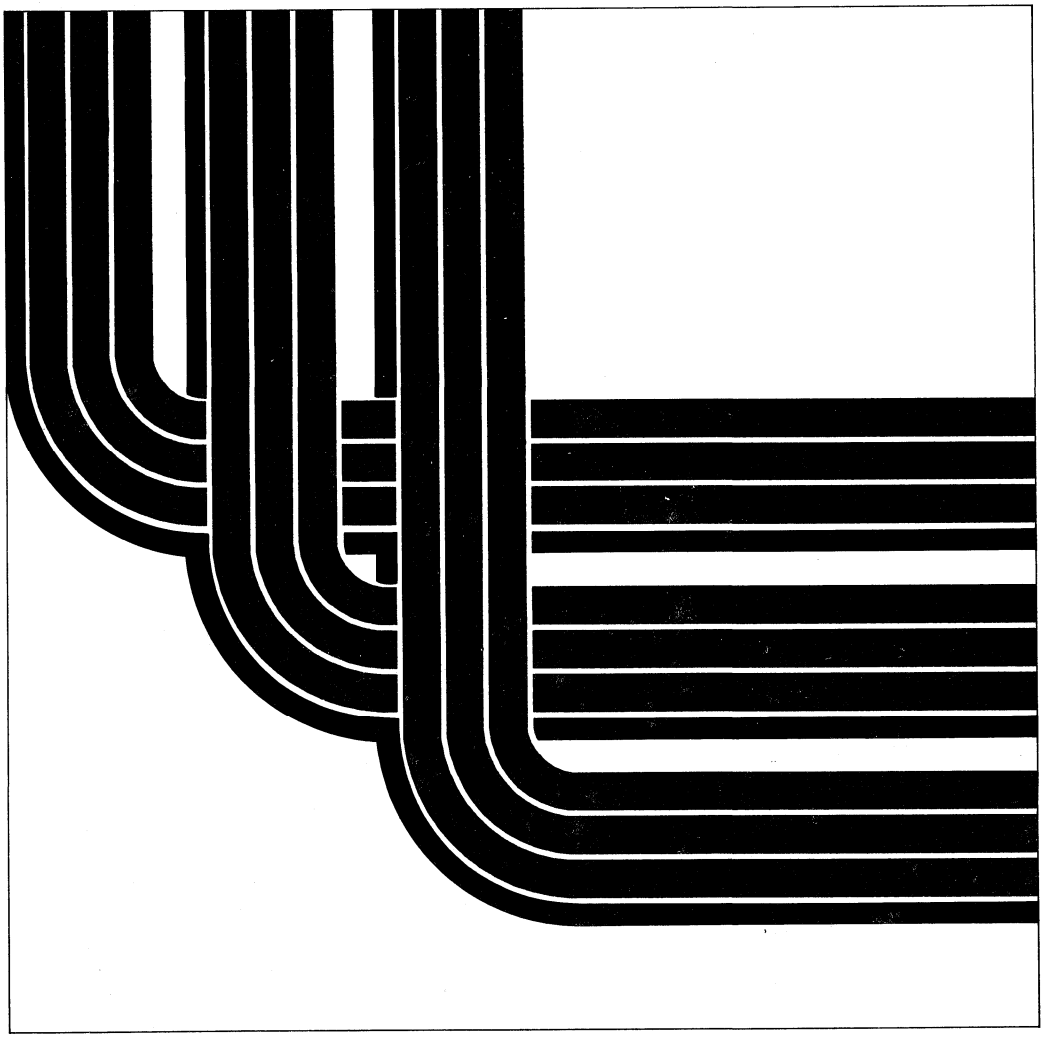
Application System/400

SC41-0012-01

Guide to Programming for Tape and Diskette

Version 2



Application Development

Application System/400

SC41-0012-01

Guide to Programming for Tape and Diskette

Version 2

Take Note!

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

Second Edition (September 1992)

This edition applies to the licensed program IBM Operating System/400 (Program 5738-SS1), Version 2 Release 2 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions. This major revision makes obsolete SC41-0012-00. Make sure you are using the proper edition for the level of the product.

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

A Customer Satisfaction Feedback form for readers' comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

Attn Department 245
IBM Corporation
3605 Highway 52 N
Rochester, MN 55901-7899 USA

or you can fax your comments to:

United States and Canada: 800+937+3430
Other countries: (+1)+507+253+5192

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you or restricting your use of it.

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

Changes or additions to the text are indicated by a vertical line (|) to the left of the change or addition.

This publication 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.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. 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". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Programming Interface Information

This guide is intended to help the customer program for tape and diskette input and output devices. This guide documents Product-Sensitive Programming Interface and Associated Guidance Information provided by Operating System/400.

Product-Sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-Sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

Application System/400	AS/400
C/400	COBOL/400
IBM	OfficeVision
Operating System/400	OS/400
RPG/400	400

About This Guide

This guide describes the tape and diskette media supported by the AS/400 system and the Operating System/400 licensed program. Characteristics and programming use of tape and diskette files are described.

Data management concepts and programming considerations for input and output spooling are described in the *Data Management Guide*, SC41-9658. Printer device file characteristics and programming use is described in the *Guide to Programming for Printing*, SC41-8194. Display file device characteristics and programming use is described in the *Guide to Programming Application and Help Displays*, SC41-0011. Programming considerations for creating backup copies and offline storage, or for transferring information to other devices or systems are described in the *Basic Backup and Recovery Guide*, SC41-0036.

You may need to refer to other IBM manuals for more specific information about a particular topic. The *Publications Guide*, GC41-9678, provides information on all the manuals in the AS/400 library.

For a list of publications related to this guide, see the "Bibliography."

Who Should Use This Guide

This guide is intended primarily for the application programmer.

Before using this guide, you should be familiar with general programming concepts and terminology, and have a general understanding of the AS/400 system and OS/400 program. Should you need an introduction to file support before reading this manual, refer to the manual *System Concepts*, GC41-9802.

Chapter 1. Introduction

Device files are files that provide access to externally attached devices such as tapes, diskettes, printers, displays, spools, and other systems that are attached by a communications line. The two device files described in this manual are:

- **Tape files** which allow access to data files on tape devices
- **Diskette files** which provide access to data files on diskette devices

Each file type has its own set of unique characteristics that determines how the file can be used and the capabilities it can provide. This manual describes the characteristics and use of tape and diskette device files for application programs. For more information about display files, printer files and spooled files, refer to the *Guide to Programming Displays*, *Guide to Programming for Printing*, and *Data Management Guide*, respectively.

When a device file is used by a program, it is referred to by a name, that identifies both the file description and, for some file types, the data itself. To be able to use tape and diskette files to their full capabilities, this manual is designed to help you understand their respective usage characteristics, configuration descriptions, error handling methods, and use in high-level language programs.

Chapter 2. Tape Support

The Application System/400* (AS/400*) system supports the following tape media for creating backup copies and offline storage of information, or for transferring information to other devices or systems:

- 1/2-inch reel
- 1/2-inch cartridge
- 8-mm cartridge
- 1/4-inch cartridge

Tape storage capacities range from 41 to 2332 megabytes, using densities ranging from 1600 to 76 200 bits per inch (bpi), depending on the tape device used.

Tape is particularly useful for storing large amounts of data:

- Reading and writing information with tape is faster than with diskette.
- More information can be stored on a single tape than on diskette.

For information about how to use tape devices for save and restore operations, see the *Basic Backup and Recovery Guide*.

Related CL Commands

The following commands are available to help maintain and use tapes. The *CL Reference* manual contains detailed descriptions of these commands.

Tape Configuration Description Commands:

CHGCTLTAP
: Change Controller Description for Tape: Changes a controller description for a tape controller.

CHGDEVTAP
: Change Device Description (Tape): Changes a device description for a tape device.

CRTCTLTAP
: Create Controller Description for Tape: Creates a controller description for a tape controller.

CRTDEVTAP
: Create Device Description (Tape): Creates a device description for a tape device.

DLTCTLD
: Delete Controller Description: Deletes a controller description.

DLTDEVD
: Delete Device Description: Deletes a device description.

DSPCTLD
: Display Controller Description: Displays a controller description.

DSPDEVD
: Display Device Description: Displays a device description.

Tape Device File Commands:

CHGTAPF
: Change Tape File: Changes certain attributes of a tape device file.

CRTTAPF
: Create Tape File: Creates a tape device file used to read and write records on tape.

DLTF
: Delete File: Deletes files.

DSPFD
: Display File Description: Displays the current characteristics of a file.

OVRTAPF
: Override with Tape File: Temporarily changes a tape file or tape file attributes specified in a program.

Other Tape Support Commands:

CHKTAP
: Check Tape: Searches a tape volume for a specific volume identifier or field label.

CPYFRMTAP
: Copy from Tape: Copies records from a tape file to an output file or a printer.

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

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

DSPTAP
: Display Tape: Displays volume and file label information, saved objects information, or the volume type (*NL or *LTM) and density for volumes without labels.

DUPTAP
: Duplicate Tape: Copies the contents of one tape to another.

INZTAP
: Initialize Tape: Initializes tapes, with or without labels, or can be used to clear all data on the tape from the load point to the end-of-tape marker.

