

BARR/TRAN

*Advanced file transfer extension to Barr RJE
communications products*

*Documentation Edition 4
Software Version 9902 or later*

Editors

Alice T. Allen
Kay Ramey Barr

Programmers

Al Hodges
Chuck Logan

Copyright © 1988-2004 by Barr Systems, Inc.
All Rights Reserved.



Barr Systems, Inc.
4500 NW 27 Avenue
Gainesville, FL 32606-7031

Phone: 800-BARR-SYS
 800-227-7797
 352-491-3100

Fax: 352-491-3141

E-mail: support@barrsystems.com
Web site: www.barrsystems.com
FTP site: ftp.barrsystems.com

The Barr logo is a registered trademark and BARR/3780, BARR/HASP, BARR/RJE, BARR/TAPE, BARR/TRAN, RJE+3270, and Communication Scope are trademarks of Barr Systems, Inc. IBM is a registered trademark and PC DOS, System/36, and AS/400 are trademarks of International Business Machines Corp. All other trademarks are the sole property of their respective owners.

August 24, 2004

Preface

The BARR/TRAN product provides extended file transfer capabilities to Barr Systems' RJE products – for example, BARR/RJE, BARR/HASP, and BARR/3780. BARR/TRAN is a mainframe program for IBM MVS, MVS/XA, MVS/ESA, and VS1/RES operating systems that works with advanced features of Barr RJE software. Although Barr RJE software manages basic file transfer operations, BARR/TRAN manages all file transfer operations.

Before installing BARR/TRAN, you will need to be familiar with the operation of your Barr RJE software. Please refer to your Barr RJE manual for installation, operating instructions, and computer-to-computer file transfer that does not require BARR/TRAN.

To simplify the subject of file transfer, the early chapters of this manual introduce new vocabulary that you will need to understand for your file transfer applications. We recommend that you read the entire manual from start to finish before you begin. Then go back and match up your applications with the examples in Chapters 4 and 5.

BARR/TRAN also works with BARR/NJE.

Barr Technical Support

Contact Barr Technical Support at 800-BARR-SYS Monday through Friday between 9 a.m. and 8 p.m. eastern time if you have questions or problems with Barr hardware or software. Technical Support will ask for your adapter serial number and software version number. When you call, please have this information on hand.

If you contact Technical Support by fax, e-mail, or the Web site support page, please include the serial number and software version number in your correspondence.

You can obtain the adapter serial number from any of these places:

- Label on the adapter box
- Blue sticker on the adapter edge that is visible from the rear of the computer
- Back of the adapter, where it is handwritten

References

Barr Publications

BARR/RJE

BARR/HASP

BARR/3780

BARR/TAPE

IBM Publications

MVS/Extended Architecture Magnetic Tape Labels and File Structure Administration (GC26-4003)

MVS/Extended Architecture Data Administration Guide (GC26-4013)

MVS/Extended Architecture System-Data Administration (GC26-4010)

MVS/Extended Architecture JCL Reference (GC28-1352)

MVS/Extended Architecture Message Library: System Codes (GC28-1157)

MVS/Extended Architecture Message Library: System Messages; Volume 1 (GC28-1376), Volume 2 (GC28-1377)

Contents

Preface	iii
1. Introduction	1
1.1 Barr RJE Software	1
1.2 BARR/TRAN Software	1
1.3 Diagram of Barr File Transfer Methods	2
2. File Structure	3
2.1 PC DOS File Formats	4
2.2 IBM MVS File Structure	6
2.3 Barr File Transfer (BFT) Format	7
3. Transformation Rules	9
3.1 Character Set Translation Rules	9
3.2 Record Length Conversion Rules	9
3.3 Carriage Control Standards	10
4. Computer-to-Mainframe File Transfer	11
4.1 Send a Computer File	11
4.2 Print a Computer File on a Mainframe Printer	13
4.3 Send a Computer Print File to an MVS Data Set with Variable-Length Records ..	14
4.4 Send a Computer Print File to an MVS Data Set with Fixed-Length Records	15
4.5 Send a Computer Binary File to an MVS Data Set	15
4.6 Send a Mainframe Object Module from Computer to Mainframe	16
5. Mainframe-to-Computer File Transfer	17
5.1 Receive a File	17
5.2 Receive an MVS Printer Data Set to the Computer	19
5.3 Receive an MVS Data Set with Variable-Length Records	20
5.4 Receive a Variable-Length MVS Data Set to the Computer with Fixed-Length Records	21
5.5 Receive an MVS Binary File to a Computer Binary File	21
5.6 Receive an MVS Object Deck to a Computer Binary File	22

6. Mainframe Installation	23
6.1 Edit INSTALL.JOB	23
6.2 Run INSTALL.JOB.....	26
A. Barr File Transfer (BFT) Format.....	27
A.1 Barr File Transfer (BFT) Format Records.....	27
A.2 Consistency Checking	35
A.3 Changing a File's Format.....	35
Index	39

1. Introduction

You can send, receive, and change the structure of files with Barr RJE (and NJE) and BARR/TRAN software. File transfer involves sending and receiving files between the computer and the mainframe. This process includes physically sending the file as well as reconciling differences between the computer file structure (PC DOS, MS-DOS) and mainframe file structures (for example, IBM MVS, VS1/RES).

1.1 Barr RJE Software

Barr RJE software—for example, BARR/RJE, BARR/HASP, and BARR/3780—already contains file transfer software required for the computer. See your Barr RJE manual for these common file transfer operations. Without additional software, you can:

- Send files of 80-character records and multiples of 80
- Receive files with record lengths of 1-255 characters and multiples of 80 characters
- Receive print files
- Send and receive binary data
- Perform computer-to-computer file transfer

1.2 BARR/TRAN Software

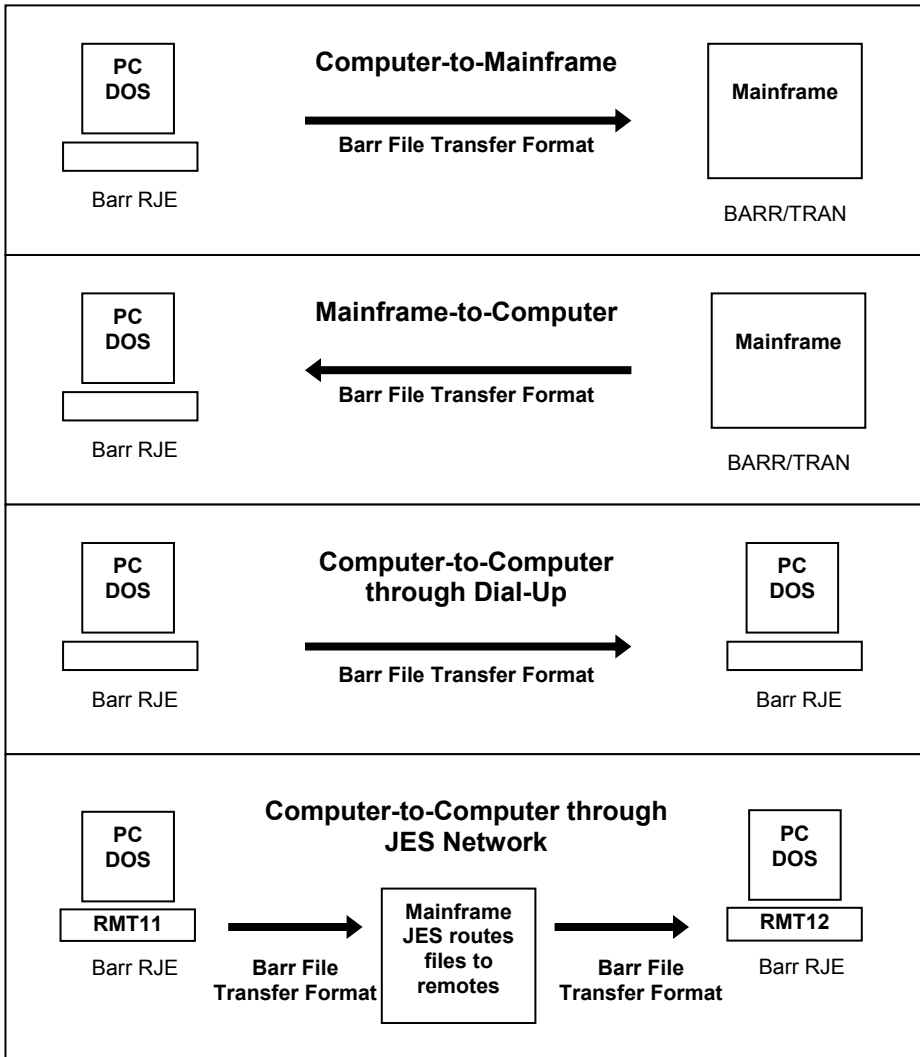
BARR/TRAN extended file transfer software contains the BARR/TRAN mainframe program that handles specialized file transfer applications. BARR/TRAN reads and writes IBM MVS files and converts the files for transmission. In other words, BARR/TRAN can manage all file transfer operations on the mainframe.

