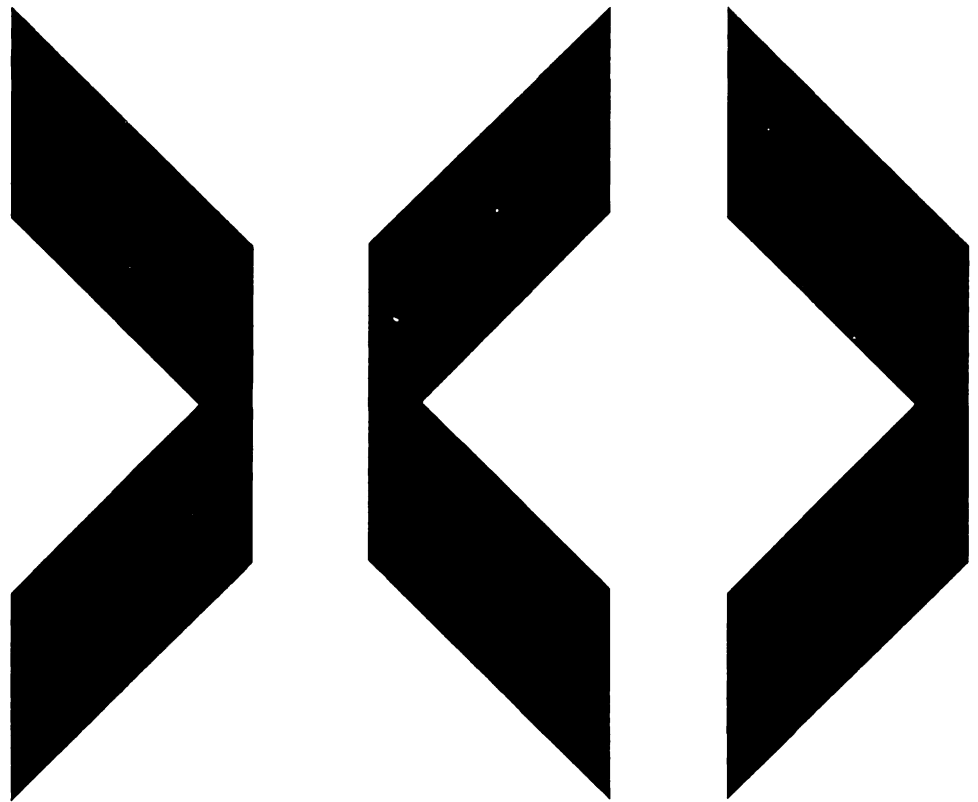




**MVS/Extended Architecture
Media Manager
Diagnosis Reference**

Licensed
Program





**MVS/Extended Architecture
Media Manager
Diagnosis Reference**

Licensed
Program

Second Edition (June 1987)

This is a major revision of, and makes obsolete, LY26-3965-0 (formerly titled MVS/XA Media Manager Diagnosis Guide and Reference).

This edition applies to Version 2 Release 3.0 of MVS/Extended Architecture Data Facility Product, Licensed Program 5665-XA2, and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Changes" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent republication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are made periodically to this publication; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370, 30xx, and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead.

Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. If you request publications from the address given below, your order will be delayed because publications are not stocked there.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

This is a licensed document that contains restricted materials of International Business Machines Corporation. © Copyright International Business Machines Corporation 1980, 1982, 1983, 1984, 1985, 1987. All rights reserved.

PREFACE

MVS/Extended Architecture Media Manager Diagnosis Reference discusses the functions performed by the media manager and how they relate to each other. It helps you to use symptoms to narrow the area of the media manager program to be examined, and to associate problem symptoms with specific functions. This information will assist you in supplying helpful detail when reporting new problems to the IBM Support Center.

ORGANIZATION

This book is organized as follows:

Chapter 1, "Program Overview" on page 3, gives an external overview of the media manager program and each of the functions it performs.

Chapter 2, "Program Functional Description" on page 5, describes the functional area of the media manager program product. This includes the function-to-function relationships, and the principal processing paths through each of the functions.

Chapter 3, "Module Directory" on page 9, provides a media manager module-to-module cross reference.

Chapter 4, "Control Blocks and Data Areas" on page 12, describes the control blocks and data area linkages used by the media manager.

Chapter 5, "Service Aids" on page 30, contains a description of the media manager dump and its output. It also describes diagnostic tools that can be used to collect information on program failures.

Appendix, "Return Codes" on page 32, lists the return codes that are provided when errors occur.

"List of Abbreviations" on page 34, defines the abbreviations used in this book.

PREREQUISITE KNOWLEDGE

In order to use this book efficiently, you should be familiar with the following topics for system control programs:

- Dump analysis
- Diagnostic practices

RELATED PUBLICATIONS

Within the text, references are made to the publications listed in the table below:

Short Title	Publication Title	Order Number
Debugging Handbook	<u>MVS/Extended Architecture Debugging Handbook</u> , Volumes 1 through 5	LC28-1164 ¹ LC28-1165 LC28-1166 LC28-1167 LC28-1168
Programming System General Information	<u>IBM Field Engineering Programming System General Information</u>	G229-2228
Service Aids	<u>MVS/Extended Architecture System Programming Library: Service Aids</u>	GC28-1159
Supervisor Services and Macro Instructions	<u>MVS/Extended Architecture System Programming Library: Supervisor Services and Macro Instructions</u>	GC28-1154
System Messages	<u>MVS/Extended Architecture Message Library: System Messages</u> , Volumes 1 and 2	GC28-1376 and GC28-1377
Utilities	<u>MVS/Extended Architecture Data Administration: Utilities</u>	GC26-4150

Note

- ¹ All five volumes may be ordered under one order number, LBOF-1015.

SUMMARY OF CHANGES

| RELEASE 3.0, JUNE 1987

The procedural diagnosis information in this manual has been moved to the new MVS/XA Data Facility Product Version 2: Diagnosis Guide. Consequentially, the title has been changed to indicate that the manual now contains only diagnosis reference information.

| NEW ENHANCEMENTS

Information was added to "Data Areas" on page 18 and the Appendix, "Return Codes" on page 32, for support of linear data sets.

RELEASE 1.0, APRIL 1985

NEW ENHANCEMENTS

- The release level keyword, which was optional in earlier releases, is now required.
- Return codes for common errors have been added in the appendix.

VERSION 2 PUBLICATIONS

The preface includes order numbers for Version 2 books.

CONTENTS

Introduction	1
Organization	1
How to Use the Reference Materials	1
Finding the Module in a Function	1
Finding the Function of a Module	2
Program Overview	3
Program Functional Description	5
CNVT Function	5
FMTWR Function	5
INIT Function	6
MMSYSTEM Function	6
PFMT Function	7
RDWR Function	7
SRV Function	8
Module Directory	9
Control Blocks and Data Areas	12
Control Block Linkages	13
Descriptions of Control Blocks and Data Areas	16
ICYCPL: Convert Parameter List	16
ICYEDB: Extent Definition Block	16
ICYIDEVB: Media Manager Initialization Device Block	16
ICYIPLST: Media Manager Initialization Parameter List	16
ICYLPMB: Logical-to-Physical Mapping Block	17
ICYMME: Media Manager Interface Block Extension	17
ICYMMIB: Media Manager Interface Block	17
ICYMMPB: Media Manager Process Block	17
Data Areas	18
MMPB	18
MMPB List Element	20
MMPB I/O Block	22
Cross-Reference for MMPB	22
ICYMMPT: Media Manager Purge Table	24
MMPT	24
Cross-Reference for MMPT	24
ICYMMRE: Media Manager Request Element	24
ICYMMSB: Media Manager Storage Block	25
MMSB	25
Cross-Reference for MMSB	26
ICYMMSV: Media Manager Storage Vector	26
MMSV	26
CROSS-REFERENCE FOR MMSV	27
ICYMMVT: Media Manager Vector Table	27
MMVT	27
Cross-Reference for MMVT	28
Service Aids	30
Appendix. Return Codes	32
List of Abbreviations	34
Index	35

FIGURES

1. Functions Invoked by Media Manager Macros	4
2. Control Block Chain, Initialization	13
3. Control Block Chain, Input/Output Processing	14
4. Media Manager Storage Block Linkages	15
5. Summary Dump Sample	30

INTRODUCTION

This publication describes the media manager in terms of its functions. For purposes of this publication, function is defined as a logically distinct group of modules that performs a major part of a program's processing.

This publication is not intended to show you the procedural steps involved in diagnosing a media manager problem, nor how to modify or repair the program logic. It will, however, give you an adequate basis for communicating with an IBM program specialist about possible program failures.

ORGANIZATION

The information contained in this book is organized to support the diagnostician in analyzing the process being performed.

This book is organized as follows:

"Program Overview" on page 3 presents an overview of the media manager, its purpose, and its functions. It is intended to provide a quick grasp of the processing performed by the media manager.

"Program Functional Description" on page 5 describes the principal processing paths through each function.

"Module Directory" on page 9 shows the module-to-module communication and the functions and sub-functions in which the module is involved. A brief description of each module is also included.

"Control Blocks and Data Areas" on page 12 describes key data areas used for communications between the functional areas, and shows the data area linkages.

"Service Aids" on page 30 contains descriptions of dumps and describes how to analyze diagnostic information for this product.

