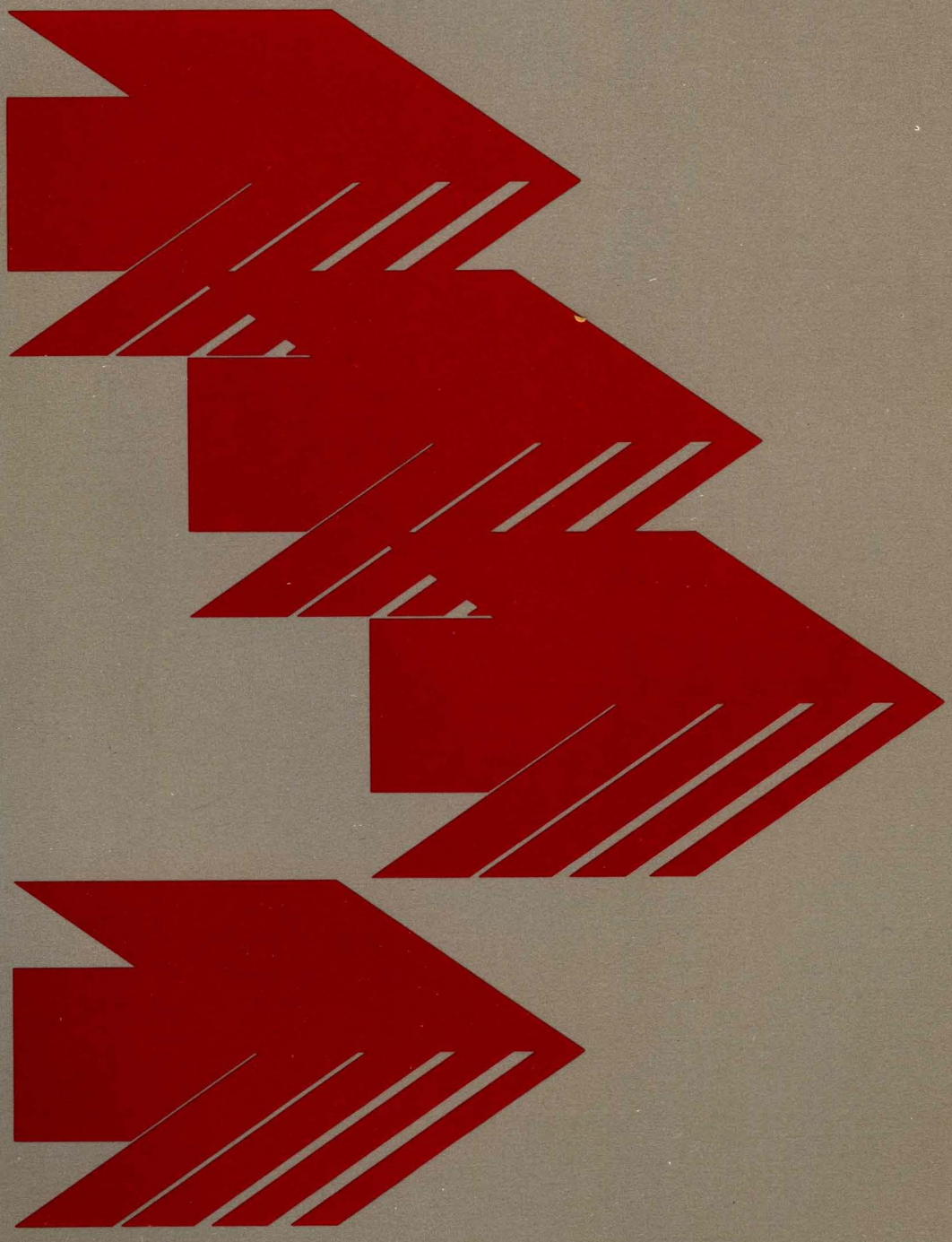
IBM 3480 Magnetic Tape Subsystem User's Reference

Storage Subsystem Library

GC35-0099-3



IBM 3480 Magnetic Tape Subsystem User's Reference

Storage Subsystem Library

GC35-0099-3

| **Fourth Edition (November 1987)**

| This is a revision of and makes obsolete GC35-0099-2.

Changes or additions to the text or illustrations will be indicated by a vertical line to the left of the change.

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 licensed program may be used. Any functionally equivalent program may be used instead.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, General Products Division, Department 61C, Tucson, Arizona, U.S.A. 85744. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This manual gives programmers information about using the IBM 3480 Magnetic Tape Subsystem. For complete information about specific macros, commands, or functions, consult the other manuals listed in "Related Publications."

Organization

This manual contains the following:

- "Chapter 1. Introduction," gives an overview of the IBM 3480 Magnetic Tape Subsystem and the software requirements for the 3480
- "Chapter 2. Tape Characteristics," describes the format of an IBM standard data set label
- "Chapter 3. Modifying Existing Code," describes the changes to code that are needed to use the 3480 including DCB parameters, JCL changes, and modifications to JFCB
- "Chapter 4. Controlling the I/O Device," describes the changes to the NOTE, POINT, and DEVTYPE macros, as well as the new SYNCDEV macro
- "Chapter 5. Displaying Messages," describes the new MSGDISP macro that allows messages to be displayed to the operator
- "Chapter 6. Managing and Controlling Data Sets," describes the changes for DFHSM when using the 3480
- "Chapter 7. Recording Error or Statistical Information," describes how to use the 3480 for the IMS/VS log tape and error and statistical information
- "Chapter 8. Restrictions to Using Sorting Applications," describes the restrictions to DFSORT when using the 3480
- "Appendix A. Data Control Block Symbolic Field Names," describes the control block fields
- "Appendix B. Checkpoint/Restart Codes," shows the return and completion codes for the IBM 3480 Magnetic Tape Subsystem when taking a checkpoint.

Prerequisite Knowledge

To use this book effectively, you should already understand data management procedures.

Related Publications

You may want to reference information presented in the publications listed below.

IBM 3480 Magnetic Tape Subsystem

- *IBM 3480 Magnetic Tape Subsystem Introduction*, GA32-0041
- *IBM 3480 Magnetic Tape Subsystem Planning and Migration Guide*, GC35-0098
- *IBM 3480 Magnetic Tape Subsystem Reference: Channel Commands, Status and Sense Bytes, and Error Recovery Procedures*, GA32-0042
- *IBM 3480 Magnetic Tape Subsystem Operator's Guide*, GA32-0066
- *Care and Handling of the IBM Magnetic Tape Cartridge*, GA32-0047
- *IBM Input/Output Equipment: Installation Manual—Physical Planning for System/360, System/370, and 4300 Processors*, TNL GN22-2317 to GC22-7064
- *Tape and Cartridge Requirements for the IBM 3480 Tape Drives*, GA32-0048.

MVS/Extended Architecture

- *MVS/Extended Architecture Checkpoint/Restart User's Guide*, GC26-4139
- *MVS/Extended Architecture System-Data Administration*, GC26-4149
- *MVS/Extended Architecture Data Administration Guide*, GC26-4140
- *MVS/Extended Architecture Data Administration: Macro Instruction Reference*, GC26-4141
- *MVS/Extended Architecture Magnetic Tape Labels and File Structure Administration*, GC26-4145
- *MVS/Extended Architecture Message Library: System Messages*, Volumes 1 and 2, GC28-1376 and GC28-1377
- *MVS/Extended Architecture Message Library: System Codes*, GC28-1157

- *MVS/Extended Architecture, MVS/System Product—JES3 Version 2, MVS/System Product—JES2 Version 2, LIC PROG 5665-291, 5740-XC6, Routing and Descriptor Codes*, GC28-1194

- *MVS/Extended Architecture Installation: System Generation*, GC26-4148

- *MVS/Extended Architecture Data Facility Product: General Information*, GC26-4142

- *MVS/Extended Architecture Conversion Notebook: MVS/System Product JES2, LIC PROG 5740-XC6; MVS/System Product JES3, LIC PROG 5665-291*, GC28-1143

- *MVS/Extended Architecture, MVS/System Product—JES3 Version 2, MVS/System Product—JES2 Version 2, LIC PROG 5665-291, 5740-XC6, Job Control Language*, GC28-1148

- *MVS/Extended Architecture MVS/System Product Version 2 5665-291 5740-XC6, Initialization and Tuning*, GC28-1149

- *MVS/Extended Architecture, MVS/System Product Version 2—JES3 MVS/System Product Version 2—JES2, LIC PROG 5665-291, 5740-XC6, Supervisor Services and Macro Instructions*, GC28-1154

- *MVS/Extended Architecture MVS/System Product—JES3 Version 2, MVS/System Product—JES2 Version 2, LIC PROG 5665-291, 5740-XC6, System Macros and Facilities*, Volumes 1 and 2, GC28-1150 and GC28-1151

- *MVS/Extended Architecture, LIC PROG 5665-291, 5740-XC6, System Modifications*, GC28-1152

- *MVS/Extended Architecture, MVS/System Product—JES3 Version 2, MVS/System Product—JES2 Version 2, 5665-291, 5740-XC6, System Management Facility*, GC28-1153

- *MVS/Extended Architecture, MVS/System Product—JES2 Version 2, 5740-XC6 Operations: JES2 Commands*, SC23-0064

- *MVS/Extended Architecture, MVS/System Product JES2 Version 2, LIC PROG 5740-XC6, Initialization and Tuning*, SC23-0065

- *MVS/Extended Architecture, MVS/System Product—JES3 Version 2, LIC PROG 5665-291, Operations: JES3 Commands*, SC23-0063

- *MVS/Extended Architecture System, MVS/System Product—JES3 Version 2, LIC PROG 5665-291, Initialization and Tuning*, SC23-0059

- *MVS/Extended Architecture Debugging, MVS/System Product—JES3 Version 2, MVS/System Product—JES2 Version 2, 5665-291, 5740-XC6,*

Handbook, Volumes 1 through 5, LC28-1164,[1] LC28-1165, LC28-1166, LC28-1167 and LC28-1168.

MVS/370

- *MVS/370 Data Facility Product, 5665-295, Checkpoint/Restart*, GC26-4054
- *MVS/370 Data Facility Product, 5665-295, System Generation Reference*, GC26-4063
- *MVS/370 Data Facility Product, 5665-295, Data Management Macro Instructions*, GC26-4057
- *MVS/370 Data Management Services, Data Facility Product 5665-295*, GC26-4058
- *MVS/370 Data Facility Product, 5665-295, Data Management*, GC26-4056
- *MVS/370 Magnetic Tape Labels and File Structure, 5665-295*, GC26-4064
- *MVS/370 Message Library: System Messages*, Volumes 1 and 2, GC28-1374 and GC28-1375
- *OS/VS Message Library: System Codes*, GC38-1008
- *OS/VS Message Library: VS2 Routing and Descriptor Codes*, GC38-1102
- *MVS/370 Installation: System Generation*, GC26-4166
- *MVS JCL MVS/System Product—JES2 Release 3.3, MVS/System Product—JES3 Release 3.1, LIC PROG 5740-XYS, -XYN*, GC28-1300
- *MVS/370 Data Facility Product: General Information*, GC26-4160
- *OS/VS2 System Programming Library: Debugging Handbook, MVS/System Products, LIC PROG 5740-XYN, -XYS, Volumes 1 through 3*, GC28-1047, GC28-1048, and GC28-1049
- *OS/VS2 MVS Supervisor Services and Macro Instructions*, GC28-0683
- *OS/VS2 MVS System Programming Library: Initialization and Tuning Guide*, GC28-0681
- *OS/VS2 MVS System Programming Library: Supervisor*, GC28-0628
- *OS/VS2 MVS System Programming Library: System Management Facilities*, GC28-1030.

---

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

VM/SP and VM/HPO

- *Virtual Machine/System Product Planning Guide and Reference*, SC19-6201
- *Virtual Machine/System Product High Performance Option Planning Guide and Reference*, SC19-6223.

VSE/SP

- *IBM Virtual Storage Extended/System Package Hardware and Software Support Extensions, Version 2 Release 1*, SC33-6184.

Other Related Publications

- *OS/VS Sort/Merge Programmer's Guide Program Product 5740-SM1*, SC33-4035
- *Data Facility Hierarchical Storage Manager: Version 2 Release 3.0 Installation and Customization Guide*, SH35-0084
- *Data Facility Hierarchical Storage Manager: Version 2 Release 3.0 System Programmer's Guide*, SH35-0085
- *Data Facility Hierarchical Storage Manager: Version 2 Release 3.0 System Programmer's Reference Guide*, SH35-0083
- *OS/VS2 MVS and Stand-Alone Versions: Input/Output Configuration Program, LIC PROG 5740-XYN, -XYS, User's Guide and Reference*, GC28-1027
- *Environmental Recording Editing and Printing (EREP) Program*, GC28-1178
- *Data Facility/Data Set Services, LIC PROG 5740-UT3, User's Guide and Reference*, SC26-3949
- *IBM Input/Output Equipment: Installation Manual—Physical Planning for System/360, System/370, and 4300 Processors*, GC22-7064
- *IBM System/370 Installation Manual: Physical Planning*, GC22-7004.

Contents

Figures

Summary of Amendments

IBM 3480 Magnetic Tape Subsystem User's Reference

Release GC35-0099-3, July 1987

This edition includes changes to correct errors or omissions in the previous edition.

Release GC35-0099-2, January 1987

This edition adds information on the 3480 Model A11 Control Unit and Model B11 Tape Drive.

There are also miscellaneous changes to correct errors or omissions in the previous edition.

Release GC35-0099-1, July 1986

This edition incorporates Technical Newsletter number GN35-0319 and adds information on the Automatic Cartridge Loader Feature.

Chapter 1. The IBM 3480 Magnetic Tape Subsystem

The IBM 3480 Magnetic Tape Subsystem consists of a 3480 Control Unit, Model A11 or A22, and up to four 3480 Tape Units, Model B11 or B22. Each tape unit contains two magnetic tape drives. With a Dual Control Unit Communications Coupler feature, two control units can be connected to allow mutual control of eight tape units (16 drives). The Automatic Cartridge Loader feature, which allows both the automatic loading of premounted tape cartridges and the manual loading of single tape cartridges, can be added to any or all of the tape units. With additional channel attachment features, each control unit can connect with up to four channels.

The subsystem can read and write data at rates up to three megabytes per second. This high data rate is achieved by recording 18 data tracks at a linear data recording density of about 38000 bytes per inch.

The control unit contains a buffer that stores the data being transferred between the tape drive and the controlling processor. The buffer reduces delays in the operations of the processor that are normally caused by the start and stop actions of individual drives. Because of the buffer, blocks of data to be written are transferred to the control unit before the tape drive is ready to accept them. The channel is then released to do other work while the data transfers from the control unit to the tape drive. During reading, data is transferred from the tape drive to the control unit buffer so as to be available when the controlling processor requests it.

