iSeries

Tape files

Version 5 Release 3

IBM

eserver

iSeries

Tape files

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 65.

Second Edition (August 2005)

This edition applies to version 5, release 3, modification 0 of IBM Operating System/400 (product number 5722–SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

Tape files 1
| What's new for V5R3 1
Print this topic. 2
Overview of tape files 2
Concepts 2
Tape data files 3
Tape device files 4
Records, blocks, and formats 4
Tape labeling 5
Use tape files 7
Initializing Tapes 7
Store data in files on tape 8
Access data from a tape device 9
Use Tape Device Files in High-Level Language Programs 15
Open Processing for Tape Device Files 16
Input/Output Processing for Tape 18
Close Processing for Tape. 18
User label processing 19
Use overrides. 20
Overriding File Attributes 21
Overriding File Names in HLL Programs . . . 22
Displaying Overrides 23
Deleting Overrides 24
File Redirection 24
Record formats 27
Example - Record format *D. 27
Example - Record format *DB 28
Example - Record format *F 29
Example - Record format *FB 30
Example - Record format *V. 31
Example - Record format *VB 31
Example - Record format *VS 33
Example - Record format *VBS 36
Example - Record format *U. 39
Reference 40
Tape file CL commands 40
Feedback Area Layouts 41
Troubleshoot tape files. 60
Handling Tape Processing Errors and Damaged Tapes 60
Related information for tape files 61
Other Tape Support Commands 61

Appendix. Notices 65
Programming Interface Information 67
Trademarks 67
Code disclaimer information. 67
Terms and conditions for downloading and printing publications 67

Tape files

This topic describes the tape media supported by the iSeries system, and the OS/400 licensed program. Characteristics and programming use of tape files are described.

See the following topics for detailed information about tape files on OS/400.

What's new for V5R3
Find out how this topic has changed since the last release.

Printable PDF
Learn how to display or print a PDF version of this topic.

Overview of Tape Files
Learn the basics of tape files.

Concepts
Learn concepts relating to tape files on OS/400®.

Use tape files
Find out how to initialize tapes, store data in files on tape, and access data from a tape device.

Use tape device files in HLL programs
Find out how to open and input/output data out of tape device files. Also, find out how to close and process user labels from tape device files.

Use overrides
Learn override rile attributes and names, and what overrides can be used for.

Examples: Record formats
Find links to examples of different record formats.

Reference
Find out the different tape file CL commands and feedback area layouts.

Troubleshoot tape files
Learn how to handle tape processing errors and damaged tapes.

Related information for tape files
Find information in the iSeries Information Center and on the Internet that offers additional topics about tape files.

What's new for V5R3

Tape files is a new topic to the Files and file systems topic in the iSeries Information Center. Technical content was taken from the Tape and Diskette Device Programming book.

To find other information about what's new or changed this release, see the Memo to Users.

Print this topic

To view or download the PDF version of this document, select Tape files (about 772 KB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF in your browser (right-click the link above).
2. Click **Save Target As...** if you are using Internet Explorer. Click **Save Link As...** if you are using Netscape Communicator.
3. Navigate to the directory in which you would like to save the PDF.
4. Click **Save**.

Downloading Adobe Acrobat Reader

You need Adobe Acrobat Reader to view or print these PDFs. You can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Overview of tape files

Device files are files that provide access to attached devices such as tapes, diskettes, printers, displays, spools, and other systems that are attached by a communications line. The device file that is described in this topic is **Tape files**, which allow access to data files on tape devices.

Each file type has its own set of unique characteristics that determines the use and capabilities the file can provide. This topic describes the characteristics and use of tape and device files for application programs. For more information about printer files, display files, and spooled files, refer to Printer Files, Display files, and the File Management topics in the iSeries Information Center.

When programs use a device file, a name represents the device file. That name identifies both the file description and, for some file types, the data itself. This manual helps you understand the following aspects of tape files so you can use their full capabilities:

- Usage characteristics
- Configuration descriptions
- Error handling methods
- Usage in high-level language programs

Concepts

Select any of the concepts from the navigation bar at left or from the following list to find out details about how tape files work on your server.

- “Tape data files” on page 3
- “Tape device files” on page 4
- “Records, blocks, and formats” on page 4
- “Tape labeling” on page 5

For information related to File management, see “Related information for tape files” on page 61. You can view or print online manuals in PDF format. You can also view other iSeries Information Center topics related to File management.

Tape data files

A tape data file contains output records produced by an application program. These files are used to store data on tape media. The tape data files are stored and accessed on tape by a tape device file.

The different data files that exist are:

- Single volume – A file that is contained on one volume of tape.
- Multivolume – A file that is contained on more than one volume of tape.
- Multifile volumes – Volumes of tape that contain more than one data file.

If you use multivolume-tape data files, then follow these conventions:

- The labels on each volume must be consistent. You cannot have standard labeled tapes and unlabeled tapes in the same tape group.
- Write all volumes and density in the same character code (EBCDIC or ASCII).
- Each tape in the group must have the same record format, block length, and record length.
- If you specify more than one tape device, then place the volumes on the devices in the sequence that is specified in the tape device file. Refer to the following reference: Figure 1. For example:
  - The data file consists of four volumes such as VOL01, VOL02, VOL03, and VOL04.
  - The tape devices specified, in order, are TAPE01, TAPE02, and TAPE03.

  Then place the volumes on a tape device as follows: VOL01 on TAPE01, VOL02 on TAPE02, VOL03 on TAPE03, and VOL04 on TAPE01.

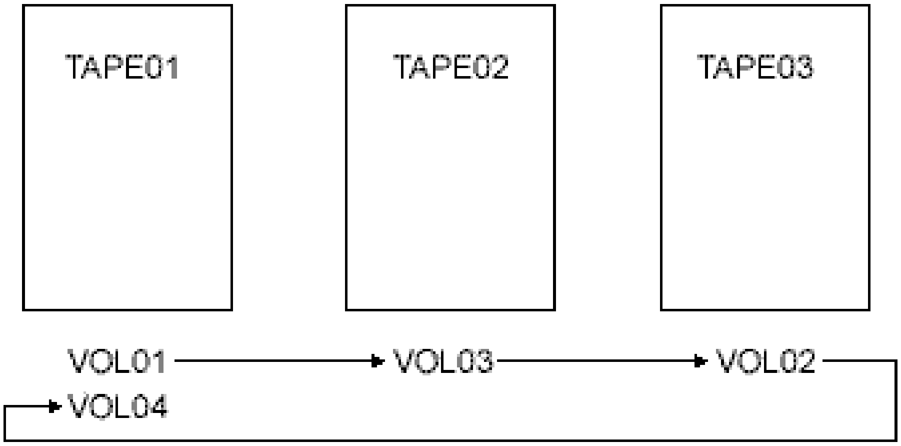


Figure 1. Multivolume-Tape-Data-File-Sequence Using Three Tape Devices

If you use volumes, in reverse order, by reading backwards, then VOL04 is on TAPE01, VOL03 on TAPE02, VOL02 on TAPE03, and VOL01 on TAPE01.

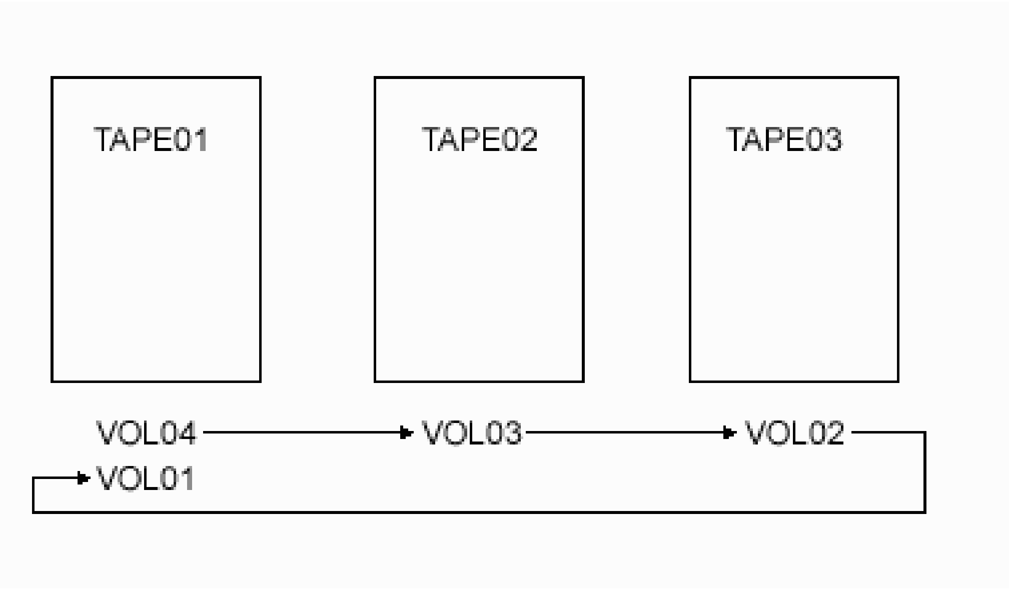


Figure 2. Sequence in Reverse Order

Tape device files

A tape device file allows access to data files on tape media. The device file contains a description of the format of the data, and a list of attributes that describe how the server should process the tape data file.

It is not necessary to have a separate device file for each tape device. Your application program can use a single device file for several different tape devices through the use of an Override Tape File (OVRTAPF) command. You can associate any number of device files with one device.

For more information about Tape device files, see:
- "IBM-Supplied Tape Device Files"
- "Create tape device file" on page 10

Note: You must vary on the configuration descriptions before using them. See the *Local Device Configuration*, SC41-5121-00 book for information about varying on configuration descriptions.

IBM-Supplied Tape Device Files

You can use the following tape device files that are shipped with the operating system:
- QTAPE (tape file)
- QTAPSRC (tape source file)

These files are all program-described as data files in library QGPL. The record format names are the same as the file names. The files contain default values for most parameters.

You can create additional tape device files to fit your needs. For example, you can create an additional tape device file to contain the specific volume and label information for a tape data file that several programs can use.

Records, blocks, and formats

This topic provides information to help you understand records, blocks, and record formats of the tapes.
- Records

Records are logical mappings of the data on a tape. It typically maps directly to the records in a database file.

- Blocks

  Blocks are physical units of data on a tape. Blocks can include a record, part of a record, or multiple records.
- Record block format

  The record block format allows the system and user to interpret the data on a tape.

To understand records, blocks, and record block formats you must know a few key terms:

- Fixed length

  The blocks on a tape have an exact (or fixed) length.
- Variable length.

  The blocks on a tape have a variable length. The block contains a header with the length of the block. Each block in a file may or may not have the same length.
- Undefined length

  The blocks on a tape have no defined length, each block can be different, and the program application interprets each block correctly.
- Blocked records

  Blocking is the process of grouping records into blocks before the system writes records on a volume. A block consists of one or more logical records. Blocking conserves storage space on a volume by reducing the number of interblock gaps in the data set. This increases processing efficiency by reducing the number of I/O operations that are required to process the data set.
- Deblocked records

  One record exists per block.
- Spanned records

  The system splits a single record (spans) into two different data blocks and writes them on the tape.
- Unspanned records

  Each record is contained within one data block.
- Interblock gap

  The physical gap on tape between two data blocks.
- Block Descriptor Word (BDW)

  One or more logical records or record segments follow a block descriptor word (BDW) in a variable length block.
- Record Descriptor Word (RDW)

  Data follows a record descriptor word (RDW) in a variable length logical record. The RDW describes the record.
- Record Segment

  Since spanned records occupy more than one block, each part of the record is a record segment.
- Segment Descriptor Word (SDW)

  Data follows a segment descriptor word (SDW) in each record segment. The SDW, similar to the RDW, describes the record segment.

In sorting out these terms, the program supports and translates certain combinations into record block formats. To view these certain combinations, select “Record formats” on page 27.

Tape labeling

The following series of diagrams provides a basic description of standard tape labeling used for the iSeries system.

In Figure 3, the INZTAP command gives the tape a volume label (marked VOL1) and writes two tape marks (TM).

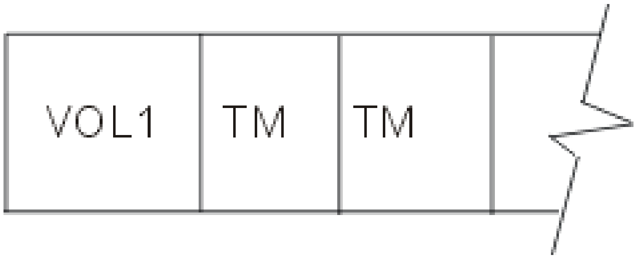


Figure 3. Volume Label and Tape Marks

When a high-level language program opens a tape file, the system does the following procedures:

1. Writes over the two tape marks that follow the VOL label with header labels HDR1 and HDR2
2. Adds a single tape mark that follows the header labels

Each header label is 80 bytes long. The first header label contains such information as the file name and date. The second header label specifies information such as record and block lengths, record block format, and buffer offset (for ASCII files).

When a high-level language program writes data to tape the system writes the data to the tape after the tape mark. Reaching the end of the file the system writes a tape mark and two end-of-file labels on the tape. The end-of-file labels contain the same information as the header labels except that the first end-of-file label (EOF1) includes the block count for the file.

Two tape marks follow the end-of-file labels as shown in the diagram below in Figure 4

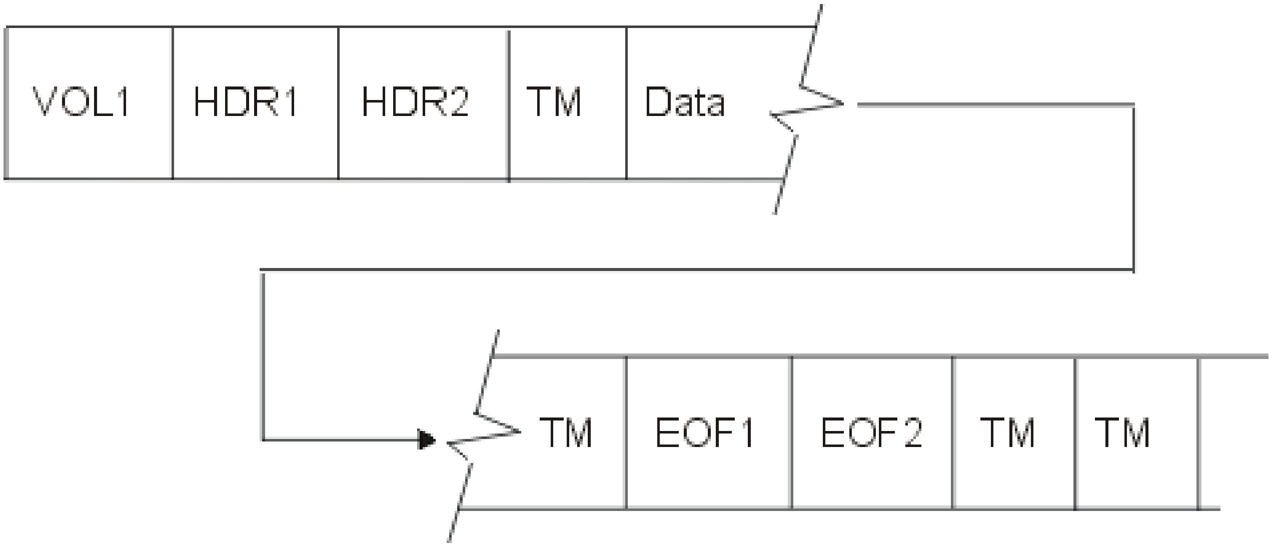


Figure 4. End of file labels

When the high-level language program adds a second file to the tape, the system creates a header label (HDR1) for the new file. This header label (HDR1) for the new file writes over the second tape mark following the end-of-file labels. The second header label, another tape mark, and the file data follow the new header label (HDR1) as illustrated below.

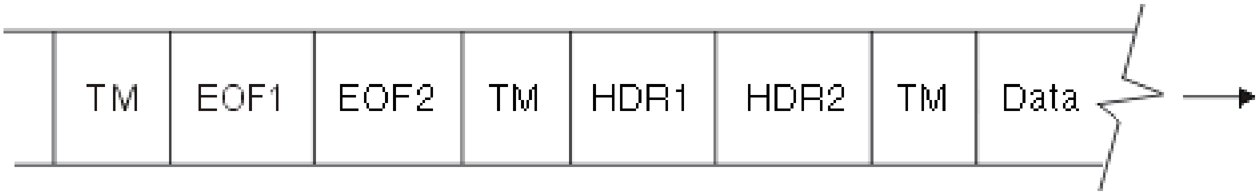
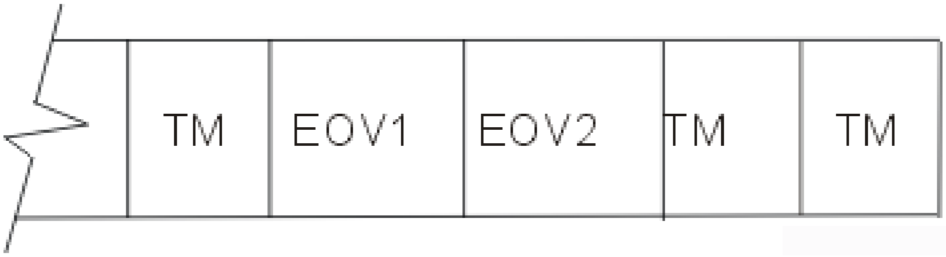


Figure 5. New Header Label

When the tape drive reaches the end of the physical tape, the system writes two tape marks that follow the end-of-volume labels. If the file is not complete, it continues on a second volume, that specifies the tape as volume 2 of the file.



Use tape files

Tape files are used for storing and accessing data on tape media. For information about how to complete those tasks and more, select any of the following topics.