HOW TO USE THE REFERENCE MATERIALS

The following sections explain how to look for information that pertains to a specific function or module.

FINDING THE MODULE IN A FUNCTION

To find out what module carries out a specific function, follow these steps:

1. Under "Program Functional Description" on page 5 and under the function heading you have chosen, find the name of the controlling module and the modules it calls in the processing sequence. One of these modules is likely to contain the code you are looking for.
2. If you want to know which modules call and are called by the particular module, look at the entry for that module under "Module Directory" on page 9.

Let's take an example. Suppose you want to know what module builds the channel program for a read request.

1. Under "Program Overview" on page 3, find the function OP=RDWR in Figure 1. The description of OP=RDWR states that

this function processes reads and writes for the data portion of records.

2. Under "Program Functional Description" on page 5, find the section on the RDWR function.
3. Look at the descriptions of the controlling module and the modules called to perform the RDWR function. The module ICYBLDCP is likely to be what you are looking for.
4. If you want to see which modules call ICYBLDCP or which modules it calls, find the entry for ICYBLDCP under "Module Directory" on page 9.

FINDING THE FUNCTION OF A MODULE

If you want to gain a general understanding of the context (or area of function) within which a module operates, the modules that call this module, and what they do, follow these steps.

1. Under "Module Directory" on page 9, opposite the name of the module you are interested in, find the names of the functions it participates in.
2. See the descriptions of the functions in "Program Functional Description" on page 5.
3. If you want a more general view, see Figure 1.

For example. If you wish to find out what media manager function uses ICYBLIST, do the following:

1. The entry in "Module Directory" on page 9 tells you that it is used by the RDWR and FMTWR functions.
2. A description of the functions is in "Program Functional Description" on page 5.

PROGRAM OVERVIEW

This section presents an overview of the media manager program, its purpose, and its functions. It also describes how the media manager interacts with other programs.

The media manager is a component of the MVS/Extended Architecture Data Facility Product (MVS/XA DFP), Program Product Number 5665-XA2.

It enables a caller to use relative byte addresses in requests to read and write data, and to pre-format a range of relative byte addresses on a direct access storage device (DASD). With relative byte addressing, the caller is independent of the device architecture and the channel programs required for I/O.

The media manager performs the following services:

- The INIT function builds and initializes the control blocks required to perform I/O.
- The RDWR function reads or writes the data portion of a logical record (control interval) on a DASD device.
- The FMTWR function writes count fields and data sequentially on a DASD device.
- The PFMT function pre-formats one or more DASD devices for a range of relative byte addresses.
- The CNVT function changes relative byte addresses to absolute device addresses (CCHHR), and vice versa.
- The MMSYSTEM function manages media manager resources. The modules used by the MMSYSTEM function are: ICYELE, ICYFRR, and ICYPURG. In addition, a system function manages media manager resources.
- The SRV function interfaces with the integrated catalog facility catalog to obtain data set information and to update that information when explicitly requested.

Macros issued by the calling program invoke most media manager functions. There are four macros: MMINIT, MMCALL, MMSRV, and MMCNVT. Figure 1 on page 4 shows the functions invoked by each of the macros and the controlling module for each function.

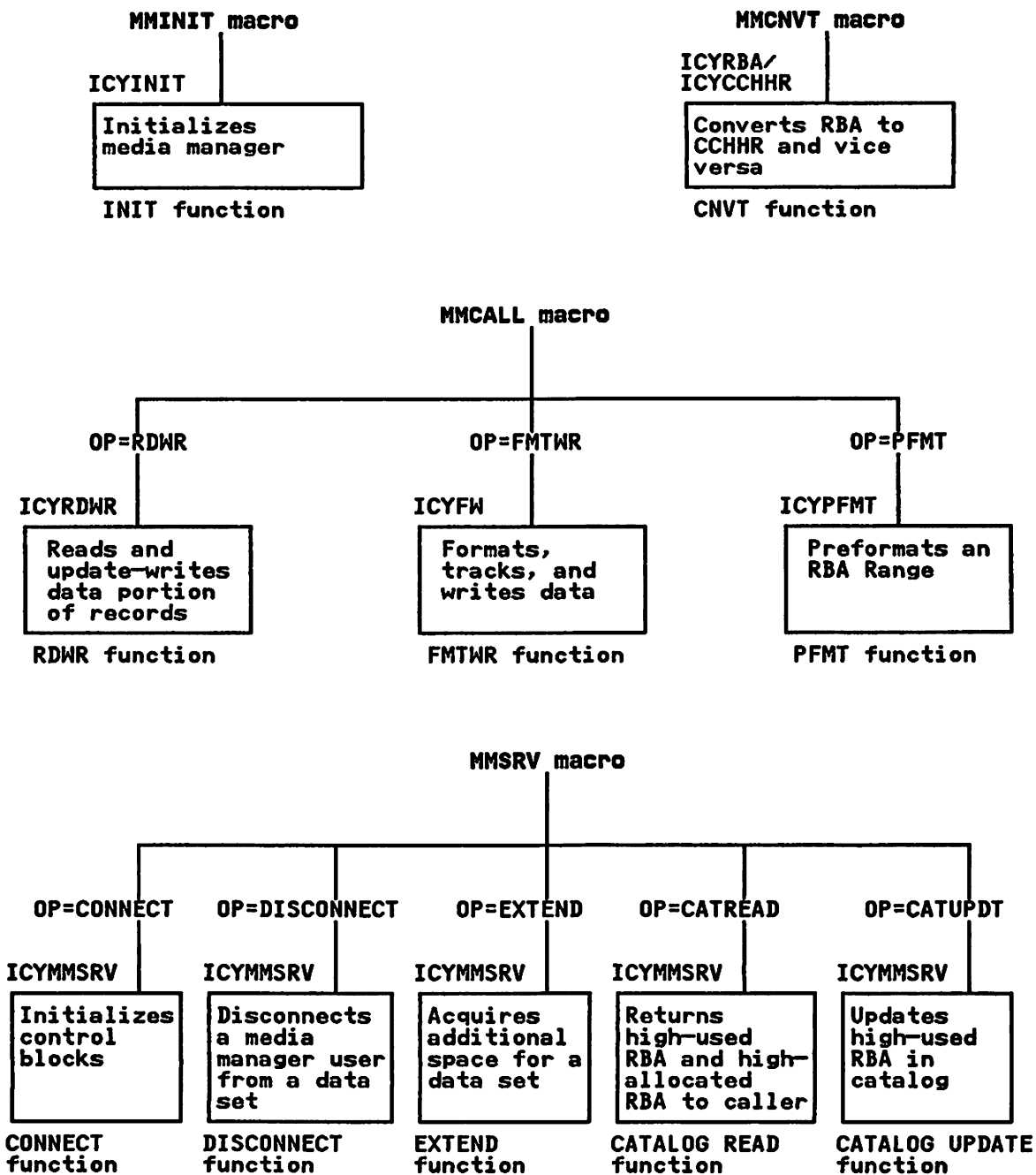


Figure 1. Functions Invoked by Media Manager Macros

PROGRAM FUNCTIONAL DESCRIPTION

This section describes the principal processing paths through each function and describes how the functions interact with each other. The functions are arranged alphabetically for easier reference.

From this section, you can relate external input to the processing of a function. The functions named in this section are those described in "Program Overview" on page 3.

The names of the functions are the titles of the main parts of this section. Under each function name is the description of the function's operation, as follows:

INPUT: Describes the input requirements for the function.

PROCESSING: Describes the processing the function performs.

RESULTS: Describes the results or output consequent to the execution of this function.

CNVT FUNCTION

INPUT: Register 0 for ICYCCHHR points to the UCB for the device containing the extent.

Register 1 points to the convert parameter list (ICYCPL).

The controlling modules are ICYRBA and ICYCCHHR.

PROCESSING: On receiving control, ICYRBA (or ICYCCHHR) proceed as follows:

- ICYRBA converts RBAs to CCHHRs.
- ICYCCHHR converts CCHHRs to RBAs.

RESULTS: RBA or CCHHR conversions are processed as specified in the calling program's request.

FMTWR FUNCTION

INPUT: Register 0 contains parameter information.

Register 1 points to the first element (MMRE) of an I/O request.

Register 2 points to the MMIB.

The controlling module is ICYFW.

PROCESSING: On receiving control, ICYFW proceeds as follows:

- Calls ICYSTOR to acquire work space and build the MMPB, indicating that requests are for format-write
- Establishes a functional recovery routine (FRR)
- Arranges the request elements in the chain in RBA sequence
- Calls ICYBLIST to build a list of control interval elements suitable for RBA-to-CCHHR conversion and channel program build
- Calls ICYRBA to convert RBAs to CCHHRs
- Calls ICYBLDCP to build I/O control blocks and channel programs

- Passes control to the I/O supervisor via the STARTIO macro to start input/output
- Returns to the calling module, and the I/O continues asynchronously

When the I/O is completed, ICYDIE/ICYNRM receives control, releases storage, and returns to the calling program via ICYPGAD, depending on the return code from the calling program's DIE exit.

When the I/O fails, ICYABN receives control and returns to the calling program always via ICYPGAD.

RESULTS: Format write requests are processed as specified by the calling program.

INIT FUNCTION

INPUT: The INIT function requires, as input, certain DASD device characteristics such as tracks per cylinder, bytes to be used on a track, and starting and ending device addresses for each extent.

