Prime®

MAGNET User's Guide

Revision 21.0

DOC10156-1LA

MAGNET User's Guide

First Edition

by

Ann Sutton

Updated for Revision 22.0

by

Ian K. Turnbull

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 22.0 (Rev. 22.0).

Prime Computer, Inc.
Prime Park
Natick, Massachusetts 01760

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer, Inc. Prime Computer, Inc., assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

This document was prepared in the United Kingdom by Technical Publications Department, International Systems Marketing and Development, Willen Lake, Milton Keynes, MK15 0DB, United Kingdom.

PRINTING HISTORY

First Edition (DOC10156-1LA) July 1987 for Master Disk Revision Level 21.0.

Update 1 (UPD10156-11A) October 1988 for Master Disk Revision Level 22.0.

CREDITS

Editorial: John Wells
Illustration: Kevin Maguire
Document Preparation: Kevin Maguire

HOW TO ORDER TECHNICAL DOCUMENTS

To order copies of documents, or to obtain a catalog and price list:

United States Customers	International
Call Prime Telemarketing, toll free, at 1-800-343-2533, Monday through Friday, 8:30 a.m. to 5:00 p.m. (EST).	Contact your local Prime subsidiary or distributor.

CUSTOMER SUPPORT

Prime provides the following toll-free numbers for customers in the United States needing service:

1-800-322-2838 (within Massachusetts)	1-800-541-8888 (within Alaska)
1-800-343-2320 (within other states)	1-800-651-1313 (within Hawaii)

For other locations, contact your Prime representative.

SURVEYS AND CORRESPONDENCE

Please comment on this manual using the Reader Response Form provided in the back of this book. Address any additional comments on this or other Prime documents to:

Technical Publications Department
Prime Computer, Inc.
500 Old Connecticut Path
Framingham, MA 01701

Contents

APPENDICES

About This Book

This MAGNET User's Guide describes the Rev.22.0 MAGNET subsystem. It is written for new or experienced MAGNET users, and for new or experienced users of magnetic tapes.

This guide describes the features of MAGNET, and explains how to use them. It also describes the characteristics of magnetic tapes and tape devices, and explains how to use PRIMOS magnetic tape commands. More magnetic tape information for the System Administrator and operator can be found in the System Administrator's Guide Volume I (DOC10131-2LA), Volume II (DOC10132-2LA), and Volume III (DOC10133-2LA), the Operator's System Overview (DOC9298-3LA), and the Data Backup and Recovery Guide (DOC10129-1LA).

HOW TO USE THIS BOOK

This book is in two parts, and contains four appendices, as follows:

PART I - PRELIMINARY INFORMATION

This part introduces you to MAGNET and provides the information that you need before you start a MAGNET session. It contains three chapters, as follows:

Chapter 1 - Introduction

This chapter briefly introduces MAGNET, explains the relationship between PRIMOS® and MAGNET, and summarizes the steps involved in a typical MAGNET session.

Chapter 2 - Tape Control

This chapter describes how to control tape drives, using the PRIMOS ASSIGN, UNASSIGN, and STATUS commands. Read this chapter if you are not familiar with these PRIMOS commands.

Chapter 3 - Initializing Tapes

This chapter describes how to initialize magnetic tapes with the PRIMOS LABEL command.

PART II - USING MAGNET

This part describes how to invoke MAGNET, and explains the function and syntax of all MAGNET's subcommands and options. It also provides tables for easy reference to subcommands or options. This part contains four chapters, as follows:

Chapter 4 - Invoking MAGNET

This chapter explains how to invoke MAGNET and describes the format of the MAGNET subcommand line.

Chapter 5 - Subcommand Syntax

This chapter provides a reference table of subcommands, followed by detailed descriptions of the function and syntax of each subcommand.

Chapter 6 - Subcommand Options

This chapter describes in detail all the options that you can use with MAGNET subcommands. It also contains a reference table which, for each option, describes the option type, the subcommands that you can use it with, and the function of the option.

Chapter 7 - Sample MAGNET Sessions

This chapter provides three examples of typical MAGNET sessions. Each step of each example is fully explained. This chapter also shows how you can create a CPL file to run a MAGNET session.

APPENDICES

There are four appendices, as follows:

Appendix A - Magnetic Tape Record Formats

This is primarily for new users of magnetic tapes. It briefly describes the physical characteristics of magnetic tapes, and describes the format of IBM and ANSI tape records.

Appendix B - Magnetic Tape Label Formats

This appendix describes the format of IBM and ANSI tape labels, for reference.

Appendix C - Character Set and Translation Tables

This appendix contains translation tables, and ASCII, EBCDIC, and BCD character set tables.

Appendix D - MAGNET Messages

This appendix lists each message that can be output by the MAGNET subsystem, and describes remedial action, where appropriate.

PRIME DOCUMENTATION CONVENTIONS

The following conventions are used in command formats, statement formats, and examples throughout this document. Examples illustrate the uses of these commands and statements in typical applications. Terminal input may be entered in either uppercase or lowercase letters.

Convention	Explanation	Example
UPPERCASE	In command or subcommand formats, words in uppercase letters indicate the actual names of commands, statements, and keywords. You can enter these in either uppercase or lowercase letters.	DECLARE
lowercase	In command formats, words in lowercase letters indicate items for which you must substitute a suitable value.	CLOSE object
Abbreviations	If a command or statement has an abbreviation, it is indicated by underlining.	STATUS DEVICE
Underlining in examples	In examples, user input is underlined, but system prompts and output are not.	OK, MAGNET [MAGNET Rev. 22.0..] >
Brackets [ ]	Brackets enclose a list of two or more optional items. Choose none, one, or more than one of these items.	MAGNET [-SILENT -OVERWRITE]
Braces { }	Braces enclose a list of items. Choose only one of these items.	MAGNET { -USER -OPERATOR }
Hyphen -	Wherever a hyphen appears as the first letter of an option, it is a required part of that option.	MAGNET -OPR

Ellipsis ...	An ellipsis indicates that the preceding item may be repeated.	option1[,option2]..

PART I

Preliminary Information

1
Introduction

MAGNET is a PRIMOS subsystem that enables you to transfer data by magnetic tape from a non-Prime operating system to PRIMOS, and vice versa.

You can use MAGNET to

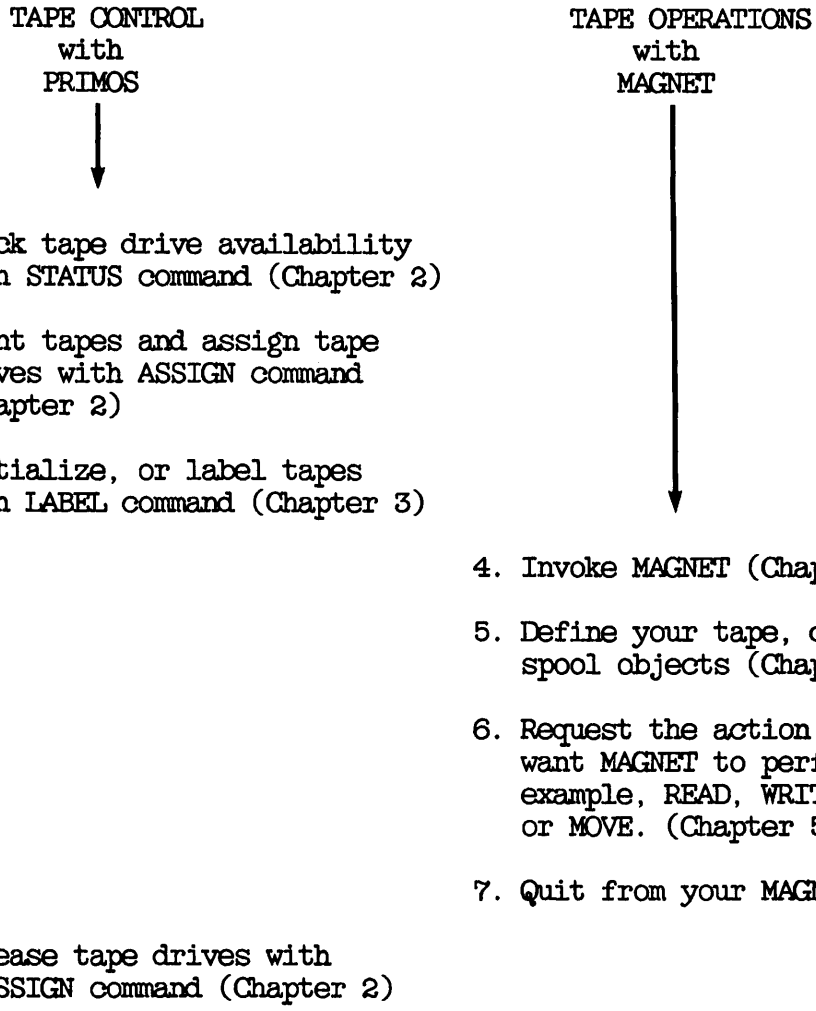
- Transfer data between disk, tape, or spooler devices at multiple destinations
- Position tapes
- Copy tapes
- Translate various character sets

The MAGNET subsystem supports variable-length records -- IBM, ANSI, and Prime -- and fixed-length records. MAGNET also supports tape operations within Batch mode, where operator intervention is expected or needed.

Because MAGNET is a PRIMOS subsystem, the commands that you use within it are called subcommands.

USING MAGNET

A MAGNET session is conducted within general magnetic tape control. You use PRIMOS commands to control tape drives and initialize tapes, and the MAGNET subsystem for detailed tape operations. The relationship between PRIMOS and MAGNET is shown in Figure 1-1, which summarizes the basic steps in a MAGNET session.



Basic Steps in a MAGNET Session
Figure 1-1

Before you use MAGNET, you need to know how to control tape drives with the PRIMOS commands STATUS, ASSIGN, and UNASSIGN: these are described in Chapter 2, TAPE CONTROL.

If you need to initialize your tape with the LABEL command, you must do so before starting your MAGNET session. The LABEL command is described in Chapter 3, INITIALIZING TAPES.

The way in which you invoke the MAGNET subsystem and use its subcommands is explained throughout Part II of this manual.

Chapter 7, SAMPLE MAGNET SESSIONS, provides three examples of MAGNET sessions and describes each example. Chapter 7 also explains how you can write and store an entire MAGNET session using a CPL program or COMINPUT (command input) file.

2
Tape Control

This chapter describes how to control tape drives using the STATUS, ASSIGN, and UNASSIGN commands: you need to know how to use these PRIMOS commands before you use MAGNET. If you are familiar with these commands, turn to Chapter 3, INITIALIZING TAPES, or Chapter 4, INVOKING MAGNET.

The three PRIMOS commands that you can use to control tape drives are

- STATUS -- which allows you to check the availability of tape drives
- ASSIGN -- which allows you to control a physical or logical tape drive, or to request operator assistance
- UNASSIGN -- which enables you to relinquish control of a tape drive

▶ STATUS

The STATUS command with the argument DEVICE, allows you to see which tape drives are in use. The command displays physical and logical device numbers for any assigned magnetic tape drives. Device numbers for tape drives that are not assigned are not included in the display. The format of the STATUS command using the DEVICE argument is:

```
STATUS DEVICE
```

A typical display might be:

```
OK, STATUS DEVICE

Device  User name                    Usrnum  Ldevice
MT0     SMITH                         1      MT0
MT1     JONES                        13      MT2
```

where:

Device -- is the physical device number.

User name -- is the login name of the user to whom the device is assigned.

Usrnum -- is that device user's number.

Ldevice -- is the logical device number being used for the drive.

▶ ASSIGN

The ASSIGN command gives you control of a specific magnetic tape drive. You can either specify a drive by device number (for example, MT0) or request any drive that meets your specified requirements: you specify requirements as options on the ASSIGN command line. (See list below.) ASSIGN also allows you to make requests to the operator; for example, to remove the write-enable ring, or to mount a tape.

Note that you cannot have more than one 60Mb cartridge tape drive assigned at a time.

The format of the ASSIGN command is:

```
{ ASSIGN }     { MTpdn [-ALIAS]      }     [options]
{ AS     }     { MTX   -ALIAS MTldn  }
```

where pdn is a physical device number, and ldn is a logical device number. Descriptions of the arguments and options follow.

Argument	Description
MTpdn	Identifies the magnetic tape (MT) unit number specified by pdn, in the range 0-7 inclusive. pdn is the physical device number assigned to each drive at system startup.
MTX	Assigns any available drive: must be accompanied by the -ALIAS MTldn option, which assigns a number (alias) to the drive for reference. The drive that is assigned depends on the other options that you specify on the command line. (Described below.)

Option	Description
-7TRK -9TRK	Indicates 7-track or 9-track tape drive; default is 9-track. Requires operator intervention. These options are usually used with the -MTX option.
-ALIAS MTldn	Specifies a drive with a logical device number ldn in the range 0-7 inclusive. The ldn is a user-defined number that is assigned to one physical drive unit. When you use this option with the MTpdn argument, ldn is mapped onto pdn in subsequent magnetic tape operations. This option is particularly useful in programs that run from command files, or in CPL programs. It allows the programs to refer to logical device numbers that remain constant, irrespective of the physical device numbers which vary according to device availability.
-DENSITY n	Requests a tape density setting, and may require operator intervention. The value of n specifies the tape density in bpi (bits per inch). Valid values for n are 800, 1600, 3200, and 6250.

Most drives can accept 800 and 1600 bpi settings. Streamer drives use 1600 and 3200 bpi. This option sets density automatically on drives that support 6250 bpi, but requires the operator to set density manually on older drives. If you do not specify -DENSITY, it defaults to 1600 bpi.

-MOUNT — Instructs the operator to mount a new tape reel on an already assigned tape drive. This option is usually used with the -TPID option.

-RETENSION — Only used with cartridge tape drives. It causes the tape to be fast-forwarded to the end of the tape, and then rewound to the beginning of the tape. This stabilizes the tape-to-head pressure and stacks the tape evenly on the reel.

-RINGOFF — Instructs the operator to remove the write ring from the tape, so that it can be read, but not written to.

-RINGON — Instructs the operator to place the write ring on the tape, so that it can be both read and written to.

-SPEED { 25 | 100 } — Selects speed for a streamer tape drive (in inches-per-second) running at 1600 bpi. The default is 25 ips. This option is ignored on other drives, including streamers that run at 3200 bpi: a speed of 50 ips is automatically set for streamers that run at 3200 bpi. See Table 2-1 for a list of speed and density settings.

-TPID id — Requests the operator to mount a reel of tape that you identify by a tape id: the id can be a maximum of eight characters. An id is a tape identifier that describes a reel of tape, and/or type of tape drive. Identifiers cannot begin with a hyphen (-), because this is a reserved character that indicates the next control argument on the ASSIGN command line. Identifiers that begin with a hyphen, or that contain other reserved characters (commas, spaces, and /*) must be placed inside quotation marks.

-WAIT — Indicates that you can wait until the drive that you requested is available.

Table 2-1
Recommended Speed and Density Settings
for Streamer Tapes

Utility	Settings Speed (ips)	Settings Density (bpi)	Usage
BRMS (BACKUP, ARCHIVE)	50	3200	For optimum capacity and speed. (Tapes not compatible with ANSI or IBM.)
BRMS TRANSPORT	25	1600	For transporting data to sites that do not support 3200 bpi.
	50	3200	For transporting data to sites that support 3200 bpi (optimum capacity and speed.)
MAGSAV MAGRST	50	3200	For optimum capacity and speed.
MAGNET	25	1600	For transferring data to sites that do not support 3200 bpi.
	50	3200	For transferring data to sites that support 3200 bpi format.
PHYSAV/ PHYRST	100	1600	For optimum speed.
PHYSAV/ PHYRST	50	3200	For optimum tape capacity.

How to Assign Tape Drives

You can assign magnetic tape drives in one of three ways:

- By physical device number (pdn), for example

  ASSIGN MTpdn [options]

- By logical device number (ldn), for example

  ASSIGN MTX -ALIAS MTldn

- By logical device number and tape characteristics, for example

  ASSIGN MTX -ALIAS MTldn -options

By Physical Device Number: When you assign a tape drive by a physical device number, you request that particular tape drive. If the drive is unavailable, the option -WAIT queues your request. In the following example, you assign magnetic tape drive MT1, where 1 is the physical device number. This is the default assignment, and does not usually require operator intervention.

```
OK, ASSIGN MT1
Device MT1 assigned.
OK,
```

When you assign a physical device, you can specify an alias: this is described in the following section. If you do not request an ldn alias, the default logical device number is the same as the physical device number of the drive. However, you still request a particular drive. In the following example, you specify physical device MT6 to be logical device MT2.

```
OK, ASSIGN MT6 -ALIAS MT2
Device MT6 assigned.
OK,
```

To display the physical-to-logical number correspondence, invoke the STATUS DEVICE command.

By Logical Device Number: When you assign a tape drive by a logical device number, you map the ldn onto the pdn.

If you use the ASSIGN command in the format

```
ASSIGN MTpdn -ALIAS MTldn
```

you specify that the drive identified by the pdn is to be referred to by the specified ldn.

For example:

```
OK, ASSIGN MT4 -ALIAS MT0
Device MT4 assigned.
OK,
```

In this example, until unassigned, the drive MT4 would be referred to as logical drive MT0. Note that the physical, not the logical, device number is returned in the message that follows the ASSIGN command. To display the physical-to-logical number correspondence, you can use the STATUS command, described at the beginning of this chapter.

When you wish to assign a logical device number to any available drive, use the command format

```
ASSIGN MTX -ALIAS MTldn
```

For example:

```
OK, ASSIGN MTX -ALIAS MT4
Device MT2 assigned.
OK,
```

would request the operator to assign any available drive as logical device 4. In this example, the operator responds to the request, and ASSIGN sends a message to your terminal to indicate that drive 2 has been assigned.

Note

Once you have specified a number as an ldn, you cannot assign another drive by that number. Just as each pdn refers to a unique drive, so must each ldn refer to one drive only. For example, if BOB assigned MT1 with an alias of MT0, he could not then assign MT0 as itself. For example:

```
OK, STAT DEV

Device  User name                              Usrnum  Ldevice


OK, ASSIGN MT1 -ALIAS MT0
Device MT1 assigned.
OK, ASSIGN MT0
The device is in use.  MT0 (asnmt$)
```

To assign drive MT0 in these circumstances, BOB would also have to give the drive an alias. For example:

```
OK, ASSIGN MT0 -ALIAS MT2
Device MT0 assigned.
OK, STAT DEV

Device  User name                              Usrnum  Ldevice
MT0     BOB                                    15      MT2
MT1     BOB                                    15      MT0
```

By Logical Device Number and Tape Characteristics: When you assign a tape drive by a logical device number and tape characteristics, you ask for a drive that can handle a particular type of tape (for example, a 9-track tape at 6250 bpi). In this instance you would give the drive a logical alias, for example

```
OK, ASSIGN MTX -ALIAS MT0 -TPID SYS -9TRK -RINGON -DENSITY 6250
```

specifies that you wish to assign a drive with the following characteristics: 9-track, read and write (no protect), and 6250 bpi. It states that you wish to mount tape SYS on the drive if a drive is assigned, and MT0 is the number you will use to refer to the assigned drive. All subsequent tape operations of the assigned drive would use MT0 as the unit number. If for any reason no drive were assignable, you would receive an error message.

Mounting Tapes

It is sometimes necessary to remove one tape from a tape drive, and replace it with another. To do this, you use the -MOUNT option: the tape drive must be assigned before you specify -MOUNT.

When you first assign the tape drive, you can use the -TPID option to specify which tape is to be placed on the drive. Subsequently, if you want to mount a different tape on the same drive, you can issue the ASSIGN command with the -MOUNT and -TPID options. This is shown in the following example.

Suppose that you assign logical drive 7 as follows

```
OK, ASSIGN MT0 -ALIAS MT7 -DENSITY 800 -TPID GRADES
```

Subsequently, to mount a different tape, called EXAMS, on the same drive, you would issue the command

```
OK, ASSIGN -ALIAS MT7 -MOUNT -TPID EXAMS
```

In this example, the operator would receive a message at the supervisor terminal, indicating that you want tape EXAMS to be mounted. The operator would respond to this request by using the REPLY command, and you would receive a message to indicate whether the mount operation was successful.

Command Messages

When you use the ASSIGN command you receive messages that report errors and status. These messages are described below, in alphabetical order.

- Bad parameter. (asnmt$)

Indicates an ASSIGN syntax error; your input is invalid. Check your command line and issue the command again.

- Device MTpdn assigned.

Your tape drive has been assigned. (Notice that this message gives the physical device number.)

- Device not available. Magtape assignment request aborted (asnmt$)

The operator cannot satisfy your assignment request. Check your command line options, and if necessary, discuss your request with the

operator.

- No Magtape assignments permitted. (asnmt$)

The operator currently cannot accept magnetic tape assignments. Wait until assignments are permitted.

- The device is in use. (asnmt$)

Another user has assigned the specified device.

- Use of 'MTX' argument not permitted at system console. (asnmt$)

The MTX argument requires a response from the supervisor terminal, and no further work can be done at the user terminal until the user receives this response. For this reason, users at the supervisor terminal cannot send themselves ASSIGN requests that involve the MTX argument.

▶ UNASSIGN

The UNASSIGN command releases a tape drive that you have previously assigned with the ASSIGN command. When you complete a magnetic tape operation, it is good practice to release the tape drive for general use as soon as possible.

The command line format for UNASSIGN is

```
{ UNASSIGN }    { MTpdn          }    [-UNLOAD]
{ UN       }    { -ALIAS MTldn   }
```

You can thus release a drive by specifying either the physical device number pdn or the logical device number ldn.

The -UNLOAD option rewinds the tape and places it offline, that is, it is unloaded. Note that the tape is not unloaded until it is fully rewound. The -UNLOAD option is not available on all drives or controllers. Find out from either your System Administrator or operator if you can use -UNLOAD at your installation.

How to Release Tape Drives

A tape drive can only be released by

- The user who assigned the drive
- The system operator

The system operator can release any drive by using the pdn argument, but can only use the -ALIAS ldn option if that operator assigned, and hence owns, the drive.

If an operator releases your tape drive, no message is displayed at your terminal. If you subsequently attempt to release the same device, an error message is displayed.

The following examples illustrate how you can use the UNASSIGN command.

Suppose that you assign the tape drives MT1 and MT0 as follows

```
OK, ASSIGN MT1 -ALIAS MT2
Device MT1 assigned.
OK, ASSIGN MTX -ALIAS MT0
Device MT0 assigned.
OK,
```

You could release physical drive MT1 by specifying either the pdn or the ldn on the UNASSIGN command line

```
OK, UNASSIGN MT1
```

or

```
OK, UNASSIGN -ALIAS MT2
```

If the command were successful, the message, "Device released" would be printed.

In this example, the pdn and ldn of MT0 are the same, so to release drive MT0 you would issue one of the following commands

```
OK, UNASSIGN -ALIAS MT0
```

or

```
OK, UNASSIGN MT0
```

Command Messages

The UNASSIGN command generates messages that report either errors or status. These messages are described below, in alphabetical order.

- Bad parameter. (usnmt$)

Indicates an UNASSIGN syntax error; your input is invalid. Check your command line, and issue the command again.

- Device not assigned. MTn (usnmt$)

The device you are attempting to release has not yet been assigned. Check that you have specified the correct device in your command line.

- Device released.

The device you had assigned is released.

3
Initializing Tapes

This chapter describes how to initialize magnetic tapes with the PRIMOS LABEL command. LABEL writes IBM (9-track EBCDIC or 7-track BCD), ANSI (9-track ASCII), or Prime nonstandard Level 1 volume labels followed by a dummy HDR1 label and two file markers. You can also use LABEL to read existing VOL1 and HDR1 labels. You have a choice of two ANSI standards for writing labels: X3.27-1978 and X3.27-1987. IBM labels are written in accordance with IBM's specifications (IBM manual GC28-6680-5).

The LABEL command also detects BRMS (Backup/Recovery Management Service) labeled tapes. You cannot read these tapes, but you can overwrite them using the -OVERWRITE option. There are four types of BRMS tapes: ARCHIVE, BACKUP, TRANSPORT, and free. LABEL overwrites free BRMS tapes without the -OVERWRITE option. Use the ARCHIVE_RELEASE, BACKUP_RELEASE, or TRANSPORT_RELEASE command to convert BRMS tapes of type ARCHIVE, BACKUP, or TRANSPORT to free tapes.

For detailed information about the format of magnetic tape labels, see Appendix B, MAGNETIC TAPE LABEL FORMATS.

▶ LABEL

The command line format for LABEL is:

```
LABEL MTn [-TYPE type] [ { -VOLSER } volume-id] [-OWNER owner]
                       { -VOLID  }
                       { -VOLUME }
      [-ACCESS access] [-option(s)]
```

Descriptions of the parameters and options follow.

Parameter	Description
MTn	The tape drive where the tape that you wish to label is located. n is a number 0 through 7. This keyword is mandatory and must be the first on the command line.
type	Specifies the type of tape that you are working with. You can specify one of five types
	-TYPE Standard_1 ANSI X3.27-1978 standard labels S1
	-TYPE ANSI87 ANSI X3.27-1987 standard labels
	-TYPE PRIME PRIME ASCII P (this is the default) ANSI A
	-TYPE B seven-track BCD (IBM)
	-TYPE E nine-track EBCDIC (IBM)
volume-id	A string that contains 1 through 6 characters, that uniquely identifies this tape reel. If you specify fewer than six characters, they are blank-padded to the right. You can substitute the keywords -VOLUME, -VOL, or -VOLSER for the keyword -VOLID.
owner	Identifies the owner of the tape. It is a string of 1 through 14 characters for ANSI labels, or 1 through 10 characters for IBM labels. If you specify fewer than 14 (or 10) characters, they are blank-padded to the right. If this keyword is omitted, the default is the user's login name. You can substitute the keyword -OWN for the keyword -OWNER.

access	A single character that defines access to this tape. The keyword ACCESS is not used by Prime software but is included for completeness. If it is omitted, it is left blank on ANSI labels. ACCESS is ignored for IBM labels.

Option	Description
-PARITY	Specifies either EVEN or ODD parity for BCD labels only (default = ODD).
-INIT	Necessary keyword for tapes previously unformatted. Also necessary for overwriting labeled tapes.
-OVERWRITE	Forces BRMS tapes to be overwritten.
-HELP	Displays information about the LABEL command.

To read existing labels, type the command:

```
LABEL MTn [-TYPE type]
```

To write labels, type the command:

```
LABEL MTn [-TYPE type] -VOLID volume-id [-OWNER owner]
          [-ACCESS access] [-INIT]
```

For example, the following command would create a scratch tape with an ANSI87 label, and volume serial ID DRB001, on device MT0.

```
OK, LABEL MT0 -TYPE ANSI87 -VOLID DRB001
```

The following example shows how you would use LABEL to write an ANSI VOL1 label on a new tape located on logical device number 3. The owner of the tape in this example is ROBINS, the volume serial ID is SP573, and the access is set to X:

```
OK, LABEL MT3 -TYPE A -VOLSER SP573  -OWNER ROBINS -ACCESS X -INIT
```

The next example shows how the command line is used to read 9-track IBM labels that already exist on the tape, which is located on logical device number 6:

```
OK, LABEL MT6 -TYPE E
```

Command Errors

When the LABEL command writes a label, the message "TAPE LABEL WAS WRITTEN SUCCESSFULLY" is displayed. On read operations, LABEL displays the volume and owner ID's, creation date, and access (for ANSI tapes only).

If you use the LABEL command incorrectly, a warning message or an error message is displayed. Errors are either the result of bad syntax in the LABEL command or a PRIMOS magnetic tape I/O error.

Warning Messages

The following LABEL command warning messages are listed and explained in alphabetical order.

- ***WARNING***
  ARCHIVE tapes cannot be overwritten. If you wish to write to this tape invoke LABEL with the -OVERWRITE option.
  LABEL will abort.

  You have tried to overwrite an ARCHIVE tape. You must either use the -OVERWRITE option when you invoke LABEL, or first release the ARCHIVE tape, using the ARCHIVE_RELEASE command (described in the Data Backup and Recovery Guide). This labels the tape as free. You can write to free tapes without using the -OVERWRITE option.

- ***WARNING***
  BACKUP tapes cannot be overwritten. If you wish to write to this tape invoke LABEL with the -OVERWRITE option.
  LABEL will abort.

  You have tried to overwrite a BACKUP tape. To overwrite a BACKUP tape, use the -OVERWRITE option when you invoke LABEL. Alternatively, you can release the tape using the BACKUP_RELEASE command (described in the Data Backup and Recovery Guide). This labels the tape as free. You can write to free tapes without using the -OVERWRITE option.

- ***WARNING***
  BRMS tapes cannot be read.
  LABEL will abort.

  You cannot read BRMS-labelled tapes. Use the ARCHIVE_RESTORE, BACKUP_RESTORE, TRANSPORT_RESTORE, or LIST_TAPE command to read BRMS-labelled tapes. (See the Data Backup and Recovery Guide.)

- ***WARNING***
  TRANSPORT tapes cannot be overwritten. If you wish to write to this tape invoke LABEL with the -OVERWRITE option.
  LABEL will abort.

  You have tried to overwrite a TRANSPORT tape. To overwrite a TRANSPORT tape, use the -OVERWRITE option when you invoke LABEL. Alternatively, you can release the tape using the TRANSPORT_RELEASE command (described in the Data Backup and Recovery Guide). This labels the tape as free. You can write to free tapes without using the -OVERWRITE option.

Error Messages

The following LABEL command error messages are listed in alphabetical order.

- -ACCESS values are ignored for type B and E labels.

  This is a warning only; Access values are ignored for IBM labels.

- A label block on the tape is not 80 characters long.

  The first record on the tape is not a valid VOL1 label.

- A VOL1 label already exists on this tape.

  Use -INIT if you wish to overwrite the existing VOL1 labels.

- An error has occurred while allocating dynamic storage.

  LABEL cannot allocate dynamic memory for internal buffers. See your System Administrator.

- An error has occurred while translating.

  Internal LABEL error.

- An error has occurred while writing to the tape.

  You may have a bad tape, or the tape may be write-locked. Clean the tape, check that it has a write-enable ring in place, or rewind the tape.

- An error has occurred while freeing dynamic storage.

  LABEL cannot free dynamic memory that is assigned to the internal buffers. See your System Administrator.

- An error occurred while reading from the tape.

  The tape drive is not ready or online, or an uncorrectable error was detected while the tape was read.

- An error occurred while rewinding the tape.

  LABEL could not rewind the tape. Check that it is online and ready.

- An indeterminate i/o error occurred.

  A bad tape is suspected. Clean your tape, or the read/write heads on your tape drive, and try the operation again.

- Error while analyzing the -ACCESS command line option.

  You should specify access with a single character.

- Error while analyzing the -OWNER command line option.

  The owner ID can be no longer than 14 characters for ANSI labels.

- Error while analyzing the -PARITY command line option.

  You must set parity to either even or odd.

- Error while analyzing the -TYPE command line option.

  Label types must be one of the characters S1, A, P, B, E, or ANSI87.

- Error while analyzing the -VOLSER command line option.

  The volume id cannot be longer than six characters.

- Error while analyzing the logical device number.

  The logical device number must be in the range MT0 through MT7.

- Error while setting up default specifications for tape drive.

  Internal LABEL error.

- Error while writing terminating file marks on tape.

  You may have a bad tape, or the tape may be write-locked. Clean the tape, check that it has a write-enable ring in place, or rewind the tape.

- "HDR1" not contained in columns 1 - 4 of second record on tape.

  When reading a HDR1 label, the string HDR1 must appear at the beginning of the second record.

- No logical device number was specified.

  You must specify the logical device number (MT0 - MT7).

- The -OWNER value is too long for type B or E labels.

  The owner ID for IBM labels cannot be longer than 10 characters.

- -PARITY is ignored for type "A" and "E" labels.

  You need to set parity only for 7-track tape drive operations, for type B labels.

- The LABEL command typed at PRIMOS command level is invalid.

  Retype the command.

- The tape drive has not been assigned.

  You must ASSIGN a tape drive before you can use LABEL.

- The tape drive is not connected.

  The tape drive is not physically connected to your system.

- The tape drive is not ready or is offline.

  You have assigned a tape drive, but your tape is not mounted on the drive or the drive is not online.

- The tape is not write-enabled (no write ring).

  You cannot write to a tape unless a write ring is present.

- "VOL1" not contained in columns 1 - 4 of first record on tape.

  When reading a VOL1 label, the string VOL1 must appear at the beginning of the record.

- -VOLSER option MUST be specified when writing a new VOL1 label.

  The Volume ID is mandatory when writing labels. It may contain a maximum of six characters.

- Your user-login-name will be used as an owner-identification.

  This is only a warning. If -OWNER is omitted, the default is your login name.

PART II

Using MAGNET

4
Invoking MAGNET

This chapter explains how to invoke MAGNET and describes the format of the MAGNET subcommand line.

THE MAGNET COMMAND LINE

To invoke MAGNET, you must be at PRIMOS command level. The format of the MAGNET command line is:

```
MAGNET [-SILENT] [-OVERWRITE]  { -USER     }
                               { -OPERATOR }
```

Option	Meaning
-SILENT	Suppresses the display of warning messages at your terminal. Only error messages are displayed. (See Appendix D, ERROR MESSAGES.)
-OVERWRITE	Overwrites BRMS-labeled tapes (ARCHIVE, BACKUP, and TRANSPORT). MAGNET can overwrite free tapes without the -OVERWRITE option. MAGNET does not read BRMS-labeled tapes.

-USER	Displays "Mount" and "Dismount" messages at your terminal.
{ -OPERATOR } { -OPR }	Displays "Mount" and "Dismount" messages at the supervisor's terminal.

Notes

If you invoke MAGNET from a user terminal and you do not specify -OPR or -USER, the default is -OPR.

All "Mount" and "Dismount" messages require a reply either from you or from the operator. If you specify -OPERATOR when you invoke MAGNET, the system operator replies to all "Mount" and "Dismount" messages. If you specify -USER when you invoke MAGNET, you must reply to all "Mount" and "Dismount" messages yourself. After you receive a "Mount" or "Dismount" message and a prompt (>) at your terminal, your reply should be one of the following:

```
> REPLY -TAPE GO

      or

> REPLY -TAPE ABORT
```

The GO response tells MAGNET that you have unloaded or loaded the tape, as directed by the message, and that you are ready to continue.

The ABORT response tells MAGNET to stop the current operation and return to subcommand level.

For example, if the end of a tape is reached, MAGNET asks you to unload the tape. Unload the tape, and issue the GO reply. MAGNET then asks you to load the next tape. When you have loaded the continuation reel, issue the GO reply again, and the MAGNET operation continues. If your tape drive supports software-controlled unload, MAGNET does not ask you to unload the tape; you receive only the prompt to load the next tape.

If you type any reply other than GO or ABORT, the "Mount" or "Dismount" message is repeated.

If you direct messages and replies to the operator's console instead of to your input command stream, you can build prepackaged magnetic tape jobs that you can run under Prime's BATCH subsystem. You can run these batch jobs at any time, and all tape mount and dismount instructions are submitted to the operator during job execution.

THE MAGNET SUBCOMMAND LINE

When you have typed the MAGNET command line and pressed the Return key, the MAGNET subsystem displays the release number and a prompt, as follows:

```
OK, MAGNET
[MAGNET, Rev. 21.0 Copyright (c) 1987 Prime Computer, Inc.]
>
```

The MAGNET prompt (>) invites you to type a subcommand line. The format of the subcommand line is:

```
>subcommand   object(s)   option=(value),...option=(value)
```

Item	Description
subcommand	Defines, manipulates, transfers, or translates data within the MAGNET subsystem.
object	The name of a tape file, disk file, or spool file. You can specify one or more objects, or none at all, depending upon the subcommand.
option	Qualifies a subcommand or object. There are five types of option: Disk, Tape, Spool, General, and Edit Tokens. The type of option that you use depends upon the object and subcommand. (For more details about options, see Chapter 6, SUBCOMMAND OPTIONS.)

Some subcommands are issued alone, others require one or more objects or options.

For detailed descriptions and reference tables of subcommands, objects, and options, see Chapter 5, SUBCOMMAND SYNTAX.

General Rules for Subcommand Lines

There are three points to remember when you type subcommand lines:

- The maximum permitted length of a subcommand line is 1024 characters.
- You can use either uppercase or lowercase alphabetic characters or a combination of both.
- Free format is allowed on the subcommand line, that is, spacing is optional.

THE BREAK KEY

MAGNET contains an internal break key handler. If you press <CTRL> and <P> at any time while you are within MAGNET, control returns to MAGNET subcommand level, where you can enter more commands. To exit from MAGNET, use the QUIT subcommand, described in Chapter 5, SUBCOMMAND SYNTAX.

5
Subcommand Syntax

This chapter describes each MAGNET subcommand. Table 5-1 lists the subcommands alphabetically, summarizes their function, and specifies which object types and which options you can use with each subcommand. Following Table 5-1, the function and syntax of each subcommand is described in detail.

Table 5-1
MAGNET Subcommands

Subcommand	Function	Object	Options	
/*	Puts comments in your MAGNET dialog.			
CLOSE	Closes a disk, tape, or spool file.	Tape Disk Spool		
COPY	Copies one or more physical files from one tape to another.	Tape	AMOUNT PRINT	
DECLARE	Identifies an object and defines its attributes.	Tape Disk Spool	AT BFACTOR BUFFERS CHARACTERS CONTROL COPIES DEFER DENSITY DISK EXCHANGE EXTERNAL FILENO FORM FORMAT LIKE	LINENOS LRECL MAXIO NEXTCHAIN OFFSET PAD PARITY POSTACTION PREACTION PROTECT RECORDNO SPOOL TAPE TRACKS VISUAL
DELETE	Deletes one or more declared objects.	Tape Disk Spool		
DISPLAY	Displays all options associated with a declared object.	Tape Disk Spool		
LIST	Lists all declared objects.		ITEMS	
LOAD	Loads a translation table.		TYPE LINES DISK	

Table 5-1 (continued)
MAGNET Subcommands

Subcommand	Function	Object	Options	
MODIFY	Changes option values or adds options to a previously declared object.	Tape Disk Spool	AT BFACTOR BUFFERS CHARACTERS CONTROL COPIES DEFER DENSITY DISK EXCHANGE EXTERNAL FILENO FORM FORMAT LIKE	LINENOS LRECL MAXIO NEXTCHAIN OFFSET PAD PARITY POSTACTION PREACTION PROTECT RECORDNO SPOOL TAPE TRACKS VISUAL
MOVE	Moves a logical file from a source object to one or more destination objects.	Tape Disk Spool		
NOISY	Enables printing of warning messages.			
POSITION	Positions a tape to a specific file number and record number.	Tape	MODE	
QUIT	Exits from MAGNET and returns you to PRIMOS command level.			
READ	Reads a file from a tape and transfers it to a disk file.	Tape Disk		
RENAME	Renames declared objects.	Tape Disk Spool		
SAVE	Saves options associated with an object in a PRIMOS global variable.	Tape Disk Spool	EXTERNAL	

Table 5-1 (continued)
MAGNET Subcommands

Subcommand	Function	Object	Options
SILENT	Suppresses printing of warning messages.		
TRANSLATE	Specifies the translation for a tape object.	Tape	A-Z (edit tokens)
WRITE	Writes a disk file onto tape.	Tape Disk	

SUBCOMMAND FUNCTION AND SYNTAX

The following sections describe the function and syntax of each subcommand. The subcommands are described in alphabetical order for easy reference.

