60496100

CONTROL DATA
CORPORATION

INTERCOM VERSION 4
REMOTE BATCH USER'S GUIDE

CDC® OPERATING SYSTEM:
NOS/BE

INSTALLATION-DEFINED PARAMETERS

Consult your instructor or site analyst to obtain the parameter values defined for your site and record them on the following table. The table lists only those installation-defined defaults that are considered useful in determining control statement values. Other default values are not alterable, affect the internal performance of the system, are not directly related to control statements, or are generally considered not necessary for normal program use.

Parameter	Default	Range
Job time limit (Tn)	____$_8$ sec.	0 to ____ seconds
I/O time limit (IOn)	____$_8$ sec.	0 to ____ seconds
Job Priority (Pn)	____	1 to ____$_8$
Central memory field length (CMn)	____	____$_8$ to ____$_8$ words (in thousands)
7-track magnetic tape units (MTn)	0	0 to ____$_8$ units
9-track magnetic tape units (NTn)	0	0 to ____$_8$ units
9-track conversion code (US/EB)	____	ASCII or EBCDIC
7-track tape data density (HY, HI, LO)	____	800, 556, or 200 b/i
9-track tape data density (HD, PE)	____	1600 or 800 b/i
Loader map	____	ON, OFF, or PART
Load file positioning	____	Rewind or no rewind
Coded punched card format	____	Hollerith (026) or ASCII (029)
Permanent file retention	____	0 to 999 days
INTERCOM time limit	____	0 to ____ seconds
INTERCOM field length	____	____$_8$ to ____$_8$ words (in thousands)

60496100

CONTROL DATA
CORPORATION

INTERCOM VERSION 4
REMOTE BATCH USER'S GUIDE

CDC® OPERATING SYSTEM:
NOS/BE

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released.
(8-31-77)	
B	Revised to include the DEFINE command and to reflect PSR summary level 473.
(8-16-78)	

Publication No.
60496100

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-								
Inside Front Cover	-								
Title Page	-								
ii	B								
iii/iv	B								
v/vi	B								
vii	B								
viii	B								
1-1	A								
1-2	A								
2-1	A								
2-2	A								
2-3	A								
2-4	A								
2-5	A								
2-6	B								
2-7	B								
2-8	B								
3-1	A								
3-2	A								
3-3	A								
3-4	A								
3-5	A								
3-6	A								
4-1	A								
4-2	A								
4-3	A								
4-4	A								
4-5	A								
4-6	A								
4-7	A								
A-1	B								
A-2	A								
B-1	B								
B-2	B								
B-3	A								
B-4	A								
B-5	A								
B-6	A								
B-7	A								
C-1	A								
C-2	A								
C-3	A								
C-4	A								
Index-1	B								
Index-2	B								
Comment Sheet	B								
Back Cover	-								

PREFACE

This manual is a guide for novice INTERCOM users in processing remote batch jobs on a CDC® CYBER 170 Series, CDC®CYBER 70, models 71, 72, 73 or 74, or CDC®6000 Series Computer System. This user's guide describes the commands and directives of INTERCOM that permit a batch job to be submitted at a remote terminal for processing on the central computer. Control statements and separators needed to process jobs on the NOS/BE Operating System are also described.

This procedurally oriented guide explains how to use INTERCOM only for remote batch processing; it does not describe all INTERCOM features. Information is presented in the order that it will likely be used: section 1 presents a general overview of remote batch processing; section 2 describes the basic INTERCOM commands needed to operate a remote batch terminal; section 3 explains job deck organization and control statement philosophy; and section 4 describes the INTERCOM commands and system control statements used to manipulate files. Examples are used extensively to illustrate how to use INTERCOM and NOS/BE commands.

The user of this manual need not have NOS/BE or INTERCOM experience, but should be familiar with both the language of the source program and the operation of the terminal being used.

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

In addition to the operator's guide for the terminal being used, more information useful to remote batch users can be found in the following publications.

Control Data Publication	Publication Number
INTERCOM Version 4 Reference Manual	60494600
NOS/BE Version 1 User's Guide	60494000
NOS/BE Version 1 Reference Manual	60493800
INTERCOM Version 4 Remote Batch Command Summary	60495400
NOS/BE Version 1 Diagnostic Handbook	60494400
LOADER Version 1 Reference Manual	60429800
CYBER Record Manager Version 1 Reference Manual	60495700
FORTRAN Extended Version 4 Reference Manual	60497800
COBOL Version 4 Reference Manual	60496800
COBOL Version 5 Reference Manual	60497100
COMPASS Version 3 Reference Manual	60492600
ALGOL Version 4 Reference Manual	60496600
BASIC Version 3 Reference Manual	19983900

CONTENTS

APPENDIXES

INDEX

FIGURES

INTRODUCTION 1

INTERCOM is a communications subsystem that operates under the NOS/BE operating system to provide computer access from a remote terminal. Two operating modes, interactive and remote batch, are available through INTERCOM. In interactive mode, files can be created and edited from a remote terminal. Interactive mode also permits interactive control statement processing: most control statements that can be submitted as part of a job can be executed one at a time through INTERCOM.

Batch mode permits job decks to be read into the NOS/BE input queue from a remote card reader. The operating system processes the job as though it had been read from a central site card reader. When processing has completed, the system writes the results on the system output file. Output resulting from execution is returned to the originating terminal for printing or punching unless directed specifically to the central site or to another terminal.

Any program submitted for NOS/BE batch processing as an input file must contain the necessary control statements at the beginning of the deck. These control statements can include any applicable operating system control statements.

REMOTE BATCH TERMINAL EQUIPMENT

Remote batch jobs can be submitted to the central site from a wide variety of terminal equipment configurations. A basic terminal facility must be capable of issuing commands to INTERCOM, as well as reading input jobs and printing results. The minimum equipment needed to process a remote batch job consists of a display terminal, card reader, and a line printer. An optional card punch permits cards to be punched at a remote batch facility. Several possible terminal configurations are shown in figure 1-1. Some terminals are wired directly into the central site computer, others are connected by dial-up telephone lines. Each terminal has slightly different operating characteristics; therefore, it is necessary for you to become familiar with the terminal that you intend to use before attempting to process a remote batch job. The Batch Terminal Operator's Guide for the terminal being used is an excellent source of terminal operating information.