Register 1 points to the calling program's parameter list that contains information needed to build EDBs and LPMBs.

The controlling module is ICYINIT.

PROCESSING: On receiving control, ICYINIT proceeds as follows:

- Builds extent definition blocks (EDBs) and logical-to-physical mapping blocks (LPMBs) as required for each extent for each volume entry specified
- Uses information in device blocks describing the physical characteristics of the data set and the device to construct LPMBs
- Initializes the media manager interface block (MMIB) with the addresses of the EDB and LPMB storage

RESULTS: The MMIB is now complete for processing calling program requests.

MMSYSTEM FUNCTION

INPUT: (No input used)

PROCESSING: The MMSYSTEM function proceeds as follows:

- ICYELE waits for notification from the system resource manager of a low system queue area (ESQA) condition and frees unused media manager storage.
- ICYFRR records information in the SYS1.LOGREC and SYS1.DUMPnn data sets for indeterminate errors or system X'COD' abends within the media manager.
- ICYPURG releases system resources held by the media manager or VSAM when a PURGE HALT request is received.

RESULTS: (No output produced)

PFMT FUNCTION

INPUT: Register 0 contains parameter information.

Register 1 points to the I/O request element (MMRE).

Register 2 points to the MMIB.

The controlling module is ICYPFMT.

PROCESSING: On receiving control, ICYPFMT proceeds as follows:

- Calls ICYSTOR to acquire work space and build the MMPB
- Establishes a functional recovery routine (FRR)
- Calls ICYPCFP to set up a channel program to format a track
- Builds the I/O control blocks to interface with the I/O supervisor
- Passes control to the I/O supervisor via STARTIO macro to start I/O
- Returns to the calling module, and I/O continues asynchronously

On return from the I/O supervisor (IOS), ICYPFMT calls ICYPCFP to format another track. If another track is required, ICYPFMT re-drives the IOS. If finished, ICYPFMT schedules ICYPGAD to return to the calling program. If an error occurs, ICYPFAPP gets control (in order to interface with user error routines, if they exist) before returning to the calling program via ICYPGAD.

RESULTS: Pre-format requests are processed as specified by the calling program.

RDWR FUNCTION

INPUT: Register 0 contains parameter information.

Register 1 points to the first element (MMRE) of an I/O request.

Register 2 points to the MMIB.

The controlling module is ICYRDWR.

PROCESSING: On receiving control, ICYRDWR proceeds as follows:

- Calls ICYSTOR to acquire work space and build the MMPB that keeps track of read/write processing
- Establishes a functional recovery routine
- Calls ICYSORT to sort the request elements in the calling program's request by relative byte address for efficiency of input/output, if specified by the calling program
- Calls ICYBLIST to build a list of control interval elements suitable for RBA-to-CCHHR conversion and channel program build
- Calls ICYRBA to convert RBAs to CCHHRs
- Calls ICYBLDCP to build I/O control blocks (SRB and IOSB) and channel programs
- Passes control to the I/O supervisor
- Returns to the calling module, and I/O continues asynchronously

When the I/O is completed, ICYDIE/ICYNRM receives control, releases storage, and returns to the calling program via ICYPGAD, depending on the return code from the calling program's DIE exit.

When the I/O fails, ICYABN receives control, and returns to the calling program, always via ICYPGAD.

RESULTS: Read/write requests are processed as specified by the calling program.

SRV FUNCTION

INPUT: The SRV function consists of five sub-functions: CONNECT, DISCONNECT, EXTEND, CATREAD, and CATUPDT. The media manager services parameter list (MMSPL) contains the DDNAME specification or the specification for the data set or a valid parameter list for sub-functions other than CONNECT. This MMSPL list is required as input. Register 1 points to the calling program's parameter list.

PROCESSING: On receiving control, ICYMMSRV proceeds as follows:

- Builds the media manager services work area (MSWA)
- Creates a dummy ACB to interface with VSAM
- Calls VSAM for CONNECT, EXTEND, DISCONNECT
 - Issues SVC 26 to obtain or update required catalog information
 - Returns to calling program

RESULTS: The sub-function is complete or a return code is passed to the calling program.

MODULE DIRECTORY

This section shows the module-to-module communication within the program. Module information is organized alphabetically and is useful when following the flow between modules and when verifying what processing has taken place. The following table lists all media manager modules, and gives for each module:

- The functions to which it belongs
- The names of the modules by which it is called
- A brief description of what it does
- The name of the modules it calls
- The name of the module to which it exits

In both the "Called by" and the "Calls" column, a name in parentheses refers to another system component, as follows:

IOS I/O supervisor
SRM system resource manager
ENF event notification facility
VSAM virtual storage access method
VSM virtual storage manager

Legend:

- < > Indicates an indirect call by scheduling an SRB.
- [] Indicates a branch.

Module	Function	Called by	Description	Calls	Exits to
ICYABN	RDWR FMTWR	(IOS)	IOS interface, abnormal end appendage		IOS +(0 or 8) ICYNRM
ICYBLDCP	RDWR FMTWR	ICYDIE ICYFW ICYPGAD ICYRDWR	Builds channel program	ICYSTOR	calling program
ICYBLIST	RDWR FMTWR	ICYFW ICYRDWR	Builds a list of control interval elements suitable for RBA conversion and channel program build	ICYSTOR	calling program
ICYCCHHR	CNVT PFMT	ICYPFAPP	Converts device address to relative byte address		calling program
ICYDIE	RDWR FMTWR	(IOS)	Interface between IOS and calling program's DIE routine for read-write and format-write requests	ICYBLDCP <ICYPGAD> ICYRBA ICYSTOR	IOS +(0, 4 or 8)

Restricted Materials of IBM
Licensed Materials - Property of IBM

Module	Function	Called by	Description	Calls	Exits to
ICYELE	MMSYSTEM	(SRM, ENF)	Upon notification from ENF, frees media manager storage blocks		calling program
ICYFRR	MMSYSTEM	(RTM)	Functional recovery routine		calling program
ICYFW	FMTWR		Processes format-write requests	ICYBLDCP ICYBLIST ICRBA ICYSTOR	calling program
ICYIEDB	INIT	ICYINIT	Builds extent definition block	ICYILPMB	calling program
ICYILPMB	INIT	ICYIEDB	Builds logical-to-physical mapping block		calling program
ICYINIT	INIT		Initializes media manager control blocks	ICYIEDB	calling program
ICYMMSRV	SRV		Access the integrated catalog facility catalog to build a media manager control block structure	(VSAM)	calling program
ICYMSTAE	SRV	(RTM)	ICYMMSRV ESTAE routine		calling program
ICYNRM	RDWR FMTWR	[ICYABN] (IOS)	IOS interface, normal end appendage		calling program
ICYPFAPP	PFMT	(IOS)	IOS interface, appendage that pre-formats by RBA range	ICYCCHHR	calling program
ICYPFCP	PFMT	ICYPFMT ICYPFDIE	Builds channel program to preformat within an RBA range	ICRBA ICYSTOR	calling program
ICYPFDIE	PFMT	(IOS)	Interface between IOS pre-format function for redriving I/O until all tracks are formatted	ICYPFCP <ICYPGAD>	Same as ICYDIE
ICYPFMT	PFMT		Processes requests to pre-format within an RBA range	ICYPFCP ICYSTOR	calling program
ICYPGAD	RDWR FMTWR	<ICYDIE> <ICYPFDIE> (IOS)	IOS interface, termination routine	ICYBLDCP ICRBA ICYSTOR	system
ICYPURG	MMSYSTEM	(IOS)	IOS interface, purge routine		calling program
ICRBA	RDWR FMTWR PFMT CNVT	ICYDIE ICYFW ICYPFCP ICYPGAD ICYRDWR	Converts relative byte address to CCHHR address		calling program

**Restricted Materials of IBM
Licensed Materials - Property of IBM**

Module	Function	Called by	Description	Calls	Exits to
ICYRDWR	RDWR		Processes read and update-write requests	ICYBLDCP ICYBLIST ICyrBA ICYSORT ICYSTOR	calling program
ICYSORT	RDWR	ICYRDWR	Sorts I/O request elements for input/output efficiency		calling program
ICYSTOR	RDWR FMTWR PFMT	ICYRDWR ICYFW ICYPFMT ICYBLIST ICYBLDCP ICYDIE ICYPGAD ICYPFCP	Acquires space in main storage for the media manager	(VSM) (lock manager)	calling program

CONTROL BLOCKS AND DATA AREAS

This section describes the data areas (control blocks) used for communication between functions, programs, or parts of the system. This information is useful for interpreting storage dumps and for determining the program's and system's status at the time of failure. Only those data areas are documented that contain fields that direct processing from one function to another.

The media manager uses control blocks and data areas to map relative byte addresses to the actual device address. The media manager also builds internal control blocks to allow control of the I/O process and to maintain storage that is allocated to the media manager.

Where appropriate, key data areas have been provided. These are listed following the related control block.