- "Initializing Tapes"
- "Store data in files on tape" on page 8
- "Access data from a tape device" on page 9

Initializing Tapes

Before you begin using tape files, you must initialize all tapes. Use the Initialize Tape (INZTAP) command to initialize tapes, with or without labels. Sometimes you use the Initialize Tape (INZTAP) command to clear all data on the tape.

Note: If the CLEAR(*YES) parameter is specified, the operation can take several hours. The program erases the tape from the beginning-of-tape to the end-of-tape.

The following example initializes the tape volume that is loaded on device TAP01 to standard label format.

```
INZTAP  DEV(TAP01) NEWVOL(BACKUP)
DENSITY(*FMT3490E)
```

To initialize the tape volume with the volume ID `BACKUP`, you specify the character code EBCDIC(by default) and set the tape format to *FMT3490E.

The examples shown below are converting from one character set to another:

- EBCDIC to ASCII
- ASCII to EDCDIC

The iSeries system can convert data with a user-specified conversion table, or with user-specified from and to CCSID values. If no conversion table or CCSID values are specified, the system uses a default data conversion table derived from the American National Standards Institute, Inc, Document ANSI X3.26-1970.

Store data in files on tape

To store data on tape:

- Create a tape device file
- Open the tape device file
- Write the data using an application program

Data files that can be stored on tape include:

- Single volume tape data file: A file that is contained on one volume of tape.
- Multivolume-tape data files: Files that are contained on more than one volume of tape.
- Multifile volumes: Volumes of tape that contain more than one data file.

For information about extending files, see “Extending Files on Tape”

Extending Files on Tape

Tape data files can be extended using the EXTEND parameter on the CRTTAPF, CHGTAPF, and OVRTAPF commands. The system does not support extending tape devices that do not support overwrite capability.

When you extend a file, any existing tape data following the specified file on the tape is no longer accessible by the system.

In the following example, a tape contains four files: FILE1, FILE2, FILE3, and FILE4. If FILE2 is extended, FILE3 and FILE4 are no longer accessible.

Note: If you specify EXTEND(*YES *CHECK) on the OVRTAPF command, the expiration date of the file (FILE3) that is following the extension of (FILE2) is checked. The expiration date will be checked *before* extending the file (FILE2). However, the expiration dates of any remaining files (FILE4) are not checked, even if EXTEND(*YES *CHECK) is specified.

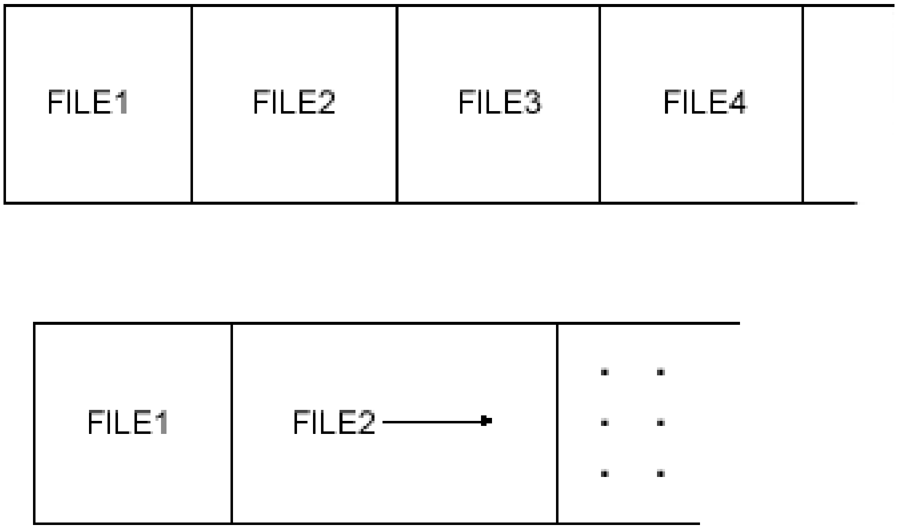


Figure 6. Extending a file

Access data from a tape device

To access data from a tape device on the system, the following objects must exist:

- A device description must exist for each tape device and tape media library device to describe the device to the system. For more information about setting up a device description for each tape device select “Set up a device description.” To learn how to set up a device description for each tape media library device select “Set up a device description for each media library device.”

For more information about creating a tape device file, select “Set up a tape device file.”

Once you have the above objects set up, you can then open the tape device file and proceed by using your application program to read the data on that file.

Set up a device description

You specify the device description using the Create Device Description (CRTDEVTAP) The device description contains information such as the device name, device type, model number, and features. For some tape devices, a tape controller description must also exist. For more information about setting up a Tape controller description, select CRTCTLTAP.

Set up a device description for each media library device

You specify the device description using the Create Device Media Library (CRTDEVMLB) commands. The device description contains information such as the device name, device type, model number, and features. For some tape devices, a tape controller description must also exist.

Set up a tape device file

Create the tape device files by using the Create Tape File (CRTTAPF) command. Tape device files describe how a device presents input data to a program, or how a program presents output data to a device. Do not confuse tape device files with the actual data files on the tape volumes. For processing volumes which contain data files, the tape device files provide a link between the application program and the tape device.

For more information about setting up a tape device file, select any of the following topics.

- “Create tape device file”
- “Specifying Tape Device File Parameters”

Create tape device file: In the following example, the program creates a tape device file, TAP05 in library QGPL, for output that is written to tape:

```
CRTTAPF  FILE(QGPL/TAP05) DEV(TAP01)
REELS(*SL) SEQNBR(3)
CODE(*EBCDIC) ENDOPT(*UNLOAD)
```

The program specifies the tape REELS parameter with the value *SL, indicating that the tape uses standard labels. The device name is TAP01. The program writes a file at sequence number three on the tape (SEQNBR parameter), in EBCDIC code (CODE parameter), and unloads it after it has processed (ENDOPT parameter).

Specifying Tape Device File Parameters: The description of the tape device file record is in the application program that uses the tape information. The system views each record as one field with a length equal to the record length.

The following section lists considerations for parameters that are specified on the CRTTAPF, CHGTAPF, and OVRTAPF commands.

DEV The name of the device description for a tape device file that identifies the devices the file can access.

VOL The volume identifiers of the tapes that are used for the device file may be specified using the VOL parameter on the CRTTAPF, CHGTAPF, and OVRTAPF commands. The volume identifiers may contain from 1 to 6 alphanumeric characters.

REELS

The REELS parameter specifies both the number of tapes that will contain the data file, and the type of label processing that is used by those tapes. Ignore the reel number during output processing or specifying a volume list. Ignore the reel number if you specify standard label processing (by using *SL on the REELS parameter).

If some of the file labels are incorrect, specify bypass label processing (*BLP). The system will check each reel for a volume label that begins with the characters VOL1. The system will ignore most other volume label information and the file labels on the tape.

For bypass label processing, each data file on the tape must contain a header label and either an end-of-file trailer label or an end-of-volume trailer label.

SEQNBR

The SEQNBR parameter specifies the sequence number of the data file on tape. The data files are numbered consecutively across all the volumes they occupy, starting with sequence number 1 for the first data file on the first volume. (Valid values for the sequence number range from 1 to 16 777 215.) Figure 7 on page 11 shows how to number files for labeled volumes that contain more than one file and contain multivolume tapes (FILEB on three volumes):

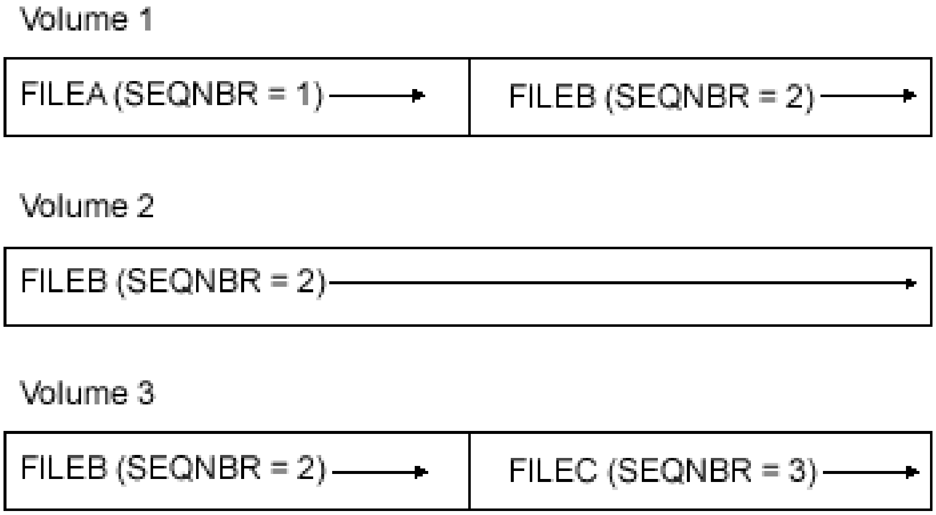


Figure 7. Data file sequence number on multivolume tapes

The sequence number specified for new standard labeled data files on tape must correspond to the physical sequence number of data files on the tape. (Specify the sequence number by the SEQNBR parameter on the CRTTAPF, CHGTAPF, and OVRTAPF commands.) This means that if files 1 and 2 exist on the tape, the next data file created must have a sequence number of 3. To create a new data file on a tape that contains the last volume of a multivolume-tape data file, the sequence number of the new data file must be the sequence number of the last data file on the multivolume tape data file plus 1. In Figure 7, the sequence number of FILEC must be 3, even though there are only two files on the last volume.

Always specify the location of a data file on tape with the SEQNBR parameter. The information specified on the LABEL parameter verifies that you found the correct data file. However, this verification only occurs after the system locates the file that is specified with the SEQNBR parameter. You cannot locate a data file on tape by label name. If you use the Check Tape (CHKTAP) command, the sequence number of the data file returns in the completion message.

You may use some special values in place of an actual sequence number:

- *NEXT: The system processes the next sequential data file on the tape. If you position the tape before the first data file, the system processes the first data file on the tape. *NEXT is useful for applications that need to read all data files on a tape. The system uses this value for tape device files that are used to read from tape. The system issues an error message when you use a tape device file to write to a tape and you specify *NEXT.
- *END: The system writes the data file to the end of the tape. The system uses this value in tape device files that are used to write to tape. The system issues an error message when you use a tape device file to read from a tape and you specify *END.

The SEQNBR parameter for an output file for which EXTEND(*NO) is specified must be one of the following:

- SEQNBR(1). This overwrites the first data file on the volume, regardless of the sequence number in the labels of the first data file already on the volume.

- A value of 1 greater than the value for a data file that already exists on the volume. This either overwrites an existing data file on the volume or adds a data file at the end of the volume.

  Note: If the tape device is a 1/4-inch or 8-mm cartridge device, the program will not overwrite the existing files.
- *END.

LABEL

The LABEL parameter specifies the data file label on the tape.

The information specified on the LABEL parameter is used for new labels created for an output file for which EXTEND(*NO) is specified. It is also used for an I/O file for which EXTEND(*YES) is specified to verify that the correct file is processed.

FILETYPE

The file type of the file to process. The value should be *DATA for a data physical file and *SRC for a source physical file. The program allows this parameter only on the CRTTAPF command.

RCDLEN

The parameter RCDLEN specifies the length of records that are used by a program using this device file. If *CALC is specified, the system attempts to calculate record length from the file header labels. The maximum record length is 32 767 bytes for fixed-length or undefined format records, and 32 759 for variable-length or spanned records. Fixed-length and undefined format output records cannot be less than 18 bytes in length.

BLKLEN

The BLKLEN parameter specifies the data block length that transfers on each I/O operation. If *CALC is specified, the system attempts to calculate block length from the file header labels. The block length must be between 18 and 524 288 bytes.

RCDBLKFMT

The RCDBLKFMT parameter specifies the format of the I/O records and blocks. Records can be:

- D-type ASCII, deblocks (*D)
- D-type ASCII, blocked (*DB)
- Fixed-length, deblocked (*F)
- Fixed-length, blocked (*FB)
- Variable-length, deblocked, unspanned (*V)
- Variable-length, blocked, unspanned (*VB)
- Variable-length, deblocked, spanned (*VS)
- Variable-length, blocked, spanned (*VBS)
- Undefined format (variable length) (*U)

The record length, block length, and record block format may not need to be specified for standard-labeled I/O tape data files specified as EXTEND(*YES). The system can take this information from the tape labels. If the program specifies a block length or record block format that does not match tape label specifications in the tape label, the system then assumes the tape label specification.

If the record length specified in the program does not match the length of the data, the system then truncates or pads the data to the length specified in the program

EXTEND

New records may be added to the end of the data file on the tape by specifying the EXTEND parameter. The destruction of all remaining data files occurs if the data file is not the last data file on the tape. The destruction of all remaining data files also occurs when you over-write an existing data file. The extension uses the label specifications for record and block length that are specified in the label. EXTEND is valid only for 1/2-inch tape devices.

By specifying EXTEND(*YES *CHECK) the system checks the expiration date of the first data file following the data file being extended.

DENSITY
The system records, in the same density, all data files on a volume. You use the DENSITY parameter only to set the output volume density when you create the first data file on a volume that is not labeled. You use the volume label on a labeled tape to determine the density format. For valid values, see the CRTTAPF, CHGTAPF, and OVRTAPF commands in the CL topic.

COMPACT
Allows the user to control device data compaction for output files. If you do not want to use data compaction, specify *NO on the COMPACT parameter. If you specify *DEVD and the device does not support data compaction, the system ignores this parameter.

CODE The CODE parameter specifies the character code (EBCDIC or ASCII) for the data that is not labeled. For standard label tapes, the volume label determines the character code. The system writes ASCII Interchange code when the character code is ASCII. The data conforms to the "American National Standard" X3.27-1978,"Magnetic Tape, and File Structure for Information Interchange".

CRTDATE
The CRTDATE parameter specifies the creation date of an input data file on a labeled tape. The system sends a message to the system operator if the creation date on the tape does not match the date in the file description.

EXPDATE
The EXPDATE parameter specifies the expiration date of an output data file on a labeled tape. The program cannot write-over the data file until the date has expired. The program considers the data file protected.

The program may create an output data file instead of extending an existing data file. When this occurs, the system compares the expiration date of the new data file to the date of the file which precedes it on the volume. If the expiration date of the new data file is later than the file preceding it on the tape, the program sends an inquiry message (CPA4036). The system operator can choose one of the following operations:

- The creation of the data file.
- Load a new tape and try again
- Allow the program to end processing

Note: Creating the data file can produce a volume for which CHECK(*FIRST) on the INZTAP command is unreliable.

If you do not want the data file to be written over, specify *PERM on the EXPDATE parameter.

ENDOPT

The ENDOPT parameter specifies where to position the magnetic tape when the program closes the tape device file. The program:

- Rewinds the magnetic tape to the load point.
- Leaves the magnetic tape as it is.
- Unloads the magnetic tape.

When you use a multivolume-tape data file and specify ENDOPT(*LEAVE), you must place the first volume on the first tape device specified in the DEV parameter. (The exception to this is for a read backward, in which case you must place the last volume on the first tape device specified). If a user opens the data file again, with the same device list, and leaves the tape on a different tape device:

- Place the tape volume on the first tape device that is specified in the DEV parameter before you open the next data file on that tape reel.

Note the following restriction when using *LEAVE processing with tape media libraries. *LEAVE processing restricts the use of the resource that has the current cartridge mounted to that same cartridge. A resource allocation timeout will occur if both of the following conditions exist:

- The device is the only resource available to the media library.
- The program issues a request to use a different cartridge.

The resource will remain unavailable to the program until:

- The program issues a command to rewind or unload the cartridge.
- The job that left the cartridge in *LEAVE processing ends.

USRLBLPGM

The command supports user header and trailer labels through the use of the USRLBLPGM parameter. USRLBLPGM identifies the user program that is used during open and close processing. See “User label processing” on page 19 for more information.

BUFOFSET

The buffer offset length for an ASCII file is specified using the BUFOFSET parameter. You can specify a buffer offset length for any ASCII input data file. You can specify a buffer offset value of

- *BLKDSC for an input or output ASCII formats *D file
- *BLKDSC for an input or output ASCII formats *DB file

to process a block with 4-digit block descriptors.

TBL A conversion table to use for data conversion is specified using the TBL parameter. If *NONE is specified, then no data conversion is performed. If *CCSID is specified, then the CCSID values specified by the FROMCCSID and TOCCISD parameters are used to determine the data conversion to perform. A conversion table can also have a special value of *DFT. If the code is *ASCII (CODE parameter) and TBL(*DFT) is specified, the data and labels will be converted between ISO/ASCII 8-bit code and EBCDIC. When the code is *EBCDIC (CODE parameter) and TBL(*DFT) is specified, the data and labels will not be converted.

FROMCCSID

This parameter is used to specify a CCSID value for the input data. The CCSID specified must be a single-byte CCSID.

TOCCSID

This parameter is used to specify a CCSID value for the output data. The CCSID value must be a single-byte CCSID.

For additional tape information, and information about using tape for save and restore operations, see the Backup and Recovery topic. Table 1 lists parameters that apply to magnetic tape and where to specify the parameters. The CL topic in the Information Center contains detailed information about how to specify these parameters on the CRTTAPF, CHGTAPF, and OVRTAPF commands.