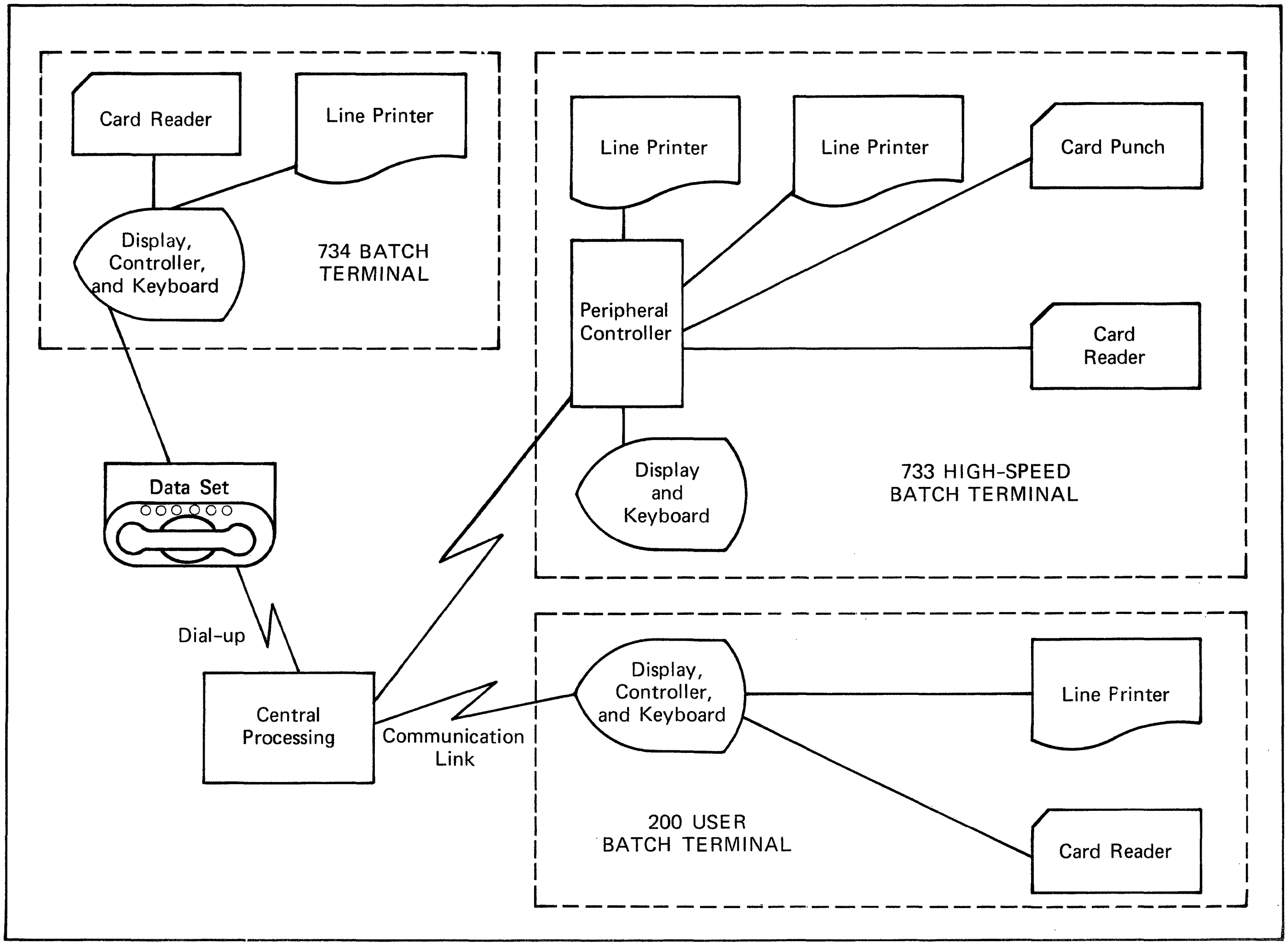


Figure 1-1. Remote Batch Terminal Equipment

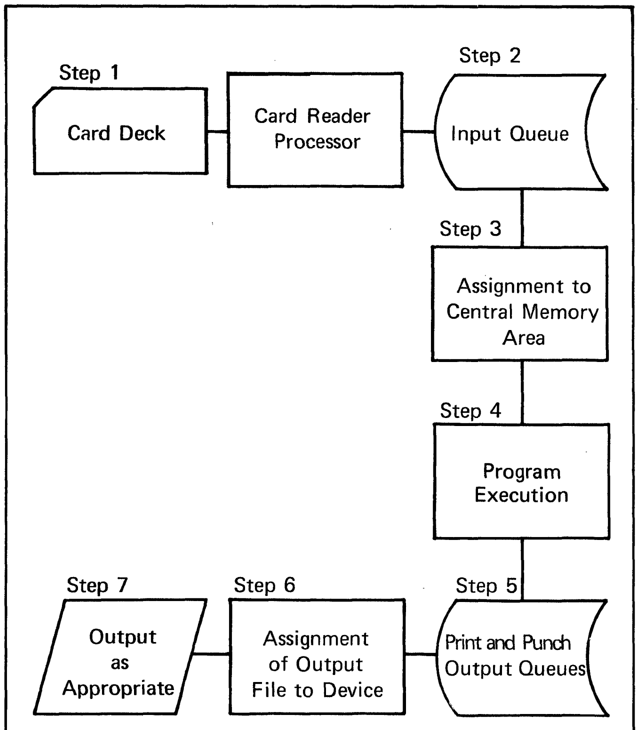
JOB PROCESSING

NOS/BE is a file-oriented operating system. All information processed by the computer is in the form of a file. A file, in the NOS/BE system, is a logically connected set of information that is accessed by its logical file name. Information to generate files can come from a number of sources including punched cards, magnetic tape, or information typed into a terminal. A job deck, complete with control statements, can be considered to be a file.

A typical remote batch job execution sequence is shown in figure 1-2. When a job is submitted to the computer from the remote card reader, the card images of the job deck reside as a file on the system disk. The operating system generates a file name for the job file by combining the first five characters of the name specified on the job statement with two unique system-generated characters. This file name distinguishes the job file from all other files in the system. When the job file has been named and loaded onto the system disk, it is said to be in the input queue. The system establishes a priority for the job by examining the job statement information. As system resources become available, the operating system selects the highest priority job from the input queue and loads it into central memory. Execution information for the job is displayed at a control point and the job is said to be in the execution queue.

Job execution begins with the first control statement following the job statement. Each control statement is executed as it is encountered until a 7/8/9 multipunch card indicates the end of the control statement section. Some statements, such as a compiler call statement, must access a file (to obtain data to compile). When no file is specified, the local file with the logical file name INPUT is assumed. The INPUT file contains the image of the job deck. Each time INPUT is read, the card images after the last 7/8/9 separator card are processed. For example, an FTN call statement, with no input file parameter specified, begins compiling the FORTRAN job from INPUT (starting with the first card image following a 7/8/9 separator).

As job execution continues, information to be printed is written on the system disk as the local file, OUTPUT; information to be punched is called PUNCH, PUNCHB, or P80C. When execution is completed, the job is cleared from the system control point and the execution queue. The OUTPUT file is given the system-assigned job name and entered into the output queue; PUNCH, PUNCHB, and P80C files are given the system-assigned job name and entered into the punch queue. As output devices become available, the operating system causes files from the output and punch queues to be physically output. Output for jobs submitted at a remote batch terminal are normally returned to that terminal.



Step 1. Job input through card reader
Step 2. Creation of job input file
Step 3. Control point assignment
Step 4. Program execution
Step 5. Assignment to output queue
Step 6. Assignment of output file to output device
Step 7. Job output (as appropriate)

Figure 1-2. Typical Job Flow

REMOTE BATCH OPERATION 2

This section describes the INTERCOM commands and displays used for remote batch processing. Information presented in this section is intended primarily to guide you in submitting batch jobs from a remote terminal, monitoring job progress during execution, and obtaining job output.

Remote batch operations described in this section can be conducted from any remote batch terminal equipped with a display terminal, card reader, and a line printer. Some terminals include a card punch; therefore, commands that affect card punching are also explained.

Throughout this section, the direction 'enter' means to key in (on the terminal keyboard) the specified characters, then press the key that causes the displayed characters to be transmitted to the central computer system.

DIAL-UP

Terminals that are not dedicated (wired directly into the computer system) must be connected through a dial-up telephone line before remote batch operation can begin. To establish telephone communication with the central computer system:

1. Lift the data set receiver from its cradle, press the TALK button, and wait for a dial tone.
2. Dial the telephone number used to connect the central computer system.
3. The data set at the other end of the line will either return a busy signal or begin to ring. If a busy signal is received, the line is already in use; hang up and wait a short time before re-dialing or dial another number. If the ring signal is returned, wait until the phone is answered and a high-pitched tone is heard.
4. Press the DATA button on the data set, and place the receiver back in its cradle. The terminal is now connected to the central computer system; a message appears on the display terminal that is similar to:

```
CONTROL DATA INTERCOM
DATE    06/07/77
TIME    10.21.28.

PLEASE LOGIN
```

LOGIN

Software communication with INTERCOM is established from a dial-up batch terminal by the LOGIN command. It is not necessary to log in at dedicated-line (hardwired) terminals before using remote batch commands; but you must be logged in to use interactive commands. LOGIN must be the first command entered after hardware communications are established at a dial-up terminal. At the conclusion of a successful log-in procedure, INTERCOM has:

Verified your right to access the system

Assigned a two-character user identifier

Made access possible to other INTERCOM commands

The sequence of entries required for log-in could vary at your installation. Generally, however, the LOGIN command asks for identification and a password that indicates authorization to use INTERCOM. Sometimes any name, such as your last name, is acceptable with a given password; at other times, a particular name is necessary. Often, a name is a department or class identification, and the password is an accounting number. Consult with your instructor or system analyst for specific instructions.

The name and password entered during log-in must conform to a range of acceptable values stored at the central site. Both of these items can consist of one to ten letters or digits. If you do not enter values that match allowed values, your log-in attempt will not be successful. Part of the function of the password is to restrict INTERCOM access to authorized users only.

To begin the log-in procedure, enter LOGIN. The system should respond with instructions for additional entries. If the system does not respond, there might be a bad telephone connection, INTERCOM might not be operating on the system, or there could be some hardware or software malfunction. Hang up and re-dial. If there is still no response when you enter LOGIN, hang up and seek help or try again later.

Continue entering the information requested by the system until a successful log-in is achieved. You are logged in when a message is displayed that is similar to:

```
05/13/76    LOGGED IN AT 14.06.29.
            WITH USER-ID CD
            EQUIP/PORT 25/06

COMMAND -
```

USER-ID is the identification code used by INTERCOM to identify users and files to which they are associated. Look for this identification when lists of jobs in the system are being scanned. Also, use this identifier when you consult with the central site operator or system analyst. The EQUIP/PORT information shows the hardware connection to the central site.

At some installations, additional messages might appear before the word COMMAND. These messages can be system bulletins prepared by installation analysts to inform users of operating hours or analysts telephone numbers. These messages could also contain instructions on how to obtain additional help or system information.

You can condense the log-in procedure by entering some or all of the items required at the same time that LOGIN is entered. Use commas to separate items when entering more than one item.

Any of the following methods could be used to log in:

Enter LOGIN.

Wait to be prompted before entering other items. The system will ask for a user name, then ask for a password.

Enter LOGIN, username.

Bécause the user name is entered with LOGIN, the system will prompt only the password entry.

Enter LOGIN, username, password.

The system responds with informative bulletins or messages.

Enter LOGIN, username, password, SUP.

The SUP parameter suppresses the information normally displayed at the end of the log-in procedure; all three characters (SUP) must be entered. The SUP parameter can be used only with this abbreviated entry.

Once the word COMMAND appears, INTERCOM is ready to receive additional user commands.

Summary of log-in (user entries are underlined):

```
PLEASE LOGIN
LOGIN
ENTER USER NAME -
SOMENAME
ENTER PASSWORD -
YOURKEY
(date, time, user-id)
(installation information can appear here)
COMMAND -
```

Summary of abbreviated entry:

```
PLEASE LOGIN
LOGIN, SOMENAME, YOURKEY, SUP
COMMAND -
```

SCREEN

Display terminals can differ in the format of the display. Some remote terminals display data as 80 columns by 13 lines, others display it as 50 columns by 20 lines, and still other terminals display data as 80 columns by 16 lines. It is possible that the default display format selected for your terminal could be wrong. When the wrong format is selected, either line or page wraparound can occur. If information that should be at the end of a line or page is displayed on a new line or at the top of the page, it means that the wrong screen size has been selected. Use the SCREEN command to change the display format of the terminal by entering:

```
SCREEN,50,20
    or
SCREEN,80,13
    or
SCREEN,80,16
```

READ

After software communications have been established, the next thing you will normally want to do is input your job through the card reader. Place your job deck in the card reader and make it ready. Card reading begins automatically at some remote batch terminals when cards are placed in the reader and it is made ready; other terminals require you to enter R (for READ). Some terminals, such as the 200 UT, require that the LOAD button on the controller be pushed to initiate card reading. The card reading operation is the same for all types of terminals.

Cards are read into an input file, which is automatically created for the job. The first card read must be a valid job statement. Section 3 contains a detailed explanation of the job statement and other basic control statements.