Each subcommand description specifies the type, or types of option that you can use with that subcommand. Option types are Disk, Tape, Spool, General, or Edit Token. For full details of all options and option types, see Chapter 6, SUBCOMMAND OPTIONS.

Note

Please note that you must define your tape, disk, or spool object with the DECLARE subcommand before you issue any other subcommand. (See DECLARE, later in this chapter.)

▶ /*

The /* subcommand puts comments in your MAGNET dialog. Its function, is the same as the PRIMOS comment command. A comment can contain a maximum of 1000 characters. The subcommand line format is

```
>/* this is a comment
>
```

▶ CLOSE

The CLOSE subcommand closes a tape, disk, or spool object that might have been left open by an abnormal condition or when you pressed the break key. For disk and spool objects, the affected files are truncated (if open for write) and are then closed.

The format of the CLOSE subcommand is

```
>CLOSE object
```

where object is the name of a tape, disk, or spool object.

In the following example, the CLOSE subcommand is used to close three objects. A MOVE subcommand is issued to move data between a tape file, TFILE, and two other objects: DFILE (a disk file) and SFILE (a spool file). The example assumes that there is an abnormal end to the MOVE subcommand, and the CLOSE subcommand is issued three times to close each of the three objects. The LIST subcommand is then used to check that the objects are closed.

```
>MOVE TFILE  SFILE  DFILE
              .
              .
              .
>LIST ITEMS=(OPEN)
         object-names                  type      open status
-------------------------------      ----      -----------
TFILE                                 Tape      Open / Read
DFILE                                 Disk      Open / Write
SFILE                                 Spool     Open / Write

>CLOSE TFILE
>CLOSE SFILE
>CLOSE DFILE

>LIST
         object-names                  type      open status
-------------------------------      ----      -----------
TFILE                                 Tape      Closed
DFILE                                 Disk      Closed
SFILE                                 Spool     Closed
```

▶ COPY

The COPY subcommand copies information from a source tape to one or more destination tapes. This is a physical, as opposed to a logical, copy function. This means that the information is copied, regardless of its content.

$$\text{>COPY source-object target-object(s) AMOUNT=}\left\{\begin{array}{c}\text{n}\\ *\end{array}\right\}\text{, [PRINT=}\left\{\begin{array}{c}\text{yes}\\ \text{no}\end{array}\right\}\text{]}$$

Descriptions of the arguments and options follow.

Argument	Description
source-object	The name of the tape object that you wish to copy. You can specify only one source object.
target-object	The name of the tape object(s) that you wish to copy to. You can specify up to seven target objects.

Options	Description
AMOUNT={ n | * }	Mandatory. Specifies the number of physical tape files that you want to copy. Enter either n or asterisk (*). n must be a numeric value in the range 1 through 32,767. An asterisk specifies that you want to copy everything on the tape.
PRINT={ yes | no }	Optional. Default = no. Specifies that the size, in bytes, of the first physical record of each file and the number of physical records in each file is displayed on your terminal as it is copied. Enter either yes or no.

Note

MAGNET positions your tape before copying it. Indicate the COPY starting position on the tape using the FILENO and RECORDNO options with the DECLARE and MODIFY subcommands. The FILENO/RECORDNO options must specify an absolute position; the values for both options must be greater than or equal to 0. If they are equal to 0, none of the tapes are pre-positioned. (For more information on tape positioning, see the POSITION subcommand, later in this chapter.)

The following example illustrates how to use the COPY subcommand and the AMOUNT option. In this example, a tape object, SYSIN, is declared, located on logical device 0. The FILENO/RECORDNO option pair instructs MAGNET to rewind the tape and position it to file number 1, record number 1. Two objects are then declared: SYSOUT, a tape object located on logical device number 1, and SYSOUT2, a tape object located on logical device number 2. With SYSOUT and SYSOUT2, the FILENO/RECORDNO option pair functions in the same way as for the SYSIN declaration. The COPY subcommand line is then issued, which copies all of SYSIN, the source object, to SYSOUT and SYSOUT2, the two target objects. Remember that all of SYSIN is copied because an asterisk (*) was specified as the value for the AMOUNT option.

```
>DECLARE SYSIN TAPE=(0), FILENO=(1), RECORDNO=(1)
>DECLARE SYSOUT TAPE=(1), FILENO=(1), RECORDNO=(1)
>DECLARE SYSOUT2 TAPE=(2), FILENO=(1), RECORDNO=(1)
>COPY SYSIN SYSOUT SYSOUT2 AMOUNT=(*), PRINT=(NO)
>
```

▶ DECLARE

You use the DECLARE subcommand to define and characterize a specified object. You must declare objects before you can use most other MAGNET subcommands.

The format of the DECLARE subcommand line is

```
>DECLARE  object option1=(value) [,option2=(value)...]
```

Descriptions of the arguments and options follow.

Argument	Description
object	Identifies a tape file, disk file, or spool file. You must always specify an object with the DECLARE subcommand. You can use any combination of a maximum of 32 alphabetic characters, numerals, and the two characters period (.) and underscore (_). Any of these characters can appear first in the object-name. Using separate DECLARE subcommands, you may declare a maximum of 100 objects.

Options	Description
option1=(value)	Mandatory. Describes an object. This is the first option that you specify on the DECLARE subcommand line, and must always be TAPE, DISK, SPOOL, LIKE, or EXTERNAL. Specify TAPE for a tape object, DISK for a DISK object, or SPOOL for a spool object. The value for TAPE must be the logical device number. The value for DISK must be a valid PRIMOS pathname or filename, and the value for SPOOL must be the banner heading that you want to be printed. Alternatively, you can use the LIKE option to declare the object as being similar to another object of the same type. The value for the LIKE option must be the name of a previously declared object. Another alternative is to use the EXTERNAL option. The value for the EXTERNAL option must be the name of a PRIMOS global variable that contains the option values that you want to use for this object.

option2=(value)	One or more Tape, Disk, or Spool options. Use only Tape options for tape objects, Disk options for disk objects, or Spool options for spool objects.

▶ DELETE

The DELETE subcommand deletes one declared object.

The subcommand line format is as follows:

```
>DELETE object
```

where object is the name of a tape, disk, or spool object.

In the following example, output from a LIST subcommand shows that there are two previously declared tape objects, STATS1 and STATS2. The command to delete STATS2 is given, and a second LIST subcommand reflects the deletion.

```
>LIST
          object-names                type       open status
-------------------------------       ----       -----------
STATS1                                TAPE       Closed
STATS2                                TAPE       Closed

>DELETE STATS2

>LIST
          object-names                type       open status
-------------------------------       ----       -----------
STATS1                                TAPE       Closed
>
```

▶ DISPLAY

The DISPLAY subcommand displays all options associated with a declared object. The subcommand line format is as follows

```
>DISPLAY object
```

where object is the name of a tape, disk, or spool object.

The subcommand output shows all possible options that you can specify

for an object. Tape options are displayed for tape objects, Disk options for disk objects, and Spool options for spool objects.

Those options that you declared previously for the specified object are displayed with their values, together with those that automatically defaulted.

The following example shows the DISPLAY output for a tape object, SYSIN. SYSIN is declared with two options and their values. The LIST subcommand confirms that SYSIN is a declared tape object. Output from DISPLAY then shows all possible options for SYSIN, and shows the values for the options that were declared (TAPE and FILENO) and for those that defaulted automatically.

```
>DECLARE SYSIN TAPE=(3), FILENO=(5)

>LIST

          object-names                    tape      open status
-----------------------------------       ----      -----------
SYSIN                                     TAPE      Closed

>DISPLAY SYSIN
***  Information for TAPE object SYSIN ***
PREV=                                   / NEXT=
PREACTION          =                      POSTACTION        =
TAPE               =      3               TRACKS            = 9
DENSITY            = 800                  CHARACTERS        = 2
FORMAT             = FIXED                PARITY            =
LRECL              =      0               BFACTOR           =      1
OFFSET             =      0               BUFFERS           =      3
MAXIO              =  10000               VISUAL            =
FILENO             =              5       RECORDNO          =        0
EXCHANGE           = NONE                 PAD               = '040'O
PROTECT            = NO
TRANSLATE     = (A*)
>
```

The next example shows the DISPLAY output for a disk object, SYSOUT. SYSOUT is declared with only one option, DISK. The LIST subcommand confirms that SYSOUT is a disk object. Output from the DISPLAY subcommand shows all possible Disk options for SYSOUT, but displays values only for DISK, LRECL, and FORMAT. (The values shown for LRECL and FORMAT are default values.)

```
>DECLARE  SYSOUT  DISK=(SUE>SOURCE>REV19>FILE2)
>LIST

          object-names                    type      open status
--------------------------------------    ----      -----------
SYSOUT                                    DISK      Closed

>DISPLAY  SYSOUT
***  Information for DISK object SYSOUT  ***
DISK  = SUE>SOURCE>REV19>FILE2
PREV=                                          / NEXT=
PREACTION          =                            POSTACTION    =
LRECL              = 0                          FORMAT        =  VAR/PRIME
>
```

The next example shows the DISPLAY output for a spool object, BONUSES. BONUSES is declared with three options, SPOOL, AT, and COPIES. The LIST subcommand confirms that BONUSES is a spool object. Output from the DISPLAY subcommand shows all possible spool options for BONUSES. Note that only the SPOOL, AT, and COPIES options have received values. Other options, such as FORMAT and LRECL, received default values.

```
>DECLARE  BONUSES  SPOOL=(HOME_OFFICE), AT=(BOSTON), COPIES=(12)
>LIST
          object-names                    type      open status
--------------------------------------    ----      -----------
BONUSES                                   Spool     Closed

>DISPLAY  BONUSES
*** Information for SPOOL object BONUSES ***
SPOOL = HOME_OFFICE
PREV=                                          / NEXT=
PREACTION          =                            POSTACTION        =
LRECL              = 0                          FORMAT            = VAR/PRIME
AT                 = BOSTON                     CONTROL           = NONE
COPIES             =     12                     DEFER             =
FORM               =                            LINENOS           = NO
```

▶ LIST

You can use the LIST subcommand to see which objects you have declared. The subcommand line format is as follows

```
>LIST  [ ITEMS=(value) ]
```

where (value) is DISK, TAPE, SPOOL, OPEN, or CLOSED.

If you specify the LIST subcommand without the ITEMS option, MAGNET displays a chart that shows the name and type of each declared object, and whether the object is open or closed.

You use the ITEMS option to limit the output from the LIST subcommand. You can direct LIST to display all tape objects, disk objects, or spool objects declared previously. Alternatively, you can direct LIST to display only those objects that are not currently in use (CLOSED) or which are in use (OPEN).

In the following example, the LIST subcommand is issued without the ITEMS option. The output shows the three declared objects TFILE, DFILE, and SFILE to be tape, disk, and spool objects, respectively. The status of each object is Closed.

```
>LIST
          object-names                type      open status
-----------------------------         ----      -----------
TFILE                                 Tape      Closed
DFILE                                 Disk      Closed
SFILE                                 Spool     Closed
>
```

▶ LOAD

You can use the LOAD subcommand to load a user-defined translation table for later use with the TRANSLATE subcommand. For more information, see TRANSLATE later in this chapter.) Each user-defined translation table is 512 bytes (256 halfwords) long.

The LOAD subcommand line format is as follows:

```
>LOAD table-name DISK=(pathname), LINES=(n), TYPE=(value)
```

Descriptions of the arguments and options follow.

Argument	Description
table-name	One of V,W,X,Y, or Z that identifies the area of memory where the translation table is to be loaded.

Option	Description
DISK=(pathname)	The pathname of the file that contains the translation table that you wish to load.
LINES=(n)	Specifies the number of lines to be read from the file named by the DISK option. Enter a number that is a divisor of 512. That is, 1, 2, 4, 8, 16, 32, 64, 128, 256, or 512.

Note

For readability, you should not use a small line count (1, 2, 4, 8, or 16), especially if you specify TYPE=(BIN).

TYPE=(value)	Specifies the interpretation of the characters in the translation table that you are loading. Enter BIN, CHAR, OCT, DEC, or HEX.

▶ MODIFY

The MODIFY subcommand changes a declared object or option. For example, using MODIFY, you can add an option to an object or change the value of an option. The MODIFY subcommand line format is as follows:

```
>MODIFY object [options=(value)...]
```

Descriptions of the arguments and options follow.

Argument	Description
object	The name of the tape file, disk file, or spool file that you wish to modify.

Options	Description
options=(value)	One or more Tape, Disk, or Spool options. Use only Tape options for tape objects, Disk options for disk objects, or Spool options for spool objects. You can also specify the LIKE, or EXTERNAL option.

▶ MOVE

The MOVE subcommand moves information simultaneously from a source object to a maximum of seven target objects. Source and target objects must be declared previously. The source object can be either a tape or disk object. Target objects can be disk, tape, or spool objects. To perform a MOVE operation, you must define a BUFFERS option value greater than or equal to 1 for each tape object that you use.

The subcommand line format for MOVE is as follows:

```
>MOVE source-object target-object(s)
```

where source-object is the name of the tape or disk object from which you are moving the information, and target-object is the name of the tape, disk, or spool object to which you want to move the information.

The following example first declares the source object, NEW_DATA, to be a disk file. Next, the first of the two intended target objects, (PLACE1), is declared; this is a tape file with characteristics as shown. The second target object, PLACE2 (a disk file) is then declared with characteristics as shown. The output from a LIST subcommand shows the objects that have just been declared. Finally, the MOVE subcommand is issued to move the information in NEW_DATA simultaneously to PLACE1, and PLACE2.

```
>DECLARE NEW_DATA DISK=(SFA>STATS), LRECL=(80)
>DECLARE PLACE1 TAPE=(0), LRECL=(100), BFACTOR=(10), BUFFERS=(3),
 FORMAT=(VAR/ANSI), FILENO=(10)
>DECLARE PLACE2 DISK=(PARIS>SOURCE>PROGRAM.FTN), LRECL=(80)
>LIST

          object-names                    type       open status
-------------------------------------     ----       -----------
NEW_DATA                                  DISK       Closed
PLACE1                                    TAPE       Closed
PLACE2                                    DISK       Closed

>MOVE  NEW_DATA  PLACE1  PLACE2
>
```

▶ POSITION

The POSITION subcommand positions a tape to a previously specified file number and record number. Positioning can be either absolute (rewind the tape first, then space forward) or relative (space forward or backward from your current position on the tape).

Before you use the POSITION subcommand, you must set the FILENO and RECORDNO options with the DECLARE or MODIFY subcommands. For absolute positioning, the values for both the FILENO and RECORDNO options must be greater than or equal to 1. For relative positioning, the two options can have any suitable values. (For details of the FILENO and RECORDNO options, see Chapter 6, SUBCOMMAND OPTIONS.)

The subcommand line format for POSITION is as follows:

```
>POSITION  object  MODE=(value)
```

Descriptions of the arguments and options follow.

Argument	Description
object	The name of the tape that you wish to position. This tape object must have been declared previously.

Options	Description
MODE=(value)	Specifies either absolute or relative positioning. Enter ABS for absolute positioning, or REL for relative positioning.

The following examples show how you can use the POSITION subcommand and the MODE option.

In the first example, T1 is declared as a tape object located on logical device 0. It contains four physical files with six physical records in each file. The values of the FILENO and RECORDNO options are set to 3 and 6, respectively. Next, the POSITION subcommand specifies absolute positioning for T1. This instructs MAGNET to rewind the tape, space forward two file markers, and space forward five records.

```
>DECLARE  T1  TAPE=(0), FILENO=(3), RECORDNO=(6)

>POSITION  T1 MODE=(ABS)

>
```

There are two ways to rewind your tape. The following example rewinds tape X by giving 1 as the value for both the FILENO and RECORDNO options, then specifying absolute positioning with the POSITION subcommand. Tape Y is rewound by giving very large, negative values for both FILENO and RECORDNO and specifying relative positioning on the POSITION subcommand line.

```
>DECLARE X TAPE=(0), FILENO=(1), RECORDNO=(1)
>DECLARE Y TAPE=(1) FILENO=(-1000000) RECORDNO=(-1000000)

>POSITION X MODE=(ABS)
>POSITION Y MODE=(REL)

>
```

▶ QUIT

The QUIT subcommand exits from MAGNET and returns you to PRIMOS. The subcommand line format follows:

```
>QUIT
OK,
```

Note

MAGNET ignores anything that you type on the subcommand line following QUIT.

▶ READ

The READ subcommand moves information from a tape file (source object) to a disk file (target object). The tape and disk files must be previously declared objects.

The subcommand line format for READ follows:

```
>READ source-object target-object
```

where source-object is the name of a previously declared tape object, and target-object is the name of a previously declared disk object.

In the following example, a tape object SYSIN is declared, located on logical device 0. The third file on the tape is specified, which has five records per physical tape block, with each record 100 characters long: by default, MAGNET is working with fixed-length records. The MODIFY subcommand is used with the BUFFERS option to specify that three internal buffers are being used by the tape object, SYSIN. To read information from the tape with the READ subcommand, the number of buffers for the tape object must be greater than or equal to 1. Next, a disk object, SYSOUT, is declared: this is a disk file called EMPLOYEES, and is in the directory SFA. Finally, the READ subcommand reads the information from SYSIN into SYSOUT.

```
>DECLARE SYSIN TAPE=(0), FILENO=(3), LRECL=(100), BFACTOR=(5)
>MODIFY SYSIN BUFFERS=(3)
>DECLARE SYSOUT DISK=(SFA>EMPLOYEES), LRECL=(100)
>READ SYSIN SYSOUT
>
```

▶ RENAME

The RENAME subcommand changes the name of a declared object. The subcommand line format is as follows:

```
>RENAME old-object new-object
```

where old-object is the name of the tape, disk, or spool object that you wish to rename, and new-object is the new name that you wish to apply to that object. In the following example, EMPNOS1 is a previously declared object, and is shown in the LIST subcommand output. The object EMPNOS1 is renamed EMPNOS2 by the RENAME subcommand, and the new object name is displayed when a second LIST subcommand is issued.

```
>LIST
          object-names                  type      open status
--------------------------------        ----      -----------
EMPNOS1                                 TAPE      Closed

>RENAME   EMPNOS1   EMPNOS2
>LIST
          object-names                  type      open status
--------------------------------        ----      -----------
EMPNOS2                                 TAPE      Closed
>
```

▶ SAVE

The SAVE subcommand saves a group of options from a previously declared object into a PRIMOS global variable. Subsequently, you can retrieve these options and apply them to another object using the DECLARE subcommand with the EXTERNAL option (see the DECLARE subcommand, earlier in this chapter.)

MAGNET saves only those options that you declared or modified to a value other than their default.

The subcommand line format is:

```
>SAVE object EXTERNAL=(global-variable-name)
```

where object is the name of the tape, disk, or spool object whose option values you wish to save, and global-variable-name is the name of the PRIMOS global variable where you wish to store the option values.

SAVE is the opposite of:

```
>DECLARE...EXTERNAL=(...)
```

In the following example, all the options previously specified for the object JJR are stored with their values in the global variable name .SUE:

```
>SAVE JJR EXTERNAL=(.SUE)
>
```

▶ SILENT
▶ NOISY

The two subcommands, SILENT and NOISY, control the display of warning messages at your terminal.

The SILENT subcommand suppresses the display of warning messages, and is equivalent to specifying -SILENT on the PRIMOS command line when you invoke MAGNET.

The NOISY subcommand enables the display of warning messages; this is the default mode for MAGNET if you do not specify -SILENT on the PRIMOS command line.

▶ TRANSLATE

The TRANSLATE subcommand is for tape objects only, and allows you to translate logical records, or parts of logical records to or from industry standard ASCII, EBCDIC, or BCD. Translation is also available for several 7-track binary formats.

For example, in an 80-character logical record, the first 40 characters could specify a person's name in EBCDIC, and the last 40 characters could specify binary information. You could specify a translation for the first 40 characters to Prime ECS, and the second 40 characters could be read as they are.

The subcommand line format for TRANSLATE is as follows:

```
>TRANSLATE object (translate string)
```

Descriptions of the arguments and options follow.

Argument	Description
object	The name of the tape object that you wish to translate.

Options	Description
(translate string)	Specifies how particular fields of a logical record are to be translated. You can specify the translate string in one of the following ways: (repetition-factor(translate string) where repetition-factor is a number that specifies how many times to repeat the (translate string). Or, you can specify the translate string without the repetition factor, as follows: (edit token { length / char }) where edit token is one of the edit tokens A through Z. You must specify either a length or a character following the edit token, where length is a number that specifies how many characters that edit token is to apply to, and character is the fill character to be used with the F edit token. The translate string must be enclosed in parentheses, with the edit token strings separated by commas.

In the following example of the TRANSLATE subcommand, a tape called BETA_TEST is being read. First, a translation of six characters from industry standard ASCII-8 to Prime ECS (A6) is specified. Next, a translation for 40 pairs of fields of six characters from industry standard ASCII-8 to Prime ECS (A6,40(A6)) is specified. This is followed by an instruction to delete the next two characters in the field (A6,40(A6,D2)). Last, any characters that remain in the logical record are translated from industry standard ASCII-8 to Prime ECS (A6,40(A6,D2),A*) .

```
>TRANSLATE BETA_TEST (A6,40(A6,D2),A*)
>
```

To change a translation, you issue another TRANSLATE command line. For example, to change the translation of BETA_TEST shown above, you might type the following:

```
>TRANSLATE BETA_TEST (B4,30(B7,D3),B*)
>
```

Remember that you can always check that the translation you specified is correct, by issuing the DISPLAY command.

▶ WRITE

The WRITE subcommand moves information from a disk file (source object) to a tape file (target object). The function of the WRITE subcommand is the opposite of READ. (READ moves information from a tape file to a disk file.) As with the READ subcommand, the disk and tape files must be previously declared objects. To write information to a tape file with the WRITE subcommand, the number of buffers for the tape object must be greater than or equal to 1.

The subcommand line format for WRITE is as follows:

```
>WRITE source-object target-object
```

where source-object is the name of a previously declared disk object, and target-object is the name of a previously declared tape object. The following example uses two previously declared objects; a tape object called SYSIN and a disk object called SYSOUT. The information in SYSOUT is to be moved into SYSIN:

```
>WRITE  SYSOUT  SYSIN
>
```

6
Subcommand Options

This chapter provides detailed descriptions of all the options that you can use with MAGNET subcommands. The descriptions are grouped by option type, as follows:

- Tape options -- options used with tape objects
- Disk Options -- options used with disk objects
- Spool options -- options used with spool objects
- General options -- options specific to only one subcommand, or options used with any object type
- Edit Tokens -- used in translate strings with the TRANSLATE subcommand only

If you need to reference an option in this chapter, but are not sure which type it is, refer to Table 6-1 first. Table 6-1 lists all MAGNET options, in alphabetical order. It lists the type, the applicable subcommand(s), and the function of each option.

Following Table 6-1, each option is described in alphabetical order within its type group, which is identified within a banner heading. If an option is of more than one type (for example, FORMAT, which you can use with tape, disk, or spool objects), it is described within each of its groups. The description in each case is specific to the option's use with that particular object type.

Table 6-1
MAGNET Subcommand Options

Option	Type	Applicable Subcommand(s)	Function
A-Z	Edit Token	TRANSLATE	Indicates how specified fields of a logical record are to be translated.
AMOUNT	General	COPY	Specifies the number of files to copy.
AT	Spool	DECLARE, MODIFY	Specifies location at which to print a spool file.
BFACTOR	Tape	DECLARE, MODIFY	Specifies the number of logical records per physical tape block.
BUFFERS	Tape	DECLARE, MODIFY	Specifies the number of buffers being used.
CHARACTERS	Tape	DECLARE, MODIFY	Specifies whether one or two characters per word are to be read or written.
CONTROL	Spool	DECLARE, MODIFY	Specifies the type of line-printer carriage control desired.
COPIES	Spool	DECLARE, MODIFY	Specifies the number of copies to be printed on a line-printer.
DEFER	Spool	DECLARE, MODIFY	Specifies the time a spool file is to be printed.
DENSITY	Tape	DECLARE, MODIFY	Specifies density of the tape being read or written.
DISK	Disk	DECLARE, MODIFY LOAD	Specifies the location of a disk file.

Table 6-1 (continued)
MAGNET Subcommand Options

Option	Type	Applicable Subcommand(s)	Function
EXCHANGE	Tape	DECLARE, MODIFY	Indicates whether, for each logical record, high and low order bytes within words are to be exchanged.
EXTERNAL	General	DECLARE, MODIFY SAVE	Indicates that additional options can be found in an external PRIMOS global variable.
FILENO	Tape	DECLARE, MODIFY	Specifies a file number on a tape object.
FORM	Spool	DECLARE, MODIFY	Specifies the type of line-printer forms to be used for printing.
FORMAT	Disk Tape Spool	DECLARE, MODIFY	Identifies the type of records being read or written.
ITEMS	General	LIST	Specifies what type of objects should be listed.
LIKE	General	DECLARE, MODIFY	Identifies a different object-name whose options, option values, and translation edit tokens are to be duplicated.
LINENOS	Spool	DECLARE, MODIFY	Specifies whether line numbers should be printed on line-printer output.
LINES	General	LOAD	Specifies the number of lines in a user translation table.
LRECL	Disk Tape Spool	DECLARE, MODIFY	Specifies the number of bytes in each logical record.

Table 6-1 (continued)
MAGNET Subcommand Options

Option	Type	Applicable Subcommand(s)	Function
MAXIO	Tape	DECLARE, MODIFY	Limits the maximum I/O transfer size.
MODE	General	POSITION	Instructs MAGNET to position a tape either absolutely or relatively.
NEXTCHAIN	Disk Tape Spool	DECLARE, MODIFY	Declares the next tape object in a chain of tape objects.
OFFSET	Tape	DECLARE, MODIFY	Specifies the size of a field, at the beginning of physical records, that contains control characters to be ignored.
PAD	Tape	DECLARE, MODIFY	Specifies a padding character.
PARITY	Tape	DECLARE, MODIFY	Identifies the parity of a seven-track tape.
POSTACTION	Tape Spool	DECLARE, MODIFY	Specifies action to perform when you close a tape or spool file.
PREACTION	Disk Tape Spool	DECLARE, MODIFY	Specifies action to perform when you open a tape, disk, or spool file.
PRINT	General	COPY	Prints the size of the first physical record in each tape file.
PROTECT	Tape	DECLARE, MODIFY	Prevents writing on a tape.

Table 6-1 (continued)
MAGNET Subcommand Options

Option	Type	Applicable Subcommand(s)	Function
RECORDNO	Tape	DECLARE, MODIFY	Specifies the number of physical records to be spaced forward or backward.
SPOOL	Spool	DECLARE, MODIFY	Defines a spool file.
TAPE	Tape	DECLARE, MODIFY	Defines a tape object.
TRACKS	Tape	DECLARE, MODIFY	Specifies the number of tracks on a tape.
TYPE	General	LOAD	Identifies the type of a translation table.
VISUAL	Tape	DECLARE, MODIFY	Identifies the external visual id of a tape reel.

TAPE OPTIONS

TAPE OPTIONS

You can use any of the following options with tape objects.

BFACTOR	LRECL	PREACTION
BUFFERS	MAXIO	PROTECT
CHARACTERS	NEXTCHAIN	RECORDNO
DENSITY	OFFSET	TAPE
EXCHANGE	PAD	TRACKS
FILENO	PARITY	VISUAL
FORMAT	POSTACTION	

BFACTOR

Value : numeric
Default: 1

BFACTOR is the blocking factor option, and applies to tape objects only. It specifies the number of logical records per physical tape block. (For detailed information about logical record sizes and blocking factors, see Appendix A, MAGNETIC TAPE RECORD FORMATS.)

In the following example, the tape file EMPNAMES has a blocking factor of 10 logical records per physical record. Each logical record contains a maximum of 80 bytes. Therefore, the maximum physical record size is 800 bytes:

```
>DECLARE EMPNAMES TAPE=(3), LRECL=(80), BFACTOR=(10)
>
```

BUFFERS

Value : numeric
Range : 0-10
Default: 3

All magnetic tape I/O is carried out through internal buffers. Each declared tape object must have its I/O buffers allocated before you can use the READ, WRITE, or MOVE subcommands. You use the BUFFERS option with the DECLARE or MODIFY subcommand to indicate how many internal buffers are to be used for magnetic tape I/O. If you specify BUFFERS with a DECLARE subcommand and later modify it to a lesser value, the

extra buffers are released (returned to the free storage list).

In the following example, a tape object called SLEDGER is declared. It is located on logical device number 2 and has 3 internal I/O buffers:

```
>DECLARE SLEDGER TAPE=(2), BUFFERS=(3)
>
```

Note

To use a tape object with the READ, WRITE, or MOVE subcommands, you must ensure that the declared BUFFERS option has a value greater than 0.

CHARACTERS

Value : 1 or 2
Default: 2

You can use the CHARACTERS option when you read or write odd-length physical tape blocks. All normal data transfers to or from a tape are executed in halfwords. Because a halfword is composed of two bytes (or characters), only an even number of characters can normally be transferred between memory and a tape. Note that an odd number of halfwords equals an even number of characters (five halfwords = 10 characters). For you to write a true odd-length physical block, the tape drive must be instructed to transfer only one character per halfword. To obtain a one-character-per-halfword string, a two-character-per-halfword string is doubled in length, and each character from the original string is placed in the right-hand byte of a halfword.

When you write a tape file, the CHARACTERS option explodes each block from a two-character-per-halfword string to a one-character-per-halfword string. When you read a tape file, each physical block is read as a one-character-per-block record and imploded to a two-character-per-halfword string. See Figure 6-1 (A) and (B).

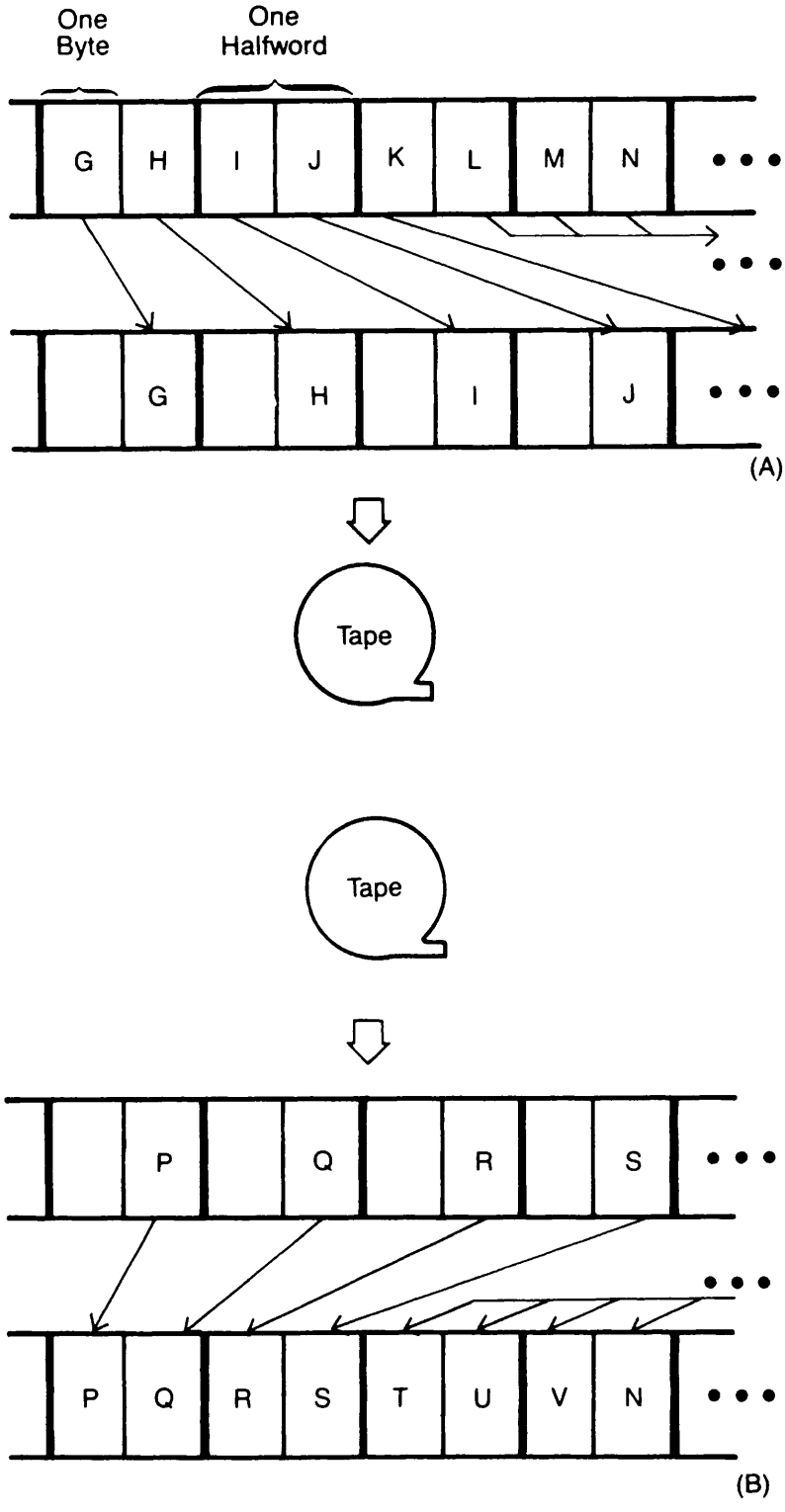
TAPE OPTIONS

Note

If the CHARACTERS and EXCHANGE options are used together when reading, CHARACTERS is processed first. When writing, EXCHANGE is processed first. Note that when you specify a value of 1, the value of LRECL multiplied by BFACTOR must be less than or equal to 6143.

In the following example, LEDGER is declared as a tape object located on logical device number 4. The logical record length is 37 and the blocking factor is 11. The file being worked upon is the 20th file on the tape. Each physical block is to be written as a true odd-length physical block.

```
>DECLARE LEDGER TAPE=(4),LRECL=(37),BFACTOR=(11),CHARACTERS=(1),
FILENO=(20)
>
```



The CHARACTERS Option
(A) Tape Output
(B) Tape Input

Figure 6-1

DENSITY

Value : 200,556,800,1600,3200, 6250, or DEFAULT
Default: Ignores density setting

The DENSITY option specifies the density of the tape that you are reading or writing. For 7-track tapes, you can specify a density of 200, 556, or 800 (bpi). For 9-track tapes, you can specify a density of 800, 1600, 3200, or 6250 (bpi), depending on the type of drive.

When you specify a density, MAGNET attempts to set that density and/or to verify that the tape drive is at the correct density. To set the default value, specify DENSITY=(DEFAULT). MAGNET then ignores the density setting and does not check it.

In the following example, DATAFILE is declared as a tape object located on logical device number 2. It has a density of 1600 bpi.

```
>DECLARE  DATAFILE  TAPE=(2), DENSITY=(1600)
>
```

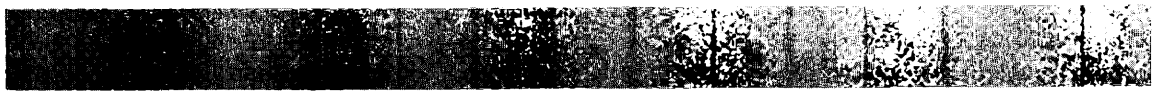
EXCHANGE

Value : NONE or PHYSICAL
Default: NONE

When you write a physical block to tape on a Prime system, each high-order byte within a 16-bit halfword is written preceding the low-order, or second, byte. Some non-Prime operating systems read or write the low-order byte first, and then read or write the high-order byte. The EXCHANGE option allows you to compensate for this. On output, EXCHANGE swaps pairs of bytes in a physical block. On input, EXCHANGE swaps each byte in a physical block with its neighbor before any other processing takes place. See Figure 6-2 (A) and (B).

Unless you specify PHYSICAL as the value for the EXCHANGE option, no bytes are swapped.

TAPE OPTIONS

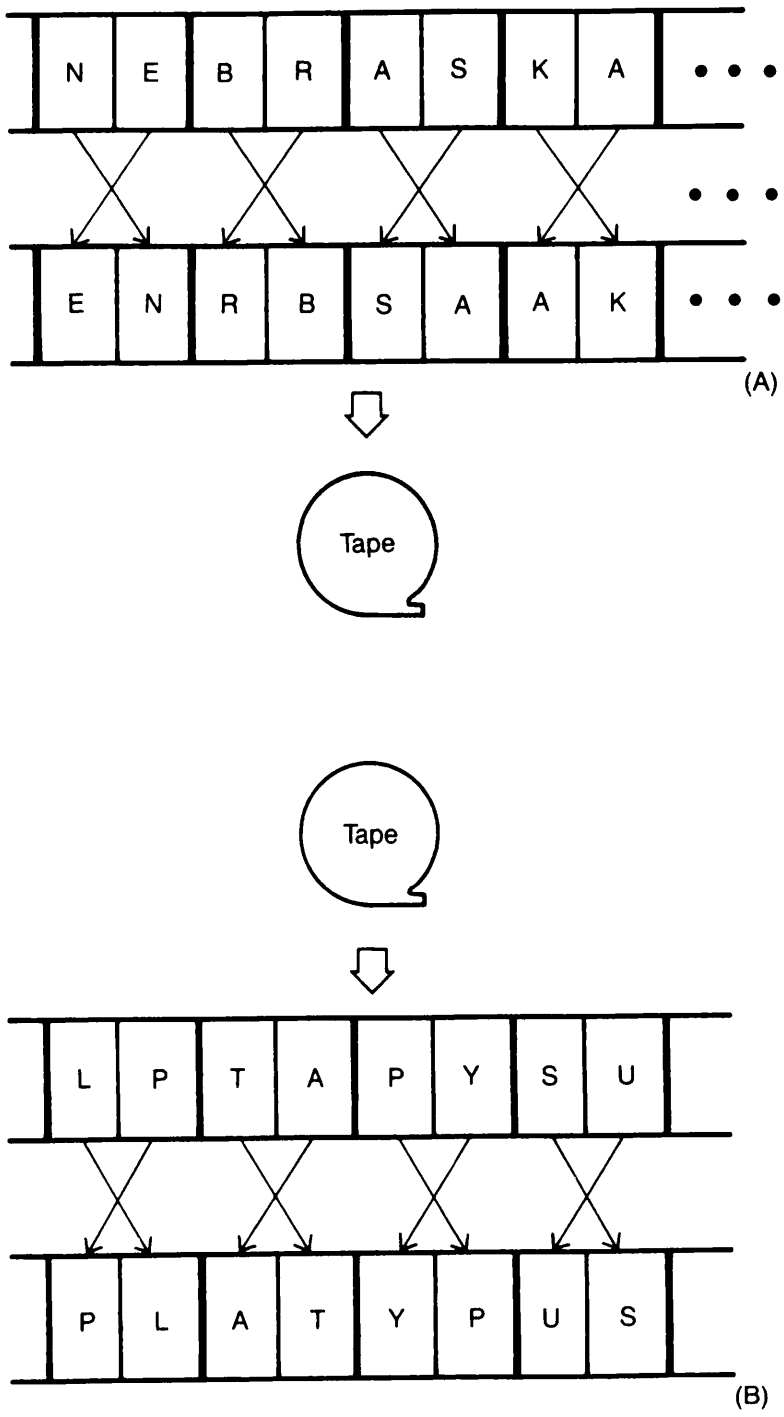


Note

If the physical block has an odd length, the last byte is not swapped.