Some file transfer applications requiring BARR/TRAN are:

- Sending records longer than 80 characters
- Sending variable-length records
- Sending a file to be printed on the mainframe with form feed and overprint
- Naming files on the destination computer
- Restructuring the file—for example, converting between variable-length and fixed-length records, converting between a file with carriage control to a file without carriage control, and changing record length

- Ensuring that the length of a binary file is the same on the computer as it is on the mainframe
- Deleting the trailer lines sometimes sent by JES

1.3 Diagram of Barr File Transfer Methods



2. File Structure

This chapter defines the file structures used when you transfer files between the PC DOS and mainframe MVS operating systems.

File Formats

BARR/TRAN file transfer actually involves three different file structures (or formats), namely PC DOS, IBM MVS, and Barr File Transfer (BFT) format. Table 2-1 describes the different file formats.

TABLE 2-1
File Formats

File Structure	Description
PC DOS	File structure used on the computer.
IBM MVS	File structure used on the IBM mainframe.
Barr File Transfer format	A general file structure, which is easily converted into PC DOS or IBM MVS file structure. This intermediate format is used when <i>any</i> file is sent or received.

Record Formats

Within each file structure, there are different methods for representing records. The term record format (RECFM) describes these different methods. Each record format is defined by character set, length format, and carriage control. Table 2-2 describes record format attributes.

TABLE 2-2
Record Format Attributes

Attributes	Description
Character set	Representation of graphic symbols as binary bits.
Length format	Fixed, variable, undefined length.
Carriage control	Vertical spacing control.

Kinds of File Structures

File structures for PC DOS, IBM MVS, and Barr File Transfer (BFT) format are shown in Table 2-3.

TABLE 2-3
File Format Attributes

Attributes	Formats		
	PC DOS	IBM MVS	BFT
Character set	ASCII, binary	EBCDIC, binary	ASCII, EBCDIC, binary
Length format	Loosely defined by ASCII carriage return characters	Data Control Block fixed, variable, or undefined	Fixed, variable, undefined
Carriage control	ASCII form feed and line feed	ASA carriage control	ASA carriage control

Because PC DOS and mainframe file structures are very different, a file's structure must be changed when it moves between the two types of machines. BARR/TRAN converts and transfers the file at the same time.

Barr RJE software converts from PC DOS file structure to BFT format when it sends a file and converts from BFT format to PC DOS when it receives a file. The BARR/TRAN mainframe program converts from MVS file structure to BFT format when it sends a file and converts from BFT format to MVS file structure when it receives a file.

Sections 2.1 through 2.3 describe file structures for the PC DOS, IBM MVS, and BFT formats. Chapter 3 explains transformation rules.

2.1 PC DOS File Formats

PC DOS uses the ASCII standard for coding character data as documented in Appendix B of your Barr RJE software manual. Unfortunately, PC DOS does not define file structures or record formats. In this section, we assign names to and describe commonly used PC DOS record formats. You can use these eight record formats with the BARR/TRAN program.

TP Printer Data in ASCII

TP record format describes a print file. It contains printable characters of varying length lines. Each line is terminated by a carriage return (**CR**), which can be followed by line feed

(**LF**) and form feed (**FF**) characters. This popular format is used by all microcomputer and minicomputer operating systems.

TP format is the same as the ASCII Send and Receive modes described in the “Software Installation” and “Operation” chapters of your Barr RJE software manual.

TA ASCII File with No Record Structure

This file does not contain any record delimiters such as line feed (**LF**) and form feed (**FF**) characters.

TFn Fixed-Length ASCII Records

TFn records contain ASCII printable characters and have a fixed length of **n**. There are no delimiters between the records. This file might not be printable because there are no **CR** or **LF** characters to advance the printer to a new line.

TDFn Fixed-Length ASCII Records with Delimiters

TDFn records contain ASCII printable characters and have a fixed length of **n**. Each record is terminated with a **CR LF**. A file in **TDFn** format can be printed, if the length of the record **n** is not greater than the printer’s carriage width.

TV Variable-Length ASCII Records with Delimiters

Each record is terminated with a **CR LF**. A file in **TV** format can be printed, if the maximum length of the record is not greater than the printer’s carriage width.

TB Binary Data with No Record Structure

TB format is an unstructured file without ASCII or EBCDIC code structure or record structure. The only attribute transferred with the file is the file length.

TBFn Fixed-Length Binary Records

TBFn contains binary records without ASCII or EBCDIC code structure that have a fixed length of **n**.

TBV Variable-Length Binary Records

A file in **TBV** format begins with a four-byte header of **76 1A LSB MSB**, where **LSB** and **MSB** represent the maximum record length expressed in binary with the least significant byte first. Additionally, each record in the file must begin and end with its binary length count. If the maximum record length in the header is less than 255, the record length code is one byte. Otherwise, it is two bytes, with the least significant byte given first.

2.2 IBM MVS File Structure

IBM MVS file structure is completely described in the JCL Data Control Block (**DCB**) specification, which has the general form:

DCB= (RECFM=r , LRECL=1 , BLKSIZE=b)

These parameters are briefly described here.

RECFM=r

The record format (**RECFM**) is determined by three characteristics – record type, blocking, and carriage control – which are specified in your JCL after the **RECFM=** keyword.

- F** Fixed-length records.
- V** Variable-length records.
- U** Undefined-length records.
- B** Blocking combines records into blocks before reading or writing.
- A** The first character of a record contains an ASA carriage control character. The standard list of carriage control characters is shown in Chapter 3 of this manual.
- S** Spanning allows variable-length records to exceed the length of the block.

LRECL=1

Logical Record Length is a number computed by:

$$l = d + a + v$$

where **d** is data length; **a** is 1 if ASA carriage control is used, otherwise 0; and **v** is 4 if the records are variable length, otherwise 0.

BLKSIZE=b

Block size is the length of the maximum block of data that will be read or written at one time. For fixed-length records, **BLKSIZE** is a multiple of the logical record size **LRECL**.

Example

This is a common format for a mainframe print file:

DCB= (RECFM=FBA , LRECL=133 , BLKSIZE=1330)

This statement indicates that the record format is fixed with 10 records to a block (1330 divided by 133) and contains ASA carriage control characters. The logical record length is

133, indicating that there are 132 data characters preceded by an ASA carriage control character.

For more information about these parameters, see your MVS JCL reference manual.