Initializing Tapes

All tapes must be initialized before they can be used by the system. The Initialize Tape (INZTAP) command is used to initialize tapes, with or without labels, and to clear all data on the tape from the load point to the end-of-tape marker.

Note: If the tape drive is an 8-mm cartridge device and CLEAR (*YES) is specified, the operation will take over 2 hours.

In the following example, a tape volume loaded on device TAP01 is initialized to standard label format.

```
INZTAP  DEV(TAP01) NEWVOL(BACKUP)
        DENSITY(1600)
```

The tape volume is initialized with the volume ID BACKUP. EBCDIC code is specified (by default), and the tape density is set at 1600 bits per inch. The AS/400 system uses data conversion tables derived from the American National Standards Institute, Inc, Document ANSI X3.26-1970.

Tape Labeling

The following series of diagrams provides a basic description of standard tape labeling used for the AS/400 system.

In Figure 2-1, the tape is given a volume label (marked VOL1). Two tape marks (TM) are written when the tape is initialized using the INZTAP command.

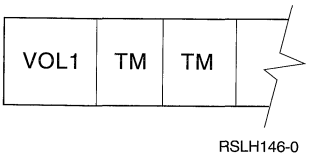
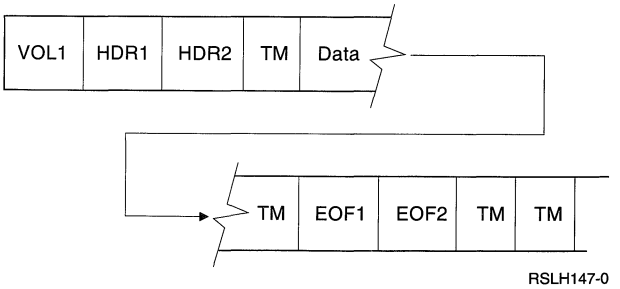


Figure 2-1. Volume Label and Tape Marks

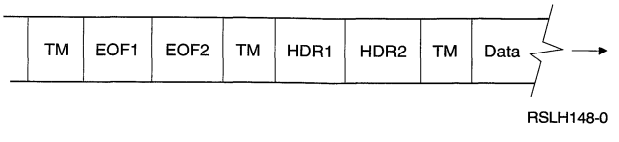
When the tape is used as an output device for a high-level language program, the two tape marks following the VOL label are written over with header labels (HDR1 and HDR2). Each header label is 80 bytes long, the first containing such information as the file name and date, the second specifying information such as record and block lengths, record block format, and buffer offset (for ASCII files). A single tape mark is added following the header labels.

Data written to the tape is added following the tape mark. When the end of the file is reached, a tape mark and two end-of-file labels are written on the tape. The end-of-file labels contain the same information included in the header labels except that the second end-of-file label (EOF2) includes the block count for the file.

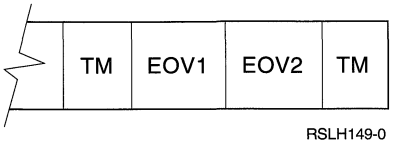
Two tape marks follow the end-of-file labels as shown in the following diagram.



If a second file is added to the tape, the second tape mark following the end-of-file labels is written over by a header label (HDR1) for the new file, followed by the second header label, another tape mark, and the file data as shown in the following diagram.



When the end of the physical tape is reached, a tape mark and two end-of-volume labels are written, followed by a final tape mark. If the file is not complete, it is continued on a second volume, starting with a volume label specifying the tape as volume 2 of the file.



Tape Data Files

Data files can be stored on tape in the following ways:

- Single volume tape data file: A file contained on one reel (volume) of tape
- Multivolume-tape data files: Files contained on more than one reel (volume) of tape
- Multifile volumes: Volumes of tape containing more than one data file

If you use multivolume-tape data files, the following conventions must be followed:

- The labels on each volume must be consistent. You cannot have standard labeled tapes and unlabeled tapes in the same tape group.
- All volumes must be written in the same character code (EBCDIC or ASCII) and density.
- Each tape in the group must have the same record format, block length, and record length.
- If more than one tape device is specified, the volumes must be put on the devices in the sequence specified in the tape device file, as shown in Figure 2-2. For example, if a data file is contained on four volumes (VOL01, VOL02, VOL03, and VOL04) and the tape devices specified, in order, are TAPE01, TAPE02, and TAPE03, the volumes must be placed on a tape drive as follows: VOL01 on TAPE01, VOL02 on TAPE02, VOL03 on TAPE03, and VOL04 on TAPE01.

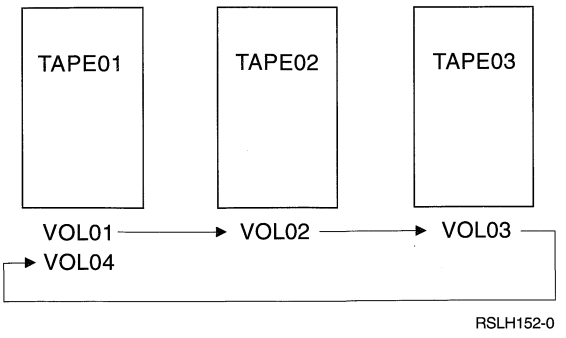
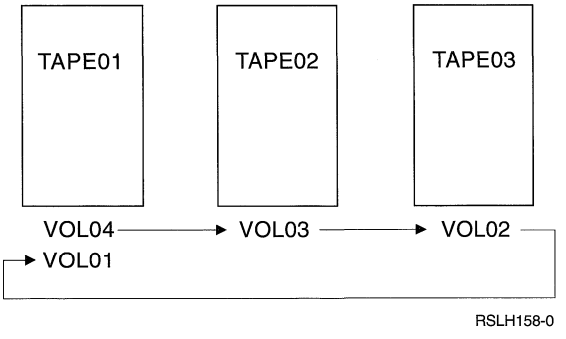


Figure 2-2. Multivolume-Tape-Data-File Sequence Using Three Tape Devices

If volumes are used in reverse order to be read backward, VOL04 is on TAPE01, VOL03 on TAPE02, VOL02 on TAPE03, and VOL01 on TAPE01.



Tape Configuration Descriptions and Device Files

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

- First, a device description must exist for each tape device to describe the device to the system. The tape device description is specified using the Create Device Description (Tape) (CRTDEVTAP) command. The tape device description contains information such as device address, device name, device type, model number, and features. For some tape devices, a tape controller description must also exist.
- Second, a device file must exist for the tape device. Tape device files are created using the Create Tape File (CRTTAPF) command and describe how input data is to be presented to a program from a device, or how output data is to be presented to the device from a program. Tape device files must not

be confused with the actual data files on the tape volumes: the tape device file provides a link between the application program and the tape device for processing a volume containing data files.

It is not necessary to have a separate device file for each tape device; you can use a single device file for several different tape devices using an Override Tape File (OVRTAPF) command. Any number of device files can be associated with one device.

Note: The configuration descriptions must be varied on before they can be used. See the *Device Configuration Guide* for information about varying on configuration descriptions.

IBM-Supplied Tape Device Files

The following tape device files are shipped with the operating system for your use:

- QTAPE (tape file)
- QTAPSRC (tape source file)

These files are all program-described 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 can be used by several programs.

Example of Creating a Tape Device File

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

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

The tape REELS parameter is specified with the value *SL, indicating that the tape uses standard labels. The device name is TAP01. The file is written at sequence number three on the tape (SEQNBR parameter), in EBCDIC code (CODE parameter), and will be unloaded after it has processed (ENDOPT parameter).

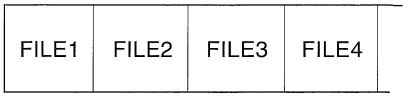
Extending Files on Tape

Tape data files on 1/2-inch tapes can be extended using the EXTEND parameter on the CRTTAPF, CHGTAPF, and OVRTAPF commands. Extending data files is not supported for 1/4-inch or 8-mm cartridge tape devices.

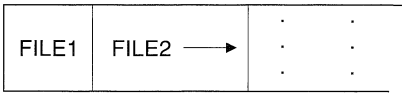
When you extend a file, any existing tape data following the specified file on the tape is no longer accessible by the system.

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: In this example, if you specify EXTEND(*YES *CHECK) on the OVRTAPF command, the expiration date of the file following the file to be extended (FILE3) 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.



RSLH159-0



RSLH160-0

Specifying Tape Device File Parameters

Tape device file records are described 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 may be specified on the CRTTAPF, CHGTAPF, and OVRTAPF commands.

DEV

The name of the device description identifying the tape devices that the tape device file can access.

VOL

The volume identifiers (from 1 through 6 alpha-numeric characters) of the tapes to be used for the device file may be specified using the VOL parameter on the CRTTAPF, CHGTAPF, and OVRTAPF commands.

REELS

The type of label processing used for the tape on which the data file is to reside and the number of tapes (reel number) that are to contain the data file are specified using the REELS parameter. The reel number is ignored for output processing or if a volume list is specified. The reel number is also ignored if standard label processing is specified (using *SL on the REELS parameter). For cartridge tape devices, *SL must be specified for the REELS parameter.

If some of the file labels are incorrect, bypass label processing (*BLP) should be specified. Each reel is checked for a standard label; however, most other volume label information and the file labels on the tape are ignored. Refer to the *CL Reference* manual for more information on label processing.