Table 1. Tape Device File Parameters

CL Parameter	Description	Specified on CRTTAPF Command	Specified on OVRTAPF Command	Specified in HLL Programs
FILE	File name	Qualified file name	File name	ILE RPG, COBOL, BASIC, PL/I, or ILE C programming languages
DEV	Device name	*NONE or list of device names	List of device names	
VOL	Volume	*NONE or list of volume identifiers	*NONE or list of volume identifiers	
REELS	Volume label type	*SL, *NL, *NS, *BLP, or *LTM	*SL, *NL, *NS, *BLP, or *LTM	

Table 1. Tape Device File Parameters (continued)

CL Parameter	Description	Specified on CRTTAPF Command	Specified on OVRTAPF Command	Specified in HLL Programs
REELS	Number of labeled tapes	Number of reels	Number of reels	
SEQNBR	Sequence number	*NEXT, *END, or sequence number of file	*NEXT, *END, or sequence number of file	
LABEL	Label	*NONE or file label	*NONE or file label	BASIC
FILETYPE	File type	*DATA or *SRC		
RCDLEN	Record length	*CALC or record length	*CALC or record length	ILE RPG, COBOL, BASIC, PL/I, or ILE C programming languages
BLKLEN	Block length	*CALC or block length	*CALC or block length	COBOL programming language
BUFOFSET	Buffer offset	*BLKDSC or buffer offset	*BLKDSC or buffer offset	
RCDBLKFMT	Record block format	*F, *FB, *V, *VB, *D, *DB, *VS, *VBS, or *U	*F, *FB, *V, *VB, *D, *DB, *VS, *VBS, or *U	COBOL, ILE C programming languages
EXTEND	Extend	*NO, *YES *CHECK, or *YES *NOCHECK	*NO, *YES *CHECK, or *YES *NOCHECK	COBOL, ILE C programming languages
DENSITY	Density	See the CL topic.	See the CL topic.	
COMPACT	Data compaction	*DEVD or *NO	*DEVD or *NO	
CODE	Character code	*EBCDIC or *ASCII	*EBCDIC or *ASCII	COBOL programming language
CRTDATE	Creation date	*NONE or date	*NONE or date	
EXPDATE	Expiration date	*NONE, date, or *PERM	*NONE, date, or *PERM	
ENDOPT	End option	*REWIND, *LEAVE or *UNLOAD	*REWIND, *LEAVE or *UNLOAD	COBOL programming language
USRLBLPGM	User label program	*NONE or qualified program name	*NONE or qualified program name	
IGCDTA	Double-byte data	N/A	*NO or *YES	
WAITFILE	File wait time	*IMMED, *CLS, or number of seconds	*IMMED, *CLS, or number of seconds	
SHARE	Shared file	*NO or *YES	*NO or *YES	
AUT	Authority	*LIBCRTAUT, *CHANGE, *ALL, *USE, *EXCLUDE, or authorization list name	N/A	
REPLACE	Replace existing file	*YES or *NO	N/A	
TEXT	Text	*BLANK or text	N/A	
TBL	Conversion table	N/A	Table name or library, *NONE, *CCSID, *DFT	
FROMCCSID	From CCSID	N/A	1 to 65533	
TOCCSID	To CCSID	N/A	1 to 65533	

Use Tape Device Files in High-Level Language Programs

A program-described device file can access a magnetic tape device. To use a tape device file with a program, you must either specify the tape file name in the program or use an Override with Tape File (OVRTAPF) command. The high-level language determines what tape parameters to use in the program.

Select any of the links below for information pertaining to the different processing types for tape device files used in high-level language.

- “Open Processing for Tape Device Files” on page 16
- “Input/Output Processing for Tape” on page 18

- “Close Processing for Tape” on page 18
- “User label processing” on page 19

Open Processing for Tape Device Files

The following considerations apply to opening tape device files:

- When the program opens a tape device file, the system merges any parameters that are specified in the file with the parameters that are specified in the program. The system then merges these parameters with the parameters that are specified on an OVRTAPF command.
- Specify the device names when you open the tape device file. If you specify DEV(*NONE) in the tape file, you must specify one or more device names on an OVRTAPF command. You can specify as many as four device names for a single tape device file (depending on how many magnetic tape devices you have).

  The record length, block length, record-block-format, and buffer offset (for an ASCII file) always return to the program in the data management open feedback area. They return in the format in which they are written in the HDR2 file header label. This information is available regardless of the type of label processing that is used for the file.
- The following data files support the read-backward operation for both single colume tape data files and multivolume-tape data files:
  - fixed (*F)
  - fixed block (*FB)
  - undefined format (*U)

  You request a read backward operation through a high-level language when you open the file. An escape message is signaled if a read-backward operation is attempted for variable-length (spanned or not spanned) or source records.

  Note: The following tape devices do not have read-backward capabilities: Use the Retrieve Device Capabilities (QTARDCAP) to determine the device capabilities of your device.
  - 9348 tape unit
  - 8-mm cartridge device
  - Some 1/4-inch cartridge devices

  Reading a data file backwards, where you specify the device and volume list, you must position the volumes on the device in reverse order. For example, a device file with DEV(QTAPE1 QTAPE2) VOL(VOL01 VOL02 VOL03) expects VOL03 on QTAPE1, VOL02 on QTAPE2, and then VOL01 on QTAPE1.

  For a read-backward operation, the end-of-file condition occurs if the system recognizes the first volume of the data file from the header labels for the following:
  - Standard label processing (*SL)
  - Bypass label processing (*BLP)

  If the system does not recognize the header labels for the first volume of the data file, or if this is a *BLP file, the system signals the end-of-file condition when:
  - The system processes the specified number of reels.
  - The system processes the number of identifiers on the VOL parameter.
- Some high-level languages allow you to specify where to position the tape when the program opens an input tape device file. This indicates whether you process the tape in the forward or in the backward direction. These rules determine the first volume of a data file:
  - HDR1 labels multivolume sequence field = 1 (ASCII or EBCDIC with no HDR2 label)

    or
  - HDR2 label volume switch indicator field = 0 (EBCDIC)

The program specifies the record length according to the information that is shown in Table 2.

- For source files, the record length used to determine the block length is the actual data length, not the data length plus 12 bytes (for sequence number and date).
- You must supply either a RCDLEN or BLKLEN parameter value for unspanned, deblock records (*F, *V, *D, *U).
- You must supply both RCDLEN and BLKLEN parameter values for spanned or blocked records (*FB, *VB, *DB, *VS, *VBS).
- When the file type specified in the tape device file is a source file:
  - The system appends a date and sequence number to each record on input operations. The date field is always 0.
  - The system removes the date and sequence number from each record on output operations

  The program can check (if the high-level language you are using allows it) to determine if the input or output file is a source file. The record length must include 12 bytes for the date and sequence number. The block length and record length ratio remains the same for a source block and data record minus the 12 bytes allocated to source files. For example, if the actual data record length is 80 the record length becomes 92 for a source file. The block length remains unchanged.
- To process input files using standard labels the system will always use the block length in the file label. The device file block length is ignored.
- Variable-length (spanned or unspanned) records and undefined format records can be used for output files. If your high-level language does not support variable-length records, then all records for an output tape device file that uses variable format are maximum length.
- Specify the sequence number so you can find the data file. You cannot locate tape data files by label name.
- When you specify both the VOL and REELS parameters the REELS parameter is ignored. If you want to use the REELS parameter (number of reels) to limit the number of input volumes that is processed, specify *NONE for the VOL parameter.

Table 2. Specifying Record Lengths by Record and Format Type

Record and Format Type	Minimum Record Length for *DATA	Minimum Record Length for *SRC	Maximum Record Length for *DATA	Maximum Record Length for *SRC	Block Length
Fixed blocked, *F, *FB, *U	18	30	32 767	32 767	Multiple of *DATA record length
Variable unblocked, *V	1	13	32 759 (See Note)	32 767	Equal to maximum *DATA record length plus 8
D-type ASCII unblocked, *D	1	13	9 995 (See Note)	10 007 (See Note)	Equal to maximum *DATA record length plus 4, plus buffer offset
Variable blocked, *VB	1	13	32 759	32 767	At least maximum *DATA record length plus 8
D-type ASCII blocked, *DB	1	13	9 995 (See Note)	10 007 (See Note)	At least maximum *DATA record length plus 4, plus buffer offset
*VS, *VBS	1	13	32 759	32 759	

Note: This is the maximum record length for a record being written to a tape. Input records can be padded to 32 767.

Input/Output Processing for Tape

The following considerations apply to I/O operations that are performed on data files:

- “Read and Write Considerations”
- “Read Considerations”
- “Force-End-of-Data Considerations”
- “Force-End-of-Volume Considerations”

Read and Write Considerations

- Specify record lengths in your program when writing variable-length records (*D, *DB, *V, *VB, *VS, *VBS, or *U specified on the RCDBLKFMT parameter on the CRTTAPF command).

  If the maximum record length on tape is shorter than the output record length (because of overrides or existing file labels):

  – The system truncates the records to the maximum length that is allowed.

  – The system sends a diagnostic message when you open a device file to indicate that output records may be truncated.
- If the program specifies a record length that differs from the actual length of the data, the system pads or truncates the data to match the program specification.

Read Considerations

- If the system does not find an end-of-file label, processing will continue until the system uses all the specified volume identifiers. If the program specifies VOL(*NONE), then the system processes the tapes until reaching the specified number of reels (REELS parameter). The system sends a message CPA5230 to the system operator message queue when all the identifiers in the VOL list are processed. When you receive this message, you can do the following:

  – Cancel processing of the device file immediately. The system will close the device file.

  – Continue, in order to process other volumes.
- When the system reads a tape block that is not a valid length, the system sends a CPF5036 notify message. If your high-level language reports this condition to your program, it can continue to process by reading another record. When you continue this way, the system skips the block that is not valid so your program does not receive any records from this block.

Force-End-of-Data Considerations

The force-end-of-data function is valid for both input and output. The force-end-of-data function for an output file forces the system to write all buffered records to tape. When this occurs, the system may write a short block on the volume. The force-end-of-data function for an input file positions the tape at the last volume for the file and signals end-of-file to the using program.

Force-End-of-Volume Considerations

The force-end-of-volume function is valid for both input and output files. It causes volumes to switch immediately, or signals end-of-file if there are no more continuation volumes for an input file.

Close Processing for Tape

When you close a tape device file, the system performs one of several functions according to what you specify in the tape device file. The system uses the ENDOPT parameter on a CRTTAPF, CHGTAPF, or an OVRTAPF command. The system can perform the functions that are listed below:

- The system rewinds the tape.
- The system leaves the tape as it is.
- The system rewinds and unloads the tape to remove it from the magnetic tape device.

If the tape operation ends abnormally:

- The position of the tape when you close the device file may be unchanged.

- The system may rewind the tape regardless of specified program instructions.

User label processing

The system uses the USRLBLPGM parameter to specify the name of a program used to process user header and trailer labels. This parameter is not valid for save and restore functions.

The system calls the program, which the USRLBLPGM parameter specifies, for open and close processing, to process every label. The system calls the program one additional time to tell the program that there are no more labels.

Figure 8 on page 20 shows the format of a tape with user labels. At open, the system will call the user-label program three times for processing the labels in the diagram. Those calls are for UHL1, UHL2, and a final time to indicate completion. At close, the system will call the user-label program three more times.

The system passes three variables to the program. The program variables have the following lengths:

- Parameter 1: 80 characters
- Parameter 2: 1 character
- Parameter 3: 244 characters

The following list shows the parameters the variables pass to the user label program:

- "Parameter 1"
- "Parameter 2"
- "Parameter 3" on page 20

Parameter 1

Position 1 - 80

User header or trailer label

- For output files the program sets this variable to the next user label to write to tape.
- For input files written to the system it sets this variable to the user label that is most recently read from the tape.

Parameter 2

Position 1

End-of-labels indication

Parameter 2 contains a character `0` or `1`, that indicates whether the label read is the last label read. For output files, the user label program sets the value. For input files, the system sets the value.

- `0` indicates that Parameter 1 contains a label.
- `1` indicates that Parameter 1 does not contain a label. There are no labels to process.

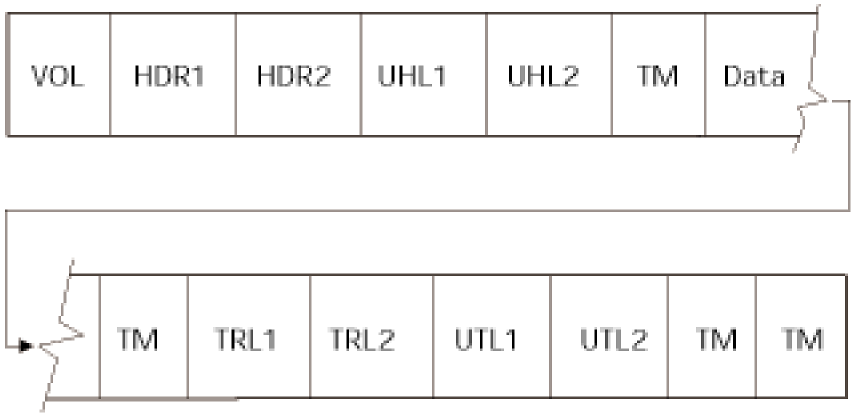


Figure 8. Tape with User Labels

Parameter 3

Position 1- 80
Current volume label

Position 81-160
The last processed HDR1 or TRL1 label.

Position 161-240
The last processed HDR2 or TRL2 label.

Position 241-242
User label number

- Output files: The number of the next user label to be written in the current header or trailer group.
- Input files: The total count of user labels read in the current header or trailer group.

Position 243
Open file option

The open file option field contains a character that indicates whether the file is open for input or for output.

- `I` Indicates that the file is an input file.
- `O` Indicates that the file is an output file.

Position 244
Expect labels

The expect labels field contains an integer indicating whether the call is returning or expecting labels from the user program.

- `0` indicates that the user program is returning labels.
- `1` indicates that the user program is expecting labels.

Use overrides

You can use overrides to temporarily change a file name, a device name associated with the file, or some of the other attributes of a file. Overrides allow you to make minor changes in how a program functions. Overrides allow you to select the data, on which they operate, without recompiling the program.

Select any of the links below for information about what can be done with an override and how to manage the overrides that currently exist.

- “Overriding File Attributes”
- “Overriding File Names in HLL Programs” on page 22
- “Deleting Overrides” on page 24
- “Displaying Overrides” on page 23
- “File Redirection” on page 24

| **Note:** When using overrides, the Open Scope parameter (OPNSCOPE) should be set to *JOB .

Overriding File Attributes

File attributes are built as a result of the following:

- Create file command. Initially, this command builds the file attributes.
- Program using the files. At compile time, the user program can specify some of the file attributes. The high-level language used by the program determines the attributes that can be used.
- Override commands. At program run time, these commands can override the file attributes previously built by the merging of the file description and the file parameters specified in the user program.

The simplest form of overriding a file is to override some attribute of the file.

For example, assume that you create a tape file OUTPUT whose attributes are:

- Use Device TAP01.
- Write the Data on the tape with a density of 1600 bits per inch (bpi)
- Use the ASCII character code type.
- When you close the tape file, the program rewinds and unloads the tape.

The Create Tape File (CRTTAPF) command looks like this:

```
CRTTAPF  FILE(QGPL/OUTPUT) DEV(TAP01)
DENSITY(1600) CODE(*ASCII) ENDOPT(*UNLOAD)
```

Your application program has the tape file OUTPUT specified with a character code type of EBCDIC and a density of 3200. However, before you run the application program, you want to change the density to 6250 bpi and the end option to *REWIND. The override command looks like this:

```
OVRTAPF  FILE(OUTPUT) DENSITY(6250)
ENDOPT(*REWIND)
```

When you call the application program, the system uses a tape density of 6250 bpi and the end option is *REWIND.

When the application program opens the file the system merges the following to form the open data path (ODP):

- The file overrides.
- The program-specified attributes.
- The file attributes.

The program uses the open data path (ODP) during the running of the program. File overrides have precedence overprogram-specified attributes. Program-specified attributes have precedence over file-specified attributes. In Figure 9 on page 22, when you open the file and you perform output operations, the program writes:

- To the device TAP01 with a density of 6250 bpi.
- In a character code type of EBCDIC.
- An end option of *REWIND.

Figure 9 explains this example.

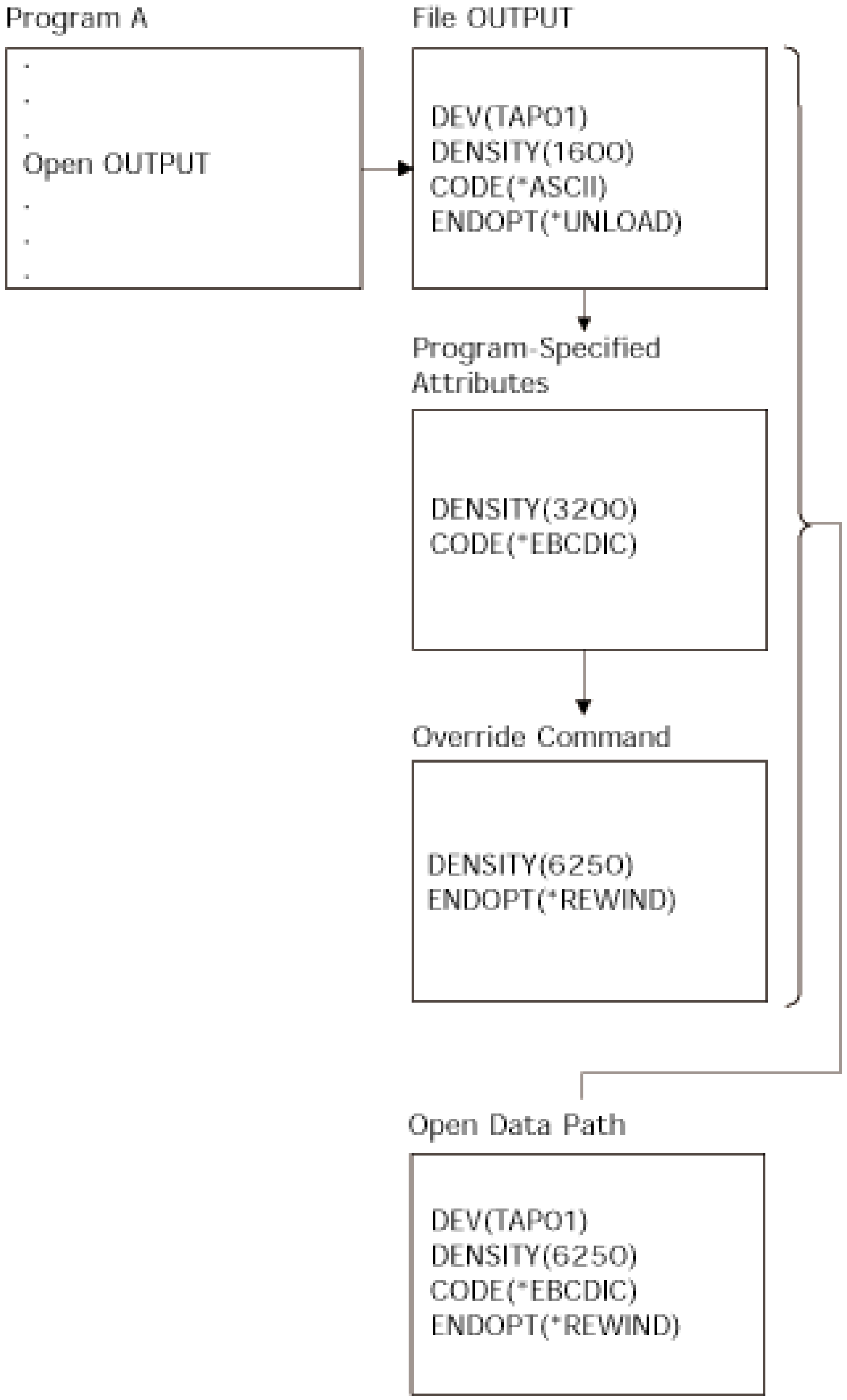


Figure 9. Overriding File Attributes

Overriding File Names in HLL Programs

Another simple form of overriding a file is to change the file that is used by the program. This can be useful for moved or renamed files after you compile the program. For example, you want to send the output from your application program to tape file TAPE20 instead of the tape file OUTPUT1 (The application program specifies OUTPUT1). Before you run the program, enter the following:

```
OVRTAPF FILE(OUTPUT1) TOFILE(TAPE20)
LABEL(FILE01) OPNSCOPE(*JOB)
```

You must create the file TAPE20 by a CRTTAPF command before you use it.