2.3 Barr File Transfer (BFT) Format

When you transfer a file from computer to mainframe, mainframe to computer, or computer to computer, your Barr RJE software converts the file to an intermediate format—the Barr File Transfer (BFT) format. On the mainframe, the BARR/TRAN program reads and writes MVS data sets and transforms them into BFT format.

In BFT format, each record is 80 characters long, and records are numbered sequentially.

BFT format contains three types of records—header records, data records, and end-of-file records. The header record is sent at the start of a file followed by the data records and the end-of-file record.

Header Record

All the attributes of the file are specified by the header record, including coding (ASCII, EBCDIC, or binary), file name, length format (fixed-length, variable-length), and whether ASA carriage control is present.

Data Records

Each data record contains a length specification followed by the record information.

End-of-File Record

The end-of-file record contains the number of cards and the number of records.

Notes:

3. Transformation Rules

BARR/TRAN automatically converts files according to transformation rules.

You use BARR/TRAN when you move a file from a source computer to a destination computer, for example, from computer to mainframe. The transformation rules described in this chapter document the conversion processes that BARR/TRAN automatically performs. This is how the process works:

1. The file is rewritten from the PC DOS file structure into Barr File Transfer (BFT) format. (Any computer or MVS file can be directly converted to the intermediate BFT format.)
2. The file is physically sent as a series of 80-character records.
3. Using the transformation rules, the file is converted from the source computer's record format to the destination computer's record format.
4. The file is rewritten from the BFT format into the IBM MVS file structure.

Transformation rules cover character set translation, record length conversion, and the treatment of printer carriage control.

3.1 Character Set Translation Rules

Barr RJE and BARR/TRAN software translate character sets according to these rules:

- If either the source or the destination character set is binary, no translation will be performed.
- Either EBCDIC-to-ASCII or ASCII-to-EBCDIC translation will be performed, if necessary.
- ASCII tab characters are left as tabs by default. To expand tab characters into blanks according to the computer standard of eight-character columns, set the **Expand FILE80 tab characters?** option to **YES** on the **Tuning - Reader Control** screen.

3.2 Record Length Conversion Rules

Barr RJE and BARR/TRAN software change record lengths according to these rules:

- Records longer than the destination length will be truncated.
- Records shorter than the destination length will be padded to the required length. ASCII and EBCDIC formats use the blank character as a pad character while binary format uses binary zero as the pad character.

3.3 Carriage Control Standards

BFT format uses the ASA carriage control character standard. If the computer is sending a printer file, the ASA character is created from the ASCII characters **FF**, **CR**, and **LF**. If the computer is receiving a print file, this process is reversed. ASA carriage control characters are defined by the standards in Table 3-1.

TABLE 3-1
Carriage Control Character Actions

Carriage Control Character	Action of characters before line is printed
(blank)	Space 1 line
0	Space 2 lines
-	Space 3 lines
+	Suppress space (overprint line)
1	Skip to stop 1 (usually advance to new page)
2	Skip to stop 2 (channel 2)
3	Skip to stop 3 (channel 3)
4	Skip to stop 4 (channel 4)
5	Skip to stop 5 (channel 5)
6	Skip to stop 6 (channel 6)
7	Skip to stop 7 (channel 7)
8	Skip to stop 8 (channel 8)
9	Skip to stop 9 (channel 9)
A	Skip to stop A (channel 10)
B	Skip to stop B (channel 11)
C	Skip to stop C (channel 12)

4. Computer-to-Mainframe File Transfer

This chapter describes how to send a file from the computer to the mainframe using Barr RJE software on the computer and BARR/TRAN on the mainframe.

When you send a file from the computer to the mainframe, you need to specify the computer file name and record format with the Barr RJE software **Send Files** command. You also need to specify the mainframe file name and record format with mainframe JCL.

In the **Send Files** command, for **pc_filename** substitute the computer file and for **format** substitute one of the eight computer formats (**TP**, **TA**, **TFn**, **TDFn**, **TV**, **TB**, **TBFn**, or **TBV**). In the JCL **FILEOUT DD** statement, supply the MVS file name with the **DSNAME=data_set_name** parameter and the record format with the **DCB** parameter, as follows:

```
DCB=(RECFM=r , LRECL=1 , BLKSIZE=b)
```

Section 4.1 lists general file transfer instructions. Sections 4.2 through 4.6 provide detailed examples.

4.1 Send a Computer File

Follow these general steps to send a file from the computer to the mainframe.

1. Write the mainframe JCL statements for your application in a file on the computer.
2. Log on to the mainframe with the Barr RJE software.
3. Use the **Send Files** command to send two files to the mainframe – the JCL file and the computer file (in one of the computer record formats described in Chapter 2) you want to transfer.

See your Barr RJE software manual for more information about the **Send Files** command. BARR/TRAN converts the file from the PC DOS file structure to the MVS file structure you specified in the JCL. You will be notified whether or not the file has been successfully received.

For example, assume you want to send a computer file named **pc_filename** with one of the computer record formats to the mainframe.

Mainframe Steps

On the mainframe, specify the MVS data with

```
DSNAME=data_set_name
```

and the record format with

```
DCB=(RECFM=r,LRECL=1,BLKSIZE=b)
```

Include these parameters in the **FILEOUT DD** statement.

```
//jobname JOB (1234,5678,2,1,9999),'your name'  
//      EXEC BARRTRAN  
//FILEOUT DD DSNAME=data_set_name,DCB=(RECFM=r,LRECL=1,BLKSIZE=b),  
//          UNIT=SYSDA,SPACE=(8000,(10,10),RLSE),DISP=(,CATLG)  
//SYSIN   DD *
```

Computer Steps

After you create a JCL file on the computer, log on to the mainframe with the Barr RJE software. Then use the **Send Files** command to send these two files to the mainframe:

- The file containing mainframe JCL
- The computer file you want to transfer

You will send the mainframe JCL file to the mainframe in front of the computer file. On the Barr software **Send Files** command line, you will substitute the name of the mainframe JCL file for **jcl_filename**.

1. From the Barr **Operation** screen, select **Send Files to RD1**.

```
Enter file names. To combine files, separate by a plus (+).  
To start a new file, leave a blank.  
To cancel, press Esc.  
Files: jcl_filename+pc filename/format
```

2. Enter the JCL file name and plus sign (+) followed by the computer file name, a slash (/), and the format code.

The job will run on the mainframe, and the BARR/TRAN job log will come back to the computer. If the condition code on the JES job log is 0000, your file was received correctly.

This is a sample BARR/TRAN job log:

```
***** BARR / TRAN JOB LOG *****  
  
BARR/TRAN Release 89  
  
OPTIONS IN EFFECT FOR THIS RUN:  
  FFILE=      0 FRECORD=  0 LFILE=  0 LRECORD=0 BUFLLEN=32,760  
  IFORMAT=AUTO OFORMAT=AUTO SKIP= 100 CONVERT=  
  RECFM=      RECFM=      0 BLKSIZE=0 FILENAME=  
  
BARR/TRAN INPUT IS SYSIN:  
  CARD DECK IN FILE-80 FORMAT, ENCODING:  
  ASCII FILE pc_filename (OPENED IN RECORD MODE)  
  WITH FORMAT=TP, A MAXIMUM RECORD LENGTH OF 32,760 BYTES, AND RECORDS  
  WITH ASA CARRIAGE CONTROL.  
  
BARR/TRAN OUTPUT IS FILEOUT:  
  EBCDIC FILE JES2.JOB03556.S0000103 (OPENED IN RECORD MODE)  
  WITH FORMAT=FBA, A MAXIMUM RECORD LENGTH OF 132 BYTES, AND RECORDS  
  WITH ASA CARRIAGE CONTROL.  
  THE FILE HAS AN OS BLOCK SIZE OF 5,320 BYTES.  
  
FINISHED READING INPUT. INPUT STATISTICS:  
  40 CARDS WERE READ; OF THESE, 0 WERE BAD CARDS SKIPPED AT THE START;  
  25 RECORDS OR BLOCKS WERE READ, AND 0 FILE MARKS WERE READ.  
  THE LARGEST RECORD OR BLOCK WAS 78 BYTES LONG; THE SMALLEST RECORD  
  OR BLOCK WAS 0 BYTES LONG.  
  
FINISHED WRITING OUTPUT. OUTPUT STATISTICS:  
  25 RECORDS OR BLOCKS WERE WRITTEN, AND 0 FILE MARKS WERE WRITTEN.  
  THE LARGEST RECORD OR BLOCK WAS 132 BYTES LONG; THE SMALLEST RECORD  
  OR BLOCK WAS 132 BYTES LONG.  
  
**** NOTE **** RECORDS WERE PADDED TO MAKE FIXED-LENGTH OUTPUT  
RECORDS.  
  
***** END OF BARR / TRAN LOG *****
```