More than one job can be stacked in the card reader for continuous reading without operator intervention. Each time an end-of-information (6/7/8/9) card is encountered, the job it terminates is placed in the input queue for NOS/BE processing. After the end-of-information card, the next card must be either another end-of-information card or a valid job statement, or the input hopper must be empty. Extra end-of-information cards are ignored if no other cards separate them. Card reading continues until the input hopper is empty. The last card read must be an end-of-information card.

If the system input queue overloads during a heavy job load, all card readers are placed in the WAIT state and the following message is displayed:

```
INPUT SUSPENDED BY SYSTEM
```

Card reading continues automatically when space becomes available, provided the printer is logically on. (Polling for input occurs only when the printer is logically on.) Some terminals continue reading cards automatically even when the printer is not logically on. If you do not want the printer logically on, and your terminal does not resume card reading automatically, enter G,CRn (n is the card reader device number). If the system input queue is still overloaded, the message INPUT SUSPENDED BY SYSTEM is displayed again. Wait a short time and reenter G,CRn.

When the first card read (or the first card after an end-of-information card in a multiple job read operation) is not a valid job statement, the following message is displayed and the entire file is ignored:

```
CRn, j j j j j j j, JOB CARD ERROR
```

where n is the card reader device number, and j j j j j j j are the first seven characters of the job statement. The job statement must be corrected and the job reread. If you happen to put a blank card at the end of your deck, it is read as a job statement because it is the first card read after the end-of-information card. Because a blank card is not a valid job statement, the following message would be displayed:

```
CR1,              , JOB CARD ERROR
```

This example assumes the card reader device number is 1 (CR1). The first seven characters of a blank job statement are blanks; therefore, blanks appear in the displayed message.

An uncorrectable parity error (or other equipment trouble) encountered while reading the cards causes the file to be ignored and the following message displayed:

```
CRn, j j j j j j j , INPUT FILE ERROR
```

where n is the card reader device number, and j j j j j j j is the seven-character job name. The job should be reread from the beginning.

When no read errors occur, card reading continues until an end-of-information is reached. If the last card read is not an end-of-information card, INTERCOM expects more cards and displays the following message:

```
CRn, NOT READY
```

More cards (or an end-of-information card) must be placed in the input hopper and the card reader made ready. Some terminals read the additional cards automatically; if the terminal in use does not, enter G,CRn.

H COMMAND

If all has gone well so far, your program has been read by the card reader and submitted to the system input queue.

You are now ready to monitor your jobs' progress through execution. The H command can be used to request a display of the queues associated with the terminal or to show the status of the terminal peripherals. If the card reader did not read your job, you might find out why by using the H,S command to display terminal status (the card reader might be logically off).

Only jobs associated with the terminal-id appear on an H display. The display is updated automatically every 10 seconds, and is terminated by entry of any command. Sometimes it is difficult to enter the desired command between display updates; Therefore, enter E (or END) to stop display updating and then enter the desired command.

In figures 2-1 through 2-5, the screen size used is 50 by 20.

To display the input queue (figure 2-1), enter:

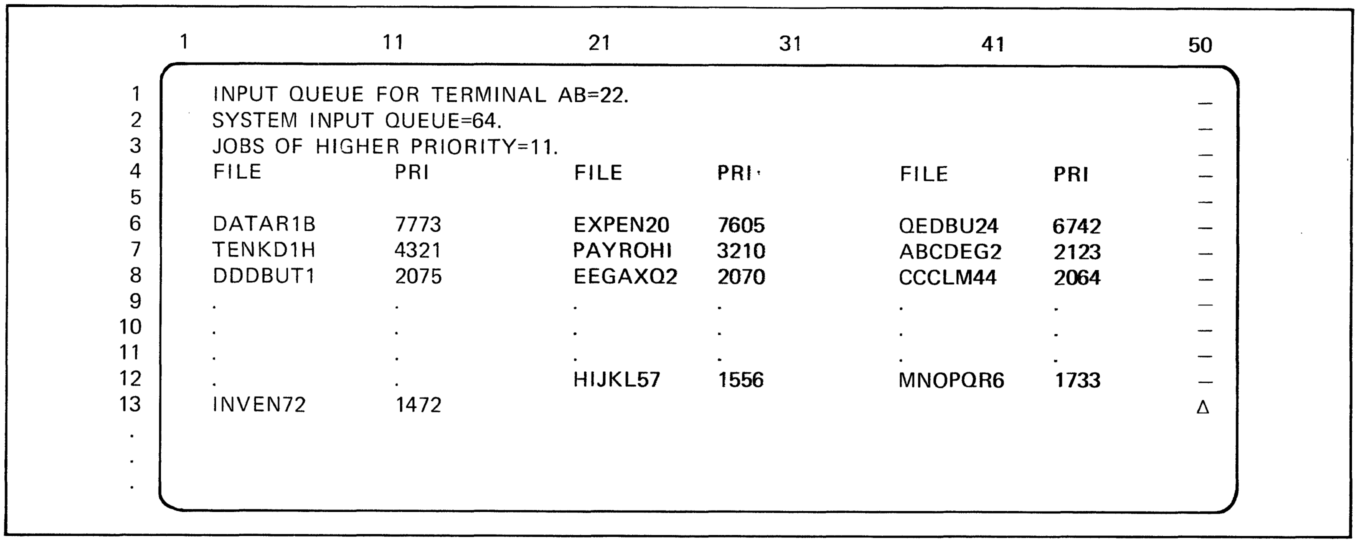
```
H,I
```

The screen displays the file name (column FILE) and priority (PRI). Priority is a numerical value assigned by the system to determine the order in which files are processed. The file with the highest PRI value is processed first. Three columns of filenames, eight to a column, are displayed from left to right in descending priority. A maximum of 24 files can be displayed. If the permanent file area is full, SUSP is displayed in the upper right corner of the display.

To display the output (print) queue (figure 2-2), enter:

```
H,O
```

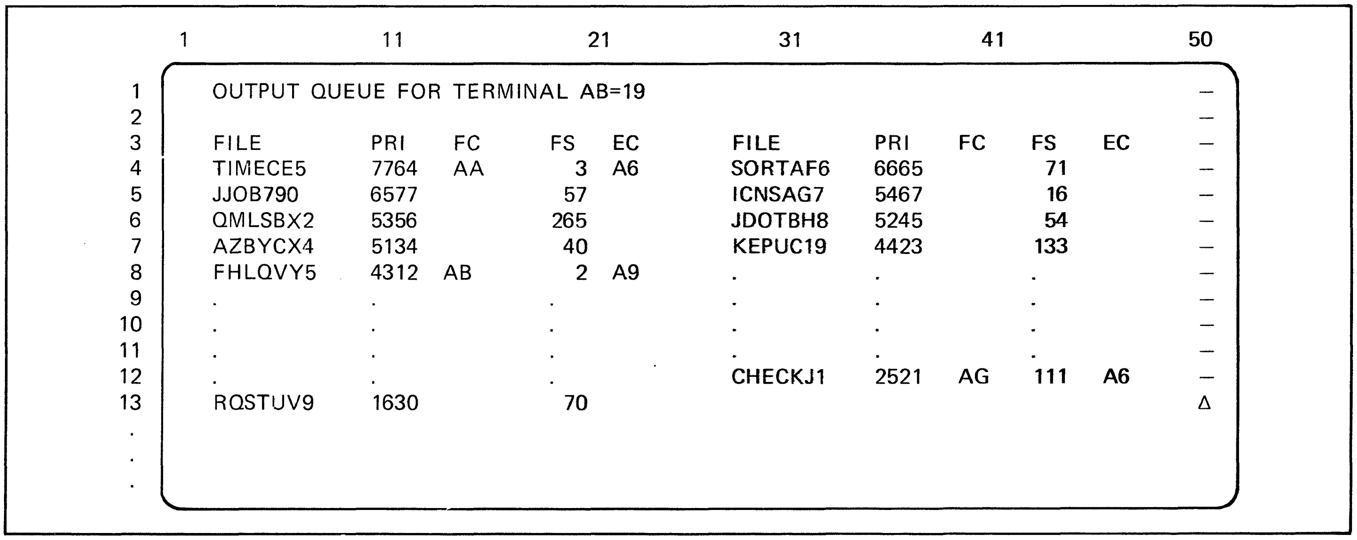
The screen displays the file name (FILE), priority (PRI), forms code (FC), file size (FS), and requested print train (EC). The external characteristics (EC) parameter is a two-character identifier used to indicate the type of print train required to correctly print the file. (The EC parameter is explained in detail in the NOS/BE 1 Reference Manual.) You will normally not have to be concerned with the external characteristics of the print train. Forms code is an arbitrary two-character identifier used to associate a print file with a desired printer form. The file size is shown in number of sectors, including repetitions, divided by 100_8. Up to 10 file names are displayed in each of the two columns, left to right in descending priority.



```
       1           11          21          31          41         50

  1    INPUT QUEUE FOR TERMINAL AB=22.
  2    SYSTEM INPUT QUEUE=64.                                        _
  3    JOBS OF HIGHER PRIORITY=11.                                   _
  4    FILE        PRI         FILE     PRI        FILE     PRI      _
  5                                                                  _
  6    DATAR1B     7773        EXPEN20  7605       QEDBU24  6742     _
  7    TENKD1H     4321        PAYROHI  3210       ABCDEG2  2123     _
  8    DDDBUT1     2075        EEGAXQ2  2070       CCCLM44  2064     _
  9    .           .           .        .          .        .        _
 10    .           .           .        .          .        .        _
 11    .           .           .        .          .        .        _
 12    .           .           HIJKL57  1556       MNOPQR6  1733     _
 13    INVEN72     1472                                              Δ
  .
  .
  .
```

Figure 2-1. Input Queue Display (H,I)



```
       1           11          21          31          41         50

  1    OUTPUT QUEUE FOR TERMINAL AB=19                               _
  2                                                                  _
  3    FILE      PRI   FC   FS  EC     FILE     PRI   FC   FS   EC   _
  4    TIMECE5   7764  AA    3  A6     SORTAF6  6665          71     _
  5    JJOB790   6577       57         ICNSAG7  5467          16     _
  6    QMLSBX2   5356      265         JDOTBH8  5245          54     _
  7    AZBYCX4   5134       40         KEPUC19  4423         133     _
  8    FHLQVY5   4312  AB    2  A9     .        .              .     _
  9    .         .           .         .        .              .     _
 10    .         .           .         .        .              .     _
 11    .         .           .         .        .              .     _
 12    .         .           .         CHECKJ1  2521  AG    111   A6 _
 13    RQSTUV9   1630       70                                       Δ
  .
  .
  .
```

Figure 2-2. Output Queue Display (H,I)

Sector and physical record unit (PRU) are used interchangeably to describe the FS parameter. Both generally contain 64 central memory words. Each word can contain 10 characters; therefore, a file sector contains approximately 640 characters. For more information on sectors, see the BSP command description.