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

SEQNBR

The sequence number of the data file on tape is specified by the SEQNBR parameter. 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, 2 for the second data file, and so on. Figure 2-3 shows how files are numbered for labeled volumes containing more than one file and multivolume tapes (FILEB is contained on three volumes):

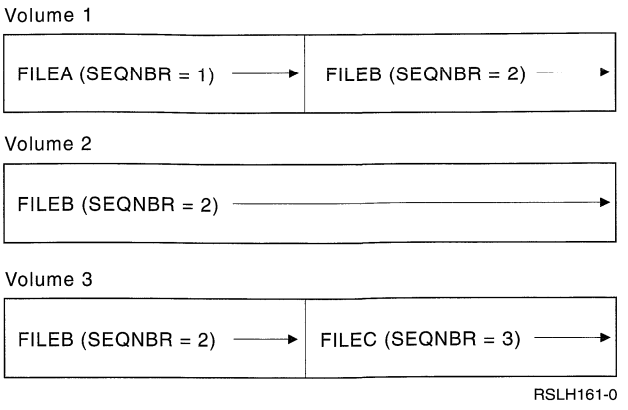


Figure 2-3. Data File Sequence Number on Multi-volume Tapes

The sequence number specified for new standard labeled data files on tape (SEQNBR parameter on CRTTAPF, CHGTAPF, and OVRTAPF commands) must correspond to the physical sequence number of data files on the tape. This means that if files 1 and 2 exist on the tape, the next data file created must have a sequence number of 3. When a new data file is created 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. As shown in the previous diagram, the sequence number of FILEC must be 3, even though there are only two files on the last volume.

The location of a data file on tape is always specified on the SEQNBR parameter. The information specified on the LABEL parameter is used to verify that the correct data file is to be processed only after the file is located using the information specified on the SEQNBR parameter. A data file on tape cannot be located by label name. If the Check Tape (CHKTAP) command is used, the sequence number of the data file is returned in the completion message.

Some special values can be used in place of an actual sequence number:

- *NEXT: The next sequential data file on the tape is processed. If the tape is currently positioned prior to the first data file, the first data file on the tape is processed. *NEXT is useful for applications that need to read all data files on a tape. This value can only be specified for tape device files used to read from tape. An error message

is issued when a tape device file is used to write to a tape and *NEXT is specified.

- *END: The data file will be written to the end of the tape. This value can only be specified in tape device files used to write to tape. An error message is issued when a tape device file is used to read from a tape and *END is specified.

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 draft 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: The existing files are not written over if the tape drive is a 1/4-inch or 8-mm cartridge device.

- *END.

LABEL
The data file label on the tape is specified using the LABEL parameter.

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 input or output file for which EXTEND(*YES) is specified to verify that the correct file is processed.

RCDLEN
The length of records to be used by a program using this device file is specified using the RCDLEN parameter. 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 length of the block of data to be transferred on each input or output operation is specified using the BLKLEN parameter. If *CALC is specified, the system attempts to calculate block length from the file header labels. The block length must be between 18 and 32 767 bytes.

RCDBLKFMT
The format of the input or output records and blocks is specified using the RCDBLKFMT parameter. Records can be:

- Fixed-length, unblocked (*F)
- Fixed-length, blocked (*FB)
- Variable-length, unblocked, unspanned (*V)
- Variable-length, blocked, unspanned (*VB)
- Variable-length, unblocked, 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 input or output tape data files specified as EXTEND(*YES) because this information is taken from the tape labels. If a block length or record block format is specified that does not match that specified in the tape label, the specification from the tape label is assumed. If the record length specified in the program does not match the length of the data, the data is padded or truncated 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. If the data file is not the last data file on the tape, all remaining data files are destroyed. (This is also true when an existing data file is written over.) The record and block length specified in the label are used for the extension. EXTEND is valid only for 1/2-inch reel 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
For 1/2 inch reel, density is expressed as the number of bits per inch (1600, 3200, 6250). For cartridge tape, density is expressed as a value (10 000, 16 000, 38 000, 43 200) or by tape format (*FMT3480, *FMT3490E, QIC120, QIC525, QIC1000). All data files on a volume are recorded in the same density. The DENSITY parameter is used only to set the output volume density when the first data file on a volume not labeled is created. The

volume label on a labeled tape is used to determine the number of bits per inch.

CODE
: The character code (EBCDIC or ASCII) for the data on tapes not labeled is specified using the CODE parameter. For standard label tapes, the volume label determines the character code. When ASCII is the interchange code, volume labels, header labels and data written by the system conform to "American National Standard" X3.27-1978, "Magnetic Tape and File Structure for Information Interchange."

CRTDATE
: The creation date of an input data file on a labeled tape is specified using the CRTDATE parameter. If the creation date written on the tape does not match the date specified in the file description, a message is sent to the system operator, who determines what should be done.

EXPDATE
: The expiration date of an output data file on a labeled tape is specified using the EXPDATE parameter. The data file cannot be written over until the date has expired. The data file is considered to be protected.

  When an output data file is created on a labeled volume (instead of extending an existing data file), the expiration date of the new data file is compared to the date of the file preceding it on the volume. If the expiration date of the new data file is later than the expiration date of the data file preceding it on the tape, an inquiry message (CPA4036) is sent. The system operator can:

  - Allow the data file to be created
  - Put on a new tape and try again
  - End processing

  Note: Creating the data file could 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
: When you use a multivolume-tape data file and specify ENDOPT(*LEAVE), the first volume must be put on the first tape device specified in the DEV parameter (except for a read backward, in which case the last volume must be put on the first tape device specified). If a user opens the data file again (using the same device list) and the tape is left on a tape device other than the first one specified in the DEV parameter, the tape reel must be moved to the first tape device specified in the DEV parameter before the next data file on that tape reel is opened.

  The ENDOPT parameter can be used to specify where magnetic tape is to be positioned when the tape device file is closed. The magnetic tape can be rewound to the load point, left as it is, or unloaded.

USRLBLPGM
: User header and trailer labels are supported through the use of the USRLBLPGM parameter. USRLBLPGM identifies the user program that is used during open and close processing. See "Processing User Labels" on page 2-12 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 format *D or *DB file to process a block with 4-digit block descriptors.

For additional tape information, and information about using tape for save and restore operations, see the *Basic Backup and Recovery Guide*.

Figure 2-4 on page 2-8 lists parameters that apply to magnetic tape and where the parameters can be specified. The *CL Reference* manual contains detailed information about how to specify these parameters on the CRTTAPF, CHGTAPF, and OVRTAPF commands.

Figure 2-4. 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	RPG/400*, COBOL/400*, BASIC, PL/I, or C/400* 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	Number of labeled tapes (reels)	Number of reels	Number of reels	
REELS	Volume label type (reels)	*SL, *NL, *NS, *BLP, or *LTM	*SL, *NL, *NS, *BLP, or *LTM	
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	RPG/400, COBOL/400, BASIC, PL/I, or C/400 programming languages
BLKLEN	Block length	*CALC or block length	*CALC or block length	COBOL/400 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/400, C/400 programming languages
EXTEND	Extend	*NO, *YES *CHECK, or *YES *NOCHECK	*NO, *YES *CHECK, or *YES *NOCHECK	COBOL/400, C/400 programming languages
DENSITY	Density	*DEVTYPE, *FMT3480, *FMT3490E, 1600, 3200, 6250, 10 000, 16 000, 38 000, 43 200, QIC120, QIC525, or QIC1000	*DEVTYPE, *FMT3480, *FMT3490E, 1600, 3200, 6250, 10 000, 16 000, 38 000, 43 200, QIC120, QIC525, or QIC1000	
CODE	Character code	*EBCDIC or *ASCII	*EBCDIC or *ASCII	COBOL/400 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/400 programming language
USRLBLPGM	User label program	*NONE or qualified program name	*NONE or qualified program name	
IGCDTA	Double-byte data	*NO or *YES	*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		
REPLACE	Replace existing file	*YES or *NO		
TEXT	Text	*BLANK or text		

Using Tape Device Files in High-Level Language Programs

A magnetic tape device can be accessed with a program-described device file. 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 you use determines what tape parameters can be specified in the program.

Open Processing for Tape Device Files

The following considerations apply to opening tape device files:

- When a tape device file is opened, any parameters specified in the file are merged (overridden) with the parameters specified in the program. These parameters are then merged (overridden) with the parameters specified on an OVRTAPF command, if specified.
- The device names must be specified when the tape device file is opened. If DEV(*NONE) is specified in the tape file, one or more device names must be specified 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) specified for the file are always returned to the program in the data management open feedback area. They are returned 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 used for the file.
- The read-backward operation is supported for fixed (*F), fixed block (*FB), and undefined format (*U) data files and for both single volume tape data files and multivolume-tape data files. It can be requested only through a high-level language when the file is opened. 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:

  - 9348 tape unit
  - 8-mm cartridge device
  - 1/4-inch cartridge devices

  If a device and volume list are specified for a data file that is to be read backward, volume positioning on devices is the same as though the data file were to be read forward. 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 for standard label processing (*SL) or bypass label processing (*BLP) if the system can recognize the first volume of the data file from the header labels. If the header labels are not recognized for the first volume of the data file or if this is a *BLP file, the end-of-file condition is signaled when the number of reels specified on the REELS parameter or the number of identifiers on the VOL parameter has been processed.
- Some high-level languages allow you to specify where the tape is to be positioned when an input tape device file is opened. This indicates if the tape is to be processed in the forward or the backward direction.