4.2 Print a Computer File on a Mainframe Printer

You can copy a normal text file (possibly with form feeds and overstruck lines) from the computer to the mainframe and print it on a mainframe printer. In this example, you need to specify the **TP** format in the Barr **Send Files** command:

```
jcl_filename+pc_filename/TP
```

You also need the following JCL:

```
//PC1      JOB (1234,5678,2,1,9999),'your name'  
/*ROUTE PRINT LOCAL  
//        EXEC BARRTRAN  
//FILEOUT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133)  
//SYSIN   DD *
```

BARR/TRAN reads from **SYSIN**, which contains the computer file, and writes to **FILEOUT**. Since **FILEOUT** is defined as a **SYSOUT** data set, the data is printed. The **/*ROUTE** statement sends the data to the local printer at the mainframe (instead of sending it back to Barr RJE software).

4.3 Send a Computer Print File to an MVS Data Set with Variable-Length Records

You can copy a print file (possibly with form feeds and overstruck lines) from the computer to an MVS data set with variable-length records. In this example, you need to specify the **TP** format in the Barr **Send Files** command:

```
jcl_filename+pc_filename/TP
```

You also need the following JCL:

```
//PC2      JOB (1234,5678,2,1,9999),'your name'  
//        EXEC BARRTRAN  
//FILEOUT DD UNIT=SYSDA,DSNAME=data_set_name,DISP=(NEW,CATLG),  
//        SPACE=(8000,(10,10),RLSE),  
//        DCB=(RECFM=VBA,LRECL=137,BLKSIZE=13704)  
//SYSIN   DD *
```

BARR/TRAN reads input from **SYSIN** and writes to the **FILEOUT** data set name.

4.4 Send a Computer Print File to an MVS Data Set with Fixed-Length Records

You can copy a print file from the computer to an MVS data set with fixed-length records. This example is similar to the one in Section 4.3, except that all records written to the output data set are automatically padded with EBCDIC spaces to the record length of the output data set.

In this example, you need to specify the **TP** format in the Barr **Send Files** command:

```
jcl_filename+pc_filename/TP
```

You also need the following JCL:

```
//PC3    JOB (1234,5678,2,1,9999),'your name'  
//      EXEC BARRTRAN  
//FILEOUT DD UNIT=SYSDA,DSNAME=data_set_name,DISP=(NEW,CATLG),  
//          SPACE=(8000,(10,10),RLSE),  
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=13300)  
//SYSIN  DD *
```

BARR/TRAN reads input from **SYSIN** and writes to the **FILEOUT** data set name.

4.5 Send a Computer Binary File to an MVS Data Set

You can copy a binary file from the computer to an MVS data set with undefined-format records. In this example, you need to specify the **TB** format in the Barr **Send Files** command:

```
jcl_filename+pc_filename/TB
```

You also need the following JCL:

```
//PC4    JOB (1234,5678,2,1,9999),'your name'  
//      EXEC BARRTRAN  
//FILEOUT DD UNIT=SYSDA,DSNAME=data_set_name,DISP=(NEW,CATLG),  
//          DCB=(RECFM=U,BLKSIZE=2000),SPACE=(8000,(10,10),RLSE)  
//SYSIN  DD *
```

RECFM=U indicates that the data set's record format is undefined. Because the computer file format is binary, no ASCII-to-EBCDIC translation is performed. This file is written as 2000-byte output records with a final short block.

4.6 Send a Mainframe Object Module from Computer to Mainframe

You can copy a mainframe object module stored as a binary computer file back to the mainframe. For example, the BARR/TRAN program is stored as an object module on the disk you received from Barr Systems.

In this example, you need to specify the **TBF80** format in the Barr **Send Files** command:

```
jcl_filename+pc_filename/TBF80
```

You also need the following JCL:

```
//PC5      JOB (1234,5678,2,1,9999),'your name'  
//        EXEC BARRTRAN  
//FILEOUT DD UNIT=SYSDA,DSNAME=*&TEMP,DISP=(NEW,PASS),  
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),  
//        SPACE=(8000,(10,10),RLSE)  
//SYSIN   DD *
```

FILEOUT defines a temporary disk data set that could later be used for input to the linkage editor.

5. Mainframe-to-Computer File Transfer

This chapter describes how to receive a file from the mainframe to the computer using BARR/TRAN on the mainframe and Barr RJE software on the computer.

When you receive a mainframe file to the computer, you need to specify the computer file name and record format and the mainframe file name and record format with mainframe JCL.

You need to specify the computer file name with the **FILENAME=pc_filename** parameter and the computer record format with the **RECFM=pc_format** parameter. (The computer formats are **TP**, **TA**, **TFn**, **TDFn**, **TV**, **TB**, **TBFn**, and **TBV**). You also need to specify the MVS file name with the **DSNAME=data_set_name** parameter. Include these parameters in the **FILEIN DD** statement.

Section 5.1 lists general file transfer instructions. Sections 5.2 through 5.6 provide detailed examples.

5.1 Receive a File

Follow these steps to receive a file from the mainframe to the computer.

1. Write the mainframe JCL statements for your application in a file on the computer. The statements specify that BARR/TRAN is to convert an MVS file to a file with one of the computer formats.
2. From the Barr RJE software **Assign Devices** screen, assign the punch device (for example, **PU1**) to a subdirectory on the computer and set the receive mode to **Transfer files**.
3. Log on to the mainframe with the Barr RJE software.
4. Use the **Send Files** command to send the JCL file to the mainframe.

See your Barr RJE software manual for more information about the **Send Files** command. BARR/TRAN transforms the file and sends it to the remote computer.

Assume you want to receive a file from the mainframe and write it to a file named **pc_filename** in one of the record formats described in Chapter 2.

Software Setup

To set up your Barr RJE software for file transfer, perform these steps. Start at the **Installation Description** screen.

1. Select **Assign Devices**.
2. Select **PU1** as the source device.
3. Select **(FILE)** as the destination device.
4. Leave **Beginning of file name** blank or enter a path followed by a backslash (\).
5. Select **Receive Mode**.
6. Select **Transfer files**.

Mainframe Steps

You need to run a job on the mainframe that will use BARR/TRAN to read the mainframe data set and route it on the computer as a file on the **PU1** stream. In the JCL, you need to specify the computer file format and file name with the **RECFM=pc_format** and **FILENAME=pc_filename** parameters. You also need to specify the mainframe MVS data set name with the **DSNAME=data_set_name** parameter.