Figure 10 explains this example.

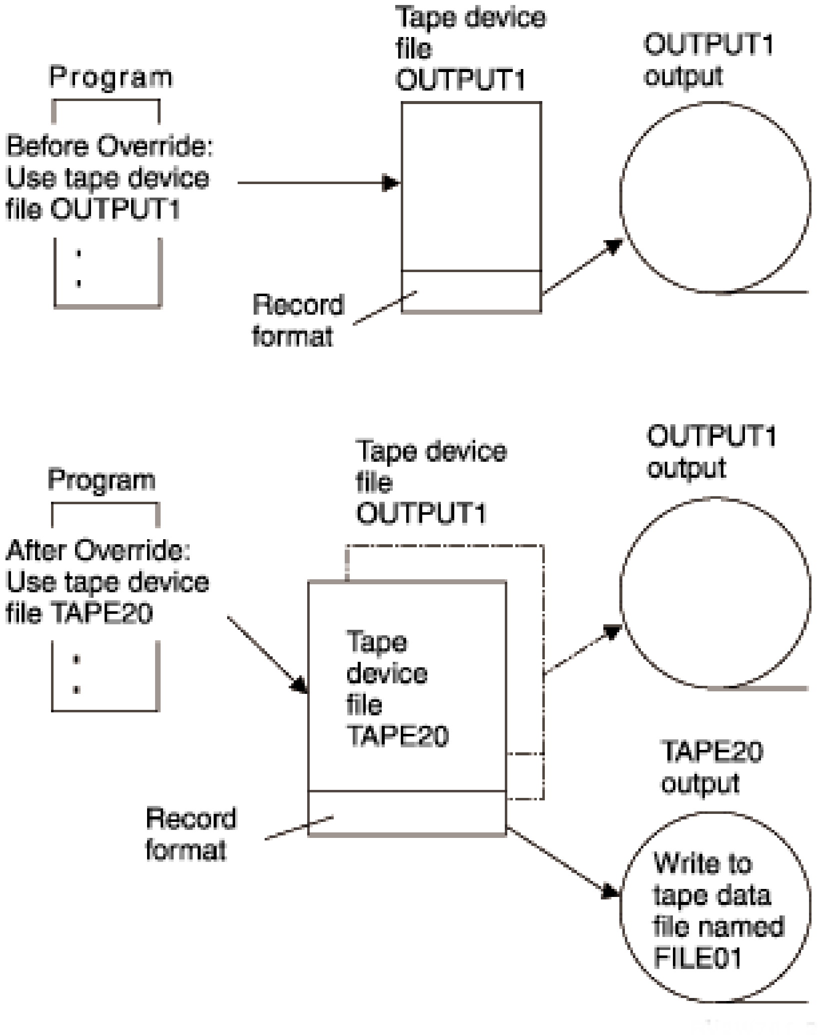


Figure 10. Overriding a File Name

You may want to override a file with a file that has a different file type. For example, you may want to override a diskette file with a display file. To determine if you can override your file with a file that has a different type, see the information under “File Redirection” on page 24. More information is available in the File Management topic in the Information Center.

Displaying Overrides

You can display all file overrides or file overrides for a specific file using the Display Override (DSPOVR) Command.

The DSPOVR command displays the overrides used by an application that either calls or transfers control to other programs. You can control which overrides are displayed. More information about displaying overrides used in application programs is available in the File Management topic.

Deleting Overrides

If you want to delete an override, you can use the Delete Override (DLTOVR) command.

The DLTOVR command, in an application that either calls or transfers control to other programs, may or may not delete the override. More information about deleting overrides in application programs is available in the File Management topic.

File Redirection

File redirection refers to using overrides to change the file name and library or the type of the file you process. For example, you can substitute:

- One tape file for another tape file.
- One diskette file for another diskette file.
- Or change from using a tape or diskette file to using a display file, printer file, ICF file and so on.

System code may or may not support file redirection. See “Recognizing Commands that Ignore or Restrict Overrides” on page 26 for rules on how system code processes overrides.

For more information about file redirection select any of the following:

- “Overriding Files with the Same File Types”
- “Overriding Files with Different File Types”
- “Recognizing Commands that Ignore or Restrict Overrides” on page 26

Overriding Files with the Same File Types

When a program replaces a file that is used with another file of the same type, the system processes the new file in the same manner as the original file. If you redirect a field level file, or any other file containing externally described data, it is typically necessary to specify the command LVLCHK(*NO) or recompile the program. With level checking that is turned off, it is still necessary that the record formats in the file be compatible with the records in the program. If the formats are not compatible, you cannot predict the results.

Overriding Files with Different File Types

If you change to a different type of file, the system ignores the device-dependent characteristics and reads or writes the records sequentially. The program must specify some device parameters in the new device file or the override. The system takes default parameters for others. This manual describes the effect of specific redirection combinations later in this section.

The system ignores any attributes specified on overrides of a different file type than the final file type. The parameters SPOOL, SHARE, OPNSCOPE, and SECURE are exceptions to this rule. The system accepts the above parameters from any override to the file, regardless of device type.

Some redirection combinations present special problems due to the specific characteristics of the device. In particular:

- Do not use file redirection for save files.
- You can redirect nonsequentially processed database files only to another database file or a DDM file.
- You can redirect display files and ICF files that use multiple devices (MAXDEV or MAXPGMDEV > 1) only to a display file or ICF file.
- Redirecting a display file to any other file type, or another file type to a display file, requires:
  - That the program be recompiled, with the override active, if there are any input-only or output only fields.

  This is necessary because the display file omits these unused fields from the record buffer, but other file types do not.

Table 3 on page 25 summarizes valid file redirections.

Table 3. File Redirections

	From-File					
To-File	**Printer**	**ICF**	**Diskette**	**Display**	**Database**	**Tape**
Printer	O*	O	O	O	O	O
ICF		I/O		I/O		
	O	O	O	O	O	O
		I	I	I	I	I
Diskette	O	O	O	O	O	O
		I	I	I	I	I
Display		I/O		I/O		
	O	O	O	O	O	O
		I	I	I	I	I
Database	O	O	O	O	O	O
		I	I	I	I	I
Tape	O	O	O	O	O	O
		I	I	I	I	I

- I=input file O=output file I/O=input/output file
- *=redirection to a different type of printer

To use Table 3 on page 25, identify the file type to override in the FROM-FILE columns and the file type to override in the TO-FILE column. The intersection specifies an I or O or both, meaning that the substitution is valid for these two file types when used as input files or as output files.

For instance, you can override a diskette output file with a tape output file, and a diskette input file with a tape input file. The chart refers to file type substitutions only. That is, you cannot change the program function by overriding an input file with an output file.

The following charts describe the specific defaults taken andwhat to ignore for each redirection combination.

From: Diskette Input

To: ICF: Records are retrieved from the ICF file one at a time.

Display: Records are retrieved from the display one at a time. Type in the data for each record and press the Enter key when the record is complete. A nonfield-level device file must be specified. Diskette label information is ignored.

Database: Records are retrieved in sequential order. Diskette label information is ignored.

Tape: Records are retrieved in sequential order. If a label value is specified in the program, that value is used as the label for the tape file.

From: Diskette output

To: ICF: Records are written to the ICF file one at a time.

Database: Records are written to the database in sequential order.

Display: Records are written to the display with each record overlaying the previous record. You can request each output record using the Enter key.

Printer: Records are printed and folding or truncating is performed as specified in the printer file.

Tape: Records are written on tape in sequential order.

From: Tape input

To: ICF: Records are retrieved from the ICF file one at a time.

Display: Records are retrieved from the display one at a time. Type in the data for each record and press the Enter key when the record is complete. A nonfield-level device file must be specified. Tape label information is ignored.

Database: Records are retrieved in sequential order. One record is read as a single field. Tape label information is ignored.

Diskette: Records are retrieved in sequential order. If a label value is specified in the program, that value is used as the label for the diskette file.

From: Tape Output

To: Printer: Records are printed, and folding or truncating is performed as specified in the printer file.

ICF: Records are written to the ICF file one at a time. Tape label information is ignored.

Diskette: The amount of data written on diskette depends on the exchange type of the diskette. If a label value is specified in the program, that value is used as the label for the diskette file.

Display: Records are written to the display with each record overlaying the previous record. You can request each output record using the Enter key.

Database: Records are written to the database in sequential order.

Recognizing Commands that Ignore or Restrict Overrides

The following commonly used commands ignore overrides entirely:

ALCOBJ
CHGOBJOWN
CPYIGCTBL
CRTDKTF
CRTDUPOBJ
CRTTAPF
DLCOBJ
DLTF
DSPFD
DSPFFD
EDTOBJAUT
GRTOBJAUT
MOVOBJ
RNMOBJ
RVKOBJAUT

The system does not apply overrides to any system files that you openas part of an end-of-routing step or end-of-job processing. For example, you cannot specify overrides for the job log file. In some cases, when you need to override something in a system file, you may be able to change it through a command other than an override command. For example, to change the output queue for a job log, the output queue can be changed before sign-off using the OUTQ parameter on the Change Job (CHGJOB) command to specify the name of the output queue for the job. If the printer file for the job log contains the value *JOB for the output queue, the output queue is the one specified for the job. The save and restore commands open the tape file with SECURE(*YES) so that the tape file overrides will be ignored.

The following commands allow overrides for the SRCFILE and SRCMBR parameters only:

CRTCMD
CRTICFF
CRTDSPF
CRTLF
CRTPF
CRTPRTF
CRTSRCPF
CRTTBL
CRTXXXPGM

(All create program commands. These commands also use overrides to determine which file will be opened by a compiled program. See the File Management topic for more information about applying overrides when compiling a program.)

The following commands allow overrides, but do not allow changing the MBR to *ALL:

CPYFRMPCD
CPYTOPCD

The following commands do not allow overrides to be applied to the display files they use. Overrides to the printer files they use should not change the file type or the file name. Various restrictions are placed on changes that may be made to printer files used by these commands, but the system cannot guarantee that all combinations of possible specifications will produce an acceptable report.

DMPTAP
(In addition to the preceding limitations, this command does not allow overrides to the file it dumps.)

DSPDKT and DSPTAP
(The display commands that display information about a file do not allow overrides to that file.)

Record formats

This topic provides examples of record formats. For information to help you understand records, blocks, and record formats of the tapes, select “Records, blocks, and formats” on page 4.

Click any of the links below to view examples of the different record formats.

- “Example - Record format *D”
- “Example - Record format *DB” on page 28
- “Example - Record format *F” on page 29
- “Example - Record format *FB” on page 30
- “Example - Record format *V” on page 31
- “Example - Record format *VB” on page 31
- “Example - Record format *VS” on page 33
- “Example - Record format *VBS” on page 36
- “Example - Record format *U” on page 39

Example - Record format *D

D-type Code ASCII, deblocks (*D).

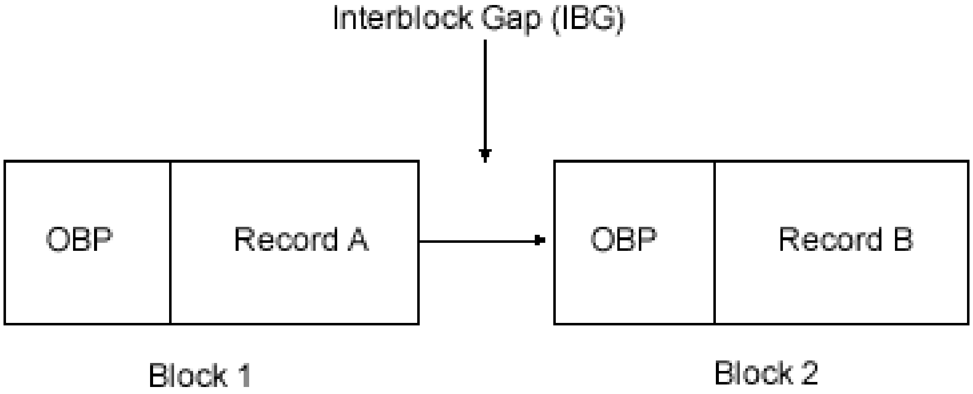


Figure 11. Blocks with Interblock Gap

OBP is an optional block prefix. The buffer offset (BUFOFSET) parameter, for the tape file, specifies the optional block prefix that can vary in length from 00 to 99. The OBP length must be constant for all blocks in the file. Each record can also contain an optional control character.

Example - Record format *DB

D-type code ASCII, blocked (*DB).

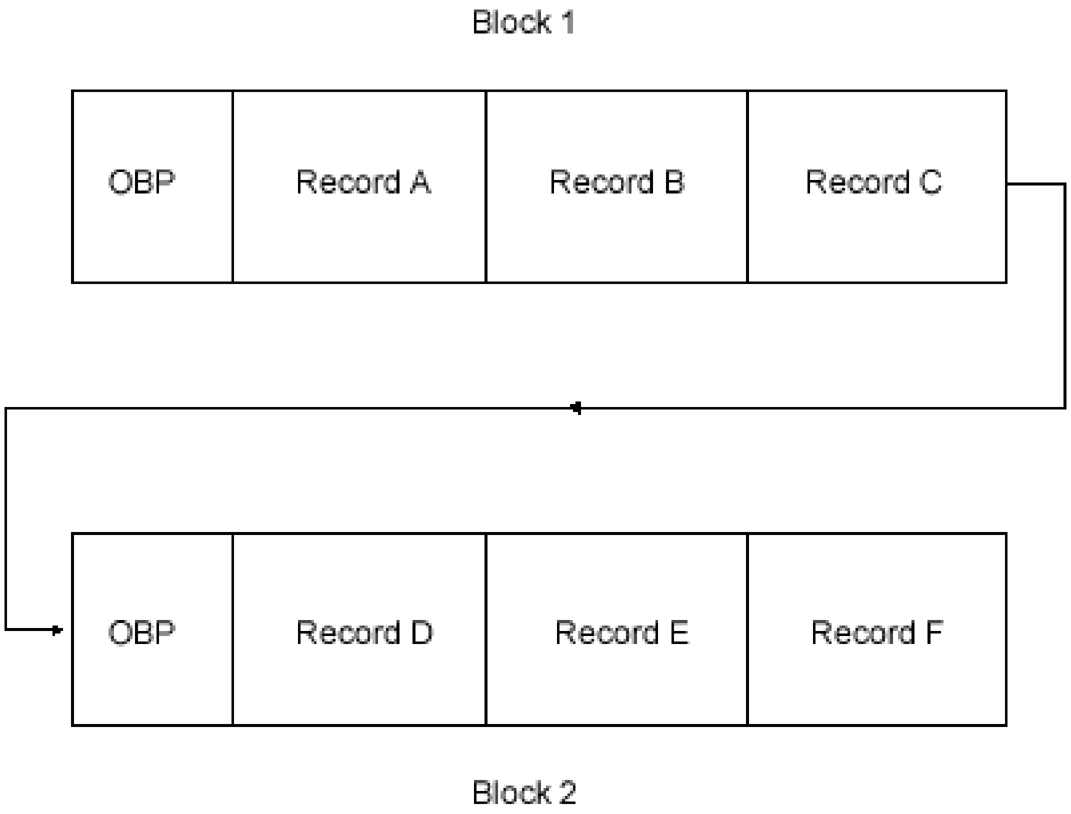


Figure 12. Records with Optional Block Prefixes

OBP is an optional block prefix. The buffer offset (BUFOFSET) parameter, for the tape file, specifies the optional block prefix that can vary in length from 00 to 99. The OBP length must be constant for all blocks in the file. Each record can also contain an optional control character.

Example - Record format *F

Fixed-length, deblocked (*F).