The rules for determining the first volume of a data file from the labels are:

  - HDR1 label multivolume sequence field = 1 (ASCII or EBCDIC with no HDR2 label)

    or

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

- The record length must be specified according to the information shown in Figure 2-5.
- 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, unblocked 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, a date and sequence number are appended to each record on input operations and are removed on output operations. (The date field is always 0.) 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 for a source file have the same ratio as the block length and record length for data records minus the 12 bytes needed for source files. For example, if the actual type data record length is 80 and block length is 800, to use the file as a source file the record length becomes 92 and the block length is unchanged at 800.
- For input file processing using standard labels, the block length in the file label is used regardless of what is specified when the device file is opened or what is specified in the device file for block length.
- 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.
- The sequence number *must* be specified so the data file can be found. Tape data files cannot be located by label name.
- If both the VOL and REELS parameters are specified, the volume list is used to control the volumes processed by the device file, even if the REELS parameter is specified in a previous override (at a higher call level than the oldest VOL parameter specification). If you want to use the REELS parameter (number of reels) to limit the number of input volumes processed, specify *NONE for the VOL parameter.

Figure 2-5. 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 performed on data files:

Read and Write Considerations:

- Record length must be specified 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 records are truncated to the maximum length allowed. A diagnostic message is sent when the device file is opened to indicate that output records may be truncated.

- If the record length specified by the program differs from the actual length of the data, the data is padded or truncated as necessary to match the program specification.

Read Considerations:

- If no end-of-file label is found, tapes are processed until the specified volume identifiers have been used. If VOL(*NONE) is specified, tapes are processed until the specified number of reels (REELS parameter) is reached. Message CPA5230 is sent to the system operator message queue when all the identifiers in the VOL list have been used. When you receive this message, you can:
  - Cancel processing of the device file immediately. The device file is closed.
  - Continue: Other volumes are processed.
- When a tape block is read that is not a valid length, a CPF5036 notify message is sent to the program. If your high-level language reports this condition to your program, it can continue processing by attempting to read another record. When you continue processing this way, the block that is not valid is skipped 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 all buffered records to be written to tape. When this occurs, a short block may be written 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 be switched immediately, or signals end-of-file if there are no more continuation volumes for an input file.

Close Processing for Tape

When a tape device file is closed, one of the following functions is performed according to what you specify in the tape device file, using the ENDOPT parameter on a CRTTAPF, CHGTAPF, or an OVRTAPF command.

- The tape is rewound.
- The tape is left as it is.
- The tape is rewound and unloaded so that it can be removed from the magnetic tape device.

If the tape operation ends abnormally, the position of the tape when the device file is closed may be unchanged, or the tape may be rewound regardless of what has been specified.

Handling Tape Processing Errors and Damaged Tapes

Tape data files can be damaged when, due to an error condition or a system failure, no tape marks or labels are written at the end of the data file on the tape. 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 and existing data may appear to be concatenated when processed for input. If the tape is labeled, an error message may be sent to the system operator when the trailer labels are read. No error is detected for unlabeled tape.
- If the new data and the existing data do not appear to be concatenated, an error message is sent to the system operator.
- If the tape contains no existing data or tape marks beyond the location of the last data block (the tape may be new or completely

deleted), the tape advances until it runs off the end of the reel when used as input.

Note: Whenever an output file is closed, the system attempts to write an end-of-tape marker and label at the end of the file even if a previous error that signaled an escape message was reported. If a file close is attempted and the system cannot write closing tape marks and labels, a message is sent to your job log.

However, if an error occurs in the process of saving a file to 1/2-inch reel or cartridge tape, the system prompts the operator to mount another tape volume or cancel the job. If the operator mounts another tape volume, the system repositions the tape a number of blocks before the error occurred and, if the volume can be closed, writes end-of-volume labels. The job then continues writing data beginning with the first block of data that was overwritten with end-of-volume labels.

If damage occurs while using a cartridge tape device, blank tape is encountered and an error message is sent to the system operator.

Processing User Labels

The USRLBLPGM parameter can be used 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 specified on the USRLBLPGM parameter during open and close processing for every label processed, plus one additional time to indicate when no more labels are being passed.

Figure 2-6 shows the format of a tape with user labels. A user label program used to process the labels in the diagram would be called three times for open processing (for UHL1, UHL2, and a final time to indicate no more labels being passed). The program would be called three more times for close processing.

Three variables are passed to the program by the system. The variables have the following lengths:

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

The following list shows the content of the variables passed to the user label program:

Parameter 1

Position 1 - 80
User header or trailer label

- For output files, this parameter indicates the next user label to be written, set by the user label program.
- For input files, this parameter indicates the user label most recently read from the tape, set by the system.

Parameter 2

Position 1
End-of-labels indication

Parameter 2 contains a character `0` or `1`, indicating whether or not the label read is the last label to be read. For output files, the value is set by the user label program; for input files, the value is set by the system.

- `0` indicates that Parameter 1 contains a label.
- `1` indicates that Parameter 1 does not contain a label. All labels have been processed.

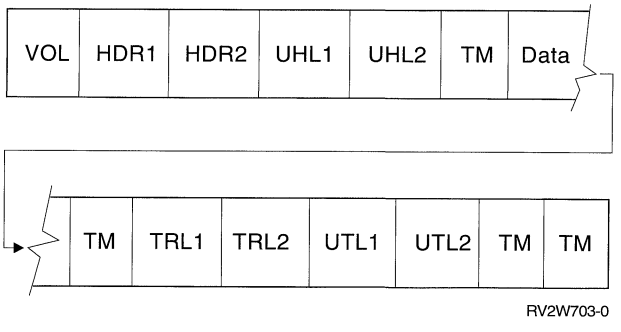


Figure 2-6. Tape with User Labels

Parameter 3

Position 1- 80
Current volume label

Position 81-160
Last processed HDR1 or TRL1 label, whichever was encountered last

Position 161-240
Last processed HDR2 or TRL2 label, whichever was encountered last

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 count of the number of user labels read in the current header or trailer group.

Position 243
Open file option

The open file option field contains a character indicating whether the file was opened as an input or output file.

- I indicates that the file was opened as an input file.
- O indicates that the file was opened as 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.

Other Tape Support Commands

Several tape support functions are provided by the following CL commands.

CHKTAP
Check Tape: Searches a tape volume for a specific volume identifier or field label.

The following command checks a tape volume loaded on device TAP01 to see if a file named TEST is located at sequence number 3.

```
CHKTAP  DEV(TAP01) SEQNBR(3) LABEL(TEST)
```

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

The following command dumps the second file, including all label information, from a tape volume loaded on device TAP02.

```
DMPTAP  DEV(TAP02) SEQNBR(2) TYPE(*ALL)
```

DSPTAP
Display Tape: Displays volume and file label information, saved objects information, or the volume type (*NL or *LTM) and density for volumes without labels.

The following command displays the volume and file label information of a tape volume loaded on device TAP01.

```
DSPTAP  DEV(TAP01)
```

DUPTAP
Duplicate Tape: Copies the contents from one tape volume to another.

The following command duplicates the tape volume mounted on device TAPE01 onto the tape volume mounted on device TAPE02.

```
DUPTAP FROMDEV(TAPE01) TODEV(TAPE02)
```

Considerations for Using the DUPTAP Command

To use the DUPTAP command your system must have two tape drives. In addition, these considerations apply:

- Nonlabeled tapes cannot be duplicated to 1/4-inch cartridge or 8-mm cartridge devices.
- The density field in the file labels is updated to reflect the true density.
- Byte 80 on the volume label on a 6157 tape is reset from a Q to a blank.
- Tape duplication is not necessarily a volume to volume relation because of the different tape capacities.

  When duplicating a tape, the tape being copied to should have a larger capacity than the original tape (the one being copied from).

Performance Considerations for Tape

The 1/4-inch tape devices are intended primarily for save and restore operations. These devices are not designed for operations that cannot process data fast enough to keep the tape moving. Performance and reliability will suffer if the tape stops and starts too often.

If you use the 1/4-inch cartridge devices for operations other than save and restore, the tape will

more likely keep moving when you do any of the following:

- Process large tape blocks (BLKLEN parameter). If the records are small, use a blocked format (specify *FB, *DB, *VB, or *VBS on the RCDBLKFMT parameter).
- Use fixed-length records, because they are processed more efficiently than variable-length records.
- Design the application to do as little processing as possible between tape read or write operations. The best application design may be one that uses a system copy file command (CPYF, CPYFRMTAP, or CPYTOTAP) to copy records between tape and a database file, with application programs that only process records in the database file.

Chapter 3. Diskette Support

Diskettes can be used to create backup copies of information, provide offline storage of files and libraries, or to transfer information to other systems or devices.

The AS/400 system supports both 5 1/4-inch and 8-inch diskettes, with the following restrictions:

- 5 1/4-inch diskettes must be double-sided, high capacity (2HC) diskettes. These diskettes have a capacity equivalent to 8-inch, double density diskettes. When initializing these diskettes, use 2D format and sector sizes 256, 512, or 1024.
- 8-inch diskettes can be any of the following three types:
  - Diskette 1 is a single-sided, single-density diskette.
  - Diskette 2 is a double-sided, single-density diskette.
  - Diskette 2D is a double-sided, double-density diskette.