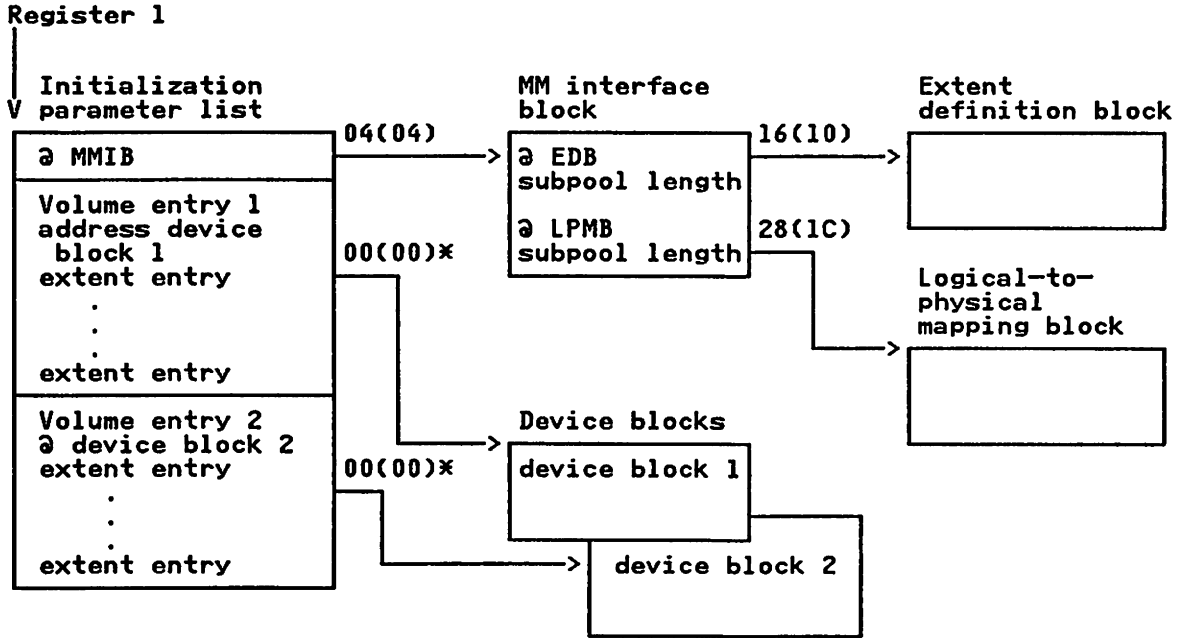
Media manager functions use control blocks and data areas as follows:

Functions	Data Areas
INIT	EDB, IDEVB, IPLST, LPMB, MMIB
RDWR, FMTWR, PFMT	EDB, LPMB, MMIB, MMPB, MMPT, MMRE, MMSB, MMSV, MMVR
CNVT	CPL (part of MMPB for use by media manager; separate data area for calling programs)
SRV	MMSPL, MSWA

Note: The prefix ICY is part of the mapping macro names on microfiche.

CONTROL BLOCK LINKAGES

Figure 2 shows a control block chain for initialization;
 Figure 3 on page 14 shows a control block chain for I/O
 processing.



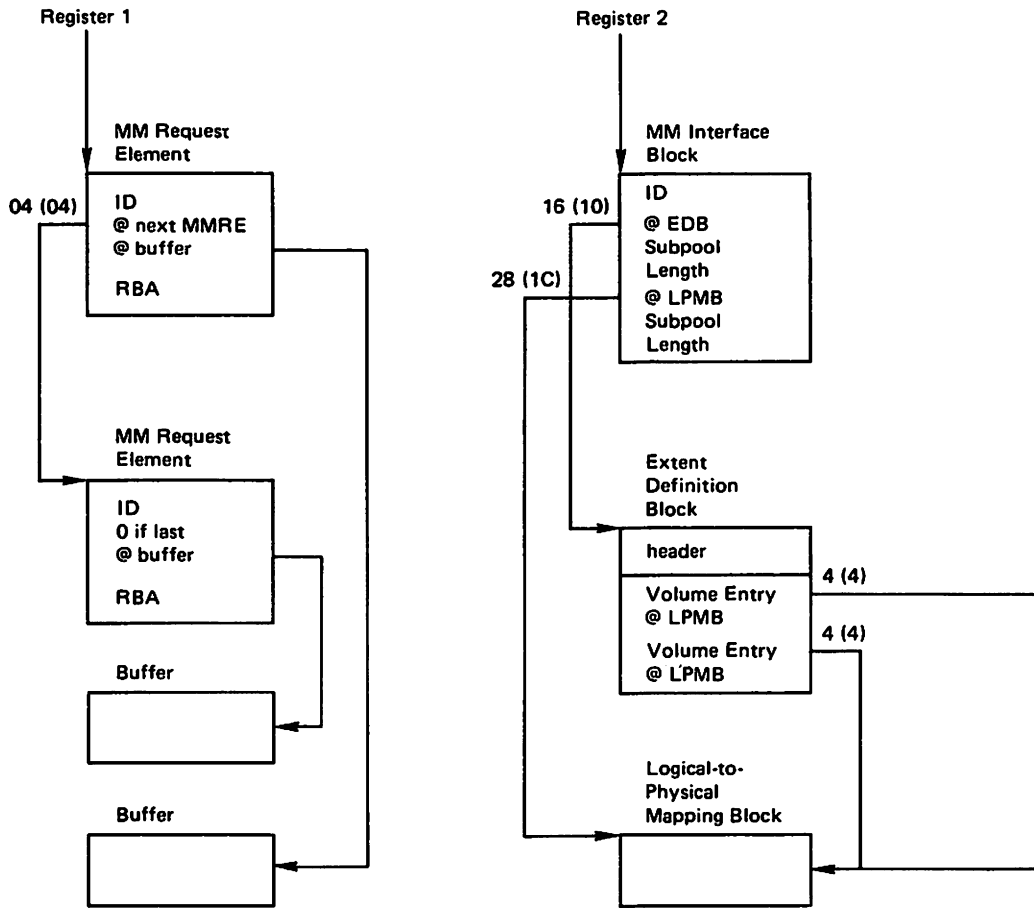
Legend:

@ Indicates pointer to.

Note:

* The offset is taken from the beginning of the entry.

Figure 2. Control Block Chain, Initialization



Notes: 1. The offset is taken from the beginning of the entry.
 2. @ is pointer to.

Figure 3. Control Block Chain, Input/Output Processing

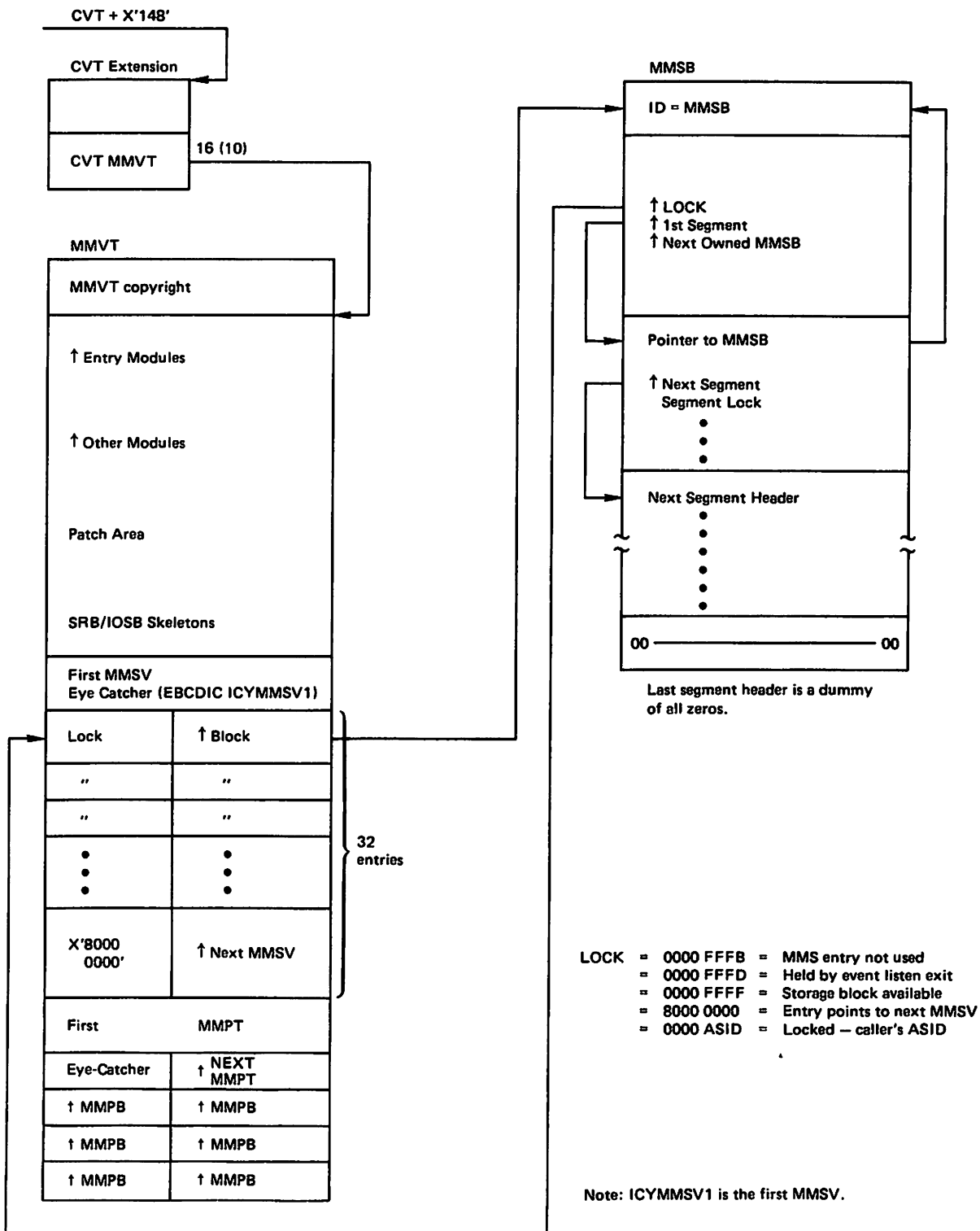


Figure 4. Media Manager Storage Block Linkages

DESCRIPTIONS OF CONTROL BLOCKS AND DATA AREAS

Following is a brief description of control blocks and data areas needed for diagnosis in the media manager.

