDDM program.

3. The “page window” defines which part should be displayed of a page that is deeper or wider than its partition.

workstation. A display screen together with attachments such as a local copy device or a tablet.

world coordinates. The user application-oriented coordinates used for drawing graphics. See also *window*.

wrap-around field. An alphanumeric field that extends to the right-hand edge of the page and continues at the start of the next row.

WTP. Write-to-programmer.

Index

Numerics

3179-G display 4
 3193 data stream format files 22, 25
 3193 display 4, 8
 3279 display 4
 3287 printer 20
 3800 printer 20, 22, 25
 3812 printer 20, 22, 25
 3816 printer 20
 3820 printer 20, 22, 25
 4224 printer 20, 21
 4250 printer 22, 25
 8815 Scanner 22, 25

A

A (After) command 38
 add image to the name table (IUAIMG) 70
 add projection to the name table (IUAPRJ) 71
 ADM5IUX map selection libraries 93
 ADMDEFS files 21
 ADMIMG files 11
 ADMIVU exec 5
 ADMPRINT files 20
 ADMU transaction 5
 ADMU5IMG image 11, 13
 ADMUCIN5 C/370 declarations 49
 ADMUPIN5 PL/I declarations 49
 After command 38
 alphanumerics 52
 Alternate cursor key 8, 29
 APL2 47
 application programming 47
 Assembler 47

B

B (Before) command 38
 Back key 9
 background, black or white 33
 Backward PF key 7
 Before command 38
 beginning a GDDM-IVU session 4
 big images 14
 bilevel image 1
 bit maps 22, 25
 bits and pixels 1
 black background 33
 black pixel retention 33, 40
 blanking-out a section of an image 27, 30
 box cursor 8, 29

brightness of scanned image 40

C

C (Copy) command 38
 C/370 47
 C/C++ 47
 call syntax 50
 calling GDDM-IVU 47
 calls, GDDM-IVU 69
 CDPFTYPE processing option 22
 Clear key 9
 clear mode 33
 Clear PF key 7
 clearing a section of an image 27, 30
 clearing images from screen 7, 9
 COBOL 47
 COLORMAS processing option 22
 combining images 27, 30
 combining pixels 33, 40
 compiling GDDM-IVU programs 67
 compression 22, 24, 25
 continuous-tone images 1, 40
 contrast of scanned image 40
 Copy command 38
 copying a section of an image 27, 32
 copying an image or projection on disk 91
 Create Projection panel 36
 cross cursor 8, 29
 Cursor blink key 8, 29
 Cursor mode key 8, 29
 Cursor move key 8
 Cursor move keys 29
 cursors on 3193 8, 29
 cut-and-paste-type changes 27, 34

D

D (Delete) command 38
 dark documents 40
 data sets
 See files
 data types of API parameters 66
 defaults for editing 32
 Delete command 38
 delete image from the name table (IUDIMG) 72
 delete projection from the name table (IUDPRJ) 72
 deleting a projection from GDDM-IVU storage 43
 deleting a section of an image 27, 30
 deleting an image from GDDM-IVU storage 19
 deleting an image or projection from disk 91

index

- description stored with image 12, 19
- description stored with projection 41, 42
- Discard Image panel 19
- Discard Projection panel 43
- discarding a projection from GDDM-IVU storage 43
- discarding an image from GDDM-IVU storage 19
- discarding an image or projection from disk 91
- disk files
 - See files
- DISOSS interface programs 85
- display terminals 4
- displaying an image 13
 - and importing 24
 - and loading 11, 12
 - and scanning 16
- document size for scanner 17
- drawing order in projections 38

E

- Edit Defaults panel 32
- Edit Erase panel 30
- Edit Fetch panel 30
- Edit Image panel 28
- Edit Transform panel 32
- editing an image 27
- editing IVU panels 93
- entering a GDDM-IVU session 4
- erasing a projection GDDM-IVU storage 43
- erasing a section of an image 27, 30
- erasing an image from GDDM-IVU storage 19
- erasing an image or projection from disk 91
- example programs 49, 85
- executing GDDM-IVU programs 67
- Exit PF key 7
- exiting from a GDDM-IVU session 5
- Export Image panel 25
- exporting 25
- Extract Source panel 39
- Extract Target panel 40
- Extract Transform panel 39
- extract, editing 27, 33
- extract, in projections 34