In the following example, assume that you declare a tape object, JJR, located on logical device number 6. The logical record length is 20 and you are working with the first file on the tape. All bytes in each halfword of the physical block are swapped because you specified PHYSICAL as the value for EXCHANGE.

```
>DECLARE JJR TAPE=(6),LRECL=(20),FILENO=(1),EXCHANGE=(PHYSICAL)
>
```



The EXCHANGE Option
(A) Tape Output
(B) Tape Input

Figure 6-2

TAPE OPTIONS

FILENO

Value : numeric
Default: 0

You use the FILENO option to identify the number of the file you wish to work with on your tape. Its value must be a number greater than or equal to 0: a value of 0 indicates that you wish to work with the file at your current position on the tape. Alternatively, you can use the FILENO option with the POSITION subcommand, to specify absolute or relative file positioning on a tape. (For more information about the POSITION subcommand, see Chapter 5, SUBCOMMAND SYNTAX.)

In the following example, OUTFILE is declared as a tape object located on logical device number 1. This example assumes that you are working with the second file on the tape:

```
>DECLARE OUTFILE TAPE=(1), FILENO=(2)
>
```

Note

When you supply a value to the FILENO option for an operation such as positioning, the option retains that value on completion of the operation. If the value is not appropriate to the next operation, you can change it with the MODIFY subcommand.

TAPE OPTIONS

FORMAT

Value : FIXED, VAR/ANSI, VAR/PRIME, or VAR/IBM
Default: FIXED

The FORMAT option identifies the type of records that you are reading from, or writing to, your tape file.

There are four possible formats that you can specify for tape objects:

- FIXED (for fixed-length records)
- VAR/ANSI (for ANSI standard variable-length records, 9-track only)
- VAR/PRIME (for Prime variable-length records)
- VAR/IBM (for IBM standard variable-length records, 9-track only)

With Prime variable-length records, the logical record length equals the physical record length. The maximum record length that you can specify is defined by the MAXIO option. It is important to note that for VAR/PRIME format, you must always specify a blocking factor of 1, using the BFACTOR option. Prime variable-length tape records have no Record Control Word (RCW) or Block Control Word (BCW). (For more information on the Prime variable-length tape format, see Appendix A, MAGNETIC TAPE RECORD FORMATS.)

In the following example, MTAPE is declared as a tape object located on logical device number 5. It is a 9-track tape with fixed-length records, each of which may have a maximum of 100 characters:

```
>DECLARE MTAPE TAPE=(5), TRACKS=(9), FORMAT=(FIXED), LRECL=(100)
>
```

TAPE OPTIONS

LRECL

Value : numeric
Range : dependent on BFACTOR (see text, below)
Default: mandatory

LRECL is the logical record length option, and you must always specify it. LRECL specifies either the actual number of bytes in each logical record (in fixed-length format) or the maximum number of bytes in each logical record (in variable-length format).

The LRECL and BFACTOR options together specify the maximum size of a tape file's physical records. MAGNET obtains the maximum size by multiplying the value of LRECL by the value of BFACTOR. The maximum size can be no larger than the MAXIO option specifies. This maximum is less if you are using ANSI or IBM variable-length records. (For detailed information about logical record sizes and blocking factors, see Appendix A, MAGNETIC TAPE RECORD FORMATS.)

In the following example, the tape file EMPNAMES has a blocking factor of 10 logical records per physical record. Each logical record contains a maximum of 80 bytes. Therefore, the maximum physical record size is 800 bytes:

```
>DECLARE  EMPNAMES  TAPE=(3), LRECL=(80), BFACTOR=(10)
>
```

TAPE OPTIONS

MAXIO

Value : numeric
Range : 2,000 - MTRS configuration value
Default: 10,000

The MAXIO option specifies the maximum number of characters that MAGNET reads or writes. All tape operations use an internal MAGNET buffer, and MAXIO specifies the size of this buffer. MAXIO cannot exceed the size specified by the Maximum Tape Record Size (MTRS) configuration directive. In the following example, a previously declared tape object, JJR, is modified so that the MAXIO option is specified as 9500. This means that the maximum number of characters requested on input or output is 9500.

```
>MODIFY JJR MAXIO=(9500)
>
```

The minimum value that you can specify for the MAXIO option is OFFSET + (LRECL * BFACTOR).

NEXTCHAIN

Value : name of valid tape object
Default: no tape chaining

It is possible to chain together a number of tape objects. The NEXTCHAIN option identifies the next object in a chain. You would chain tape objects if, for example, you wanted to specify which tape should be mounted when you reach the end of the tape you are currently using. If you do not specify a chain of tapes, and the end-of-tape is detected, MAGNET requests that a new tape be mounted. However, since MAGNET does not initially recognize any characteristics of the new tape, it has no way of knowing if the new tape is the correct one. Consequently, using multireel tape files without specifying the NEXTCHAIN option is useful only for unlabeled tapes.

When you specify a chain of tapes, you should include the BUFFERS option for the first object in the chain only. If you specify BUFFERS for any object after the first in the chain, those buffers are released when that tape object is activated. Other options that specify physical characteristics of the tape (such as TAPE, TRACKS, and DENSITY) or logical characteristics of the file (such as LRECL, BFACTOR, and FORMAT) are copied from the first object in the chain.

TAPE OPTIONS

To identify the next tape to be mounted, you should specify an external visual identification by using the VISUAL option. (For more information, see the description of the VISUAL option, later in this chapter.)

In the following example, a tape object, TAXES, is declared with a number of options. It is the first object in a chain; the next object in the chain is TAXES2. TAXES2 has similar option values as TAXES except for the VISUAL option. Note that, because TAXES2 is the last object in the chain, NEXTCHAIN is not specified for TAXES2. When end-of-tape is detected on TAXES, the tape is rewound and a message is displayed on the operator's console. After the new tape specified by TAXES2 is mounted and the operator replies to MAGNET's message, processing continues with the object TAXES2.

```
>DECLARE TAXES TAPE=(0), LRECL=(100), BFACTOR=(20)
>MODIFY TAXES DENSITY=(6250), FORMAT=(VAR/ANSI)
>MODIFY TAXES VISUAL=(X125)
>MODIFY TAXES BUFFERS=(7), MAXIO=(12000)
>MODIFY TAXES NEXTCHAIN=(TAXES2)
>/*
>DECLARE TAXES2 TAPE=(0), VISUAL=(X126)
>
```

OFFSET

Value : numeric
Range : 0-99
Default: 0

The OFFSET option specifies the length, in characters, of any field in a physical tape block that precedes the data portion of that block.

In the following example, the tape object, TEST, is declared, located on logical device number 1. The file to be worked with is number 12, and each logical record is 80 characters long. The offset field preceding the data portion of each block is 10 characters long. Note that, in this example, because the BFACTOR option is not specified, it defaults to a value of 1. However, because OFFSET=(10) is specified, each physical block in this tape file is 90 characters long.

TAPE OPTIONS

```
>DECLARE TEST TAPE=(1),FILENO=(12),LRECL=(80),OFFSET=(10)
>
```

Note

MAGNET uses the declared OFFSET value only during read operations.

PAD

Value : ('x't) (where t = A,B,D,E,H,I,O, or P, and 'x' depends on t)
Default: ASCII space character (octal 040)

You use the PAD option to specify which padding character MAGNET is to use to use to fill a logical record when you write files to tape.

The PAD option is specified in the following manner:

```
PAD=('xxx't)
```

where the letters x specify one or more characters and t is the type indicator. The single quotation marks must be included.

Eight types of padding can be selected. Padding types are indicated by the following single letters, which you can enter either in upper or lower case.

A	industry-standard ASCII-8
B	BCD
D	decimal
E	EBCDIC
H	hexadecimal
I	binary
O	octal
P	Prime ECS

Thus, to specify an EBCDIC blank as a padding character, you can use any of the following:

```
PAD=(' 'E)              (for EBCDIC)
PAD=('01000000'I)       (for binary)
PAD=('100'O)            (for octal)
PAD=('064'D)            (for decimal)
PAD=('40'H)             (for hexadecimal)
```

If you specify type A, B, E or P, you can enter only one character between the quotes.

If you specity type I, you must enter eight digits between the quotes.

If you specify type D or O, you must enter three digits between the quotes.

If you specify type H, you can enter only two digits between the quotes.

Note that if you select A, B, E, or P, uppercase and lowercase characters are significant within the quotes. That is,

```
PAD=('K'e)
```

is not equivalent to

```
PAD=('k'e)
```

The first results in a padding value of octal 322, while the second would result in a padding value of octal 222.

Because BCD is only a 64-bit code, if you specify lowercase characters they are mapped to uppercase pad characters.

TAPE OPTIONS

PARITY

Value : ODD or EVEN
Default: mandatory for 7-track tape objects

The PARITY option specifies either odd or even parity for a 7-track tape. It is a 7-track tape option only; 9-track tapes are always written with odd parity. There is no default for this option, and you must always specify it when you declare a 7-track tape object.

In the following example, STATS is declared as a tape object located on logical device number 2. It is a 7-track tape with odd parity:

```
>DECLARE STATS TAPE=(2), TRACKS=(7), PARITY=(ODD)
>
```

POSTACTION

Value : REWIND or UNLOAD
Default: No postaction

The POSTACTION option specifies the action that MAGNET is to take when you close a tape file. After a tape file has been closed, you may rewind and/or unload the tape reel.

The following example declares and modifies a tape object, STEVEP. It operates with file number 42 on the tape, and the POSTACTION option specifies a REWIND value. This would instruct MAGNET to rewind the tape after closing tape file number 42:

```
>DECLARE STEVEP TAPE=(7), POSTACTION=(REWIND)
>MODIFY STEVEP LRECL=(80), FILENO=(42)
>
```

The next example again works with file number 42 in the tape object, STEVEP. The POSTACTION option specifies an UNLOAD value. This would instruct MAGNET to rewind and unload (dismount) the tape after closing tape file number 42:

```
>DECLARE STEVEP TAPE=(7), POSTACTION=(UNLOAD)
>MODIFY STEVEP LRECL=(80), FILENO=(42)
>
```

TAPE OPTIONS

PREACTION

Value : REWIND
Default: No preaction

You can use the PREACTION option to rewind a reel of tape before searching for and opening a tape file. This is useful if your tape is positioned beyond the point at which you want to work.

In the following example, STEVEP is declared as a tape file located on logical device number 7. Each logical record in the file is 80 characters long. The PREACTION option specifies a REWIND value. This instructs MAGNET to rewind the tape before it opens file number 23.

```
>DECLARE STEVEP TAPE=(7), PREACTION=(REWIND)
>MODIFY STEVEP LRECL=(80), FILENO=(23)
>
```

PROTECT

Value : YES or NO
Default: NO

The PROTECT option is the software equivalent of a write-enable ring on a reel of magnetic tape. If you specify the PROTECT option with a value of YES, you cannot write to your tape. If you specify the PROTECT option with a value of NO, you can write to your tape only if a write-permit ring is present.

In the following example, VITAL_INFO is declared as a tape object located on logical device 0, and it is protected from writing:

```
>DECLARE VITAL_INFO TAPE=(0), PROTECT=(YES)
>
```

TAPE OPTIONS

RECORDNO

Value : numeric
Default: 1

The RECORDNO option identifies the number of physical records that you wish to position either forward or backward. You use this option only in conjunction with the POSITION and COPY subcommands. (The POSITION and COPY subcommands are described in Chapter 5, SUBCOMMAND SYNTAX.)

The value that you specify with RECORDNO is either a positive or a negative number. You use a positive number to position the tape forwards, and a negative number to position the tape backwards.

In the following example, BAR is declared as a tape object located on logical device 0. RECORDNO indicates to the POSITION subcommand that the tape is to be positioned five filemarkers backward, and 17 physical records forward:

```
>DECLARE BAR TAPE=(0), FILENO=(-5), RECORDNO=(17)
>POSITION BAR MODE=(ABS)
>
```

TAPE

Value : numeric
Range : 0-7
Default: mandatory

You specify the TAPE option only if the object-name preceding it on the subcommand line refers to a tape file. The value you assign to this option is the logical device number of the tape drive on which the tape is mounted. Remember to assign your tape drives at PRIMOS command level before you invoke MAGNET. Chapter 2, Setting Up Tapes describes how to do this, using the ASSIGN command.

In the following example, the DECLARE subcommand defines the object-name TFILE to be a tape file. It is located on logical device (tape drive) number 7.

```
>DECLARE TFILE TAPE=(7)
>
```

TRACKS

Value : 7 or 9
Default: 9

The TRACKS option identifies the number of tracks on your declared tape.

In the following example, EMPNAMES is declared as a 9-track tape, located on logical device number 6.

```
>DECLARE EMPNAMES TAPE=(6), TRACKS=(9)
>
```

VISUAL

Value : 8-character alphanumeric string
Default: 8 spaces

The VISUAL option specifies the external visual identification of the tape reel that you are using. The external visual identification is located on the tape reel enclosure. VISUAL simplifies the identification of tape reels for MOUNT and DISMOUNT requests. To use the VISUAL option, enter a string of eight characters that can be any combination of the following:

- Uppercase and/or lowercase alphabetic characters
- Numbers
- Any other characters, except: right and left parentheses; equal signs; commas; apostrophes; and blanks

In the following example, BLEDGER is declared as a tape object located on logical device number 7. The external visual identification of this tape reel is ACCOUNTS2:

```
>DECLARE BLEDGER TAPE=(7), VISUAL=(ACCOUNTS2)
>
```

DISK OPTIONS

DISK OPTIONS

You can use any of the following options with disk objects.

DISK
FORMAT
LRECL
PREACTION

DISK

Value : valid PRIMOS pathname or filename
Default: mandatory

You specify the DISK option only if the object-name preceding it on the subcommand line refers to a disk file. The value you assign to this option must be a PRIMOS filename or pathname.

In the following example, the object-name DFILE is a PRIMOS disk file, SOCSECNUMS, in the directory SFA:

```
>DECLARE  DFILE  DISK=(SFA>SOCSECNUMS)
>
```

FORMAT

Value : FIXED, VAR/PRIME, UNCOMPRESSED
Default: VAR/PRIME

The FORMAT option identifies the type of records that you are reading from, or writing to, your disk file.

There are three possible formats that you can specify for disk objects:

- FIXED (for fixed-length records)
- VAR/PRIME (for Prime variable-length records, ASCII files)
- UNCOMPRESSED (for binary files and partially translated records)

FIXED format is for reading and writing ASCII records. When reading from disk, the record is truncated or space-padded to the record length. When writing to disk, multiple spaces are compressed and trailing spaces are removed.

VAR/PRIME format records are similar to FIXED format records, except that they are not space-padded. However, if you move information from disk to a FIXED format tape the records are space-padded.

The UNCOMPRESSED format is used for processing records that contain binary fields. This format reads fixed length records and ignores record terminators.

In the following example, DFILE is declared as a disk object with fixed-length records, each of which may have a maximum of 120 characters:

```
>DECLARE DFILE DISK=(SFA>SOCSECNUMS), FORMAT=(FIXED), LRECL=(120)
>
```

LRECL

Value : even number
Range : 2 - 12,288
Default: 0

LRECL is the logical record length option, and specifies either the size in characters of each record in a binary disk file, or the maximum size in characters of each record in an ASCII disk file.

In the following example, X is declared as a disk file, SALESFILE, and each logical record in the disk file is 80 characters long:

```
>DECLARE X DISK=(*>SALESFILE), LRECL=(80)
>
```

Note

With a binary disk file, you must also specify FORMAT=(UNCOMPRESSED). For ASCII files, you should specify FORMAT=(VAR/PRIME). For more information, see the discussion of the FORMAT option, earlier in this chapter.

DISK OPTIONS

PREACTION

Value : APPEND
Default: No preaction

When you open a disk file, you can use the PREACTION option to position the tape to the end of the disk file so that new information read into it will be concatenated at the end.

In the following example, a disk object, SUEA, is declared with the filename SUBSFILE. The PREACTION option has the value APPEND, which instructs MAGNET to position to the end of SUBSFILE when it is opened.

```
>DECLARE SUEA DISK=(SUBSFILE), PREACTION=(APPEND)
>
```

Note

The PREACTION option with the APPEND value is useful only when you read into a disk file.

SPOOL OPTIONS

You can use any of the following options with spool objects.

AT	FORM	POSTACTION
CONTROL	FORMAT	PREACTION
COPIES	LINENOS	SPOOL
DEFER	LRECL	

AT

Value : alphanumeric string
Range : maximum 32 characters
Default: spaces

The AT option specifies the location at which to print a spool file. The value of the AT option must be a string of a maximum of 32 alphanumeric characters that identifies the print location. These characters can be any combination of the following:

- Uppercase and/or lowercase alphabetic characters
- Numbers
- Any other characters, except: right and left parentheses; equal signs; commas; apostrophes; and blanks

In the following example, the spool object GROSS_PAY is declared, to be printed at a location called BOSTON:

```
>DECLARE GROSS_PAY  SPOOL=(EMPLOYEES),  AT=(BOSTON)
```

SPOOL OPTIONS

CONTROL

Value : FORTRAN or NONE
Default: NONE

The CONTROL option specifies the type of carriage control used by a spool file.

In the following example, the spool object PURPLE is declared with carriage control type FORTRAN.

```
>DECLARE PURPLE SPOOL=(COLORS), CONTROL=(FORTRAN)
>
```

COPIES

Value : numeric
Range : 1 - 32,767
Default: 1

The COPIES option specifies the number of copies of your spool file to print.

In the following example, 23 copies of the declared spool object GROSS_PAY are required:

```
>DECLARE GROSS_PAY  SPOOL=(EMPLOYEES),  COPIES=(23)
```

DEFER

Value : hhmm
Default: spaces

The DEFER option specifies the time to print a spool file, and is useful if you wish to avoid printing a spool file at a busy period. The time must be specified in 24-hour format (for example, 11:30 p.m. would be specified as 2330).

The following example specifies that a spool file is to be printed at 10:15 p.m.

```
>MODIFY PURPLE  DEFER=(2215)
>
```

FORM

Value : alphanumeric character string
Range : maximum 6 characters
Default: default form type

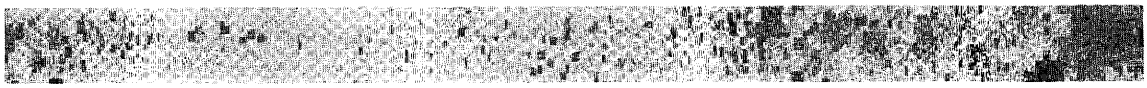
The FORM option specifies the type of line printer forms on which to print your spool file. The value of this option is a string of no more than 6 characters, representing the form type. These characters can be a combination of any of the following:

- Upper and/or lowercase alphabetic characters
- Numbers
- Any other characters, except: right and left parentheses; equal signs; commas; apostrophes; and blanks

In the following example, a spool object named TOTALS is declared with the AT, COPIES and FORM options:

```
>DECLARE TOTALS  SPOOL=(RECEIPTS),  COPIES=(12),
FORM=(3PART), AT=(FRAMINGHAM)
```

SPOOL OPTIONS



FORMAT

Value : FIXED or VAR/PRIME
Default: VAR/PRIME

The FORMAT option identifies the type of records that you are writing to your spool file.

There are two possible formats that you can specify for spool objects:

- FIXED (for fixed-length records)
- VAR/PRIME (for Prime variable-length records, ASCII files)

Prime variable-length records are not blank-padded.

In the following example, X is declared as a spool object with banner MEMOS. It is a Prime variable-length file with records that may have a maximum of 80 characters:

```
>DECLARE X SPOOL=(MEMOS), LRECL=(80), FORMAT=(VAR/PRIME)
>
```

LINENOS

Value : YES or NO
Default: NO

The LINENOS option enables you to direct the spooler to print line numbers for every line printed in a spool file.

In the following example, a spool object, CHARTS87, is declared with line numbers:

```
>DECLARE CHARTS87 SPOOL=(COURSES), LINENOS=(YES)
>
```

LRECL

Value : numeric
Range : maximum 132
Default: mandatory

LRECL is the logical record length option, and specifies the maximum character length of each print line in your spool file. You must always specify the value of LRECL for spool files.

In the following example, SALES is declared as a spool object with a banner heading of AREA1. The maximum character length of each line is declared as 64:

```
>DECLARE SALES SPOOL=(AREA1), LRECL=(64)
```

POSTACTION

Value : NOEJECT
Default: page ejection

The POSTACTION option specifies the action that MAGNET is to take when you close a spool file. After a spool file has been closed, you may or may not require a page eject when print is complete. If you do not require a page eject, specify NOEJECT as the value for the POSTACTION option. The default is an automatic page ejection.

In the following example, a spool object, REPORTS, is declared with the banner heading REP1, 10 copies, and no page ejection when print is complete.

```
>DECLARE REPORTS SPOOL=(REP1),COPIES=(10),POSTACTION=(NOEJECT)
>
```

SPOOL OPTIONS

PREACTION

Value : SUPPRESS
Default: print banner

The PREACTION option specifies the action that MAGNET is to take when you open a spool file. The only preaction that you can specify is SUPPRESS, which suppresses the printing of a banner heading.

In the following example, a spool object, EXP, is declared with banner DEPT2, 12 copies and a print location of OZ. The PREACTION option specifies that the banner is not to be printed.

```
>DECLARE EXP SPOOL=(DEPT2),COPIES=(12),AT=(OZ),
PREACTION=(SUPPRESS)
```

SPOOL

Value : alphanumeric string
Range : maximum 32 characters
Default: mandatory

You specify the SPOOL option only if the object-name preceding it on the subcommand line refers to a spool file. The value you assign to this option must be an alphanumeric string of no more than 32 characters. The value of the SPOOL option is printed as a banner on your spooler output.

In the following example, PRODUCTS is declared as a spool object. The SPOOL option value, AUTOS, appears as the banner on the line printer output:

```
>DECLARE PRODUCTS SPOOL=(AUTOS)
>
```

GENERAL OPTIONS

Six of the options in this category have functions that are specific to one subcommand only. The remaining two options, EXTERNAL and LIKE, must be used in a particular sequence on the DECLARE subcommand line. The eight General options are

AMOUNT	LINES
EXTERNAL	MODE
ITEMS	PRINT
LIKE	TYPE

AMOUNT

Value : numeric or *
Range : 1 - 32,767
Default: mandatory

You use the AMOUNT option only with the COPY subcommand. It identifies the number of physical tape files to copy. There is no default for AMOUNT. You must specify this option on the COPY subcommand line.

There are two possible values that you can specify for the AMOUNT option, either a numeric value (n) or an asterisk (*). A numeric value identifies the number of physical files to copy. If you wish to copy everything on your tape until you reach the first filemarker after the physical end-of-tape (EOT), then you specify an asterisk (*). If the physical EOT is detected, and if you did not specify AMOUNT=(*), then a single file marker is read, copying stops, and MAGNET requests a new source tape. Similarly, if the physical EOT is detected on a destination tape, MAGNET writes a single file marker and requests a new tape.

The following example shows how you can use the COPY subcommand and the AMOUNT option. In this example, a tape object, SYSIN, is declared, located on logical device 0. The FILENO/RECORDNO option pair instructs MAGNET to rewind the tape and position it to file number 1, record number 1.

GENERAL OPTIONS

Two objects are declared: SYSOUT, a tape object located on logical device number 1, and SYSOUT2, a tape object located on logical device number 2. With SYSOUT and SYSOUT2, the FILENO/RECORDNO option pair functions in the same way as for the SYSIN declaration. The COPY subcommand line is then issued, which copies all of SYSIN, the source object, to SYSOUT and SYSOUT2, the two target objects. Remember that all of SYSIN is copied because an asterisk (*) is specified as the value for the AMOUNT option.

```
>DECLARE SYSIN TAPE=(0), FILENO=(1), RECORDNO=(1)
>DECLARE SYSOUT TAPE=(1), FILENO=(1), RECORDNO=(1)
>DECLARE SYSOUT2 TAPE=(2), FILENO=(1), RECORDNO=(1)
>COPY SYSIN SYSOUT SYSOUT2 AMOUNT=(*), PRINT=(NO)
>
```

EXTERNAL

Value : PRIMOS Global Variable name
Default:

You use the EXTERNAL option only with the DECLARE, MODIFY, or SAVE subcommands.

When you use MAGNET, there may be specific sets of options and their associated values that you will want to use more than once. The EXTERNAL option allows you to save and reload these options so that you do not have to retype them each time. The options and values are stored in PRIMOS global variables.

To save your options and option values, use the SAVE subcommand with the EXTERNAL option.

To retrieve your stored options and option values, use the DECLARE or MODIFY subcommand with the EXTERNAL option.

The value that you specify for the EXTERNAL option must be a valid PRIMOS global variable name, that comprises a string of no more than 32 characters. The first character in the string must always be a period (.). The other characters in the string can be any combination of the following:

- Uppercase and/or lowercase alpha characters
- Numbers
- Underscores
- Periods

In the following example, a global variable file, IO.GVAR is first defined at PRIMOS command level. Output from a LIST_VAR command shows that the two variables .SYSIN and .SYSOUT are defined. MAGNET is then invoked, and two objects are declared, SYSIN and SYSOUT. The EXTERNAL option for each of these objects is specified, the values being .SYSIN and .SYSOUT, respectively. This instructs MAGNET to retrieve the global variable values in .SYSIN and .SYSOUT and put them into the internal variables SYSIN and SYSOUT. Next, the LIST subcommand is issued, which confirms that SYSIN and SYSOUT are declared objects. Finally, the DISPLAY subcommand output for both SYSIN and SYSOUT shows all the values associated with variables SYSIN and SYSOUT.

GENERAL OPTIONS

```
OK,DEFINE_GVAR IO.GVAR
OK,LIST_VAR
.SYSOUT                 DISK=(TESTDIR>SFA>FILE3),LRECL=(80)
.SYSIN                  TAPE=(0),BUFFERS=(2),LRECL=(80),FILENO=(1)
OK,MAGNET
[MAGNET, Rev. 21.0]
>DECLARE SYSIN EXTERNAL=(.SYSIN)
>DECLARE SYSOUT EXTERNAL=(.SYSOUT)
>LIST

          object-names                  type      open status
--------------------------------      ----      -----------
SYSIN                                 TAPE      Closed
SYSOUT                                DISK      Closed


>DISPLAY SYSIN
*** Information for TAPE object SYSIN ***
PREV=                                       / NEXT=
PREACTION          =                         POSTACTION    =
TAPE               =      0                  TRACKS        = 9
DENSITY            = 800                     CHARACTERS    = 2
FORMAT             = FIXED                   PARITY        =
LRECL              =     80                  BFACTOR       =     1
OFFSET             =      0                  BUFFERS       =     2
MAXIO              =  10000                  VISUAL        =
FILENO             =              1          RECORDNO      =         0
EXCHANGE           = NONE                    PAD           = '040'O
PROTECT            = NO
TRANSLATE          = (A*)
>DISPLAY SYSOUT
*** Information for DISK object SYSOUT ***
DISK = TESTDIR>SFA>FILE3
PREV=                                       / NEXT=
PREACTION          =                         POSTACTION    =
LRECL              =     80                  FORMAT        = VAR/PRIME
>QUIT
OK,
```

GENERAL OPTIONS

With the DECLARE or MODIFY subcommands, you may specify EXTERNAL first in the option list on the subcommand line. However, no matter where EXTERNAL is positioned in the option list, MAGNET ignores any options that follow it. If you specify EXTERNAL as the first option with a DECLARE subcommand, you must define the TAPE, DISK, or SPOOL options first within the CPL variable.

There are a few points to remember when you use the EXTERNAL option:

- A global variable cannot contain an EXTERNAL option. You can use EXTERNAL only on the MAGNET subcommand line.
- Options retrieved from a global variable overwrite any of the same options already declared or modified.
- Global variables cannot contain more than 1024 characters.
- You may specify the same name for an EXTERNAL value and an object.
- You cannot store Edit Tokens in a PRIMOS global variable. Therefore, you cannot save or retrieve them using the EXTERNAL option.

For more information about the EXTERNAL option, see the descriptions of DECLARE and SAVE in Chapter 5, SUBCOMMAND SYNTAX.

ITEMS

Value : TAPE, DISK, SPOOL, OPEN or CLOSED
Default: all open and closed objects are listed

The ITEMS option limits the output from the LIST subcommand. You can direct LIST to display all tape objects, disk objects, or spool objects declared previously. Alternatively, you can direct LIST to display only those objects that are not currently being used (CLOSED) or which are being used (OPEN). An object might remain open if, for example, you pressed the BREAK key while data was being transferred.

GENERAL OPTIONS

The following example requests that only spool objects are displayed:

```
>LIST  ITEMS=(SPOOL)

          object-names                type       open status
-------------------------------     ----       -----------
SFILE                                 Spool      Closed
>
```

LIKE

Value : name of a declared object
Default: no options copied

You can use the LIKE option with the DECLARE or MODIFY subcommand to copy the option values and translation edit tokens of one object to another object. For example, if you are using a multireel tape file, you must use two or more tape objects (depending on the number of tape reels) linked by the NEXTCHAIN option. (See the description of the NEXTCHAIN option, earlier in this chapter.) Chained tapes usually have many options in common, such as LRECL, BFACTOR, FORMAT, TRACKS, or PROTECT. Rather than retype these options and their values repeatedly, you can use LIKE to duplicate them.

As an example, consider the following set of DECLARE and MODIFY statements:

```
>DECLARE INFILE  TAPE=(0), LRECL=(100), BFACTOR=(20),
FORMAT=(VAR/ANSI)
>MODIFY INFILE VISUAL=(0131811), BUFFERS=(4)
>MODIFY INFILE NEXTCHAIN=(INFILE2)
>TRANSLATE INFILE (A70,6(F*),3(A6,2(F)))
>/*
>DECLARE INFILE2  TAPE=(0), LRECL=(100), BFACTOR=(20),
FORMAT=(VAR/ANSI)
>MODIFY INFILE2 VISUAL=(0131812), BUFFERS=(4)
>TRANSLATE INFILE2 (A70,6(F*),3(A6,2(F)))
```

This sequence of statements can be shortened dramatically by typing:

```
>DECLARE INFILE  TAPE=(0), LRECL=(100), BFACTOR=(20),
FORMAT=(VAR/ANSI)
>MODIFY INFILE VISUAL=(0131811), BUFFERS=(4)
>TRANSLATE INFILE (A70,6(F*),3(A6,2(F)))
>/*
>DECLARE INFILE2 LIKE=(INFILE), VISUAL=(0131812)
>MODIFY INFILE2 NEXTCHAIN=(INFILE2)
>
```

In this example, all options, option values, and translation edit tokens associated with INFILE are duplicated in INFILE2. Remember, however, that INFILE2 is an object in its own right. If the BUFFERS option were modified in INFILE, for example, the update would not be reflected in INFILE2. Another MODIFY command would have to be issued to change the BUFFERS value for INFILE2. Similarly, two DELETE commands would be necessary to erase these objects and return their associated memory to the free memory pool. A LIST command would list two objects, not one, and two DISPLAY commands would be necessary to view each object's option values.

LINES

Value : 1, 2, 4, 8, 16, 32, 64, 128, 256, or 512
Default: Mandatory

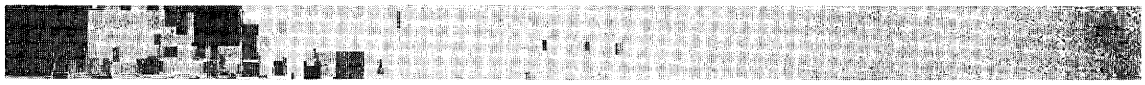
You use the LINES option only with the LOAD subcommand, and it is mandatory to that command. The LINES option tells MAGNET how many lines to read in the translation table that you name as the value for DISK in the LOAD subcommand line.

The number of binary, octal, decimal, or hexadecimal numbers (or characters) that must appear on each line of the translation table is 512 divided by the value of the LINES option.

Note

For readability, you should not use a small line count (1, 2, 4, 8, or 16).

GENERAL OPTIONS



MODE

Value : ABS or REL
Default: mandatory

The MODE option specifies either absolute or relative tape positioning on a POSITION subcommand line.

The following examples show how you can use the POSITION subcommand and the MODE option.

T1 is declared as a tape object located on logical device 0. It contains four physical files with six physical records in each file. The values of the FILENO and RECORDNO options are set to 3 and 6, respectively. The POSITION subcommand specifies absolute positioning for T1. This instructs MAGNET to rewind the tape, space forward two file markers, and then space forward five records.

```
>DECLARE  T1  TAPE=(0), FILENO=(3), RECORDNO=(6)

>POSITION  T1 MODE=(ABS)

>
```

There are two ways to rewind your tape. The following example rewinds tape X by giving 1 as the value for both the FILENO and RECORDNO options, then specifying absolute positioning with the POSITION subcommand. Tape Y is rewound by giving very large, negative values for both FILENO and RECORDNO and specifying relative positioning on the POSITION subcommand line.

```
>DECLARE X TAPE=(0), FILENO=(1), RECORDNO=(1)
>DECLARE Y TAPE=(1) FILENO=(-1000000) RECORDNO=(-1000000)

>POSITION X MODE=(ABS)
>POSITION Y MODE=(REL)

>
```

GENERAL OPTIONS

PRINT

Value : YES or NO
Default: NO

You can use the PRINT option only with the COPY subcommand. The PRINT option specifies that the size, in bytes, of the first physical record of each tape file is printed on your terminal as it is copied.

In the following example, the COPY subcommand line is issued. This copies all of the source object SYSIN to the two target objects SYSOUT and SYSOUT2. All of SYSIN is copied because an asterisk (*) is specified as the value for the AMOUNT option.

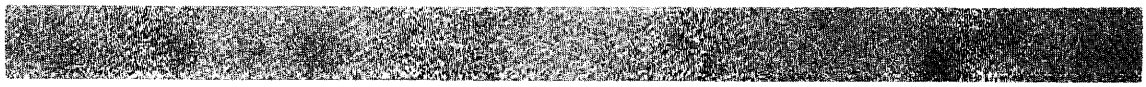
```
>COPY SYSIN SYSOUT SYSOUT2 AMOUNT=(*), PRINT=(YES)
>
```

Because the above example specifies PRINT=(YES), a table similar to the following is printed as the copy operation proceeds:

File #	Bytes Read	Total Blocks Read
1	200	250
2	350	97
3	790	1123
.	.	.
.	.	.
.	.	.

When MAGNET has copied the first block, it displays the file number and the number of bytes read. When MAGNET has completed copying a file, the total number of physical records in that file is displayed in the third column of the table.

GENERAL OPTIONS



TYPE

Value : BIN, CHAR, OCT, DEC, or HEX
Default: mandatory

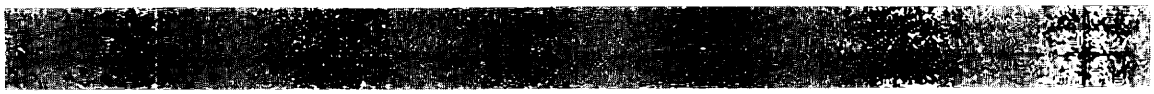
The TYPE option is mandatory to the LOAD subcommand, and specifies the interpretation of the characters in the translation table that you name as the value for DISK in the LOAD subcommand line.

You must specify one of the following five values:

- BIN (binary number character strings)
- CHAR (ASCII character strings)
- OCT (octal number character strings)
- DEC (decimal number character strings)
- HEX (hexadecimal number character strings)

All binary digit strings must be eight digits (or characters) long. All octal and decimal digit strings must be three digits (or characters) long. All hexadecimal digit strings must be two digits (or characters) long.

EDIT TOKENS



EDIT TOKENS

Edit tokens are part of the TRANSLATE subcommand line. Each edit token specifies how you want particular fields of a logical record to be translated. An edit token is an alpha character followed by either a number or an asterisk.

Note

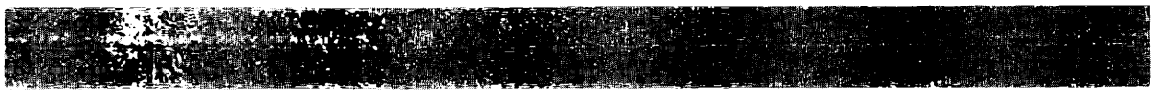
The edit token F is the only token that can be followed by any character except a comma (,), parentheses (), or apostrophe (').

A translate string is a group of edit tokens separated by commas. Each edit token or translate string may be preceded by an optional repetition factor, which is always a number. All MAGNET edit tokens are listed in Table 6-2, and described in detail in the following paragraphs.

Table 6-2
Translation Edit Tokens

Edit Token	Function
A	Translates Prime ECS to and from 8-bit industry-standard ASCII
B	Translates Prime ECS to and from BCD code
C	Moves the column position within fields of logical records
D	Deletes characters in logical records
E	Translates Prime ECS to and from EBCDIC code
F	Specifies fill characters
G	Specifies seven-track packing code 2424
H	Specifies seven-track packing code 4242
I	Specifies seven-track packing code 2466
J	Specifies seven-track packing code 4266
K	Specifies seven-track packing code 6246
L	Specifies seven-track packing code 6426
M	Specifies seven-track packing code 6624
N	Specifies seven-track packing code 6642
O	Specifies no translation
P	Translates Prime ECS to and from 7-bit Prime ASCII
Q - R	Not used
S	Translates Prime ECS to and from 7-bit industry-standard ASCII
T - U	Not used
V - Z	Specify user codes

EDIT TOKENS



Edit Token A

All characters specified with edit token A translate to or from industry-standard 8-bit ASCII (ASCII-8). When you read from a tape, the specified characters are translated from ASCII-8 to Prime ECS. When you write to a tape, the specified characters are translated from Prime ECS to ASCII-8. For example, if you have 80-character ASCII-8 logical records on an input tape, and you want to translate them to Prime ECS, you can specify the following:

```
(A80)
```

Note that you may also specify this translation as (80(A1)), but this format is processed much more slowly.

Edit Token B

The edit token B translates 7-track BCDIC code to Prime ECS, and vice versa. For example, if you have declared a tape object named BANKST consisting of 120-character logical records, your TRANSLATE subcommand line may be

```
>TRANSLATE BANKST (B120)
>
```

If you were using BANKST as an input object, this would translate each of the 120 characters in a logical record from 7-track BCDIC code to Prime ECS. Again, note that you can also use the form (120(B1)), but it is processed much more slowly.