ICYCPL: CONVERT PARAMETER LIST

Pointed to by: Register 1.

Storage Location: Specified by calling program.

Description: Used by the convert routines to change RBA to CCHHR and vice versa.

Created: By calling program.

Released: By calling program.

ICYEDB: EXTENT DEFINITION BLOCK

Pointed to by: ICYMMIB.

Storage Location: Specified by calling program or obtained in MMINIT processing.

Description: Describes data set extents. There are one or more extent definition blocks for each data set. The extent descriptions in an EDB are in ascending RBA sequence. For each extent, the description includes the address of the associated LPMB, the starting and ending RBA, and the address of the UCB for the device associated with this extent.

Created: By execution of the MMINIT macro.

Released: By calling program.

ICYIDVB: MEDIA MANAGER INITIALIZATION DEVICE BLOCK

Pointed to by: ICYIPLST.

Storage Location: Specified by calling program.

Description: Contains device-related data set information for a device type. Includes the length of a block, the unit of allocation, the physical block size, the control interval size, and the control area size for a specific device type.

Created: By calling program.

Released: By calling program.

ICYIPLST: MEDIA MANAGER INITIALIZATION PARAMETER LIST

Pointed to by: Register 1.

Storage Location: Specified by calling program.

Description: Has a pointer to the MMIB. Contains data set information for one or more volumes. Each volume entry includes the addresses of the unit control block (UCB) and device block for the device on which the volume is to be mounted; the low and high RBAs; and the beginning and ending cylinder and head addresses.

Created: By calling program.

Released: By calling program.

**Restricted Materials of IBM
Licensed Materials - Property of IBM**

ICYLPMB: LOGICAL-TO-PHYSICAL MAPPING BLOCK

Pointed to by: ICYEDB.

Storage Location: Specified by calling program or obtained in MMINIT processing.

Description: Maps the descriptions of the data set extents in an EDB to the physical characteristics of the device type on which the volume resides.

Created: By execution of the MMINIT macro.

Released: By calling program.

ICYMME: MEDIA MANAGER INTERFACE BLOCK EXTENSION

Pointed to by: ICYMMIB, ICYMME.

Storage Location: Specified by calling program.

Description: Describes an extension of MMIB storage.

Created: By calling program.

Released: By calling program.

ICYMMIB: MEDIA MANAGER INTERFACE BLOCK

Pointed to by: ICYMMPB, ICYIPLST, Register 2.

Storage Location Register 2: Specified by calling program.

Description: Provides linkages between the media manager and the calling program requesting I/O services. Includes pointers to the EDB and LPMB, the length of those blocks, and sub-pool information; and the addresses of the calling program's exit routines.

Created: By calling program.

Released: By calling program.

ICYMMPB: MEDIA MANAGER PROCESS BLOCK

Pointed to by: IOSB (IOSUSE); register 11 for the majority of media manager processing.

Storage Location: System queue area, within an MMSB.

Description: Describes the process being performed. Flags indicate whether the process is read, update-write, format-write, or pre-format. A 1-byte module identifier in field MMPMODID of the MMPB at offset X'1C' identifies the last media manager module in control.

Created: During request initialization.

Released: At request termination.

DATA AREAS

MMPE

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	564	MMPB	MEDIA MANAGER PROCESS BLOCK
0 (0)	4	MMPBID	EBCDIC IDENTIFIER 'MMPB'
4 (4)	12	MMPBPARM	INPUT PARAMETERS R0, R1, R2
4 (4)	4	MMPREG0	REGISTER 0 AT ENTRY
4 (4)	3	MMPFLG1	INPUT FLAGS
	1... ..	MMPNBFC	DO NOT DO BUFFER ADDRESS CHECK
	.1.. ..	MMPNSRT	DO NOT SORT MMRE CHAIN
	..1.	MMPREAL	MMRE HAS REAL (NOT VIRTUAL) BUFFER ADDRESSES
	...1	MMPNFRR	FRR NOT WANTED FOR FRONT END (USE EUT FRR)
 1...	MMPNDSBL	DO NOT DISABLE IN FRONT END
1..	MMPACSEQ	SEQUENTIAL ACCESS
1.	MMPDEPNO	REQUESTS ARE NOT DEPENDENT
1	MMPBYPC	BYPASS CACHE LOAD
7 (7)	1	MMPKEY	PROTECT KEY FOR I/O
8 (8)	4	MMPMMRE	POINTER TO MMRE CHAIN
12 (C)	4	MMPMMIB	POINTER TO MMIB
16 (10)	4	MMPUSAV	USERS SAVE AREA REGISTER
20 (14)	4	MMPASID	ASID FOR COMPARE AND SWAP
20 (14)	2	MMPASIDR	RESERVED
22 (16)	2	MMPASIDH	ASID USED FOR SDUMP
24 (18)	4	MMPPTCBA	PURGE TCB ADDRESS
28 (1C)	96	MMPCLEAR	PORTION OF MMPB TO BE CLEARED AT MMPB CREATION
28 (1C)	1	MMPMODID	FUNCTION CODE PORTION OF THE RETURN CODE
29 (1D)	1	MMPFLG2	GENERAL FLAGS (SOME ARE DECLARED BELOW AS ABNORMAL)
	1... ..	MMPERR	AT LEAST ONE MMRE IN THE REQUEST WAS IN ERROR

Restricted Materials of IBM
 Licensed Materials - Property of IBM

Offset Dec. (Hex.)	Length	Name	Description
	.1..	MMPDEP	DEPENDENCIES EXIST BETWEEN MMRES IN THE REQUEST
	..1.	MMPNTERM	DO NOT ENTER USERS TERMINATION ROUTINE
	...1	MMPFMTWR	FORMAT WRITE REQUEST
 1...	MMPPRFMT	PREFORMAT REQUEST
1..	MMPFINIO	ALL IOSBS FINISHED
1.	MMP PCK	PCK IN IOS OR FRR ENTERED
1	MMPPURGE	PURGE ENABLED
29 (1D)	1	MMPABNL	FLAGS DECLARED ABNORMALLY
	1...	MMPERRAB	SAME AS MMPERR
	.111 11..		
1.	MMP PCKAB	SAME AS MMP PCK
30 (1E)	1	MMP2FLG	FLAGS FOR USE BY SECOND LEVEL MODULES
31 (1F)	1	MMPFLG3	RESERVED
	1...	MMPIOP	I/O PREVENTED
	.111 1111		NOT USED
32 (20)	4	MMPMMSB	ADDRESS OF FIRST OWNED MMSB (LAST MMSB ACQUIRED)
36 (24)	4	MMPFSEG	ADDRESS FIRST STORAGE SEGMENT (LAST SEGMENT ACQUIRED)
40 (28)	4	MMPFREEP	FIRST FREE BYTE
44 (2C)	4	MMPFREEL	FREE LENGTH
48 (30)	4	MMPLISNP	ADDRESS OF LIST ELEMENT FOR NEXT CI TO PROCESS
52 (34)	4	MMPBLKCT	PHYSICAL RECORDS
56 (38)	4	MMPIOSBS	NUMBER OF IOSBS STARTED
60 (3C)	4	MMPIOSBF	NUMBER OF IOSBS FINISHED
64 (40)	2	MMPIOSBC	NUMBER OF IOSBS
66 (42)	2	MMPLISTC	NUMBER OF ELEMENTS IN MMPLIST
68 (44)	2	MMPCPAL	LENGTH OF CHANNEL PROGRAM AREA
70 (46)	1	MMPFRLK	LOCK FOR ICYFREE. 00 = DISABLED, FF = ENABLED
71 (47)	1	MMP SMSK	SAVE AREA FOR SYSTEM MASK
72 (48)	4	MMPCPAP	ADDRESS CHANNEL PROGRAM AREA
76 (4C)	4	MMPRETCD	RETURN CODE

Offset Dec. (Hex.)	Length	Name	Description
76 (4C)	2	MMPRETSS	SSSS
78 (4E)	1	MMPRETF	FF
79 (4F)	1	MMPRETCC	CC
80 (50)	4	MMPURET	RETURN CODE FROM USER
84 (54)	4	MMPMMPT	ADDRESS OF MMPT ENTRY
88 (58)	4	MMPIPIB	ADDRESS OF IPIB IF PURGE IN PROGRESS
92 (5C)	4	MMPUPURG	ADDRESS OF USER PURGE EXIT
96 (60)	4	MMPFRRWA	ADDRESS OF FRR WORK AREA
100 (64)	24	MMPLIST1	FIRST ELEMENT IN MMPLIST
124 (7C)	232	MMPIO1	FIRST IOSB AREA
356 (164)	72	MMPMSAV	SAVE AREA FOR CALLING ICYSTOR AND ICYRBA
428 (1AC)	64	MMPSAVE2	GENERAL SAVE AREA
492 (1EC)	8	MMP2WA	WORK AREA
500 (1F4)	64	MMPSAVE3	SAVE AREA FOR FRONT END WHEN BACK END MAY BE PROCESSING. MAY NOT BE USED BY BACK END