Use the following JCL:

```
//jobname JOB (1234,5678,2,1,9999),'your name'  
//          EXEC BARRTRAN,PARM='RECFM=pc_format,FILENAME=pc_filename'  
//FILEIN   DD DSNAME=data_set_name,DISP=SHR
```

Computer Steps

After you create a JCL file on the computer, log on to the mainframe with the Barr RJE software. Then use the **Send Files** command to send the JCL file to the mainframe. The JCL file executes BARR/TRAN, which transforms the mainframe data set, and sends it to the **PU1** device that you specified in the Barr RJE software.

Follow these steps to send the JCL file to the mainframe.

1. From the Barr **Operation** screen, select **Send Files to RD1**.
2. Type the name of the JCL file.
3. Press ENTER.

If the condition code on the JES job log is 0000, your file was received correctly.

This is a sample BARR/TRAN job log:

```

      * * * * B A R R / T R A N   J O B   L O G   * * * *
BARR/TRAN  Release 89
OPTIONS IN EFFECT FOR THIS RUN:
FFILE=      0  FRECORD=  0  LFILE=      0  LRECORD  0  BUFLLEN=32,760
IFORMAT=AUTO  OFORMAT=AUTO_ SKIP=    100  CONVERT=
RECFM=      TP  RECLEN=   0  BLKSIZE=   0  FILENAME=pc_filename
BARR/TRAN INPUT IS FILEIN:
EBCDIC FILE data_set_name                (OPENED IN RECORD MODE)
WITH FORMAT=FBA, A MAXIMUM RECORD LENGTH OF 132 BYTES, AND RECORDS
WITH ASA CARRIAGE CONTROL. THE FILE HAS AN OS BLOCK SIZE OF 5,484
BYTES.
BARR/TRAN OUTPUT IS SYSOUT:
CARD DECK IN FILE-80 FORMAT, ENCODING:
EBCDIC FILE pc_filename                    (OPENED IN RECORD MODE)
WITH FORMAT=TP, A MAXIMUM RECORD LENGTH OF 132 BYTES, AND RECORDS
WITH ASA CARRIAGE CONTROL.
FINISHED READING INPUT.  INPUT STATISTICS:
25 RECORDS OR BLOCKS WERE READ, AND 0 FILE MARKS WERE READ.  THE
LARGEST RECORD OR BLOCK WAS 132 BYTES LONG; THE SMALLEST RECORD OR
BLOCK WAS 132 BYTES LONG.
FINISHED WRITING OUTPUT.  OUTPUT STATISTICS:
40 OUTPUT CARDS WERE WRITTEN.
25 RECORDS OR BLOCKS WERE WRITTEN, AND 0 FILE MARKS WERE WRITTEN.
THE LARGEST RECORD OR BLOCK WAS 132 BYTES LONG; THE SMALLEST RECORD
OR BLOCK WAS 132 BYTES LONG.
      * * * * E N D   O F   B A R R / T R A N   L O G   * * * *

```

5.2 Receive an MVS Printer Data Set to the Computer

You can copy a variable-length MVS data set with the format DCB=(RECFM=VBA, LRECL=137, BLKSIZE=8000) and write it to a normal computer print file with form feeds and overstruck lines.

You need the following JCL:

```
//MF1 JOB (1234,5678,2,1,9999),'your name'  
// EXEC BARRTRAN,PARM='RECFM=TP,FILENAME=pc_filename'  
//FILEIN DD DSNAME=data_set_name,DISP=SHR
```

In the **EXEC BARRTRAN** statement, **RECFM=TP** specifies that the computer file format will be ASCII with variable-length records, **CR LF** delimiters, and printer carriage control characters. **FILENAME=pc_filename** specifies the name of the computer file that will be created. BARR/TRAN reads input from **FILEIN**. The Barr RJE software writes the file in **TP** format and names it **pc_filename**.

5.3 Receive an MVS Data Set with Variable-Length Records

You can copy a variable-length MVS data set created with the format **DCB=(RECFM=VBA, LRECL=137, BLKSIZE=8000)** and write it to a normal computer text file without form feeds or overstruck lines.

You need the following JCL:

```
//MF2 JOB (1234,5678,2,1,9999),'your name'  
// EXEC BARRTRAN,PARM='RECFM=TV,FILENAME=pc_filename'  
//FILEIN DD DSNAME=data_set_name,DISP=SHR
```

In the **EXEC BARRTRAN** statement, **RECFM=TV** specifies that the computer file format will be ASCII with variable-length records, **CR LF** delimiters, and no printer carriage control characters. **FILENAME=pc_filename** specifies the name of the computer file that will be created. BARR/TRAN reads input from **FILEIN**. The Barr RJE software writes the file in **TV** format and names it **pc_filename**.

5.4 Receive a Variable-Length MVS Data Set to the Computer with Fixed-Length Records

You can copy a variable-length MVS data set created with the format **DCB=(RECFM=VBA, LRECL=137, BLKSIZE=8000)** and write it to a computer file with fixed-length records. This is the same as the example in Section 5.3, except that the computer file has fixed-length records 70 bytes long.

You need the following JCL:

```
//MF3 JOB (1234,5678,2,1,9999), 'your name'  
// EXEC BARRTRAN, PARM='RECFM=TDF70, FILENAME=pc_filename'  
//FILEIN DD DSNAME=data_set_name, DISP=SHR
```

In the **EXEC BARRTRAN** statement, **RECFM=TDF70** specifies that the computer file format will be ASCII with fixed-length records, **CR LF** delimiters (for editing and printing), no printer carriage control characters, and a record length of 70. (If the trailing number was not specified, the record length of the source, 132 bytes, would have been used.)

Barr RJE software pads records shorter than 70 bytes with ASCII blanks and truncates records longer than 70 bytes.

5.5 Receive an MVS Binary File to a Computer Binary File

You can copy a binary file from the mainframe and write it as a computer file with binary format.

You need the following JCL:

```
//MF4 JOB (1234,5678,2,1,9999), 'your name'  
// EXEC BARRTRAN, PARM='RECFM=TB, FILENAME=pc_filename'  
//FILEIN DD DSNAME=data_set_name, DISP=SHR
```

BARR/TRAN is executed with the **RECFM=TB**, which means the computer file format will be binary file no record structure.

5.6 Receive an MVS Object Deck to a Computer Binary File

You can copy an object deck from the mainframe to the computer, and write it to a binary file with fixed-length records. (Programs on the mainframe are stored in object deck format.)

You need the following JCL:

```
//MF5   JOB (1234,5678,2,1,9999),'your name'  
//      EXEC BARRTRAN,PARM='RECFM=TBF,FILENAME=pc_filename'  
//FILEIN DD DSN=member,DISP=SHR
```

BARR/TRAN is executed with **RECFM=TBF**, which means the computer file format is binary with fixed-length records containing no delimiters. **FILEIN** specifies the member of the object library. Because the computer record length was not specified, the MVS record length is used.

6. Mainframe Installation

Installing BARR/TRAN on the mainframe involves two steps. You need to edit and then run the installation program. The BARR/TRAN software files are included on your Barr software disk. When you install the Barr software, these software files are automatically copied to the C:\BARR\TRAN folder on your computer.

6.1 Edit INSTALL.JOB

The mainframe side of file transfer consists of the BARR/TRAN program that runs under the mainframe operating systems MVS and VS1/RES. You need to create a new library on the mainframe for storing BARR/TRAN. Ask a systems programmer to help you set up a library and add a cataloged procedure, as described in this section.

The INSTALL.JOB file contains the JCL (Job Control Language) required to create a mainframe library and install the BARR/TRAN program. Use a program editor to modify INSTALL.JOB to add to your computing center's accounting and file naming conventions.