EDIT TOKENS

Edit Token C

Edit token C repositions MAGNET's internal pointer to another field location in a logical record. For example, assuming input from a tape, the string

```
(A80,C10,A10)
```

causes the following:

- Translates character columns 1 through 80 from ASCII-8 to Prime ECS
- Repositions to column 10
- Translates character columns 10 through 19 from ASCII-8 to Prime ECS

In this example, output from the translation process is 90 characters even though the length of the input logical record is only 80 characters. Note that columns 10 through 19 are translated twice and appear twice in the output. However, the following example results in fewer characters:

```
(A40,C71,A10)
```

Here, character columns 41 through 70 are not translated and are effectively deleted from the output.

Edit Token D

Edit token D allows you to delete characters. For example, the string

```
(A40,D30,A10)
```

performs the same task as the second example for edit token C. In other words, (A40,C71,A10) is equivalent to (A40,D30,A10).

EDIT TOKENS

Edit Token E

Edit token E translates characters to or from EBCDIC code. When you read a tape, the specified characters are translated from EBCDIC code to Prime ECS. When you write to a tape, the specified characters are translated from Prime ECS to EBCDIC. For example, the string

```
(E100)
```

translates 100 characters from EBCDIC to Prime ECS or vice versa.

Edit Token F

The F edit token allows you to specify fill characters. These fill characters may be any symbols except the following:

- Comma ,
- Left and right parentheses ( )
- Single quote '

Note that alpha characters are always translated to uppercase. For example, the string:

```
(A10,10(F*),C20,A20,10(FX))
```

causes the following:

EDIT TOKENS

- Translates ten characters using A format
- Inserts ten asterisks (*) into the output record
- Positions to column 20
- Translates 20 characters (columns 20 through 39) using A format
- Inserts 10 Xs into the output record

In this example, the output record would contain 50 characters. If you specify the F edit token by itself, without any fill character, a blank is inserted. The string

```
(20(F))
```

inserts 20 blanks into the output record.

Edit Tokens G Through N

The eight edit tokens G through N specify 7-track packing and unpacking codes. Edit tokens G through N and their corresponding packing codes are as follows:

Edit Token	Packing Code
G	2424
H	4242
I	2466
J	4266
K	6246
L	6426
M	6624
N	6642

The repetition factor you specify for these edit tokens is a byte count. Thus, if on input you wish to pack three binary numbers in K format, you specify (K9). You use the number 9 because each binary number on input occupies three bytes. On output, each binary number occupies only two bytes or one 16-bit halfword. Similarly, G and H format binary numbers occupy four bytes on input from a tape, and occupy two bytes on output to disk. When you write to a tape, binary numbers are expanded from two bytes to four bytes for G and H formats, and to three bytes for I, J, K, L, M, and N formats. When you determine logical record lengths for tape and disk objects, you must take into account 7-track binary number expansion and compression.

In the following example, 20 binary numbers are translated using H format:

```
(H80)
```

Edit Token O

You use the edit token O to specify no translation in your TRANSLATE subcommand line. For example, the subcommand line

```
>TRANSLATE ACCOUNTS (O100)
>
```

transfers each logical record in ACCOUNTS through the translation mechanism without any translation (assuming that each logical record is 100 characters long).

Edit Token P

The edit token P translates Prime ECS to and from 7-bit Prime ASCII (Prime-7). When you read from or write to tape, characters that have their high bit set off translate to "?". For example, the string

```
(P100)
```

translates 100 characters to Prime-7 code.

EDIT TOKENS

Edit Tokens Q and R

The edit tokens Q and R are not currently used.

Edit Token S

The edit token S translates Prime ECS to and from industry-standard 7-bit ASCII (ASCII-7). When you read from tape, characters that have their high bit set on translate to "?". When you write to tape, characters that have their high bit set off translate to "?". For example, the string

(S100)

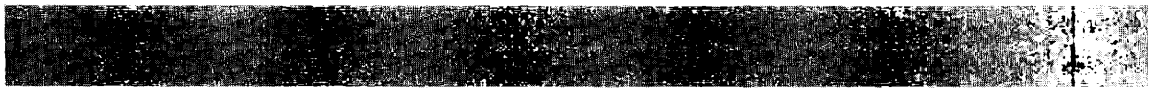
translates 100 characters to ASCII-7 code.

Edit Tokens T and U

The edit tokens T and U are not used.

Edit Tokens V through Z

The five edit tokens V through Z specify user-defined translation codes. You use these edit tokens in the same way that you use edit tokens A, B, E, O, P, and S. One character of input is translated according to the table you loaded with a LOAD subcommand and becomes one character of output. (For information about the LOAD subcommand, see Chapter 5, SUBCOMMAND SYNTAX.)



Repetition Factor *

You can use the repetition factor (*) to specify translation for the remaining characters of a logical record without specifying an absolute number. For example, the subcommand line

```
>TRANSLATE MTAPE1 (A*)
>
```

translates all the characters in each logical record of MTAPE1 according to A format. The subcommand line

```
>TRANSLATE MTAPE2 (E29,O10,3(F),E*)
>
```

causes the following:

- Translates twenty-nine characters according to E format
- Does not translate the next ten characters or bytes
- Inserts three blanks
- Translates all remaining characters according to E format

7
Sample MAGNET Sessions

This chapter provides examples and explanations of three MAGNET sessions, and a CPL program.

DOCUMENTATION CONVENTIONS USED IN THE EXAMPLES

To enhance readability, user responses in the following sample sessions are not underlined. With the exception of the system and copyright messages, everything that follows the OK, and > prompts is user input.

In each example, line numbers are added to the left of the command lines so that you can reference the appropriate explanation in the succeeding section.

EXAMPLE ONE - WRITING A TAPE IN STANDARD ASCII

This example writes a file to tape at the current tape position using fixed length records blocked ten to a physical record.

```
OK, MAGNET
   [MAGNET Rev. 21.0 Copyright (c) 1986, Prime Computer, Inc.]
1  > DCL TAPE TAPE=(0),LRECL=(132),BFACTOR=(10)
2  > DCL DISK DISK=(INPUT_DATA),LRECL=(132)
3  > MOVE DISK TAPE
    Object name = DISK :   Operation complete.
    Object name = DISK :   35 Logical records read.
    Object name = TAPE :   4 Physical blocks written.
    Object name = TAPE :   35 Logical records written.
   > QUIT
```

Explanation of Example One

1. TAPE is declared as a tape object with a record length of 132, and a blocking factor of 10. The following options have been allowed to default:

   ```
   Format             = Fixed
   Fileno./Recordno.  = 0/1 (no positioning)
   ```

2. DISK is declared as a disk object with a record length of 132. The following option has been allowed to default:

   ```
   Format             = VAR/PRIME
   ```

3. The MOVE subcommand instructs MAGNET to read data from the object DISK and write it to the object TAPE. Data is written at the current tape position because when TAPE was declared in line 1, the FILENO and RECORDNO options defaulted to 0 and 1.

EXAMPLE TWO - READING THE THIRD FILE ON AN EBCDIC TAPE

This example reads the third file on a tape. Each logical record has a fixed length of 80 characters.

```
OK, MAGNET
   [MAGNET Rev.  21.0 Copyright (c) 1986,  Prime Computer,  Inc.]
1  > DCL TAPE TAPE=(0),LRECL=(80),FILENO=(3)
2  > DCL DISK DISK=(OUTPUT_FILE),LRECL=(80)
3  > TRANSLATE  TAPE  (E*)
4  > MOVE TAPE DISK
    Object name = TAPE :  Operation  complete.
    Object name = TAPE :  250 Physical  blocks  read.
    Object name = TAPE :  5000 Logical records  read.
    Object name = DISK :  5000 Logical records written.
   > Q
```

Explanation of Example Two

1. TAPE is declared as a tape object with a record length of 80 characters. The FILENO option is specified with a value of 3. The following options have been allowed to default:

Format	= Fixed
Recordno	= 1

   Because the Fileno/Recordno option pair is 3/1, the tape is positioned to the beginning of the third file. Blocking Factor is not used when reading tapes.

2. DISK is declared as a disk object with a record length of 80 characters. The following option has been allowed to default:

Format	= VAR/PRIME

3. The TRANSLATE subcommand instructs MAGNET to translate every character of the third file on the tape to EBCDIC format.

4. The MOVE subcommand writes the data from the tape file to the disk file.

EXAMPLE THREE - READ A LABELED TAPE CONTAINING VARIABLE LENGTH RECORDS

On an ANSI labeled tape, the data portion of each file is preceded by a label group and terminated with a label group. Consequently, three tape files comprise each logical file, and the data is contained in the second tape file. This example shows how to read files on a labeled tape.

```
OK, MAGNET
   [MAGNET Rev. 21.0 Copyright (c) 1986, Prime Computer, Inc.]
1  > DCL TAPE TAPE=(0),FILENO=(2),LRECL=(132),FORMAT=(VAR/ANSI)
2  > DCL DISK DISK=(FILE1),LRECL=(132)
3  > MOVE TAPE DISK
    Object name = TAPE :   Operation complete.
    Object name = TAPE :   5 Physical blocks read.
    Object name = TAPE :   401 Logical records read.
    Object name = DISK :   401 Logical records written.
4  > MODIFY TAPE FILENO=(5)
5  > MODIFY DISK DISK=(FILE2)
6  > MOVE TAPE DISK
    Object name = TAPE :   Operation complete.
    Object name = TAPE :   7 Physical blocks read.
    Object name = TAPE :   623 Logical records read.
    Object name = DISK :   623 Logical records written.
   > QUIT
```

Explanation of Example Three

1. TAPE is declared as a tape object with the following options:

Record length (LRECL)	= 132	
Format	= VAR/ANSI	Ansi format variable length records
File No./Record No.	= 2/1	Position to the third file. Rec no. defaults to 1.

   Blocking Factor is not used when reading tapes.

2. DISK is declared as a disk object with a record length of 132. The Format option is allowed to default to VAR/PRIME.

3. The data portion of the first logical file is read into the disk file.

4. The file number is changed to 5, which coresponds to the data portion of the second logical file.

5. The disk file pathname is changed so that the first file is not overwritten.

6. The second logical file is read.

EXAMPLE FOUR - CPL TO READ AN IBM LABELED TAPE

This example shows how you can create a CPL program to run a MAGNET session.

```
1  &S  unit_number  :=  [RESPONSE  'Enter  Unit Number']
2  &S  logical_file_no := [RESPONSE 'Enter file number']
3  &S  actual_file_no := [calc  (%logical_file_no%  -  1)  *  3  +  2]
4  &S disk_file_name := [RESPONSE 'Enter disk file pathname']
5  &S  rec_length  :=  [RESPONSE 'Enter record Length' 80]

   &data MAGNET
6  DCL TAPE TAPE=(%unit_number%),FILENO=(%Actual_file_no%),
   FORMAT=(VAR/IBM)
7  DCL DISK DISK=(%disk_file_name%),LRECL=(%rec_length%)
8  MODIFY TAPE LRECL=(%rec_length%)
9  TRANSLATE TAPE (E*)
10 MOVE TAPE DISK
   QUIT
   &end
```

Explanation of Example Four

1. The tape drive number is requested from the user.
2. The logical file number is requested.
3. The physical file number is calculated by P = (L-1)*3 + 2
4. The disk file name is requested.
5. The record length is requested.
6. The tape object is declared using requested values.
7. The disk object is declared.
8. The tape record length is set.
9. The EBCDIC translation is specified.
10. The tape is read into the disk file.

APPENDICES

A

Magnetic Tape Record Formats

This appendix is for inexperienced users of magnetic tapes. It provides elementary information about the physical characteristics of magnetic tapes, and describes the different formats of magnetic tape records.

INTRODUCTION TO MAGNETIC TAPES

Magnetic tapes in computer applications are used to read and write digital information.

Magnetic tape is a convenient medium for data storage. It is compact, easy to handle, low cost, and can be used repeatedly. Data recorded on magnetic tape can only be accessed serially. This means that each record is accessed in turn. Magnetic tape is therefore an ideal medium for storing data that you need to reference in sequence, for example a mailing list, but not so useful for data that you need to access randomly. Magnetic tape is frequently used to transport software from one computer site to another, and to store backup copies of data files and operating systems that reside permanently on magnetic disk.

Physical Characteristics

A computer tape is a long, narrow piece of plastic film, thinly coated on one side with metal oxide. To record information, certain areas of

the coating are magnetized. The magnetized and non-magnetized areas represent binary 1s and 0s (called bits.) The magnetized areas remain on the tape, either until the tape is remagnetized by recording new information over the old information, or until the tape is demagnetized by a bulk erasing device. Because tapes are susceptible to magnetic fields, they must be stored carefully to guard against destruction of information.

Computer tape is wound onto a reel. The reel usually contains a removable ring on one side called the write-enable ring. If the ring is in place, the tape can be read or written. If the ring is removed, the tape can be read but no information can be written.

A computer tape is 1/2 inch wide and ranges from 600 to 2400 feet long. The usual sizes available from suppliers are 600, 1200, or 2400 feet. Cartridge tapes are 1/4 inch wide.

Density, Tracks, and Frames

The storage capacity of a magnetic tape is determined by the density at which characters are recorded. Tape density is measured in terms of bits per inch (bpi). The most common densities are 800, 1600, and 6250 bits per inch.

A magnetic tape contains either seven or nine parallel tracks, running the length of the tape. Characters are written across the tape. That is, each character is recorded as a pattern of bits, one bit per track. The area needed to hold one bit per track across the width of the tape is called a tape frame (or tape character). A 7-track tape frame has seven bits, one for each track, and a 9-track tape frame has nine bits, one for each of its nine tracks. 9-track tapes and 7-track tapes are the same width, but 9-track tapes have narrower tracks.

Both 7-track and 9-track tapes have one track set aside for checking parity. This is called the parity track. Each tape frame thus has a parity bit.

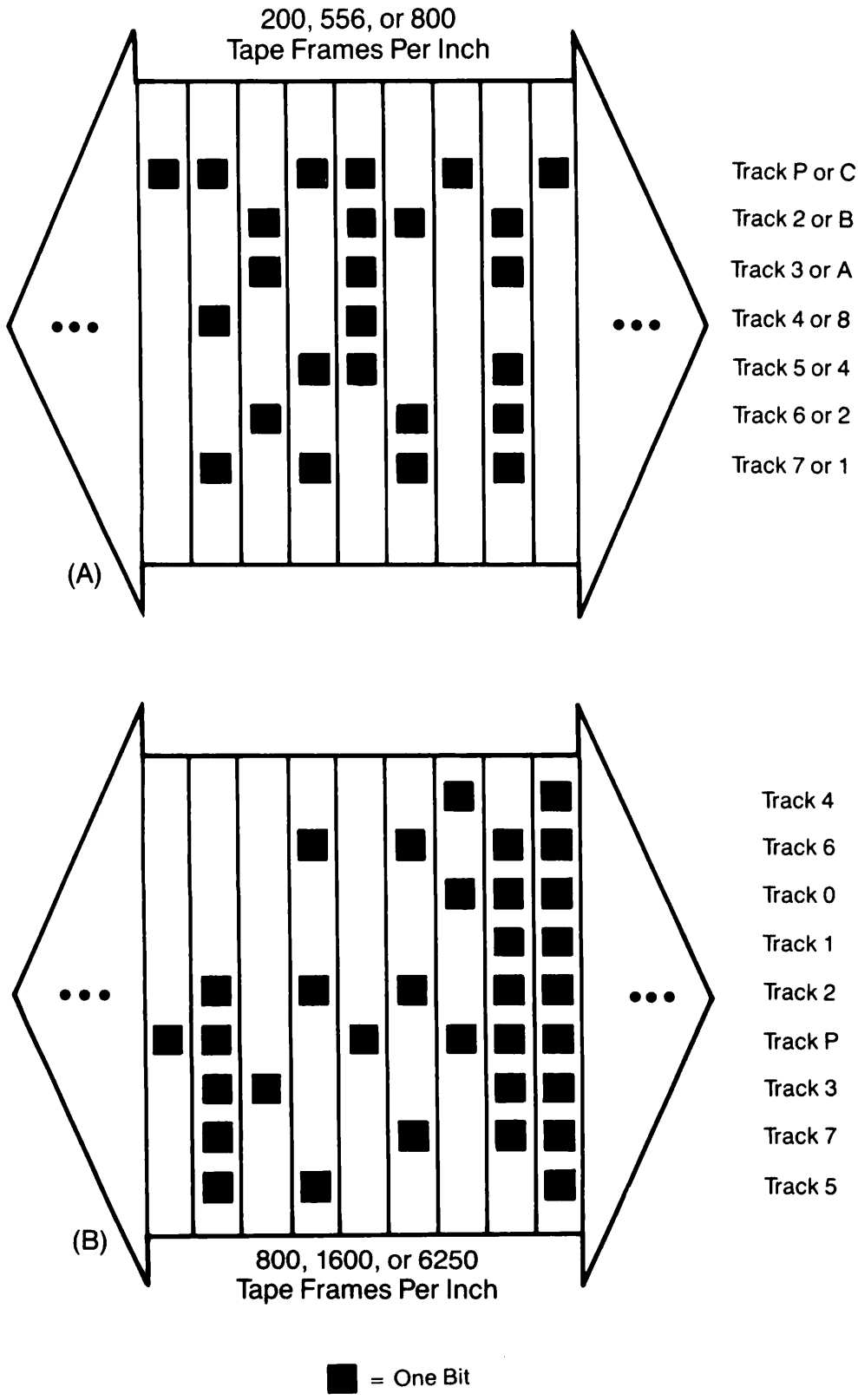
The parity bit is used by the system to check the validity of the information stored on the tape. There are four possible settings for a parity bit:

- Always on; error if a parity-bit-off condition occurs.
- Always off; error if a parity-bit-on condition occurs.
- Even parity; the data bits for a frame are added. If the result is even, the parity bit is set to 0; otherwise, it is set to 1.
- Odd parity; the data bits for a frame are added. If the result is even, the parity bit is set to 1; otherwise, it is set to 0.

7-track tapes can be read or written with either the even or odd parity selections. Prime tape drives always write odd parity for 9-track tapes.

9-track tapes are more common than 7-track tapes.

Figure A-1 shows how characters are stored on 7-track and 9-track tape.



Formats for (A) 7-track Tape, (B) 9-track Tape
Figure A-1

Records, Gaps, and Blocks

The computer can read and write only a certain number of characters during a single read or write operation. A group of characters read or written at one time is called a physical tape record or physical block. Physical records are separated by inter-record gaps (IRGs.) The inter-record gap is at least half an inch long, and can sometimes be as long as 3 inches.

If you write many small records to tape, you also create a large number of inter-record gaps. For example, if you write 80-character records at a density of 800 bpi, each record will occupy 1/10 of an inch. Each IRG will be at least five times as large as each record, so you waste space. This problem can be solved by blocking the 80-character logical records together to form a smaller number of physical records.

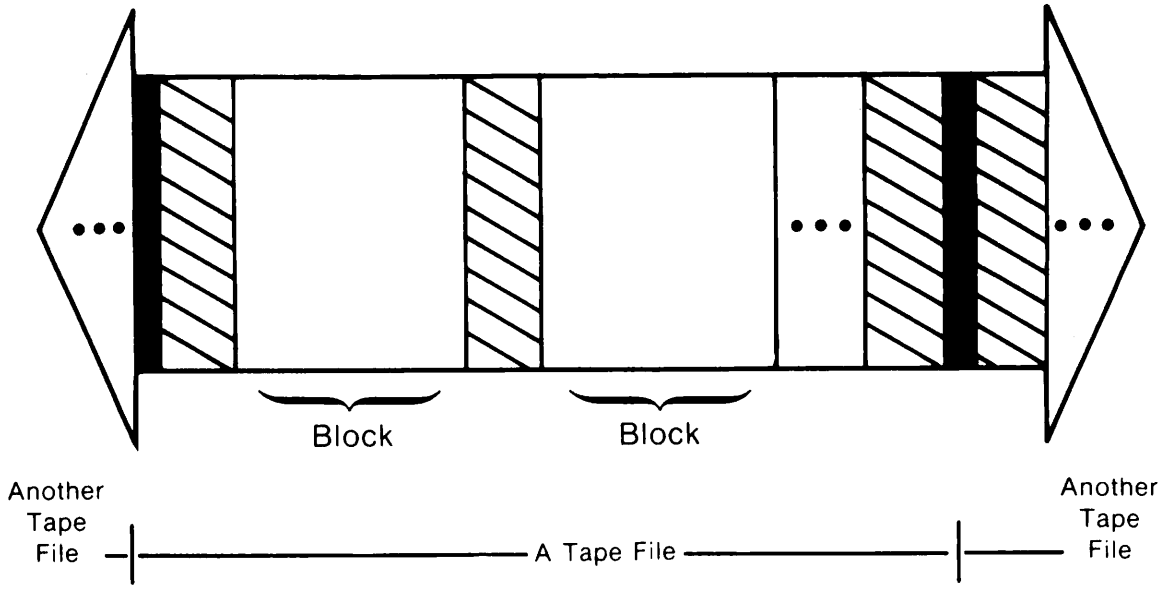
The maximum size of Prime physical tape records is defined by the Maximum Tape Record Size (MTRS) configuration directive. The maximum value for MTRS is 32K bytes, and the default value is 6K bytes. If the maximum size is configured as 32K bytes (32,768 characters), the 80-character logical records mentioned above could be blocked so that 409 fit into each physical record. The blocking factor in this case is 409. Each physical record written is 32,720 characters long (409 * 80), and 48 characters are unused.

The characters in a physical tape record appear to the tape drive to be one large record. User programs must perform the blocking and unblocking of physical-to-logical and logical-to-physical translations.

Tape Files

Blocks are grouped to form a tape file. There can be more than one file on a tape. The file limits are marked on the tape by special control characters. These characters are called file markers or tape markers. A file marker is composed of approximately 3 inches of blank tape followed by a special file marker control character. The entire file marker area is at least 4 inches long. The term for the blank tape area plus the file marker control character is file marker gap, but the term file marker (or tape marker) is used in this manual.

When a tape contains more than one file, separated by file markers, the tape is said to have a logical tape format. The logical tape format is shown in Figure A-2.



Logical Tape Format
Figure A-2

Character Codes

A series of eight bits is called a byte, and a byte represents a character; for example, a letter or numeral. The interpretation of the contents of a byte depends on the character code used. There are three standard character codes, as follows:

- The American National Standard Code for Information Interchange (ASCII)
- The Extended Binary Coded Decimal Interchange Code (EBCDIC)
- The Binary Coded Decimal Interchange Code (BCDIC or BCD)

ASCII is the code used on Prime computers. EBCDIC is used on IBM systems, and BCDIC is used primarily as a 7-track tape interchange code. There are 7-bit and 8-bit versions of the ASCII code. The Prime versions of this code are called, respectively, Prime-7 and Prime Extended Character Set (Prime ECS). To write information currently encoded in ASCII to a 7-track tape, the information must first be translated to BCDIC, which is a 6-bit code. Prime ECS is an 8-bit code, and therefore only one quarter of the Prime ECS characters can be translated to BCDIC characters. The characters that cannot be translated include lowercase letters, some special characters, and all additional characters provided by the extension of ASCII from a 7-bit to an 8-bit code.

Some characters cannot readily be translated to meaningful characters in another code. The usual solution is to switch to a character that is recognized as not translatable.

The 7-bit ASCII code fully defines 128 characters. However, most computers that use ASCII have eight-bit bytes. Prime software sets the remaining (high-order) bit to a 1. This fact must be kept in mind when you transfer information in 7-bit character code between Prime systems and other ASCII-based systems that set high-order bits to 0.

For detailed information about character code translation, see Appendix C, CHARACTER SET AND TRANSLATION TABLES.

LOGICAL RECORDS

Earlier sections of this appendix have described the physical aspects of magnetic tapes, such as bits, frames, records, files, tracks, and density. This section explains logical records in more detail.

Records, whether stored on magnetic disks or tapes, cards or paper tape, take one of two forms: either fixed-length or variable-length. Fixed-length records are all the same size. If user data does not completely fill a fixed-length record, empty character positions are set to any pad character. Variable-length records, however, can range in size from zero characters to a user-defined maximum. Both

fixed-length and variable-length records can be blocked on magnetic tapes.

Under PRIMOS, the maximum size of a physical record is limited to 16,384 halfwords (or 32,768 bytes or characters). Thus, the maximum size of a fixed-length logical record is 32,768 characters if the blocking factor is 1. If the record size is decreased, the blocking factor can be increased.

All records in a file that contains fixed-length records must be the same size except for the last record, which may be smaller.

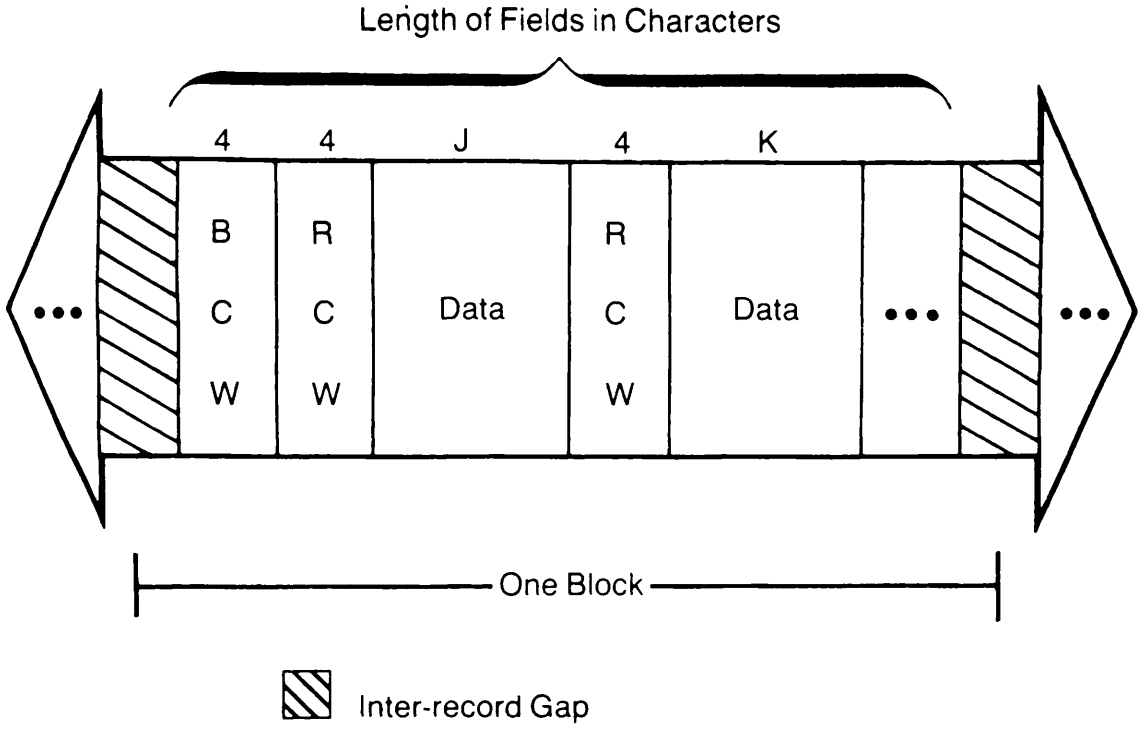
Variable-length records can be written in different formats:

- IBM standard variable
- American National Standards Institute (ANSI) standard variable
- IBM standard spanned variable
- ANSI standard spanned variable
- Prime variable-length

Each of these formats is discussed in the following sections.

IBM Standard Variable-length Records

The maximum size of an IBM variable record under PRIMOS is 35,752 characters. Smaller maximum size limits result in larger possible blocking factors. The tape format for variable-length records is a 4-character block control word (BCW) followed by a block of records. As shown in Figure A-3, each record is composed of a 4-character record control word (RCW) and data.



IBM Standard Variable-length Records
Figure A-3

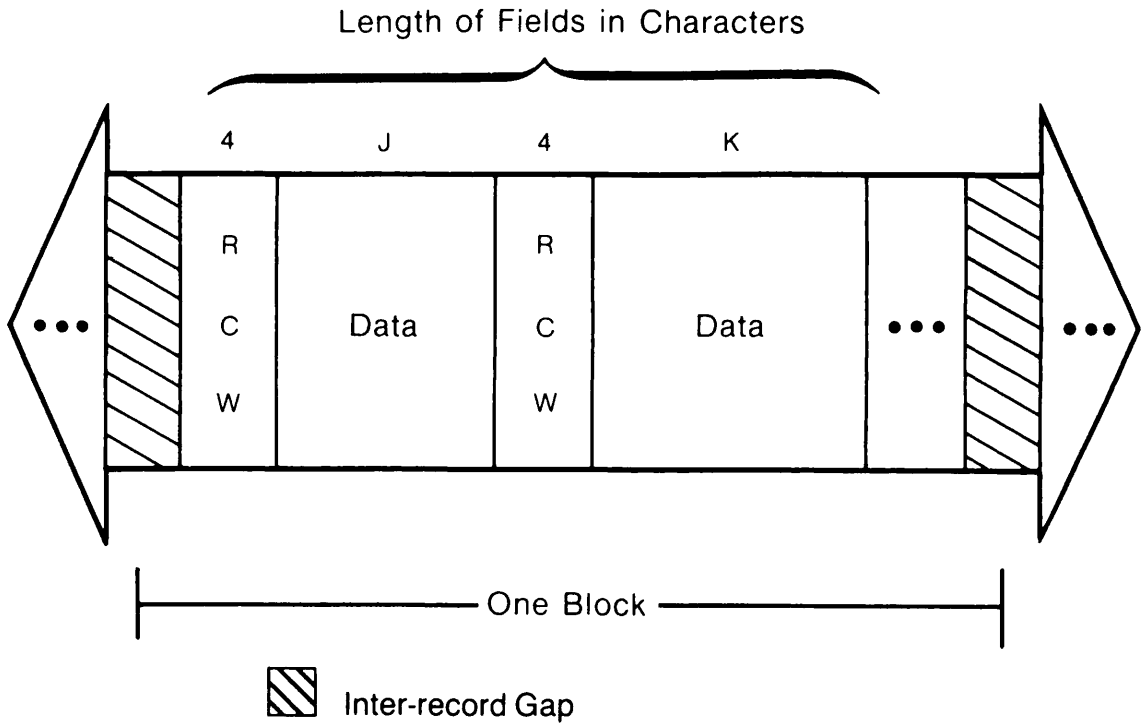
In Figure A-3, the length of the data portion of one record is shown as J, and the other data record's length is shown as K. These two lengths are not necessarily equal. The only constraint on record length is that the sum of the BCW, plus all the RCWs, and all the data portions must not exceed 32,760 bytes.

The BCW contains the length of the entire physical block, or record. The length is stored as an unsigned 16-bit integer in the first two bytes. The second two bytes are reserved and are set to 0. It is important to note that the BCW includes the length of the BCW itself, which is four bytes. Also, on IBM systems, the number stored in the BCW should be between 8 through 32,760. The RCW has the same format as that of the BCW, but the length is that of the logical record plus the length of the RCW (four bytes).

ANSI Standard Variable-length Records

The maximum size of an ANSI variable record under PRIMOS is 32,764 characters. Smaller maximum size limits result in larger possible blocking factors. This format is similar to the IBM variable-length

format except that there is no block control word (BCW). As shown in Figure A-4, each record is composed of a record control word (RCW) plus a data portion.



ANSI Standard Variable-length Records
Figure A-4

In Figure A-4, the length of the data portion of one record is shown as J while the other data record's length is shown as K. These two lengths are not necessarily equal. The only constraint on the record length is that the sum of the RCWs and all the data portions must not exceed 32,768 bytes. The format of the ANSI RCW is not the same as IBM's RCW. The length is an integer, recorded as a 4-byte character string that completely fills the RCW. There are no corresponding reserved bytes as there are in the IBM format.

IBM Standard Spanned Variable-length Records

This section describes IBM standard spanned records for your information only. MAGNET does not support spanned records.

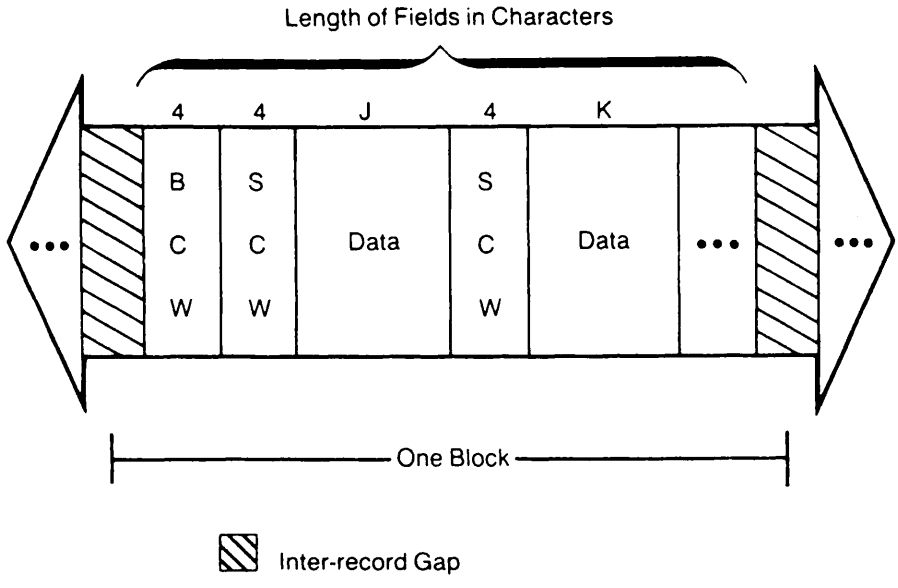
Spanned records are similar to the IBM standard variable-length records. Logical record lengths greater than the PRIMOS physical length maximum are possible, because a logical record may be continued

over two or more physical records. In other words, a 100,000-byte logical record would be written on tape as three physical tape blocks, each containing the 32,760-byte maximum, and a fourth block containing the remainder. Each piece of the logical record is called a segment. Instead of a record control word (RCW), each segment is preceded by a segment control word (SCW). The SCW contains two pieces of information: the length of the data segment, including the SCW itself, and a segment descriptor. The format of the IBM SCW is shown in Figure A-6. The Segment Control Code (SCC) within the segment descriptor indicates whether the associated segment is the first segment of the logical record, the last segment, a middle segment, or the entire logical record. Figure A-5 shows the IBM standard spanned variable-length record format.

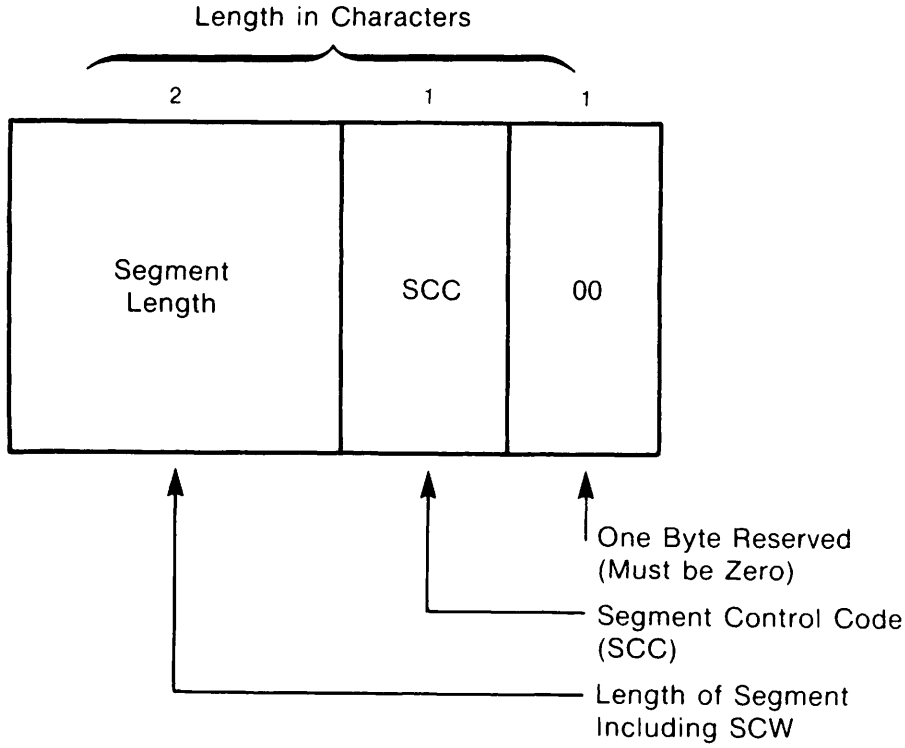
Spanned records may be blocked. However, no more than one segment of a record may appear in a block. In Figure A-5, the data segment with length J is the last segment of a record, and the data segment with length K is either the first or only segment of that record.

The SCC is one byte (eight bits) long. The first (high-order) six bits are 0 and the last two bits specify the relative position of the segment. Table A-1 lists the SCCs and the corresponding segment positions.

Like IBM variable-length records, IBM spanned records contain a block control word. The format is exactly the same as for variable-length records.



IBM Standard Spanned Variable-length Records
Figure A-5



IBM Segment Control Word (SCW)
Figure A-6

Table A-1
IBM Segment Control Codes

Code	Relative Position of Segment
00	Complete logical record
01	First segment of a multisegment record
10	Last segment of a multisegment record
11	Middle segment of a multisegment record (not the first or the last)

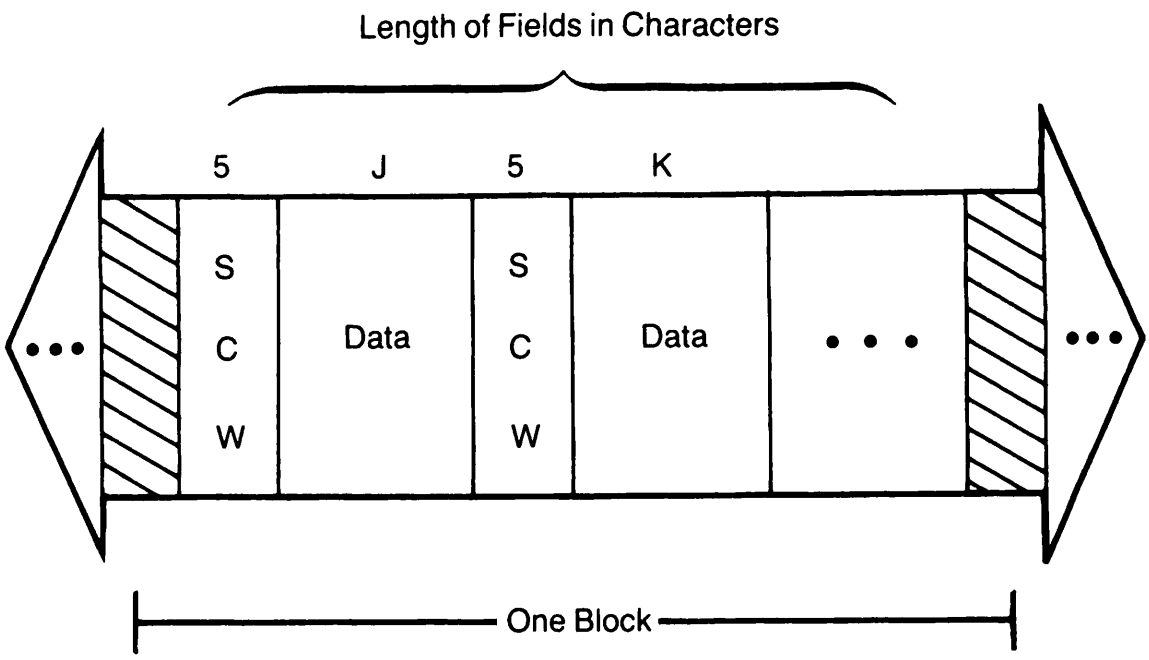
ANSI Standard Spanned Variable-length Records

This section describes ANSI standard spanned records for your information only. MAGNET does not support spanned records.