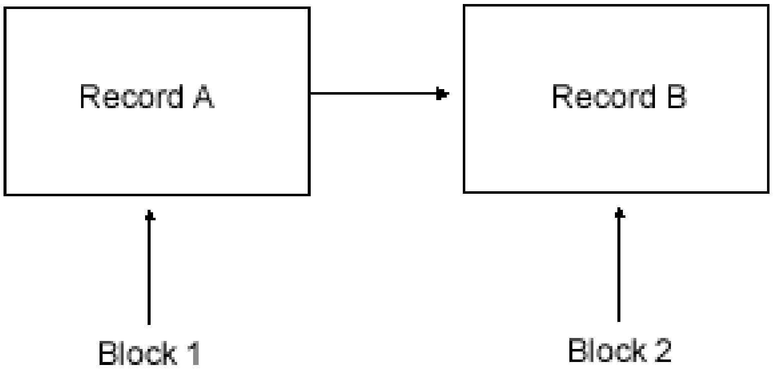


*Figure 13. Fixed-length, deblocked (*F)*

Example - Record format *FB

Fixed-length, blocked (*FB).

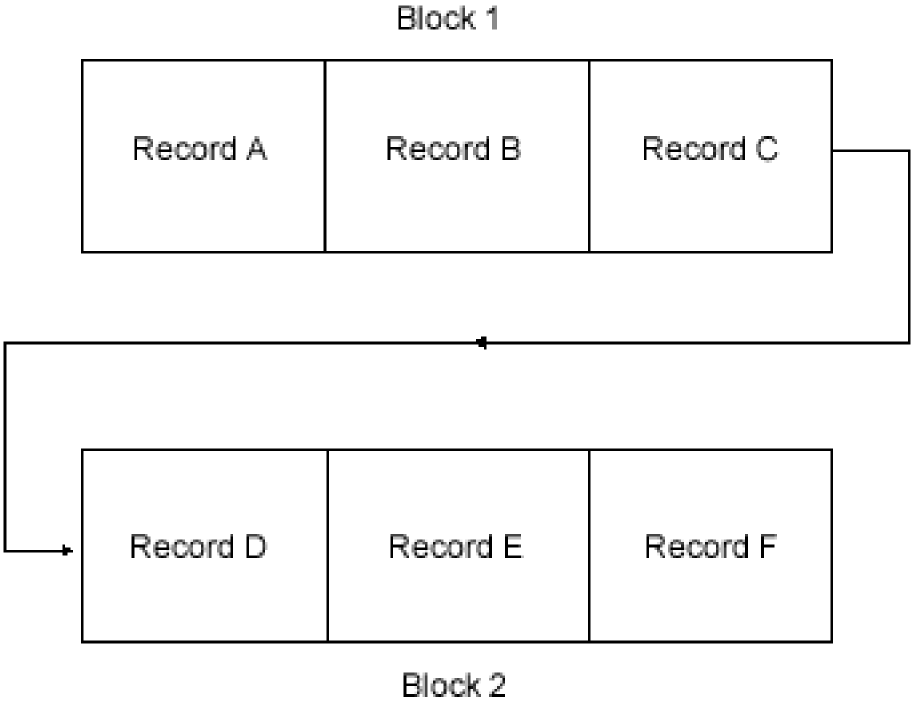


*Figure 14. Fixed-length, block (*FB)*

Example - Record format *V

Variable-length, deblocked, unspanned (*V).

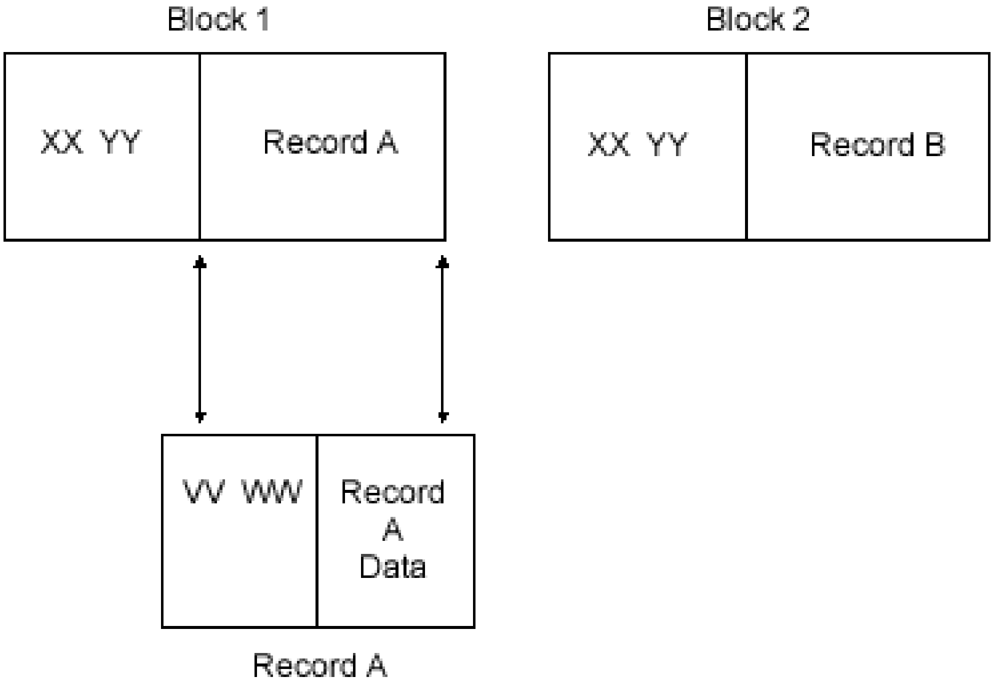


*Figure 15. Variable-length, deblocked, unspanned (*V)*

XX and YY make up the BDW. XX is the length of the record plus the length of the BDW (4 bytes). YY are currently reserved fields and must be 00. XX should be the actual length of the data block that is written.

Record A has its own mapping by including the RDW. VV and WW make up the RDW. VV is the length of the record plus the length of the RDW (4 bytes). WW are currently reserved fields and must be 00. VV should be the actual block length (the value in XX) minus 4 bytes (size of BDW). OS/400 will pad blocks to make the 18 byte block limit. This occurs if the record data is less than 10 bytes (10 bytes plus 8 bytes of header information is the 18 byte block limit). OS/400 will pad the block with a byte of x'80' followed by bytes of x'00'.

Note: Record format *V tapes created by OS/400 that contain record data that is less than 10 bytes are not directly compatible with OS/390® and z/OS®. To process these tape with OS/390 and z/OS you will need to code RECFM=VS or RECFM=VBS on the DD statement even though the records are not spanned. This will allow OS/390 and z/OS to read the tapes without error.

Example - Record format *VB

Variable-length, blocked, unspanned (*VB).

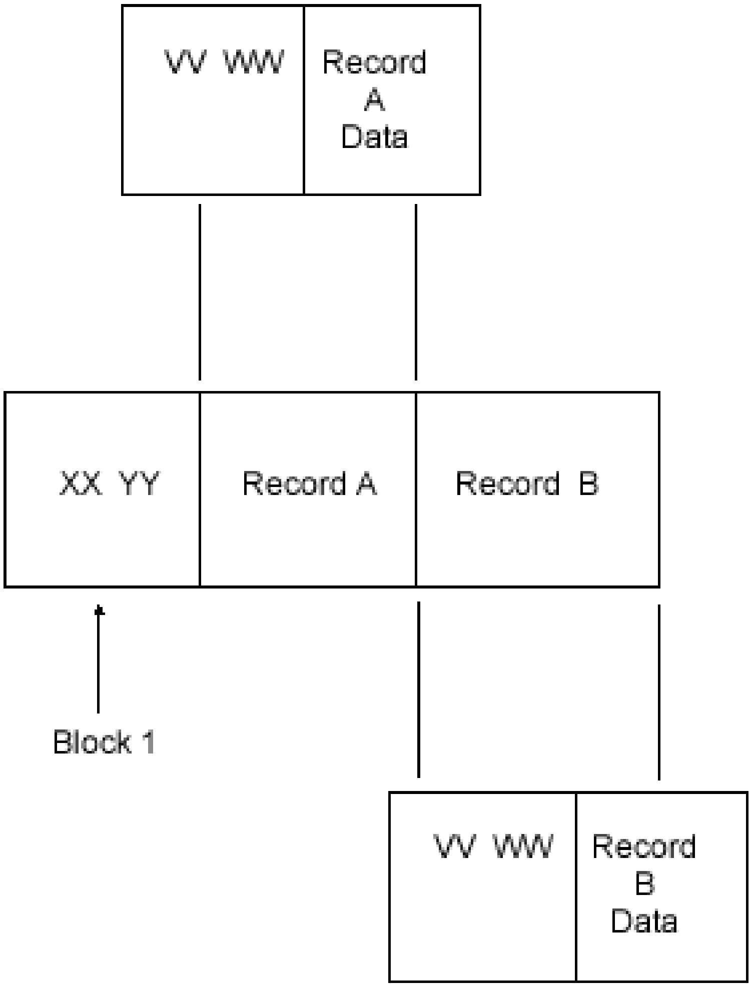


*Figure 16. Variable-length, blocked, unspanned (*VB)*

XX and YY make up the BDW. XX is the length of all records plus the length of the BDW (4 bytes). YY are currently reserved fields and must be 00. XX should be the actual length of the data block that is written.

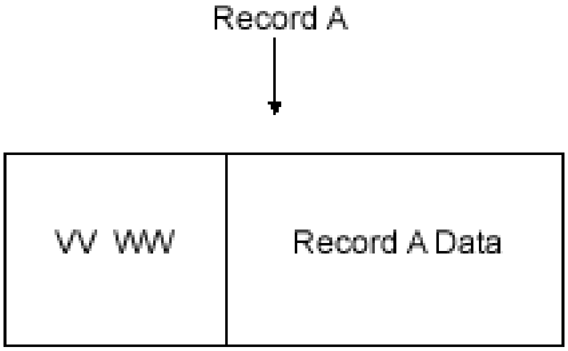


Figure 17. Record A mapping

Record A has its own mapping by including the RDW. VV and WW make up the RDW. VV is the length of the record plus the length of the RDW (4 bytes). WW are currently reserved fields and must be 00. VV should be the actual length of record data A plus the length of the RDW (4 bytes). The actual length of the data block equals the sum of:

- The VV value for ALL records.
- The length of the BDW (4 bytes).

Example - Record format *VS

Variable-length, deblocked, spanned (*VS).

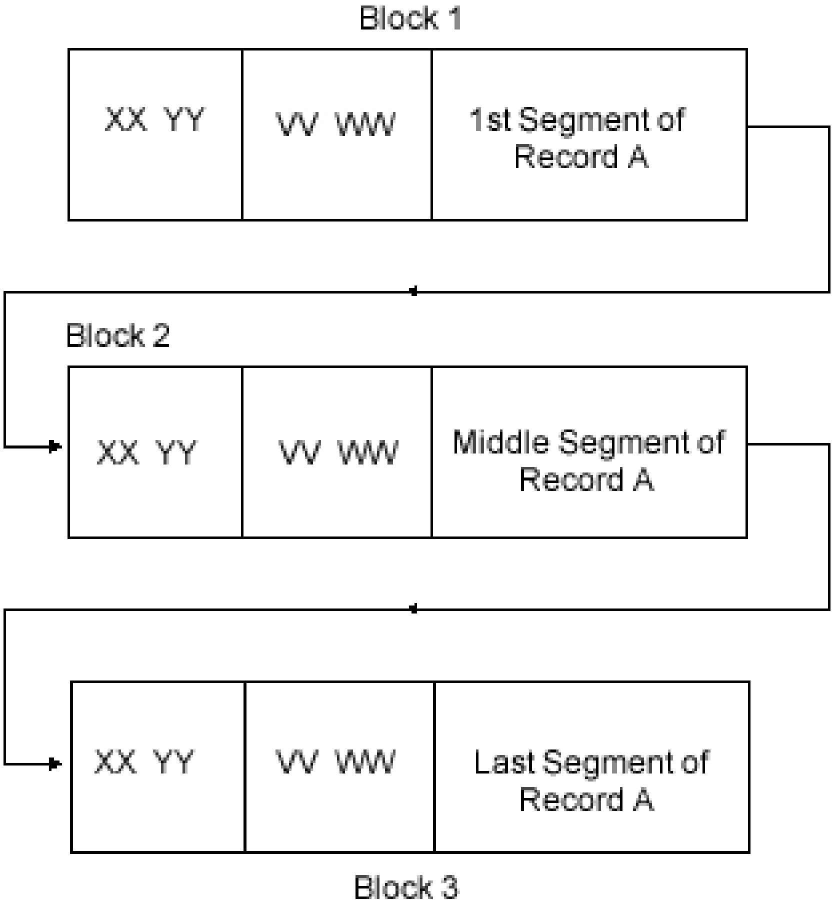


*Figure 18. Variable-length, deblocked, spanned (*VS)*

XX and YY make up the BDW for each data block. XX is the length of all records in each block plus the length of the BDW (4 bytes). YY are currently reserved fields and must be 00. XX should be the actual length of the data block that is written.

Note that logical record A spans over three actual data blocks on the tape.

By breaking down each piece of record A, you can see that each segment has its own characteristics.

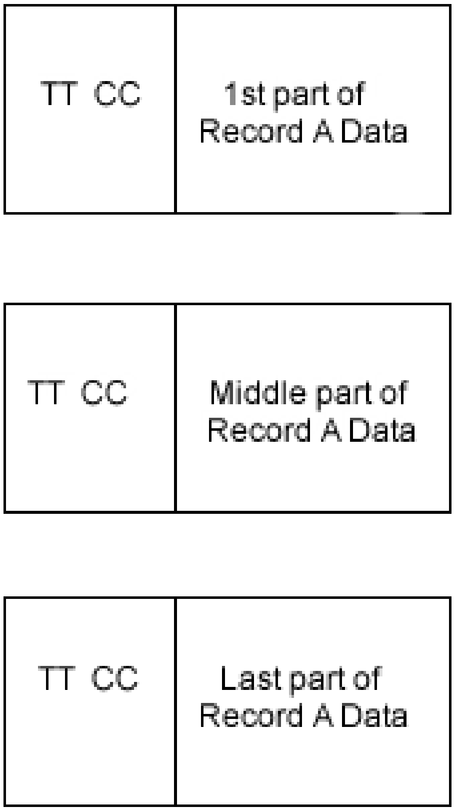

TT CC
1st part of Record A Data
TT CC
Middle part of Record A Data
TT CC
Last part of Record A Data

Figure 19. Segments of record A

Note that the actual mapping of the entire block 3 is the following:

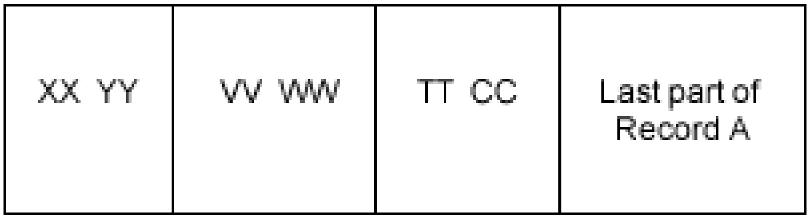


Figure 20. Mapping of block 3

Each segment of record A has its own by mapping by including the SDW. TT and CC make up the SDW. TT is the length of the record plus the length of the SDW (4 bytes). CC is the segment control character. The first byte of the CC defines which part of the record the segment is. The values of the control character can be:

- 00 binary - Complete logical record
- 01 binary - First segment of a multi-segment record

- 10 binary - Last segment of a multi-segment record
- 11 binary - Middle segment of a multi-segment record

The second byte of the control character is reserved, and should be 0. TT should be the actual length of record data segment plus the length of the SDW (4 bytes).

From a user's point of view, a logical view of record A is the same as other records defined above.

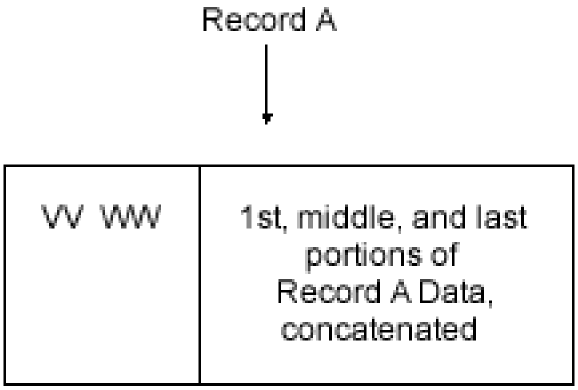

Record A
VV WW
1st, middle, and last portions of Record A Data, concatenated

Figure 21. Logical view of record A

Example - Record format *VBS

Variable-length, blocked, spanned (*VBS).