Contents of INSTALL.JOB

The text of the INSTALL.JOB file is listed as follows (with an explanation by line number in the following paragraphs). The line numbers are for reference only and do not appear in the actual file.

```

1 //INSTALL JOB (1234,5678,2,1,0),'your name'
2 /**
3 /** INSTALL BARRTRAN ON MAINFRAME
4 /**
5 /** UPLOAD OBJECT DECK FROM PC AND LINK INTO
6 /** A LOAD LIBRARY.
7 /**
8 //LINK EXEC PGM=LINKEDIT,PARM='LIST,MAP'
9 //SYSPRINT DD SYSOUT=A
10 //SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(200,20))
11 //SYSLMOD DD UNIT=SYSDA,DSNAME=user.barrlib,DISP=(NEW,CATLG),
12 // SPACE=(1024,(50,20,1),RLSE)
13 //SYSLIN DD *
14 ##BARRTRAN.OBJ/B
15 SETCODE AC(1)
16 ENTRY NOAUTH
17 NAME BARRTRAN(R)
18 /**
19 /** TRY TO ACCESS BARRTRAN
20 /**
21 //TEST EXEC PGM=BARRTRAN,COND=(0,LT)
22 //STEPLIB DD DSNAME=user.barrlib,DISP=SHR
23 //SYSPRINT DD SYSOUT=A
24 //FILEIN DD DSNAME=user.barrlib(BARRTRAN),DISP=SHR
25 //SYSOUT DD DUMMY,DCB=(RECFM=FB,BLKSIZE=800,LRECL=80)
26 /**
27 /** INSTALL A CATALOGED PROCEDURE FOR BARRTRAN.
28 /**
29 /**
30 //ADD EXEC PGM=IEBGENER,COND=(0,LT)
31 //SYSPRINT DD SYSOUT=A
32 //SYSIN DD DUMMY
33 //SYSUT2 DD DSNAME=SYS1.PROCLIB(BARRTRAN),DISP=SHR
34 //SYSUT1 DD DATA
35 //BARRTRAN PROC
36 /**
37 /** BARRTRAN IS BARR SYSTEMS' FILE TRANSFER PROGRAM BETWEEN
38 /** MAINFRAME AND PC.
39 /**
40 // EXEC PGM=BARRTRAN
41 //STEPLIB DD DSNAME=user.barrlib,DISP=SHR
42 //SYSPRINT DD SYSOUT=A
43 //SYSOUT DD SYSOUT=B
44 /*

```

Explanation by Line Number

Selected lines are described as follows, including all those that require editing. Refer to the line with the corresponding number.

1 The numbers in parentheses are the job accounting parameters, which can vary from system to system. This example contains the account number (1234,5678) and requests two seconds of CPU time, up to 1000 lines of printed output, and no punch output.

2-17 The first step of the job link-edits into a program library the object modules BARR/TRAN sent as card images from the computer.

11-12 The **//SYSLMOD DD** statement defines the program library that is to contain the file transfer programs. Replace **user.barrlib** with the library name you have chosen. As the statement is coded here, a new program library is created on any available disk volume with a block size assigned by the linkage editor and cataloged. Any unused space in the library is released after the job step is finished.

To install the BARR/TRAN program in an existing library, code the **//SYSLMOD** statement:

```
//SYSLMOD DD DSNAME=existing_library.name,DISP=SHR
```

14 **##BARRTRAN.OBJ/B** is an imbedded **Send Files** command. When the Barr software sends the JCL file, this line is replaced with the contents of the BARRTRAN.OBJ file. The **/B** option sends it as a binary file.

16 If BARR/TRAN is installed in an authorized library, replace line 16 with

```
ENTRY AUTH
```

18-25 The second step of the job tests to make sure that BARR/TRAN was properly installed. It runs the BARR/TRAN program, instructs it to read itself as input, and throws away the card output. If the BARR/TRAN log shows no errors, the program is correctly installed.

22 The **//STEPLIB DD** statement tells MVS to look in the library just created for the program BARR/TRAN. As usual, replace **user.barrlib** with the name for your system.

24 Again, replace **user.barrlib** with the right name, but leave **BARRTRAN** in parentheses exactly as it is.

26-44 The third step of the job installs the cataloged procedure **BARRTRAN**.

- 33** Replace **SYS1.PROCLIB** with the name of the procedure library storing the BARRTRAN cataloged procedure. You must be authorized to modify procedure libraries.
- 41** Again, replace **user.barrlib** with the right name.

Using BARR/TRAN with BARR/TAPE

BARR/TRAN is also the mainframe support for the BARR/TAPE product. To use BARR/TAPE to read and write tape labels on the mainframe, specify **SYSIMOD** to be an authorized program library and replace line 16 with **ENTRY AUTH**.

6.2 Run INSTALL.JOB

After you review and edit the INSTALL.JOB file, you need to run it. Follow these steps.

1. Set C:\BARR\TRAN to be the current computer folder.
2. Then log on to the mainframe with the Barr RJE software and run INSTALL.JOB.
3. If you specified **ENTRY AUTH**, a systems programmer or system administrator must use the Authorized Program Facility (APF) to authorize the library containing BARR/TRAN. Skip this step if you did not specify **ENTRY NOAUTH**.

A. Barr File Transfer (BFT) Format

Barr File Transfer (BFT) format represents a file as a series of 80-byte records.

BFT format also includes some information about the source and/or the destination file—an operating system code, a record format, a maximum record length, and a system-specific area. When a file is represented in BFT format, the source system may use these fields to describe the source file's format or to describe the format of the file to create at the destination. When a file is reconstituted from BFT format, the destination system may create a new file in the format suggested by these fields or may ignore them. This appendix refers to the format described by these fields as the file's native format.

A.1 Barr File Transfer (BFT) Format Records

A BFT format file contains four types of records described in this section: Start-of-File (SOF), Start-of-Record (SOR), Continuation-of-Record (COR), and End-of-File (EOF) records.

Start-of-File (SOF) Record

The first record in a BFT format file is a Start-of-File (SOF) record.

TABLE A-1
Start-of-File (SOF) Record Fields

Field	Column	Length	Type	Contents
TYPE	1	1	number	0 = SOF record.
RECNO	2	4	number	Record number of first record.
LENGTH	6	2	number	Maximum logical record length. 0 = file's native record length is unknown but should be less than 32,760 bytes. 1 to 32,760 = native record length for the file not including ASA carriage control. Records might be longer or shorter than this length because of deferred format conversions. A file without record structure must be partitioned into BFT format records for sending. Partitioning can be arbitrary, for example, to maximize packing efficiency.

Field	Column	Length	Type	Contents
CHARSET	8	1	number	1 = binary; 2 = ASCII; 3 = EBCDIC. ASCII and EBCDIC data contain only printable characters and can be converted from one character set to another. Binary data cannot be converted without losing information.
FORMAT	9	1	number	1 = no ASA; 2 = ASA; 3 = no record structure. A file with record structure may or may not contain ASA carriage control. The first byte of ASCII or EBCDIC data may contain a valid ASA carriage control character in the same character set as the rest of the record; binary data may not have carriage control. This ASA byte is counted in the physical record length specified in the LENGTH field of an SOR record; however, the ASA byte is not counted in the logical record length specified in the LENGTH field of an SOF record or in user messages.
SYSTEM	10	1	number	The file's native operating system. 0 = unknown operating system 1 = PC DOS & family 2 = MVS & family This field describes the native operating system of the file; its value determines the format of the other system-specific fields.
RECFM	11	1	The format of the file on its native system. Possible values are given as bit patterns (with the high-order bit on the left), decimal numbers, and JCL mnemonics. See Table A-2 for the legal values for RECFM if SYSTEM=PC DOS. See Table A-3 for the legal values for RECFM if SYSTEM=MVS. The MVS record formats use the EBCDIC character set and record structure. Barr software inputs, but does not output, machine carriage control.
NAME	12	50	ASCII	The file name, left justified and padded with blanks on the right.