To display the punch queue (figure 2-3), enter:

H,P

The screen displays the file name (FILE), priority (PRI), forms code (FC), and file size (FS). The file names are displayed the same as those in the print queue.

To display the execution queue (figure 2-4), enter:

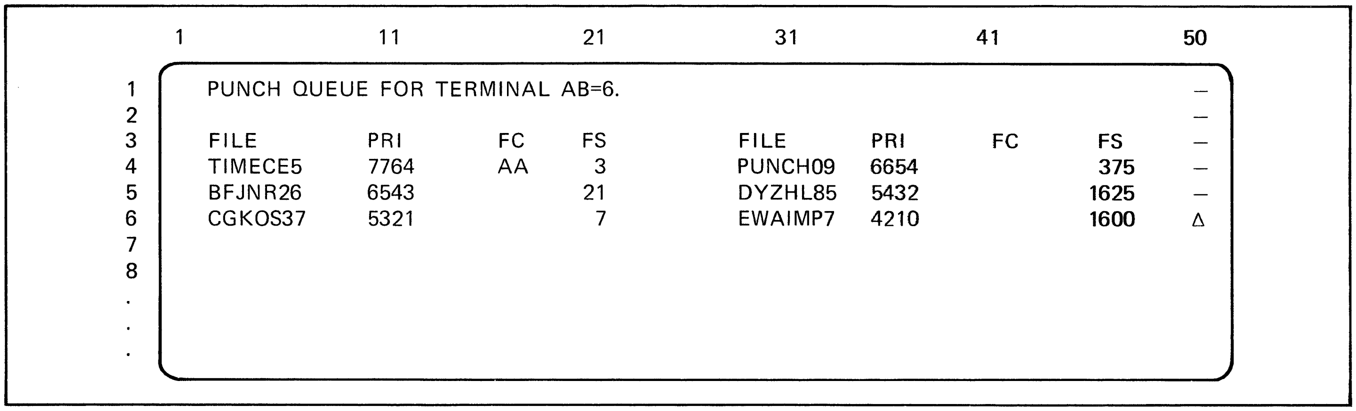
H,E

The screen displays the job name, priority (PRI), status, field length (FL), and the number of CPU seconds that have elapsed (ET) since the job started execution. Field length is an octal integer shown in hundreds (577 = 57 700_8). The status is displayed by one of the following codes:

W-SWAP	Waiting for a swap.
W-MEMORY	Waiting for memory.
W-PFILE	Waiting for a permanent file.
W-DEVICE	Waiting for a device.
W-OPRTR	Waiting for operator action.
W-INTRCM	Waiting for INTERCOM.
W-P PACK	Waiting for a permanent pack.
W-MMFRME	Waiting for a multimainframe.
EXECUTING	Executing at a control point.
W-SCHED	Waiting for scheduler action.

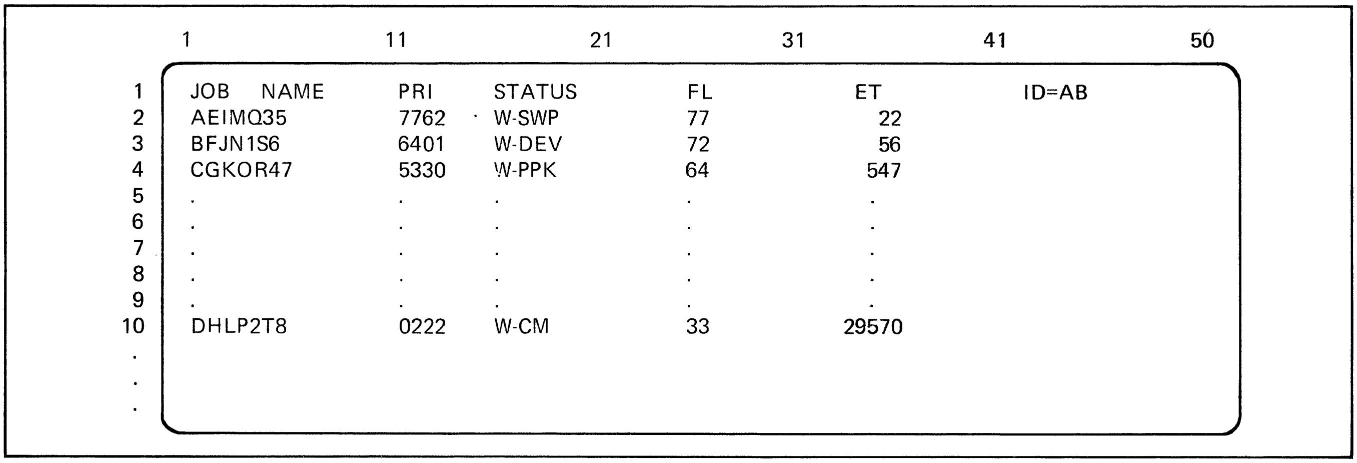
To display the terminal's status (figure 2-5), enter:

H,S



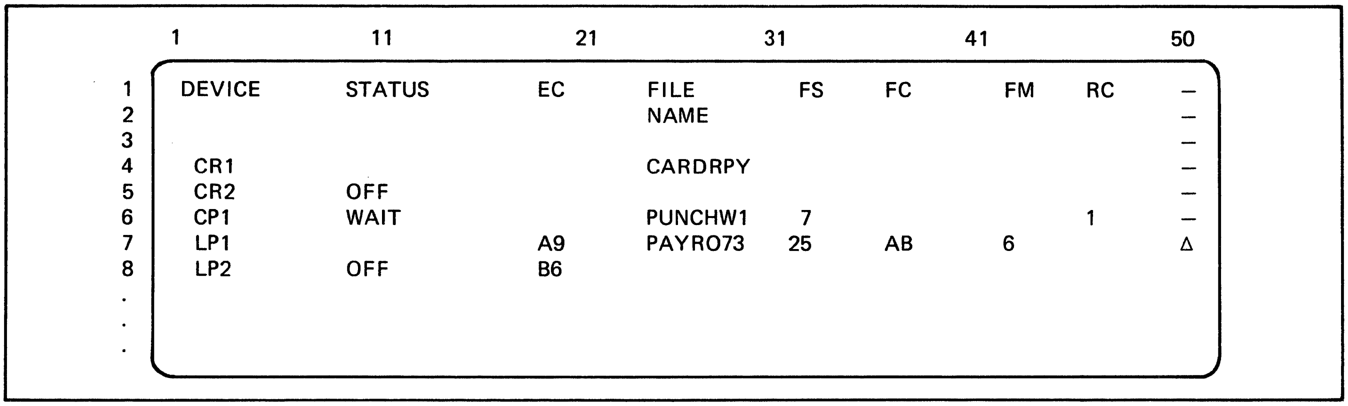
```
      1         11         21         31         41         50
  1   PUNCH QUEUE FOR TERMINAL AB=6.                          —
  2                                                           —
  3   FILE      PRI      FC   FS     FILE      PRI   FC    FS  —
  4   TIMECE5   7764     AA    3     PUNCH09   6654       375  —
  5   BFJNR26   6543          21     DYZHL85   5432      1625  —
  6   CGKOS37   5321           7     EWAIMP7   4210      1600  Δ
  7
  8
  .
  .
  .
```

Figure 2-3. Punch Queue Display (H,P)



```
       1         11         21         31         41         50
   1   JOB  NAME  PRI    STATUS      FL        ET       ID=AB
   2   AEIMQ35    7762 · W-SWP       77        22
   3   BFJN1S6    6401   W-DEV       72        56
   4   CGKOR47    5330   W-PPK       64       547
   5   .          .      .           .          .
   6   .          .      .           .          .
   7   .          .      .           .          .
   8   .          .      .           .          .
   9   .          .      .           .          .
  10   DHLP2T8    0222   W-CM        33     29570
   .
   .
   .
```

Figure 2-4. Execution Queue Display (H,E)



```
        1            11              21             31             41            50

   1    DEVICE       STATUS          EC             FILE     FS     FC     FM    RC    –
   2                                                NAME                               –
   3                                                                                   –
   4     CR1                                        CARDRPY                            –
   5     CR2         OFF                                                               –
   6     CP1         WAIT                           PUNCHW1   7                   1    –
   7     LP1                         A9             PAYRO73  25     AB     6           Δ
   8     LP2         OFF             B6
   .
   .
   .
```

Figure 2-5. Device Status Display (H,S)

The screen displays the device name (LPn, CPn, or CRn), its status (ON, OFF, ERROR, SUSP, WAIT,WAIT/E, or GO), and (for printers) its print train along with the file name, the number of physical record units (PRUs) shown in hundreds ($25=2500_8$) remaining to be printed (FS), and the forms code (FC). The status display also shows an octal integer code that identifies the forms control matrix (FM) selected. The forms control matrix is used to control line spacing on some high-speed printers. The last column shown on the status display is the repeat count (RC); it indicates the number of additional output file copies desired.