Because the 3480 subsystem indicates that the writing action is complete as soon as the write data has been transferred to the buffer, the subsystem also provides a tape-write-immediate mode of writing data to tape. Tape-write-immediate mode forces the subsystem to write the data on the tape before it indicates that the action is complete. Tape-write-immediate mode should be used only when the data is critical because this mode reduces subsystem performance.

The 3480 subsystem provides a high-speed search function. With high-speed search, you can search for specific blocks on the tape (with the subsystem disconnected from the channel) at much higher speed than is used for reading and writing. When the block is found, the subsystem reconnects to the channel to signal that it is ready for the next command.

Each tape drive has a message display on its operator's panel. The messages on this display can keep the operator informed of the status of the tape drive. In an MVS system, the program can write any necessary messages to the drive message display.

The tape drives can be assigned to specific channels by program action. This assignment ability replaces the mechanical switching used for channel assignment for earlier tape drives. For more information on channel assignment, see *IBM 3480 Magnetic Tape Subsystem Planning and Migration Guide.*

IBM 3480 Magnetic Tape Subsystem Operating Mode

In MVS systems, the 3480 subsystem can operate as a 3480 in full function mode, or it can operate in 3420 compatibility mode. System generation controls the mode in which the 3480 operates. **Both the 3480 in full function mode and the 3480 in 3420 compatibility mode cannot both be generated at the same time on the same system.** If you are generating a 3480 in full function mode, you cannot generate compatibility mode for any other 3480s on the system. However, catalog entries for both the 3480 and the 3420C are synonymous, and both entries can exist in the same catalog. This allows you to share catalogs among systems or migrate to a full function 3480 without changing catalog entries. Each system uses the catalog entry appropriate to the 3480 device type that was generated on that system. VM and VSE/SP support for the 3480 is comparable to that provided in MVS compatibility mode.

When the 3480 is operating in compatibility mode, the following 3480 features are supported:

- Tape write-immediate mode (OPTCD = W)
- Full error recovery (ERP) support
- Buffered read and write mode
- Hardware-initiated message display.

When the 3480 is operating in compatibility mode, the following 3480 features are **not** supported:

- High-speed search (block locate)
- Program-initiated message display (MSGDISP macro)
- Dynamic device reconfiguration (DDR)
- Processor assignment
- User-requested synchronization
- Automatic cartridge loader in System mode.

For more information on the functions of the 3480 in full function or compatibility mode, see *IBM 3480 Magnetic Tape Subsystem Introduction* and *IBM 3480 Magnetic Tape Subsystem Planning and Migration Guide.*

Conventions Used

The following list of symbols are used to define the format of the macros used for the 3480, but you should never type the symbols in the actual macro statement.

- Braces ({ }) indicate a required field or parameter. When a group of items appears within braces, you must choose one of the items.
- Brackets ([ ]) indicate an optional field or parameter.
- Items separated by a vertical bar (|) represent alternative items. Unless otherwise stated, no more than one of the alternative items can be selected.
- An ellipsis (...) indicates that multiple entries of the type immediately preceding the ellipsis are allowed.
- Other punctuation (parentheses, commas, slashes, spaces, and so forth) must be entered as shown.
- **Boldface** type indicates the exact characters to be entered. Such items must be entered exactly as shown.
- *Italic* type indicates fields to be supplied by you.
- Underscored type indicates a default option. If the parameter is omitted, the default is assumed.

Chapter 2. Tape Characteristics

The 3480 subsystem uses compact cartridges containing 1/2-inch (12.65-mm) chromium dioxide tape. The cartridge is about 25% the size of a standard 2400-foot (731-meter) reel of magnetic tape, but it stores up to 20% more data than the reel. Only a single density is available for tape cartridges and is used by the system for reading and writing.

Changed Format for the IBM Standard Data-Set Label 2 (HDR2/EOF2/EOV2)

The following fields contain changed data for the IBM standard file label 2:

Field 6, Tape Density (1 byte)

- Contents: A code indicates the recording density of the tape. The code is equivalent to the DEN parameter value on the data definition (DD) statement. Use code = 0 for the 3480.
- Processing: The field is neither used nor verified. When data management creates labels, the information for this field is obtained from the JFCB.

Field 13, Reserved (8 bytes)

- Contents: For 2400 and 3410 tape drives (for MVS/370 only), the entire field is reserved for possible future use. For 3420 drives, bytes 40 through 42 are reserved, byte 43 contains the model number, and bytes 44 through 47 contain the last four digits of the serial number of the creating tape unit. For the 3480 subsystem, bytes 40 through 42 are reserved, bytes 43 through 46 contain the last four digits of the serial number of the control unit, and byte 47 contains the device address (for MVS/370) or the device number (for MVS/XA).

  The serial numbers in the header and trailer labels may not be the same if the data set is opened for update or if DDR is used to swap tape units while the data set is being created.
- Processing: A unique number identifying the recording unit is read from the tape during open processing, converted into hexadecimal, and inserted into the UCBCTD field in the UCB tape extension.

Recording Mode for the IBM Standard Data Set Label 2 (HDR2/EOF2/EOV2)

If the Improved Data Recording Capability feature is installed, enabled, and invoked on the subsystem, the existing two-byte field, which is defined as the tape recording technique (previously used for seven-track tapes), indicates the data-set attribute for the IBM standard label file 2. This field is FL2TRTCH and defined in the IECDSECT macro:

```
'P ' (P followed by a blank)   The file is written in compacted mode.
'  ' (two blanks)              The file is not written in compacted mode.
```

Chapter 3. Modifying Existing Code in MVS Systems

Introducing the 3480 subsystem into your operations can require some modifications to your existing code. Such areas as DCBs, DD statements, JCL, and JFCBs should be examined for needed changes.

Selecting Data Set Options

For each data set you want to process, there must be a corresponding DCB and DD statement. The characteristics of the data set and device-dependent information can be supplied by either source. Also, the DD statement must supply data set identification, device characteristics, space allocation requests, and related information as specified in the appropriate JES documentation. You establish the logical connection between a DCB and a DD statement by specifying the name of the DD statement in the DDNAME field of the DCB macro, or by completing the field yourself before opening the data set.

DCB Parameters

After you have specified the data set characteristics in the DCB macro, you can change them only by changing the DCB during processing. The fields of the DCB discussed below are common to most data organizations and access techniques. (For more information about the DCB fields, see appropriate data management/administration macro instruction reference.)

Note: When using the IBM 3480, you need to consider the effect of the OPTCD=W and the DEN options.

Density Option (DEN)

The tape density parameter (DEN) specifies the recording density in bits-per-inch-per-track. When DEN is not specified, the highest density capable by the unit is used.

Note: The 3480 does not need the DEN option. If UNIT=3480, the DEN option is ignored. In compatibility mode, if DEN is specified, you can specify only DEN=4.

Write-Validity-Check Option (OPTCD=W)

You can specify the write-validity-check option in either the DCB parameter of the DD statement or the DCB macro. OPTCD=W causes the 3480 to operate in tape-write-immediate mode.

For buffered tape devices, write validity check delays the device-end interrupt until the data is physically on tape. When you use the write validity check option, you do not benefit from the performance advantage of buffering. This causes the maximum I/O rate to be reduced to about 10 blocks per second.

The OPTCD=W parameter of the DCB can be used to request that data buffering in the control unit be bypassed when writing data. It ensures that each block of data is physically written on the tape and read-back checked, before the control unit gives write-completion indication to the processor. (The read-back check is done by the hardware while the data is being written).

Generally, you use the write-validity-check option to make the application's write operations go directly to the tape instead of storing the data in the control unit buffer. OPTCD=W is generally only meaningful with BSAM and EXCP.

Coding the Block Size and Logical Record Length Operands

The 3480 provides good performance for all block sizes, but is designed to provide *optimal* performance for block sizes in the range of 8 to 32 kilobytes.

Constructing a Data Control Block

The access method support that exists for data sets on magnetic tape is used for the 3480. QSAM, BSAM, and EXCP are supported. The default mode used is buffered data transfer for both read and write.

Macro Instructions Available by Access Method

MACRO INSTRUCTION	EXCP	BSAM	QSAM
DCB	X	X	X
DCBD	X	X	X
MSGDISP	X	X	X
NOTE		X	
POINT		X	
SYNCDEV	X	X	X

Specifying JCL Options

The W option for OPTCD requests a validity check for write operations on the 3480. For a discussion of the OPTCD=W option, see "Write-Validity-Check Option (OPTCD=W)" on page 8.

Use the UNIT parameter to specify:

- The specific device you want the system to assign to a data set
- The types of devices you want the system to assign to a data set
- The number of devices you want the system to assign to a data set
- That the system is to assign data sets to the same unit another data set uses (unit affinity)
- That the system is not to request the operator to mount the volume until the data set is opened (deferred volume mounting).

If you request the 3480 subsystem in compatibility mode, code UNIT=3400-9 or the proper esoteric name. If you request the 3480 subsystem in full function mode, code UNIT=3480 or the proper esoteric name.

For further information on the use of the UNIT parameter, see *MVS/Extended Architecture, MVS/System Product—JES3 Version 2, MVS/System Product—JES2 Version 2, LIC PROG 5665-291, 5740-XC6, Job Control Language* or *MVS JCL MVS/System Product—JES2 Release 3.3, MVS/System Product—JES3 Release 3.1, LIC PROG 5740-XYS, -XYN.*

Modifying the Job File Control Block (JFCB)

The OPEN macro instruction requires access to information that you have supplied in a data definition (DD) statement. This information is stored by the system in a job file control block (JFCB).

In certain applications, you might have to modify the contents of a JFCB before issuing an OPEN macro instruction. For example, you might want to specify the physical block identifier in the JFCB (explicitly or by default) to permit a high-speed search of a specific data block on a tape volume. For further information about the JFCB, see *MVS/370 System Programming Library: Data Management* or *MVS/Extended Architecture System-Data Administration.*

Setting the Header Label Block ID in the JFCB for High-Speed Search

To use the high-speed search function of the 3480 subsystem on a standard label tape data set, you must specify the block ID for the data by modifying certain fields in the JFCB and using OPEN TYPE=J to open the data set. The ability to use the high-speed search function is supported only by 3480 software support.

The following procedure should be used to modify the JFCB for a high-speed search:

1. Use the RDJFCB macro to read the JFCB into your work area.

2. Set the JFCPOSID bit in field JFCBFLG3 to indicate that a block ID is provided.

3. Move the block ID of the first standard header label of the tape to the field JFCRBIDO. (The block ID must have been saved earlier, for example, from the ID available at close time.)

4. Use the OPEN macro with TYPE=J to open the data set.

If the JFCPOSID bit is set off, OPEN moves the tape normally, as though the high-speed search function is not active.

If the fast positioning bit is set on but no block ID is found in JFCRBIDO, OPEN moves the tape normally and inserts the block ID of the first label header record into JFCRBIDO.

If the JFCPOSID bit is set on but the block ID in JFCRBIDO does not match the block ID on the tape, OPEN processing fails and error message IEC147I indicates that the block ID could not be found on the tape.

If the block ID specified does not exist, the control unit searches for the block or file mark preceding the one that the Locate Block command specified. The drive moves the tape to a position to write data following the previous block or file mark.

During CLOSE processing, if the JFCPOSID bit is on, the block ID for the first label header record of the **next** data set (which may not exist) is inserted into the JFCRBIDO field. Therefore, if you deallocate the 3480 device and plan to use the current block ID for subsequent processing, you should save the block ID before you close the data set.

Note that if dynamic unallocation is specified (with SVC 99, with FREE=CLOSE in the DD statement, or with the FREE option in the CLOSE macro), the block ID for the next data set is not available to your program. This is because dynamic unallocation frees the JFCB.

Note: The data set sequence number (as coded in LABEL=(seqno,SL)) must be specified as usual (either explicitly or by default) when you move the tape at high-speed.

Example of How to Code OPEN(J)

The example program, following, shows how to add another data set to an existing multifile volume. The last time this program was processed (after the predecessor file was created), the file number and the BLOCKID of the file to be added are stored in a small control data set on DASD.

The example program opens the control data set and retrieves the file sequence number and the BLOCKID of the file to be created. File number and BLOCKID are stored into the JFCB of the new tape file. (see Figure 1.)

The subsequent OPEN TYPE=J performs a high-speed search to the beginning of the new tape file. The new file is created and closed. After CLOSE, the BLOCKID of the next file (the one to be created next time) is picked from the JFCB, the file sequence number is increased, and both are stored back into the control data set.

The fields in the JFCB that are referred to in the example program are:

Offset Dec.	Name	Length	Comment
+56	JFCRBIDO	4	Physical block identifier of SL HDR1 to be processed by Open.
+108	JFCRBIDC	4	Physical block identifier of final tape mark. The same as the first SL HDR1 of the next data set on the tape volume.
+163	JFCBFLG3	1	New bit defined for the 3480.
	JFCPOSID	-	Bit 3 (hex 10) indicates request for physical block identifier processing in Open/Close/EOV.

Figure 1. JFCB Layout

```
         ...
         ...
CDSDCB   DCB     DDNAME=CDSDD,DSORG=PS,MACRF=(GM,PM))
TAPEDCB  DCB     DDNAME=TAPEDD,DSORG=PS,MACRF=(PM),EXLST=JFCBP
         ...
         DS      0F
JFCBP    DC      X'87',AL3(JFCB)        OPEN J INDICATOR AND JFCB ADDR
JFCB     DS      CL176                  JFCB
         ORG     JFCB
         IEFJFCBN
         ORG     ,
*
CDSBUF   DS      0CL80                  CONTROL DATA SET RECORD
CDSFLSQ  DS      CL2                    SEQUENCE NUMBER OF NEW FILE
CDSRBID  DS      CL4                    BLOCKID OF BEGINNING OF FILE
         ORG     ,
         ...
         ...
         OPEN    CDSDCB                 OPEN CONTROL DATA SET
         GET     CDSDCB,CDSBUF          READ CONTROL DATA SET
         CLOSE   CDSDCB                 CLOSE CONTROL DATA SET
         ...
         ...
         RDJFCB  TAPEDCB                GET JFCB OF TAPE FILE
         MVC     JFCBFLSQ,CDSFLSQ       PUT FILE SEQ. NO. INTO JFCB
         MVC     JFCRBIDO,CDSRBID       PUT BLOCKID INTO JFCB
         OI      JFCBFLG3,JFCPOSID      SET BLOCKID BIT ON
         OPEN    (TAPEDCB,(OUTPUT)),TYPE=J OPEN/ POSITION HIGH SPEED
LP1      ...
         ...
         PUT     TAPEDCB,....           CREATE NEW FILE
         B       LP1
         ...
EOF      CLOSE   TAPEDCB
         ...
         ...
         RDJFCB  TAPEDCB                GET UPDATED JFCB OF TAPEDD
         LH      R1,CDSFLSQ             GET FILE SEQUENCE NUMBER
         AH      R1,=H'1'               INCREMENT BY 1
         STH     R1,CDSFLSQ             STORE SEQ. NO. OF NEXT FILE
         MVC     CDSRBID,JFCRBIDC       STORE BLOCKID OF NEXT FILE
         OPEN    (CDSDCB,(OUTPUT))      OPEN CONTROL DATA SET
         PUT     CDSDCB,CDSBUF          UPDATE CONTROL DATA SET
         CLOSE   CDSDCB                 CLOSE CONTROL DATA SET
         ...
         ...
```

MVS Block Count Checking

When using MVS to read any type of magnetic tape volume that has IBM standard or ANSI tape labels, certain block count checking is done. The block count checking that is performed for all magnetic tape devices is described in *MVS/370 Magnetic Tape Labels and File Structure*, and *MVS/Extended Architecture Magnetic Tape Labels and File Structure Administration*, DFP Version 1, and DFP Version 2.