Field	Column	Length	Type	Contents
MISC	62	5	System-specific information; this area's format depends on the value of SYSTEM. Binary zeros here should be interpreted as default values for all operating systems.
ATTR	62	1	The DOS directory attribute byte for computer users (see Figure A-1).
TIME	63	2	Time of last change to the file (see Figure A-2).
DATE	65	2	Date of last change to the file (see Figure A-3).
BLKSIZE	62	2	number	Native block size for MVS files. Note that, for mainframe data sets, the physical blocking of the file is not important; the file is always transmitted as logical records. It is a two-byte numeric field, stored with the most significant byte first.
ID	73	8	ASCII	FILE80## record format ID string.

TABLE A-2
SYSTEM=PC DOS Legal Values for the RECFM Field

Binary	Decimal	JCL	Description
0000 1000	8	TB	Binary file with no record structure.
0000 1100	12	TBF	Fixed-length binary records with no delimiters.
0001 0000	16	TV	Variable-length ASCII records; terminated by CR LF (formerly TD).
0001 0001	17	TP	Variable-length ASCII records; terminated by CR LF, may include FF and some records with only a CR.
0001 0100	20	TDF	Fixed-length ASCII records; terminated by CR LF.
0001 1000	24	TA	ASCII file with no record structure (formerly T).
0001 1100	28	TF	Fixed-length ASCII records with no delimiters.

TABLE A-3

SYSTEM=MVS Legal Values for RECFM Field (same as RECFM field in DCB)

Binary	Decimal	JCL	Description
0010 0000	32	D	Variable-length on an ANSI tape
0010 0100	6	DA	ASA carriage control
0010 1000	40	DS	Spanned records
0010 1100	44	DSA	Spanned records, ASA carriage control
0011 0000	48	DB	Blocked records
0011 0100	52	DBA	Blocked records, ASA carriage control
0011 1000	56	DBS	Blocked records, spanned records
0011 1100	60	DBSA	Blocked records, spanned records , ASA carriage control
0100 0000	64	V	Variable-length
0100 0010	66	VM	Machine carriage control
0100 0100	68	VA	ASA carriage control
0100 1000	72	VS	Spanned records
0100 1010	74	VSM	Spanned records, machine carriage control
0100 1100	76	VSA	Spanned records, ASA carriage control
0101 0000	80	VB	Blocked records
0101 0010	82	VBM	Blocked records, machine carriage control
0101 0100	84	VBA	Blocked records, ASA carriage control
0101 1000	88	VBS	Blocked records, spanned records
0101 1010	90	VBSM	Blocked records, spanned records, machine carriage control
0101 1100	92	VBSA	Blocked records, spanned records, ASA carriage control
0110 0000	96	VT	Track overflow
0110 0010	98	VTM	Track overflow, machine carriage control
0110 0100	100	VTA	Track overflow, ASA carriage control

Binary	Decimal	JCL	Description
0111 0000	112	VBT	Blocked records, track overflow
0111 0010	114	VBTM	Blocked records, track overflow, machine carriage control
0111 0100	116	VBTA	Blocked records, track overflow, ASA carriage control
1000 0000	128	F	Fixed-length
1000 0010	130	FM	Machine carriage control
1000 0100	132	FA	ASA carriage control
1000 1000	136	FS	Standard format blocks
1000 1010	138	FSM	Standard, machine carriage control
1000 1100	140	FSA	Standard, ASA carriage control
1001 0000	144	FB	Blocked records
1001 0010	146	FBM	Blocked records, machine carriage control
1001 0100	148	FBA	Blocked records, ASA carriage control
1001 1000	152	FBS	Blocked records, standard
1001 1010	154	FBSM	Blocked records, standard, machine carriage control
1001 1100	156	FBSA	Blocked records, standard, ASA carriage control
1010 0000	160	FT	Track overflow
1010 0010	162	FTM	Track overflow, machine carriage control
1010 0100	164	FTA	Track overflow, ASA carriage control
1011 0000	176	FBT	Blocked records, track overflow
1011 0010	178	FBTM	Blocked records, track overflow, machine carriage control
1011 0100	180	FBTA	Blocked records, track overflow, ASA carriage control
1100 0000	192	U	Undefined-format (variable length)
1100 0010	194	UM	Machine carriage control

Binary	Decimal	JCL	Description
1100 0100	196	UA	ASA carriage control
1110 0000	224	UT	Track overflow
1110 0010	226	UTM	Track overflow, machine carriage control
1110 0100	228	UTA	Track overflow, ASA carriage control

FIGURE A-1
DOS Directory Attribute Byte

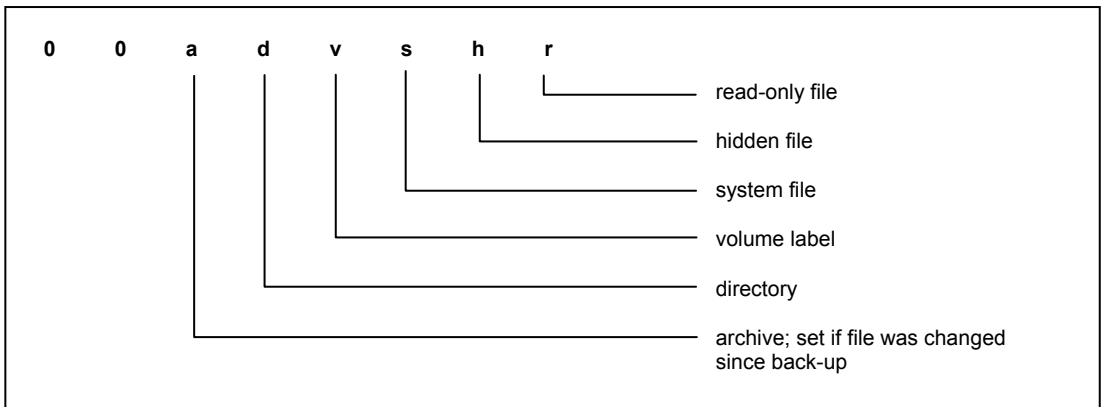


FIGURE A-2
TIME Field Diagram

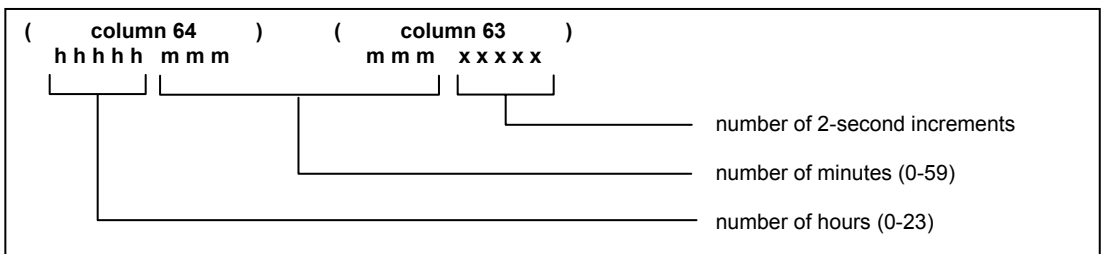
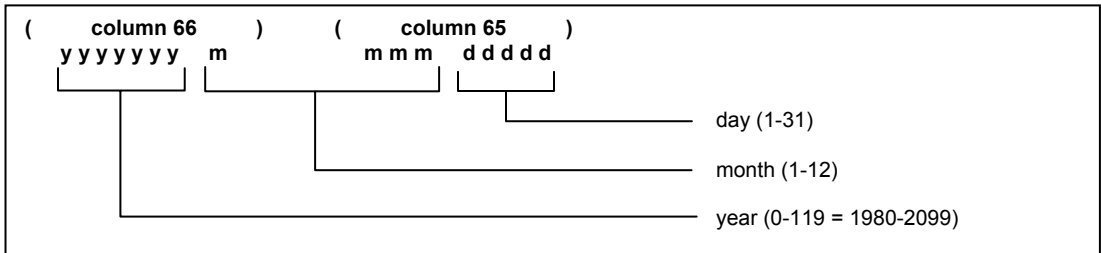


FIGURE A-3
DATE Field Diagram



Start-of-Record (SOR) Record

The SOR record contains descriptor information and data. Continue additional data on Continuation-of-Record (COR) records.