Double-density means that a diskette contains information on both sides and with two times the amount of information stored in the same space as a diskette 1. A diskette 2D holds approximately four times the amount of information as a diskette 1.

For information about how to use the diskette devices for save/restore operations, see the *Basic Backup and Recovery Guide*.

Related CL Commands

The following commands are available to help maintain and use diskettes. The *CL Reference* manual contains detailed descriptions of these commands.

Diskette Device Description Commands:

CHGDEVDKT
: Change Device Description (Diskette): Changes a device description for a diskette device.

CRTDEVDKT
: Create Device Description (Diskette): Creates a device description for a diskette device.

DLTDEVD
: Delete Device Description: Deletes a device description.

Diskette Device File Commands:

CHGDKTF
: Change Diskette File: Changes certain attributes of a diskette device file.

CRTDKTF
: Create Diskette File: Creates a diskette device file used to read and write records on diskette.

DLTF
: Delete File: Deletes files.

DSPFD
: Display File Description: Displays the current characteristics of a file.

OVRDKTF
: Override with Diskette File: Overrides a diskette device file or the file attributes specified in a program.

Other Diskette Support Commands:

CHKDKT
: Check Diskette: Checks a diskette for a specific volume identifier, file label, or both.

CLRDKT
: Clear Diskette: Clears all files on the diskette by deleting the labels and creates an expired file, labeled DATA, that includes the entire diskette. The data is not deleted.

CPYFRMDKT
: Copy from Diskette: Copies records from a diskette file to an output file or a printer.

CPYTODKT
: Copy to Diskette: Copies records to a diskette file from a physical, logical, tape, diskette, or spooled inline file.

DLTDKTLBL
: Delete Diskette Label: Deletes a file label from a diskette.

DSPDKT
Display Diskette: Displays the volume and file labels, or save and restore information on a diskette.

DUPDKT
Duplicate Diskette: Copies information from one diskette to one or more diskettes.

INZDKT
Initialize Diskette: Initializes a diskette, to write identification information and to format the diskette for use by the system.

RNMDKT
Rename Diskette: Changes the volume and owner identifiers in the volume label.

Diskette Exchange Types

Diskette data must use one of the following exchange types, and the diskettes must have the attributes (diskette type, sector size, and record length) appropriate for that exchange type. The exchange types are:

- Basic Exchange

  Basic exchange data sets may be exchanged between systems capable of reading and writing both the IBM* Diskette 1 and the IBM Diskette 2. The sector size must be 128 bytes. The length of the records can be from 1 to 128 bytes, with one record per sector.

- H Exchange

  Type H exchange data sets may be exchanged between systems capable of reading and writing the IBM Diskette 2D. The sector size must be 256 bytes. The length of the records can be from 1 to 256 bytes, with one record per sector.

- E Exchange

  Type E exchange data sets force the using system to examine each field in the header label. E exchange data sets are created by save/restore operations.

- I Exchange

  Type I exchange data sets may be used for:

  - Diskette 1 or Diskette 2 with sector sizes of 128, 256, or 512 bytes
  - Diskette 2D with sector sizes of 256, 512, or 1024 bytes

  The record length can be from 1 to 4096 bytes. One, more than one, or part of a record may be contained in one sector, or one record may span across sectors, depending on sector size and record length.

For these exchange types, each record in the file is fixed-length (each record contains the same number of bytes).

Initializing Diskettes

Before you can use a diskette on the system it must be initialized and have a volume label written on it. Diskettes are normally initialized and ready for use when you receive them.

The Initialize Diskette (INZDKT) command initializes a diskette to a specified format. You may want to use the INZDKT command to reinitialize a diskette in the following situations:

- If you want to change the sector size on a diskette to a size that is acceptable for a specified exchange type (basic, H, or I) or to accommodate save/restore.
- If you encounter errors while reading or writing a diskette. Initializing the diskette may correct the errors or assign an alternate cylinder to be used in place of a defective cylinder. A **cylinder** is a set of tracks on a diskette that can be read without changing the position of the read/write head. If you encounter more than two defective cylinders during the process of preparing a diskette, you receive a message stating that the diskette is unusable. If you receive this message, you should discard the diskette.

Initializing a diskette deletes the data contained on the diskette. If you want to save the data, you must do so before initializing the diskette. You can use the Duplicate Diskette (DUPDKT) command to copy the entire diskette to another diskette, or the Copy File (CPYF) command to copy each file on the diskette to a database file or to another media. You may also use the CPYFRMDKT and CPYTODKT commands to copy data from or to a diskette.

The AS/400 system uses data conversion tables derived from the American National Standards Institute, Inc, Document ANSI X3.26-1970.

Multivolume-Diskette Data Files

Depending on the size of your data files, you may place several files on one diskette, called a multi-file volume. If a single data file occupies more than one volume, it is called a multivolume-diskette data file.

If you use multivolume-diskette data files, the following conventions must be followed:

- All volumes of a multivolume set must have the same format (1, 2, or 2D) as the first volume.
- All volumes must have the same sector size (128, 256, 512, or 1024 bytes).
- All volumes must be written in the same character code (EBCDIC or ASCII).
- For an input file, the file exchange types must be the same on each volume of a multivolume set.
- For an input file, the length of the records in the file must be the same on each volume of the file.
- Except for the first volume, multivolume-diskette data files cannot be written to diskettes that contain active data files. An **active data file** is a diskette data file that has an expiration date greater than the system date. If you attempt to do this, a message is sent to the system operator that allows the active data files to be ignored (written over).
- A data file cannot be written to a diskette data file that has an extended label area. (Extended label areas—cylinders that have been allocated to contain additional data set labels—are not supported by the system.)

When you use a diskette for spooled output, make sure the output is written to the correct diskette. You can do this by specifying the VOL parameter when the output is spooled. The advantage of spooling output is that more than one job can be running at the same time while output is produced.

Note: The volume sequence number field in the diskette file label for multivolume-diskette data files has only two positions. Volume 100 is indicated by 00, after which the numbers return (wrap around) to 01. An informational message is sent to the system operator each time the numbers return to 01.

To use diskettes for other than saving and restoring, the following conventions must be observed:

- Each volume identifier can be from 1 through 6 alphanumeric characters.
- If you have multivolume-diskette data files, a volume identifier can apply to more than one volume. However, you are not required to use the same volume identifier for every volume in a multivolume diskette group.
- Files are not written to diskettes with extended label areas.

Diskette Device Descriptions and Device Files

To access data from a diskette on the system, two objects must exist:

- First, a device description must exist for each diskette device to describe the device to the system. The diskette device description is created using the Create Device Description (Diskette) (CRTDEVDKT) command. The diskette device description contains information such as device address, device name, device type, model number, and features.
- Second, a device file must exist for the diskette device. The device file describes how input data is presented to a program from the device, or how output data is to be presented to the device from a program. Diskette device files must not be confused with the actual data files on the diskette volumes: the diskette device file provides a link between the application program and the diskette device for processing the diskettes containing the data files.

Diskette device files must be program-described, meaning that fields and records are described in the program that processes the device file, rather than in the device file itself. It is not necessary to have a separate device file for each diskette device; you can use a single device file for several different diskette devices by using an override command. Any number of diskette device files can be associated with one diskette device.

Note: The device must be varied on before it can be used. This function can be performed auto-

matically when the system is started or you can use the appropriate vary command. See the *Device Configuration Guide* for information about varying on device descriptions.

IBM-Supplied Diskette Device Files

The following diskette device files are shipped with the operating system for your use:

- QDKT (diskette file)
- QDKTSRC (diskette source file)

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

Note: The device (DEV) for these diskette device files uses the default value *NONE. You must use the CHGDKTF or OVRDKTF commands to specify a diskette device before using these files.

Example of Creating a Diskette Device File

You can create additional device files to fit your needs. For example, you can create additional device files to:

- Direct output to a special output queue
- Contain the specific volume and label information for a diskette data file that can be used by several programs
- Spool output or not spool output

In the following example, a diskette device file, DKFILE, is created for output that is written to diskette.

```
CRTDKTF  FILE(DKFILE) DEV(DKT01) CODE(*ASCII)
```

The diskette device (specified on the DEV parameter) is DKT01, and the diskette data file will be written in ASCII code. All other parameters on the CRTDKTF command use their default values.

Because the default values were assumed in this example, the diskette volume, file label, and the creation date of the data file on the diskette must be specified in another CL command or in each program that uses the device file.

Specifying Diskette Device File Parameters

In diskette device files, the data in each record is described in the application program. The system views each record as one field with a length equal to the record length.