MMPB LIST ELEMENT

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	24	MMPLIST	CI LIST ELEMENTS FOR WHICH I/O WILL BE DONE
0 (0)	4	MMPMMREP	ADDRESS OF THE MMRE
4 (4)	1	MMPLISTF	FLAGS
	1...	MMPLREQE	LAST LIST ENTRY FOR REQUEST
	.1..	MMPLSEGE	LAST LIST ENTRY FOR SEGMENT
	..1.	MMPLWRIT	WRITE THIS CI (OFF FOR READ)
	...1 1...	MMPRSV01	RESERVED
1..	MMPLSP	SEARCH PREV. CHANNEL PROGRAM
1.	MMPLCNTL	THIS IS A CONTROL ELEMENT
1	MMPLXERR	EXTENT ERROR FOR LIST ELEMENT
5 (5)	1	MMPLCI	RELATIVE CI IN MMRE (0 TO 14)
6 (6)	8	MMPLB	BEGINNING BBCCHHR & SECTOR

Restricted Materials of IBM
 Licensed Materials - Property of IBM

Offset Dec. (Hex.)	Length	Name	Description
6 (6)	7	MMPLBSK	BEGINNING BBCCHHR
6 (6)	2	MMPLBB	BEGINNING BB
6 (6)	1	MMPLSTF1	FLAGS SET BY RBA CONVERT FOR CHANNEL PROGRAM BU
	1...	MMPLCTIG	THIS CI CONTIGUOUS WITH PRIOR CI
	.1..	MMPLSCYL	CI ON SAME CYLINDER AS PREVIOUS CI
	..1.	MMPLSTRK	CI ON SAME TRACK AS PREVIOUS CI
	...1	MMPLSEXT	CI IN SAME EXTENT AS PREVIOUS CI
	... 1..	MMPLSRW	CI IS SAME READ/WRITE OPERATION AS PREVIOUS CI
1..	MMPLECKD	EXTENT IF FOR FULL ECKD
1..	MMPLLR	LOCATE RECORD SUPPORTED
1.	MMPLDE	DEFINE EXTENT SUPPORTED
1	MMPRSV02	RESERVED
7 (7)	1	MMPLSTF2	FLAGS FOR USE BY CHAN PROG BUILD
8 (8)	5	MMPLBCHR	BEGINNING CCHHR
8 (8)	4	MMPLBRBA	BEGINNING RBA
8 (8)	4	MMPLBCH	BEGINNING CCHH
8 (8)	2	MMPLBC	BEGINNING CC
10 (A)	2	MMPLBH	BEGINNING HH
12 (C)	1	MMPLBR	BEGINNING R
13 (D)	1	MMPLSSEC	STARTING SECTOR
14 (E)	6	MMPLE	ENDING CCHHR & SECTOR
14 (E)	5	MMPLECHR	ENDING CCHHR
14 (E)	4	MMPLECH	ENDING CCHH
14 (E)	2	MMPLEC	ENDING CC
16 (10)	2	MMPLEH	ENDING HH
18 (12)	1	MMPLER	ENDING R
19 (13)	1	MMPLESEC	ENDING SECTOR
20 (14)	4	MMPLEDBP	ADDRESS OF EDB EXTENT FOR THIS CONTROL INTERVAL
20 (14)	4	MMPLNEXT	ADDRESS OF MORE LIST ELEMENTS (VALID IF MMPLCNT ON)

MMPB I/O BLOCK

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	232	MMPIO	SRB/IOSB AREA. THE IOSB IS THE FIRST ITEM IN THE AREA AND IOSUSE POINTS TO THE MMPB
0 (0)	108	MMPIOSB	IOSB
108 (6C)	44	MMPSRB	SRB
152 (98)	4	MMPIONP	ADDRESS OF NEXT MMPIO AREA
156 (9C)	5	MMPIOCHR	CCHHR WORK AREA FOR USE BY NORMAL APPENDAGE
156 (9C)	2	MMPIOC	CC
158 (9E)	2	MMPIOH	HH
160 (A0)	1	MMPIOR	R
161 (A1)	1	MMPIOF2	FOR USE BY DIE AND APPENDAGES
162 (A2)	2	MMPIOF1	FLAGS SPECIFIC TO THIS IOSB
	1... ..	MMPIOERR	AT LEAST ONE MMRE IN ERROR
	.1... ..	MMPCPINC	SEGMENT HAS BEEN BROKEN FOR ERROR RECOVERY AND INCOMPLETE WHEN I/O FINISHES
164 (A4)	64	MMPIOSAV	SAVE AREA FOR APPENDAGES
228 (E4)	4	MMPCFRBA	ADDRESS OF FIRST CI LIST ENTRY FOR PORTION OF THE SEGMENT CURRENTLY SCHEDULED TO IOS

CROSS-REFERENCE FOR MMPB

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
MMPABNL	29 (1D)	MMPACSEQ	4 X'04'	MMPASID	20 (14)
MMPASIDH	22 (16)	MMPASIDR	20 (14)	MMPB	0 (0)
MMPBID	0 (0)	MMPBLKCT	52 (34)	MMPBPARAM	4 (4)
MMPBYPCCL	4 X'01'	MMPCFRBA	228 (E4)	MMPCLEAR	28 (1C)
MMPCPAL	68 (44)	MMPCPAP	72 (48)	MMPCPINC	162 X'40'
MMPDEP	29 X'40'	MMPDEPNO	4 X'02'	MMPERR	29 X'80'
MMPERRAB	29 X'80'	MMPFINIO	29 X'04'	MMPFLG1	4 (4)
MMPFLG2	29 (1D)	MMPFLG3	31 (1F)	MMPFMTWR	29 X'10'
MMPFREEL	44 (2C)	MMPFREEP	40 (28)	MMPFRLK	70 (46)

Restricted Materials of IBM
 Licensed Materials - Property of IBM

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
MMPFRRWA	96 (60)	MMPFSEG	36 (24)	MMPIO	0 (0)
MMPIOC	156 (9C)	MMPIOCHR	156 (9C)	MMPIOERR	162 X'80'
MMPIOF1	162 (A2)	MMPIOF2	161 (A1)	MMPIOH	158 (9E)
MMPIONP	152 (98)	MMPIOP	31 X'80'	MMPIOR	160 (A0)
MMPIOSAV	164 (A4)	MMPIOSB	0 (0)	MMPIOSBC	64 (40)
MMPIOSBF	60 (3C)	MMPIOSBS	56 (38)	MMPIOI	124 (7C)
MMPIPIB	88 (58)	MMPKEY	7 (7)	MMPLB	6 (6)
MMPLBB	6 (6)	MMPLBC	8 (8)	MMPLBCH	8 (8)
MMPLBCHR	8 (8)	MMPLBH	10 (A)	MMPLBR	12 (C)
MMPLBRBA	8 (8)	MMPLBSK	6 (6)	MMPLCI	5 (5)
MMPLCNTL	4 X'02'	MMPLCTIG	6 X'80'	MMPLDE	6 X'02'
MMPLE	14 (E)	MMPLEC	14 (E)	MMPLECH	14 (E)
MMPLECHR	14 (E)	MMPLECKD	6 X'04'	MMPLEDBP	20 (14)
MMPLEH	16 (10)	MMPLER	18 (12)	MMPLESEC	19 (13)
MMPLISNP	48 (30)	MMPLIST	0 (0)	MMPLISTC	66 (42)
MMPLISTF	4 (4)	MMPLIST1	100 (64)	MMPLLR	6 X'04'
MMPLNEXT	20 (14)	MMPLREQE	4 X'80'	MMPLSCYL	6 X'40'
MMPLSEGE	4 X'40'	MMPLSEXT	6 X'10'	MMPLSP	4 X'04'
MMPLSRW	6 X'08'	MMPLSSEC	13 (D)	MMPLSTF1	6 (6)
MMPLSTF2	7 (7)	MMPLSTRK	6 X'20'	MMPLWRIT	4 X'20'
MMPLXERR	4 X'01'	MMPMMIB	12 (C)	MMPMMPT	84 (54)
MMPMMRE	8 (8)	MMPMMREP	0 (0)	MMPMMSB	32 (20)
MMPMODID	28 (1C)	MMPNBFCK	4 X'80'	MMPNDSBL	4 X'08'
MMPNFRR	4 X'10'	MMPNSRT	4 X'40'	MMPNTERM	29 X'20'
MMPPCK	29 X'02'	MMPPCKAB	29 X'02'	MMPPRFMT	29 X'08'
MMPPTCBA	24 (18)	MMPPURGE	29 X'01'	MMPREAL	4 X'20'
MMPREG0	4 (4)	MMPRETCC	79 (4F)	MMPRETCD	76 (4C)
MMPRETFF	78 (4E)	MMPRETSS	76 (4C)	MMPRSV01	4 X'18'
MMPRSV02	6 X'01'	MMPSAVE2	428 (1AC)	MMPSAVE3	500 (1F4)
MMPSMSAV	356 (164)	MMPSMSK	71 (47)	MMPSRB	108 (6C)
MMPUPURG	92 (5C)	MMPURET	80 (50)	MMPUSAV	16 (10)
MMP2FLG	30 (1E)	MMP2WA	492 (1EC)		

ICYMMPT: MEDIA MANAGER PURGE TABLE

Pointed to by: ICYMMPT.