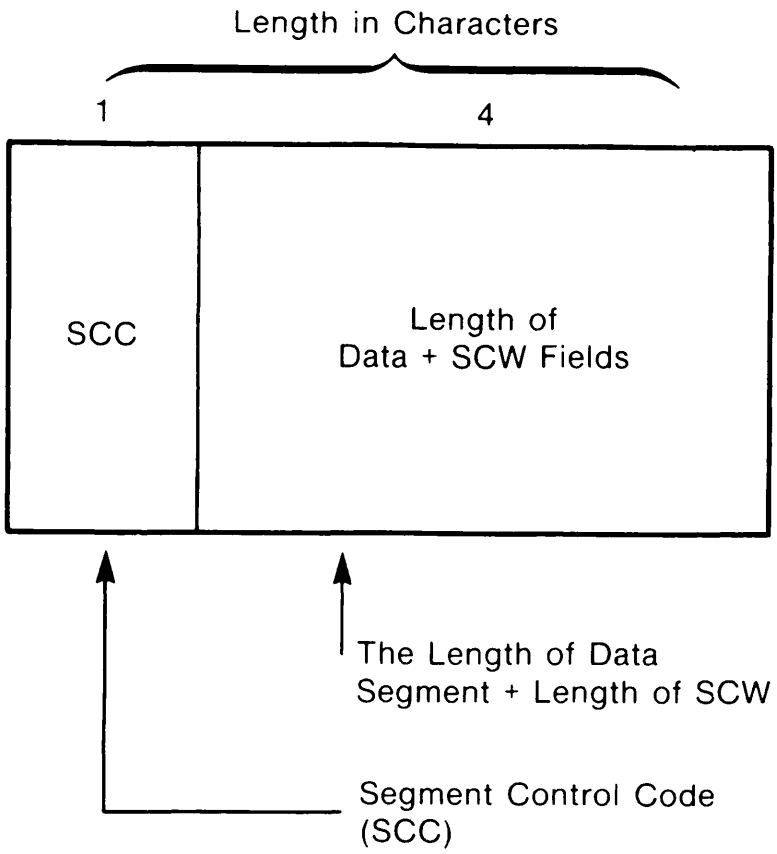
ANSI standard spanned variable-length records are similar to ANSI standard variable-length records. Logical record lengths greater than the PRIMOS physical length maximum are possible because a logical record may be continued over two or more physical records. In other words, a 100,000-byte logical record is written on tape as three physical tape blocks, each containing the 32,768-byte maximum, and a fourth block containing the remainder. Each piece of the logical record is called a segment. Instead of a record control word, each segment is preceded by a Segment Control Word (SCW). The SCW contains two pieces of information: the length of the data segment, including the SCW itself, plus a segment descriptor. The Segment Control Code (SCC) within the segment descriptor indicates whether the associated segment is the first segment of the logical record, the last segment, a middle segment, or the entire logical record. The ANSI-spanned format is very similar to the IBM-spanned format, except that there are no block control words on each physical block and the segment control word has a slightly different form. Figure A-7 shows the ANSI-spanned format.

Spanned records may be blocked. However, no more than one segment of a record may appear in a block. In Figure A-7, the data segment with length J is the last segment of a record, and the data segment with length K is either the first or only segment of that record. Note that the ANSI segment control words are each five bytes long. The format of the ANSI SCW is shown in Figure A-8.

The SCC is one byte (eight bits) long, and is a character instead of a binary number (as in the IBM SCC.) It indicates the relative position of the segment within a multisegment record. Table A-2 lists the SCCs and the corresponding segment positions.



ANSI Standard Spanned Variable-length Records
Figure A-7



ANSI Segment Control Word (SCW)
Figure A-8

Table A-2
ANSI Segment Control Codes

Code	Relative Position of Segment
0	Complete logical record
1	First segment of a multisegment record
2	Middle segment of a multisegment record (not the first or the last)
3	Last segment of a multisegment record

Prime Variable-length Format

Prime variable-length format provides to Prime FORTRAN and PL/I users a variable-length capability that is easier to use than the standard formats. In this format, a physical record is equivalent to a logical record (a blocking factor of one). Whatever is contained in a physical tape record actually becomes one logical record. No block, record, or segment control words are necessary, but the maximum record size is limited to 32,768 bytes.

B

Magnetic Tape Label Formats

This appendix describes the format of IBM and ANSI tape labels. Please note that MAGNET cannot interpret tape labels. If you wish to label your tape, use the PRIMOS LABEL command, described in Chapter 3, INITIALIZING TAPES.

INTRODUCTION

Tape labels identify a tape, and provide processing information about tape reels and the files that they contain. As with record standards, there are two tape label standards. The first is the standard of IBM. The second is an ANSI standard. The two standards are almost the same. The difference is that IBM labels are written in either 9-track EBCDIC or 7-track BCDIC, whereas ANSI labels are written in 9-track ASCII only. The other differences are in the naming or length of certain fields within a label. You have a choice of two ANSI standards: X3.27-1978 and X3.27-1987.

Labels are simply tape records. They are usually 80 characters long. The first four characters identify the label type. There are several label types:

- Volume labels - VOLn, UVLn, EOVn
- Header labels - HDRn, UHLn
- Trailer labels - EOFn, UTLn

Volume labels identify the reel of tape, the owner of the reel, and the accessibility of the reel. Other information may be stored in user fields.

Header and trailer labels identify and provide information about each file on the tape. Such information includes the file-identifier, the creation and expiration dates, block counts, and set, sequence, generation, and version numbers.

VOLUME LABELS

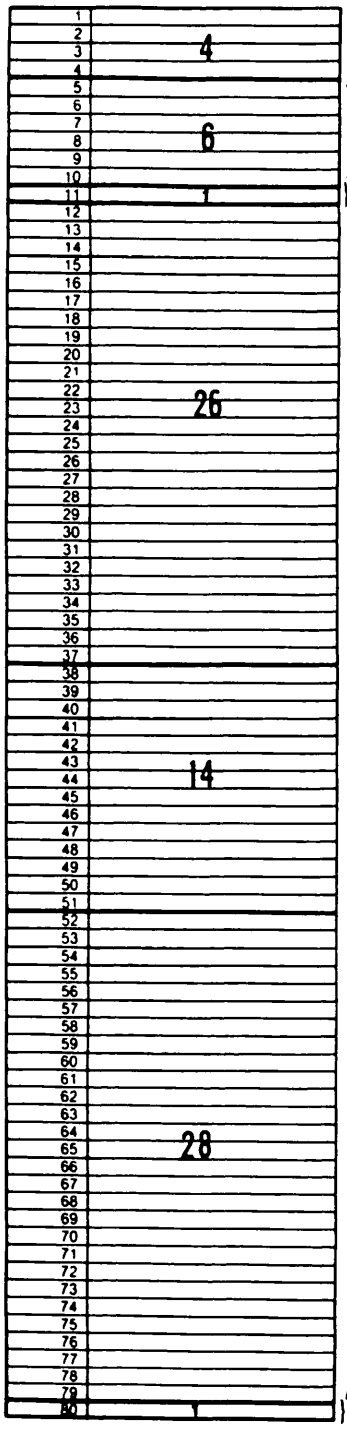
There are three types of volume label: VOLn, UVLn, and EOVn labels.

VOLn Labels

VOLn labels occur at the beginning of a reel of tape. There can be a maximum of nine of these labels, numbered VOL1 through VOL9, although certain IBM operating systems allow only up to eight of these labels (VOL1-VOL8). If the tape has any standard labels, it must have a VOL1 label as its first record. There need not be more than one VOLn label, but if there are additional VOLn labels, they must occur in sequence, VOL2, VOL3, etc. The format of the ANSI standard VOL1 label is shown in Figures B-1(a) and B-1(b). The format of the IBM standard VOL1 label is shown in Figure B-2.

Other Volume Labels, User File Header, and User File Trailer Labels

The format of IBM and ANSI VOL2 through VOL9, and the UVL1 through UVL9 labels, UHLn, and UTLn are shown in Figure B-3. Because some IBM systems limit the number of VOL and UVL labels, it is best to check first whether there is a limit when you write tapes that are targeted for IBM systems. The UVL labels immediately follow the last of the VOL labels on the tape. The VOL-UVL label group is called the volume header label group.



CONTENTS OF FIELDS

The String "VOL 1"

Volume Serial Identifier (VOLSER) — identifies a specific reel of tape

Accessibility — an installation-dependent field indicating the type of reel access. A space indicates no restrictions. Other characters may indicate read-only, execute-only, etc.

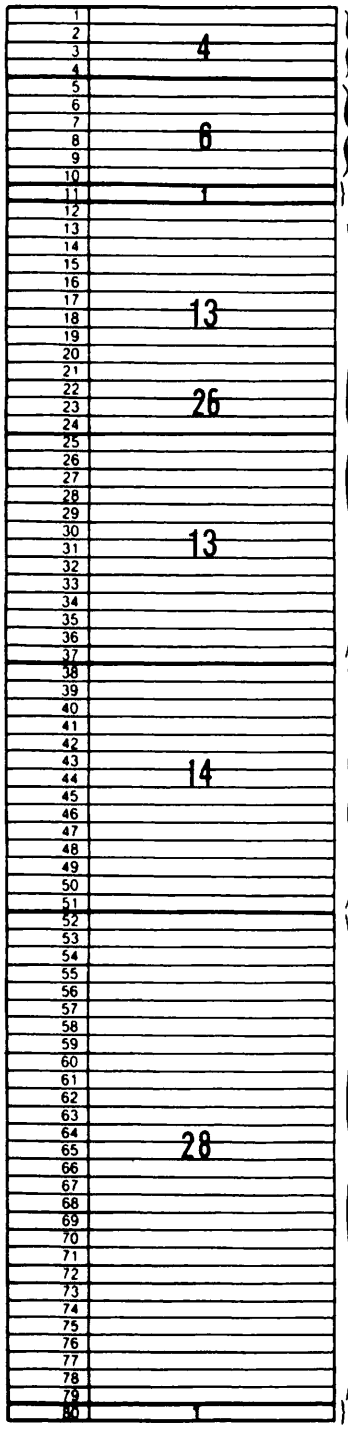
Reserved by ANSI for future information, recorded as spaces

Owner-Identification — identifies the owner of the reel

Reserved by ANSI for future information, recorded as spaces

ANSI Standard Version — recorded as a "3" to indicate that **Prime** software conforms to the X3.27-1978 standard.

ANSI Standard VOL1 Label (X3.27-1978)
Figure B-1(a)



CONTENTS OF FIELDS

The String VOL 1

Volume Serial Identifier (VOLSER) — identifies a specific reel of tape

Accessibility — an installation-dependent field indicating the type of reel access. A space indicates no restrictions. Other characters may indicate read only, execute-only, etc.

Reserved by ANSI for future information, recorded as spaces.

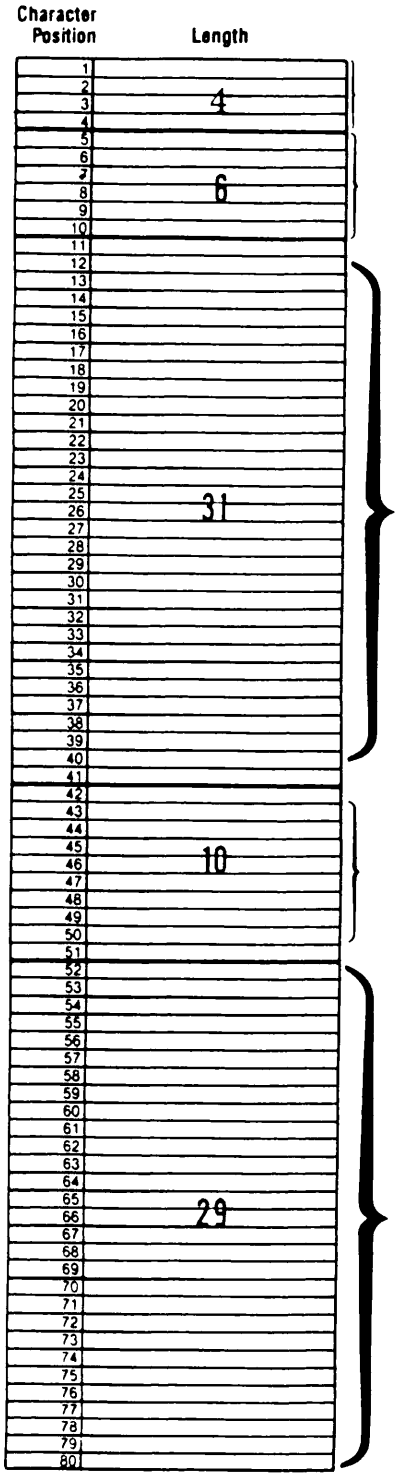
Implementation ID — identifies the implementation under which the tape was created.

Owner-Identification — identifies the owner of the reel

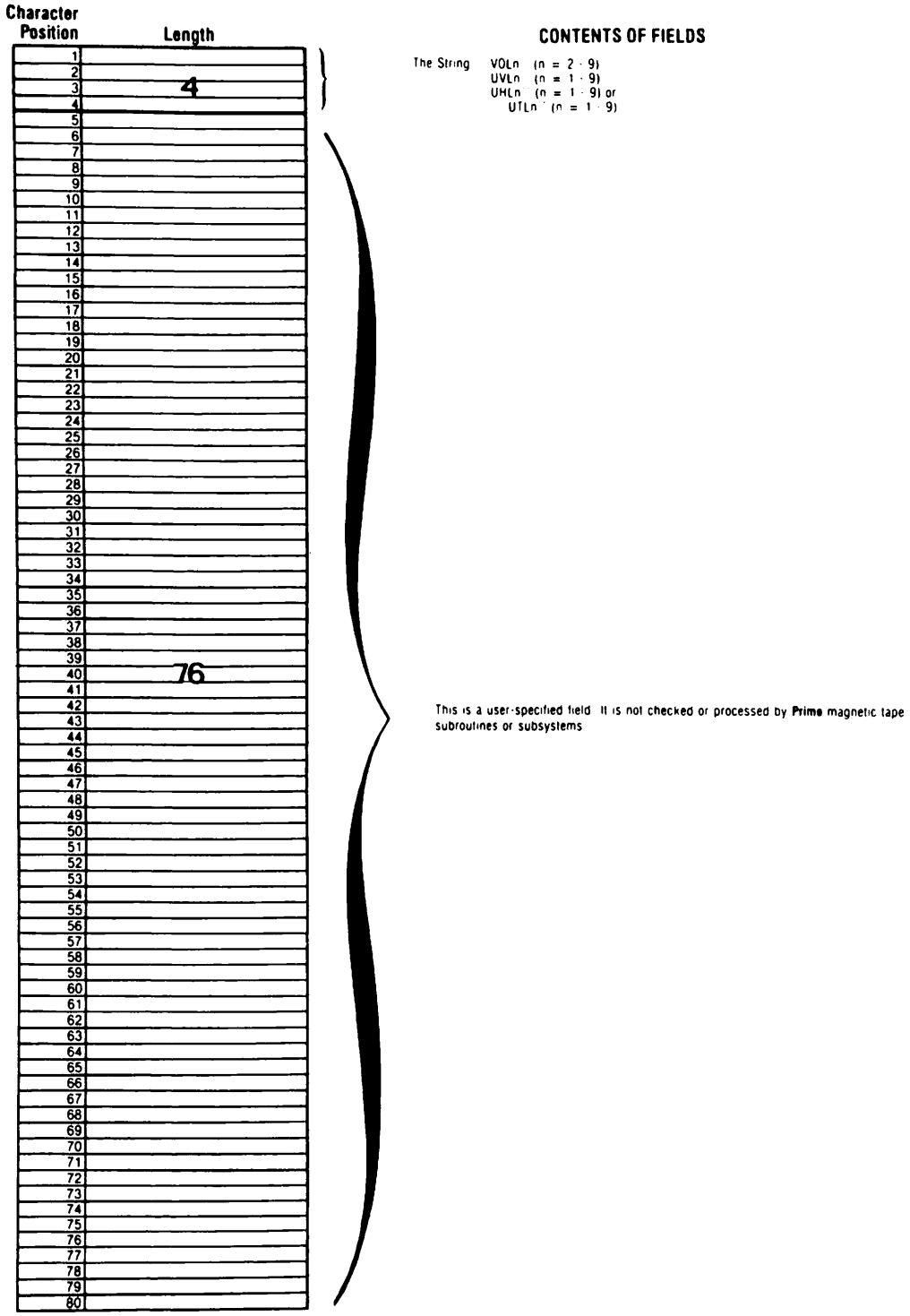
Reserved by ANSI for future information, recorded as spaces

ANSI Standard Version — recorded as a "4" to indicate that Prime software conforms to the X3.27-1987 standard.

ANSI Standard VOL1 Label (X3.27-1987)
Figure B-1(b)



IBM Standard VOL1 Label
Figure B-2



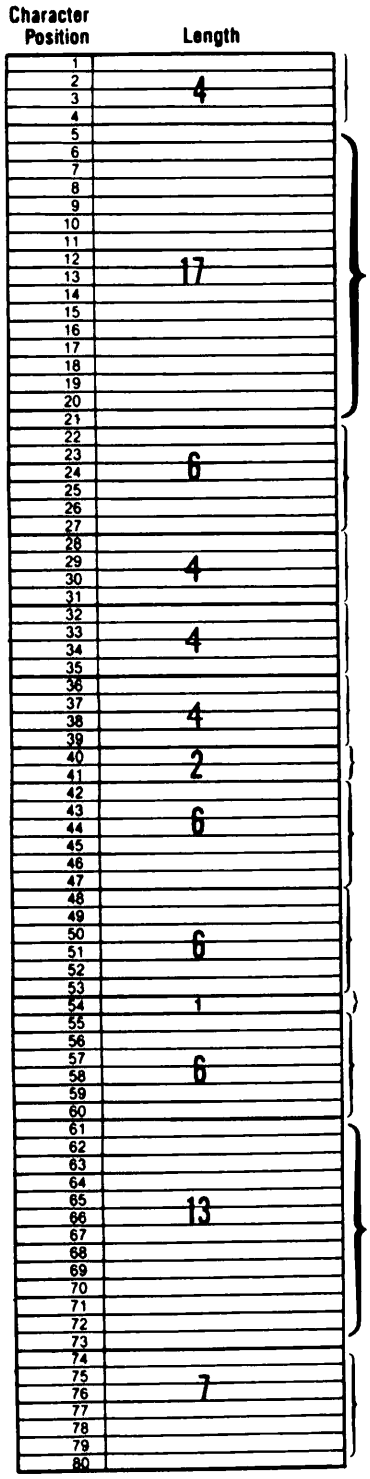
IBM and ANSI VOL2-VOL9, and UVLn, UHLn, and UTLn Labels
Figure B-3

FILE HEADER LABELS

There are two types of FILE HEADER labels: the HDRn and the UHLn labels.

A labeled tape must include at least an HDR1 label before each file. There can be a maximum of nine HDRn labels, but, just like VOLn labels, they must occur in order. If there are any UHLn labels, they must occur after the last HDRn label. A sequenced group of HDRn and UHLn labels is called a file header label group. Figure B-4 shows the format of the ANSI standard HDR1 label. Figure B-5 shows the format of the IBM standard HDR1 label. The HDR2 label contains additional information about the tape file. As stated earlier, the IBM and ANSI labels differ. Figures B-6(a) and B-6(b) show the format of the ANSI standard HDR2 label. Figure B-7 shows the format of the IBM standard HDR2 label.

All the HDRn (n = 3, 4, 5, 6, 7, 8, and 9) labels and all the UHLn labels follow the same format as the VOLn (n = 3-9) and UVLn labels. The first four characters contain the label identifier, and the other 76 characters are reserved for the user. The beginning-of-file label group is terminated by a single file marker.



CONTENTS OF FIELDS

The String "HDR1"

File-Identifier — identifies individual tape files

File-Set Identifier — the same information as that in the volume-identifier field on the VOL 1 label. For multi-volume files, it contains the volume identifier of the first tape reel of the set

File-Section Number — identifies the order of a particular reel in a multi-reel set (0001 - 9999)

File-Sequence Number — identifies the order of a file within a set of files created simultaneously ("0001" - "9999")

Generation-Number — indicates file generation or edition (0001 - 9999) or spaces)

Generation-Version — indicates file version or printing (00 - 99 or spaces)

Creation Date — the day and year the file was created. The format is Δyyddd where
Δ = a blank
yy = the year (00 - 99)
ddd = the day of the year (00 - 366)

Expiration-Date — the day and year the file expires. The format is the same as that for the creation date field

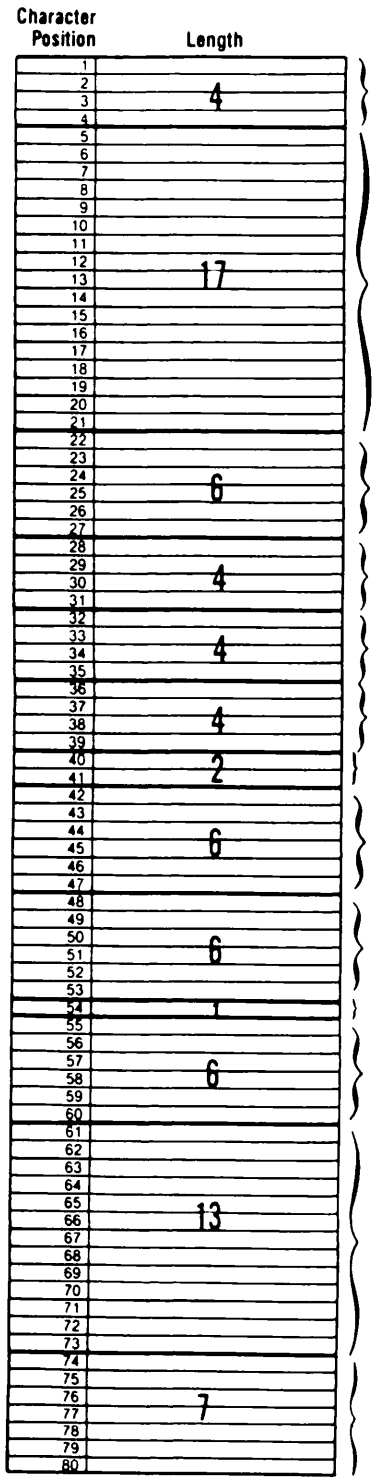
Accessibility — is installation dependent

Block Count — recorded as "000000

System Code — identifies the computing system that wrote the file. **Prime** software writes the string "PRIMOSΔΔΔΔΔΔ" where the "Δ" 's represent spaces

Reserved by ANSI for future information, recorded as spaces

ANSI Standard HDR1 Label
Figure B-4



CONTENTS OF FIELDS

The String "HDR1"

Data Set-Identifier — identifies individual tape files

Data Set Serial Number — the same information as that in the **Volume-Identifier** field on the VOL 1 label. For multi-volume files, it contains the volume identifier of the first tape reel of the set

Volume Sequence Number — identifies the order of a particular reel in a multi-reel set ("0001"-"9999")

Data Set-Sequence Number — identifies the order of a file within a set of files created simultaneously ("0001"-"9999")

Generation-Number — indicates file generation or edition ("0001"-"9999" or spaces)

Generation-Version — indicates file version or printing ("00"-"99" or spaces)

Creation Date — the day and year the file was created. The format is "bbyyddd" where
b = a blank
yy = the year ("00"-"99")
ddd = the day of the year ("001"-"366")

Expiration-Date — the day and year the file expires. The format is the same as that for the Creation Date field

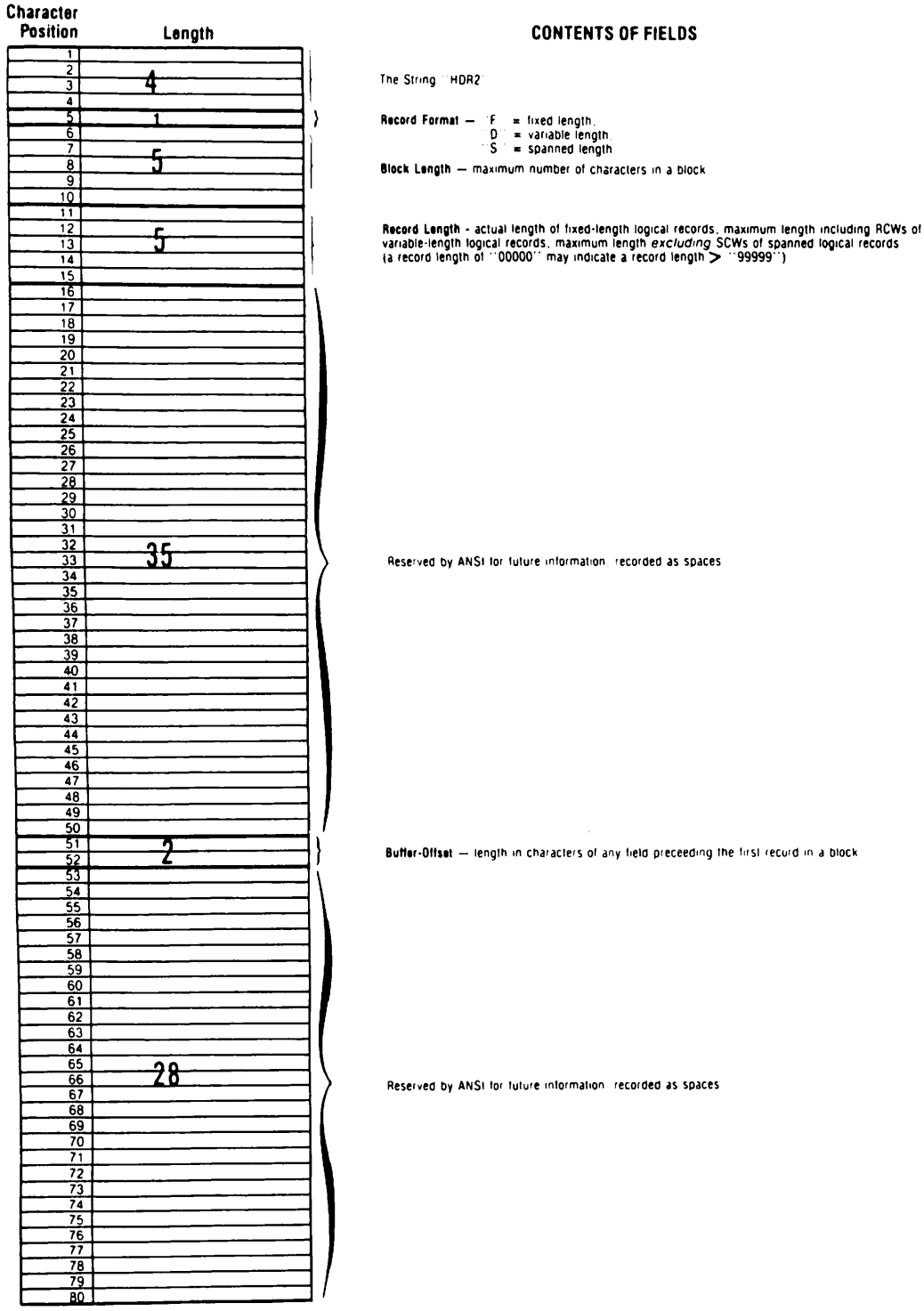
Accessibility — is installation dependent

Block Count — recorded as "000000"

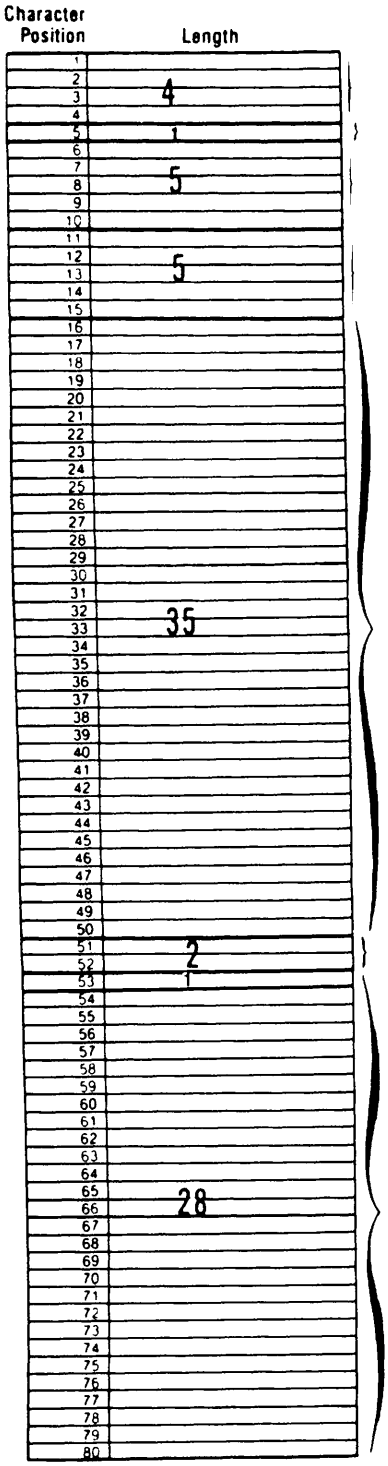
System Code — identifies the computing system that wrote the file. **Prime** software writes the string "PRIMOSbbbbbbb" where the "b"'s represent spaces

Reserved by IBM for future information, recorded as spaces

IBM Standard HDR1 Label
Figure B-5



ANSI Standard HDR2 Label (X3.27-1978)
Figure B-6(a)



CONTENTS OF FIELDS

The String "HDR2"

Record Format — F = fixed length
D = variable length
S = spanned length

Block Length — maximum number of characters in a block

Record Length - actual length of fixed-length logical records, maximum length including RCWs of variable-length logical records, maximum length *excluding* SCWs of spanned logical records (a record length of "00000" may indicate a record length > 99999)

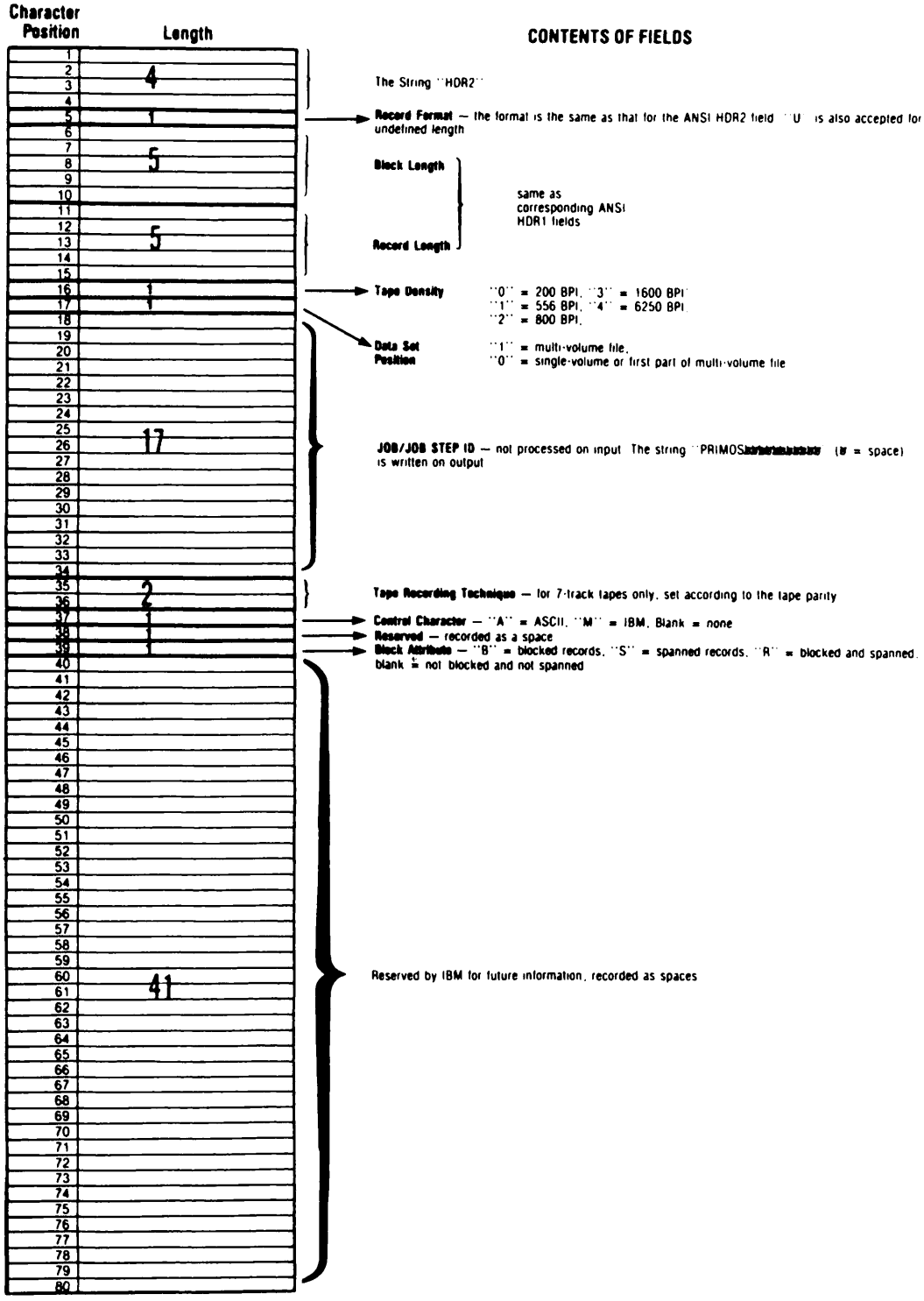
Reserved by ANSI for future information, recorded as spaces

Buffer-Offset — length in characters of any field preceeding the first record in a block

Code — space = ASCII
"A" = any other code

Reserved by ANSI for future information, recorded as spaces

ANSI Standard HDR2 Label (X3.27-1987)
Figure B-6(b)



IBM Standard HDR2 Label
Figure B-7

FILE AND VOLUME TRAILER LABELS

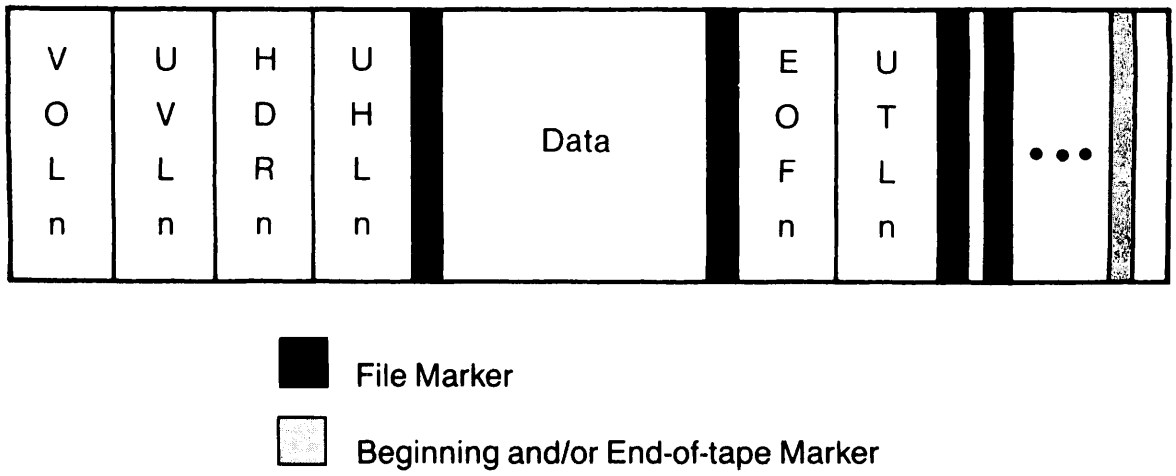
The EOF1 label follows exactly the same format as the HDR1 label, except that the block count field contains the number of blocks that were written in the file. The EOF2 label follows exactly the same format as the HDR2 label. EOF3-EOF9 and UTLn labels follow the same format as VOL3-VOL9 and UVLn labels. The first four characters identify the label type, and the other 76 characters are reserved for the user. There is no limit on the number of UHLn or UTLn labels that can be used.

The EOVn labels are written whenever it becomes necessary to mount another tape when a currently mounted tape has reached its EOT marker. Prime software does not allow EOV3-EOV9 as well as UTLn labels. On an EOT during output, the EOV1 and possibly EOV2 labels are written, followed by the appropriate number of file markers, and the tape is then rewound. The EOV1 format is the same as the EOF1 format. The EOV2 label is the same as the EOF2 label except for the EOV2 identifier.

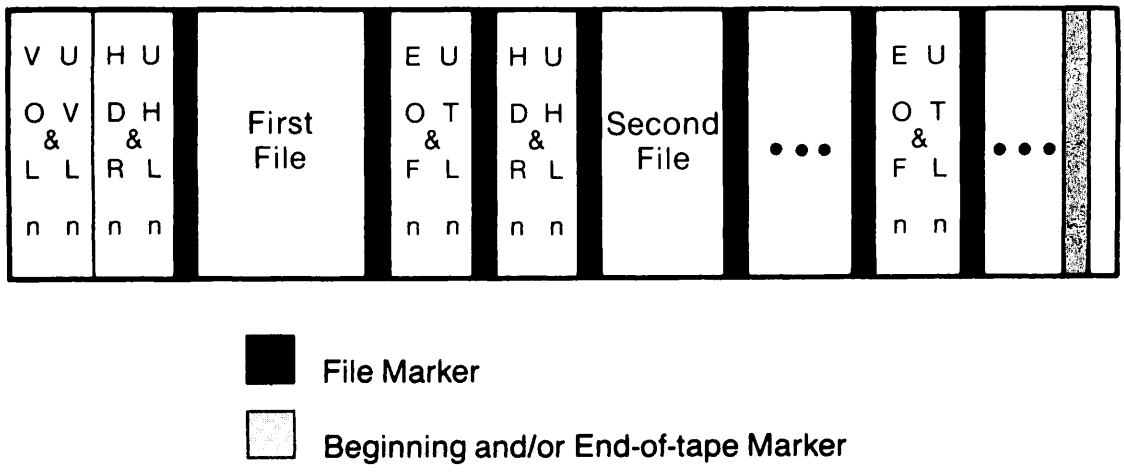
LABEL GROUPS

Tape labels are written together in groups called label groups. The first information that must appear on a labeled tape is the volume header label group. Volume labels consist of at least the VOL1 label. VOL2 through VOL9 labels may also be present followed by UVL1 through UVL9, if desired. The next information that appears on the tape is the first file's file header label group, which consists of an HDR1 label, possibly followed by HDR2 through HDR9 and any number of UHL labels. These labels are followed by a file marker, then the actual data records for the file. The file is followed by a file mark, an end-of-file label group consisting of EOF1 and possibly EOF2 through EOF9, and then any number of UTLn labels. If the file does not fit on one physical tape, the EOFn labels are replaced by EOVn labels. The last labels for a file are followed by a single file marker (unless the file is the last file on the tape and ends before the end-of-tape marker.) In that case, two marks are written. The label and tape file formats are shown in Figures B-8, B-9, B-10, and B-11.