Q

Additional information concerning jobs being processed by the system can be obtained by examining the batch processing queues. The Q command is an interactive command; therefore, if you are using a hard-wired terminal and are not logged in, the Q command cannot be used.

The system queues might contain more information that can be displayed on the terminal screen at once. After the Q command is entered and the first page of the Q display appears on the screen, request the next page by entering a blank character. To enter a blank character, press the key that causes a displayed character to be transmitted to the central computer, without first striking a character key.

If the information displayed by the Q command does not fit the screen (wraparound occurs), use the SCREEN command to reformat the display.

To obtain a count of jobs in the input, output, and execution queues, enter:

```
Q
```

To obtain a list of all job names in the input, output, punch, execution, and JANUS queues, enter:

```
Q,A
```

To obtain a display of the Q command syntax, enter:

```
Q,S
```

To display all jobs in the system input queue, enter:

```
Q,I
```

To display all jobs in the system output queue, enter:

```
Q,O
```

To display all jobs in the system punch queue, enter:

```
Q,P
```

To display all jobs in the system execution queue, enter:

```
Q,E
```

To display all jobs associated with JANUS, enter:

```
Q,J
```

ON

The ON command turns the specified peripheral device logically on. At log-in time, the card reader is initially on; the output devices are initially off. If you are the first person to use the terminal after log-in, or if the line printer is logically off as a result of a previous operation, you must turn the line printer logically on before the output will be printed.

The line printer is the default device for the ON command; therefore, to turn the printer logically on, enter:

```
ON
```

Line printer 1 can also be turned logically on by entering:

```
ON,LP1
```

To turn card reader 1 logically on, enter:

```
ON,CR1
```

OFF

The OFF command turns the specified equipment logically off, negating the effect on the ON command. Cards cannot be read when the card reader is logically off, nor can output be sent to devices that are logically off.

The line printer is the default device for the OFF command; therefore, to turn the printer logically off, enter:

```
OFF
```

Line printer 1 can also be turned logically off by entering:

```
OFF,LP1
```

To turn card reader 1 logically off, enter:

```
OFF,CR1
```

DEFINE

The DEFINE command specifies attributes of a print or punch device. If the device is already configured, DEFINE modifies attributes of the device. If the device is not configured, DEFINE configures it with the specified attributes.

To specify the nondefault characteristics of a print or punch device, enter:

$$\text{DEFINE,eqo=}\left\{\begin{matrix}\text{IP}\\\text{NIP}\end{matrix}\right\}\text{,FCfc,}\left\{\begin{matrix}\text{BANON}\\\text{BANOFF}\end{matrix}\right\}$$

All parameters are optional. The eqo parameter is positional; the other parameters are order-independent. Braces indicate only one item can be chosen.

eqo	Output device. Default is LP1.
IP	Impact printer. Selection is required only for 714 terminals when the type of printer is other than the installation-defined default.
NIP	Nonimpact printer. Selection is required only for 714 terminals when the type of printer is other than the installation-defined default.
FCfc	Forms code that identifies either the form paper or the card stock in the output device, or an installation-defined procedure. This is a two-character installation-defined parameter. Files requiring certain forms are printed or punched only at devices having forms codes that match the forms code of the file. The default is no forms code.
BANON	File identification banner pages are printed at the beginning of each file. Because of terminal hardware restrictions, the system produces a banner page that differs slightly from a banner page printed at the central site. Other pages of the listing can differ in appearance if the terminal printer does not recognize all of the printer carriage control characters supported for central site devices (refer to appendix A). BANON is the default.
BANOFF	Suppress printing of file identification banner pages.

If you enter a DEFINE command for an already configured device, only the attributes specified are affected. Unspecified attributes are not reset to system defaults.

Some devices are already configured when you enter the first batch command of the terminal session. The number of these devices configured by default and the type and number of additional devices allowed varies according to the protocol in use at the terminal. The set of default devices is an installation parameter that can be changed to suit local requirements. The released default devices are a card reader (CR1) and a line printer (LP1). For terminals with additional devices, you must enter a DEFINE command to configure each additional device.

The type and number of output devices allowable for the various protocols are given in table 2-1.

TABLE 2-1. ALLOWABLE OUTPUT DEVICES

Terminal Protocol	Card Punches	Line Printers
Mode 4A	none	LP1
Mode 4C	none	LP1-LP3
Mode 2 LSBT, MSBT	CP1	LP1
Mode 2 HSBT	CP1	LP1, LP2

CONTIN

If printing of the output file does not start when job execution is completed and the printer is physically ready, not busy, and logically on, you are probably using a terminal that delays printing for approximately 1 minute after the last entry. To start printing without the delay, monitor the H,O display and when the job is in the output queue, enter:

```
C or CONTIN
```

INTERRUPT

While data is being read or printed, the keyboard on some terminals is locked so that data cannot be overwritten; therefore, to enter a command when the terminal is actively printing or reading cards, you must first interrupt the operation. The interrupt procedure varies widely among terminal types. Check the terminal operator's guide for the proper procedure to be used at your terminal.

In most cases, the active operation resumes automatically after a command is entered. If it does not, you can resume the operation by entering:

```
C or CONTIN
```

Many of the commands described in the remainder in this section require the active operation to be interrupted if they are used at a terminal with a locked keyboard. If your terminal does not have the locked keyboard feature, ignore the direction to interrupt the operation; enter the commands as directed.

WAIT

You can temporarily halt the reading, printing, or punching operation by using the WAIT command. If input or output is not in progress when WAIT is entered, no new jobs are read, printed, or punched until the GO command is entered.

The default equipment for the WAIT command is line printer 1; therefore, to halt printing on LP1, interrupt and enter:

```
WAIT
```

To halt card reader 1 temporarily, interrupt and enter:

```
WAIT,CR1
```

To halt all printing, punching, and reading, interrupt and enter:

```
WAIT,ALL
```

GO

Operations halted temporarily by the WAIT command resume when the GO command is entered. The GO command is also required to clear the WAIT status and restart any operations suspended by one of the following error conditions:

```
DEVICE NOT READY

JOB STATEMENT ERROR

INPUT FILE ERROR

OUTPUT FILE ERROR
```

The GO command verb can be abbreviated to G; therefore, to restart printing at line printer 1 after WAIT has temporarily halted operations, enter:

```
G,LP1 or GO,LP1
```

To restart reading cards on card reader 1 after a job statement error, enter:

```
G,CR1 or GO,CR1
```

END

The END command stops file reading, punching, or printing. The file on the device specified by the END command is dropped. On line printers, the remaining data is discarded and only the dayfile, if any, is printed. The repeat count is honored. If a second END command is entered, printing terminates immediately; the dayfile is not printed and repeat counts are not honored. The END command verb can be shortened to E. The END command has no default parameter; therefore, the device to which the END command is directed must always be specified.

To terminate output on line printer 1, interrupt and enter:

```
E,LP1 or END,LP1
```

To drop the job currently being read on card reader 1, interrupt and enter:

```
E,CR1 or END,CR1
```

The END command entered with no device specified is used to terminate a refreshing display so that a longer command can be entered. For example, if the last command entered was the H,O command, you can stop the output queue display from being updated every 10 seconds by entering:

```
E or END
```

BSP

You can backspace a print or punch file currently being processed at the terminal by using the BSP command. Files are backspaced in units of 10 octal sectors. A sector, which is also referred to as a physical record unit (PRU), refers to a portion of a disk storage unit that can store 64 central memory words. Each 60-bit central memory word can contain 10 characters; therefore, 640 characters can be stored in a file sector. A file sector contains enough characters to print approximately five lines on the printer. Of course, if less than the maximum number of characters are printed per line, more lines could be printed. Because files are backspaced in units of 10 octal sectors, each BSP parameter will backspace 5120 characters. The number of pages or cards backspaced depends on the number of characters per page or card.

To backspace line printer 1 ten sectors, interrupt and enter:

```
BSP,LP1,1
```

To backspace card punch 1 twenty sectors, interrupt and enter:

```
BSP,CP1,2
```

REW

The REW command rewinds an output file and returns it to the output queue.

To rewind an output file on line printer 1, interrupt and enter:

```
REW,LP1
```

To rewind an output file on card punch 1, interrupt and enter:

```
REW,CP1
```

The file on the device specified is rewound and the device is turned logically off. You can then divert the file to another terminal or to the central site (refer to the DIVERT command, section 4). The device must be turned logically on for output to continue.

REP

The REP command causes the file on the designated device to be reprinted or repunched a specified number of additional times. An octal integer from 1 to 37 specifies the number of additional copies required. When no number is specified, 1 is assumed. REP command entries are cumulative; the maximum number of copies remaining to be processed cannot exceed 37_8.

To obtain two additional copies of the file on line printer 1, interrupt (if printing is in progress) and enter:

```
REP,LP1,2
```

RTN

The RTN command permits an active print or punch file to be rewound and returned to the output queue with a new priority. If a new priority is not specified, the file is returned to the output queue with the old priority. Printing or punching of the file terminates immediately. The residual repeat count is saved. Line printer 1 is the default device for the RTN command. The range of values for the priority is 0 through 7777$_8$.

Unless the priority is zero, printing or punching of the file terminates, the file is rewound, and printing or punching resumes. When the priority is zero, the device is logically turned off. You can then divert the file to another terminal or to the central site (refer to the DIVERT command, section 4). Output does not continue until the device is logically turned on with the ON command.