TABLE A-4
Start-of-Record (SOR) Record Fields

Field	Column	Length	Type	Contents
TYPE	1	1	number	1 = (SOR) Start of record.
RECNO	2	4	number	Record number of this record.
LENGTH	6	2	number	Physical record length or number of bytes of data including any ASA carriage control. When a mainframe file with variable-length records is represented in BFT format, the record descriptor words are not included in the logical records.
STEXT	8	73	The first 73 bytes of data from the record, padded with ASCII blanks, if necessary. All the numeric fields contain unsigned binary integers with the most significant byte first. Any columns in a record not used by fields will contain ASCII spaces (blanks).

Continuation-of-Record (COR) Record

As many COR records as needed are used to hold all data that does not fit on the SOR record.

TABLE A-5
Continuation-of-Record (COR) Record Fields

Field	Column	Length	Type	Contents
TYPE	1	1	number	2 = Continuation-of-Record record (COR).
SEQ	2	1	number	Sequence number of this data record. The first one has sequence = 2 (the SOR record has implied sequence 1); the numbers roll over to 0 after they reach 255.
CTEXT	3	78		The next 78 bytes of data from the record, padded with ASCII blanks, if necessary.

End-of-File (EOF) Record

The last record in the stream is the EOF record, which contains checksums for the data sent.

TABLE A-6
End-of-File (EOF) Record Fields

Field	Column	Length	Type	Contents
TYPE	1	1	number	3 = End-of-File record (EOF).
RECNO	2	4	number	Total number of records sent.
LINES	6	4	number	Total number of BFT format records sent, including the SOF and EOF records.
LAST	10	1	number	1 (other values reserved).

A.2 Consistency Checking

When a file in BFT format is received, verify the following:

- All records are in proper format (an exception is JES information at the beginning).
- The record number for the first record in the SOF record matches that in the first SOR record.
- Record numbers for successive records are in order.
- Sequence numbers for continuation records are in order.
- The correct number for continuation records are present for each record.
- Checksums in the EOF record match totals maintained by the receiver.
- Any unnecessary records sent by JES before or after the file should be ignored when data is processed.

A.3 Changing a File's Format

If you are changing the format of a file when converting to or from BFT format, refer to this section.

TABLE A-7
Character Set Conversion

FROM	TO		
	ASCII	EBCDIC	Binary
ASCII	—	Complete ASCII-to-EBCDIC translation	No translation of character set
EBCDIC	Complete ASCII-to-EBCDIC translation	—	No translation of character set
Binary	No translation of character set	No translation of character set	—

TABLE A-8
Record Structure Conversion

FROM	TO		
	Records without carriage control	Records with carriage control	No record structure
Records without carriage control	—	Skip 1 line code is added to all records.	Data is written with no record structure.
Records with carriage control	ASA characters are removed from all records.	—	ASA characters and record structure are removed.
No record structure	Data is divided into records using a maximum record length specification.	Data is divided into records using a maximum record length specification, and Skip 1 line code is added to all records.	—

When converting between ASCII and EBCDIC, the data is translated using the appropriate translation tables in the Barr RJE product. Any ASA characters are also translated. When converting between binary and either ASCII or EBCDIC, the character set code is changed but no other conversion takes place. (This “conversion” between binary and another character set is defined so that both EBCDIC and binary data in MVS data sets can be handled even when a unique character set is associated with each record format code.)

All MVS record formats are assigned the EBCDIC character set. Copying binary data to an MVS data set results in no translation; copying ASCII data results in translation. There are separate computer record formats for ASCII and binary data.

Converting Between Files With and Without Record Structure

When converting from a file with record structure to a file without, record boundaries are lost. Any ASA information in the source file is thrown away.

A file with no record structure in BFT format is divided up arbitrarily into BFT format records. A BFT format record length of 775 bytes (10 records per record with a last short record) gives 97 percent data bytes in the BFT format stream.

When converting from a file with no record structure to a file with record structure, the maximum record length of the destination file is used to divide the data into records. (The last one may be short; if the destination has a fixed-length format, it must then be padded.) If the destination file also needs ASA information, a plus (no-space code) is added to the first record; a blank (single-space code) is added to all other records.

Shortening or Lengthening Records

When shortening records to fit a smaller maximum record length, records are truncated on the right.

When lengthening records to make up a fixed-length record file, records are padded on the right with blanks in the character set of the destination file, or with binary zeroes if the destination file is binary.

Deferred Format Conversions

A BFT format file image may or may not have a character set and format (CHARSET and FORMAT) matching those implied by its native record format (SYSTEM and RECFM). Also, records longer than LENGTH (though not longer than 32,760 bytes) can be transmitted. This provision allows deferred conversions, where data is converted to the destination format as late as possible.

Notes:

Index

A

Adapter · iii
Adapter serial number · iii
APF · 26
ASCII tab characters · 9
ASCII-to-EBCDIC translation · 9
Authorized library · 25, 26

B

Barr file transfer methods · 2
Barr RJE software · iii, 1, 4, 7, 26
Barr software disk · 23
BARR/TAPE · 26
BARR/TRAN · 26
BARR/TRAN job log · 12, 13, 19
BARR/TRAN software · 1, 9, 23
Binary · 5, 6, 15, 21, 22, 25, 28, 29, 30
Binary file · 15, 21, 22
Block size · 6, 25, 29

C

C:\BARR\TRAN directory · 23
Carriage control · 1, 10
Cataloged procedure · 23, 25
Character set translation · 9
Computer requirements · 1
Condition code · 12, 18

D

Data Control Block (DCB) · 4, 6
Data record · 7
Destination computer · 9
DSNAME · 11

E

EBCDIC-to-ASCII translation · 9

F

File name · 11, 17
File transfer · 17, 23, 27
FILEOUT DD statement · 11, 12
Form feed · 1, 13, 14

H

Header record · 7
Help - contact Technical Support · iii

I

INSTALL.JOB · 23, 26

J

JCL · 23
JES trailer lines, deleting · 1
Job Control Language (JCL) · 23

L

Library · 22, 23, 25, 26
Lines · 5, 13, 25
Logical Record Length · 6, 7

M

Mainframe file name · 11, 17
Mainframe object module · 16
MVS · 11, 12, 14
MVS data set · 15, 18, 20, 21
MVS object deck · 22
MVS printer data · 19
MVS/ESA · iii
MVS/XA · iii

N

Naming files · 1

O

Operating systems · iii, 3, 5, 23

Overprint · 1

P

Print file · 14, 15

R

Record format · 11, 36

Record format (RECFM) · 12, 16, 17

Record length conversion · 9

S

Send Files command · 11

Serial number · iii

Software version · iii

Source computer · 9

STEPLIB DD statement · 25

SYS1.PROCLIB · 26

SYSLMOD DD statement · 25

T

TA file format · 5

TB file format · 5

TBFn file format · 5

TBV file format · 6

TDFn file format · 5

Technical Support · iii

TFn file format · 5

TP file format · 5

Transformation rules · 4, 9

TV file format · 5

V

Version, software · iii

VS1/RES · iii, 1, 23