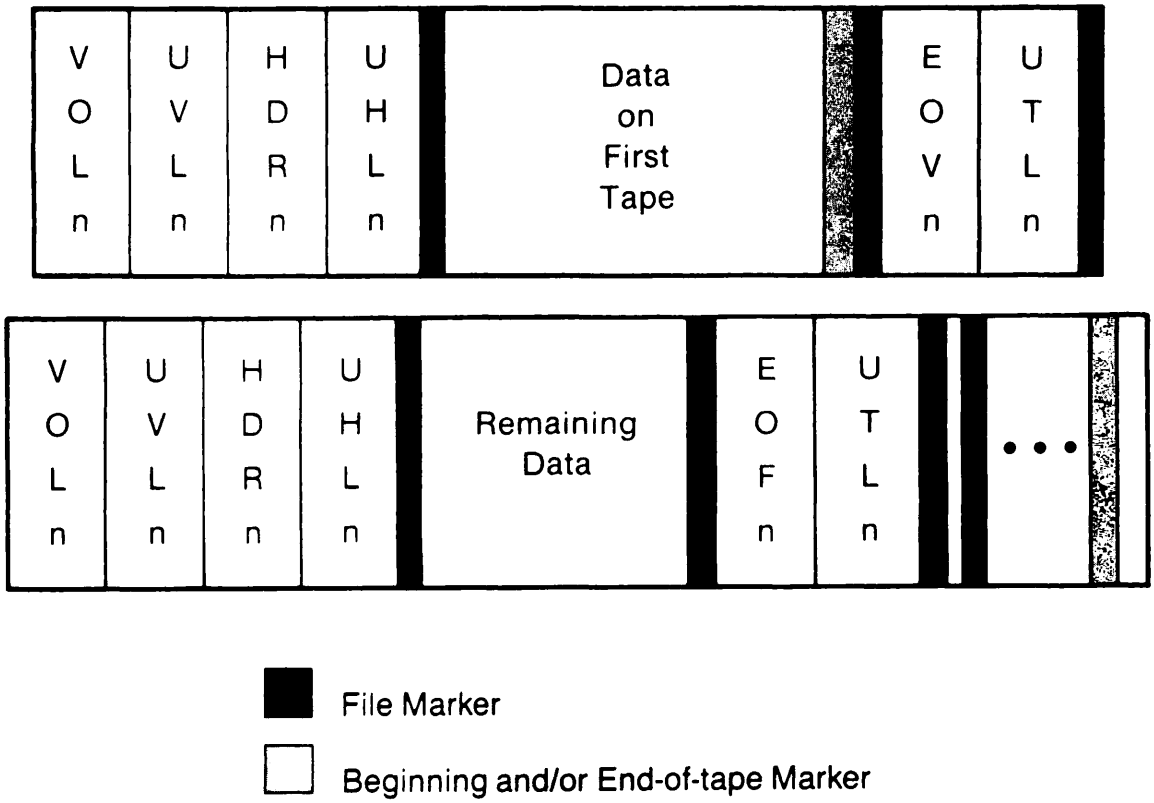
If all the file header and file trailer groups on a tape contain only HDR1 and EOF1/EOV1 labels, the tape is said to have Level 1 labels only. If the tape contains HDR2 and EOF2/EOV2 labels in addition to the Level 1 labels, the tape is said to contain Level 2 labels. A tape that contains Level 2 labels and HDRn, EOFn, EOVn n>2), UHLn, or UTLn labels is said to contain Level 3 labels. The terms Level 1 tape, Level 2 tape, and Level 3 tape are also sometimes used. A Level 4 tape may contain all Level 3 labels plus spanned records.



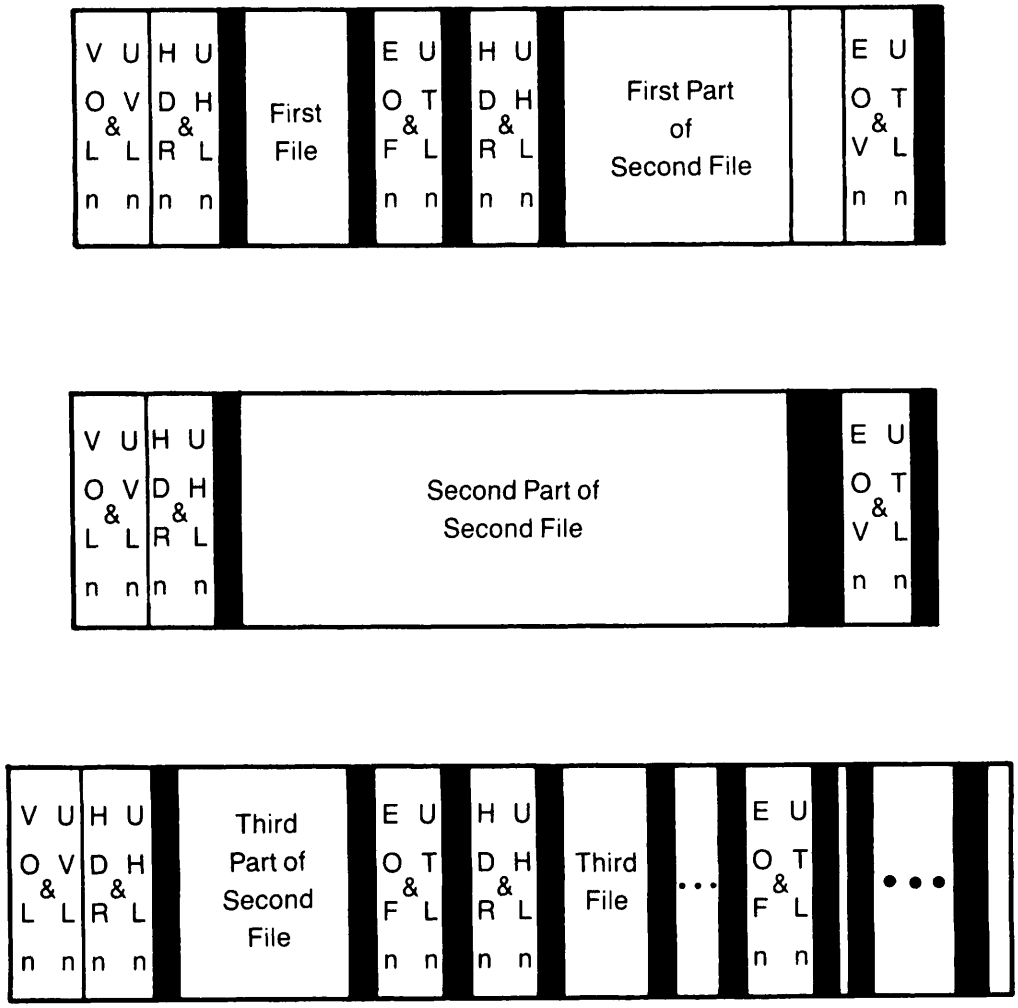
Tape Label and File Configuration -- Single Reel, Single File
Figure B-8



Tape Label and File Configuration -- Single Reel, Multiple Files
Figure B-9



Tape Label and File Configuration -- Multiple Reels, Single File
Figure B-10



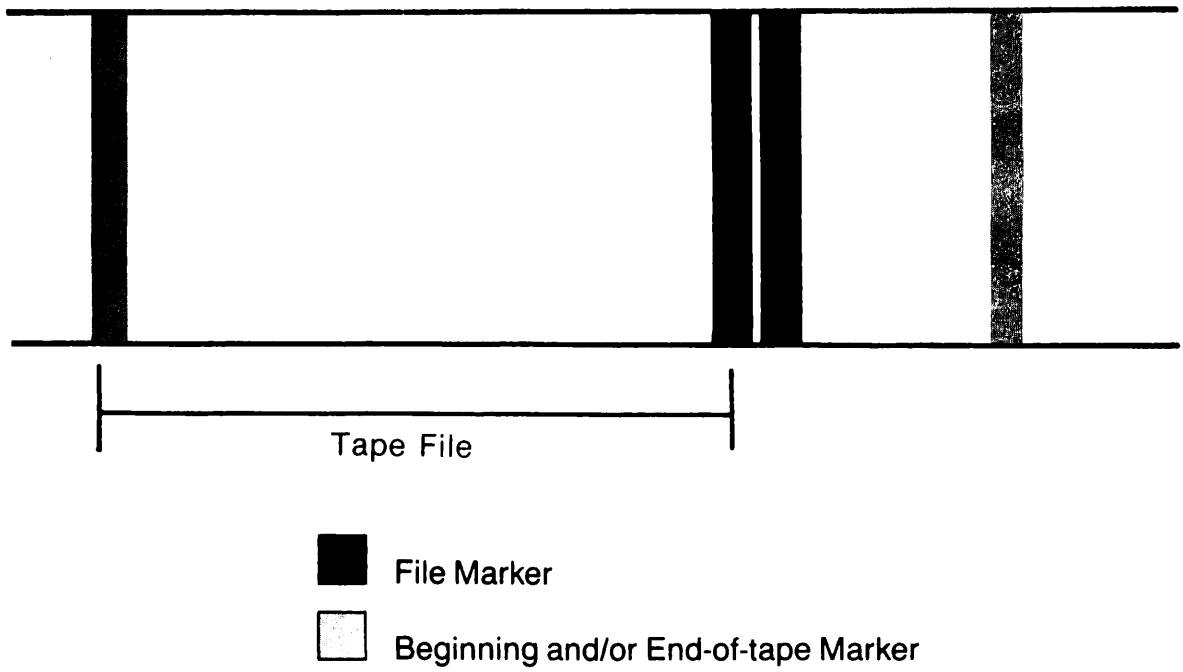
Tape Label and File Configuration -- Multiple Reels, Multiple Files
Figure B-11

UNLABELED TAPES

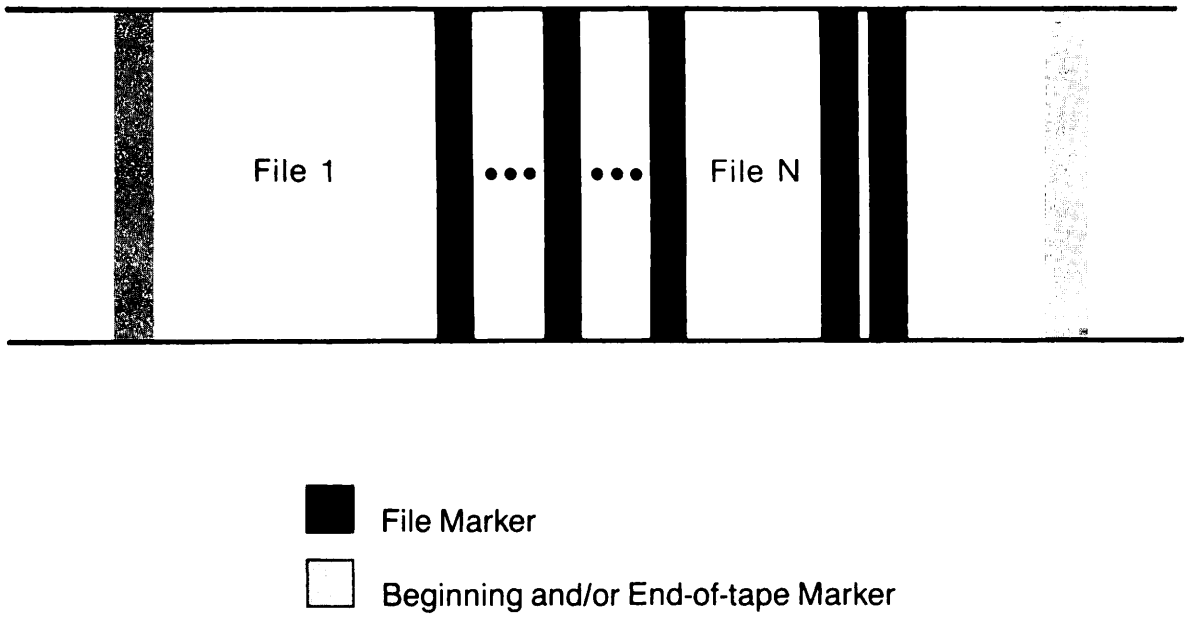
Like labeled tapes, unlabeled tapes also follow certain standard formats, but they are harder to process than labeled tapes. There are two reasons for this:

- Tape labels contain information about record formats and lengths, blocking factors, and multi-reel operations.
- Tape labels provide some protection for error checking, and they delimit data files. That is, the labels state the number of tape blocks contained in the file, and files are delimited by four file markers -- two at each end: one marker before and after the HDRn label group, and one marker at each end of the EOFn or EOVn group.

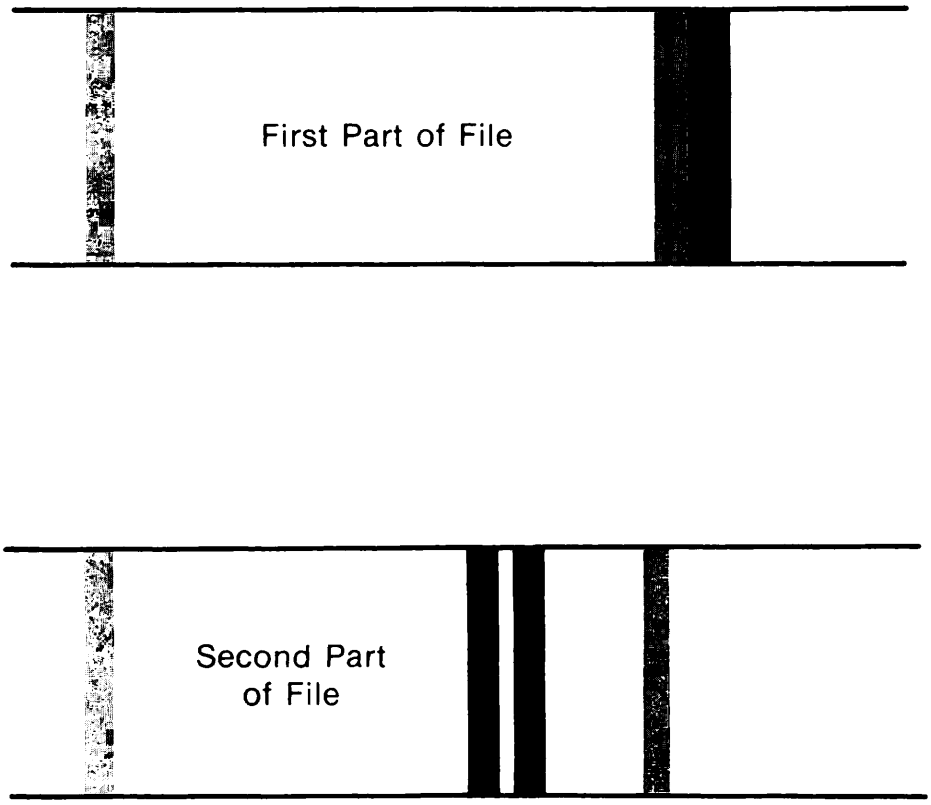
On unlabeled tapes, a single file marker signals the end of a file. If the end of a file is also the end of all the data on the tape, another file marker is written. If the end-of-tape marker is detected when writing, a single file marker is written whether or not the entire file has been written to tape. It is assumed that another tape must be mounted to complete processing. Unlabeled tape file formats are shown in Figures B-12, B-13, B-14, and B-15.



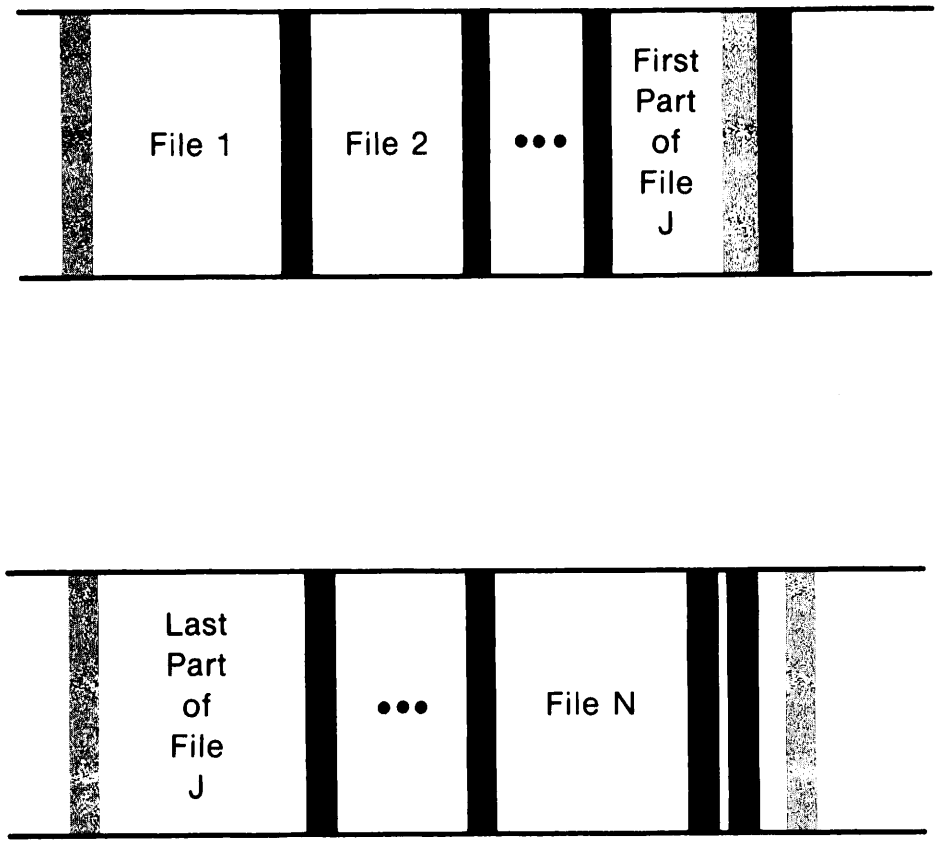
Unlabeled Tape File Configuration -- Single Reel, Single File
Figure B-12



Unlabeled Tape File Configuration -- Single Reel, Multiple Files
Figure B-13



Unlabeled Tape File Configuration -- Multiple Reels, Single File
Figure B-14



Unlabeled Tape File Configuration -- Multiple Reels, Multiple Files
Figure B-15

C

Character Set and Translation Tables

The character set tables provided in this Appendix are drawn from the following sources:

- ASCII -- American National Standards Institute standard X3.4-1977.
- BCD -- Coded Character Sets, History and Development, by C. E. Mackenzie, 1980, Addison-Wesley Publishing.
- EBCDIC -- IBM System/370 Reference Summary, Third Edition, February 1974, (Publication No. GX20-1850-2).

CHARACTER SET TABLES

Industry-standard ASCII

This table lists the industry-standard 8-bit ASCII character set (ASCII-8) and the 7-bit ASCII character set (ASCII-7). ASCII-8 runs from octal 000 through octal 377. ASCII-7 is a subset of ASCII-8, and runs from octal 000 through octal 177.

^ => Control key depressed

8-Bit Octal Code	Char		Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
000	NULL	^@	0000	040	Sp	0200
001	SOH	^A	0004	041	!	0204
002	STX	^B	0010	042	"	0210
003	ETX	^C	0014	043	#	0214
004	EOT	^D	0020	044	$	0220
005	ENQ	^E	0024	045	%	0224
006	ACK	^F	0030	046	&	0230
007	BEL	^G	0034	047	'	0234
010	BS	^H	0040	050	(	0240
011	HT	^I	0044	051	)	0244
012	LF	^J	0050	052	*	0250
013	VT	^K	0054	053	+	0254
014	FF	^L	0060	054	,	0260
015	CR	^M	0064	055	-	0264
016	SO	^N	0070	056	.	0270
017	SI	^O	0074	057	/	0274
020	DLE	^P	0100	060	0	0300
021	DC1	^Q	0104	061	1	0304
022	DC2	^R	0110	062	2	0310
023	DC3	^S	0114	063	3	0314
024	DC4	^T	0120	064	4	0320
025	NAK	^U	0124	065	5	0324
026	SYN	^V	0130	066	6	0330
027	ETB	^W	0134	067	7	0334
030	CAN	^X	0140	070	8	0340
031	EM	^Y	0144	071	9	0344
032	SUB	^Z	0150	072	:	0350
033	ESC	^[	0154	073	;	0354
034	FS	^\	0160	074	<	0360
035	GS	^]	0164	075	=	0364
036	RS	^^	0170	076	>	0370
037	US	^_	0174	077	?	0374

Industry-standard ASCII (continued)

8-Bit Octal Code	Char	Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
100	@	0400	140	`	0600
101	A	0404	141	a	0604
102	B	0410	142	b	0610
103	C	0414	143	c	0614
104	D	0420	144	d	0620
105	E	0424	145	e	0624
106	F	0430	146	f	0630
107	G	0434	147	g	0634
110	H	0440	150	h	0640
111	I	0444	151	i	0644
112	J	0450	152	j	0650
113	K	0454	153	k	0654
114	L	0460	154	l	0660
115	M	0464	155	m	0664
116	N	0470	156	n	0670
117	O	0474	157	o	0674
120	P	0500	160	p	0700
121	Q	0504	161	q	0704
122	R	0510	162	r	0710
123	S	0504	163	s	0714
124	T	0520	164	t	0720
125	U	0524	165	u	0724
126	V	0530	166	v	0730
127	W	0534	167	w	0734
130	X	0540	170	x	0740
131	Y	0544	171	y	0744
132	Z	0550	172	z	0750
133	[	0554	173	{	0754
134	\	0560	174	|	0760
135	]	0564	175	}	0764
136	^	0570	176	~	0770
137	_	0574	177	DEL	0774

Industry-standard ASCII (continued)

8-Bit Octal Code	Char	Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
200	RES1	1000	240	NBSP	1200
201	RES2	1004	241	INVE	1204
202	RES3	1010	242	CENT	1210
203	RES4	1014	243	PND	1214
204	IND	1020	244	CURR	1220
205	NEL	1024	245	YEN	1224
206	SSA	1030	246	BBAR	1230
207	ESA	1034	247	SECT	1234
210	HTS	1040	250	DIA	1240
211	HTJ	1044	251	COPY	1244
212	VTS	1050	252	FOI	1250
213	PLD	1054	253	LAQM	1254
214	PLU	1060	254	NOT	1260
215	RI	1064	255	SHY	1264
216	SS2	1070	256	TM	1270
217	SS3	1074	257	MACN	1274
220	DCS	1100	260	DEGR	1300
221	PU1	1104	261	PLMI	1304
222	PU2	1110	262	SPS2	1310
223	STS	1114	263	SPS3	1314
224	CCH	1120	264	AAC	1320
225	MW	1124	265	LCMU	1324
226	SPA	1130	266	PARA	1330
227	EPA	1134	267	MIDD	1334
230	RES5	1140	270	CED	1340
231	RES6	1144	271	SPS1	1344
232	RES7	1150	272	MOI	1350
233	CSI	1154	273	RAQM	1354
234	ST	1160	274	FR14	1360
235	OSC	1164	275	FR12	1364
236	PM	1170	276	FR34	1370
237	APC	1174	277	INVQ	1374

Industry-standard ASCII (continued)

8-Bit Octal Code	Char	Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
300	UCAG	1400	340	LCAG	1600
301	UCAA	1404	341	LCAA	1604
302	UCAC	1410	342	LCAC	1610
303	UCAT	1414	343	LCAT	1614
304	UCAD	1420	344	LCAD	1620
305	UCAR	1424	345	LCAR	1624
306	UCAE	1430	346	LCAE	1630
307	UCCC	1434	347	LCCC	1634
310	UCEG	1440	350	LCEG	1640
311	UCEA	1444	351	LCEA	1644
312	UCEC	1450	352	LCEC	1650
313	UCED	1454	353	LCED	1654
314	UCIG	1460	354	LCIG	1660
315	UCIA	1464	355	LCIA	1664
316	UCIC	1470	356	LCIC	1670
317	UCID	1474	357	LCID	1674
320	UETH	1500	360	LETH	1700
321	UCNT	1504	361	LCNT	1704
322	UCOG	1510	362	LCOG	1710
323	UCOA	1514	363	LCOA	1714
324	UCOC	1520	364	LCOC	1720
325	UCOT	1524	365	LCOT	1724
326	UCOD	1530	366	LCOD	1730
327	MULT	1534	367	DIV	1734
330	UCOO	1540	370	LCOO	1740
331	UCUG	1544	371	LCUG	1744
332	UCUA	1550	372	LCUA	1750
333	UCUC	1554	373	LCUC	1754
334	UCUD	1560	374	LCUD	1760
335	UCYA	1564	375	LCYA	1764
336	UTHN	1570	376	LTHN	1770
337	LGSS	1574	377	LCYD	1774

Prime ASCII

This table lists the Prime Extended Character Set (Prime ECS) and the Prime 7-bit ASCII character set (Prime-7). Prime ECS runs from octal 000 through octal 377. Prime-7 is a subset of Prime ECS, and runs from octal 200 through octal 377.

F => Valid filename character
R => Reserved command line character
^ => Control key depressed

8-Bit Octal Code	Char	Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
000	RES1	0000	040	NBSP	0200
001	RES2	0004	041	INVE	0204
002	RES3	0010	042	CENT	0210
003	RES4	0014	043	PND	0214
004	IND	0020	044	CURR	0220
005	NEL	0024	045	YEN	0224
006	SSA	0030	046	BBAR	0230
007	ESA	0034	047	SECT	0234
010	HTS	0040	050	DIA	0240
011	HTJ	0044	051	COPY	0244
012	VTS	0050	052	FOI	0250
013	PLD	0054	053	LAQM	0254
014	PLU	0060	054	NOT	0260
015	RI	0064	055	SHY	0264
016	SS2	0070	056	TM	0270
017	SS3	0074	057	MACN	0274
020	DCS	0100	060	DEGR	0300
021	PU1	0104	061	PLMI	0304
022	PU2	0110	062	SPS2	0310
023	STS	0114	063	SPS3	0314
024	CCH	0120	064	AAC	0320
025	MW	0124	065	LCMU	0324
026	SPA	0130	066	PARA	0330
027	EPA	0134	067	MIDD	0334
030	RES5	0140	070	CED	0340
031	RES6	0144	071	SPS1	0344
032	RES7	0150	072	MOI	0350
033	CSI	0154	073	RAQM	0354
034	ST	0160	074	FR14	0360
035	OSC	0164	075	FR12	0364
036	PM	0170	076	FR34	0370
037	APC	0174	077	INVQ	0374

Prime ASCII (continued)

8-Bit Octal Code	Char	Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
100	UCAG	0400	140	LCAG	0600
101	UCAA	0404	141	LCAA	0604
102	UCAC	0410	142	LCAC	0610
103	UCAT	0414	143	LCAT	0614
104	UCAD	0420	144	LCAD	0620
105	UCAR	0424	145	LCAR	0624
106	UCAE	0430	146	LCAE	0630
107	UCCC	0434	147	LCCC	0634
110	UCEG	0440	150	LCEG	0640
111	UCEA	0444	151	LCEA	0644
112	UCEC	0450	152	LCEC	0650
113	UCED	0454	153	LCED	0654
114	UCIG	0460	154	LCIG	0660
115	UCIA	0464	155	LCIA	0664
116	UCIC	0470	156	LCIC	0670
117	UCID	0474	157	LCID	0674
120	UETH	0500	160	LETH	0700
121	UCNT	0504	161	LCNT	0704
122	UCOG	0510	162	LCOG	0710
123	UCOA	0504	163	LCOA	0714
124	UCOC	0520	164	LCOC	0720
125	UCOT	0524	165	LCOT	0724
126	UCOD	0530	166	LCOD	0730
127	MULT	0534	167	DIV	0734
130	UCOO	0540	170	LCOO	0740
131	UCUG	0544	171	LCUG	0744
132	UCUA	0550	172	LCUA	0750
133	UCUC	0554	173	LCUC	0754
134	UCUD	0560	174	LCUD	0760
135	UCYA	0564	175	LCYA	0764
136	UTHN	0570	176	LTHN	0770
137	LGSS	0574	177	LCYD	0774

Prime ASCII (continued)

8-Bit Octal Code	Char		Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
200	NULL	^@	1000	240	Sp	1200
201	SOH	^A	1004	241	!	1204 R
202	STX	^B	1010	242	"	1210 R
203	ETX	^C	1014	243	#	1214 F
204	EOT	^D	1020	244	$	1220 F
205	ENQ	^E	1024	245	%	1224 R
206	ACK	^F	1030	246	&	1230 FR
207	BEL	^G	1034	247	'	1234 R
210	BS	^H	1040	250	(	1240 R
211	HT	^I	1044	251	)	1244 R
212	LF	^J	1050	252	*	1250 F
213	VT	^K	1054	253	+	1254
214	FF	^L	1060	254	,	1260 R
215	CR	^M	1064	255	-	1264 FR
216	SO	^N	1070	256	.	1270 F
217	SI	^O	1074	257	/	1274 F
220	DLE	^P	1100	260	0	1300 F
221	DC1	^Q	1104	261	1	1304 F
222	DC2	^R	1110	262	2	1310 F
223	DC3	^S	1114	263	3	1314 F
224	DC4	^T	1120	264	4	1320 F
225	NAK	^U	1124	265	5	1324 F
226	SYN	^V	1130	266	6	1330 F
227	ETB	^W	1134	267	7	1334 F
230	CAN	^X	1140	270	8	1340 F
231	EM	^Y	1144	271	9	1344 F
232	SUB	^Z	1150	272	:	1350 R
233	ESC	^[	1154	273	;	1354 R
234	FS	^\	1160	274	<	1360
235	GS	^]	1164	275	=	1364 R
236	RS	^^	1170	276	>	1370
237	US	^_	1174	277	?	1374

Prime ASCII (continued)

8-Bit Octal Code	Char	Octal Code When in Left Byte	8-Bit Octal Code	Char	Octal Code When in Left Byte
300	@	1400 R	340	`	1600 R
301	A	1404 F	341	a	1604
302	B	1410 F	342	b	1610
303	C	1414 F	343	c	1614
304	D	1420 F	344	d	1620
305	E	1424 F	345	e	1624
306	F	1430 F	346	f	1630
307	G	1434 F	347	g	1634
310	H	1440 F	350	h	1640
311	I	1444 F	351	i	1644
312	J	1450 F	352	j	1650
313	K	1454 F	353	k	1654
314	L	1460 F	354	l	1660
315	M	1464 F	355	m	1664
316	N	1470 F	356	n	1670
317	O	1474 F	357	o	1674
320	P	1500 F	360	p	1700
321	Q	1504 F	361	q	1704
322	R	1510 F	362	r	1710
323	S	1514 F	363	s	1714
324	T	1520 F	364	t	1720
325	U	1524 F	365	u	1724
326	V	1530 F	366	v	1730
327	W	1534 F	367	w	1734
330	X	1540 F	370	x	1740
331	Y	1544 F	371	y	1744
332	Z	1550 F	372	z	1750
333	[	1554 R	373	{	1754 R
334	\	1560 R	374	|	1760
335	]	1564 R	375	}	1764 R
336	^	1570 R	376	~	1770 R
337	_	1574 F	377	DEL	1774

EBCDIC

Decimal	Octal	Hex.	Char.	Decimal	Octal	Hex.	Char.
000	000	00	NULL	032	040	20	DS
001	001	01	SOH	033	041	21	SOS
002	002	02	STX	034	042	22	FS
003	003	03	ETX	035	043	23	
004	004	04	PF	036	044	24	BYP
005	005	05	HT	037	045	25	LF
006	006	06	LC	038	046	26	ETB
007	007	07	DEL	039	047	27	ESC
008	010	08		040	050	28	
009	011	09		041	051	29	
010	012	0A	SMM	042	052	2A	SM
011	013	0B	VT	043	053	2B	CU2
012	014	0C	FF	044	054	2C	
013	015	0D	CR	045	055	2D	ENQ
014	016	0E	SO	046	056	2E	ACK
015	017	0F	SI	047	057	2F	BEL
016	020	10	DLE	048	060	30	
017	021	11	DC1	049	061	31	
018	022	12	DC2	050	062	32	SYN
019	023	13	TM	051	063	33	
020	024	14	RES	052	064	34	PN
021	025	15	NL	053	065	35	RS
022	026	16	BS	054	066	36	UC
023	027	17	IL	055	067	37	EOT
024	030	18	CAN	056	070	38	
025	031	19	EM	057	071	39	
026	032	1A	CC	058	072	3A	
027	033	1B	CU1	059	073	3B	CU3
028	034	1C	IFS	060	074	3C	DC4
029	035	1D	IGS	061	075	3D	NAK
030	036	1E	IRS	062	076	3E	
031	037	1F	IUS	063	077	3F	SUB

Blanks in the character column represent characters unsupported by Prime. All unsupported characters are translated into the question mark (?).

EBCDIC (continued)

Decimal	Octal	Hex.	Char.	Decimal	Octal	Hex.	Char.
064	100	40	SP	096	140	60	-
065	101	41		097	141	61	/
066	102	42		098	142	62	
067	103	43		099	143	63	
068	104	44		100	144	64	
069	105	45		101	145	65	
070	106	46		102	146	66	
071	107	47		103	147	67	
072	110	48		104	150	68	
073	111	49		105	151	69	
074	112	4A	cent	106	152	6A	|
075	113	4B	.	107	153	6B	,
076	114	4C	<	108	154	6C	%
077	115	4D	(	109	155	6D	_
078	116	4E	+	110	156	6E	>
079	117	4F		111	157	6F	?
080	120	50	&	112	160	70	
081	121	51		113	161	71	
082	122	52		114	162	72	
083	123	53		115	163	73	
084	124	54		116	164	74	
085	125	55		117	165	75	
086	126	56		118	166	76	
087	127	57		119	167	77	
088	130	58		120	170	78	
089	131	59		121	171	79	`
090	132	5A	!	122	172	7A	:
091	133	5B	$	123	173	7B	#
092	134	5C	*	124	174	7C	@
093	135	5D	)	125	175	7D	'
094	136	5E	;	126	176	7E	=
095	137	5F		127	177	7F	"

Blanks in the character column represent characters unsupported by Prime. All unsupported characters are translated into the question mark (?).

EBCDIC (continued)

Decimal	Octal	Hex.	Char.	Decimal	Octal	Hex.	Char.
128	200	80		160	240	A0	
129	201	81	a	161	241	A1	~
130	202	82	b	162	242	A2	s
131	203	83	c	163	243	A3	t
132	204	84	d	164	244	A4	u
133	205	85	e	165	245	A5	v
134	206	86	f	166	246	A6	w
135	207	87	g	167	247	A7	x
136	210	88	h	168	250	A8	y
137	211	89	i	169	251	A9	z
138	212	8A		170	252	AA	
139	213	8B		171	253	AB	
140	214	8C		172	254	AC	
141	215	8D		173	255	AD	
142	216	8E		174	256	AE	
143	217	8F		175	257	AF	
144	220	90		176	260	B0	
145	221	91	j	177	261	B1	
146	222	92	k	178	262	B2	
147	223	93	l	179	263	B3	
148	224	94	m	180	264	B4	
149	225	95	n	181	265	B5	
150	226	96	o	182	266	B6	
151	227	97	p	183	267	B7	
152	230	98	q	184	270	B8	
153	231	99	r	185	271	B9	
154	232	9A		186	272	BA	
155	233	9B		187	273	BB	
156	234	9C		188	274	BC	
157	235	9D		189	275	BD	
158	236	9E		190	276	BE	
159	237	9F		191	277	BF	

Blanks in the character column represent characters unsupported by Prime. All unsupported characters are translated into the question mark (?).

EBCDIC (continued)

Decimal	Octal	Hex.	Char.	Decimal	Octal	Hex.	Char.
192	300	C0	{	224	340	E0	\
193	301	C1	A	225	341	E1	
194	302	C2	B	226	342	E2	S
195	303	C3	C	227	343	E3	T
196	304	C4	D	228	344	E4	U
197	305	C5	E	229	345	E5	V
198	306	C6	F	230	346	E6	W
199	307	C7	G	231	347	E7	X
200	310	C8	H	232	350	E8	Y
201	311	C9	I	233	351	E9	Z
202	312	CA		234	352	EA	
203	313	CB		235	353	EB	
204	314	CC		236	354	EC	
205	315	CD		237	355	ED	
206	316	CE	yen	238	356	EE	
207	317	CF		239	357	EF	
208	320	D0	}	240	360	F0	0
209	321	D1	J	241	361	F1	1
210	322	D2	K	242	362	F2	2
211	323	D3	L	243	363	F3	3
212	324	D4	M	244	364	F4	4
213	325	D5	N	245	365	F5	5
214	326	D6	O	246	366	F6	6
215	327	D7	P	247	367	F7	7
216	330	D8	Q	248	370	F8	8
217	331	D9	R	249	371	F9	9
218	332	DA		250	392	FA	
219	333	DB		251	373	FB	
220	334	DC		252	374	FC	
221	335	DD		253	375	FD	
222	336	DE		254	376	FE	
223	337	DF		255	377	FF	

Blanks in the character column represent characters unsupported by Prime. All unsupported characters are translated into the question mark (?).

BCD

Decimal	Octal	Hex.	Char.	Decimal	Octal	Hex.	Char.
000	000	00	0	032	040	20	-
001	001	01	1	033	041	21	J
002	002	02	2	034	042	22	K
003	003	03	3	035	043	23	L
004	004	04	4	036	044	24	M
005	005	05	5	037	045	25	N
006	006	06	6	038	046	26	O
007	007	07	7	039	047	27	P
008	010	08	8	040	050	28	Q
009	011	09	9	041	051	29	R
010	012	0A		042	052	2A	!
011	013	0B	=	043	053	2B	$
012	014	0C	'	044	054	2C	*
013	015	0D	:	045	055	2D	]
014	016	0E	>	046	056	2E	;
015	017	0F		047	057	2F	(
016	020	10	spc	048	060	30	+
017	021	11	/	049	061	31	A
018	022	12	S	050	062	32	B
019	023	13	T	051	063	33	C
020	024	14	U	052	064	34	D
021	025	15	V	053	065	35	E
022	026	16	W	054	066	36	F
023	027	17	X	055	067	37	G
024	030	18	Y	056	070	38	H
025	031	19	Z	057	071	39	I
026	032	1A		058	072	3A	?
027	033	1B	,	059	073	3B	.
028	034	1C		060	074	3C	)
029	035	1D		061	075	3D	[
030	036	1E	\	062	076	3E	<
031	037	1F		063	077	3F	

Blanks in the character column represent characters unsupported by Prime. All unsupported characters are translated into spaces (spc).