To rewind, return, and delay the printing of a file on line printer 2 until another output file with a priority greater than 100$_8$ has printed, interrupt and enter:

```
RTN,LP2,100
```

To rewind and return, with the same priority, a file on card punch 1, interrupt and enter:

```
RTN,CP1
```

To rewind and return, with the same priority, a file on line printer 1, interrupt and enter:

```
RTN
```

SUP

Normally, printing at a remote batch terminal is controlled by a carriage control character in the first character position of each print line.

Use the SUP command to suppress carriage control and cause the file to be single-spaced. Line printer 1 is the default device for the SUP command. To suppress carriage control on the file currently printing on line printer 1, interrupt and enter:

```
SUP
```

To suppress carriage control on line printer 2, enter:

```
SUP,LP2
```

Normal carriage control is restored when the file has completed printing.

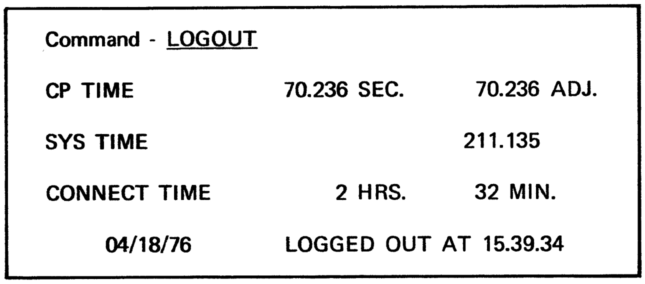
LOGOUT

LOGOUT is the last command entered when all work is completed. It disassociates a dial-up terminal from INTERCOM. No commands can be entered at a dial-up terminal after LOGOUT until the LOGIN command is used to reestablish communications. LOGOUT is not necessary at a hard-wired terminal; it prohibits access to interactive commands.

To execute the LOGOUT command, enter:

```
LOGOUT
```

When LOGOUT is processed, information summarizing session time is displayed as shown in figure 2-6.



```
Command - LOGOUT

CP TIME               70.236 SEC.      70.236 ADJ.

SYS TIME                               211.135

CONNECT TIME              2 HRS.        32 MIN.

    04/18/76          LOGGED OUT AT 15.39.34
```

Figure 2-6. Session Time Summary Display

CP TIME indicates how many seconds were used by the central processing unit at the central site for all operations during this terminal session.

SYS TIME is computed from a formula that reflects all system resources used.

CONNECT TIME shows the elapsed time, in number of hours and minutes, between the LOGIN command and LOGOUT command execution. These times are used by the system to determine accounting charges for users of INTERCOM. You need not be concerned with them unless you have been asked to keep your own or a class log of terminal work.

After you log out, the terminal is normally ready for another user to log in.

HANG-UP

To break telephone communications with the central computer system:

1. Lift the data set receiver from its cradle.
2. Press the TALK button.
3. Place the receiver back on its cradle.

JOB DECK STRUCTURE 3

NOS/BE is a file-oriented operating system; therefore, all job decks submitted for processing must be in the form of a file. This section describes how to organize that file.

Although an input file (job deck) need consist of only one section (the control statement section), most jobs have more than one section. Therefore, the following description assumes a file containing three sections. The first section is made up of system control cards and is called the control statement section; the second section contains the source program; data comprises the third section. A 7/8/9 multiple-punched card separates each section.

SEPARATORS

Two types of separators are used in a job deck. A card with a 7/8/9 multiple-punch in column 1 is used to separate the different logical sections of a job deck; it is called an end-of-section (EOS) card. This card was formerly called the end-of-record (EOR), and could be referred to as such in some other documents. A compiler or assembler treats a 7/8/9 card encountered during processing as a partition or file end.

A job deck is terminated by an end-of-information (EOI) indicator, which is a card with a 6/7/8/9 multiple-punch in column 1. This card serves to terminate the input file for each job deck. Most computer centers provide prepunched EOI cards of a special color to serve as visual separators for each job deck. Only one EOI card should appear in a given job deck. Multiple-punched separator cards are shown in figure 3-1.

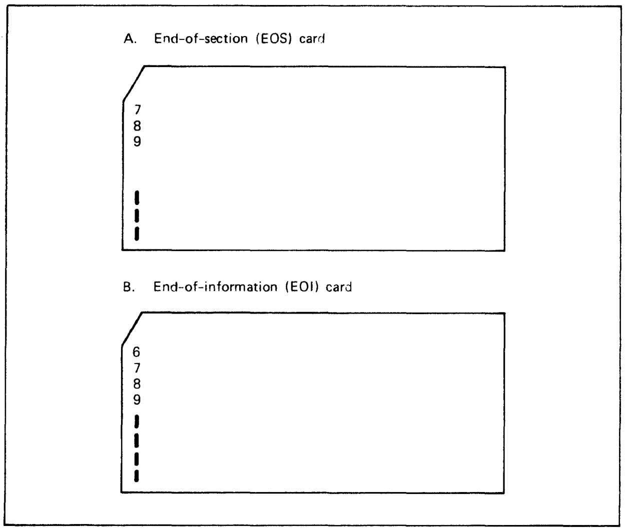

A. End-of-section (EOS) card
7
8
9
B. End-of-information (EOI) card
6
7
8
9

Figure 3-1. Punched Separator Cards

CONTROL STATEMENT SECTION

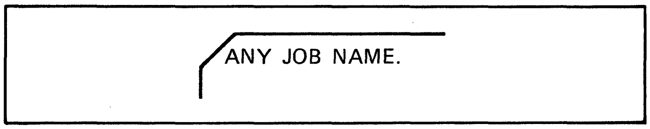
Control statements are instructions to the operating system or its loader. All job control statements that apply to a particular job must appear within the control statement section; when they do not appear in this section they are not processed unless special control statement directives are used. Control statement cards are examined and executed, one at a time, in the order in which they occur in the control statement section. Information contained on a control statement begins in column 1, each parameter is separated by a comma or a left parenthesis, and the final parameter is terminated with a period or a right parenthesis.

The control statement section must begin with a job statement. The type and order of additional cards is determined by the operation to be performed by the job.

A complete description of job control statements is presented in the NOS/BE 1 Reference Manual. Only those statements commonly used by beginning remote batch users are described here.

JOB STATEMENT

The job statement must be the first statement in the control statement section; it is used to name the file (job) and allocate various system resources. Job name is the only required parameter in this statement. When a job name is the only parameter used in the job statement, it must be terminated with a period. A minimum required job statement is shown in figure 3-2. Any name (regardless of its length) that starts in column 1, begins with a letter, and is comprised of only letters, numbers and blanks is a legal job name.

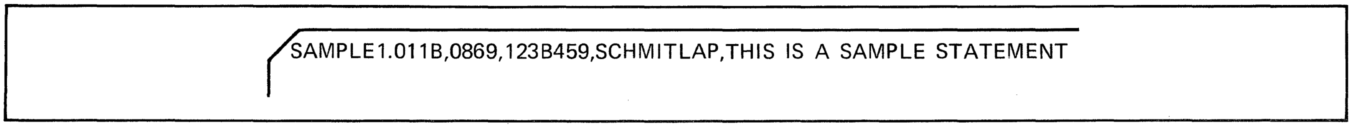


```
ANY JOB NAME.
```

Figure 3-2. Minimum Job Statement

When additional parameters are not supplied, the predetermined parameters established by the computer center are assumed for resources such as time limit and central memory storage. These installation-defined values are generally sufficient for the majority of jobs submitted by beginning users. It is worthwhile to find out what the installation-defined parameters are and record them on the inside front cover of this manual.

In any job statement, the terminating period can be followed by comments and installation accounting information. An example of a job statement with accounting data and a comment is shown in figure 3-3. Check with your local computer center consulting staff or instructor for local accounting details.



```
SAMPLE1.011B,0869,123B459,SCHMITLAP,THIS IS A SAMPLE STATEMENT
```

Figure 3-3. Job Statement with Accounting Data and a Comment

In addition to the job name, the job statement can contain the optional parameters described below.

Execution Time Limit (T)

T specifies the maximum number of central processor seconds allowed a job. Some installations choose to specify time in octal seconds, others use decimal seconds. Ten seconds is a good rule-of-thumb value for small, student-type FORTRAN Extended or COBOL jobs. Larger jobs, of course, take more time to process. Inspecting the output from a previous job can help to better estimate the amount of time required to process subsequent jobs. Setting the T parameter to a reasonable value improves job processing priority and thereby improves throughput.

Gross overestimates of time requirements can degrade processing priority. Excessively long execution time limit wastes system resources if execution enters an infinite loop.

Input/output Time Limit (IO)

The IO parameter limits the amount of time that a job spends in input and output operations. Input/output time limit is expressed in either octal or decimal seconds, depending on the installation. This time limit applies only to the input/output operations performed while the job is in execution; it does not apply to operations required to get the job into or out of central memory. The installation default for this time is usually more than sufficient for most job processing. Decreasing the value from the installation-defined default value does not significantly affect job priority; therefore, this parameter is generally not specified in the job statement.

Central Memory Storage (CM)

NOS/BE automatically assigns storage space in central memory to any incoming job on the basis of a value determined when the operating system was installed in the computer. The amount of continuous central memory space is called field length. The default field length varies from installation to installation. Except for unusual circumstances, field length need not be specified in the job statement. For directions on using this parameter, refer to the NOS/BE 1 Reference Manual.

Processing Priority (P)

When a job is submitted to the computer, it is automatically assigned a processing priority with respect to other jobs in the system. Normally, this priority is based on several factors, such as the amount of time expected to process the job, the amount of central memory needed, and the priority requested in the job statement. The use of the priority parameter in the job statement is determined by the installation; consult the computer operations center.

Tape Scheduling (MT/NT)

When a tape unit is required for job execution, indicate MT or NT in the job statement, along with the maximum number of tape units you need at one time. MT indicates 7-track tape units; NT indicates 9-track tape units.