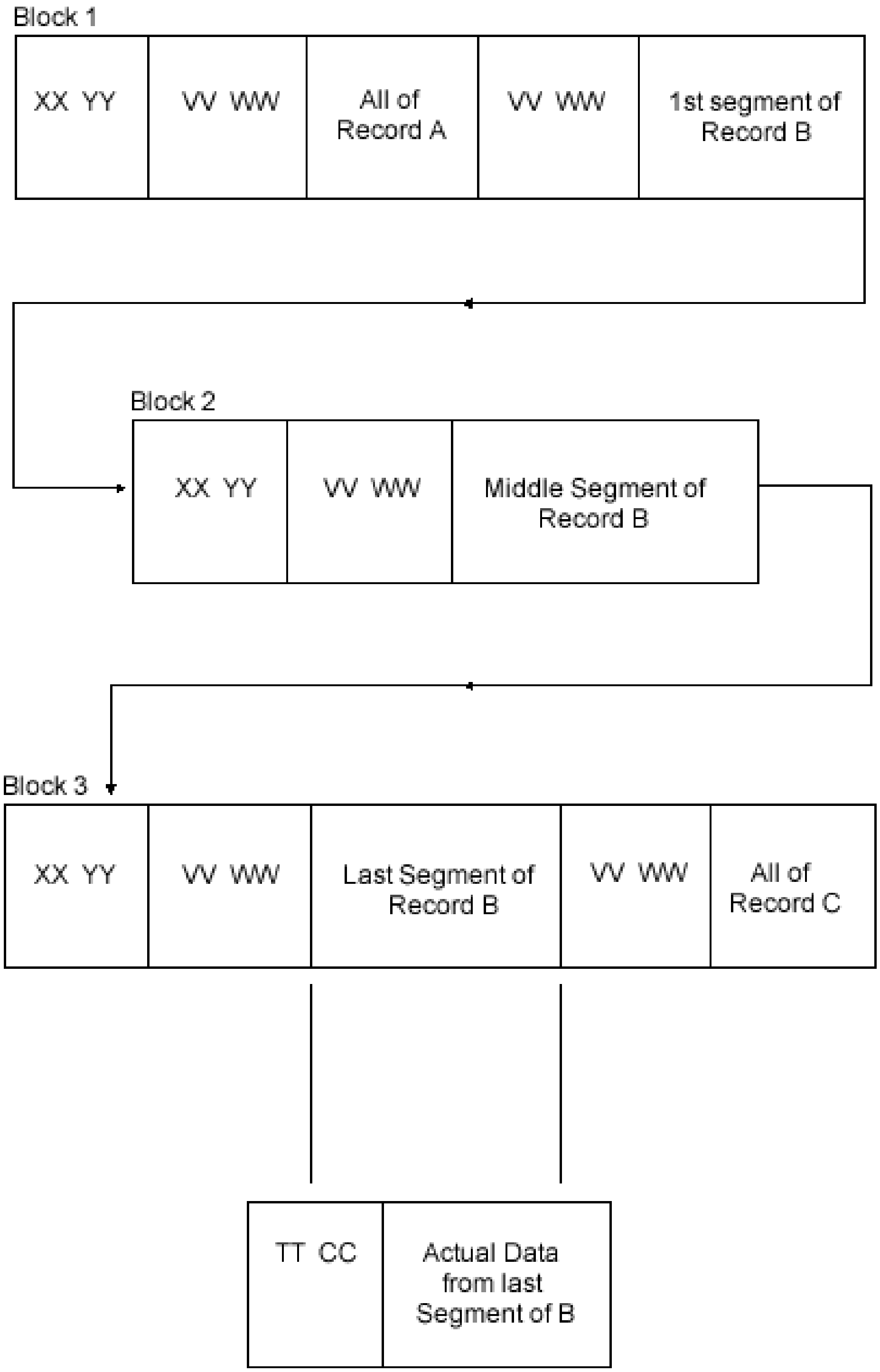


*Figure 22. Variable-length, blocked, spanned (*VBS)*

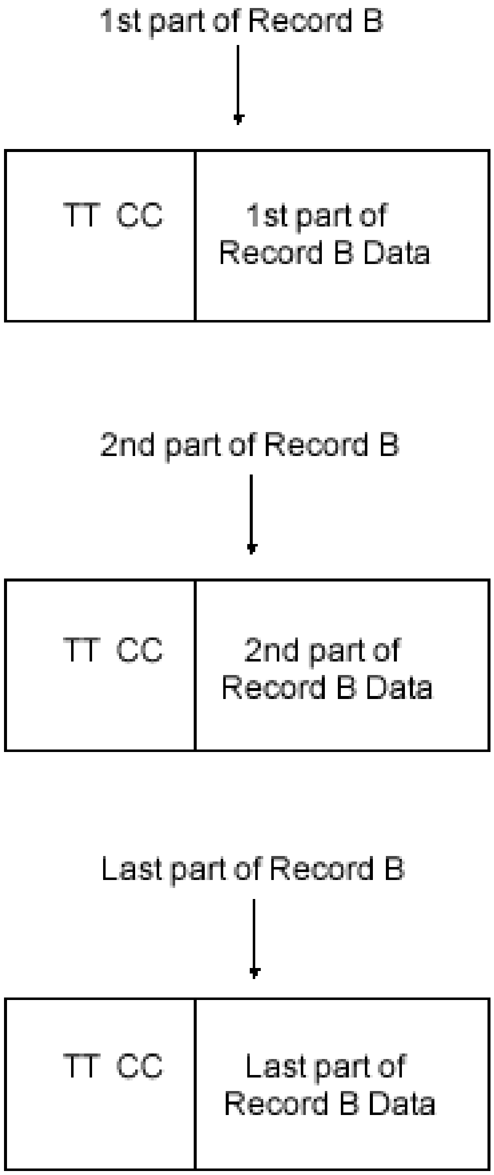

1st part of Record B
TT CC
1st part of Record B Data
2nd part of Record B
TT CC
2nd part of Record B Data
Last part of Record B
TT CC
Last part of Record B Data

Figure 23. Parts of record B

XX and YY make up the BDW for each data block. XX is the length of all records in each block plus the length of the BDW (4 bytes). YY are currently reserved fields and must be 00. XX should be the actual length of the data block that is written.

Note that logical record B spans across three actual data blocks on the tape.

Each segment of record B has its own by mapping by including the SDW. TT and CC make up the SDW. TT is the length of the record plus the length of the SDW (4 bytes). CC is the segment control character. The first byte of the CC defines which part of the record the segment is. The values of the control character can be:

- 00 binary - Complete logical record
- 01 binary - First segment of a multi-segment record

- 10 binary - Last segment of a multi-segment record
- 11 binary - Middle segment of a multi-segment record

The second byte of the control character is reserved, and should be 0. TT should be the actual length of record data segment plus the length of the SDW (4 bytes).

From a user's point of view, a logical view of record B is the the same as other records defined above.

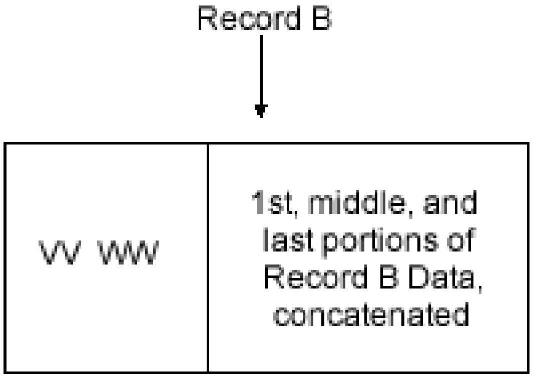

Record B
VV WW
1st, middle, and last portions of Record B Data, concatenated

Figure 24. Logical view of record B

Example - Record format *U

Undefined format (variable length) (*U).

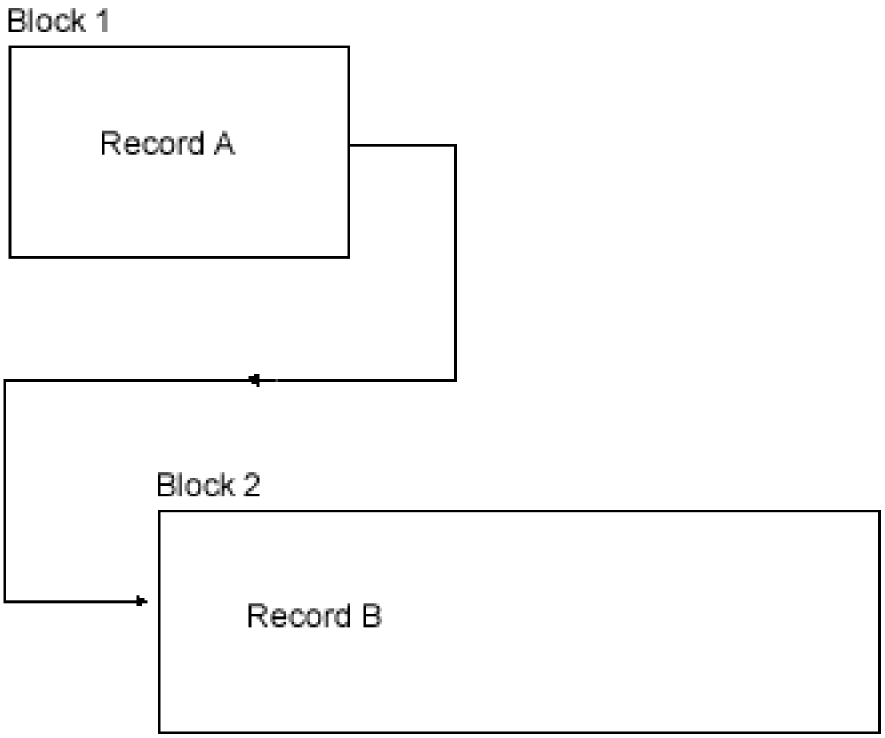


*Figure 25. Undefined format (variable length) (*U)*

Undefined records do not have a standard definition, and the contents of Records A and B are defined by the application. The application may define its own descriptors with lengths similar to BDWs.

Reference

Select any of the reference topics on the navigation bar or from the list below for quick access to specific reference information.

- "Tape file CL commands"
- "Feedback Area Layouts" on page 41

Tape file CL commands

The following commands are available to help maintain and use tapes:

- "Tape Configuration Description Commands"
- "Tape Device Files Commands" on page 41

The CL topic contains detailed descriptions of these commands.

Tape Configuration Description Commands

CHGCTLTAP
Change Controller Description (Tape): The command changes the controller description for a tape controller.

CHGDEVTAP
Change Device Description (Tape): The command changes the device description for a tape device.

CFGDEVMLB
Configure Device Media Library (Tape): The command configures the connection between the media library device and the communication interfaces of the robotics.

CHGDEVMLB
Change Device Media Library (Tape): The command changes the device description for a media library device.

CRTCTLTAP
Create Controller Description (Tape): The command creates a controller description for a tape controller.

CRTDEVTAP
Create Device Description (Tape): The command creates a device description for a tape device.

CRTDEVMLB
Create Device Media Library (Tape): The command creates a device description for a media library device.

DLTCTLD
Delete Controller Description: The command deletes a controller description.

DLTDEVD
Delete Device Description: The command deletes a device description.

DSPCTLD
Display Controller Description: The command displays a controller description.

DSPDEVD
Display Device Description: The command displays a device description.

DSPLANMLB
Display LAN Media Library (Tape): The command displays the LAN information necessary to configure a library manager.

Tape Device Files Commands

CHGTAPF
Change Tape File: The command changes certain attributes of a tape device file.

CRTTAPF
Create Tape File: The command creates a tape device file that is used to read and write records on tape.

DLTF Delete File: The command deletes files.

DSPFD
Display File Description: The command displays the current characteristics of a file.

OVRTAPF
Override with Tape File: The command temporarily changes a tape file or tape file attributes that are specified in a program.

Feedback Area Layouts

This appendix contains Product-Sensitive Programming Interface and Associated Guidance Information.

Figures in this appendix describe the open and I/O feedback areas associated with tape or diskette files. The program presents the following information for each item in these feedback areas:

- Offset, which is the number of bytes from the start of the feedback area to the location of each item

- Data Type
- The program gives the length in number of bytes
- Contents, which is the description of the item and the valid values for it
- File type, which is an indication of what file types are valid for each item.

For more information about various feedback areas, select any of the following:
- "Open Feedback Area"
- "I/O Feedback Area" on page 57

For more information about feedback area layouts for files other than tape and diskette files, refer to the File Management topic in the Information Center.

The support provided by the high-level language you are using determines how to access this information and how the program represents the data types. See your high-level language manual for more information.

Open Feedback Area

The open feedback area is the part of the open data path (ODP) that contains general information about the file after the program opens the file. It also contains file-specific information, depending on the file type. The open feedback area contains information about each device or communications session defined for the file. The program sets this information during open processing and the program may update the information as it performs other operations.

The following links provide access to tables and more information about open feedback areas.
- Open Feedback Area Table
- "Device Definition List" on page 45
- "Volume Label Fields" on page 46
- "IBM Standard Volume Label (VOL1)" on page 47
- "IBM Standard Data Set Label 1 (HDR1/EOV1/EOF1)" on page 48
- "IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2)" on page 49
- "IBM Standard User Labels (UHL1-UHL8 or UTL1-UTL8)" on page 52
- "Other IBM Standard Labels" on page 52
- "ISO/ANSI Standard Volume Label (VOL1)" on page 52
- "ISO/ANSI Standard Data Set Label 1 (HDR1/EOV1/EOF1)" on page 53
- "ISO/ANSI Standard Data Set Label 2 (HDR2/EOV2/EOF2)" on page 55
- "ISO/ANSI Standard User Labels (UHL and UTL)" on page 57

Table 4. Open Feedback Area

Offset	Data Type	Length	Contents	File Type
0	Character	2	Open data path (ODP) type: **DS** Display, tape, ICF, save, printer or diskette file that is not being spooled. **DB** Database member. **SP** Printer or diskette file being spooled or inline data file.	Tape and diskette
2	Character	10	Name of the file being opened. If the ODP type is DS, this is the name of the device file or save file. If the ODP type is SP, this is the name of the device file or the inline data file.	Tape and diskette
12	Character	10	Name of the library containing the file. For an inline data file, the value is *N.	Tape and diskette

Table 4. Open Feedback Area (continued)

Offset	Data Type	Length	Contents	File Type
22	Character	10	Name of the spooled file. The name of a database file containing the spooled input or output records.	Diskette being spooled
32	Character	10	Name of the library in which the spooled file is located.	Diskette being spooled
42	Binary	2	Spooled file number.	Diskette being spooled
44	Binary	2	Maximum record length.	Tape and diskette
46	Character	2	Reserved.	
48	Character	10	Member name. If ODP type SP, the member name in the file named at offset 22.	Diskette
58	Binary	4	Reserved.	
62	Binary	4	Reserved.	
66	Binary	2	File type: **1** Display **2** Printer **4** Diskette **5** Tape **9** Save **10** DDM **11** ICF **20** Inline data **21** Database	Tape and diskette
68	Character	3	Reserved.	
71	Binary	2	Not applicable to tape and diskette.	
73	Binary	2	Not applicable to tape and diskette.	
75	Binary	4	Not applicable to tape and diskette.	
79	Character	2	Not applicable to tape and diskette.	
81	Character	1	Not applicable to tape and diskette.	
82	Character	1	Source file indication. **Y** File is a source file. **N** File is not a source file.	Tape and diskette
83	Character	10	Reserved.	
93	Character	10	Reserved.	
103	Binary	2	Offset to volume label fields of open feedback area.	Tape and diskette
105	Binary	2	Maximum number of records that can be read or written in a block when using blocked record I/O.	Tape and diskette
107	Binary	2	Not applicable to tape and diskette.	
109	Binary	2	Blocked record I/O record increment. Number of bytes that must be added to the start of each record in a block to address the next record in the block.	Tape and diskette
111	Binary	4	Reserved.	
115	Character	1	Miscellaneous flags. **Bit 1:** Reserved.	

Table 4. Open Feedback Area (continued)

Offset	Data Type	Length	Contents	File Type
			Bit 2: File shareable **0** File was not opened shareable. **1** File was opened shareable (SHARE(*YES)).	Tape and diskette
			Bit 3: Not applicable to tape and diskette.	
			Bit 4: Not applicable to tape and diskette.	
			Bit 5: Not applicable to tape and diskette.	
			Bit 6: Field-level descriptions This is always 0 for tape and diskette.	Tape and diskette
			Bit 7: DBCS-capable file **0** File is not DBCS-capable. **1** File is DBCS-capable.	Tape and diskette
			Bit 8: Not applicable to tape and diskette.	
116	Character	10	Not applicable to tape and diskette.	
126	Binary	2	File open count. If the file has not been opened shareable, this field contains a 1. If the file has been opened shareable, this field contains the number of programs currently attached to this file.	Tape and diskette
128	Binary	2	Reserved.	
130	Binary	2	Not applicable to tape and diskette.	
132	Character	1	Miscellaneous flags.	
			Bit 1: Not applicable to tape and diskette	
			Bit 2: Not applicable to tape and diskette.	
			Bit 3: Not applicable to tape and diskette.	
			Bit 4: Not applicable to tape and diskette.	
			Bit 5: Not applicable to tape and diskette.	
			Bit 6: User buffers **0** System creates I/O buffers for the program. **1** User program supplies I/O buffers.	Tape and diskette
			Bits 7: Reserved.	
			Bits 8: Not applicable to tape and diskette.	
133	Character	2	Open identifier. The value is unique for a full open of a file (SHARE(*NO)) or the first open of a file with (SHARE(*YES)). This is used for display and ICF files, but is set up for all file types. It allows you to match this file to an entry on the associated data queue.	Tape and diskette

Table 4. Open Feedback Area (continued)

Offset	Data Type	Length	Contents	File Type
135	Binary	2	Maximum record format length, including both data and file-specific information such as: first-character forms control, option indicators, response indicators, source sequence numbers, and program-to-system data. If the value is zero, then use the field at offset 44.	Tape and diskette
137	Character	2	Not applicable to tape and diskette.	
139	Character	1	Not applicable to tape and diskette.	
140	Character	6	Reserved.	
146	Binary	2	Number of devices defined for this ODP. For displays, this is determined by the number of devices defined on the DEV parameter of the Create Display File (CRTDSPF) command. For ICF, this is determined by the number of program devices defined or acquired with the Add ICF Device Entry (ADDICFDEVE) or the Override ICF Device Entry (OVRICFDEVE) command. For all other files, it has the value of 1.	Tape and diskette
148	Character		Device name definition list. See "Device Definition List" for a description of this array.	Tape and diskette

Device Definition List: The device definition list part of the open feedback area is an array structure. Each entry in the array contains information about each device or communications session that you attach to the file. The number at offset 146 of the open feedback area shows the number of entries in this array. The device definition list begins at offset 148 of the open feedback area. The offsets shown for it are from the start of the device definition list rather than the start of the open feedback area.