The following diskette-device-file attributes may be specified for the CRTDKTF, CHGDKTF, or OVRDKTF command:

- Spooling information for output files, including:
  - Output queue name
  - Maximum number of records that can be spooled for the data file
  - When output is made available to a writer program (at job end, at file end, or immediately as it is spooled)
  - Whether the output is to be held on the output queue until the system operator releases it
  - Whether the output is to be saved after it is produced

  Note: If you do not spool output, you must ensure that the device which the output is to be written to is assigned to your device file.
- The device with which the device file can be used. You can specify a device name on the create device file command. The device name can also be specified using the OVRDKTF or CHGDKTF command.
- The wait time for file allocation. The number of seconds the system is to wait for the file resources to be allocated when the device file is opened.
- Whether or not the open data path (ODP) for the device file can be shared.
- The volume identifiers of the diskettes to be used for the device file (specifying VOL(*NONE) causes volume checking to be ignored).
- The data file label on the diskette.
- The character code (EBCDIC or ASCII) for the data on the diskettes.
- The creation date of an input data file. If the creation date written on the diskette does not match the date specified in the diskette device file, a message is sent to the system operator, who determines what should be done. The

date must be specified in the format specified in the system value QDATFMT. However, the specified date is put in the diskette label in the format, yymmdd.

- The exchange type (basic, H, or I) to be used when creating an output file on the diskette. This attribute is not used when processing an input file.
- The expiration date of an output data file on diskette. The expiration date means that the data file cannot be written over until the date has expired. The file is considered to be protected. The date must be specified in the format specified in the system value QDATFMT. However, the specified date is stored in the diskette label as yymmdd.

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

Figure 3-1 lists the parameters that apply to diskettes and where the parameters can be specified. The *CL Reference* manual contains detailed information about how to specify these parameters on the CRTDKTF, CHGDKTF and OVRDKTF commands.

Using Diskette Device Files in High-Level Language Programs

A diskette device can be accessed from a program using a program-described diskette device file. To use the diskette device file with a program, you must either specify the diskette device file name in the program or use an Override with Diskette File (OVRDKTF) command.

To use the diskette device directly, you specify the device name and the default SPOOL(*NO) parameter. The high-level language you use determines what diskette parameters can be specified in the program. If the parameters are not specified in the file description or program, they can be specified in an OVRDKTF command.

Figure 3-1 lists the parameters that apply to diskettes and where the parameters can be specified.

Figure 3-1 (Page 1 of 2). Diskette File Parameters

CL Parameter	Description	Specified on CRTDKTF Command	Specified on OVRDKTF Command	Specified in HLL Programs
FILE	File name	Qualified device file name	Device file name	RPG/400, COBOL/400, BASIC, PL/I, and C/400 programming languages
DEV	Device name	*NONE or list of device names	List of device names	
VOL	Volume	*NONE or volume identifier	*NONE or volume identifier	
LABEL	Label	*NONE or file label	File label	PL/I
FILETYPE	File type	*SRC or *DATA		
EXCHTYPE	Exchange type	*STD, *BASIC, *H, or *I	*STD, *BASIC, *H, or *I	
CODE	Code	*EBCDIC or *ASCII	*EBCDIC or *ASCII	COBOL/400 programming language
CRTDATE	Creation date	*NONE or date	*NONE or date	
EXPDATE	Expiration date	*NONE, *PERM, or date	*NONE, *PERM, or date	
SPOOL	Spool data	*NO or *YES	*NO or *YES	
OUTQ	Output queue	Qualified name	Qualified name	
MAXRCDS	Maximum records	*NOMAX or maximum records	*NOMAX or maximum records	
SCHEDULE	Schedule	*FILEEND, *JOBEND or *IMMED	*FILEEND, *JOBEND or *IMMED	
HOLD	Hold	*NO or *YES	*NO or *YES	
SAVE	Save	*NO or *YES	*NO or *YES	

Figure 3-1 (Page 2 of 2). Diskette File Parameters

CL Parameter	Description	Specified on CRTDKTF Command	Specified on OVRDKTF Command	Specified in HLL Programs
OUTPTY	Output priority	*JOB or output priority	*JOB or output priority	
USRDTA	User data	*BLANK or user data	*BLANK or user data	
IGCDTA	Double-byte data	*NO or *YES	*NO or *YES	
WAITFILE	File wait time	*IMMED, *CLS or number of seconds	*IMMED, *CLS or number of seconds	
SHARE	Shared file	*YES or *NO	*YES or *NO	PL/I
AUT	Authority	*CHANGE, *ALL, *USE, *EXCLUDE, or authorization list name		
REPLACE	Replace existing file	*YES or *NO		
TEXT	Text	*BLANK or text		
	Record length			RPG/400, COBOL/400, BASIC, PL/I, and C/400 programming languages

Open Processing for Diskette Device Files

The following considerations apply to opening diskette device files:

- When a diskette device file is opened, the parameters specified in the file are merged (overridden) with the parameters specified in the program. These parameters are then merged (overridden) with the parameters specified on an OVRDKTF command, if specified.
- The device name must be specified when the diskette device file is opened if the file is not to be spooled; that is, the default SPOOL(*NO) is specified. If DEV(*NONE) is specified in the diskette device file, the device name must be specified on an OVRDKTF command.
- For an input file, the program may specify the record length, but it is not a requirement. If record length is not specified or is specified with a length of 0, the system will determine the record length from the data file label on the diskette. If the program specifies a record length that is not equal to the length of the records in the data file, the records are padded or truncated to the length specified in the program. A diagnostic message is sent stating that the records are padded if the program record length is greater than the length of the records in the data file.
- For output files, record length must be specified in the program. When the record length specified in the program exceeds that for which the diskette is formatted, a diagnostic message is sent to your program, and the records are truncated.

  Maximum record lengths supported by exchange types are:

Basic exchange	128 bytes
H exchange	256 bytes
I exchange	4096 bytes

- The file label name is required when the file is opened. If the diskette device file was created with LABEL(*NONE) specified, the label must be supplied with the OVRDKTF command. The label name must not exceed 8 characters for files in exchange type BASIC, H, or I. The first character must be alphabetic (A through Z, #, $, or @). All remaining characters in the label name can be any character (A through Z, 0 through 9, #, $, or @) except a blank. The period character (.) is valid only in labels on E exchange files. If the period appears in a label on a BASIC, H, or I exchange file, an error is returned.

  A data file will not be written with a label name that is not valid. If such a label name is encountered for an input file, a diagnostic message is sent and an attempt is made to read the file.
- When the data file type specified in the diskette device file is a source file, a date and

sequence number are added to each record on input operations and are removed on output operations. (The date field is always 0.) The program can specify (if the high-level language you are using allows it) that the system should check to determine if the input or output file is a source file. The record length specified in a program that uses a source file must include 12 bytes for source data. For example, if the data is 80 bytes long, a record length of 92 must be specified in the program.

- A volume identifier specifies the diskette to process. If no volume identifier is specified, the diskette is processed without checking the volume identifier. When a volume identifier is specified, the correct volume must be loaded. If not, a message that requests that the correct diskette be put in is sent to the system operator.
- For multivolume-diskette data files, all volumes must be on the same type of diskette, must have the same sector size, must have the same record length, and must be written in the same character code. For output files, all volumes after the first volume must be written to diskettes that do not contain active files. If active files exist, a message is sent to the system operator that allows the active files to be ignored (overwritten).
- When an output file is created, all expired files on a diskette are written over; no messages are sent to your program. An expired file is a file having an end date less than or equal to the system date.

I/O Processing

The following considerations apply to input/output (I/O) operations performed on diskette data files:

Read and Write Considerations

- When a volume exchange occurs while processing multivolume-diskette data files, a message identifying the current volume identifier is sent to your program.
- If the record length specified in the program does not match the length of the data, the data is padded or truncated to match the record length specified in the program.

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 does not write any data on diskette.
- The force-end-of-data function for an input file positions the diskette at the last volume of the file and signals end-of-file to the program using the file. Refer to the chapter on spool support in the *Data Management Guide* for information about how this function is handled for a SPOOL(*YES) output file.

Close Processing

When a diskette device file is closed, the labels for output data are written on the diskette before the file resources are deallocated.

Handling Diskette Errors

If you encounter an error on the input diskette while using the DUPDKT command, the operation is ended without any data being written to the output diskette.

If you encounter an error on the input diskette while using the CPYF, CPYTODKT, or CPYFRMDKT command, all the data in the data file before the record that caused the error is copied. The remaining data in the data file is not copied.

Chapter 4. Overriding Device Files and Device File Attributes

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 to the way a program functions or to select the data on which it operates without having to recompile the program.

Determining Whether or Not to Use Overrides

Use the following properties of overrides to help you decide if overrides are appropriate for the task you want to perform:

- Overrides remain in effect only for the job or program for which they are issued. They do not permanently change the attributes of a file.
- Overrides have no effect on the other jobs that may be running at the same time.
- Overrides that are to be applied must be specified before the file is opened by a program or before a program that opens a file is compiled.
- Override commands can be entered interactively from a display station or as part of a batch job.
- Override commands can be included in a control language (CL) program, or they can be issued from other programs via a call to the program QCMDEXC.
- Overrides are affected by call levels.

Tape and Diskette Override Commands

You can process override functions for files using the following CL commands:

OVRDKTF
: Override with Diskette File: Overrides (replaces) the diskette file named in the program, overrides certain parameters of a diskette file that is used by the program, or overrides the file and certain parameters of the file to be processed.

OVRTAPF
: Override with Tape File: Overrides (replaces) the tape file named in the program, overrides certain attributes of a file that is used by the program, or overrides the file and certain attributes of the file to be processed.