Job Statement Terminator

All job statement parameters, except the job name, can appear anywhere in the parameter list. A separator comma or left parenthesis is necessary between parameters, and a terminating period or right parenthesis must be placed at the end. Equivalent job statement parameters are shown in the following example:

```
STATEMENT 1,T10,P6,MT2,NT1.
STATEMENT 2,NT1,MT2,P6,T10.
STATEMENT 3(T10,NT1,MT2,P6)
```

Comments

General remarks and accounting information, such as user name, department number, or special account number, can be inserted after the terminator in a job statement. Comments can begin in any column following the terminator, and can extend as far as the 80-column limit. Any characters, including the comma and period, can be used in comments. Comments are for your use or installation information only; they are printed as part of your job dayfile output, and have no effect on job processing.

Several possible job statements containing comments are shown in the following example:

```
ANY JOBNAME,T10,P6,MT1,NT3. COMMENTS.
YOUR JOB NAME,T10,NT1,MT2. DEPT. 1234,PAT.
GYHOOUAAT(T20,MT3) THIS IS A FORTRAN JOB.
```

COMPILER AND ASSEMBLER CALL STATEMENTS

Compiling or assembling programs under NOS/BE is relatively simple. Each compiler or assembler can be thought of as a file accessible to all jobs in the computer. To execute any of these files, enter a control statement with the special name of the desired compiler or assembler. The compiler or assembler reads your program as data and converts the source program to an intermediate type of machine instruction called relocatable binary code. The relocatable binary code is generally written to a disk file under the name LGO; however, the relocatable code can be written on magnetic tape for storage, or written to the PUNCHB file to produce binary card decks. The relocatable binary code is processed by a special program called the LOADER.

Various options that control operations during compilation or assembly, such as listing the program, generating cross-reference maps, and reading source code from a file other than INPUT, are available for each member of the product set. Some of these options are covered in the following explanation. The reference manual for a particular compiler or assembler contains a complete description of parameters that can appear on the control statement.

FORTRAN Extended

The control statement that calls for compilation of a FORTRAN Extended source program consists of the characters FTN, followed by an optional parameter list. Default values are assigned to any unspecified parameters. FTN control statement formats are shown in figure 3-4. The FORTRAN Extended 4 Reference Manual contains a detailed explanation of all available parameters.

COBOL 4

The control statement that calls for compilation of a COBOL 4 source program consists of the characters COBOL,

```
FTN.

FTN,I=lfn,B=lfn,R=n,OPT=m.
```

I=lfn — Specifies the file containing the source program to be compiled. The default parameter is I=INPUT. When I is specified without providing an lfn, the file named COMPILE is assumed.

B=lfn — Specifies the file on which the computer writes the binary object code. The default parameter is B=LGO. When B=0 is specified, no object code is produced.

R=n — Selects the desired reference map:

0	Produces no map
1	Produces a short map consisting of symbols, addresses, and properties
2 or R	Produced a long map, which is the same as a short map with references by line number and a DO-loop map added; this option is usually selected for debugging runs
3	Produces a long map with the addition of common block numbers and equivalence groups

The default parameter is defined by the installation.

OPT=m — Indicates the level of object code optimization:

0	Specifies fast compilation, slow execution
1	Specifies standard compilation
2	Specifies slow compilation, fast execution

Most users select OPT=0 for debugging, and OPT=1 or 2 for production runs. The default parameter is defined by the installation.

All parameters are optional.

Figure 3-4. FTN Control Statement Formats

followed by an optional parameter list enclosed in parentheses. When no parameters are given, COBOL is followed by a period. COBOL 4 control statement formats are shown in figure 3-5. Other available parameters and key names not included in the figure are explained in the COBOL 4 Reference Manual.

An example of a COBOL 4 control statement using the L default option in combination with X and R codes is:

```
COBOL(LXR)
```

```
COBOL.

COBOL(I=lfn,B=lfn,L=lfn,U)
```

I=lfn — Specifies the file containing the card images of the source program to be compiled. The default parameter is I=INPUT.

B=lfn — Specifies the file on which the compiler writes the binary object code. Object decks are obtained by specifying B=PUNCHB. The default parameter is B=LGO. When B=0 is specified, no object code is produced.

L=lfn — Specifies the file on which the computer writes the listable output and diagnostics. Four types of diagnostics are available:

C	Catastrophic; compilation terminates immediately
E	Serious errors (usually syntax); compilation continues, but the program cannot be executed
U	Unconventional errors (usually syntax); compilation continues, and the program can be executed
T	Trivial errors (usually syntax); the program can be executed

When L=0 is specified, the listing is suppressed, except for type C and E diagnostics. The normal listing includes the source program and C and E diagnostics. The letter L can be specified in combination with any or all of the following codes:

X	Extended diagnostics; includes types T and U, as well as C and E diagnostics
R	Data-name and procedure-name cross-reference list
C	Items copied from the library (see the COBOL 4 Reference Manual)
O	Object code in octal
M	Data map

The default parameter is L=OUTPUT.

U — Specifies the collating sequence. When U is specified, the compiler selects the installation-defined collating sequence (usually ASCII, rather than CDC COBOL). The default is the CDC COBOL collating sequence.

All parameters are optional.

Figure 3-5. COBOL Control Statement Formats

COBOL 5

The control statement that calls for compilation of a COBOL 5 source program consists of the characters COBOL5, followed by an optional parameter list enclosed in parentheses. When no parameters are given, COBOL5 is followed by a period. COBOL5 control statement formats are shown in figure 3-6. A complete listing and explanation of parameters and key names is contained in the COBOL 5 Reference Manual.

```
COBOL5.

COBOL5(I=lfn,B=lfn,L=lfn,LO=lopt)
```

I=lfn Specifies the file containing card images of the source program to be compiled. The default parameter is I=INPUT.

B=lfn Specifies the file on which the compiler writes the binary object code. Object decks are obtained by specifying B=PUNCHB. The default parameter is B=LGO. When B=0 is specified, no object code is produced.

L=lfn Specifies the file on which the source listing, diagnostics, and information selected by the LO parameter are written. The default parameter is L=OUTPUT. When L=0 is specified, the listing is suppressed, except for fatal diagnostics.

LO=lopt Specifies the listing options:

M A map that correlates program entities, attributes such as data class and size, and physical storage is listed

O Generated object code with COMPASS mnemonic is listed

R Cross-reference of program entities, and locations of definitions and use within the program are listed.

S Source program is listed

0 None of the information that can be be selected by O, R, M, or S is listed

Multiple options can be selected with slashes between options. The default parameter is LO=S.

All parameters are optional.

Figure 3-6. COBOL 5 Control Statement Formats

An example of a COBOL5 control statement using the default values of I, B, and L and selecting the LO = options S, M, and R is:

```
COBOL5(LO=S/R/M)
```

ALGOL

The control statement that calls for compilation of an ALGOL source program consists of the characters ALGOL, followed by an optional parameter list enclosed in parentheses. ALGOL is followed by a period when no parameters are given. Parameters can be specified by key letters only, or in the form key=lfn. Formats of the ALGOL control statement are shown in figure 3-7. A complete explanation of key letters is contained in the ALGOL 4 Reference Manual.

```
ALGOL.

ALGOL(I=lfn,B=lfn,L=lfn)
```

I=lfn Specifies the file containing the card images of the source program to be compiled. The default parameter is I=INPUT.

B=lfn Specifies the file on which the compiler writes the binary object code. This parameter must be specified if the program is to proceed immediately into execution (assuming no compiler program errors occur). The default parameter is B=LGO. No object code is produced when B=0 is specified.

L=lfn Specifies the file on which the compiler writes the listable output and diagnostics. The default parameter is L=OUTPUT. When L=0 is specified, only diagnostics are printed.

All parameters are optional.

Figure 3-7. ALGOL Control Statement Formats

An example of an ALGOL control statement that compiles from the file COMPILE, lists the source text and diagnostics on the file OUTPUT, and writes the object code on file PUNCHB (which results in punched binary cards at job termination), is:

```
ALGOL(I=COMPILE,B=PUNCHB)
```

BASIC

The control statement that calls for compilation of a BASIC source program consists of the characters BASIC, followed by an optional parameter list enclosed in parentheses. BASIC is followed by a period when no parameters are given. Formats of the BASIC control statement are shown in figure 3-8. A complete explanation of optional parameters is contained in the BASIC 3 Reference Manual.

The following example shows a BASIC control statement that compiles from file INPUT, writes execution output, source listing, and diagnostics to file OUTPUT, writes the object code to file PUNCHB, and does not execute; the program can be executed by using a PUNCHB control statement:

```
BASIC (L,B=PUNCHB)
```

COMPASS Assembler

The control statement that calls for assembly of a COMPASS source program consists of the characters COMPASS, followed by an optional parameter list enclosed in parentheses. COMPASS must be followed by a period when no parameters are given. COMPASS control statement formats are shown in figure 3-9. Other key names and parameters not included in the figure are explained in the COMPASS 3 Reference Manual.

A COMPASS control statement is shown in the following example; the source deck is on the INPUT file, full listings are on the OUTPUT file, and the object deck is punched:

```
COMPASS (B=PUNCHB)
```

```
BASIC.

BASIC(I=lfn,B=lfn,L=lfn)
```

I=lfn Specifies the file containing the card images of the source program to be compiled. The default parameter is I=INPUT.

B=lfn Specifies the file on which the compiler writes the binary object code. When B is specified, the program is compiled but not executed; it can be executed by an lfn control statement. A default file named BIN is assumed when B is specified but no lfn is provided. When B is omitted, no relocatable code is produced; the program is compiled directly to memory and executed if no compilation errors occur.

L=lfn Specifies the file on which the computer writes the source listing, diagnostics, and execution output. The default parameter is L=OUTPUT. When L is omitted, diagnostics and execution output are written to file OUTPUT and no source listing is produced.