Table 5. Device Definition List

Offset	Data Type	Length	Contents	File Type
0	Character	10	Program device name. For database files, the value is DATABASE. For printer or diskette files being spooled, the value is *N. For save files, the value is *NONE. For ICF files, the value is the name of the program device from the ADDICFDEVE or OVRICFDEVE command. For all other files, the value is the name of the device description.	Tape and diskette
10	Character	50	Reserved.	
60	Character	10	Device description name. For printer or diskette files being spooled, the value is *N. For save files, the value is *NONE. For all other files, the value is the name of the device description.	Tape and diskette
70	Character	1	Device class. **hex 01** Display **hex 02** Printer **hex 04** Diskette **hex 05** Tape **hex 09** Save **hex 0B** ICF	Tape and diskette
71	Character	1	Device type. hex 08 Spooled hex 1A 9347 Tape Unit hex 1B 9348 Tape Unit hex 1C 9331-1 Diskette Unit	

Table 5. Device Definition List (continued)

Offset	Data Type	Length	Contents	File Type
			hex 1D 9331-2 Diskette Unit	
			hex 2A 6346 Tape Unit	
			hex 2B 2440 Tape Unit	
			hex 2C 9346 Tape Unit	
			hex 2D 6331 Diskette Unit	
			hex 2E 6332 Diskette Unit	
			hex 3A 3430 Tape Unit	
			hex 3B 3422 Tape Unit	
			hex 3C 3480 Tape Unit	
			hex 3D 3490 Tape Unit	
			hex 49 6367 Tape Unit	
			hex 4A 6347 Tape Unit	
			hex 4E 6341 Tape Unit	
			hex 4F 6342 Tape Unit	
			hex 50 6133 Diskette Unit	
			hex 53 6366 Tape Unit	
			hex 54 7208 Tape Unit	
			hex 5A 6343 Tape Unit	
			hex 5B 6348 Tape Unit	
			hex 5C 6368 Tape Unit	
			hex 64 6344 Tape Unit	
			hex 65 6349 Tape Unit	
			hex 66 6369 Tape Unit	
			hex 67 6380 Tape Unit	
			hex 68 6378 Tape Unit	
			hex 69 6390 Tape Unit	
			hex 70 6379 Tape Unit	
			hex 71 9331-11 Tape Unit	
			hex 72 9331-12 Tape Unit	
			hex 73 3570 Tape Unit	
			hex 74 3590 Tape Unit	
			hex 75 6335 Tape Unit	
			hex 76 1/4-inch Cartridge Tape[1]	
			hex 77 1/2-inch Cartridge Tape[1]	
			hex 78 1/2-inch Reel Tape[1]	
			hex 79 8mm Cartridge Tape[1]	
72	Binary	2	Not applicable to tape and diskette.	
74	Binary	2	Not applicable to tape and diskette.	
76	Character	2	Not applicable to tape and diskette.	
78	Character	1	Not applicable to tape and diskette.	
79	Character	1	Not applicable to tape and diskette.	
80	Character	50	Reserved.	

Note:

1. If the device is not listed above, use one of the following general categories:
   - hex 76—1/4″ Cartridge Tape
   - hex 77—1/2″ Cartridge Tape
   - hex 78—1/2″ Reel Tape
   - hex 79—8mm Cartridge Tape

Volume Label Fields:

Table 6. Volume Label Fields

Offset	Data Type	Length	Contents	File Type
0	Character	128	Volume label of current volume.	Tape and diskette
128	Character	128	Header label 1 of the opened file.	Tape and diskette
256	Character	128	Header label 2 of the opened file.	Tape

IBM Standard Volume Label (VOL1):

Table 7. Format of the IBM Standard Volume Label (VOL1) for Tape

Offset	Data Type	Length	Contents	File Type
0	Character	3	Label identifier	
3	Character	1	Label number	
4	Character	6	Volume identifier (Volume Serial Number)	
10	Character	1	Volume access (security)	
11	Character	5	VTOC Pointer (not used)	
16	Character	21	Reserved	
37	Character	14	Owner Name and Address Code (Owner identifier)	
51	Character	29	Reserved	

The IBM® standard volume label (VOL1) is 80 characters in length. The system uses it to identify the tape volume, tape volume owner, and security of the tape volume’s contents. It is always the first block of data on the tape volume if the tape is a standard labeled tape. The program records the volume label in EBCDIC.

The contents and function of each of the fields is described below.

- Label identifier

  The characters VOL identify this label as a volume label. The system reads this field to verify that it mounts a standard labeled tape. The system also verifies that this label is a volume label. The system writes this field to the tape when you use the Initialize Tape (INZTAP) command and specify the new volume (NEWVOL) parameter.
- Label number

  The relative position of the label within a set of labels of the same type. The label number is always 1 for the IBM standard volume label.
- Volume identifier (Volume Serial Number)

  A unique identification code to identify the logical tape volume. The system writes the value that is specified for the new volume (NEWVOL) parameter when you use the Initialize Tape (INZTAP) command. For media library devices, the program recommends that you match the logical volume identifier with the external bar code identifier on the cartridges. The value may range from 1 to 6 alphanumeric characters (left justified and padded with blanks if less than 6). The alphanumeric character set includes A-Z, 0-9, @, $, and #. If a program specifies a value for a command in the VOL parameter, the system verifies this field to match the specified value.
- Volume access (security)

  The program considers the volume secure (from processing) if the volume security field is a blank, character zero, or hex zero. *SECOFR authority can process the secured volumes.
- VTOC Pointer (not used)

  Not used by OS/400.
- Owner Name and Address Code (Owner identifier)

The owner identifier of the tape volume. The system writes a value to this field when you use the OWNER parameter on the Initialize Tape (INZTAP) command. The purpose of the field is to write the owner identifier of the volume or to write information about the contents of the volume.

IBM Standard Data Set Label 1 (HDR1/EOV1/EOF1):

Table 8. Format of the IBM Standard Data Set Label 1 (HDR1/EOV1/EOF1)

Offset	Data Type	Length	Contents	File Type
0	Character	3	Label identifier	
3	Character	1	Label number	
4	Character	17	Data Set (File) Identifier	
21	Character	6	Aggregate volume identifier	
27	Character	4	Aggregate volume sequence number	
31	Character	4	Data Set (File) sequence number	
35	Character	4	Generation Number (not used)	
39	Character	2	Generation Version Number (not used)	
41	Character	6	Creation Date	
47	Character	6	Expiration Date	
53	Character	1	Data Set (File) Security (not used)	
54	Character	6	Block Count, Low Order (Trailer labels only)	
60	Character	13	System Code (Trailer labels only)	
73	Character	3	Reserved	
76	Character	4	Block Count, High Order (Trailer labels only)	

The IBM standard data set label 1 (HDR1/EOV1/TRL1) is 80 characters in length, and you use it

To identify each data set. The program records the data set label in EBCDIC.

The program describes the contents and function of each of the fields below.

- Label identifier

  The characters identify the type of data set label.

  – HDR - Header Label (the beginning of a data set)

  – EOF - Trailer Label (the end of a data set)

  – EOV - Trailer Label (the end of a data set that is continued on another volume)

- Label number

  The relative position of this label, within a set of labels of the same type; it is always 1 for the IBM data set label 1.

- Data Set Identifier (file name)

  A unique identification code to identify the data set (file). If the data set ID is less than 17 characters, it is left justified, and you pad it with blanks.

- Aggregate volume identifier

  This field contains the volume identifier from the volume labels of the first volume in a multivolume data set.

- Aggregate volume sequence number

  This field contains the relative volume number of this volume in a multivolume data set. The field is 0001 for a single volume data set.

- Data Set (File) sequence number

  This field indicates the relative position of the data set within a multiple data set group. The value will be in EBCDIC displayable characters for 0001–9999. For numbers larger than 9999 that will not fit in the 4–character field as an EBCDIC displayable character set, the first byte will a ’?’ (’6F’x) for EBCDIC labels. The last three bytes will be a binary number from 1 to 64000.

- Generation Number that is not used.

  If the data set is a part of a generation data group, this field contains a number that indicates absolute generation number. The OS/400 does not use this field.
- Generation Version Number (not used)

  If the data set is part of a generation data group, this field contains a number that indicates the version number of the generation. OS/400 does not use this field.
- Creation Date

  The creation date of the data set. The program shows the date in the format cyyddd, where:

  c = century (blank=19; 0=20; 1=21; and so on)

  yy = year (00-99)

  ddd = day of the current year (001-366)

  Note that the century code gives the first two digits of the year, not the actual century. For example, a blank which translates into 19 indicates a year in the 1900s, not in the nineteenth century.
- Expiration Date

  The date the program considers the data set expired. Expiration date refers to a data set that is allowed to be overwritten. The user specifies the date in the open tape file that is used in writing the tape. You ignore the expiration date on input, and verify the expiration date on output by OS/400. The date is in the format cyyddd, where:

  c = century (blank=19; 0=20; 1=21; and so on)

  yy = year (00-99)

  ddd = day of the current year (001-366)

  Note that the century code gives the first two digits of the year, not the actual century. For example, a blank which translates into 19 indicates a year in the 1900s, not in the nineteenth century.
- Data Set (File) Security that is not used.

  A code number indicates the security status of the data set. OS/400 does not use this field. The following values indicate data set security values that are created by other systems:

  0 No password protection

  1 Password protection (required for read/write/deletion)

  3 Password protection that is required for write/deletion.
- Block Count, Low Order (Trailer labels only)

  The field in the trailer labels contains the low order six digits of the number of data blocks in the data set on the current volume. The field in the header labels contains hex zeros.
- System Code (Trailer labels only)

  A unique code that identifies the system that created the data set.

  IBMOS400
  IBM OS/400

  IBM OS/VS 370
  IBM MVS™ operating system
- Block Count, High Order (Trailer labels only)

  The field in the trailer labels contains the high order four digits of the number of data blocks in the data set on the current volume. The field in the header labels contains hex zeros.

IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2):

Table 9. Format of the IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2)

Offset	Data Type	Length	Contents
0	Character	3	Label identifier

Table 9. Format of the IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2) (continued)

Offset	Data Type	Length	Contents
3	Character	1	Label number
4	Character	1	Record Format
5	Character	5	Block Length
10	Character	5	Record Length
15	Character	1	Tape Density/Format
16	Character	1	Data Set Position
17	Character	17	Job/Job Step Identification
34	Character	2	Tape Recording Technique
36	Character	1	Control Character
37	Character	1	Reserved
38	Character	1	Block Attribute
39	Character	3	Reserved
42	Character	5	Device Serial Number (not used)
47	Character	1	Checkpoint Data Set Identifier
48	Character	22	Reserved
70	Character	10	Large Block Length

The IBM standard data set label 2 (HDR2/EOV2/TRL2) is 80 characters in length and the program uses it to identify additional information about the data set. The program records the data set label in EBCDIC.

The program describes the contents and function of each of the fields below.

- Label identifier

  The characters identify the type of data set label.

  HDR Header Label (the beginning of a data set)

  EOF Trailer Label (the end of a data set)

  EOV Trailer Label (the end of a data set that is continued on another volume)

- Label number

  The relative positions of this label within the set of labels of the same type; The Label number is always 2 for the IBM data set label 2.

- Record Format

  An alphabetic character that indicates the format of the records in the data set. While the operating system reads from the tape, the Record format field tells the operating system how to interpret the blocks of data the program reads.

  F Fixed length records

  V Variable length records

  U Undefined length records

- Block Length

  A number indicating the block length (in bytes) of the data blocks on the tape. The number in this field can range from 18 to 32767 on OS/400. For numbers greater than 32767, the Large Block Length field allows values up to 512 Kbytes.

- Record Length

  A number that indicates the record length, in bytes, of the logical records on the tape volume. The interpretation of the number depends on the Record Format field.

  F Fixed length records.

  V Variable length records.

U Undefined length records

- Tape Density/Format

  A code indicating the record density/format of the tape volume.

  3 1600 bpi

  4 6250 bpi

  5 3200 bpi

  blank all other densities/formats

- Data Set Position

  A code indicating a volume switch is as follows:

  0 No volume switch has occurred

  1 A volume switch previously occurred

- Job/Job Step Identification

  This field identifies the job and job step that created or extended the data set. OS/400 does not use this field.

- Tape Recording Technique

  This field indicates the tape recording technique used in creating the data set.

  blank No Improved Data Recording Capability (IDRC) used.

  'P ' Improved Data Recording Capability (IDRC) used.

- Control Character

  A printer control code indicating whether the program uses a control character set to create the data set and the type of control characters used:

  A Contains ANSI control characters

  M Contains machine control characters

  blank Contains no control characters

- Block Attribute

  A code indicating the block attribute used to create the data set:

  B Blocked records

  S Spanned records, if the record format byte is V

  S Standard records, if the record format byte is F

  R Blocked and spanned records, if the record format byte is V

  R Blocked and standard records, if the record format byte is F

  blank Records that are not blocked and not spanned, or records that are not blocked and not standard

- Device Serial Number (not used)

  The serial number of the device that writes the volume. Header and trailer labels may have different serial numbers if the program extends the data set. OS/400 does not use this field.

- Checkpoint Data Set Identifier

  This byte contains the character C if the data set is a secure checkpoint data set. The byte is blank if the data set is not a secure data set checkpoint.

- Large Block Length

  A number indicating the block length (in bytes) of the data blocks on the tape. The number in this field can range from 18 to 524288 on OS/400. For numbers up to 32767, the Block Length field also contains the block length in bytes.

IBM Standard User Labels (UHL1-UHL8 or UTL1-UTL8):

Table 10. Format of the IBM Standard User Labels (UHL1-UHL8 or UTL1-UTL8)

Offset	Data Type	Length	Contents
0	Character	3	Label identifier
3	Character	1	Label number
4	Character	76	User data

The IBM standard user labels (UHL1-UHL8 or UTL1-UTL8) is 80 bytes in length. The program uses them to identify additional information about the data set. All information in the user labels is user data. The user labels directly follow the header/trailer group and can be up to 8 user labels per data set.

The program describes the contents and function of each of the fields below.

- Label identifier

  The characters identify the type of data set label.

  UHL User Header Label (the beginning of a data set)

  UTL User Trailer Label (the end of a data set)
- Label number

  The relative position of this label within the set of labels of the same type; there is a limit of 8 user labels per data set.
- User Data

  You contain any user information in the user label. The program ignores the information in this set by OS/400, but passes it to a specified user label program.

Other IBM Standard Labels: OS/400 only supports labels described above. However, it reads and ignores volume labels (VOL2-VOL8) and file labels (HDR3-HDR8) created on other operating systems.

ISO/ANSI Standard Volume Label (VOL1):

Table 11. Format of the ISO/ANSI Standard Volume Label (VOL1), Version 3

Offset	Data Type	Length	Contents	File Type
0	Character	3	Label identifier	
3	Character	1	Label number	
4	Character	6	Volume identifier (Volume Serial Number)	
10	Character	1	Accessibility	
11	Character	26	Reserved	
37	Character	14	Owner Identifier	
51	Character	28	Reserved	
79	Character	1	Label Standard Level	

The ISO/ANSI standard volume label (VOL1) is 80 characters in length. The program uses the label to identify the tape volume, tape volume owner, and security of the tape volume's contents. It is always the first block of data on the tape volume if the tape is a standard labeled tape. The field definition follows the industry standards as understood and interpreted by IBM: (OS/400 supports for input only):

- ANSI X3.27-1978, level 4
- ISO 1001-1979,level 4

The system records the volume label in ASCII.

the program describes the contents and function of each of the fields below. This version of the ISO/ANSI standard is Version 3.

- Label identifier

  The characters VOL identify this label as a volume label. The system reads this field to verify that a standard labeled tape is mounted, and that this label is a volume label. When you use the initialized tape (INZTAP) command and specify the new volume (NEWVOL) parameter, the system writes this field to the tape.
- Label number

  The relative position of this label within the set of labels of the same type. The Label number is always 1 for the Version 3 volume label.
- Volume identifier

  A unique identification code to identify the logical tape volume. The system writes the value specified for the new volume (NEWVOL) parameter on the Initialize Tape (INZTAP) command in this position. For media library devices, the logical volume identifier should match the external bar code identifier on the cartridges. The value may be from 1 to 6 alphanumeric characters (left justified and padded with blanks if less than 6). The alphanumeric character set includes A-Z, 0-9, @, $, and #. If the program specifies a value for a command in the VOL parameter, the system verifies that this field matches the specified value.
- Accessibility

  The system considers the volume secure (from processing) if the volume security field is a blank. A program with *SECOFR authority can process secured volumes.
- Owner Identifier

  The owner identifier of the tape volume. The system writes a value to this field through the OWNER parameter on the Initialize Tape (INZTAP) command. You use the field to write the owner identifier of the volume or to write information about the contents of the volume. If the identifier is less than 14 bytes, the system justifies the value and pads it with blanks.
- Label Standard Level

  Identifies the version of ISO/ANSI standards. For Version 3, the program places a 3 in this field.

ISO/ANSI Standard Data Set Label 1 (HDR1/EOV1/EOF1):

Table 12. Format of the ISO/ANSI Standard Data Set Label 1 (HDR1/EOV1/EOF1)

Offset	Data Type	Length	Contents	File Type
0	Character	3	Label identifier	
3	Character	1	Label number	
4	Character	17	File Identifier	
21	Character	6	File Set Identifier	
27	Character	4	File Section Number	
31	Character	4	File Sequence Number	
35	Character	4	Generation Number (not used)	
39	Character	2	Version Number (not used)	
41	Character	6	Creation Date	
47	Character	6	Expiration Date	
53	Character	1	Accessibility	
54	Character	6	Block Count, Low Order (Trailer labels only)	
60	Character	13	System Code (Trailer labels only)	
73	Character	7	Reserved	

The ISO/ANSI standard data set label 1 (HDR1/EOV1/TRL1) is 80 characters in length and is used to identify each data set. The system records the data set label in ASCII.

The manual describes the contents and function of each of the fields below.

- Label identifier

  The characters identify the type of data set label.