For the 3480, MVS does block count checking under more circumstances. These are when the data set is written and for all label types. If there is a block count error on the tape, the system will attempt to discover it when the tape is written, rather than when it is later read.

This additional block count checking is possible because the 3480 appends a block identifier to each block written on the tape and provides the block identifier to the system using the Read Block ID channel command.

For 3480, MVS compares the number of blocks written or read in a data set to DCBBLKCT when it is being maintained. The comparison is done during CLOSE, EOV and FEOV operations and results in an ABEND when the DCBBLKCT and block ID data do not match. This comparison is not done for an MVS checkpoint data set or one for which OPTCD=H (imbedded DOS checkpoints) is specified.

With BSAM and QSAM, the system maintains DCBBLKCT during tape processing. For EXCP and EXCPVR, DCBBLKCT may also be maintained.

With standard labels (IBM and ANSI) the block count is recorded in the trailer labels. The installation's NSL (nonstandard label) routines may also record the block count in the trailer labels.

For tape devices other than 3480, output data sets cannot be checked for the correct number of blocks.

The block count check that compares the DCBBLKCT field to the trailer label is performed only on input data sets with standard labels. Thus, two block count checks are performed for 3480 input data sets with standard labels.

The additional 3480 block count checking provides some additional return codes for ABEND messages and may require you to change some of your programs. The additional ABEND return codes are explained in "ABEND Return Codes" on page 73. Programming changes that may be required include user exits for the additional block count check, EXCP routines, and certain user programs.

User Exits

For those data sets that receive both block count checks, the check using the 3480-written block ID is performed before the check that compares the trailer label to DCBBLKCT. If both checks fail and you want to prevent both ABENDs, you must handle each ABEND independently. You can prevent the 3480 block count check ABEND in the DCB ABEND exit and can prevent the DCBBLKCT to trailer label mismatch ABEND in either the DCB ABEND exit or in the block count unequal exit.

EXCP Routines

DCBBLKCT is assumed to be present and valid in all cases except for an EXCP DCB that does not have a device interface section of at least three words. For EXCP routines this means that any of bits 12, 13, or 14 of DCBMACRF must be set to 1. Bit 5 of MACRF must also be 1 to indicate that the routine is maintaining an accurate block count.

For further information concerning the use of EXCP and EXCPVR, see *MVS/370 System Programming Library: Data Management*, and *MVS/Extended Architecture System-Data Administration*, DFP Version 1, and DFP Version 2.

User Programs

A few user programs written for an environment where the system does less block count checking may not operate successfully. Programs that may fail are:

- Programs that do not cause the DCBBLKCT field to reflect the correct tape position when one of the following is true: the tape is unlabeled, the tape has non-standard labels, or BLP is used.
- Programs that depend on the block count exit to handle known block count discrepancies. The mismatch between DCBBLKCT and the count calculated from the unit causes an ABEND to be issued before the mismatch between DCBBLKCT and the tape label would cause the block count exit to be called.

The following are changes that you can make to such programs:

- Keep an accurate block count in DCBBLKCT. For BSAM and QSAM, the access method takes care of this except with some uses of the CNTRL macro. With EXCP, the system will maintain DCBBLKCT if IOBINCAM specifies the number of blocks read or written with each EXCP. If IOBINCAM is zero, your program can maintain DCBBLKCT. A positive or negative value indicates the number of blocks going forward or backward, respectively, on the tape.
- Set all of bits 12, 13, and 14 of DCBMACRF to zero and/or set bit 5 of DCBMACRF to zero. This may cause the program to be less reliable because the block count is not checked.
- Supply a DCB ABEND exit routine (see Data Administration Guide) to handle the 117, 137, and 214 ABEND.
- Supply an installation DCB ABEND exit routine (see Data Administration Guide) to handle the 117, 137, and 214 ABENDs.
- Supply an ESTAE exit routine to handle the new ABENDs. This action is least desirable because the DCB becomes unusable and subsequent volumes cannot easily be handled.

Performance Options for IEBGENER

BUFNO is a supported parameter for DCB specifications for IEBGENER. IEBGENER provides better performance when multiple buffers permit more data blocks to be read and written with each I/O operation initiated to a device, and allows the potential for more overlap of reading with writing. IEBGENER provides a default value of 5 for the number of buffers.

When one or both of the data sets are on DASD and has a small block size, the performance of IEBGENER may be further improved by increasing the number of buffers above the default value, up to one more than the maximum number of blocks per track.

IEBGENER uses the DCB parameter NCP=nn to determine the number of buffers if the BUFNO parameter is not specified. If the parameter used for BSAM specifies a number greater than 99, the value 99 is used.

If INPUT data sets are concatenated, BUFNO is ignored and one buffer is used.

IEBGENER Storage Requirements

Using multiple buffers for IEBGENER increases the amount of virtual storage needed to run the program. You may need to change or add a REGION parameter for the additional storage to avoid 80A system ABENDs. You can calculate the region size by using the following formula:

```
50K + ((2+SYSUT1 DCBBUFNO)*(SYSUT1 DCBBLKSIZE)) +
      ((2+SYSUT2 DCBBUFNO)*(SYSUT2 DCBBLKSIZE))
```

Note: If BUFNO is not specified, use a value of 5 in the calculation. This formula yields slightly large values for region size to allow for growth and different MVS environments.

The following is an example of how to use the formula to calculate the amount of virtual storage needed:

```
SYSUT1 BLKSIZE=2K
SYSUT1 BUFNO=20
SYSUT2 BLKSIZE=32K
SYSUT2 BUFNO=Not specified, default is used

 Virtual region = 50K + ((2+20)*(2K)) +
                  ((2+5)*(32K))

REGION=318K
```

INVOKING IEBGENER

The following is an example of JCL to invoke IEBGENER:

```
//COPYJOB  JOB
//         EXEC PGM=IEBGENER,REGION=318K
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT1   DD DSNAME=X.FILE,UNIT=3380,
//            DISP=OLD,VOL=SER=X13380,
//            DCB=(BUFNO=20,RECFM=FB,LRECL=2000,BLKSIZE=2000)
//SYSUT2   DD DSNAME=X.FILE.TAPEVER,UNIT=3480,
//            DISP=(NEW,KEEP),
//            DCB=(RECFM=FB,LRECL=2000,BLKSIZE=32000)
```

Chapter 4. Controlling the I/O Device

In 3480 full-function mode, two modified macros allow you to perform high-speed searches for data blocks on tape. A new macro allows you to force the control unit to write all the buffered write data onto the tape before continuing with other commands.

Controlling High-Speed Search (NOTE/POINT)

The NOTE and POINT macros are modified for the 3480 and parameter TYPE=ABS is added for full-function mode.

How the Physical Block Identifier Works

Before using the NOTE and POINT macros for high-speed search, you should understand how data is stored on the 3480 tape.

The 3480 control unit records a block identifier (BLOCKID) with each block written to tape. A BLOCKID contains a 1-byte physical reference value and a 3-byte absolute block number. The block identifier is increased by 1 for every successive block on the tape, including labels and tape marks.

The absolute block number for the first block on tape is hex 000000.

Although a BLOCKID is normally recorded on tape, it is invisible to the Read command. Read operations do not transfer the BLOCKID into main storage with the record. The program does not have to create BLOCKIDs when writing data.

The NOTE macro retrieves the BLOCKID. The POINT macro uses a known BLOCKID to perform a high-speed search for a specific block. The new TYPE=ABS keyword in the NOTE and POINT macros cause these functions.

Retrieving the BLOCKID

The NOTE macro returns two BLOCKID values, the current block position and depth of buffering, to the application program.

The first BLOCKID returned (in register 0) by NOTE represents one of the following:

- The data block to be passed between the processor and the subsystem in either read or write mode
- The last data block sent to the processor in read backward mode.

The second BLOCKID returned (in register 1) by NOTE represents one of the following:

- The next data block to be written to the tape from the control unit buffer for write mode
- The next data block to be read from the tape to the control unit buffer for read mode
- The most recent data block that was read from the tape to the control unit buffer in read-backward mode.

To obtain the number of blocks in the buffer in use for the specific DCB, subtract the 20 low-order bits in register 0 from the 20 low-order bits in register 1.

- If the result is negative, the tape is in read mode
- If the result is zero, no blocks are buffered
- If the result is positive, the tape is in write or read-backward mode.

Performing the High-Speed Search

The POINT macro moves the tape to a specific block. If the TYPE=ABS keyword is used, the search is done at high speed. If TYPE=REL is used, the search is done at normal speed.

When high speed is used to move the tape to the end of a data set, for example, to add a block to an existing data set, the BLOCKID that is used in the POINT macro might not yet exist. In this instance, the subsystem tries to move the tape to a position after a block with the preceding block number. If this block is also missing, an error results.

Usually, the POINT macro only uses BLOCKIDs that were previously retrieved by the NOTE macro. If the device type is not known when specifying TYPE=ABS, substitute zeros and the search is done at normal speed.

Comparing Conventional Search and High-Speed Search

This section discusses the differences in application program logic between conventional search (NOTE/POINT with TYPE=REL) and high-speed search (NOTE/POINT with TYPE=ABS).

The format of the BLOCKID used for high-speed search is different from the format of the relative block address used for conventional-speed search.

If TYPE=ABS is used with a device that does not support the high-speed search facility, an error return code is returned in register 15. If device independence is required, this condition must be checked and an alternate program path (using conventional-speed search) must be provided. Conventional-speed search is device-independent and can also be used with the 3480. However, the two search speeds cannot be mixed for a data set.

Note: All NOTE and POINT macro instructions for a given file must use the same TYPE keyword.

When applications are converted from conventional-speed search to high-speed search, the following differences exist:

- With TYPE=ABS, the NOTE and POINT macros provide a return code in register 15. The return code should be checked by the program.
- With TYPE=REL, no return code is provided in register 15.

The NOTE macro is usually used to locate the current logical position within the data set.

- With TYPE=ABS, this information is returned in register 0
- With TYPE=REL, this information is returned in register 1.

The information returned describes the current block in the data set, which can be either the next block that would process or the previous block that was just processed. For normal forward read or write processing:

- The **next** block for TYPE=ABS
- The **previous** block for TYPE=REL.

When the information is used later in a POINT macro instruction, it refers to the **next** block that would process for both TYPE=ABS and TYPE=REL.

Examples of How to Code NOTE and POINT

Example 1

This example includes Program A and Program B. Program A creates a tape file. While writing the data, Program A must save the information about the position of a certain block (for example, number 500) and pass that information to Program B. Program B moves the tape to block number 500 and reads it. You can select high-speed search or conventional search.

High-speed search implementation

Program A writes 499 blocks to the tape, then calls the NOTE macro instruction with the TYPE=ABS parameter. The information returned in register 0 refers to block number 500. Program A then continues to write from block number 500 until the end of the file. The information obtained by NOTE is saved and passed to Program B.

Program B uses this information as a search argument in a POINT macro instruction with the TYPE=ABS parameter. After that, a READ macro instruction retrieves block number 500.

Conventional search implementation

Program A writes 500 blocks to the tape, then calls the NOTE macro instruction with the TYPE=REL parameter. The information returned in register 1 refers to block number 500. Program A then continues to write from block number 501 until the end of the file. The information obtained by NOTE is saved to be passed to Program B.

Program B uses this information as a search argument in a POINT macro instruction with the TYPE=REL parameter. After that, a READ macro instruction retrieves block number 500.

Example 2

The following is an example of using NOTE and POINT for high-speed search. This example shows how an application reading through a data set on tape can reach a point at which information is required from another block in the same data set.

The NOTE macro is used to save the current tape position (it returns the BLOCKID of the successor block on the tape). Assuming the BLOCKID of the required block is provided within the block just read, issuing a POINT macro will perform a high-speed search within the data set to reach that block. After the block is read, a second POINT macro (using the saved BLOCKID) brings the tape back to the original position where sequential processing is resumed.

```
         ...
DCBIN    DCB     DDNAME=TAPEIN,DSORG=PS,MACRF=(RP)
*-------------------------------------
BUFFER   DS      0CL800                      NORMAL READ BUFFER
         ...
REFBLKID DS      CL4                         BLOCKID REFERENCE
         ...
         ORG     BUFFER +800
*-------------------------------------
RBUFFER  DS      CL800                       BUFFER FOR REFERENCED BLOCK
*-------------------------------------
BLOCKID  DS      F                           BLOCKID SAVE AREA
         ...
         ...
         OPEN    (DCBIN,(INPUT))             OPEN TAPE
         ...
LP1      READ    DECBIN1,SF,DCBIN,BUFFER,800 READ DATA FROM TAPE
         CHECK   DECBIN1
         ...                                 PROCESS THIS BLOCK
         ...
         L       2,REFBLKID                  IS THERE A NONSEQUENTIAL
         LTR     2,2                         BLOCK REFERENCED?
         BZ      LP1                         NO
         ...
         NOTE    DCBIN,TYPE=ABS              REMEMBER TAPE POSITION
         LTR     15,15                       ALL OKAY?
         BNZ     NOTEERR
         ST      0,BLOCKID                   SAVE TAPE POSITION
         ...
         POINT   DCBIN,(2),TYPE=ABS          GO TO REFERENCED BLOCK
         LTR     15,15                       ALL OKAY?
         BNZ     POINTERR
         ...
         READ    DECBIN2,SF,DCBIN,RBUFFER,800 READ REFERENCED BLOCK
         CHECK   DECBIN2
         ...                                 PROCESS THIS BLOCK
         ...
         POINT   DCBIN,BLOCKID,TYPE=ABS      GO BACK TO OLD PLACE
         LTR     15,15                       ALL OKAY?
         BNZ     POINTERR
         B       LP1                         RESUME SEQUENTIAL PROCESSING
         ...
EOF      CLOSE   DCBIN
         ...
NOTEERR  DS      0H                          NOTE ERROR RTN
         ...
POINTERR DS      0H                          POINT ERROR RTN
         ...
```

NOTE—Provide Relative Position (BPAM and BSAM)

The NOTE macro instruction causes the system to return the position of the last block read from or written into a data set. All input and output operations using the same data control block must be tested for completion before the NOTE macro instruction is issued.