TRANSLATION TABLES

BCD to Prime ECS

Binary	Octal	Decimal	Hex.	Characters
10110000 10110001	130261	45233	B0B1	0 1
10110010 10110011	131263	45747	B2B3	2 3
10110100 10110101	132265	46261	B4B5	4 5
10110110 10110111	133267	46775	B6B7	6 7
10111000 10111001	134271	47289	B8B9	8 9
00000000 10111101	000275	00189	00BD	=
10100111 10111010	123672	42938	A7BA	' :
10111110 00000000	137000	38630	BE00	>
10100000 10101111	120257	41135	A0AF	spc /
11010011 11010100	151724	54228	D3D4	S T
11010101 11010110	152726	54742	D5D6	U V
11010111 11011000	153730	55256	D7D8	W X
11011001 11011010	154732	55770	D9DA	Y Z
00000000 10101100	000254	00172	00AC	,
10101000 00000000	124000	43008	A800	(
11011100 00000000	156000	56618	DC00	\
10101101 11001010	126712	44490	ADCA	- J
11001011 11001100	145714	52172	CBCC	K L
11001101 11001110	146716	52686	CDCE	M N
11001111 11010000	147720	53200	CFD0	O P
11010001 11010010	150722	53714	D1D2	Q R
10100001 10100100	120644	41380	A1A4	! $
10101010 11011101	125335	43741	AADD	*]
10111011 00000000	135400	47872	BB00	;
10101011 11000001	125701	43969	ABC1	+ A
11000010 11000011	141303	49859	C2C3	B C
11000100 11000101	142305	50373	C4C5	D E
11000110 11000111	143307	50887	C6C7	F G
11001000 11001001	144311	51401	C8C9	H I
10111111 10101110	137656	49070	BFAE	? .
10101001 11011011	124733	43483	A9DB	) [
10111100 00000000	136000	48128	BC00	<
00000000 00000000	000000	00000	0000	(No lowercase or control characters. This line repeats 95 additional times.)

Blanks in the character column represent characters unsupported by Prime. All unsupported characters are translated into spaces (spc).

Prime ECS to BCD

Binary	Octal	Decimal	Hex.	Characters
00010000 00010000	010020	04112	1010	spc (No extended, lower case, or control characters. This line repeats 79 additional times.)
00010000 00101010	010052	04138	102A	spc !
00010000 00010000	010020	04112	1010	" #
00101010 00010000	025020	10768	2B10	$ %
00010000 00001100	010014	04208	100C	& '
00011100 00111100	016074	07228	1C3C	( )
00101100 00110000	02060	11312	2C3D	* +
00011011 00100000	015440	06944	1B20	, -
00111011 00010001	035421	15121	3B11	. /
00000000 00000001	000001	00001	0001	0 1
00000010 00000011	001003	00515	0203	2 3
00000100 00000101	002005	01029	0405	4 5
00000110 00000111	003007	01543	0607	6 7
00001000 00001001	004011	02057	0809	8 9
00001101 00101110	006456	03374	0D2E	: ;
00111110 00001011	037013	15883	3E0B	< =
00001110 00111010	007072	03642	0E3A	> ?
00010000 00110001	010061	04145	1031	@ A
00110010 00110011	031063	12851	3233	B C
00110100 00110101	032065	13365	3435	D E
00110110 00110111	033067	13879	3637	F G
00111000 00111001	034071	14393	3839	H I
00100001 00100010	020442	08482	2122	J K
00100011 00100100	021444	08996	2324	L M
00100101 00100110	022446	09510	2526	N O
00100111 00101000	023450	10024	2728	P Q
00101001 00010010	024422	10514	2912	R S
00010011 00010100	011424	04884	1314	T U
00010101 00010110	012426	05398	1516	V W
00010111 00011000	013430	05912	1718	X Y
00011001 00111101	014475	06821	193D	Z [
00011110 00101101	017055	07725	1E2D	\ ]
00010000 00010000	010020	04112	1010	^ _
00010000 00010000	010020	04112	1010	spc (No lowercase or control characters. This line repeats 15 additional times.)

Unsupported characters are translated into spaces (spc).

Prime ECS to EBCDIC

Binary	Octal	Decimal	Hex.	Positions		Characters
01101111 01101111	067557	28527	6F6F	000	001	? ? (No extended characters. This line repeats 16 times.)
01001010 01101111	045157	19055	4ABF	034	035	cent ?
01101111 11001110	067716	28622	6FCE	036	037	? yen
01101111 01101111	067557	28527	6F6F	038	039	? ? (No extended characters. This line repeats 44 times.)
00000000 00000001	000001	00001	0001	128	129	NULL SOH
00000010 00000011	001003	00515	0203	130	131	STX ETX
00110111 00101101	033455	14125	372D	132	133	EOT ENQ
00101110 00101111	027057	11823	2E2F	134	135	ACK BEL
00010110 00000101	013005	05637	1605	136	137	BS HT
00100101 00001011	022413	09483	250B	138	139	LF VT
00001100 00001101	006015	03085	0C0D	140	141	FF CR
00001110 00001111	007017	03599	0E0F	142	143	SO SI
00010000 00010001	010021	04113	1011	144	145	DLE DC1
00010010 01101111	011157	04719	126F	146	147	DC2 ?
00111100 00111101	036075	15421	3C3D	148	149	DC4 NAK
00110010 00100110	031046	12838	3226	150	151	SYN ETB
00011000 00011001	014031	06169	1819	152	153	CAN EM
00111111 00100111	037447	16167	3F27	154	155	SUB ESC
00011100 00011101	016035	07197	1C1D	156	157	IFS IGS
00011110 00011111	017037	07711	1E1F	158	159	IRS IUS
01000000 01011010	040132	16474	405A	160	161	SP !
01111111 01111011	077573	32635	7F7B	162	163	" #
01011011 01101100	055554	23404	5B6C	164	165	$ %
01010000 01111101	050175	20605	507D	166	167	& '
01001101 01011101	046535	19805	4D5D	168	169	( )
01011100 01001110	056116	23630	5C4E	170	171	* +
01101011 01100000	065540	27488	6B60	172	173	, -
01001011 01100001	045541	19297	4B61	174	175	. /
11110000 11110001	170361	61681	F0F1	176	177	0 1
11110010 11110011	171363	62195	F2F3	178	179	2 3
11110100 11110101	172365	62709	F4F5	180	181	4 5
11110110 11110111	173367	63223	F6F7	182	183	6 7
11111000 11111001	174371	63737	F8F9	184	185	8 9
01111010 01011110	075136	31326	7A5E	186	187	: ;
01001100 01111110	046176	19582	4C7E	188	189	< =
01101110 01101111	067157	28271	6E6F	190	191	> ?
01111100 11000001	076301	31937	7CC1	192	193	@ A
11000010 11000011	141303	49859	C2C3	194	195	B C

Prime ECS to EBCDIC (continued)

Binary		Octal	Decimal	Hex.	Positions		Characters	
11000100	11000101	142305	50373	C4C5	196	197	D	E
11000110	11000111	143307	50887	C6C7	198	199	F	G
11001000	11001001	144311	51401	C8C9	200	201	H	I
11010001	11010010	150722	53714	D1D2	202	203	J	K
11010011	11010100	151724	54228	D3D4	204	205	L	M
11010101	11010110	152726	54742	D5D6	206	207	N	O
11010111	11011000	153730	55256	D7D8	208	209	P	Q
11011001	11100010	154742	55778	D9E2	210	211	R	S
11100011	11100100	161744	58340	E3E4	212	213	T	U
11100101	11100110	162746	58854	E5E6	214	215	V	W
11100111	11101000	163750	59368	E7E8	216	217	X	Y
11101001	11000000	164700	59840	E9C0	218	219	Z	[
11100000	11010000	160320	57552	E0D0	220	221	\	]
01011111	01101101	057555	24429	5F6D	222	223	^	_
01111001	10000001	074601	31105	7981	224	225	`	a
10000010	10000011	101203	33411	8283	226	227	b	c
10000100	10000101	102205	33925	8485	228	229	d	e
10000110	10000111	103207	34439	8687	230	231	f	g
10001000	10001001	104211	34953	8889	232	233	h	i
10010001	10010010	110622	37266	9192	234	235	j	k
10010011	10010100	111624	37780	9394	236	237	l	m
10010101	10010110	112626	38294	9596	238	239	n	o
10010111	10011000	113630	38808	9798	240	241	p	q
10011001	10100010	114642	39330	99A2	242	243	r	s
10100011	10100100	121644	41892	A3A4	244	245	t	u
10100101	10100110	122646	42406	A5A6	246	247	v	w
10100111	10101000	123650	42920	A7A8	248	249	x	y
10101001	11000000	124700	43456	A9C0	250	251	z	{
01101010	11010000	065320	27344	6AD0	252	253	|	}
10100001	00000111	120407	41223	A107	254	255	~	DEL

EBCDIC to Prime ECS

Binary		Octal	Dec.	Hex.	Positions		Characters	
10000000	10000001	100201	32897	8081	000	001	NULL	SOH
10000010	10000011	101203	33411	8283	002	003	STX	ETX
10111111	10001001	137611	49033	BF89	004	005	?	HT
10111111	11111111	137777	49151	BFFF	006	007	?	DEL
10111111	10111111	137677	49087	BFBF	008	009	?	?
10111111	10001011	137613	49035	BF8B	010	011	?	VT
10001100	10001101	106215	35981	8C8D	012	013	FF	CR
10001110	10001111	107217	36495	8E8F	014	015	SO	SI
10010000	10010001	110221	37009	9091	016	017	DLE	DC1
10010010	10111111	111277	37567	92BF	018	019	DC2	?
10111111	10001010	137612	49034	BF8A	020	021	?	LF
10001000	10111111	104277	35007	88BF	022	023	BS	?
10011000	10011001	114231	39065	9899	024	025	CAN	EM
10111111	10111111	137677	49087	BFBF	026	027	?	?
10011100	10011101	116235	40093	9C9D	028	029	FS	GS
10011110	10011111	117237	40607	9E9F	030	031	RS	US
10111111	10111111	137677	49087	BFBF	032	033	?	?
10111111	10111111	137677	49087	BFBF	034	035	?	?
10111111	10001010	137612	49034	BF8A	036	037	?	LF
10010111	10011011	113633	38811	979B	038	039	ETB	ESC
10111111	10111111	137677	49087	BFBF	040	041	?	?
10111111	10111111	137677	49087	BFBF	042	043	?	?
10111111	10000101	137605	49029	BF85	044	045	?	ENQ
10000110	10000111	103207	34439	8687	046	047	ACK	BEL
10111111	10111111	137677	49087	BFBF	048	049	?	?
10010110	10111111	113277	38591	96BF	050	051	SYN	?
10111111	10111111	137677	49087	BFBF	052	053	?	?
10111111	10000100	137604	49028	BF84	054	055	?	EOT
10111111	10111111	137677	49087	BFBF	056	057	?	?
10111111	10111111	137677	49087	BFBF	058	059	?	?
10010100	10010101	112225	38037	9495	060	061	DC4	NAK
10111111	10011010	137632	49050	BF9A	062	063	?	SUB
10100000	10111111	120277	41151	A0BF	064	065	SP	?
10111111	10111111	137677	49087	BFBF	066	067	?	?
10111111	10111111	137677	49087	BFBF	068	069	?	?
10111111	10111111	137677	49087	BFBF	070	071	?	?
10111111	10111111	137677	49087	BFBF	072	073	?	?
00100010	10101110	021256	8878	22AE	074	075	cent	.
10111100	10101000	136250	48296	BCA8	076	077	<	(
10101011	11111100	125774	44028	ABFC	078	079	+	|
10100110	10111111	123277	42687	A6BF	080	081		?
10111111	10111111	137677	49087	BFBF	082	083	?	?
10111111	10111111	137677	49087	BFBF	084	085	?	?
10111111	10111111	137677	49087	BFBF	086	087	?	?
10111111	10111111	137677	49087	BFBF	088	089	?	?
10100001	10100100	120644	41380	A1A4	090	091	!	$
10101010	10101001	125251	43689	AAA9	092	093	*	)
10111011	11011110	135736	48094	BBDE	094	095	;	^
10101101	10101111	126657	44463	ADAF	096	097	-	/

EBCDIC to Prime ECS (continued)

Binary		Octal	Dec.	Hex.	Positions		Characters	
10111111	10111111	137677	49087	BFBF	098	099	?	?
10111111	10111111	137677	49087	BFBF	100	101	?	?
10111111	10111111	137677	49087	BFBF	102	103	?	?
10111111	10111111	137677	49087	BFBF	104	105	?	?
11111100	10101100	176254	64684	FCAC	106	107	¦	,
10100101	11011111	122737	42463	A5DF	108	109	%	_
10111110	10111111	137277	48831	BEBF	110	111	>	?
10111111	10111111	137677	49087	BFBF	112	113	?	?
10111111	10111111	137677	49087	BFBF	114	115	?	?
10111111	10111111	137677	49087	BFBF	116	117	?	?
10111111	10111111	137677	49087	BFBF	118	119	?	?
10111111	11100000	137740	49120	BFE0	120	121	?	
10111010	10100011	135243	47779	BAA3	122	123	:	#
11000000	10100111	140247	49319	C0A7	124	125	@	'
10111101	10100010	136642	48546	BDA2	126	127	=	"
10111111	11100001	137741	49121	BFE1	128	129	?	a
11100010	11100011	161343	58083	E2E3	130	131	b	c
11100100	11100101	162345	58597	E4E5	132	133	d	e
11100110	11100111	163347	59111	E6E7	134	135	f	g
11101000	11101001	164351	59625	E8E9	136	137	h	i
10111111	10111111	137677	49087	BFBF	138	139	?	?
10111111	10111111	137677	49087	BFBF	140	141	?	?
10111111	10111111	137677	49087	BFBF	142	143	?	?
10111111	11101010	137752	49130	BFEA	144	145	?	j
11101011	11101100	165754	60396	EBEC	146	147	k	l
11101101	11101110	166756	60910	EDEE	148	149	m	n
11101111	11110000	167760	61424	EFF0	150	151	o	p
11110001	11110010	170762	61938	F1F2	152	153	q	r
10111111	10111111	137677	49087	BFBF	154	155	?	?
10111111	10111111	137677	49087	BFBF	156	157	?	?
10111111	10111111	137677	49087	BFBF	158	159	?	?
10111111	11111110	137776	49150	BFFE	160	161	?	~
11110011	11110100	171764	62452	F3F4	162	163	s	t
11110101	11110110	172766	62966	F5F6	164	165	u	v
11110111	11111000	173770	63480	F7F8	166	167	w	x
11111001	11111010	174772	63994	F9FA	168	169	y	z
10111111	10111111	137677	49087	BFBF	170	171	?	?
10111111	10111111	137677	49087	BFBF	172	173	?	?
10111111	10111111	137677	49087	BFBF	174	175	?	?
10111111	10111111	137677	49087	BFBF	176	177	?	?
10111111	10111111	137677	49087	BFBF	178	179	?	?
10111111	10111111	137677	49087	BFBF	180	181	?	?
10111111	10111111	137677	49087	BFBF	182	183	?	?
10111111	10111111	137677	49087	BFBF	184	185	?	?
10111111	10111111	137677	49087	BFBF	186	187	?	?
10111111	10111111	137677	49087	BFBF	188	189	?	?
10111111	10111111	137677	49087	BFBF	190	191	?	?
11111011	11000001	175701	64449	FBC1	192	193	{	A
11000010	11000011	141303	49859	C2C3	194	195	B	C

EBCDIC to Prime ECS (continued)

Binary		Octal	Dec.	Hex.	Positions		Characters	
11000100	11000101	142305	50373	C4C5	196	197	D	E
11000110	11000111	143307	50887	C6C7	198	199	F	G
11001000	11001001	144311	51401	C8C9	200	201	H	I
10111111	10111111	137677	49087	BFBF	202	203	?	?
10111111	10111111	137677	49087	BFBF	204	205	?	?
00100101	10111111	022677	9663	25BF	206	207	yen	?
11111101	11001010	176712	64970	FDCA	208	209	}	J
11001011	11001100	145714	52172	CBCC	210	211	K	L
11001101	11001110	146716	52686	CDCE	212	213	M	N
11001111	11010000	147720	53200	CFD0	214	215	O	P
11010001	11010010	150722	53714	D1D2	216	217	Q	R
10111111	10111111	137677	49087	BFBF	218	219	?	?
10111111	10111111	137677	49087	BFBF	220	221	?	?
10111111	10111111	137677	49087	BFBF	222	223	?	?
11011100	10111111	156277	56511	DCBF	224	225	\	?
11010011	11010100	151724	54228	D3D4	226	227	S	T
11010101	11010110	152726	54742	D5D6	228	229	U	V
11010111	11011000	153730	55256	D7D8	230	231	W	X
11011001	11011010	154732	55770	D9DA	232	233	Y	Z
10111111	10111111	137677	49087	BFBF	234	235	?	?
10111111	10111111	137677	49087	BFBF	236	237	?	?
10111111	10111111	137677	49087	BFBF	238	239	?	?
10110000	10110001	130261	45233	B0B1	240	241	0	1
10110010	10110011	131263	45747	B2B3	242	243	2	3
10110100	10110101	132265	46261	B4B5	244	245	4	5
10110110	10110111	133267	46775	B6B7	246	247	6	7
10111000	10111001	134271	47289	B8B9	248	249	8	9
11111100	10111111	176277	64703	FCBF	250	251	|	?
10111111	10111111	137677	49087	BFBF	252	253	?	?
10111111	10111111	137677	49087	BFBF	254	255	?	?

Industry-standard ASCII-8 to Prime ECS

Binary		Octal	Decimal	Hex.	Characters	
10000000	10000001	100201	32897	8081	NULL	SOH
10000010	10000011	101203	33411	8283	STX	ETX
10000100	10000101	102205	33925	8485	EOT	ENQ
10000110	10000111	103207	34439	8687	ACK	BEL
10001000	10001001	104211	34953	8889	BS	HT
10001010	10001011	105213	35467	8A8B	LF	VT
10001100	10001101	106215	35981	8C8D	FF	CR
10001110	10001111	107217	36495	8E8F	SO	SI
10010000	10010001	110221	37009	9091	DLE	DC1
10010010	10010011	111223	37523	9293	DC2	DC3
10010100	10010101	112225	38037	9495	DC4	NAK
10010110	10010111	113227	38551	9697	SYN	ETB
10011000	10011001	114231	39065	9899	CAN	EM
10011010	10011011	115233	39579	9A9B	SUB	ESC
10011100	10011101	116235	40093	9C9D	FS	GS
10011110	10011111	117237	40607	9E9F	RS	US
10100000	10100001	120241	41121	A0A1	sp	!
10100010	10100011	121243	41635	A2A3	"	#
10100100	10100101	122245	42149	A4A5	$	%
10100110	10100111	123247	42663	A6A7	&	'
10101000	10101001	124251	43177	A8A9	(	)
10101010	10101011	125253	43691	AAAB	*	+
10101100	10101101	126255	44205	ACAD	,	-
10101110	10101111	127257	44719	AEAF	.	/
10110000	10110001	130261	45233	B0B1	0	1
10110010	10110011	131263	45747	B2B3	2	3
10110100	10110101	132265	46261	B4B5	4	5
10110110	10110111	133267	46775	B6B7	6	7
10111000	10111001	134271	47289	B8B9	8	9
10111010	10111011	135273	47803	BABB	:	;
10111100	10111101	136275	48317	BCBD	<	=
10111110	10111111	137277	48831	BEBF	>	?
11000000	11000001	140301	49345	C0C1	@	A
11000010	11000011	141303	49859	C2C3	B	C
11000100	11000101	142305	50373	C4C5	D	E
11000110	11000111	143307	50887	C6C7	F	G
11001000	11001001	144311	51401	C8C9	H	I
11001010	11001011	145313	51915	CACB	J	K
11001100	11001101	146315	52429	CCCD	L	M
11001110	11001111	147317	52943	CECF	N	O
11010000	11010001	150321	53457	D0D1	P	Q
11010010	11010011	151323	53971	D2D3	R	S
11010100	11010101	152325	54485	D4D5	T	U
11010110	11010111	153327	54999	D6D7	V	W

Industry-standard ASCII-8 to Prime ECS (continued)

Binary	Octal	Decimal	Hex.	Characters	
11011000 11011001	154331	55513	D8D9	X	Y
11011010 11011011	155333	56027	DADB	Z	[
11011100 11011101	156335	56541	DCDD	\	]
11011110 11011111	157337	57055	DEDF	^	_
11100000 11100001	160341	57569	E0E1	`	a
11100010 11100011	161343	58083	E2E3	b	c
11100100 11100101	162345	58597	E4E5	d	e
11100110 11100111	163347	59111	E6E7	f	g
11101000 11101001	164351	59625	E8E9	h	i
11101010 11101011	165353	60139	EAEB	j	k
11101100 11101101	166355	60653	ECED	l	m
11101110 11101111	167357	61167	EEEF	n	o
11110000 11110001	170361	61681	F0F1	p	q
11110010 11110011	171363	62195	F2F3	r	s
11110100 11110101	172365	62709	F4F5	t	u
11110110 11110111	173367	63223	F6F7	v	w
11111000 11111001	174371	63737	F8F9	x	y
11111010 11111011	175373	64251	FAFB	z	{
11111100 11111101	176375	64765	FCFD	|	}
11111110 11111111	177377	65279	FEFF	~	DEL
00000000 00000001	000001	00001	0001	RES1	RES2
00000010 00000011	001003	00515	0203	RES3	RES4
00000100 00000101	002005	01029	0405	IND	NEL
00000110 00000111	003007	01543	0607	SSA	ESA
00001000 00001001	004011	02057	0809	HTS	HTJ
00001010 00001011	005013	02571	0A0B	VTS	PLD
00001100 00001101	006015	03085	0C0D	PLU	RI
00001110 00001111	007017	03599	0E0F	SS2	SS3
00010000 00010001	010021	04113	1011	DCS	PU1
00010010 00010011	011023	04627	1213	PU2	STS
00010100 00010101	012025	05141	1415	CCH	MW
00010110 00010111	013027	05655	1617	SPA	EPA
00011000 00011001	014031	06169	1819	RES5	RES6
00011010 00011011	015033	06683	1A1B	RES7	CSI
00011100 00011101	016035	07197	1C1D	ST	OSC
00011110 00011111	017037	07711	1E1F	PM	APC
00100000 00100001	020041	08225	2021	NBSP	INVE
00100010 00100011	021043	08739	2223	CENT	PND
00100100 00100101	022045	09253	2425	CURR	YEN
00100110 00100111	023047	09767	2627	BBAR	SECT
00101000 00101001	024051	10281	2829	DIA	COPY
00101010 00101011	025053	10795	2A2B	FOI	LAQM
00101100 00101101	026055	11309	2C2D	NOT	SHY
00101110 00101111	027057	11823	2E2F	TM	MACN
00110000 00110001	030061	12337	3031	DEGR	PLMI
00110010 00110011	031063	12851	3233	SPS2	SPS3
00110100 00110101	032065	13365	3435	AAC	LCMU
00110110 00110111	033067	13879	3637	PARA	MIDD
00111000 00111001	034071	14393	3839	CED	SPS1

Industry-standard ASCII-8 to Prime ECS (continued)

Binary	Octal	Decimal	Hex.	Characters
00111010 00111011	035073	14907	3A3B	MOI RAQM
00111100 00111101	036075	15421	3C3D	FR14 FR12
00111110 00111111	037077	15935	3E3F	FR34 INVQ
01000000 01000001	040101	16449	4041	UCAG UCAA
01000010 01000011	041103	16963	4243	UCAC UCAT
01000100 01000101	042105	17477	4445	UCAD UCAR
01000110 01000111	043107	17991	4647	UCAE UCCC
01001000 01001001	044111	18505	4849	UCEG UCEA
01001010 01001011	045113	19019	4A4B	UCEC UCED
01001100 01001101	046115	19533	4C4D	UCIG UCIA
01001110 01001111	047117	20047	4E4F	UCIC UCID
01010000 01010001	050121	20561	5051	UETH UCNT
01010010 01010011	051123	21075	5253	UCOG UCOA
01010100 01010101	052125	21589	5455	UCOC UCOT
01010110 01010111	053127	22103	5657	UCOD MULT
01011000 01011001	054131	22617	5859	UCOO UCUG
01011010 01011011	055133	23131	5A5B	UCUA UCUC
01011100 01011101	056135	23645	5C5D	UCUD UCYA
01011110 01011111	057137	24159	5E5F	UTHN LGSS
01100000 01100001	060141	24673	6061	LCAG LCAA
01100010 01100011	061143	25187	6263	LCAC LCAT
01100100 01100101	062145	25701	6465	LCAD LCAR
01100110 01100111	063147	26215	6667	LCAE LCCC
01101000 01101001	064151	26729	6869	LCEG LCEA
01101010 01101011	065153	27243	6A6B	LCEC LCED
01101100 01101101	066155	27757	6C6D	LCIG LCIA
01101110 01101111	067157	28271	6E6F	LCIC LCID
01110000 01110001	070161	28785	7071	LETH LCNT
01110010 01110011	071163	29299	7273	LCOG LCOA
01110100 01110101	072165	29813	7475	LCOC LCOT
01110110 01110111	073167	30327	7677	LCOD DIV
01111000 01111001	074171	30841	7879	LCOO LCUG
01111010 01111011	075173	31355	7A7B	LCUA LCUC
01111100 01111101	076175	31869	7C7D	LCUD LCYA
01111110 01111111	077177	32383	7E7F	LTHN LCYD

Prime ECS to Industry-standard ASCII-8

Binary		Octal	Decimal	Hex.	Characters	
10000000	10000001	100201	32897	8081	RES1	RES2
10000010	10000011	101203	33411	8283	RES3	RES4
10000100	10000101	102205	33925	8485	IND	NEL
10000110	10000111	103207	34439	8687	SSA	ESA
10001000	10001001	104211	34953	8889	HTS	HTJ
10001010	10001011	105213	35467	8A8B	VTS	PLD
10001100	10001101	106215	35981	8C8D	PLU	RI
10001110	10001111	107217	36495	8E8F	SS2	SS3
10010000	10010001	110221	37009	9091	DCS	PU1
10010010	10010011	111223	37523	9293	PU2	STS
10010100	10010101	112225	38037	9495	CCH	MW
10010110	10010111	113227	38551	9697	SPA	EPA
10011000	10011001	114231	39065	9899	RES5	RES6
10011010	10011011	115233	39579	9A9B	RES7	CSI
10011100	10011101	116235	40093	9C9D	ST	OSC
10011110	10011111	117237	40607	9E9F	PM	APC
10100000	10100001	120241	41121	A0A1	NBSP	INVE
10100010	10100011	121243	41635	A2A3	CENT	PND
10100100	10100101	122245	42149	A4A5	CURR	YEN
10100110	10100111	123247	42663	A6A7	BBAR	SECT
10101000	10101001	124251	43177	A8A9	DIA	COPY
10101010	10101011	125253	43691	AAAB	FOI	LAQM
10101100	10101101	126255	44205	ACAD	NOT	SHY
10101110	10101111	127257	44719	AEAF	TM	MACN
10110000	10110001	130261	45233	B0B1	DEGR	PLMI
10110010	10110011	131263	45747	B2B3	SPS2	SPS3
10110100	10110101	132265	46261	B4B5	AAC	LCMU
10110110	10110111	133267	46775	B6B7	PARA	MIDD
10111000	10111001	134271	47289	B8B9	CED	SPS1
10111010	10111011	135273	47803	BABB	MOI	RAQM
10111100	10111101	136275	48317	BCBD	FR14	FR12
10111110	10111111	137277	48831	BEBF	FR34	INVQ
11000000	11000001	140301	49345	C0C1	UCAG	UCAA
11000010	11000011	141303	49859	C2C3	UCAC	UCAT
11000100	11000101	142305	50373	C4C5	UCAD	UCAR
11000110	11000111	143307	50887	C6C7	UCAE	UCCC
11001000	11001001	144311	51401	C8C9	UCEG	UCEA
11001010	11001011	145313	51915	CACB	UCEC	UCED
11001100	11001101	146315	52429	CCCD	UCIG	UCIA
11001110	11001111	147317	52943	CECF	UCIC	UCID
11010000	11010001	150321	53457	D0D1	UETH	UCNT
11010010	11010011	151323	53971	D2D3	UCOG	UCOA
11010100	11010101	152325	54485	D4D5	UCOC	UCOT
11010110	11010111	153327	54999	D6D7	UCOD	MULT
11011000	11011001	154331	55513	D8D9	UCOO	UCUG
11011010	11011011	155333	56027	DADB	UCUA	UCUC

Prime ECS to Industry-standard ASCII-8 (continued)

Binary		Octal	Decimal	Hex.	Characters	
11011100	11011101	156335	56541	DCDD	UCUD	UCYA
11011110	11011111	157337	57055	DEDF	UTHN	LGSS
11100000	11100001	160341	57569	E0E1	LCAG	LCAA
11100010	11100011	161343	58083	E2E3	LCAC	LCAT
11100100	11100101	162345	58597	E4E5	LCAD	LCAR
11100110	11100111	163347	59111	E6E7	LCAE	LCCC
11101000	11101001	164351	59625	E8E9	LCEG	LCEA
11101010	11101011	165353	60139	EAEB	LCEC	LCED
11101100	11101101	166355	60653	ECED	LCIG	LCIA
11101110	11101111	167357	61167	EEEF	LCIC	LCID
11110000	11110001	170361	61681	F0F1	LETH	LCNT
11110010	11110011	171363	62195	F2F3	LCOG	LCOA
11110100	11110101	172365	62709	F4F5	LCOC	LCOT
11110110	11110111	173367	63223	F6F7	LCOD	DIV
11111000	11111001	174371	63737	F8F9	LCOO	LCUG
11111010	11111011	175373	64251	FAFB	LCUA	LCUC
11111100	11111101	176375	64765	FCFD	LCUD	LCYA
11111110	11111111	177377	65279	FEFF	LTHN	LCYD
00000000	00000001	000001	00001	0001	NULL	SOH
00000010	00000011	001003	00515	0203	STX	ETX
00000100	00000101	002005	01029	0405	EOT	ENQ
00000110	00000111	003007	01543	0607	ACK	BEL
00001000	00001001	004011	02057	0809	BS	HT
00001010	00001011	005013	02571	0A0B	LF	VT
00001100	00001101	006015	03085	0C0D	FF	CR
00001110	00001111	007017	03599	0E0F	SO	SI
00010000	00010001	010021	04113	1011	DLE	DC1
00010010	00010011	011023	04627	1213	DC2	DC3
00010100	00010101	012025	05141	1415	DC4	NAK
00010110	00010111	013027	05655	1617	SYN	ETB
00011000	00011001	014031	06169	1819	CAN	EM
00011010	00011011	015033	06683	1A1B	SUB	ESC
00011100	00011101	016035	07197	1C1D	FS	GS
00011110	00011111	017037	07711	1E1F	RS	US
00100000	00100001	020041	08225	2021	sp	!
00100010	00100011	021043	08739	2223	"	#
00100100	00100101	022045	09253	2425	$	%
00100110	00100111	023047	09767	2627	&	'
00101000	00101001	024051	10281	2829	(	)
00101010	00101011	025053	10795	2A2B	*	+
00101100	00101101	026055	11309	2C2D	,	-
00101110	00101111	027057	11823	2E2F	.	/
00110000	00110001	030061	12337	3031	0	1
00110010	00110011	031063	12851	3233	2	3
00110100	00110101	032065	13365	3435	4	5
00110110	00110111	033067	13879	3637	6	7

Prime ECS to Industry-standard ASCII-8 (continued)

Binary	Octal	Decimal	Hex.	Characters
00111000 00111001	034071	14393	3839	8 9
00111010 00111011	035073	14907	3A3B	: ;
00111100 00111101	036075	15421	3C3D	< =
00111110 00111111	037077	15935	3E3F	> ?
01000000 01000001	040101	16449	4041	@ A
01000010 01000011	041103	16963	4243	B C
01000100 01000101	042105	17477	4445	D E
01000110 01000111	043107	17991	4647	F G
01001000 01001001	044111	18505	4849	H I
01001010 01001011	045113	19019	4A4B	J K
01001100 01001101	046115	19533	4C4D	L M
01001110 01001111	047117	20047	4E4F	N O
01010000 01010001	050121	20561	5051	P Q
01010010 01010011	051123	21075	5253	R S
01010100 01010101	052125	21589	5455	T U
01010110 01010111	053127	22103	5657	V W
01011000 01011001	054131	22617	5859	X Y
01011010 01011011	055133	23131	5A5B	Z [
01011100 01011101	056135	23645	5C5D	\ ]
01011110 01011111	057137	24159	5E5F	^ _
01100000 01100001	060141	24673	6061	` a
01100010 01100011	061143	25187	6263	b c
01100100 01100101	062145	25701	6465	d e
01100110 01100111	063147	26215	6667	f g
01101000 01101001	064151	26729	6869	h i
01101010 01101011	065153	27243	6A6B	j k
01101100 01101101	066155	27757	6C6D	l m
01101110 01101111	067157	28271	6E6F	n o
01110000 01110001	070161	28785	7071	p q
01110010 01110011	071163	29299	7273	r s
01110100 01110101	072165	29813	7475	t u
01110110 01110111	073167	30327	7677	v w
01111000 01111001	074171	30841	7879	x y
01111010 01111011	075173	31355	7A7B	z {
01111100 01111101	076175	31869	7C7D	| }
01111110 01111111	077177	32383	7E7F	~ DEL

Prime ECS to Industry-standard ASCII-7

Binary	Octal	Decimal	Hex.	Characters	
00111111 00111111	37477	16191	3F3F	?	? (No extended characters. This line repeats 63 times.)
00000000 00000001	000001	00001	0001	NULL	SOH
00000010 00000011	001003	00515	0203	STX	ETX
00000100 00000101	002005	01029	0405	EOT	ENQ
00000110 00000111	003007	01543	0607	ACK	BEL
00001000 00001001	004011	02057	0809	BS	HT
00001010 00001011	005013	02571	0A0B	LF	VT
00001100 00001101	006015	03085	0C0D	FF	CR
00001110 00001111	007017	03599	0E0F	SO	SI
00010000 00010001	010021	04113	1011	DLE	DC1
00010010 00010011	011023	04627	1213	DC2	DC3
00010100 00010101	012025	05141	1415	DC4	NAK
00010110 00010111	013027	05655	1617	SYN	ETB
00011000 00011001	014031	06169	1819	CAN	EM
00011010 00011011	015033	06683	1A1B	SUB	ESC
00011100 00011101	016035	07197	1C1D	FS	GS
00011110 00011111	017037	07711	1E1F	RS	US
00100000 00100001	020041	08225	2021	sp	!
00100010 00100011	021043	08739	2223	"	#
00100100 00100101	022045	09253	2425	$	%
00100110 00100111	023047	09767	2627	&	'
00101000 00101001	024051	10281	2829	(	)
00101010 00101011	025053	10795	2A2B	*	+
00101100 00101101	026055	11309	2C2D	,	-
00101110 00101111	027057	11823	2E2F	.	/
00110000 00110001	030061	12337	3031	0	1
00110010 00110011	031063	12851	3233	2	3
00110100 00110101	032065	13365	3435	4	5
00110110 00110111	033067	13879	3637	6	7
00111000 00111001	034071	14393	3839	8	9
00111010 00111011	035073	14907	3A3B	:	;
00111100 00111101	036075	15421	3C3D	<	=
00111110 00111111	037077	15935	3E3F	>	?
01000000 01000001	040101	16449	4041	@	A
01000010 01000011	041103	16963	4243	B	C
01000100 01000101	042105	17477	4445	D	E
01000110 01000111	043107	17991	4647	F	G
01001000 01001001	044111	18505	4849	H	I
01001010 01001011	045113	19019	4A4B	J	K
01001100 01001101	046115	19533	4C4D	L	M
01001110 01001111	047117	20047	4E4F	N	O
01010000 01010001	050121	20561	5051	P	Q
01010010 01010011	051123	21075	5253	R	S
01010100 01010101	052125	21589	5455	T	U

Prime ECS to Industry-standard ASCII-7 (continued)

Binary	Octal	Decimal	Hex.	Characters	
01010110 01010111	053127	22103	5657	V	W
01011000 01011001	054131	22617	5859	X	Y
01011010 01011011	055133	23131	5A5B	Z	[
01011100 01011101	056135	23645	5C5D	\	]
01011110 01011111	057137	24159	5E5F	^	_
01100000 01100001	060141	24673	6061	`	a
01100010 01100011	061143	25187	6263	b	c
01100100 01100101	062145	25701	6465	d	e
01100110 01100111	063147	26215	6667	f	g
01101000 01101001	064151	26729	6869	h	i
01101010 01101011	065153	27243	6A6B	j	k
01101100 01101101	066155	27757	6C6D	l	m
01101110 01101111	067157	28271	6E6F	n	o
01110000 01110001	070161	28785	7071	p	q
01110010 01110011	071163	29299	7273	r	s
01110100 01110101	072165	29813	7475	t	u
01110110 01110111	073167	30327	7677	v	w
01111000 01111001	074171	30841	7879	x	y
01111010 01111011	075173	31355	7A7B	z	{
01111100 01111101	076175	31869	7C7D	|	}
01111110 01111111	077177	32383	7E7F	~	DEL

Industry-standard ASCII-7 to Prime ECS

Binary	Octal	Decimal	Hex.	Characters	
10000000 10000001	100201	32897	8081	NULL	SOH
10000010 10000011	101203	33411	8283	STX	ETX
10000100 10000101	102205	33925	8485	EOT	ENQ
10000110 10000111	103207	34439	8687	ACK	BEL
10001000 10001001	104211	34953	8889	BS	HT
10001010 10001011	105213	35467	8A8B	LF	VT
10001100 10001101	106215	35981	8C8D	FF	CR
10001110 10001111	107217	36495	8E8F	SO	SI
10010000 10010001	110221	37009	9091	DLE	DC1
10010010 10010011	111223	37523	9293	DC2	DC3
10010100 10010101	112225	38037	9495	DC4	NAK
10010110 10010111	113227	38551	9697	SYN	ETB
10011000 10011001	114231	39065	9899	CAN	EM
10011010 10011011	115233	39579	9A9B	SUB	ESC
10011100 10011101	116235	40093	9C9D	FS	GS
10011110 10011111	117237	40607	9E9F	RS	US
10100000 10100001	120241	41121	A0A1	sp	!
10100010 10100011	121243	41635	A2A3	"	#
10100100 10100101	122245	42149	A4A5	$	%
10100110 10100111	123247	42663	A6A7	&	'
10101000 10101001	124251	43177	A8A9	(	)
10101010 10101011	125253	43691	AAAB	*	+
10101100 10101101	126255	44205	ACAD	,	-
10101110 10101111	127257	44719	AEAF	.	/
10110000 10110001	130261	45233	B0B1	0	1
10110010 10110011	131263	45747	B2B3	2	3
10110100 10110101	132265	46261	B4B5	4	5
10110110 10110111	133267	46775	B6B7	6	7
10111000 10111001	134271	47289	B8B9	8	9
10111010 10111011	135273	47803	BABB	:	;
10111100 10111101	136275	48317	BCBD	<	=
10111110 10111111	137277	48831	BEBF	>	?
11000000 11000001	140301	49345	C0C1	@	A
11000010 11000011	141303	49859	C2C3	B	C
11000100 11000101	142305	50373	C4C5	D	E
11000110 11000111	143307	50887	C6C7	F	G
11001000 11001001	144311	51401	C8C9	H	I
11001010 11001011	145313	51915	CACB	J	K
11001100 11001101	146315	52429	CCCD	L	M
11001110 11001111	147317	52943	CECF	N	O
11010000 11010001	150321	53457	D0D1	P	Q
11010010 11010011	151323	53971	D2D3	R	S
11010100 11010101	152325	54485	D4D5	T	U
11010110 11010111	153327	54999	D6D7	V	W
11011000 11011001	154331	55513	D8D9	X	Y
11011010 11011011	155333	56027	DADB	Z	[
11011100 11011101	156335	56541	DCDD	\	]
11011110 11011111	157337	57055	DEDF	^	_

Industry-standard ASCII-7 to Prime ECS (continued)

Binary	Octal	Decimal	Hex.	Characters
11100000 11100001	160341	57569	E0E1	` a
11100010 11100011	161343	58083	E2E3	b c
11100100 11100101	162345	58597	E4E5	d e
11100110 11100111	163347	59111	E6E7	f g
11101000 11101001	164351	59625	E8E9	h i
11101010 11101011	165353	60139	EAEB	j k
11101100 11101101	166355	60653	ECED	l m
11101110 11101111	167357	61167	EEEF	n o
11110000 11110001	170361	61681	F0F1	p q
11110010 11110011	171363	62195	F2F3	r s
11110100 11110101	172365	62709	F4F5	t u
11110110 11110111	173367	63223	F6F7	v w
11111000 11111001	174371	63737	F8F9	x y
11111010 11111011	175373	64251	FAFB	z {
11111100 11111101	176375	64765	FCFD	| }
11111110 11111111	177377	65279	FEFF	~ DEL
10111111 10111111	137677	49087	BFBF	? ?
				(No extended characters. This line repeats 63 times.)