Storage Location: Nucleus or system queue area.

Description: Contains addresses of active media manager process blocks.

Created: As required during request processing.

Released: Not released.

MMPT

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	256	ICYMMPT	MEDIA MANAGER PURGE TABLE
0 (0)	8	MPTHDR	HEADER
0 (0)	4	MPTID	ID = MMPT
4 (4)	4	MPTNEXT	NEXT MMPT
8 (8)	248	MPTMMPB	POINTER TO MMPB OR ZERO IF NONE

CROSS-REFERENCE FOR MMPT

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
ICYMMPT	0 (0)	MPTHDR	0 (0)	MPTID	0 (0)
MPTMMPB	8 (8)	MPTNEXT	4 (4)		

ICYMMRE: MEDIA MANAGER REQUEST ELEMENT

Pointed to by: ICYMMPB, Register 1.

Storage Location: Specified by calling program.

Description: Used by the calling program to communicate requests for data to the media manager. Flags indicate whether a read or a write is requested. Includes the number of control intervals in the request, and the address of the next request element, if applicable.

Created: By calling program.

Released: By calling program.

**Restricted Materials of IBM
Licensed Materials - Property of IBM**

ICYMMSB: MEDIA MANAGER STORAGE BLOCK

Pointed to by: ICYMMSV.

Storage Location: System queue area.

Description: Describes the storage blocks that are owned and managed by the media manager.

Created: As required during request processing.

Released: By ICYELE when signaled by the system resource manager if the block is not in use.

MMSB

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	4096	MMSB	MEDIA MANAGER STORAGE BLOCK
0 (0)	16	MMSHDR	HEADER
0 (0)	4	MMSID	CHARACTERS MMSB
4 (4)	4	MMSLOCKP	ADDRESS OF LOCK FOR BLOCK (IN MMSV)
8 (8)	4	MMSFSEG	ADDRESS OF FIRST SEGMENT
12 (C)	4	MMSMMSB	ADDRESS OF NEXT MMSB ON OWNED CHAIN
16 (10)	4080	MMSBSEGS	AREA FOR SEGMENTS
0 (0)	16	MMSBSEG	ALLOCATED OR FREE SEGMENT
0 (0)	16	MMSHDR	SEGMENT HEADER
0 (0)	4	MMSMMSB	ADDRESS OF MMSB
4 (4)	4	MMSCHN	ADDRESS OF NEXT SEGMENT IN THIS MMSB
8 (8)	4	MMSUCHN	ADDRESS OF NEXT SEGMENT USED BY THIS REQUEST
12 (C)	4	MMSASID	ASID OF THE OWNER OF THIS SEGMENT. FFFF IF FREE
16 (10)	0	MMSSTG	REST OF SEGMENT

CROSS-REFERENCE FOR MMSB

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
		MMSB	0 (0)	MMSBSEG	0 (0)
MMSBSEGS	16 (10)	MMSFSEG	8 (8)	MMSHDR	0 (0)
MMSID	0 (0)	MMSLOCKP	4 (4)	MMSMMSB	12 (C)
MMSSASID	12 (C)	MMSSCHN	4 (4)	MMSSHDR	0 (0)
MMSSMMSB	0 (0)	MMSSSTG	16 (10)	MMSSUCHN	8 (8)

ICYMMSV: MEDIA MANAGER STORAGE VECTOR

Pointed to by: ICYMMSV.

Storage Location: Nucleus or system queue area.

Description: Contains the addresses of media manager storage blocks.

Created: As required during request processing.

Released: Not released.

MMSV

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	256	ICYMMSV	MEDIA MANAGER STORAGE VECTOR
0 (0)	256	MMSVENTS	INDIVIDUAL ENTRIES
0 (0)	8	MMSVENT	MMSV ENTRY
0 (0)	4	MMSVLOCK	LOCK FOR MMSB
	1... ..	MMSVCHN	CHAIN ENTRY (BITS 1 THRU 31 are 0's)
4 (4)	4	MMSVADDR	ADDR IF MMSB OR NEXT MMSV

CROSS-REFERENCE FOR MMSV

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
		ICYMMSV	0 (0)	MMSVADDR	4 (4)
MMSVCHN	0 X'80'	MMSVENT	0 (0)	MMSVENTS	0 (0)
MMSVLOCK	0 (0)	END OF	MMSV		

ICYMMVT: MEDIA MANAGER VECTOR TABLE

Pointed to by: CVTMMVT.

Storage Location: Nucleus.

Description: Contains addresses of media manager routines, a patch area, the first MMSV prefixed by the characters ICYMMSV1, and the first MMPT.

Created: By link-editing media manager into the nucleus.

Released: Not released.

MMVT

Offset Dec. (Hex.)	Length	Name	Description
0 (0)	2	MMVT	MEDIA MANAGER VECTOR TABLE
0 (0)	2	MMVLEN	TOTAL LENGTH OF MMVT
2 (2)	0	MMVCPYRT	COPYRIGHT INFORMATION
2 (2)	1		BLANK
3 (3)	11	MMVCOMP	COMPONENT ID
0 (0)	992	MMVTC	POINTED TO BY CVTMMVT
0 (0)	4	MMVPATCH	M/M PATCH AREA LOCATION
4 (4)	4	MMVIOSB	M/M IOSB SKELETON LOCATION
8 (8)	4	MMVINIT	M/M INITIALIZATION ROUTINE
12 (C)	4	MMVRDWR	M/M READ AND UPDATE WRITE
16 (10)	4	MMVFW	M/M FORMAT WRITE
20 (14)	4	MMVPFMT	M/M PREFORMAT BY RBA RANGE
24 (18)	4	MMVRBA	M/M RBA TO CCHHR CONVERT
28 (1C)	4	MMVCCHHR	M/M CCHHR TO RBA CONVERT
32 (20)	4	MMVMMSRV	MEDIA MANAGER SERVICES
36 (24)	4	MMV553I	ADDRESS OF MODULE IGG0553I

Offset Dec. (Hex.)	Length	Name	Description
40 (28)	4	MMVSM	M/M STORAGE MANAGER
44 (2C)	4	MMVELE	M/M EVENT LISTEN EXIT
48 (30)	4	MMVSORT	M/M REQUEST ELEMENT SORT
52 (34)	4	MMVBLIST	M/M CI LIST BUILD
56 (38)	4	MMVBLDCP	M/M CHANNEL PROGRAM BUILD
60 (3C)	4	MMVPFCP	M/M PREFORMAT CP BUILD
64 (40)	4	MMVDIE	M/M DISABLED INTERRUPT EXIT
68 (44)	4	MMVPFDIE	M/M PREFORMAT DIE
72 (48)	4	MMVNRM	M/M NORMAL END APPENDAGE
76 (4C)	4	MMVABN	M/M ABNORMAL END APPENDAGE
80 (50)	4	MMVPFAPP	M/M PREFORMAT APPENDAGE
84 (54)	4	MMVPGAD	M/M TERMINATION ROUTINE
88 (58)	4	MMVPURG	M/M PURGE ROUTINE
92 (5C)	4	MMVFRR	M/M FUNCTIONAL RECOVERY ROUTINE
96 (60)	4	MMVIEDB	M/M EDB INITIALIZATION
100 (64)	4	MMVILPMB	M/M LPMB INITIALIZATION
104 (68)	4	MMVRSV02	RESERVED
108 (6C)	256	MMVPTCH1	M/M PATCH AREA
364 (16C)	108	MMVIOSB1	M/M IOSB SKELETON
472 (1D8)	8	MMVMSVE	EYE-CATCHER
480 (1E0)	256	MMVMMSV1	M/M FIRST STORAGE VECTOR
736 (2E0)	256	MMVMPT1	M/M FIRST PURGE TABLE

CROSS-REFERENCE FOR MMVT

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
MMVABN	76 (4C)	MMVBLDCP	56 (38)	MMVBLIST	52 (34)
MMVCCHHR	28 (1C)	MMVCOMP	3 (3)	MMVCPYRT	2 (2)
MMVDIE	64 (40)	MMVELE	44 (2C)	MMVFRR	92 (5C)
MMVFW	16 (10)	MMVIEDB	96 (60)	MMVILPMB	100 (64)
MMVINIT	8 (8)	MMVIOSB	4 (4)	MMVIOSB1	364 (16C)
MMVLEN	0 (0)	MMVMPT1	736 (2E0)	MMVMMSRV	32 (20)

**Restricted Materials of IBM
Licensed Materials - Property of IBM**

Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)	Name	Offset/ Equ Value Dec. (Hex.)
MMVMMSVE	472 (1D8)	MMVMMSV1	480 (1E0)	MMVNRM	72 (48)
MMVPATCH	0 (0)	MMVPFAPP	80 (50)	MMVPFCP	60 (3C)
MMVPFDIE	68 (44)	MMVPFMT	20 (14)	MMVPGAD	84 (54)
MMVPTCH1	108 (6C)	MMVPURG	88 (58)	MMVRBA	24 (18)
MMVRDWR	12 (C)	MMVRSV02	104 (68)	MVSM	40 (28)
MMVSORT	48 (30)	MMVT	0 (0)	MMVTC	0 (0)
MMV553I	36 (24)				

APPENDIX. RETURN CODES

For certain errors that may occur when a media manager function is invoked, or a termination routine is called, return codes will be provided.