HDR Header Label (the beginning of a data set)

EOF Trailer Label (the end of a data set)

EOV Trailer Label (the end of a data set that is continued on another volume)

- Label number

  The relative position of this label within the set of labels of the same type; it is always 1 for the data set label 1.
- File Identifier

  A unique identification code to identify the data set (file). If the data set ID is less than 17 characters, it is left justified and padded with blanks.
- File Set Identifier

  This field contains the volume identifier from the volume labels of the first volume in a multivolume data set.
- File Section Number

  This field contains the relative volume number of this volume in a multivolume data set. The field is 0001 for a single volume data set.
- File Sequence Number

  This field indicates the relative position of the data set within a multiple data set group. The value will be in EBCDIC displayable characters for 0001–9999. For numbers larger than 9999 that will not fit in the 4–character field as an EBCDIC displayable character set, the first byte will a '?' ('6F'x for EBCDIC labels. The last three bytes will be a binary number from 1 to 64000.
- Generation Number (not used)

  If the data set is a part of a generation data group, this field contains a number that indicates a absolute generation number. OS/400 does not use this field.
- Version Number (not used)

  If the data set is part of a generation data group, this field contains a number indicating the version number of the generation. OS/400does not use this field.
- Creation Date

  The creation date of the data set. the program shows the date in the format cyyddd, where:

  c = century (blank=19; 0=20; 1=21; and so on)

  yy = year (00-99)

  ddd = day of the current year (001-366)

  Note that the century code gives the first two digits of the year, not the actual century. For example, a blank which translates into 19 indicates a year in the 1900s, not in the nineteenth century.
- Expiration Date

  The program considers the date in which the data set expires. Expired refers to a data set being allowed to be overwritten. The user specifies the date in the open tape file used in writing the tape. The OS/400 ignores the expiration date on input, and verifies the expiration date on output. The date is in the format cyyddd, where:

  c = century (blank=19; 0=20; 1=21; and so on)

  yy = year (00-99)

  ddd = day of the current year (001-366)

  Note that the century code gives the first two digits of the year, not the actual century. For example, a blank which translates into 19 indicates a year in the 1900s, not in the nineteenth century.
- Accessibility

  A code number indicating the security status of the data set. OS/400 does not use this field. The following values indicate data set security values created by other systems:

  blank No data set access protection

1 Password protection (Required for read/write/deletion)

3 Password protection (Required for write/deletion)

other character
Protected volume, no access possible

- Block Count (Trailer labels only)

  The field in the trailer labels contains the number of data blocks in the data set on the current volume. The field in the header labels contains hex zeros.

- System Code (Trailer labels only)

  A unique code that identifies the system that created the data set.

  IBMOS400
  IBM OS/400

  IBM OS/VS 370
  IBM MVS operating system

ISO/ANSI Standard Data Set Label 2 (HDR2/EOV2/EOF2):

Table 13. Format of the IBM Standard Data Set Label 2 (HDR2/EOV2/EOF2)

Offset	Data Type	Length	Contents
0	Character	3	Label identifier
3	Character	1	Label number
4	Character	1	Record Format
5	Character	5	Block Length
10	Character	5	Record Length
15	Character	35	Reserved for Operating System
50	Character	2	Buffer offset
52	Character	28	Reserved

The ISO/ANSI standard data set label 2 (HDR2/EOV2/TRL2) is 80 characters in length and is used to identify additional information about the data set. The system records the data set label in ASCII.

The manual describes the contents and function of each of the fields below.

- Label identifier

  The characters identify the type of data set label.

  HDR Header Label (the beginning of a data set)

  EOF Trailer Label (the end of a data set)

  EOV Trailer Label (the end of a data set that is continued on another volume)

- Label number

  The relative position of this label within the set of labels of the same type; it is always 2 for the data set label 2.

- Record Format

  An alphabetic character that indicates the format of the records in the data set. The record format indicates to the operating system how to interpret the blocks of data that the program reads from the tape volume.

  F Fixed length records

  V Variable length records

  U Undefined length records

- Block Length

A number indicating the block length, in bytes, of the data blocks on the tape. The number in this field can range from 18 to 2048. This block length should include the buffer offset and padding.

Note that the 18-byte to 2048 byte limit on block length is an ISCII/ASCII standard. You may specify larger blocks (up to 9999 bytes) with the agreement of the interchange parties. However, for tapes with Version 3 labels, exceeding the 2048 byte limit may create incompatible tapes.

Interpretation of the number depends on the associated record format field as follows:

Record format F
Maximum block length

Record format D
Maximum block length that includes the 4 byte length indicator in the records and the optional block prefix.

Record format S
Maximum block length that includes the optional block prefix, plus one or more pairs of 5 byte segment control words and segments.

- Record Length

  A number indicating the record length, in bytes, of the logical records on the tape volume. The interpretation of the number depends on the Record Format field.

  – Record format F

    Actual record length

  – Record format D

    Maximum record length that includes the 4 byte length indicator in the records

  – Record format S

    Maximum record length that excludes all the 5 byte segment control words that describe the record. If the record length is larger than 99999, this field is 0.

- Reserved for Operating System

  The content of this field is optional for each operating system. OS/400 has chosen similar meaning to the same bytes in the IBM Standard Data Set Label 2.

  – Tape Density (1 byte)

    3 1600 bpi

    4 3200 bpi

    5 6250 bpi

    blank All other densities/formats

  – Data Set Position (1 byte)

    A code indicating a volume switch is as follows:

    0 No volume switch has occurred

    1 A volume switch previously occurred

  – Job/Job Step Identification (17 bytes)

    This field identifies the job and job step that created or extended the data set. OS/400 does not use this field.

  – Tape Recording Technique (2 bytes)

    This field indicates the tape recording technique used in creating the data set.

    blank No Improved Data Recording Capability (IDRC) used.

    'P ' Improved Data Recording Capability (IDRC) used.

  – Control Character (1 byte)

A printer control code indicating whether the program uses a control character set to create the data set and the type of control characters the program uses:

A Contains ISO/ANSI control letters

blank Contains no control characters

- Buffer Alignment Block (1 byte)

  OS/400 does not use this field.

- Block Attribute

  A code indicating the block attribute used to create the data set:

  B Blocked records

  blank Records not blocked

- Reserved (11 bytes)

  OS/400 does not use this field.

- Buffer Offset

  The length of the block prefix (from 0 to 99). Used to determine the length of an optional prefix that may be a part of a physical block on tape. The version of the prefix for variable and spanned record formats is known as a block descriptor word (BDW). A BDW is always 4 bytes long and contains the block length of the physical record it describes, including the BDW.

ISO/ANSI Standard User Labels (UHL and UTL):

Table 14. Format of the ISO/ANSI Standard User Labels (UHL and UTL)

Offset	Data Type	Length	Contents
0	Character	3	Label identifier
3	Character	1	Label number
4	Character	76	User data

The ISO/ANSI standard user labels (UHL and UTL) is 80 bytes in length and is used to identify additional information about the data set. All information in the user labels is user data. The user labels directly follow the header/trailer group and can be up to any number of user labels per data set.

The program describes the contents and function of each of the fields below.

- Label identifier

  The characters identify the type of data set label.

  UHL User Header Label (the beginning of a data set)

  UTL User Trailer Label (the end of a data set)

- Label number

  The relative position of this label within the set of labels of the same type; there is a no limit of user labels per data set.

- User Data

  This field specifies any user information in the user label. The system ignores the information in this set but passes it to a specified user label program.

Other ISO/ANSI Labels: ISO/ANSI standards also allow up to 9 volume labels, up to 9 header and trailer labels, but OS/400 will not create these ″extra″ labels. OS/400 will read and ignore these labels if created on another operating system.

I/O Feedback Area

The system communicates the results of I/O operations to the program using OS/400 messages and I/O feedback information. The system updates the I/O feedback area for every I/O operation unless your program uses a blocked record I/O. In that case, the system updates the feedback area only when you read or write a block of records. Some of the information reflects the last record in the block. Other information, such as the count of I/O operations, reflects the number of operations on blocks of records and not the number of records. See your high-level language manual to determine if your program uses blocked record I/O.

The I/O feedback area consists of two parts: "Common I/O Feedback Area" and a file-dependent area. The file-dependent area varies by the file type.

Common I/O Feedback Area:

Table 15. Common I/O Feedback Area

Offset	Data Type	Length	Contents
0	Binary	2	Offset to file-dependent feedback area.
2	Binary	4	Write operation count. Updated only when a write operation completes successfully. For blocked record I/O operations, this count is the number of blocks, not the number of records.
6	Binary	4	Read operation count. Updated only when a read operation completes successfully. For blocked record I/O operations, this count is the number of blocks, not the number of records.
10	Binary	4	Write-read operation count. Updated only when a write-read operation completes successfully.
14	Binary	4	Other operation count. Number of successful operations other than write, read, or write-read. Updated only when the operation completes successfully. This count includes update, delete, force-end-of-data, force-end-of-volume, change-end-of-data, release record lock, and acquire/release device operations.
18	Character	1	Reserved.
19	Character	1	Current operation. **hex 01** Read or read block or read from invited devices **hex 02** Read direct **hex 03** Read by key **hex 05** Write or write block **hex 06** Write-read **hex 07** Update **hex 08** Delete **hex 09** Force-end-of-data **hex 0A** Force-end-of-volume **hex 0D** Release record lock **hex 0E** Change end-of-data **hex 0F** Put delete **hex 11** Release device **hex 12** Acquire device
20	Character	10	Not applicable to tape and diskette.
30	Character	2	Device class: Byte 1: hex 00 Database hex 01 Display hex 02 Printer hex 04 Diskette

Table 15. Common I/O Feedback Area (continued)

Offset	Data Type	Length	Contents	
			hex 05	Tape
			hex 09	Save
			hex 0B	ICF
			Byte 2 (if byte 1 contains hex 00):	
			hex 00	Nonkeyed file
			hex 01	Keyed file
			Byte 2 (if byte 1 does not contain hex 00 and contains either hex 04 or hex 05):	
			hex 08	Spooled
			hex 1A	9347 Tape Unit
			hex 1B	9348 Tape Unit
			hex 1C	9331-1 Diskette Unit
			hex 1D	9331-2 Diskette Unit
			hex 2A	6346 Tape Unit
			hex 2B	2440 Tape Unit
			hex 2C	9346 Tape Unit
			hex 2D	6331 Diskette Unit
			hex 2E	6332 Diskette Unit
			hex 3A	3430 Tape Unit
			hex 3B	3422 Tape Unit
			hex 3C	3480 Tape Unit
			hex 3D	3490 Tape Unit
			hex 49	6367 Tape Unit
			hex 4A	6347 Tape Unit
			hex 4E	6341 Tape Unit
			hex 4F	6342 Tape Unit
			hex 50	6133 Diskette Unit
			hex 53	6366 Tape Unit
			hex 54	7208 Tape Unit
			hex 5A	6343 Tape Unit
			hex 5B	6348 Tape Unit
			hex 5C	6368 Tape Unit
			hex 64	6344 Tape Unit
			hex 65	6349 Tape Unit
			hex 66	6369 Tape Unit
			hex 67	6380 Tape Unit
			hex 68	6378 Tape Unit
			hex 69	6390 Tape Unit
			hex 70	6379 Tape Unit
			hex 71	9331-11 Diskette Unit
			hex 72	9331-12 Diskette Unit
			hex 73	3570 Tape Unit
			hex 74	3590 Tape Unit
			hex 75	6335 Tape Unit
			hex 76	1/4-inch Cartridge Tape[1]
			hex 77	1/2-inch Cartridge Taped[1]
			hex 78	1/2-inch Reel Tape[1]
			hex 79	8mm Cartridge Tape[1]

Table 15. Common I/O Feedback Area (continued)

Offset	Data Type	Length	Contents
32	Character	10	Device name. The name of the device for which the operation just completed. Supplied only for display, printer, tape, diskette, and ICF files. For printer or diskette files being spooled, the value is *N. For ICF files, the value is the program device name. For other files, the value is the device description name.
42	Binary	4	Length of the record processed by the last I/O operation (supplied only for an ICF, display, tape, or database file). On ICF write operations, this is the record length of the data. On ICF read operations, it is the record length of the record associated with the last read operation.
46	Character	80	Reserved.
126	Binary	2	Number of records retrieved on a read request for blocked records or sent on a write or force-end-of-data or force-end-of-volume request for blocked records. Supplied only for database, diskette, and tape files.
128	Binary	2	Record length. For output, the field value is the record format length, including first-character forms control, option indicators, source sequence numbers, and program-to-system data. If the value is zero, use the field at offset 42. For input, the field value is the record format length, including response indicators and source sequence numbers. If the value is zero, use the field at offset 42.
130	Character	2	Reserved.
132	Binary	4	Current block count. The number of blocks of the tape data file already written or read. For tape files only.
136	Character	8	Reserved.

Note:

1. If the device is not listed above, use one of the following general categories:
   - hex 76—1/4″ Cartridge Tape
   - hex 77—1/2″ Cartridge Tape
   - hex 78—1/2″ Reel Tape
   - hex 79—8mm Cartridge Tape

Troubleshoot tape files

This page is designed to help you solve problems related to tape files that you may be experiencing. See “Handling Tape Processing Errors and Damaged Tapes” for more information regarding tape errors.

Handling Tape Processing Errors and Damaged Tapes

When the system does not write tape marks or labels at the end of the data file, damage to the data files can occur. This can happen due to an error condition or a system failure. If this happens when writing a file to a 1/2-inch tape device, the following occurs when you try to read the data file:

- The new data and existing data may appear concatenated when processed for input. When the tape has labels the system sends an error message to the system operator when the program reads the trailer labels. The system does not detect an error for unlabeled tape.
- If the new data and the existing data do not appear concatenated, then the system sends an error message to the system operator.
- If the tape you use as input meets one of the following conditions, then the tape advances off the end of the reel:

- The tape does not contain existing data or tape marks beyond the location of the last data block.
- The tape is new.
- The tape is completely erased.

Note: Whenever you close an output file:

- The system attempts to write an end-of-tape marker and label at the end of the file.
- If the system cannot write closing tape marks and labels, the system sends a message to your job log.

When an error occurs when you save data by using a SAVxxx command, the system prompts the operator to load another tape or cancel the job. If the operator loads another tape volume:

- The system repositions the tape a number of blocks before the error occurred.
- The system writes end-of-volume labels.

The job then continues to write data starting with the first block of data that was overwritten with end-of-volume labels.

If damage occurs while using a cartridge tape device, and you encounter a blank tape, the program sends an error message to the system operator.

Related information for tape files

Listed below are the iSeries™ manuals and Information Center topics that relate to Tape files.

- "Other Tape Support Commands"
- File Management
- Backup and Recovery
- Manage Tape Libraries

Other Tape Support Commands

ADDTAPCTG
Add Tape Cartridge: The command adds the specified cartridge identifiers to a usable category.

CHGJOBMLBA
Change Job MLB Attributes: The command allows a user to change the media library resource allocation attributes for a job.

CHGTAPCTG
Change Tape Cartridge: The command changes the specified cartridge from any category to the specified category.

CHKTAP
Check Tape: The command searches a tape volume for a specific volume identifier or file label.

CPYFRMTAP
Copy from Tape: The command copies records from a tape file to an output file or a printer.

CPYTOTAP
Copy to Tape: The command copies records to a tape file from a physical, logical, tape, diskette, or spooled inline data file.

CRTTAPCGY
Create Tape Category: The command creates a user-defined category name and assigns it to a system name.

DLTTAPCGY
Delete Tape Category: The command deletes a user-defined category name that was previously

created with the Create Tape Category (CRTTAPCYG) command. The command will not delete the category if a cartridge currently uses that category.

DMPTAP
Dump Tape: The command dumps label information, data blocks, or both, from a tape with or without a label.

DSPTAP
Display Tape: The command displays volume label information,file label information, and saved object information for standard label tapes. The command displays both the volume type and density for volumes without labels.

DSPTAPCGY
Display Tape Category: The command allows the user to display the categories that are defined through the Create Tape Category (CRTTAPCGY) command.

DSPTAPCTG
Display Tape Cartridge: The command displays the attributes of tape cartridges.

DSPTAPSTS
Display Tape Status: The command displays the slot information for the library device and tape device information for the tape devices that are attached to the library device.

DUPTAP
Duplicate Tape: The command copies the contents of one tape to another.

INZTAP
Initialize Tape: The command initializes tapes, with or without labels, or clears all the data on the tape from the load point to the end-of-tape marker.

RMVTAPCTG
Remove Tape Cartridge: The command removes the specified cartridge identifiers from their current category, or the specified category, and places them in the eject (*EJECT) category. The command can move the specified cartridge to the Convenience I/O station or to the High Capacity Output station. The eject category is not a valid category for I/O operations. The Eject category cartridges are disallowed in the tape device.

SETTAPCGY
Set Tape Category: The command allows the user to set a category on a tape device within the specified media library.

WRKMLBRSCQ
The command allows a user to work with the resource allocation requests for the specified media library device.

WRKTAPCTG
Work with Tape Cartridges: The command allows the user to work with a list of tape cartridges.

The following iSeries manuals and Information Center topics contain information you may need. The manuals list their title and base order number.

The Backup and Recovery topic provides the system programmer with information about planning and accomplishing a backup and recovery strategy and how to recover from disk unit failures and disaster. It includes information about setting up and managing:

- Journaling, access path protection, and commitment control
- User auxiliary storage pools (ASPs)
- Disk protection (device parity, mirrored, and checksum)

It also includes topics on advanced backup and recovery, such as save-while-active support, saving and restoring to a different release, and programming tips and techniques.

The Manage Tape Libraries topic in the Information Center provides information about tasks performed with an automated tape library (ATL).The citation describes recommended methods for ATL design and use.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM Corp., 2004. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2004. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This information documents intended Programming Interfaces that allow the customer to write programs to obtain the services of tape files.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Application System/400
AS/400
e (logo)
IBM
iSeries
Operating System/400
OS/400
400
MVS Operating System
OS/390
z/OS

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

Code disclaimer information

This document contains programming examples.

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you ″AS IS″ without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

Printed in USA