The NOTE macro instruction is automatically provided when a partitioned data set is used (DSORG=PO or POU), but when a sequential data set (BSAM) is used, the use of NOTE/POINT macro instructions must be indicated in the MACRF operand of the DCB macro instruction.

The NOTE macro instruction cannot be used for SYSOUT data sets.

The NOTE macro is written:

[symbol]	**NOTE**	dcb address **[,TYPE={ABS|REL}]**

dcb address—RX-type address, (2−12), or (1)
The dcb address operand specifies the address of the data control block opened for the partitioned or sequential data set being processed.

TYPE={ABS|<u>REL</u>}

ABS
specifies that, after NOTE processes successfully (contents of register 15 is 0), register 0 contains the physical block identifier for the next data block waiting for transfer between *main storage* and the control unit buffer, and register 1 contains the physical block identifier of the next data block waiting for transfer between the control unit buffer and the *tape drive.*

If you subtract the low-order 20 bits of register 1 from the low-order 20 bits of register 0, the remainder is the number of data blocks left in the control unit buffer. A negative remainder means the buffer is in read mode, and a positive remainder means the buffer is in either write or read-backward mode. A zero remainder means that no data is buffered.

<u>REL</u>
causes the system to return the relative position of the last block read from or written into a data set. The position, in terms of the current volume, is returned in register 1. The block number is in binary, right-adjusted in register 1 with high-order bits set to zero. Do not use a NOTE macro instruction for tapes without standard labels when:

- The data set is opened for RDBACK (specified in the OPEN macro instruction) or
- The DISP parameter of the DD statement for the data set specifies DISP=MOD.

Completion Codes from NOTE

When you have specified the ABS parameter and the system returns control to your problem program, the low-order byte of register 15 contains a return code; the low-order byte of register 0 contains a reason code:

Return Code (15)	Reason Code (0)	Meaning
00 (hex 00)		Successful completion.
04 (hex 04)		Device does not support block identifier.
08 (hex 08)	01 (hex 01)	Incorrect parameter.
08 (hex 08)	02 (hex 02)	Incorrect DCB or a DEBCHK error.
08 (hex 08)	03 (hex 03)	Environmental error.
08 (hex 08)	11 (hex 0B)	Unsuccessful call to ESTAE macro.
08 (hex 08)	12 (hex 0C)	Unsuccessful GETMAIN request.
12 (hex 0C)		Input/output error.

POINT—Moving Tape to a Relative Block Position (BPAM and BSAM)

The POINT macro starts the next READ or WRITE operation at the specified data set block on the current volume. Before you issue the POINT macro, ensure that all input and output operations using the same data control block are tested for completion. If you are processing a data set that was opened for UPDAT, you must issue a READ macro immediately after the POINT macro. If you are processing an output data set, you must issue a WRITE macro immediately after the POINT macro before you close the data set, unless you already issued the CLOSE macro (with **TYPE=T** specified) before the POINT macro.

Note: If you specify the TYPE=T option in the CLOSE macro and you do not issue a WRITE macro before you close the data set, use the end-of-data location that is determined by TCLOSE.

The POINT macro is written:

[symbol]	**POINT**	dcb address ,block address **[,TYPE={ABS|REL}]**

dcb address—RX-type address, (2−12), or (1)
The dcb address operand specifies the address of the data control block for the opened data set that is to be moved into position.

block address—RX-type address, (2 – 12), or (0)

The block address operand indicates which block in the data set is to be processed next.

When TYPE = ABS is specified, the block address operand specifies the address (on a fullword boundary) of a fullword that contains the physical block identifier of the block that is to be processed next. This physical block identifier is provided as output from prior NOTE macro processing.

When TYPE = REL is specified or defaults, the block address operand specifies the address (on a fullword boundary) of a fullword that contains the relative address of the block that is to be processed next. In the relative address specification, the block number is in binary and is right-adjusted in the fullword with the high-order bits set to 0; add 1 if reading tape backward. Do not use the POINT macro instruction for tapes without standard labels when:

- The data set is opened for RDBACK or
- The DD statement for the data set specifies DISP = MOD.

When an end-of-data condition is encountered on magnetic tape, you must not issue the POINT macro instruction unless you have first moved the tape into position for processing within your data set; otherwise, the POINT operation will be unsuccessful. (Issuing CLOSE TYPE = T is an easy method to use to accomplish tape position in your EODAD routine.)

The first block of a magnetic tape cartridge data set is always specified by the hexadecimal value 01000000.

TYPE = {ABS|REL}

indicates whether the block address operand is a physical block identifier or a relative address.

ABS

indicates that the block address operand specifies an address (on a fullword boundary) of a fullword containing a physical block identifier of the block that is to be processed next.

REL

indicates that the block address operand specifies an address (on a fullword boundary) of a fullword containing the relative address of the block is to be processed next.

Note: POINT cannot be used for SYSIN or SYSOUT data sets.

If the tape position is not correct or if the block identification is not in the correct format, the error analysis (SYNAD) routine is given control when the next CHECK macro instruction is processed.

Completion Codes from POINT

When you have specified the ABS parameter and the system returns control to your problem program, the low-order byte of register 15 contains a return code; the low-order byte of register 0 contains a reason code:

Return Code (15)	Reason Code (0)	Meaning
00 (hex 00)		Successful completion.
04 (hex 04)		Device does not support block identifier.
08 (hex 08)	01 (hex 01)	Incorrect parameter.
08 (hex 08)	02 (hex 02)	Incorrect DCB or a DEBCHK error.
08 (hex 08)	03 (hex 03)	Environmental error.
08 (hex 08)	11 (hex 0B)	Unsuccessful call to ESTAE macro.
08 (hex 08)	12 (hex 0C)	Unsuccessful GETMAIN request.
12 (hex 0C)		Input/output error.

Controlling Data Synchronization (SYNCDEV)

Synchronization means ensuring that data is written on the tape before other commands are performed. A synchronization command to the 3480 subsystem forces the control unit to write any buffered write data to the addressed tape drive before it accepts any other commands. Certain other commands, such as Write Tape Mark, Rewind, Rewind Unload, Locate, and any read command issued after buffered write commands, cause the control unit buffer to synchronize by writing all buffered write data to the tape before the control unit performs the command.

The SYNCDEV macro controls data synchronization for devices that support buffered write mode with full function programming. The macro is only for output data sets. Read buffering is always done automatically and cannot be changed by the SYNCDEV macro.

The SYNCDEV macro operates on two levels: the logical level and the physical level. On the logical level, the macro controls the buffers established by the access method programming. These buffers can be controlled as to the level of buffering. On the physical level, the macro controls the control unit buffer. The control unit buffer cannot be controlled as to the level of buffering. Once the control unit receives a synchronization command for a tape drive, *all* the buffered write data for that drive is written on the tape before any other command is accepted for that drive.

If you want to wait until a critical record is physically written from the control unit buffer onto the tape, you can use the SYNCDEV macro.

If QSAM is used with more than one main storage buffer or blocked record, then issuing a SYNCDEV only clears the control unit buffers, but **not** any logical records that the access method has waiting for service. Therefore, synchronization on the logical record level can be achieved only for EXCP or BSAM, or QSAM and BUFNO=1 and unblocked records.

If you want to limit the number of data blocks that can be waiting in the control unit buffer, the SYNCDEV macro can be used to specify a threshold value.

Because the control unit does not keep the last threshold value (BUFBLK) given to it, every time synchronization is needed, SYNCDEV must be used.

Synchronization is done by comparing the BUFBLK parameter value and the actual buffering depth (number of blocks buffered in the control unit for this data set). If BUFBLK is less, all data in the buffer will be written to tape before control is returned to the program. A zero BUFBLK value causes an unconditional write of all buffered data.

The two ways to use the SYNCDEV macro include:

- To request information regarding synchronization
- To demand that synchronization occur, based on the number (depth of buffering) used in the BUFBLK parameter (0 means direct synchronization), or, depending on MF-type given in a halfword field addressed by the ABUFBLK parameter.

Error codes are returned in register 15 to the macro.

Example of How to Code SYNCDEV

```
         ...
         ...
DCBOUT   DCB     DDNAME=TAPEOUT,DSORG=PS,MACRF=(WP)
BUFFER   DS      CL800                          WRITE BUFFER
SWITCH1  DC      X'00'                          SWITCH
CRITICAL EQU     X'01'
         ...
         ...
SAMP2    DS      0H
         OPEN    (DCBOUT,(OUTPUT))              OPEN FILE
*                                               DISPLAY A USER MESSAGE
         MSGDISP RDY,DCB=DCBOUT,TXT='OKAY 1'
         LTR     15,15                          ALL OKAY?
         BNZ     MSGERR                         NO
         ...
LP1      ...                                    BUILD BUFFER TO WRITE
         ...                                    SOME BLOCKS MAY BE CRITICAL
         ...                                    SWITCH1 WILL BE SET TO X'01'
         WRITE   DECBOT1,SF,DCBOUT,BUFFER,800   WRITE DATA
         CHECK   DECBOT1
         TM      SWITCH1,CRITICAL               CRITICAL DATA BLOCK?
         BZ      LP1                            NO
         ...
*                                               WAIT FOR ALL DATA IN BUFFER
*                                               TO BE PHYSICALLY WRITTEN
         SYNCDEV DCB=DCBOUT,INQ=NO,BUFBLK=0
         LTR     15,15                          ALL OKAY?
         BNZ     SYNCERR                        NO
         MVI     SWITCH1,0                      RESET SWITCH
         B       LP1                            RETURN TO LOOP
         ...
END      CLOSE   DCBOUT                         CLOSE DATA SET
         ...
         ...
MSGERR   DS      0H                             MSGDISP ERROR RTN
         ...
SYNCERR  DS      0H                             SYNCDEV ERROR RTN
         ...
```

SYNCDEV—Synchronize Device

The SYNCDEV macro instruction allows you to control data synchronization for the 3480 subsystem, which supports buffered write mode. Data records in the control unit buffer may not yet be on tape when your program is ready to send more. There is no way to determine how much data is left in the buffer, and it is time-dependent to tape motion. You can use the SYNCDEV macro to either:

- Request information regarding synchronization
- Demand synchronization if the specified number of data blocks are buffered.

  If more blocks are buffered than were specified, the system stays in control until all the blocks are written on the tape or it detects an I/O error.

  If the same number or fewer blocks are buffered, buffering is not affected.

Note: Demands for synchronization are ignored if the drive is in read mode.

The SYNCDEV macro is written:

[symbol]	**SYNCDEV**	**DCB** = addr **[,{ABUFBLK** = addr| **BUFBLK** = {maximum buffer depth|0}**]** **[,INQ** = {**YES**|**NO**}**]**

The following describes the operands that can be specified for SYNCDEV.

DCB = *addr*—A-type address or (2 – 12)
specifies the address of the data control block.

ABUFBLK = *addr*|**BUFBLK** = {*maximum buffer depth*|*0*}
specifies the maximum number of data blocks that can be buffered.

ABUFBLK = *addr*—A-type address or (2 – 12)
specifies the address (on a halfword boundary) of a halfword that contains a value that specifies the maximum number of data blocks that can be buffered.

BUFBLK = *maximum buffer depth*
specifies the maximum number of data blocks that can be buffered. This number can be an absolute value from 0 to 65535. The BUFBLK value can be in the 2 low-order bytes of a register (2 – 12).

0
If neither ABUFBLK nor BUFBLK is specified, the number of data blocks that can be buffered defaults to 0, and no data blocks are buffered.

INQ={YES|NO}
specifies whether this is a request for information about the degree of synchronization or a request for synchronization.

YES
specifies an inquiry as to how many data blocks are in the buffer.

If the previous operation completed successfully, register 0 contains the number of buffered physical blocks.

NO
specifies a request for synchronization based on the number of data blocks that can be buffered as specified in ABUFBLK or BUFBLK.

Note: Do not use this option in 31-bit residence mode; it requires a 24-bit addressing mode parameter list.

SYNCDEV—List Form

The list form of the SYNCDEV macro is written:

[symbol]	**SYNCDEV**	**[DCB=**addr**]** **[,BUFBLK=**{maximum buffer depth|0}**]** **[,INQ={YES|NO}]** **,MF=L**

The following describes the operands that can be specified for the list form of SYNCDEV.

DCB=*addr*—A-type address
specifies the address of the data control block.

BUFBLK=*maximum buffer depth|0*
specifies the maximum number of data blocks that can be buffered. This number can be an absolute value from 0 to 65535. If BUFBLK is not specified, the number of data blocks that can be buffered defaults to 0, and no data blocks are buffered.

INQ={YES|NO}
specifies whether this is a request for information about the degree of synchronization or a request for synchronization.

YES
specifies an inquiry as to how many data blocks are in the buffer.

NO
specifies a request for synchronization based on the number of data blocks that can be buffered as specified in BUFBLK.

MF=L

generates a 24-bit addressing-mode parameter list that contains no executable instructions. The list can be used as input and can be modified by the execute form of the SYNCDEV macro.

Note: Do not use this option in 31-bit residence mode; it requires a 24-bit addressing-mode parameter list.

SYNCDEV—Execute Form

The execute form of the SYNCDEV macro is written:

[symbol]	**SYNCDEV**	**[DCB=**addr**]** **[,{ABUFBLK=**addr| **BUFBLK=**{maximum buffer depth|0}**]** **[,INQ={YES|NO}]** **,MF=(E,addr)**

The following describes the operands that can be specified for the execute form of SYNCDEV.

DCB=*addr*—A-type address or (2–12)

specifies the address of the data control block.

ABUFBLK=*addr***\|BUFBLK={***maximum buffer depth***\|***0***}**

specifies the maximum number of data blocks that can be buffered.

ABUFBLK=*addr*—A-type address or (2–12)

specifies the address (on a halfword boundary) of a halfword that contains a value that specifies the maximum number of data blocks that can be buffered.

BUFBLK=*maximum buffer depth*

specifies the maximum number of data blocks that can be buffered. This number can be an absolute value from 0 to 65535. The BUFBLK value can be in the 2 low-order bytes of a register (2–12).

INQ={YES\|NO}

specifies whether this is a request for information about the degree of synchronization or a request for synchronization.

YES

specifies an inquiry as to how many data blocks are in the buffer.

NO

specifies a request for synchronization based on the number of data blocks that can be buffered as specified in ABUFBLK or BUFBLK.

Register 0 contains the number of buffered physical blocks if the previous operation completed successfully.

MF = (E,*addr*)
specifies the execute form of SYNCDEV.

addr—**A-type address, RX-type address, or (2 – 12)**
specifies the 24-bit addressing-mode address for the parameter list.