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 attributes that can be specified depend on the high-level language in which the program is written.)
- 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:

- Device TAP01 is used
- Data is written on the tape with a density of 1600 bits per inch (bpi)
- The ASCII character code type is used
- When the tape file is closed, the tape is rewound and unloaded.

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

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

The tape file OUTPUT is specified in your application program 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)
```

Then you call the application program, and a tape density of 6250 bpi is used and the end option is *REWIND.

When the application program opens the file, the file overrides, program-specified attributes, and file attributes are merged to form the open data path (ODP) which is used during the running of the program. File overrides have precedence over program-specified attributes. Program-specified attributes have precedence over file-specified attributes. In this example, when the file is opened and output operations are performed, the device TAP01 is written to using a density of 6250 bpi, a character code type of EBCDIC and an end option of *REWIND.

Figure 4-1 explains this example.

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 files that have been moved or renamed after the program has been compiled. For example, you want the output from your application program to be sent to tape file TAPE20 instead of the tape file OUTPUT1 (OUTPUT1 is specified in the application program). Before you run the program, enter the following:

```
OVRTAPF FILE(OUTPUT1) TOFILE(TAPE20)
  LABEL(FILE01)
```

The file TAPE20 must have been created by a CRTTAPF command before it can be used.

Figure 4-2 on page 4-3 explains this example.

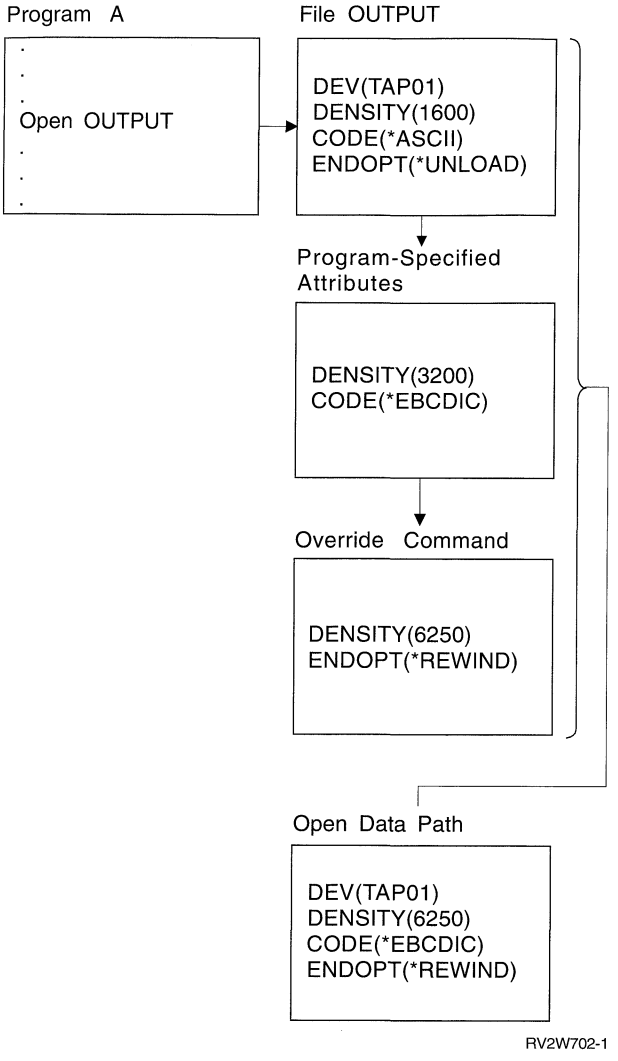


Figure 4-1. Overriding File Attributes

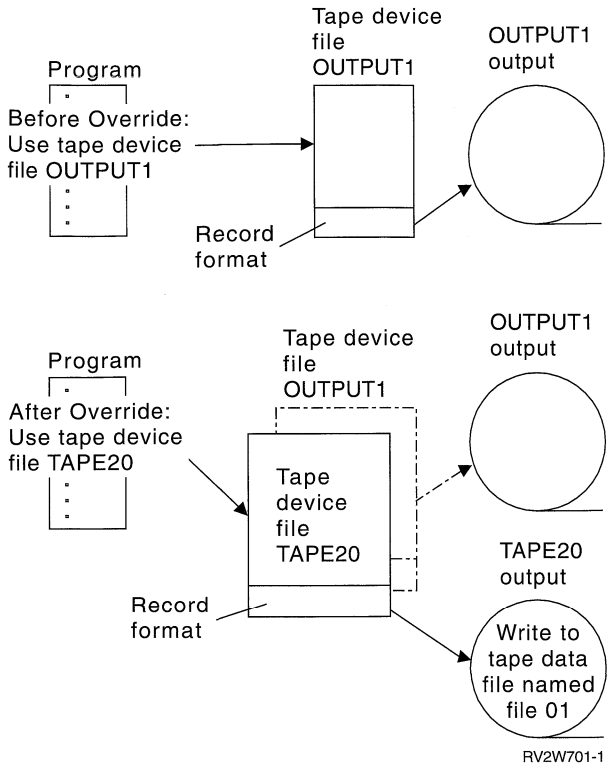


Figure 4-2. Overriding a File Name

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 your file can be overridden with a file that has a different type, see the information under "File Redirection" on page 4-4. More information is available in the *Data Management Guide*.

Overriding Both File Names and File Attributes in HLL Programs

This form of overriding files is simply a combination of overriding file attributes and overriding file names or types. With this form of override, you can override the file that is to be used in a program and you can also override the attributes of the overriding file.

For example, you want the output from your application program directed to the tape file REPORTS instead of the tape file OUTPUT1 (OUTPUT1 is specified in the application program). In addition to having the application program use the tape file REPORTS, you wish to override the density to 6250 bpi. Assume the file REPORTS was created with the following command:

```
CRTTAPF FILE(REPORTS) DEV(TAP01) DENSITY(1600)
```

Before you run the program, type the following command:

```
OVRTAPF FILE(OUTPUT1) TOFILE(REPORTS)
  DENSITY(6250)
```

Then call the application program, and data is written using the tape file REPORTS with a density of 6250 bpi.

This is *not* equal to the following two override commands:

```
Override 1   OVRTAPF  FILE(OUTPUT1)
             TOFILE(REPORTS)
Override 2   OVRTAPF  FILE(REPORTS)
             DENSITY(6250)
```

Only one override is applied for each call level for an open of a particular file, so if you want to override the file that is used by the program and also override the attributes of the overriding file from one call level, you must use a single command. If two overrides are used, override 1 causes the output to be directed to the tape file REPORTS, but override 2 is ignored.

Deleting Overrides

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

If you use the DLTOVR command in an application that either calls or transfers control to other programs, the override may or may not be deleted. More information about deleting overrides in application programs is available in the *Data Management Guide*.

Displaying Overrides

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

If you use the DSPOVR command to display 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 *Data Management Guide*.

File Redirection

File redirection refers to using overrides to change the file name and library or the type of the file to be processed. For example, you can substitute one tape file for another, one diskette file for another, 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 4-6 for rules on how system code processes overrides.

Overriding Files with the Same File Types

When you replace the file that is used in a program with another file of the same type, the new file is processed in the same manner as the original file. If a field level file, or any other file containing externally described data is redirected, it usually is necessary to either specify LVLCHK(*NO) or recompile the program. With level checking 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, the results cannot be predicted.

Overriding Files with Different File Types

If you change to a different type of file, the device-dependent characteristics are ignored and records are read or written sequentially. Some device parameters must be specified in the new device file or the override. Defaults are taken for others. The effect of specific redirection combinations is described later in this section.

Any attributes specified on overrides of a different file type than the final file type are ignored. The parameters SPOOL, SHARE, and SECURE are exceptions to this rule. They are accepted from any override applied to the file, regardless of device type.

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

- File redirection is not recommended for save files.
- Nonsequentially processed database files can be redirected only to another database file or a DDM file.
- Display files and ICF files that use multiple devices (MAXDEV or MAXPGMDEV > 1) can be redirected 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 cr output-only fields. This is necessary because the display file omits these fields from the record buffer in which they are not used, but other file types do not.

Figure 4-3 on page 4-4 summarizes valid file redirections.

Figure 4-3. File Redirections

	From-File					
To-File	**Printer**	**ICF**	**Dis-kette**	**Display**	**Data-base**	**Tape**
Printer	O*	O	O	O	O	O
ICF	O	I/O O I	O	I/O O I	O	O I
Diskette	O	O I	O	O I	O	O I
Display	O	I/O O I	O	I/O O I	O	O I
Database	O	O I	O	O I	O	O I
Tape	O	O I	O	O I	O	O I

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

To use this chart, identify the file type to be overridden in the FROM-FILE columns and the file type overriding 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 and what is ignored 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. Refer to Chapter 3, "Diskette Support" for a description of exchange types.

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

Overrides are not applied to any system files that are opened as part of an end-of-routing step or end-of-job processing. For example, overrides cannot be specified 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 could 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 following commands allow overrides for the SRCFILE and SRCMBR parameters only:

CRTCMD	CRTPF
CRTICFF	CRTPRTF
CRTDSPF	CRTSRCPF
CRTLF	CRTTBL
	CRTXXXPGM

(All create program commands. These commands also use overrides to determine which file will be opened by a compiled program. See *Data Management Guide* 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.)

Appendix A. 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 following information is presented 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
- Length, which is given 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 each item is valid for

For more information about feedback area layouts for files other than tape and diskette files, refer to the *Data Management Guide*.

The support provided by the high-level language you are using determines how to access this information and how the data types are represented. 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 it has been opened. It also contains file-specific information, depending on the file type, plus information about each device or communications session defined for the file. This information is set during open processing and may be updated as other operations are performed.

Figure A-1 (Page 1 of 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 file not being spooled, or diskette file 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
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.	

Figure A-1 (Page 2 of 4). Open Feedback Area

Offset	Data Type	Length	Contents	File Type
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. Bit 2: File shareable 0 File was not opened shareable. 1 File was opened shareable (SHARE(*YES)). Bit 3: Not applicable to tape and diskette. Bit 4: Not applicable to tape and diskette. Bit 5: Not applicable to tape and diskette.	Tape and diskette

Figure A-1 (Page 3 of 4). Open Feedback Area

Offset	Data Type	Length	Contents	File Type
			Bit 6: Field-level descriptions	Tape and diskette
			This is always 0 for tape and diskette.	
			Bit 7: DBCS-capable file	Tape and diskette
			0 File is not DBCS-capable. 1 File is DBCS-capable.	
			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	Tape and diskette
			0 System creates I/O buffers for the program. 1 User program supplies I/O buffers.	
			Bits 7: Reserved.	
			Bits 8: Not applicable to tape and diskette.	
133	Character	2	Open identifier. The value is unique for a full (non-shared) open operation of a file. 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
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.	

Figure A-1 (Page 4 of 4). Open Feedback Area

Offset	Data Type	Length	Contents	File Type
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" on page A-4 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 attached to the file. The number of entries in this array is determined by the number at offset 146 of the open feedback area. 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.

Figure A-2 (Page 1 of 2). 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

Figure A-2 (Page 2 of 2). Device Definition List

Offset	Data Type	Length	Contents	File Type
71	Character	1	Device type.	
			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 7208-2 Tape Unit	
			hex 5A 6343 Tape Unit	
			hex 5B 6348 Tape Unit	
			hex 5C 6368 Tape Unit	
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.	

Volume Label Fields

Figure A-3. 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

I/O Feedback Area

The results of I/O operations are communicated to the program using Operating System/400* (OS/400*) messages and I/O feedback information. The I/O feedback area is updated for every I/O operation unless your program is using blocked record I/O. In that case, the feedback area is updated only when a block of records is read or written. 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.

Common I/O Feedback Area

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

Figure A-4 (Page 1 of 2). 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 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 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

Figure A-4 (Page 2 of 2). Common I/O Feedback Area

Offset	Data Type	Length	Contents
			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 7208-2 Tape Unit hex 5A 6343 Tape Unit hex 5B 6348 Tape Unit hex 5C 6368 Tape Unit
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.

Bibliography

The following AS/400 manuals contain information you may need. The manuals are listed with their full title and base order number. When these manuals are referred to in this guide, the short title listed is used.

- *Basic Backup and Recovery Guide*, SC41-0036, provides the system programmer with information to plan a backup and recovery strategy. Also included are procedures to implement your backup and recover strategy, how to add disk units to an existing auxiliary storage pool (ASP), and how to recover from disk unit failures.

  Short title: *Basic Backup and Recovery Guide*.

- *Communications: Intersystem Communications Function Programmer's Guide*, SC41-9590 provides the application programmer with the information needed to write application programs that use AS/400 communications and ICF files. It also contains information on data description specifications (DDS) keywords, system-supplied formats, return codes, file transfer support, and program examples.

  Short title: *ICF Programmer's Guide*.

- *Communications: Operating System/400* Communications Configuration Reference*, SC41-0001 provides information about configuration commands and defining lines, controllers, and devices.

  Short title: *OS/400* Communications Configuration Reference*.

- *Data Description Specifications Reference*, SC41-9620 provides the application programmer with detailed descriptions of the entries and keywords needed to describe database files (both logical and physical) and certain device files (for displays, printers, and ICF) external to the user's programs.

  Short title: *DDS Reference*.

- *Data Management Guide*, SC41-9658 provides the application programmer with information about using files in application programs. It describes fundamental structure and concepts of data management support on the AS/400 system, copying files, and temporarily changing files when running an application program through overrides and file redirection.

  Short title: *Data Management Guide*.

- *Database Guide*, SC41-9659 provides the application programmer or system programmer with a detailed discussion of the AS/400 database organization, including information on how to create, describe, and manipulate database files on the system.

  Short title: *Database Guide*.

- *Device Configuration Guide*, SC41-8106 provides the system operator or system administrator with information on how to do an initial local hardware configuration and how to change that configuration. It also contains conceptual information for device configuration, and planning information for device configuration on the 9406, 9404, and 9402 System Units.

  Short title: *Device Configuration Guide*.

- *Distributed Data Management Guide*, SC41-9600 provides the application programmer or system programmer with information about remote file processing. It describes how to define a remote file to OS/400 DDM (distributed data management), how to create a DDM file, what file utilities are supported through DDM, and the requirements of OS/400 DDM as related to other systems.

  Short title: *DDM Guide*.

- *Guide to Programming Application and Help Displays*, SC41-0011 provides information about creating and maintaining screens for applications, creating online help information, and working with display files on the AS/400 system.

  Short title: *Guide to Programming Displays*.

- *Guide to Programming for Printing*, SC41-8194 provides information on how to understand and control printing: printing elements and concepts, printer file support, print spooling support, printer connectivity, advanced function printing, and printing with personal computers.

  Short title: *Guide to Programming for Printing*.

- *Programming: Control Language Programmer's Guide*, SC41-8077 provides a wide-ranging discussion of programming topics, including a general discussion of objects and libraries, control language (CL) programming, controlling flow and communicating between programs, working with objects in CL programs, and creating CL programs. Other topics include predefined and immediate messages and message handling, defining and creating user-defined commands and menus, and application testing, including debug mode, breakpoints, traces, and display functions.

  Short title: *CL Programmer's Guide*.

- *Programming: Control Language Reference*, SC41-0030 provides a description of the control language (CL) and its commands. Each command is defined including its syntax diagram, parameters, default values, and keywords.

  Short title: *CL Reference*.

- *Security Reference*, SC41-8083 provides the system programmer (or someone who is assigned the responsibilities of a security officer) with information about system security concepts, planning for security, and setting up security on the system.

  Short title: *Security Reference*.

- *System Concepts*, GC41-9802 provides the programmer and system user with information about the concepts related to the overall design and use of the AS/400 system and its operating system. This manual includes general information about topics such as user interface, object management, work management, system management, data management, data base, communications, environments, OfficeVision/400*, PC Support/400, and architecture.

  Short title: *System Concepts*.

- *System Operator's Guide*, SC41-8082 provides information about how to use the system unit control panel and console, send and receive messages, respond to error messages, start and stop the system, use control devices, work with program temporary fixes (PTFs), and process and manage jobs on the system.

  Short title: *Operator's Guide*.

Index

Numerics

A

B

C

D

E

F

H

I

L

S

T

U

V

W

Customer Satisfaction Feedback

Application System/400
Guide to Programming
for Tape and Diskette
Version 2

Publication No. SC41-0012-01

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

_____

_____

_____

_____

_____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937+3430**
  Other countries: **(+1)+507+253+5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

_____ _____
Name Address

_____ _____
Company or Organization

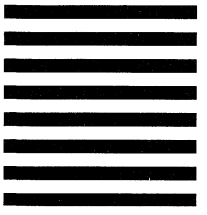
_____ _____
Phone No.

Customer Satisfaction Feedback
SC41-0012-01

Cut o
Along

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape **Please do not staple** Fold and Tape

SC41-0012-01

Cut or
Along

Customer Satisfaction Feedback

Application System/400
Guide to Programming
for Tape and Diskette
Version 2

Publication No. SC41-0012-01

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

_____

_____

_____

_____

_____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937+3430**
  Other countries: **(+1)+507+253+5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Customer Satisfaction Feedback
SC41-0012-01

IBM®

Cut
Alon

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape | **Please do not staple** | Fold and Tape

Cut o
Along

Customer Satisfaction Feedback

Application System/400
Guide to Programming
for Tape and Diskette
Version 2

Publication No. SC41-0012-01

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

_____

_____

_____

_____

_____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937+3430**
  Other countries: **(+1)+507+253+5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Customer Satisfaction Feedback
SC41-0012-01

IBM®

Cut
Alon

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape | **Please do not staple** | Fold and Tape

Cut
Alon

Customer Satisfaction Feedback

Application System/400
Guide to Programming
for Tape and Diskette
Version 2

Publication No. SC41-0012-01

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

_____

_____

_____

_____

_____

May we contact you to discuss your responses? __ Yes __ No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937+3430**
  Other countries: **(+1)+507+253+5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

_____ _____
Name Address

_____ _____
Company or Organization

_____ _____
Phone No.

Customer Satisfaction Feedback
SC41-0012-01

IBM®

Cut o
Along

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 245
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-7899

Fold and Tape **Please do not staple** Fold and Tape

SC41-0012-01

Cut o
Along

Program Number: 5738-SS1

Printed in Ireland by Printech International plc

SC41-0012-01