F

- fetching an image 27, 30
- files
 - format 11, 66
 - listing and managing 66, 91
 - loading into GDDM-IVU storage 10
 - names of 13
 - restricting access 67
- flipping pixels 27, 32, 40
- FORTTRAN 47

- Forward PF key 7
- full screen image 7
- Fwd key 9

G

- GDDM Internet home page x
- graphics 52
- gray-level 1

H

- halftone image 1, 40
- help information, online 6
- Help key 6
- help panel
 - removing from screen 7
 - tailoring 93
- hierarchy, GDDM 52
- home page for GDDM x
- Home panel 5
- Home PF key 7
- HRIDOCNM processing option 22
- HRIFORMT processing option 22
- HRIPSIZE processing option 22
- HRISPILL processing option 22
- HRISWATH processing option 22

I

- I (Insert) command 38
- ICU (Interactive Chart Utility) 91
- image
 - introduction and examples 1
 - query identifier (IUQIID) 78
 - query name (IUQINM) 78
- image files
 - See files
- image identifier
 - query (IUQIID) 78
- image name
 - query (IUQINM) 78
- Image View Facility (IVF) 85
- Import Image panel 23
- Import Unformatted panel 24
- importing 22
- information, online 6
- initialize GDDM-IVU (IUNIT) 73
- initiating a GDDM-IVU session 4
- input of images and projections from disk files
 - See loading, importing
- Input Selection panel 10
- Insert command 38
- instructions, online 6
- Interactive Chart Utility 91

Internet home page for GDDM x
 intricate documents 40
 invoke GDDM-IVU (IUIVU) 74
 IUAIMG (add image to the name table) 70
 example 52
 IUAPRJ (add projection to the name table) 71
 example 52
 IUDIMG (delete image from the name table) 72
 examples 53
 IUDPRJ (delete projection from the name table) 72
 IUNIT (initialize GDDM-IVU) 73
 IUIVU (invoke GDDM-IVU) 74
 examples 49, 53
 IUMLO (set the maximum number of loaded
 objects) 75
 examples 53
 IUPFK (set PF key definitions) 76
 examples 51, 53
 IUQIID (query image identifier) 78
 examples 53
 IUQINM (query image name) 78
 IUQLNO (query the number of loaded objects) 80
 IUQLO (query the currently loaded object identifiers and
 names) 79
 IUQMLO (query the maximum number of loaded
 objects) 80
 IUQPFK (query PF key definitions) 81
 IUQPID (query projection identifier) 82
 IUQPNM (query projection name) 82
 IUTERM (terminate GDDM-IVU) 83
 IVF (Interactive View Facility) 85
 IVU panels, editing 93

J

jumping partitions 9

K

keyboard of 3193 8
 keys, PF 6

L

languages, programming 47
 large images 14
 light documents 40
 line documents 1
 LIST38PP files 21
 LIST4250 files 21
 listing and managing disk files 66, 91
 Load Image panel 12
 Load Projection panel 42
 Load Resolution panel 11
 loading a GDDM image 10

loading a projection 42

M

M (Move) command 38
 managing disk files 66, 91
 maps 93
 menu
 removing from screen 7, 9
 tailoring 93
 Menu PF key 7
 merging images 27, 30
 merging pixels 33, 40
 message line, removing and restoring 9
 mirror image 27, 32, 40
 mix mode 33, 40
 mixing images 27, 30
 MMR (modified modified read) 22, 24, 25
 Move command 38
 moving a section of an image 27
 MSL (map specification library) 93
 multiple extracts of images 38
 multiple images 27, 30

N

name table
 add image to (IUAIMG) 70
 add projection to (IUAPRJ) 71
 delete image from (IUDIMG) 72
 delete projection from (IUDPRJ) 72
 deleting images (IUTERM) 83
 deleting projections (IUTERM) 83
 query projection identifier (IUQPID) 82
 query projection name (IUQPNM) 82
 query the currently loaded object identifiers and
 names (IUQLO) 79
 query the maximum number of loaded objects
 (IUQMLO) 80
 query the number of loaded objects (IUQLNO) 80
 set maximum number of loaded objects
 (IUMLO) 75
 names of images 13
 names of projections 36
 national-language substitution letters for MSL
 names 93
 negating an image 27, 32, 40
 nicknames for printing and plotting 21
 No menu PF key 7