Note: Do not use this option in 31-bit residence mode; it requires a 24-bit addressing mode parameter list.

Completion Codes from SYNCDEV

When the system returns control to your problem program, the low-order byte of register 15 contains a return code; the low-order byte of register 0 contains a reason code:

Return Code (15)	Reason Code (0)	Meaning
00 (hex 00)		Successful completion. Register 0 contains the number of data blocks in the control unit buffer.
04 (hex 04)	01 (hex 01)	Incorrect parameter.
04 (hex 04)	02 (hex 02)	Incorrect DCB or a DEBCHK error.
04 (hex 04)	03 (hex 03)	Environmental error.
04 (hex 04)	04 (hex 04)	Incorrect input to NOTE.
04 (hex 04)	05 (hex 05)	Device does not support buffering.
04 (hex 04)	11 (hex 0B)	Unsuccessful call to ESTAE macro.
04 (hex 04)	12 (hex 0C)	Unsuccessful GETMAIN request.
08 (hex 08)		Permanent I/O error during read block ID or synchronize command.
12 (hex 0C)		Permanent I/O error on the last channel program with loss of data.

Note: If you specified a SYNAD option in the DCB and issue a PUT or CHECK macro after this error occurs, your program cannot enter the SYNAD routine.

Obtaining I/O Device Characteristics (DEVTYPE)

Use the DEVTYPE macro instruction to request information relating to the characteristics of an I/O device and to cause this information to be placed into a specified area. (The results of a DEVTYPE macro instruction processed before a checkpoint is taken should not be considered valid after a checkpoint/restart occurs.) The IHADVA macro maps the data returned by the DEVTYPE macro. For further information about the DEVTYPE macro, see *MVS/370 System Programming Library: Data Management* or *MVS/Extended Architecture System—Data Administration.*

For the IBM 3480, the DEVTYPE macro, device characteristics, and particular output include:

- The device
  - 3480 (18-track)
- Maximum record size (word 1, in decimal)
  - 32760
- DEVTAB (words 2, 3, and 4, in hexadecimal)
  - Not applicable
- RPS (word 5, in hexadecimal)
  - Not applicable.

The UCBTYP field for the 3480 is:

- '78008080' for full function mode
- '33008003' for compatibility mode.

Chapter 5. Displaying Messages (MSGDISP)

The MSGDISP macro allows you to display a message on the 3480 using full function programming support. Any hardware-generated message on the display (except the error message CHK xy) is overlaid when an MSGDISP macro is issued.

Displaying a Message from any User Program (MSGDISP RDY)

This function displays an application program message.

The MSGDISP macro with the RDY parameter allows every user program to display a message. The display request can be issued to any tape device that is allocated and opened by the program.

The text supplied in the TXT parameter is displayed in positions 2 through 7 of the message display while the tape data set is open. The text is shown in parentheses. The message text is also logged to the tape pool operator console.

The following is an example of how to use the MSGDISP RDY macro:

Example of How to Code MSGDISP RDY

```
         ...
         ...
DCBOUT   DCB     DDNAME=TAPEOUT,DSORG=PS,MACRF=(WP)
BUFFER   DS      CL800                          WRITE BUFFER
SWITCH1  DC      X'00'                          SWITCH
CRITICAL EQU     X'01'
         ...
         ...
SAMP2    DS      0H
         OPEN    (DCBOUT,(OUTPUT))              OPEN FILE
*                                               DISPLAY A USER MESSAGE
         MSGDISP RDY,DCB=DCBOUT,TXT='OKAY 1'
         LTR     15,15                          ALL OKAY?
         BNZ     MSGERR                         NO
         ...
LP1      ...                                    BUILD BUFFER TO WRITE
         ...                                    SOME BLOCKS MIGHT BE CRITICAL
         ...                                    SWITCH1 WILL BE SET TO X'01'
         WRITE   DECBOT1,SF,DCBOUT,BUFFER,800   WRITE DATA
         CHECK   DECBOT1
         TM      SWITCH1,CRITICAL               CRITICAL DATA BLOCK?
         BZ      LP1                            NO
         ...
*                                               WAIT FOR ALL DATA IN CU BUFF
*                                               TO BE PHYSICALLY WRITTEN
         SYNCDEV DCB=DCBOUT,INQ=NO,BUFBLK=0
         LTR     15,15                          ALL OKAY?
         BNZ     SYNCERR                        NO
         MVI     SWITCH1,0                      RESET SWITCH
         B       LP1                            RETURN TO LOOP
         ...
END      CLOSE   DCBOUT                         CLOSE DATA SET
         ...
         ...
MSGERR   DS      0H                             MSGDISP ERROR RTN
         ...
SYNCERR  DS      0H                             SYNCDEV ERROR RTN
         ...
```

MSGDISP—Displaying a Ready Message

The MSGDISP macro is written:

[symbol]	MSGDISP	RDY ,DCB=addr [,TXT={'msgtxt'|addr}]

RDY
specifies that text supplied in the TXT parameter be displayed in positions 2 through 7 of the display while the data set is open. The display is steady (not flashing) and is enclosed in parentheses. The display is also written to the tape pool console (routing code 3, descriptor code 7).

DCB=*addr*
specifies the address of a DCB opened to a data set on the mounted volume. If more than one device is allocated, the message display is directed to the device containing the volume currently in use.

Note: If more than one device or more than one volume is allocated, you can update a message display after an end-of-volume condition by using the EOV exit specified in a DCB exit list. In the case of a concatenated data set with unlike characteristics, the open DCB exit can be used to update the display.

addr—RX-type address, A-type address, or (2–12)
specifies an in-storage address of the opened DCB.

TXT={'msgtxt'|*addr***}**
specifies up to six characters be displayed in positions 2 through 7. If TXT is not specified, blanks are displayed.

'msgtxt'
specifies the 1- to 6-character text. The text must be enclosed in apostrophes.

addr—RX-type address, A-type address, or (2–12)
specifies an in-storage address of an area containing the text to be displayed.

MSGDISP—Displaying a Ready Message List Form

The list form of the MSGDISP macro is written:

[symbol]	**MSGDISP**	**[RDY]** **[,DCB=**addr**]** **,MF=L** **[,TXT={'msgtxt'|addr}]**

RDY
specifies that text supplied in the TXT parameter be displayed in positions 2 through 7 while a data set is open. The display is steady (not flashing) and is enclosed in parentheses. The display is also written to the tape pool console (routing code 3, descriptor code 7).

DCB=*addr*
specifies the address of a DCB opened to a data set on the mounted volume. If more than one device is allocated, the message display is directed to the device containing the volume currently in use.

Note: If more than one device or more than one volume is allocated, you can update a message display after an end-of-volume condition by using the EOV exit specified in a DCB exit list. In the case of a concatenated data set with unlike characteristics, the open DCB exit can be used to update the display.

addr—A-type address
specifies an in-storage address of the opened DCB.

MF=L
specifies the list form of MSGDISP. This generates a parameter list that contains no executable instructions. The list can be used as input to and can be modified by the execute form of the macro.

TXT={'msgtxt'|addr}
specifies up to six characters be displayed in positions 2 through 7. If TXT is not specified, blanks are displayed.

'msgtxt'
specifies the 1 to 6 character text. Enclose the text in apostrophes.

addr—A-type address
specifies an in-storage address of an area containing the text to be displayed.

MSGDISP—Displaying a Ready Message Execute Form

The execute form of the MSGDISP macro is written:

[symbol]	**MSGDISP**	**RDY** **[,DCB=**addr**]** **,MF=(E,**addr**)** **[,TXT=**{'msgtxt'|addr}**]**

RDY
specifies that text supplied in the TXT parameter be displayed in positions 2 through 7 while a data set is open. The display is steady (not flashing) and is enclosed in parentheses. The display is also written to the tape pool console (routing code 3, descriptor code 7).

DCB=*addr*
specifies the address of a DCB opened to a data set on the mounted volume. If more than one device is allocated, the message display is directed to the device containing the volume currently in use.

Note: If more than one device or more than one volume is allocated, you can update a message display after an end-of-volume condition by using the EOV exit specified in a DCB exit list. In the case of a concatenated data set with unlike characteristics, the open DCB exit can be used to update the display.

addr—RX-type address, A-type address, or (2–12)
specifies an in-storage address of the opened DCB.

MF=(E,*addr***)**
specifies that execute form of MSGDISP and an existing parameter list be used.

addr—RX-type address, (1), or (2–12)
specifies an in-storage address of the parameter list.

TXT={'msgtxt'|*addr***}**
specifies up to six characters be displayed in positions 2 through 7. If TXT is not specified, blanks are displayed.

'msgtxt'
specifies the 1 to 6 character text. Enclose the text in apostrophes.

addr—RX-type address, A-type address, or (2–12)
specifies an in-storage address of an area containing the text to be displayed.

Displaying a Message from Authorized Programs

The other MSGDISP macros require the program to be in supervisor state, have a storage protect key of 0 through 7, or be authorized by the authorized program facility (APF).

The following functions are available for APF authorized programs:

- Displaying a mount volume message

  The MOUNT parameter specifies that an M be displayed in position 1 of the display area during a mount request. The next six characters contain the volume serial number and the last character contains the label type. The display flashes repeatedly until the volume is loaded and the device is ready. The M is not displayed if the volume is already mounted.

- Displaying a verify volume message

  The VERIFY parameter displays the serial number and label type of the mounted volume to indicate that the volume was verified and accepted.

- Displaying a demount volume message

  The DEMOUNT parameter displays a volume disposition indicator and volume serial number until a volume is demounted by the operator. Optionally, another volume to be mounted instead of the one to be demounted can also be specified. If this option is used, the demount and mount messages appear alternately on the display.

- Resetting a display

  The RESET parameter clears the display of any message generated by a previous MSGDISP request (an asterisk appears on the display). Then any waiting hardware-generated messages are shown.

- Generalizing a display

  The GEN parameter displays one or two 8-byte messages of any content. The messages can be steady or flashing. Two messages can be specified to be swapped either repeatedly or at the time the operator inserts or removes a cartridge.

For MVS/XA, the IOSLEVEL (priority) of the request can be specified with the FORCE parameter. IOSLEVEL support replaces single level I/O quiescing with multilevel quiescing; the higher the IOSLEVEL value, the greater the user's priority to control the device.

The MSGDISP macro generates a parameter list as input to an SVC routine. The contents of the parameter list are described in *OS/VS2 System Programming Library: Debugging Handbook* under the section that describes SVC routines.

MSGDISP can be coded in the standard, execute, and list forms.

MSGDISP—Displaying a Mount Message

The format for specifying MSGDISP with the MOUNT parameter is:

[symbol]	MSGDISP	MOUNT ,UCB=addr [,FORCE={NO|YES|n|keyword|(reg)}] [,LABEL={'A'|'N'|'S'|'X'|addr}] [,MF={L|(E,addr)}][1] [,SER={'volser'|addr}] [,TEXT={NO|YES}] [,WAIT={NO|YES}]

[1]If the MF parameter is not specified, the standard form of the macro is used.

MOUNT
specifies that an M be displayed in position 1 of the display area during a mount request. The M is followed by a volume serial number and label type. The data flashes repeatedly until a volume is loaded and ready. If the device is ready at the time a mount request is issued, the M is not displayed.

UCB=*addr*—RX-type address, or (2 – 12)
specifies a register containing the UCB address for the device.

For MVS/370 users:

FORCE=
specifies whether and when a display request processes for a device while I/O is being quiesced.

NO
specifies that a display request cannot process for a device while I/O is being quiesced.

YES
specifies that a display request processes for a device while I/O is being quiesced.

For MVS/XA users:

FORCE=
specifies the priority (IOSLEVEL) for the I/O of the request to be processed. The higher the IOSLEVEL value, the greater the priority for the display to be processed.

If the FORCE parameter is not specified, the default is FORCE=NO.

NO

specifies that a display request cannot process for a device if I/O is being quiesced. The IOSLEVEL is set to the installation default, as indicated in the CVTIONLV field of the CVT.

YES

specifies that a display request processes for a device even though I/O is being quiesced. The IOSLEVEL is set to 9, the highest priority.

n

specifies a decimal number from 1 to 9 be used as the IOSLEVEL value. A high number indicates a higher priority request for the device.

keyword

specifies a label equated to an IOSLEVEL value:

NORMAL	1
QUIESCE	2
DAVV	3
DDR	4
DYNPATH	5
UNCRSV	6
CHPRCVY	7
SCHRCVY	8
FDEV	9

(*reg*)

Specifies that a value of 1 through 9 be passed in the low-order byte of registers 2 through 12.

For all users:

LABEL=

specifies label type of the mounted volume be displayed in position 8. If an unknown label type other than a blank is specified, a ? is displayed.

addr—RX-type address, A-type address, or (2−12)

specifies an in-storage address of an area containing an A, N, S, or X (see explanations below for these characters). For MF=L, only an A-type address can be specified.

'A'

specifies ISO/ANSI/FIPS (AL) or ISO/ANSI/FIPS with user (AUL) labels. Specify in apostrophes.

'N'

specifies no labels (NL), LTM (DOS), or bypass label processing (BLP). Specify in apostrophes.

'S'
specifies IBM Standard (SL) or IBM Standard with user (SUL) labels. Specify in apostrophes.

'X'
specifies nonstandard (NSL) labels. Specify in apostrophes.

MF=
specifies either the execute or list form of MSGDISP. If this parameter is not specified, the standard form of the macro is used.

L
specifies the list form of MSGDISP. This generates a parameter list that does not contain any executable instructions. The list can be used as input to and can be modified by the execute form of the macro.

(E,*addr*)
specifies that the execute form of the macro and an existing parameter list be used.

addr—RX-type address, (1), or (2−12)
specifies an in-storage address of the parameter list.

SER=
specifies the serial number of the volume to be mounted. The serial number is displayed in positions 2 through 7. If SER is not specified, the system supplies the volume serial number. If the serial number is not available, a scratch volume is used, unless the volume use attribute indicates a default of PRIVAT.

addr—RX-type address, A-type address, or (2−12)
specifies an in-storage address of the volume serial number. For MF=L, only an A-type address can be specified.

'volser'
specifies the volume serial number as a literal. Specify in apostrophes.

TEST=
specifies if the UCB is to be tested to determine whether the device supports a message display before calling the message display SVC routine.

NO
specifies that the UCB is to be tested by the SVC routine.

YES
specifies that the UCB is to be tested before the SVC call.

Note: TEST=YES requires that the UCB mapping macro (IEFUCBOB) is to be included in the source code.

WAIT=
specifies when control is returned to the caller.

NO
specifies that control is returned before I/O is complete. I/O return codes are not returned, and I/O errors are recorded in the same manner as any permanent error by the error recovery procedure.

YES
specifies that control is returned after I/O is complete.