Return codes are located in register 15 when control is returned to the program that called the media manager; in register zero, when the media manager calls an exit routine.

Return codes will have the format ccccffss; cccc is the return code, ff is the module that detected the error, and ss is a status byte.

The status byte represents the following:

<u>ss field</u>	<u>Error Description</u>
00	No error.
04	Warning.
08	Extent error.
0C	Logic error.
10	Permanent I/O error.
14	Error can't be determined.

In combination with one of these status bytes, return codes in the cccc field indicate the following errors:

<u>cccc ff ss</u>	<u>Error Description</u>
0000 -- 00	No errors.
0000 -- 04	No errors. DIE exit is receiving control in SRB mode.
0010 -- 08	The end of a CI is out of its extent.
0014 -- 08	Extent not found.
0018 -- 08	Extent not active.
0420 -- 08	Overlapping extents.
0424 -- 08	Invalid ending RBA.
0428 -- 08	Block size is too large for the device.
042C -- 08	Inconsistent CI sizes.
0004 -- 0C	Two writes to the same RBA.
0008 -- 0C	Two reads to the same buffer.
0010 -- 0C	MMRE specifies output but MMIB specifies input only.
000C -- 0C	An MMRE specifies multiple CIs to be sorted.
001C -- 0C	Load real address error. Buffers not fixed.
0020 -- 0C	Starting RBA not on CI boundary—preformat only.
0024 -- 0C	Insufficient storage for channel program due to media manager storage constraints.
0400 -- 0C	GETMAIN error for LPMB.
0408 -- 0C	Insufficient storage supplied for LPMB.
0410 -- 0C	GETMAIN error for EDB.
0418 -- 0C	Insufficient storage supplied for EDB.
0434 -- 0C	The parameter list is incomplete.
0010 -- 10	I/O prevented.
0020 -- 10	Permanent I/O error.
0030 -- 14	Error can't be determined.
0034 -- 14	Indeterminate error - I/O has been started.

**Restricted Materials of IBM
Licensed Materials - Property of IBM**

The ff field of the return code identifies the module that has detected the error. These codes may also be found in the MMPMODID field of the MMPB.

<u>ff field</u>	<u>Module</u>
--	ICYMMSRV
--	ICYMMVTC
--	ICYMSTAE
01	ICYRDWR
02	ICYFW
03	ICYPFMT
04	ICYRBA
05	ICYCCHHR
06	ICYSTOR
07	ICYSORT
08	ICYBLIST
09	ICYELE
0A	ICYBLDCP
0B	ICYPFCP
0C	ICYDIE
0D	ICYPFDIE
0E	ICYNRM
0F	ICYABN
10	ICYPFAPP
11	ICYPGAD
12	ICYPURG
13	ICYFRR
40	ICYINIT
41	ICYIEDB
42	ICYILPMB

LIST OF ABBREVIATIONS

ACDS.	alternate control data set.	K.	1024 (a kilobyte=1024 bytes).
APAR.	authorized program analysis report.	LPMB.	logical-to-physical mapping block.
CCHHR.	cylinder-cylinder head-head record.	MMIB.	media manager interface block.
CDS.	control data set.	MMPB.	media manager process block.
CI.	control interval.	MMPT.	media manager purge table.
CSI.	consolidated software inventory.	MMRE.	media manager request element.
CVT.	communication vector table.	MMSB.	media manager storage block.
DASD.	direct access storage device.	MMSV.	media manager storage vector.
DIE.	disabled interrupt exit.	MMVT.	media manager vector table.
EDB.	extent definition block.	RBA.	relative byte address.
ENF.	event notification facility.	RMID.	replace module identifier.
ESQA.	extended system queue area.	SDWA.	system diagnostic work area.
EWS.	early warning system.	SMP.	System Modification Program.
FMID.	function modification identifier.	SMP/E.	System Modification Program Extended.
FRR.	functional recovery routine.	SRB.	service request block.
IOS.	I/O supervisor.	SRM.	system resource manager.
IOSB.	I/O supervisor block.	SSF.	Software Support Facility.
JCL.	job control language.	UCB.	unit control block.
		VSM.	virtual storage manager.

**Restricted Materials of IBM
Licensed Materials - Property of IBM**

INDEX

A

ACB 8
ACDS, defined 34
APAR, defined 34

C

CATREAD 8
CATUPDT 8
CCHHR, defined 34
CDS, defined 34
channel programs 3
CI, defined 34
CNVT function 3, 5
CONNECT 8
control blocks 12
 description 16
 linkages 13
control interval elements 5
CSI, defined 34
CVT, defined 34
COD abend 6

D

DASD, defined 34
data areas 12
 description 16
device architecture 3
device characteristics 6
Diagnosis Reference 1
 how to use 1
 preface 1
DIE, defined 34
DISCONNECT 8

E

EDB 6, 16, 17
EDB, defined 34
ENF 9
ENF, defined 34
ESQA, defined 34
EWS, defined 34
EXTEND 8
extent definition blocks 6

F

FMID, defined 34
FMTWR function 3, 5
FRR 5
FRR, defined 34
function 1
 CNVT 5
 finding modules in 1
 FMTWR 5
 INIT 6
 MMSYSTEM 6
 PFMT 7
 RDWR 7
 SRV 8
functional description 5
functional recovery routine 5

I

ICYABN 6, 9
ICYBLDCP 7, 9
ICYBLIST 5, 7, 9
ICYCCHHR 5, 9
ICYCCHHR. 5
ICYCPL 5
ICYCPL: Convert Parameter List 16
ICYDIE 9
ICYDIE/ICYNRM 6
ICYEDB: Extent Definition Block 16
ICYELE 6, 10
ICYFRR 6, 10
ICYFW 10
ICYIDEVB: Media Manager Initialization
 Device Block 16
ICYIEDB 10
ICYILPMB 10
ICYINIT 6, 10
ICYIPLST: Media Manager Initialization
 Parameter List 16
ICYLPMB: Logical-to-Physical Mapping
 Block 17
ICYMME: Media Manager Interface Block
 Extension 17
ICYMMIB: Media Manager Interface
 Block 17
ICYMMPB: Media Manager Process Block 17
ICYMMPT: Media Manager Purge Table 24
ICYMMRE: Media Manager Request
 Element 24
ICYMMSB: Media Manager Storage Block 25
ICYMMSRV 8, 10
ICYMMSV: Media Manager Storage
 Vector 26
ICYMMVT: Media Manager Vector Table 27
ICYMSTAE 10
ICYNRM 10
ICYPFAPP 7
ICYPFCP 10
ICYPFDIE 7, 10
ICYPGAD 6, 7, 10
ICYPURG 6, 10
ICYRBA 5, 10

ICYRDWR 11
ICYSORT 11
ICYSTOR 5, 7, 11
INIT function 3, 6
IOS 9
IOS, defined 34
IOSB 17
IOSB, defined 34
IOSUSE 17

J

JCL, defined 34

K

K, defined 34

L

linkages, control block 13
logical-to-physical mapping blocks 6
LPMB 6, 17
LPMB, defined 34

M

media manager 3
 control blocks 12
 data areas 12
 module directory 9
 overview 3
 service aids 30
MMIB 16
MMIB, defined 34
MMPB 5
MMPB, defined 34
MMPMODID 17
MMPT, defined 34
MMRE, defined 34
MMSB, defined 34
MMSV, defined 34
MMSYSTEM function 3, 6
MMVT, defined 34
module directory 9
modules 2
 finding the function 2
MSWA 8

P

PFMT function 3, 7
PURGE HALT 6

R

RBA, defined 34
RDWR function 3, 7
relative byte addresses 3, 12
RMID, defined 34

S

SDWA, defined 34
service aids 30
SMP/E, defined 34
SMP, defined 34
SRB, defined 34
SRM 9
SRM, defined 34
SRV function 3, 8
SSF, defined 34
STARTIO 7
SVC 26 8
SYS1.DUMPnn 6
SYS1.LOGREC 6

U

UCB, defined 34

V

VSAM 9
VSM 9
VSM, defined 34

W

work area 8

**Restricted Materials of IBM
Licensed Materials—Property of IBM**
(Except for Customer-Originated Materials)
© Copyright IBM Corp. 1980, 1987
LY26-3965-1

**Reader's
Comment
Form**

MVS/XA Media Manager Diagnosis Reference

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Do not use this form to request IBM publications. If you do, your order will be delayed because publications are not stocked at the address printed on the reverse side. Instead, you should direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

If you have applied any technical newsletters (TNLs) to this book, please list them here: _____

Chapter/Section _____

_____ Page No. _____

Comments:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

If you want a reply, please complete the following information.

Name _____ Phone No. (____) _____

Company _____

Address _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and tape

Please do not staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150



Fold and tape

Please do not staple

Fold and tape





MVS/Extended Architecture
Media Manager
Diagnosis Reference

Restricted Materials of IBM
Licensed Materials—Property of IBM
© Copyright IBM Corp. 1980, 1987
File Number S370-37

LY26-3965-01



Printed in U.S.A.