Prime ECS to Prime-7 / Prime-7 to Prime ECS

Binary		Octal	Decimal	Hex.	Characters	
10111111	10111111	137677	49087	BFBF	?	? (No extended characters. This line repeats 63 times.)
10000000	10000001	100201	32897	8081	NULL	SOH
10000010	10000011	101203	33411	8283	STX	ETX
10000100	10000101	102205	33925	8485	EOT	ENQ
10000110	10000111	103207	34439	8687	ACK	BEL
10001000	10001001	104211	34953	8889	BS	HT
10001010	10001011	105213	35467	8A8B	LF	VT
10001100	10001101	106215	35981	8C8D	FF	CR
10001110	10001111	107217	36495	8E8F	SO	SI
10010000	10010001	110221	37009	9091	DLE	DC1
10010010	10010011	111223	37523	9293	DC2	DC3
10010100	10010101	112225	38037	9495	DC4	NAK
10010110	10010111	113227	38551	9697	SYN	ETB
10011000	10011001	114231	39065	9899	CAN	EM
10011010	10011011	115233	39579	9A9B	SUB	ESC
10011100	10011101	116235	40093	9C9D	FS	GS
10011110	10011111	117237	40607	9E9F	RS	US
10100000	10100001	120241	41121	A0A1	sp	!
10100010	10100011	121243	41635	A2A3	"	#
10100100	10100101	122245	42149	A4A5	$	%
10100110	10100111	123247	42663	A6A7	&	'
10101000	10101001	124251	43177	A8A9	(	)
10101010	10101011	125253	43691	AAAB	*	+
10101100	10101101	126255	44205	ACAD	,	-
10101110	10101111	127257	44719	AEAF	.	/
10110000	10110001	130261	45233	B0B1	0	1
10110010	10110011	131263	45747	B2B3	2	3
10110100	10110101	132265	46261	B4B5	4	5
10110110	10110111	133267	46775	B6B7	6	7
10111000	10111001	134271	47289	B8B9	8	9
10111010	10111011	135273	47803	BABB	:	;
10111100	10111101	136275	48317	BCBD	<	=
10111110	10111111	137277	48831	BEBF	>	?
11000000	11000001	140301	49345	C0C1	@	A
11000010	11000011	141303	49859	C2C3	B	C
11000100	11000101	142305	50373	C4C5	D	E
11000110	11000111	143307	50887	C6C7	F	G
11001000	11001001	144311	51401	C8C9	H	I
11001010	11001011	145313	51915	CACB	J	K
11001100	11001101	146315	52429	CCCD	L	M
11001110	11001111	147317	52943	CECF	N	O
11010000	11010001	150321	53457	D0D1	P	Q
11010010	11010011	151323	53971	D2D3	R	S

Prime ECS to Prime-7/Prime-7 to Prime ECS (continued)

Binary		Octal	Decimal	Hex.	Characters	
11010100	11010101	152325	54485	D4D5	T	U
11010110	11010111	153327	54999	D6D7	V	W
11011000	11011001	154331	55513	D8D9	X	Y
11011010	11011011	155333	56027	DADB	Z	[
11011100	11011101	156335	56541	DCDD	\	]
11011110	11011111	157337	57055	DEDF	^	_
11100000	11100001	160341	57569	E0E1	`	a
11100010	11100011	161343	58083	E2E3	b	c
11100100	11100101	162345	58597	E4E5	d	e
11100110	11100111	163347	59111	E6E7	f	g
11101000	11101001	164351	59625	E8E9	h	i
11101010	11101011	165353	60139	EAEB	j	k
11101100	11101101	166355	60653	ECED	l	m
11101110	11101111	167357	61167	EEEF	n	o
11110000	11110001	170361	61681	F0F1	p	q
11110010	11110011	171363	62195	F2F3	r	s
11110100	11110101	172365	62709	F4F5	t	u
11110110	11110111	173367	63223	F6F7	v	w
11111000	11111001	174371	63737	F8F9	x	y
11111010	11111011	175373	64251	FAFB	z	{
11111100	11111101	176375	64765	FCFD	|	}
11111110	11111111	177377	65279	FEFF	~	DEL

D
MAGNET Messages

This appendix lists and describes all messages that can be output by the MAGNET subsystem, and, where applicable, describes the remedial action.

WARNING MESSAGES

The following warning messages are presented in alphabetical order.

- ***WARNING***
  ARCHIVE tapes cannot be overwritten. If you wish to write to this tape invoke MAGNET with the -OVERWRITE option.
  MAGNET will abort.

  Description: You have tried to overwrite an ARCHIVE tape.

  Suggested User Action: To overwrite an ARCHIVE tape, use the -OVERWRITE option when you invoke MAGNET. Alternatively, you can use the ARCHIVE_RELEASE command (described in the Data Backup and Recovery Guide) to release the tape. This labels the tape as free. You can write to free tapes without using the -OVERWRITE option.

- ***WARNING***
  BACKUP tapes cannot be overwritten. If you wish to write to this tape invoke MAGNET with the -OVERWRITE option.
  MAGNET will abort.

  Description: You have tried to overwrite a BACKUP tape.

  Suggested User Action: To overwrite a BACKUP tape, use the -OVERWRITE option when you invoke MAGNET. Alternatively, you can use the BACKUP_RELEASE command (described in the Data Backup and Recovery Guide) to release the tape. This labels the tape as free. You can write to free tapes without using the -OVERWRITE option.

- ***WARNING***
  BRMS tapes cannot be read.
  MAGNET will abort.

  Description: You cannot read BRMS-labeled tapes.

  Suggested User Action: You must use the ARCHIVE_RESTORE, BACKUP_RESTORE, TRANSPORT_RESTORE, or LIST_TAPE command to read BRMS-labeled tapes. (See the Data Backup and Recovery Guide.)

- ***WARNING***
  Discrepancy in logical record x
  LRECL specified: a LRECL returned: b

  Description: A read operation has found a discrepancy in the specified logical record size (a) and the actual logical record size (b) on logical record x.

  Suggested User Action: You can either choose to ignore this message, or respecify the logical record size.

- ***WARNING***
  TRANSPORT tapes cannot be overwritten. If you wish to write to this tape invoke MAGNET with the -OVERWRITE option.
  MAGNET will abort.

  Description: You have tried to overwrite a TRANSPORT tape.

  Suggested User Action: To overwrite a TRANSPORT tape, use the -OVERWRITE option when you invoke MAGNET. Alternatively, you can use the TRANSPORT_RELEASE command (described in the Data Backup and Recovery Guide) to release the tape. This labels the tape as free. You can write to free tapes without using the -OVERWRITE option.

- WARNING: Data is not recorded at specified density.

  Description: You have read data from tape successfully, although the recording density of the data does not match the density you have specified.

  Suggested User Action: You can ignore this warning, but if you want to avoid the message, set DENSITY to the correct value.

- WARNING: Not all drives of this type support 800 BPI.

  Description: Older TELEX drives do not support 800 bpi. If I/O errors occur while you are using 800 bpi, the reason may be that your tape drive does not support 800 bpi.

ERROR MESSAGES

MAGNET error messages are listed below in alphabetical order. A description is provided for each message, and the suggested user action is also provided where practicable.

- A disk file is read-protected.

  Suggested User Action: If you own the directory where the disk file(s) are located, change the protection attributes to enable Read (R) access. If you are a non-owner, ask either the owner or the System Administrator to change the protection attributes.

- A disk file is write-protected.

  Suggested User Action: If you own the directory where the disk file(s) are located, change the protection attributes to enable Write (W) access. If you are a non-owner, ask either the owner or the System Administrator to change the protection attributes.

- A positioning sub-operation has aborted.

  Description: An error occurred while a tape was being positioned to its correct file and/or record location.

  Suggested User Action: Check that the correct tape is mounted and that the correct density is set. You may try rewinding the tape before you reissue the subcommand. If the problem persists, it usually indicates that your tape is corrupt.

- A run-away tape condition exists.

  Description: Either a read forward-space file, or a read forward-space record operation is executing indefinitely; this condition eventually pulls the tape from the supply reel. This occurs only when your tape is blank (read forward-space record) or when no more filemarkers are found on the tape (read forward-space file).

  Suggested User Action: Press the terminal BREAK key immediately. Deassign the tape drive(s) and immediately reassign them. This action halts the tape drive(s) suddenly. You should then reenter MAGNET to continue your magnetic tape operations.

- A spool object cannot be an information source.

  Description: You typed a MOVE subcommand and specified the source object as a spool object. Spool objects can be used only for output.

  Suggested User Action: Retype the MOVE subcommand, and specify only a disk or a tape object as the source.

- A subcommand has aborted.

  Description: A MAGNET subcommand was unable to operate as you requested. Usually, before you receive this message, you receive one that indicates the specific type of problem.

  Suggested User Action: Reenter MAGNET and try your MAGNET operation(s) again. If the problem persists, contact your System Administrator.

- A tape dismount (unload) operation has aborted.

  Description: Either you or the operator typed REPLY -TAPE ABORT in response to a dismount request.

  Suggested User Action: None, if the REPLY -TAPE ABORT response was intentional.

- A tape drive is not ready and/or is offline.

  Suggested User Action: Make sure that the tape drive is powered-on, a tape has been mounted, and that the online button has been pressed.

- A tape is hardware write-protected.

  Suggested User Action: Put a write-enable ring on your tape(s).

- A tape is software write-protected.

  Suggested User Action: Set the value of the PROTECT option for your tape(s) to NONE.

- A tape mount operation has aborted.

  Description: Either you or the operator typed REPLY -TAPE ABORT in response to a mount request.

  Suggested User Action: None, if the REPLY -TAPE ABORT response was intentional.

- Abnormal halt. Contact your system administrator.

  Description: An unusual and irretrievable condition has occurred within MAGNET.

  Suggested User Action: Save all pertinent information, such as output from COMOUTPUT, DMSTK, and PM commands. (See the PRIMOS Commands Reference Guide.) Give all this information to your System Administrator.

- Abnormal halt. Some disk or spool files may be open.

  Description: An unusual and irretrievable condition has occurred within MAGNET.

  Suggested User Action: Save all pertinent information such as output from COMOUTPUT, DMSTK, and PM commands. (See the PRIMOS Commands Reference Guide.) Give all this information to your System Administrator. In addition, check for open disk files.

- All options following EXTERNAL are ignored.

  Suggested User Action: Because all options that follow the EXTERNAL option in your DECLARE or MODIFY subcommand have been ignored, use the DISPLAY subcommand to check that all option values are correct. If necessary, use the MODIFY subcommand to change option values.

- An I/O error has occurred in a destination object.

  Description: This message may indicate one of the following problems:

  - A tape drive is not assigned.
  - A tape drive is not connected.
  - An invalid command was sent to the tape drive.
  - A destination object in the READ, WRITE, or MOVE subcommand you just issued was left open due to a previous error.
  - An uncorrectable error occurred while writing a record to a destination object.

  Suggested User Action: Make sure that you have assigned all tape drives you plan to use. Make sure that any tape drives you have assigned are valid. If your system contains only one tape drive and it is connected to the first controller, that drive can be identified only by 0, 1, 2, or 3. Make sure that the density switches on all drives are set correctly. Make sure that all disk files are closed before you invoke MAGNET. If the error persists, it may indicate that a tape is dirty or creased.

- An I/O error has occurred in the source object.

  Description: This message may indicate one of the following problems:

  - A tape drive is not assigned.
  - A tape drive is not connected.
  - An invalid command was sent to the tape drive.
  - The source object in the READ, WRITE, or MOVE subcommand you just issued was left open due to a previous error.
  - An uncorrectable error occurred while reading a record from the source object.

  Suggested User Action: Make sure you have assigned all tape drives you plan to use. Make sure that any tape drives you have assigned are valid. If your system contains only one tape drive and it is connected to the first controller, that drive can be identified only by 0, 1, 2, or 3. Make sure that the density switches on all drives are set correctly. Make sure that all disk files are closed before you invoke MAGNET. If the error persists, it may indicate that your tape contains corrupt data.

- An error occurred while saving into a variable.

  Description: An error occurred when the SAVE subcommand tried to save an object's options and values.

  Suggested User Action: Make sure that your global variable file is activated and not full. Try the SAVE subcommand again. If the problem persists, contact your System Administrator.

- An invalid command line option will be ignored.

  Suggested User Action: No corrective action is necessary. However, if you incorrectly typed -SILENT or -USER you should give the QUIT subcommand and reenter MAGNET with the correct MAGNET command line options.

- An invalid translation table was specified.

  Suggested User Action: Retype the LOAD subcommand, specifying V, W, X, Y, or Z as the user translation table.

- An object-name is too long or too short.

  Suggested User Action: Retype the subcommand line, making sure that any object-name(s) specified are no longer than 32 characters, and no shorter than one character.

- An object was already declared.

  Suggested User Action: You cannot declare an object more than once. Use the MODIFY subcommand to change option values.

- An object was not previously declared.

  Suggested User Action: Declare the object(s).

- Circular chaining is not permitted.

  Description: The values you specified for the PREVCHAIN and/or NEXTCHAIN options are the same as the object-name you are currently specifying in a DECLARE or MODIFY subcommand.

  Suggested User Action: Execution continues, but the NEXTCHAIN and/or PREVCHAIN options are not set, you must modify them.

- Data conversion error.

  Description: Bad character(s) were detected while reading from a disk file into a user translation table.

  Suggested User Action: Check your disk file for valid characters. Binary numbers may contain only the characters 0 or 1. Octal numbers may contain only the characters 0 through 7. Decimal numbers may contain only the characters 0 through 9. Hexadecimal numbers may contain only the characters 0 through 9 and A through F. Check for correct spacing of these numbers. Blanks can be considered part of a numeric field and will cause this error to occur.

- Density may not be set on this tape drive.

  Description: The tape drive does not support the checking or setting of density by software.

  Suggested User Action: Use DEN=(DEFAULT) with this tape drive.

- Density may only be changed at load point.

  Description: The tape drive is not set to the specified density, but the tape is not at load point and therefore the density cannot be changed.

  Suggested User Action: Rewind the tape and rewrite at the correct density, or ignore density by setting DEN=(DEFAULT).

- Disk options are invalid for non-disk objects.

  Suggested User Action: Retype the subcommand line with valid tape or spool options.

- Error closing a tape file. Hardware status is 'nnnnn'

  Description: You can receive this message for one of several reasons. The tape drive might have gone offline or become not ready; the end-of-tape (EOT) marker might have been detected; there might be an error in the tape drive, controller or both; the tape might have a bad spot and MAGNET could not read from or write to the tape even after 10 attempts. The hardware status is displayed as 5 octal digits to denote the last status value returned from the tape driver (T$MT).

  Suggested User Action: This is usually a severe error. It might be necessary to rewind the tape and reissue your WRITE or MOVE command. If, after rewriting to the tape, you receive the same message, and you notice that the tape has not reached the

EOT marker, the tape probably contains a bad spot. If you had been reading or moving from the tape, you can rectify this problem by issuing a CLOSE command.

- Error opening a tape file. Hardware status is 'nnnnn'

  Description: MAGNET could not open a tape file for one of several reasons. Your tape drive might be offline or not ready; there might be an error in the tape drive, the controller, or both; the tape might be corrupt and MAGNET could not read from or write to the tape even after ten attempts; the end-of-tape marker might have been detected; or the file number specified by the FILENO option might be larger than the number of files on the tape. The hardware status is displayed as 5 octal digits to denote the last status value returned from the tape driver (T$MT).

  Suggested User Action: Specify a PREACTION=(REWIND) option in a MODIFY command. Reissue the READ, WRITE, or MOVE subcommand.

- Error reading a tape file. Hardware status is 'nnnnn'

  Description: This message usually indicates a bad spot on the tape. The hardware status is displayed as 5 octal digits to denote the last status value returned from the tape driver (T$MT).

  Suggested User Action: Retry the READ or MOVE operation. If the error message persists, the tape is corrupt and cannot be read.

- Error while analyzing the AMOUNT option.

  Description: The COPY subcommand found an error when it analyzed the AMOUNT option.

  Suggested User Action: Retype the subcommand line with the AMOUNT option corrected.

- Error while analyzing the AT option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the AT option.

  Suggested User Action: Retype the subcommand line with the AT option corrected.

- Error while analyzing the BFACTOR option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the BFACTOR option.

  Suggested User Action: Retype the subcommand line with the BFACTOR option corrected.

- Error while analyzing the BUFFERS option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the BUFFERS option.

  Suggested User Action: Retype the subcommand line with the BUFFERS option corrected.

- Error while analyzing the CHARACTERS option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the CHARACTERS option.

  Suggested User Action: Retype the subcommand line with the CHARACTERS option corrected.

- Error while analyzing the CONTROL option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the CONTROL option.

  Suggested User Action: Retype the subcommand line with the CONTROL option corrected.

- Error while analyzing the COPIES option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the COPIES option.

  Suggested User Action: Retype the subcommand line with the COPIES option corrected.

- Error while analyzing the DEFER option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the DEFER option.

  Suggested User Action: Retype the subcommand line with the DEFER option corrected.

- Error while analyzing the DENSITY option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the DENSITY option.

  Suggested User Action: Retype the subcommand line with the DENSITY option corrected.

- Error while analyzing the DISK option.

  Description: The DECLARE, MODIFY, or LOAD subcommand found an error when it analyzed the DISK option.

  Suggested User Action: Retype the subcommand line with the DISK option corrected.

- Error while analyzing the EXCHANGE option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the EXCHANGE option.

  Suggested User Action: Retype the subcommand line with the EXCHANGE option corrected.

- Error while analyzing the EXTERNAL option.

  Description: The DECLARE, MODIFY, or SAVE subcommand found an error when it analyzed the EXTERNAL option.

  Suggested User Action: Retype the subcommand line with the EXTERNAL option corrected.

- Error while analyzing the FILENO option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the FILENO option.

  Suggested User Action: Retype the subcommand line with the FILENO option corrected.

- Error while analyzing the FORM option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the FORM option.

  Suggested User Action: Retype the subcommand line with the FORM option corrected.

- Error while analyzing the FORMAT option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the FORMAT option.

  Suggested User Action: Retype the subcommand line with the FORMAT option corrected.

- Error while analyzing the ITEMS option.

  Description: The LIST subcommand found an error when it analyzed the ITEMS option.

  Suggested User Action: Retype the subcommand line with the ITEMS option corrected.

- Error while analyzing the LIKE option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the LIKE option.

  Suggested User Action: Retype the subcommand line with the LIKE option corrected.

- Error while analyzing the LINENOS option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the LINENOS option.

  Suggested User Action: Retype the subcommand line with the LINENOS option corrected.

- Error while analyzing the LINES option.

  Description: The LOAD subcommand found an error when it analyzed the LINES option.

  Suggested User Action: Retype the subcommand line with the LINES option corrected.

- Error while analyzing the LRECL option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the LRECL option.

  Suggested User Action: Retype the subcommand line with the LRECL option corrected.

- Error while analyzing the MAXIO option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the MAXIO option.

  Suggested User Action: Retype the subcommand line with the MAXIO option corrected.

- Error while analyzing the MODE option.

  Description: The POSITION subcommand found an error when it analyzed the MODE option.

  Suggested User Action: Retype the subcommand line with the MODE option corrected.

- Error while analyzing the NEXTCHAIN option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the NEXTCHAIN option.

  Suggested User Action: Retype the subcommand line with the NEXTCHAIN option corrected.

- Error while analyzing the OFFSET option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the OFFSET option.

  Suggested User Action: Retype the subcommand line with the OFFSET option corrected.

- Error while analyzing the PAD option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the PAD option.

  Suggested User Action: Retype the subcommand line with the PAD option corrected.

- Error while analyzing the PARITY option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the PARITY option.

  Suggested User Action: Retype the subcommand line with the PARITY option corrected.

- Error while analyzing the POSTACTION option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the POSTACTION option.

  Suggested User Action: Retype the subcommand line with the POSTACTION option corrected.

- Error while analyzing the PREACTION option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the PREACTION option.

  Suggested User Action: Retype the subcommand line with the PREACTION option corrected.

- Error while analyzing the PRINT option.

  Description: The COPY subcommand found an error when it analyzed the PRINT option.

  Suggested User Action: Retype the subcommand line with the PRINT option corrected.

- Error while analyzing the PROTECT option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the PROTECT option.

  Suggested User Action: Retype the subcommand line with the PROTECT option corrected.

- Error while analyzing the RECORDNO option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the RECORDNO option.

  Suggested User Action: Retype the subcommand line with the RECORDNO option corrected.

- Error while analyzing the SPOOL option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the SPOOL option.

  Suggested User Action: Retype the subcommand line with the SPOOL option corrected.

- Error while analyzing the TAPE option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the TAPE option.

  Suggested User Action: Retype the subcommand line with the TAPE option corrected.

- Error while analyzing the TRACKS option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the TRACKS option.

  Suggested User Action: Retype the subcommand line with the TRACKS option corrected.

- Error while analyzing the TYPE option.

  Description: The LOAD subcommand found an error when it analyzed the TYPE option.

  Suggested User Action: Retype the subcommand line with the TYPE option corrected.

- Error while analyzing the VISUAL option.

  Description: The DECLARE or MODIFY subcommand found an error when it analyzed the VISUAL option.

  Suggested User Action: Retype the subcommand line with the VISUAL option corrected.

- Error while analyzing TRANSLATE token(s).

  Description: The list of edit tokens in your TRANSLATE subcommand contains one or more errors. For example, you may have typed A$ instead of A4.

  Suggested User Action: Retype the TRANSLATE subcommand with all errors corrected.

- Error while chaining tapes.

  Description: You had specified a NEXTCHAIN option value for a tape object. However, when it became necessary to change tapes, the next object could not be found.

  Suggested User Action: Use the DECLARE subcommand to declare all objects in a chain. Reissue the READ, WRITE, or MOVE subcommand.

- Error while closing a disk file.

  Suggested User Action: Use the PRIMOS CLOSE command to close the disk file.

- Error while closing a spool file.

  Suggested User Action: This is not usually a severe error because, at this point, any information that you wanted to move to the spool subsystem has been moved. You can either attempt to use the CLOSE subcommand within MAGNET or, after typing QUIT and returning to PRIMOS, issue a PRIMOS CLOSE command.

- Error while freeing dynamic memory.

  Description: An error occurred either when the space occupied by a tape, spool, or disk object, was returned to the free storage area, or when buffers (BUFFERS option) were returned to the free storage area.

  Suggested User Action: If this occurred during a DELETE operation, issue a LIST subcommand to ensure that the object was deleted. If the object was not deleted, reissue the DELETE subcommand. If this error occurred during an operation other than DELETE, try the operation again. If the problem persists, contact your System Administrator.

- Error while opening a disk file.

  Suggested User Action:

  - If you are trying to read from a disk file, check that the file exists and that the protection and/or access control for this file is correctly set.

  - If you are trying to write to a disk file, check for proper protection and/or access control. Check that the disk is not full.

- Error while opening a spool file.

  Description: The spool queue is full, or the disk is full.

  Suggested User Action: Contact your System Administrator.

- Error while parsing the option list.

  Suggested User Action: Retype your DECLARE or MODIFY subcommand, and correct any errors.

- Error while reading from a disk file.

  Suggested User Action: Check that the protection and/or access control for this disk file is correctly set.

- Error while reading subcommand.

  Description: Indicates a possible terminal I/O error.

  Suggested User Action: Retype the subcommand line. If the problem persists, contact your System Administrator.

- Error while setting up default options.

  Suggested User Action: Reissue your DECLARE subcommand. If the problem persists, contact your System Administrator.

- Error while writing to a disk file.

  Description: This message can indicate a number of problems: for example, a full disk, a network difficulty, or insufficient access rights.

  Suggested User Action: Exit MAGNET, and check your access rights. If that is not the problem, check for a full disk, and check that all systems on the network are running.

- Error while writing to a spool file.

  Description: This message can indicate a number of problems: for example, a full disk, a network difficulty, or insufficient access rights.

  Suggested User Action: Exit MAGNET, and check your access rights. If that is not the problem, check for a full disk, and check that all systems on the network are running.

- Error writing a tape file. Hardware status is 'nnnnn'

  Description: This message usually indicates a bad spot on the tape. The hardware status is displayed as 5 octal digits to denote the last status value returned from the tape driver (T$MT).

  Suggested User Action: Retry the WRITE or MOVE operation. If the error message persists, the tape is corrupt and cannot be read.

- External names must begin with periods.

  Suggested User Action: Retype the SAVE, DECLARE, or MODIFY subcommand with a correct PRIMOS global variable name as the value for the EXTERNAL option.

- EXTERNAL option within global variable is invalid.

  Description: The PRIMOS global variable named in the EXTERNAL option contains the word EXTERNAL. Only one level of EXTERNAL is allowed.

  Suggested User Action: Correct your global variable and exclude the word EXTERNAL.

- EXTERNAL variable name not found.

  Description: The PRIMOS global variable named in the EXTERNAL option was not found.

  Suggested User Action: Make sure that you have the correct global variable file activated. Next, make sure that the particular global variable exists in the file.

- File not found.

  Description: The file that you specified does not exist.

  Suggested User Action: If you are working with a tape object, try rewinding the tape first.

- First option not DISK/TAPE/SPOOL, EXTERNAL, or LIKE.

  Suggested User Action: Retype the DECLARE subcommand line with DISK, TAPE, or EXTERNAL as the first option specified.

- Global variable file bad or uninitialized.

  Description: This message usually indicates that you do not have a global variable file activated.

  Suggested User Action: Activate your global variable file at PRIMOS level with the DEFINE_GVAR command. If the problem persists, your global variable file contains corrupt data and must be recreated.

- Global variable file is too small or full.

  Description: There is not enough room left in your global variable file to save an object with its option values.

  Suggested User Action: To make room in your global variable file, use the PRIMOS command DELETE_VAR to delete some variables.

- Illegal operation on a QIC-02 cartridge tape.

  Description: You cannot use the POSITION subcommand with cartridge tapes.

- Invalid character(s) in object-name.

  Suggested User Action: Retype the subcommand line using valid characters in the object-name(s).

- Invalid density for this tape drive.

  Description: This tape drive does not support the specified density.

  Suggested User Action: Use a valid density.

- Invalid number of files to copy.

  Suggested User Action: Reissue the COPY subcommand and specify a valid number of files to copy.

- Invalid operation for a disk or spool object.

  Suggested User Action: Reissue the COPY or POSITION subcommand, specifying tape objects only.

- Invalid operation for a tape object.

  Suggested User Action: Reissue the READ or WRITE subcommand with disk and tape objects in the correct order.

- Invalid PRIMOS command line.

  Suggested User Action: Check the command line that you have entered, then retype the correct PRIMOS command line.

- Invalid subcommand (ignored).

  Description: You have typed an invalid subcommand line.

  Suggested User Action: Retype the subcommand line. If the problem persists, contact your System Administrator.

- Invalid subcommand or characters.

  Suggested User Action: Retype the subcommand line.

- n Logical records read.

  Description: This message should always be preceded by the message "Operation complete." n indicates the number of logical records read from a tape or disk file.

- n Logical records written.

  Description: This message should always be preceded by the message "Operation complete." n is the number of logical records written to a tape or disk file.

- MAXIO is greater than the configured system limit.

  Description: MAXIO cannot be larger than the value specified by the Maximum Tape Record Size (MTRS) configuration directive.

  Suggested User Action: You can alter MAXIO, or alter MTRS and reboot the system.

- MAXIO is less than (LRECL x BFACTOR) + OFFSET

  Description: The total block length (calculated by multiplying the logical record length by the blocking factor, and adding the offset) is greater than the expected maximum physical record length.

  Suggested User Action: Increase the size of MAXIO. (Maximum size allowed = MTRS configuration directive.)

- Memory allocation error: delete some objects.

  Description: You have used all available memory for objects.

  Suggested User Action: Either delete some objects or reduce the value of the BUFFERS option for declared tape objects.

- Multiple objects specify the same tape.

  Description: In a COPY or MOVE subcommand, two or more objects specify the same tape.

  Suggested User Action: Use the MODIFY subcommand to change the value of the TAPE option. Then, reissue the MOVE or COPY subcommand.

- No more room exists to declare this object.

  Description: No more room is available in the free storage area to declare this object.

  Suggested User Action: Delete some objects or modify the values of the BUFFERS option of any previously declared tape objects.

- No objects have been declared.

  Suggested User Action: You must first declare all objects before using them.

- No room is available in the object list.

  Description: You are attempting to declare more than the maximum 100 objects.

  Suggested User Action: Delete any objects that you no longer need.

- Not enough characters in input buffer.

  Description: This message indicates that not enough characters are left to be translated.

  Suggested User Action: Check your translation string.

- Not enough objects have been specified.

  Suggested User Action: Reissue the READ, WRITE, MOVE, or COPY subcommand and specify at least two object-names.

- Object is open - cannot delete. Use CLOSE command.

  Description: The object that you are trying to delete is still open.

Suggested User Action: Use the CLOSE subcommand to close the object, then reissue the DELETE subcommand.

- Object-name specified by LIKE option not declared.

  Description: The name specified as the value for the LIKE option was not previously declared.

  Suggested User Action: Retype the command and specify as a value for the LIKE option, the name of another object that was previously declared.

- Object-name specified by LIKE option points to self.

  Description: The name specified as the value for the LIKE option is the object-name itself.

  Suggested User Action: Retype the DECLARE or MODIFY command and use a different name with the LIKE option.

- One or more objects are not of type DISK.

  Suggested User Action: Reissue the subcommand, making sure that at least one of the objects you specified is a disk object. For the READ subcommand, the second object you specify must be a disk object. For the WRITE subcommand, the first object you specify must be a disk object.

- One or more objects are not of type TAPE.

  Suggested User Action: Reissue the subcommand, making sure that at least one of the objects you specified is a tape object. For the COPY subcommand, all objects must be tape objects. For the POSITION subcommand, the one object you specify must be a tape object. For the READ subcommand, the first object must be a tape object. For the WRITE subcommand, the second object must be a tape object.

- One or more options will be ignored.

  Suggested User Action: Use the DISPLAY subcommand to check that all option values are correct. Then use the MODIFY subcommand to modify option values if necessary.

- Operation complete.

  Description: This message indicates that an I/O operation (READ, WRITE POSITION, COPY, or MOVE) has successfully completed.

- Option conflict (DENSITY TRACKS?).

  Description: A conflict exists between some of your specified options.

  Suggested User Action: Check your options for compatibility and change any incompatible option values with the MODIFY subcommand.

- Options must be specified.

  Description: You have typed a DECLARE or MODIFY subcommand without specifying any options.

  Suggested User Action: Retype the subcommand line specifying options.

- n Physical blocks read.

  Description: This message should always be preceded by the message "Operation complete". n indicates the number of physical blocks read from a tape file.

- n Physical blocks written.

  Description: This message should always be preceded by the message "Operation complete". n is the number of physical blocks written to a tape.

- QUIT. To exit from MAGNET, use the "QUIT" command.

  Description: You typed CONTROL-P (the BREAK key). This is not an error. If any operation was in progress, for example, a MOVE subcommand, it is not restartable. That is, any operation that is interrupted must be reissued.

  Suggested User Action: Type LIST ITEMS=(OPEN) to obtain a list of any object(s) left open. If you interrupted any subcommand, you must reissue the command again from the beginning; there is no restart capability.

- Same object-name appears more than once.

  Suggested User Action: Reissue the READ, WRITE, MOVE, or COPY subcommand so that object-names on the subcommand line are displayed only once.

- Specified record not found.

  Description: The POSITION subcommand was unable to position to the record number you specified.

  Suggested User Action: This may be an acceptable condition. If not, change the value of the RECORDNO option with the MODIFY subcommand and reissue the subcommand.

- Spool options are illegal for nonspool objects.

  Description: You specified a spool option while you were attempting to declare or modify a tape or disk object.

  Suggested User Action: Retype the command to specify only disk options for disk objects, and tape options for tape objects.

- Subsystem aborting- Check for open files.

  Description: A subcommand has failed. Usually, prior to receiving this message, you receive one that indicates the specific error. MAGNET cannot recover from this error, and consequently has returned you to PRIMOS command level.

  Suggested User Action: Check for open files. Attempt to correct the indicated error. Reenter MAGNET and issue your subcommand again. If the problem persists, contact your System Administrator.

- Subsystem aborting- no files are open.

  Description: A subcommand just issued has aborted. Usually, prior to receiving this error message you should have received one that indicates the specific error. MAGNET cannot recover from this error, and consequently has returned you to PRIMOS command level.

  Suggested User Action: Attempt to correct the indicated error. Reenter MAGNET and try your subcommand again. If the problem persists, contact your System Administrator.

- Tape drive not assigned.

  Suggested User Action: Assign all tape drive(s) at PRIMOS command level with the ASSIGN command. Then reenter MAGNET and reissue the command(s).

- Tape objects require at least one buffer.

  Description: You have not specified a BUFFERS option for one or more tape objects. Because the default value for this option is 0, the READ, WRITE, or MOVE subcommand could not be executed.

- Tape options are invalid for nontape objects.

  Suggested User Action: Retype the subcommand line with valid disk or spool options.

- The AMOUNT option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The BFACTOR option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The BUFFERS option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The CHARACTERS option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The CLOSE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The COPY subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The DECLARE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The DELETE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The DENSITY option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The DISK option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The DISPLAY subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The EXTERNAL option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The FORMAT option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The LINES option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The LIST subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The LOAD subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The LRECL option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The MAXIO option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The MODIFY subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The MOVE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The NEXTCHAIN option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The OFFSET option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The PARITY option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The POSITION subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The POSTACTION option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The PREACTION option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The PREVCHAIN option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The PROTECT option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The QUIT subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The READ subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The RENAME subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The SAVE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The SPOOL option value for this object is bad.

  Description: This error message may indicate an internal (severe) MAGNET error or a difficulty with the spool subsystem.

  Suggested User Action: Attempt to DELETE any objects you were using. Then try to declare these same objects again and reissue any command(s) that you typed prior to receiving this message. If you receive this message again or encounter any difficulties while using the DELETE and/or DECLARE commands, exit MAGNET and report the problem to your System Administrator.

- The TAPE option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The tape drive is not set to the specified density.

  Description: The Kennedy tape drive allows software checking of the selected density, but not software selection of density.

  Suggested User Action: Alter the density with the switch on the tape drive's front panel.

- The TRACKS option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The TRANSLATE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The TYPE option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The VISUAL option is not initialized.

  Suggested User Action: If you meant to, give this option a value. Otherwise, ignore this message.

- The WRITE subcommand will abort.

  Suggested User Action: Prior to receiving this message, you should have received a message to indicate the specific error. Correct the problem and reissue the subcommand.

- The wrong tape is mounted.

  Suggested User Action: Try the entire subcommand operation again with the correct tape mounted.

- This object has too many options to save.

  Description: The object you specified in the SAVE subcommand contains option values that exceed the maximum 1,024 characters allowed for a PRIMOS global variable.

  Suggested User Action: Set a few options back to their default values, and reissue the SAVE subcommand.

- TRANSLATION has not been specified.

  Suggested User Action: Specify a translation (use the TRANSLATE subcommand) if you meant to. Otherwise, ignore this message.

- Uncorrectable tape control error.

  Description: A tape control operation was unsuccessful.

  Suggested User Action: Check for a correct density setting on the tape drive. If the density setting is correct, your tape is either corrupt or improperly formatted. There may also be a problem with either the tape drive or controller. If the problem persists, contact your System Administrator.

- Uncorrectable tape read error.

  Description: A tape READ operation was unsuccessful after 10 attempts.

  Suggested User Action: Check for a correct density setting on the tape drive. If the density setting is correct, your tape is either corrupt or improperly formatted. There may also be a problem with either the tape drive or controller. If the problem persists, contact your System Administrator.

- Uncorrectable tape write error.

  Description: A tape WRITE operation was unsuccessful after 10 attempts.

  Suggested User Action: Either your tape is corrupt or you are attempting to write records greater than the maximum I/O window size allowed on your system. There may also be a problem with either the tape drive or controller. If the problem persists, contact your System Administrator.

- Up to 8 object-names may be specified.

  Suggested User Action: This is a warning only. You may specify no more than eight object-names on the MOVE or COPY subcommand lines. Any other object-names are ignored.

- Warning: additional errors may occur.

  Description: You issued the MODIFY subcommand with an invalid option. This could cause additional errors.

INDEX

Index

U

V

W

SURVEY

READER RESPONSE FORM

DOC10156-1LA MAGNET User's Guide

Your feedback will help us continue to improve the quality, accuracy, and organization of our publications.

1. How do you rate this document for overall usefulness?

   ☐ *excellent* ☐ *very good* ☐ *good* ☐ *fair* ☐ *poor*

2. What features of this manual did you find most useful?

   _____
   _____
   _____
   _____
   _____
   _____

3. What faults or errors in this manual gave you problems?

   _____
   _____
   _____
   _____
   _____
   _____

4. How does this manual compare to equivalent manuals produced by other computer companies?

   ☐ *Much better* ☐ *Slightly better* ☐ *About the same*
   ☐ *Much worse* ☐ *Slightly worse* ☐ *Can't judge*

5. Which other companies' manuals have you read?

   _____
   _____

Name:_____ Position:_____

Company:_____

Address:_____

_____

_____ Postal Code:_____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

Prime™

Attention: Technical Publications
Bldg 10
Prime Park, Natick, Ma. 01760