MSGDISP—Displaying a Verify Message

The format for specifying MSGDISP with the VERIFY parameter is:

[symbol]	**MSGDISP**	**VERIFY** **,UCB=**addr **[,FORCE={NO|YES|**n|keyword|**(reg)}]** **[,LABEL={'A'|'N'|'S'|'X'|**addr**}]** **[,MF={L|(E,**addr**)}]**[1] **[,SER={**'volser'|addr**}]** **[,TEST={NO|YES}]** **[,WAIT={NO|YES}]**

[1]If the MF parameter is not specified, the standard form of the macro is used.

VERIFY
specifies that the serial number and label type of a volume that has been accepted be displayed in positions 2 through 8. Position 1 remains blank. The display continues until the next display request is processed.

UCB=*addr*—RX-type address, or (2−12)
specifies a register containing the UCB address for the device.

For MVS/370 users:

FORCE=
specifies whether a display request processes for a device while I/O is being quiesced.

NO
specifies that a display request cannot process for a device while I/O is being quiesced.

YES
specifies that a display request processes for a device while I/O is being quiesced.

For MVS/XA users:

FORCE=
specifies the priority (IOSLEVEL) for the I/O of the request to be processed. The higher the IOSLEVEL value, the greater the priority for the display to be processed.

If the FORCE parameter is not specified, the default is FORCE=NO.

NO
specifies that a display request cannot process for a device if I/O is being quiesced. The IOSLEVEL is set to the installation default, as indicated in the CVTIONLV field of the CVT.

YES
specifies that a display request processes for a device even though I/O is being quiesced. The IOSLEVEL is set to 9, the highest priority.

n
specifies a decimal number from 1 to 9 be used as the IOSLEVEL value. A high number indicates a higher priority request for the device.

keyword
specifies a label equated to an IOSLEVEL value:

NORMAL	1
QUIESCE	2
DAVV	3
DDR	4
DYNPATH	5
UNCRSV	6
CHPRCVY	7
SCHRCVY	8
FDEV	9

(*reg*)
Specifies that a value of 1 through 9 be passed in the low order byte of register 2 through 12.

For all users:

LABEL=
specifies label type of the mounted volume be displayed in position 8. If an unknown label type other than a blank is specified, a ? displays.

addr—RX-type address, A-type address, or (2 – 12)
specifies an in-storage address of an area containing an A, N, S, or X (see explanations below for these characters). For MF=L, only an A-type address can be specified.

'A'
specifies ISO/ANSI/FIPS (AL) or ISO/ANSI/FIPS with user (AUL) labels. Specify in apostrophes.

'N'
specifies no labels (NL), LTM (DOS), or bypass label processing (BLP). Specify in apostrophes.

'S'
specifies IBM Standard (SL) or IBM Standard with user (SUL) labels. Specify in apostrophes.

'X'
specifies Nonstandard (NSL) labels. Specify in apostrophes.

MF=
specifies either the execute or list form of MSGDISP. If this parameter is not specified, the standard form of the macro is used.

L
specifies the list form of MSGDISP, which generates a parameter list that does not contain any executable instructions. The list can be used as input to and can be modified by the execute form of the macro.

(E,*addr*)
specifies that the execute form of the macro and an existing parameter list be used.

addr—RX-type address, (1), or (2 – 12)
specifies an in-storage address of the parameter list.

SER=
specifies the serial number of the volume that has been verified. The serial number is displayed in positions 2 through 7. If SER is not specified, the system supplies the volume serial number. If the serial number is not available, a scratch volume is used, unless the volume use attribute indicates a default of PRIVAT.

addr—RX-type address, A-type address, or (2 – 12)
specifies an in-storage address of the volume serial number. For MF=L, only an A-type address can be specified.

'volser'
specifies the volume serial number as a literal. Specify in apostrophes.

TEST=
specifies whether the UCB is to be tested to determine if the device supports a message display before calling the message display SVC routine.

NO
specifies that the UCB is tested by the SVC routine.

YES
specifies that the UCB is tested before the SVC call.

Note: TEST=YES requires that the UCB mapping macro (IEFUCBOB) be included in the source code.

WAIT=
specifies when control is returned to the caller.

NO
specifies that control is returned before I/O is complete. I/O return codes are not returned, and I/O errors are recorded in the same manner as any permanent error by the error recovery procedure.

YES
specifies that control is returned after I/O is complete.

MSGDISP—Displaying a Demount Message

The format for specifying MSGDISP with the DEMOUNT parameter is:

[symbol]	MSGDISP	DEMOUNT ,UCB=addr [,DISP={'D'|'K'|'R'|addr}] [,FORCE={NO|YES|n|keyword|(reg)}] [,MF={L|(E,addr)}][1] [,MLABEL={'A'|'N'|'S'|'addr}] [,MSER={'volser-to-mount'|addr}] [,SER={'volser'|addr}] [,TEST={NO|YES}] [,WAIT={NO|YES}]

[1]If the MF parameter is not specified, the standard form of the macro is used.

DEMOUNT
specifies that a volume disposition indicator be displayed in position 1 until a volume is demounted. As an option, the indicator can be followed by the serial number of the volume to be demounted. The display flashes repeatedly. If a volume is not mounted on the device when the display request is processed, blanks display.

The demount message can be alternated with a mount message for the next volume by specifying the MSER parameter.

UCB=*addr*—RX-type address, or (2–12)
specifies a register containing the UCB address for the device.

DISP=
specifies a character to be displayed in position 1 representing the volume disposition.

addr—RX-type address, A-type address, or (2–12)
specifies an in-storage address of an area containing a D, K, or R (see explanations below for these characters). For MF=L, only an A-type address can be specified.

'D'
Demount a public volume. Specify in apostrophes.

Note: 'D' is also displayed when an invalid character is specified or when the volume use attribute is unknown (as in an automatic volume recognition (AVR) error in reading a label).

'K'
Keep a private volume and return it to the library. Specify in apostrophes.

'R'

Retain a private volume near the device for further use. Specify in apostrophes.

For MVS/370 users:

FORCE=

specifies whether a display request processes for a device while I/O is being quiesced.

NO

specifies that a display request cannot process for a device while I/O is being quiesced.

YES

specifies that a display request processes for a device while I/O is being quiesced.

For MVS/XA users:

FORCE=

specifies the priority (IOSLEVEL) for the I/O of the request to be processed. The higher the IOSLEVEL value, the greater the priority for the display to be processed.

If the FORCE parameter is not specified, the default is FORCE=NO.

NO

specifies that a display request cannot process for a device if I/O is being quiesced. The IOSLEVEL is set to the installation default, as indicated in the CVTIONLV field of the CVT.

YES

specifies that a display request processes for a device even though I/O is being quiesced. The IOSLEVEL is set to 9, the highest priority.

n

specifies a decimal number from 1 to 9 be used as the IOSLEVEL value. A high number indicates a higher priority request for the device.

keyword

specifies a label that equates to an IOSLEVEL value:

NORMAL	1
QUIESCE	2
DAVV	3
DDR	4
DYNPATH	5
UNCRSV	6
CHPRCVY	7
SCHRCVY	8
FDEV	9

(*reg*)

Specifies that a value of 1 through 9 be passed in the low order byte of register 2 through 12.

For all users:

MLABEL=

specifies that the label type of the volume to be loaded and made ready following a demount be displayed in position 8. If an unknown label type other than a blank is specified, a ? is displayed. This parameter can be specified only if the MSER parameter is specified.

addr—RX-type address, A-type address, or (2−12)

specifies an in-storage address of an area containing an 'A', 'N', 'S', or 'X' (see explanations below for these characters). For MF=L, only an A-type address can be specified.

'A'

specifies ISO/ANSI/FIPS (AL) or ISO/ANSI/FIPS with user (AUL) labels. Specify in apostrophes.

'N'

specifies no labels (NL), LTM (DOS), or bypass label processing (BLP). Specify in apostrophes.

'S'

specifies IBM Standard (SL) or IBM Standard with user (SUL) labels. Specify in apostrophes.

'X'

specifies Nonstandard (NSL) labels. Specify in apostrophes.

MF=

specifies either the execute or list form of MSGDISP. If this parameter is not specified, the standard form of the macro is used.

L

specifies the list form of MSGDISP, which generates a parameter list that does not contain any executable instructions. The list can be used as input to and can be modified by the execute form of the macro.

(E,*addr*)

specifies that the execute form of the macro and an existing parameter list be used.

addr—RX-type address, (1), or (2−12)

specifies an in-storage address of the parameter list.

MSER=

specifies that a mount message for the next volume be alternated with the demount message. The two messages continue to alternate until the current volume is demounted, at which time the mount message flashes repeatedly until a volume is loaded and ready. If no volume is

mounted at the time the demount message is processed with the mount message, only the mount message flashes until a volume is loaded and ready.

addr—RX-type address, A-type address, or (2−12)
specifies an in-storage address of the serial number of the volume to be mounted. For MF=L, only an A-type address can be specified.

'volser-to-mount'
specifies the volume serial number (of the volume to be mounted) as a literal. Specify in apostrophes.

SER=
specifies the serial number of the volume to be demounted. The serial number displays in positions 2 through 7. If SER is not specified, the system supplies the volume serial number. If the serial number is not available, a scratch volume is used, unless the volume use attribute indicates a default of PRIVAT.

addr—RX-type address, A-type address, or (2−12)
specifies an in-storage address of the volume serial number. This paramater is not valid for the MF=L form. For MF=L, only an A-type address can be specified.

'volser'
specifies the volume serial number as a literal. Specify in apostrophes.

TEST=
specifies whether the UCB is to be tested to determine if the device supports a message display before calling the message display SVC routine.

NO
specifies that the UCB be tested by the SVC routine.

YES
specifies that the UCB be tested before the SVC call.

Note: TEST=YES requires that the UCB mapping macro (IEFUCBOB) be included in the source code.

WAIT=
specifies when control is returned to the caller.

NO
specifies that control is returned before I/O is complete. I/O return codes are not returned, and I/O errors are recorded in the same manner as any permanent error by the error recovery procedure.

YES
specifies that control is returned after I/O is complete.

MSGDISP—Resetting the Message Display

The format for specifying MSGDISP with the RESET parameter is:

[symbol]	**MSGDISP**	**RESET** **,{UCB=**addr**|,UCBL=**addr**}** **[,FORCE={NO|YES|**n**|**keyword**|(**reg**)}]** **[,MF={L|(E,**addr**)}]**[1] **[,TEST={NO|YES}]** **[,WAIT={NO|YES}]**

[1]If the MF parameter is not specified, the standard form of the macro is used.

RESET
specifies that all existing data on the display be cleared. If WAIT=NO is specified, the display does not clear if a demount was the last service requested.

After being cleared, the display changes to show the *internal* status message for the device (for example, a message indicating if the device is ready).

UCB=*addr*—RX-type address, or (2−12)
specifies a register containing the UCB address for the device.

UCBL=*addr*—RX-type address, A-type address, (0), or (2−12)
specifies the address of a list containing a maximum of 64 words. Each word contains the address of a UCB representing a device with a display to be reset. The list is ended by the high-order bit in the last address word being set. If an error is encountered while processing the list, register 1 points to the subject UCB when control is returned to the caller.

UCBL cannot be specified with TEST=YES and WAIT=NO.

For MVS/370 users:

FORCE=
specifies whether a display request processes for a device while I/O is being quiesced.

NO
specifies that a display request cannot process for a device while I/O is being quiesced.

YES
specifies that a display request processes for a device while I/O is being quiesced.

For MVS/XA users:

FORCE=

specifies the priority (IOSLEVEL) for the I/O of the request to be processed. The higher the IOSLEVEL value, the greater the priority for the display to be processed.

If the FORCE parameter is not specified, the default is FORCE=NO.

NO

specifies that a display request cannot process for a device if I/O is being quiesced. The IOSLEVEL is set to the installation default, as indicated in the CVTIONLV field of the CVT.

YES

specifies that a display request processes for a device even though I/O is being quiesced. The IOSLEVEL is set to 9, the highest priority.

n

specifies a decimal number from 1 to 9 be used as the IOSLEVEL value. A high number indicates a higher priority request for the device.

keyword

specifies a label equated to an IOSLEVEL value:

NORMAL	1
QUIESCE	2
DAVV	3
DDR	4
DYNPATH	5
UNCRSV	6
CHPRCVY	7
SCHRCVY	8
FDEV	9

(*reg*)

Specifies that a value of 1 through 9 be passed in the low order byte of register 2 through 12.

For all users:

MF=

specifies either the execute or list form of MSGDISP. If this parameter is not specified, the standard form of the macro is used.

L

specifies the list form of MSGDISP, which generates a parameter list that does not contain any executable instructions. The list can be used as input to and can be modified by the execute form of the macro.

(E,*addr*)

specifies that the execute form of the macro and an existing parameter list be used.

addr—RX-type address, (1), or (2 – 12)
specifies the address of the parameter list.

TEST=

specifies whether the UCB is to be tested to determine if the device supports a message display before calling the message display SVC routine.

NO

specifies that the UCB be tested by the SVC routine.

YES

specifies that the UCB be tested before the SVC call.

Note: TEST=YES is not allowed if the UCBL parameter is specified. TEST=YES requires that the UCB mapping macro (IEFUCBOB) be included in the source code.

WAIT=

specifies when control is returned to the caller.

NO

specifies that control is returned before I/O is complete. I/O return codes are not returned, and I/O errors are recorded in the same manner as any permanent error by the error recovery procedure.

WAIT=NO is not allowed when the UCBL parameter is specified.

YES

specifies that control is returned after I/O is complete.

Note: Demount messages can be reset only if WAIT=YES is specified.

MSGDISP—Providing the Full Range of Display Options

The format of MSGDISP when specifying the GEN parameter is:

[symbol]	**MSGDISP**	**GEN** **,UCB** = addr **[,FLASH** = **{STEADY|STEADY2|** **BLINK|BLINK2|ALT}]** **[,FORCE** = **{NO|YES|**n|keyword|**(reg)}]** **[,MF** = **{L|(E,**addr**)}]**[1] **[,TEST** = **{NO|YES}]** **[,TXT** = **{**'msgtxt'|addr**}]** **[,TXT2** = **{**'altmsgtxt'|addr**}]** **[,VOL** = **{STATIC|REMOVE|INSERT|SWAP}]** **[,WAIT** = **{NO|YES}]**

[1]If the MF parameter is not specified, the standard form of the macro is used.

GEN
specifies the full range of display options.

UCB = *addr*—RX-type address, or (2 – 12)
specifies a register containing the UCB address for the device.

FLASH =
specifies how to display messages.

Note: If VOL = SWAP is specified, messages are shown as if FLASH = ALT was specified.

STEADY
specifies that the primary message (TXT) be shown steady (not flashing).

STEADY2
specifies that the alternate message (TXT2) be shown steady (not flashing).

BLINK
specifies that the primary message (TXT) flash repeatedly at a rate of approximately two seconds on and a half second off.