O

objects
 See files
 OFDSTYPE processing option 22

index

online help information 6
order of drawing in projections 38
output of images and projections
 See saving, exporting, printing, plotting
Output PF key 7
Output Selection panel 18
overpainting 33, 40

P

page printer 20, 22, 25
page segments 22, 25
page, GDDM 52
panels
 help, removing from screen 7
 menu, removing from screen 7, 9
 tailoring 93
panels, editing IVU 93
paper size for scanner 17
Partition enlarge key 9
Partition jump key 9
partition, menu, removing 7, 9
Personal Services/370 85
PF keys 6, 9
 information area, removing and restoring 9
 query definitions (IUQPFK) 81
 resetting and disabling 51
 set definitions (IUPFK) 76
photographs 1
 scanning 40
pixels 2
 illustration 2
 introduction 1
 merging 33, 40
 resolution when importing 24
 resolution when loading 11
 resolution when scanning 17
 reversing 27, 32, 40
PL/I 47
plotting an image 22
Print Image panel 20
printing 4, 20, 22, 25
processing options (procopts) 22
programming with GDDM-IVU calls 47
projection files
 See files
projections 34
 altering 38
 compared with editing 27, 34
 creating 35
 deleting from name table (IUDPRJ) 72
 discarding 43
 loading 42
 query identifiers (IUQPID) 82
 query names (IUQPNM) 82
 saving 41

Projectn PF key 7
PSEG38PP files 21
PSEG4250 files 21

Q

query image identifier (IUQIID) 78
query image name (IUQINM) 78
query PF key definitions (IUQPFK) 81
query projection identifier (IUQPID) 82
query projection name (IUQPNM) 82
query the currently loaded object identifiers and names (IUQLO) 79
query the maximum number of loaded objects (IUQMLO) 80
query the number of loaded objects (IUQLNO) 80

R

R (Repeat) command 38
reflecting an image 27, 32, 40
removing images from screen 9
renaming an image or projection on disk 91
Repeat command 38
resolution of image
 when importing from disk file 24
 when loading from disk file 11
 when scanning 17
restoring images and projections from disk files
 See loading
Return PF key 7
reversing an image 27, 32, 40
reversing pixels 27, 32, 40
REXX 47
rotating an image 27, 32, 40

S

Save Image panel 18
Save Projection panel 41
saving
 a projection 41
 an image in GDDM format 18
 an image in non-GDDM format 25
scale mode 33, 40
scaling an image 27, 32, 40
Scan Control panel 17
Scan Image panel 16
scanner 4
 document size 17
 resolution 17
scanning 1, 4, 16, 40
screen, filling image with 7, 9
scrolling 7, 9
segments 22, 25

- set PF key definitions (IUPFK) 76
- set the maximum number of loaded objects (IUMLO) 75
- size of image 12, 17, 29
- size of images 14
- sizing (scaling) an image 27, 32, 40
- source image
 - editing 27
 - in projections 34
- starting a GDDM-IVU session 4
- stopping a GDDM-IVU session 5
- storing images and projections in disk files
 - See saving, exporting
- subroutine calls to GDDM-IVU 47
- summary of GDDM-IVU functions 10
- syntax of GDDM-IVU calls 50

T

- tailoring GDDM-IVU panels 93
- target, editing 27
- target, in projections 34
- terminals 4
- terminate GDDM-IVU (IUTERM) 83
- terminating a GDDM-IVU session 5
- test page 45
- text 52
- tone of scanned image 40
- transforming an image 27, 32, 40
- trimming an image 27
- turning an image 27, 32, 40

U

- uncompressed image files 22, 24, 25
- Undo PF key 7
- unformatted image data 22
- unformatted image files 25

V

- View Image panel 13
- viewing an image 13
 - and importing 24
 - and loading 11, 12
 - and scanning 16

W

- white background 33
- white pixel retention 33, 40

Sending your comments to IBM

GDDM-IVU

Image View Utility

SC33-0479-02

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-IVU

Image View Utility

SC33-0479-02

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:

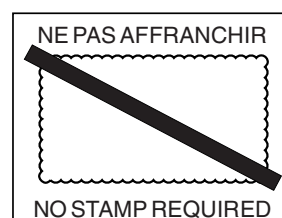
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

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.).

2 Fold along this line

By air mail
Par avion

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

**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 _____

4 Fasten here with adhesive tape**1** Cut along this line**1** Cut along this line



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

SC33-0479-02