BLINK2
specifies that the alternate message (TXT2) flash repeatedly at a rate of approximately two seconds on and a half second off.

ALT
specifies that the primary and alternate messages (TXT and TXT2) display alternately and repeat at a rate of approximately two seconds on and a half second off.

For MVS/370 users:

FORCE=
specifies whether a display request processes for a device while I/O is being quiesced.

NO
specifies that a display request cannot process for a device while I/O is being quiesced.

YES
specifies that a display request processes for a device while I/O is being quiesced.

For MVS/XA users:

FORCE=
specifies the priority (IOSLEVEL) for the I/O of the request to be processed. The higher the IOSLEVEL value, the greater the priority for the display to be processed.

If the FORCE parameter is not specified, the default is FORCE=NO.

NO
specifies that a display request cannot process for a device if I/O is being quiesced. The IOSLEVEL is set to the installation default, as indicated in the CVTIONLV field of the CVT.

YES
specifies that a display request processes for a device even though I/O is being quiesced. The IOSLEVEL is set to 9, the highest priority.

n
specifies a decimal number from 1 to 9 be used as the IOSLEVEL value. A high number indicates a higher priority request for the device.

keyword
specifies a label that equals an IOSLEVEL value:

NORMAL	1
QUIESCE	2
DAVV	3
DDR	4
DYNPATH	5
UNCRSV	6
CHPRCVY	7
SCHRCVY	8
FDEV	9

(*reg*)
Specifies that a value of 1 through 9 be passed in the low order byte of register 2 through 12.

For all users:

MF=
specifies either the execute or list form of MSGDISP. If this parameter is not specified, the standard form of the macro is used.

L
specifies the list form of MSGDISP, which generates a parameter list that does not contain any executable instructions. The list can be used as input to and can be modified by the execute form of the macro

(E,*addr*)
specifies that the execute form of the macro and an existing parameter list be used.

addr
specifies an in-storage address of the parameter list. Specify either an RX-type address or a register in the range of 2 through 12.

TEST=
specifies whether the UCB is to be tested to determine if the device supports a message display before calling the message display SVC routine.

NO
specifies that the UCB be tested by the SVC routine.

YES
specifies that the UCB be tested before the SVC call.

Note: TEST=YES requires that the UCB mapping macro (IEFUCBOB) be included in the source code.

TXT=
specifies eight characters be shown in positions 1 through 8 of the display. The default for TXT is all blanks.

addr—RX-type address, A-type address, or (2−12)
specifies an in-storage address of an area containing the eight characters. For MF=L, only an A-type address can be specified.

'msgtxt'
specifies the eight characters as literals. Specify in apostrophes.

TXT2=
specifies eight alternate characters to be shown in positions 1 through 8 of the display. The default for TXT2 is all blanks.

addr—RX-type address, A-type address, or (2 – 12)
specifies an in-storage address of an area containing the eight characters. For MF = L, only an A-type address can be specified.

'altmsgtxt'
specifies the eight characters as literals. Specify in apostrophes.

VOL =
specifies the manner in which messages display, based on volume status.

STATIC
specifies that messages display without regard to volume status until the next message request processes, or until the next command initiates volume movement.

REMOVE
specifies that messages display until the current volume is demounted. This parameter is ignored if a volume is not mounted at the time the request is processed.

INSERT =
specifies that messages display until a volume is present, the tape is threaded, and the active/inactive switch is in the active position. This parameter is ignored if a volume is loaded and ready at the time the request is processed.

SWAP
specifies that messages always display as if FLASH = ALT was specified. The data from TXT and TXT2 display alternately and repeat until the current volume is demounted, at which time only TXT2 flashes until a new volume is loaded and ready. If no volume is mounted when this parameter processes, only TXT2 data flashes until a new volume is loaded and ready.

WAIT =
specifies when control is returned to the caller.

NO
specifies that control is returned before I/O is complete. I/O return codes are not returned, and I/O errors are recorded in the same manner as any permanent error by the error recovery procedure.

YES
specifies that control is returned after I/O is complete.

Return Codes from MSGDISP

When control is returned to the caller of MSGDISP, register 15 contains one of the following return codes:

Code	Meaning
00(00)	Successful completion.
04(04)	Device does not support MSGDISP.
08(08)	Unauthorized request (failed TESTAUTH for proper authority level) or invalid input parameters (including DCB or UCB). Reason Code (in Register 0): 01 Invalid Parameter 02 Invalid DCB or DEBCHK error 03 Environmental error 04 Authorization violation 05 Invalid UCB 06 Invalid request 11 Unsuccessful ESTAE macro call 12 Unsuccessful GETMAIN request
12(0C)	I/O error (I/O supervisor posted the request for an error) *Note: An I/O error occurs for load display if the drive display has a hardware failure.*

If you specified RESET UCBL for return codes 4 and 12, register 1 contains a pointer to the UCB in which the error was found.

Modifying MSGDISP for Other Languages

The messages produced by the 3480 drive hardware are available in several languages. For the software messages generated by MOUNT, DEMOUNT, and VERIFY parameters of the MSGDISP macro, all constants are kept in a CSECT (IGXMSG01) in virtual storage as part of the extended SVC routine IGX00030. The CSECT is in English; however, you can change the software-generated messages to another language. These changes are done using the AMASPZAP service aid program (see Figure 2 and Figure 3 on page 59). Some system components supply their own messages which will be used instead of the entries in CSECT IGXMSG01. Modifications to the message table in IGXMSG01 may not be used under these circumstances.

Note: Because some of the software messages consist of well-known single-character abbreviations (for example, M for mount), you might want to leave them in the English version.

```
//ZAPJOB JOB ,(USERID),MSGCLASS=A
//************************************************
//* SYS1.LPALIB(IGX00030) - CSECT: IGXMSG01      ***
//************************************************
//STEP1     EXEC PGM=AMASPZAP
//SYSPRINT  DD  SYSOUT=*
//SYSLIB    DD  DSN=SYS1.LPALIB,DISP=SHR
//SYSIN     DD  *
NAME IGX00030 IGXMSG01
VER    0020  E2C3D9E3,C3C8
REP    0020  C1C2C3C4,C5C6
```

Figure 2. Sample Message Display Language Modification

Offset In Hex	Value	Indicator Function
0	blank	reserved
1	(	RDY identifier
2	)	RDY identifier
3	M	MOUNT
4	blank	VERIFY
5	blank	reserved
6	blank	RESET
7	blank	GEN
8	blank	reserved
9	blank	reserved
A	D	DEMOUNT,DISP=D
B	K	DEMOUNT,DISP=K
C	R	DEMOUNT,DISP=R
D	blank	reserved
E	blank	reserved
F	blank	reserved
10	S	SL label
11	N	NL label (also BLP and LTM)
12	A	AL label
13	X	NSL label
14	?	Unknown label type
•		
•		
20-25	SCRTCH	SER/MSER default
26-2B	PRIVAT	SER/MSER default

Figure 3. Description of CSECT IGXMSG01

Chapter 6. Managing and Controlling Data Sets

The Data Facility Hierarchical Storage Manager (DFHSM) provides space management, backup, and recovery functions to manage data sets on a hierarchy of storage devices that have different costs of storing data, different amounts of data stored, and different speeds of accessing the data. This storage hierarchy includes direct access storage devices (DASD), mass storage devices, and magnetic tape devices. As a continuously running task under OS/VS2 MVS JES2 or JES3, DFHSM can automatically monitor storage devices.

Tape cartridges mounted on 3480 tape drives can be used as migrate and backup volumes. The migrate and backup versions of a data set can span more than one tape volume. The space manager can designate whether the migrate and backup versions made from data sets on a given primary volume must reside on tape.

Specifying Esoteric Unit Names

You cannot specify an esoteric unit name in a user unit table that consists of a mixture of 3480 and 3420 tape devices. Do not add 3480 tape devices to 3420 esoteric names you have already defined because DFHSM rejects each invalid unit. If DFHSM rejects all esoteric unit names, there is no user unit table. For more information about specifying esoteric unit names, see USERUNITTABLE parameter of the SETSYS command.

Recycling Tapes

If you regularly perform volume backup on primary volumes, data sets on those volumes are backed up many times. As this occurs, the percentage of valid data on a tape backup volume decreases as old backup versions are deleted from the volume.

The RECYCLE command can consolidate the valid data on tape backup volumes by moving the valid data sets from the tape volumes to tape spill backup volumes, enabling the tape volumes to be used again. When a volume is successfully recycled, it is deleted from DFHSM control or redefined as an unassigned backup volume. You can move valid data sets from a 3420 tape backup volume to a 3480 tape spill backup volume and vice versa.

Establishing or Changing the Values of DFHSM Control Parameters (SETSYS)

When you start DFHSM, a subset of DFHSM control parameters is established by defaults. You can override DFHSM defaults by specifying one or more SETSYS commands in the ARCCMD*xx* PARMLIB member used when you start DFHSM. You could then issue the SETSYS command with specific parameter values after DFHSM is started to change the current defaults until you restart DFHSM. See DFHSM Installation Verification Procedure in *Data Facility Hierarchical Storage Manager: Version 2 Release 3.0 Installation and Customization Guide* for an example of how to set up the ARCMD*xx* PARMLIB member.

If you do not specify the SETSYS command, DFHSM does not do any automatic space management or backup. Therefore, if you want to take advantage of the automatic functions of DFHSM, use the SETSYS command.

Do not confuse DFHSM defaults with the SETSYS command defaults. Except for certain values, there are no SETSYS command defaults. A SETSYS command has no required parameters, so, unless you indicate a specific parameter value for the SETSYS command, the DFHSM control parameter is the default.

The 3480 has changes to the parameters listed in the following sections.

Specifying the Tape Density When Allocating Scratch Tapes (DENSITY)

Explanation: DENSITY(2|3|4) is an optional parameter identifying which tape density to specify the first time DFHSM requests that a scratch tape be mounted.

If you specify a density of 2, 3, or 4 for the 3480 subsystem, DFHSM ignores it.

Specifying the Type of Unit for Mounting a Scratch Tape (UNITNAME)

Explanation:
UNITNAME(3400-3|3400-4|3400-5|3400-6|3400-9|3480|(*esoteric unit name*) is an optional parameter identifying the type of unit that should be specified the first time DFHSM requests that a scratch tape be mounted.

The types of units you can request for mounting the scratch tape are 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, or a name you specify in the USERUNITTABLE parameter of the SETSYS command. You specify 3400-9 when your 3480 subsystem simulates your 3420 tape drives. You specify 3480 when you use all the functions of the 3480 subsystem.

Abbreviations: The TSO abbreviation convention applies for UNITNAME. There are no additional abbreviations.

SETSYS Defaults: None.

DFHSM Defaults: If you do not specify this parameter on any SETSYS command, the DFHSM default is 3400-6.

Note: You cannot specify an esoteric unit name that represents DASD; it must represent tape.

DFHSM does not use UNITNAME when it requests that a scratch volume be mounted while continuing from another volume. Instead, DFHSM uses the same unit the volume was mounted on.

If you specify DENSITY and UNITNAME, the density must match the density for that type of unit. If you specify UNIT and do not specify DENSITY, DFHSM uses the highest density for the specified unit.

If you specify an esoteric unit name with the UNITNAME parameter, you must also have identified the esoteric unit name to DFHSM with the USERUNITTABLE parameter. You can specify the USERUNITTABLE parameter when you specify the UNITNAME parameter or you could have specified USERUNITTABLE with a previous SETSYS command during this startup.

Specifying Esoteric Tape Unit Name to DFHSM (USERUNITTABLE|NOUSERUNITTABLE)

Explanation: USERUNITTABLE(*unit:...*)|NOUSERUNITTABLE are mutually exclusive, optional parameters specifying esoteric tape unit names to DFHSM.

USERUNITTABLE specifies all the esoteric tape unit names to be identified to DFHSM. For *unit...*, specify the esoteric names of all the tape units you want to use. If you specify this parameter, you must specify at least one unit.

NOUSERUNITTABLE specifies that no esoteric tape unit names are to be identified to DFHSM. Any previously defined esoteric names are no longer in effect.

Abbreviations: The TSO abbreviation convention applies for USERUNITTABLE and NOUSERUNITTABLE. There are no additional abbreviations.

SETSYS Defaults: None.

DFHSM Defaults: If you do not specify this parameter on any SETSYS command, the DFHSM default is NOUSERUNITTABLE.

Note: You must make sure that the tape unit you specify can read and write any tape written by any other tape unit that belongs to the same esoteric group. DFHSM does not verify this for you. You must define all esoteric tape unit names during system I/O generation. Do not remove from a subsequent system I/O generation those esoteric unit names DFHSM already used during its tape processing. If you do, DFHSM cannot allocate the tape volume whose esoteric unit name you specified when you issued the ADDVOL command.

Because DFHSM rejects mixed esoteric unit names, do not change your 3420 tape esoteric unit names to include the 3480 tape devices even if your 3480 subsystem simulates a 3420 tape unit. If DFHSM rejects an esoteric unit name, it does not reject the rest of the user unit table. Each time you specify USERUNITTABLE, the valid esoteric tape unit names identified with this parameter replace any esoteric tape unit names identified with a previous SETSYS USERUNITTABLE command. In other words, the user unit table is rebuilt. However, if a GETMAIN failure occurred, the previous user unit table is still valid, because DFHSM does not replace the user unit table until it has verified all the entries in the new user unit table.

Adding or Changing the Volumes Managed by DFHSM (ADDVOL)

You issue the ADDVOL command to add new volumes to the list of volumes that DFHSM manages or owns. You also issue the ADDVOL command to change the attributes of a volume that DFHSM already manages or owns. You must issue the ADDVOL command for each volume you want DFHSM to manage or own. In addition, you must respecify primary and migration level 1 volumes again each time you start DFHSM.

Specifying the Type of Device (UNIT)

Explanation: **UNIT**(*unittype*) is a required parameter specifying the type of unit on which the volume can be mounted.

For *unittype*, substitute the type of unit on which the volume can be mounted. The valid types of units for the 3480 are: 3480 or a unit specified with the USERUNITTABLE parameter of the SETSYS command. You specify 3400-9 when the 3480 subsystem simulates your 3420 tape drives. You specify 3480 when you use all the functions of the 3480 subsystem.

Abbreviations: The TSO abbreviation convention applies for UNIT. There are no additional abbreviations.

Defaults: None.

Note: If you specify an esoteric unit name that does not exist in the user unit table, the command fails.

Specifying the Tape Density (DENSITY)

Explanation: DENSITY(2|3|4) is an optional parameter specifying the density of the tape backup volume.

If you specify a density of 2, 3, or 4 for the 3480 subsystem, DFHSM ignores it.

Chapter 7. Recording Error or Statistical Information

Using EREP Reports

Some of the maintenance of the 3480 subsystem relies on the records of errors and statistical information recorded by the processor. Editing, printing, and system exception reports are included in EREP.

The error records (OBR records) and statistics records (MDR records) in SYS1.LOGREC contain the 32 sense bytes that the 3480 control unit sends to the processor. An error code is included in the sense byte information to aid the service representative in locating the cause of the unit check. MDR and OBR records have the same format for full function and 3420 compatibility modes. Error records from different systems can be combined in the EREP system exception reports.

Using Error Recovery Procedures (ERP)

Error recovery procedures (ERPs) support the new sense byte structure of the 3480. The format of the error messages has been changed slightly.

With the 3480, most of the tape error recovery sequences are initiated by the control unit. For this reason, and because the control unit provides an error symptom code, ERP programs are simplified.

Using IMS/VS Logging

IMS/VS has not changed any code for the 3480 but you might want to use tape-write-immediate mode (DCB=OPTCD=W) for the log tape. (You should carefully evaluate the effect of this on performance.) You can continue to place log data on unbuffered tape drives.

IMS/VS Release 1.3 replaces the log tape with a disk log data set. The 3480 is an ideal device for archiving the IMS disk log.

Chapter 8. Restrictions to Using Sorting Applications (DFSORT)

The following information is the support for IBM DFSORT. For the 3480 subsystem, DFSORT always uses buffered mode for both reading and writing input and output tapes.

SORTWKnn DD statement

The SORTWKnn DD statements describe the characteristics of the data sets used as intermediate storage areas for records to be sorted; they also indicate the location of these data sets.

Note: The Sort program prior to Release 7.0 does not accept 3480 devices for the SORTWKnn files.

SORTOUT DD Statement

The SORTOUT DD statement describes the characteristics of the data set in which the sorted or merged records are to be placed, and indicates its location.

Note: If LABEL=RETPD is specified in the SORTOUT DD statement for a standard labeled tape, the DCB parameters must also be specified. If the DCB parameters are not specified, the tape can be opened twice. OPTCD=W should not be specified for a 3480 tape unit. If it is specified for a full function 3480 tape unit, the request is overridden. If it is specified for a 3480 operating in 3420 compatibility mode (specified in JCL as 3400-9), the request is not overridden, but performance might be reduced.

Appendix A. Data Control Block Symbolic Field Names

Data control block fields contain information that defines the data characteristics and device requirements for a data set. Each of the fields described show the values that result from specifying various options in the DCB macro instruction. These fields can be referred to by the problem program through the use of a DCBD macro instruction that creates a dummy control section (DSECT) for the data control block. Four-byte fields that contain addresses are aligned on fullword boundaries. If the problem program inserts an address into a field, the address must be inserted into the low-order three bytes of the field without changing the high-order byte.

The contents of some fields in the data control block depend on the device and access method being used. A separate description is provided when the contents of the field are not common to all device types and access methods.

Magnetic Tape Interface

Offset	Bytes and Alignment	Field Name	Description
16(10)	1	DCBTRTCH	Tape recording technique for 7-track tape

Code		
0010 0011	E	Even parity
0011 1011	T	BCD/EBCDIC translation
0001 0011	C	Data conversion
0010 1011	ET	Even parity and translation

Offset	Bytes and Alignment	Field Name	Description
17(11)	.1	DCBDEVT	Device type.
		1000 0001	2400 series magnetic tape unit (7-track or 9-track, for MVS/370 only)
		1000 0011	3400 series magnetic tape unit
		1000 0000	3480 Magnetic Tape Subsystem
18(12)	..1	DCBDEN	Tape density of magnetic tape units*

Code	7-track	9-track
0000 0011	200 BPI	Not applicable
0100 0011	556 BPI	Not applicable
1000 0011	800 BPI	800 BPI
1100 0011	Not applicable	1600 BPI
1101 0011	Not applicable	6250 BPI

* For 3480s operating in full function mode, the DCBDEN parameter is not applicable. For 3480s operating in 3420 compatibility mode, the value '1100 0011' indicates the data set is to be processed in tape-write-immediate mode.

Appendix B. Programming Messages and Codes

Return Code from the CHKPT Macro Instruction for the 3480

For further information about checkpoint/restart, see *Checkpoint/Restart User's Guide.*

Code (Hex)	Meaning
28	I/O error during SYNCDEV caused by a user data set.

ABEND Completion Codes Issued by Checkpoint/Restart for the 3480

Code (Hex)	Meaning
03F	(MVS/370 only) System abend code 03F indicates that a user data set on a 3480 subsystem could not be synchronized. An I/O error occurred while writing data from a previous channel program to the media (deferred-write error).
33F	(MVS/XA only) System abend code 33F indicates that a user data set on a 3480 subsystem could not be synchronized. An I/O error occurred while writing data from a previous channel program to the media (deferred-write error).

ABEND Return Codes

The ABEND return codes for the following messages reflect conditions that can occur with 3480 block count checking. See *MVS/Extended Architecture Message Library: System Messages* or *MVS/370 Message Library: System Messages* for the complete text of the messages and explanations of the messages and any other return codes associated with the message.

Message IEC022I 137-rc

Code (Hex)	Meaning
18	An I/O error occurred while the system was positioning a magnetic tape data set at the first data record or on a Read Block ID command used to establish the tape position.
38	An I/O error occurred during a Read Block ID command used to establish the tape position.

Message IEC023I 237-rc

Code (Hex)	Meaning
0C	The block count in the DCB does not match the actual count of blocks processed as determined by the position of the tape.

Message IEC145I 413-rc

Code (Hex)	Meaning
38	An I/O error occurred during a Read Block ID command used to establish the tape position.

Message IEC210I 214-rc

Code (Hex)	Meaning
0C	An I/O error occurred during a Read Block ID command used to establish the tape position.
10	The block count in the DCB does not match the actual count of blocks processed as determined by the position of the tape.

Message IEC218I 117-rc

Code (Hex)	Meaning
38	An I/O error occurred during a Read Block ID command used to establish the tape position.
3C	The block count in the DCB does not match the actual count of blocks processed as determined by the position of the tape.

Glossary

This glossary defines the special terms, abbreviations, and acronyms that are used in this publication. It does not include all terms previously established for the IBM System/370 or its operating systems; therefore, if you do not find the term you are looking for, refer to the index or to the *Vocabulary for Data Processing, Telecommunications, and Office Systems,* GC20-1699.

This glossary includes definitions from:

- The *American National Dictionary for Information Processing,* published by the Computer and Business Equipment Manufacturers Association. This material is reproduced from the American National Dictionary for Information Processing, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. **These definitions are identified by an asterisk (*).**

- The *ISO Vocabulary of Data Processing,* developed by the International Standards Organization, Technical Committee 97, Subcommittee 1. Definitions from published sections of this vocabulary are identified by the symbol (ISO) preceding the definition. Definitions from draft proposals and working papers under development by the ISO/TC97 vocabulary subcommittee are identified by the symbol "(TC97)", indicating that final agreement has not yet been reached among its participating members.

access method. A technique for moving data between processor storage and input/output devices.

archiving application. The retention of records, in machine-readable form, for historical purposes.

argument. (1) * (ISO) Any value of an independent variable.

automatic cartridge loader. A feature for the 3480 tape drive. This feature attaches to the front of the drive and allows both the automatic loading of premounted tape cartridges, and the manual loading of single tape cartridges.

backup-and-recovery application. The short-term retention of records to be used for restoring essential business and system files when vital data has been lost because of program or system errors or malfunctions.

basic sequential access method (BSAM). An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

beginning of tape. * (ISO) The location on a magnetic tape that indicates the beginning of the permissible recording area.

block. * A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps (IBG) and each block may contain one or more records.

BOT. Beginning of tape.

BSAM. Basic sequential access method

buffer. * A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another.

capacity. See media capacity.

CCW. Channel command word

channel. See data channel.

channel command. An instruction that directs a data channel, control unit, or device to perform an operation or set of operations.

channel command word (CCW). A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs data channel operations.

channel path. A communication path along which signals can be sent between a controlling computer and I/O devices.

channel path group. A group of identified channel paths over which the controlling computer and an I/O device can communicate.

channel status word (CSW). An area in storage that provides information about the termination of input/output operations.

command. See channel command.

contingent allegiance. A condition in which a device owes a response to a specific channel path because of a unit check.

conversion. The process of changing from one method of data processing to another or from one data processing system to another.

CSW. Channel status word.

data. * Any representations such as characters or analog quantities to which meaning is, or might be, assigned.

data base. (TC97) A set of data that is sufficient for a given purpose or for a given data processing system.

data channel. A device that connects a processor and main storage with I/O control units.

data control block (DCB). A control block used by access method routines in storing and retrieving data.

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.

DCB. Data control block.

DDR. Dynamic device reconfiguration.

data migration. (1)The moving of data from an online device to an offline or low-priority device, as determined by the system or as requested by the user. (2)The moving of data from an I/O device to another I/O device with different characteristics. Contrast with staging.

deferred unit check. A condition in which a device returns a unit check indication for the controlling computer, which pertains to an error an event that occurred asynchronously with the channel commands. The deferred unit check may not refer to the command that receives the indication.

disband group. To remove the grouping of a channel path group.

drive loaded. A condition of a tape drive in which a tape reel or cartridge inserted in the drive and the tape has been threaded to the beginning-of-tape position.

dump. (ISO) To write the contents of storage, or of a part of storage, usually from an internal storage to a external medium, for a specific purpose such as to allow other use of storage, as a safeguard against faults or errors, or in connection with debugging.

dynamic device reconfiguration (DDR). A facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

effective recording density. The number of user bytes per unit of length of the recording medium.

ERP. Error recovery procedures

error recovery procedures (ERP). Procedures designed to help isolate and, where possible, to recover from errors in equipment. The procedures are often used in conjunction with programs that record the statistics of machine malfunctions.

EXCP. Execute channel program.

extended contingent allegiance. A condition caused by a permanent buffered-write error in which the drive responds only to the channel path group from which the write command was received. The extended contingent allegiance continues until a controlling computer in the channel path group retrieves the unwritten data from the buffer or issues a tape motion command.

file. * (ISO) A set of related records, treated as a unit, for example, in stock control, a file could consist of a set of invoices.

format. * (ISO) The arrangement or layout of data on a data medium.

interchange application. The preparation of tapes for use on other systems or devices, either local or remote, or the use of tape data prepared by another system.

JCL (job control language). * A problem-oriented language designed to express statements in a job

that are used to identify the job or describe its requirements to an operating system.

journalizing. Recording transactions against a data set so that the data set can be reconstructed by applying transactions in the journal against a previous version of the data set.

load point. The beginning of the recording area on magnetic tape.

logical end-of-tape. A point on the tape where written data normally ends.

magnetic recording. * (ISO) A technique of storing data by selectively magnetizing portions of a magnetizable material.

magnetic tape. (TC97) A tape with a magnetizable surface layer on which data can be stored by magnetic recording.

magnetic tape drive. (ISO) A mechanism for moving magnetic tape and controlling its movement.

media capacity. The amount of data that can be contained on a storage medium, such as tape, expressed in bytes of data.

migration. See conversion.

OBR. Outboard recorder.

outboard recorder (OBR). A feature that records pertinent data on the system recorder file when an unrecoverable I/O error occurs.

physical end-of-tape. A point on the tape beyond which the tape is not permitted to move.

primed. Pertaining to a condition of a tape drive when the controlling computer has addressed the drive but the drive was not in a ready state.

processing application. The completion of a systematic sequence of operations performed on data to accomplish a specific purpose.

QSAM. Queued sequential access method.

queued sequential access method (QSAM). An extended version of the basic sequential access method (BSAM). When this method is used, a queue is formed of input data blocks that are awaiting processing or output data blocks that have been processed and are awaiting transfer to auxiliary storage or to an output device.

quiesce. To bring a device or system to a halt by a rejection of new requests for work.

read-ahead queue. In ACF/TCAM, an area of main storage from which an application program obtains work units in advance of their being requested by the application.

record. * (ISO) A collection of related data or words, treated as a unit; for example, in stock control, each invoice could constitute one record.

synchronization. The process of coordinating the activities of the controlling computer and the 3480 subsystem to obtain the condition in which the buffer is empty and the tape is in the correct position for the next operation.

tape cartridge. A container holding magnetic tape that can be processed without separating it from the container.

task I/O table (TIOT). A control block constructed by job management to provide I/O support routines with pointers to JFCBs and allocated devices.

TIOT. Task I/O table

volume. (1)(ISO) A certain portion of data, together with its data carrier, that can be handled conveniently as a unit. (2)(ISO) A data carrier that is mounted and demounted as a unit; for example, a reel of magnetic tape, a disk pack.

Index

D

E

F

G

H

I

S

T

U

V

W

Numerics

IBM 3480 Magnetic Tape Subsystem
User's Reference

READER'S
COMMENT
FORM

Order No. GC35-0099-3

This manual is part of a library that serves as a reference source for systems 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: Copies of IBM publications are not stocked at the location to which this form is addressed. Please 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.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

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 them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY THE ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85744

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

IBM 3480 Magnetic Tape Subsystem, User's Reference (File No. S370-05/4300-05) Printed in U.S.A. GC35-0099-3

IBM 3480 Magnetic Tape Subsystem
User's Reference

READER'S
COMMENT
FORM

Order No. GC35-0099-3

This manual is part of a library that serves as a reference source for systems 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: Copies of IBM publications are not stocked at the location to which this form is addressed. Please 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.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

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 them directly to the address in the Edition Notice on the back of the front cover or title page.)

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY THE ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85744

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

IBM 3480 Magnetic Tape Subsystem, User's Reference (File No. S370-05/4300-05) Printed in U.S.A. GC35-0099-3

IBM 3480 Magnetic Tape Subsystem
User's Reference

READER'S
COMMENT
FORM

Order No. GC35-0099-3

This manual is part of a library that serves as a reference source for systems 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: Copies of IBM publications are not stocked at the location to which this form is addressed. Please 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.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

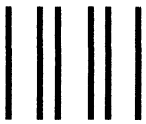
Number of latest Newsletter associated with this publication: _____

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 them directly to the address in the Edition Notice on the back of the front cover or title page.)

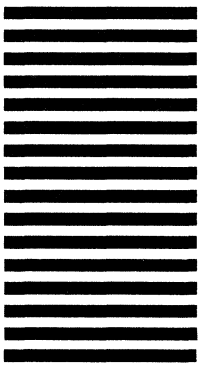
Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY THE ADDRESSEE

International Business Machines Corporation
Department 61C
P.O. Box 27155
Tucson, AZ, USA 85744

Fold and Tape | Please Do Not Staple | Fold and Tape

IBM®

IBM 3480 Magnetic Tape Subsystem, User's Reference (File No. S370-05/4300-05) Printed in U.S.A. GC35-0099-3

IBM 3480 Magnetic Tape Subsystem
User's Reference

GC35-0099-3

File Number S370-05/4300-05
Printed in U.S.A.

GC35-0099-03