All parameters are optional.

Figure 3-8. BASIC Control Statement Formats

```
COMPASS.

COMPASS(I=lfn,B=lfn,L=lfn)
```

I=lfn Specifies the file from which the assembler reads the source program. When I is specified without an lfn, the file named COMPILE is assumed. The default parameter is I=INPUT.

B=lfn Specifies the file on which the assembler writes the binary object code. When object decks are desired, use B=PUNCHB. The default parameter is B=LGO. When B=0 is specified, no binary output is produced.

L=lfn Specifies the file on which the assembler writes the complete listing. The default parameter is L=OUTPUT. When L=0 is specified, only brief listings are produced.

All parameters are optional.

Figure 3-9. COMPASS Control Statement Formats

LOAD AND EXECUTE CONTROL STATEMENTS

A load and execute control statement and a compiler or assembler call statement are both file execution statements and, therefore, are treated the same by the NOS/BE operating system. They both request the system to find, load, and execute the object program file of the name specified by the execution statement. For example, an FTN statement tells the system to find the file named FTN (which contains the FORTRAN Extended compiler in object program format), load it, and execute it.

The difference between the way compiler call statements and load and execute control statements are used is that a compiler call statement produces an object program from a source program, whereas a load and execute control statement rewinds, loads, and executes an object program.

Default Parameter (LGO)

When the compiler is brought into central memory because of a compiler call statement and the default file parameter is selected, the compiler reads the source program from the file INPUT, compiles it into machine language, and writes it on a file named LGO. The load and execute control statement gives the name of the file (LGO) containing the object program to be rewound, loaded, and executed. The following example shows a compiler call statement, with default parameters selected and the corresponding load and execute control statement:

COBOL. (Compiler call statement)

LGO. (Load and execute statement)

Optional Parameter

When an optional parameter is specified on the compiler call statement, the compiler writes the binary object program on the file named by the optional parameter. The load and execute control statement gives the name of the file containing the object program and causes it to be rewound, loaded, and executed. The following example shows a compiler call statement and the corresponding load and execute control statement that might be used to select an optional object program file named SAMPLE:

FTN,B=SAMPLE. (Compiler call statement)

SAMPLE. (Load and execute statement)

Binary Decks

Some terminals are able to read binary program decks. If a binary deck is the object program from a previous compilation, no compiling is done when the binary deck is read (no compiler call statement is used); therefore, the system reads the binary program into the default file INPUT. To load and execute the program, specify INPUT on the load and execute control statement. A job statement and a load and execute control statement that you could use to read, load, and execute a binary program are shown in figure 3-10. A binary program in card form must be terminated by two consecutive 7/8/9 cards if another program or data is to follow. When the binary program is the last section in the deck, a 6/7/8/9 card is used instead of two 7/8/9 cards.

Permanent File

A previously compiled program might be stored as a permanent file. To execute a permanent file program (it is already in binary object code), you must first make it available to your job with the ATTACH statement. Once the program file is attached, load and execute it by specifying the file name in the load and execute control statement. An example of control statements to attach, load, and execute a job that has been made a permanent file is shown in figure 3-11.

FILE CONTROL STATEMENTS

Additional control statements that you can use to manipulate files are described in section 4.

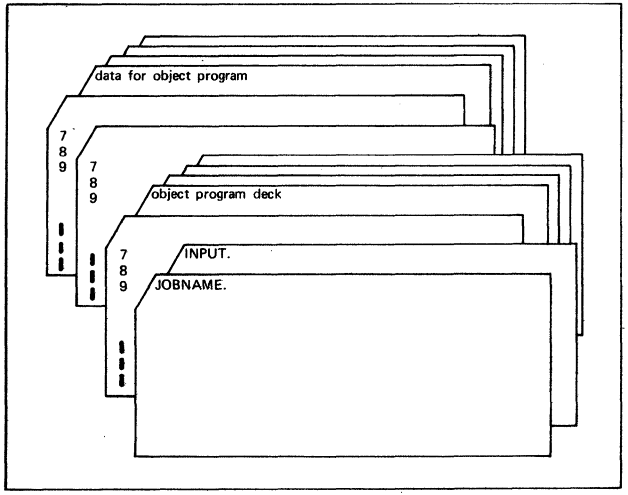


Figure 3-10. Statements to Read, Load, and Execute a Binary Deck

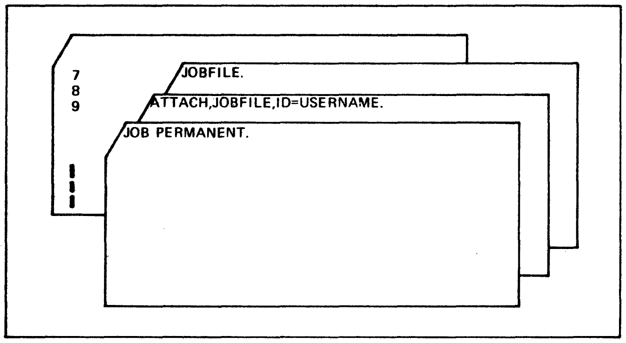


Figure 3-11. Statements to Attach, Load, and Execute a Permanent File

PROGRAM SECTION

The program section of your job deck contains the program cards. When the job contains more than one individually compiled program, they make up separate program sections. A program used in the program section can be either a source program or an object program. A source program is used as data by the compiler; an object program is read by the card reader, loaded into memory, and executed without having to be compiled.

Source program construction depends on the programming language being used; refer to the proper programming language reference manual for a description of program deck format.

A 7/8/9 multiple-punched separator card is required after each program section. A representative job deck that could be used to run a source program is shown in figure 3-12. A typical job using an object deck is shown in figure 3-13. Neither program used in the examples requires data for execution.

DATA SECTION

Data required by your source or object program is contained in the data section. A main program or subroutine can be associated with more than one data section. Each data section must be terminated with a 7/8/9 multiple-punched card. An example of deck structure for a source program requiring data is shown in figure 3-14. Additional examples are shown in appendix C.

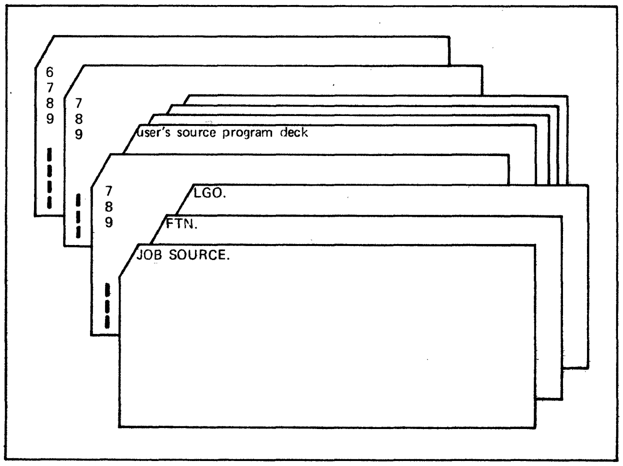


Figure 3-12. Sample Source Deck Format

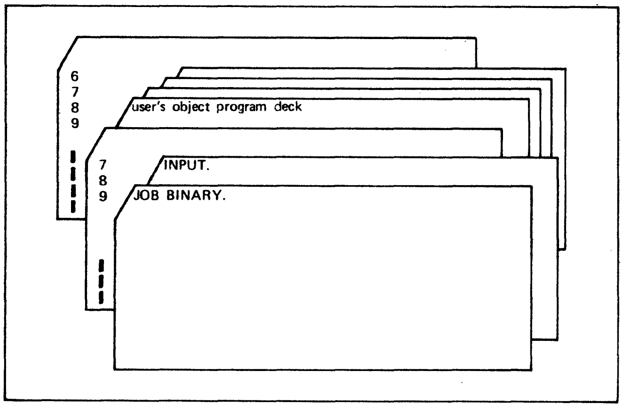


Figure 3-13. Sample Object Deck Format

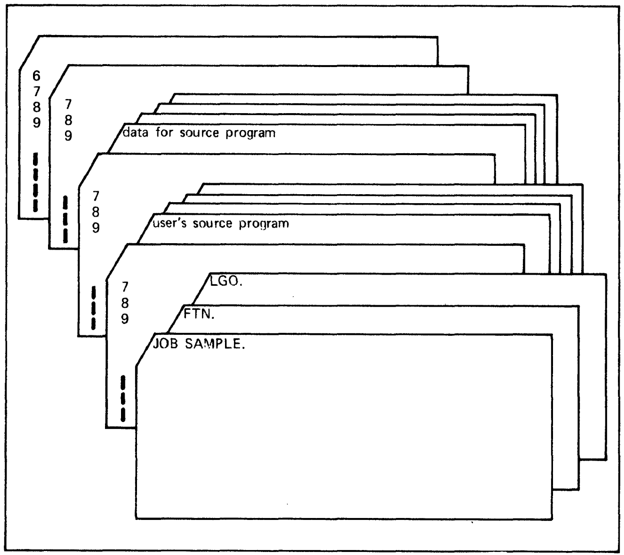


Figure 3-14. Sample Deck Format for Source Program and Data

FILE OPERATION 4

Files residing in the system input, output, and execution queues can be manipulated from a remote batch terminal by using INTERCOM commands. Files used by a job during execution can be manipulated by using system control statements. This section demonstrates how to use INTERCOM remote batch commands and system control statements to manipulate files.

INTERCOM COMMANDS

Jobs submitted from remote batch terminals reside in the system input queue. At execution time, they are placed in the execution queue, and output from their execution is placed in the output and punch queues. Even though a job is under operating system control while in the queues, the terminal-id continues to be associated with the job unless the terminal or central site operator specifically breaks that association with a DIVERT command. As a result, a terminal that disconnects or goes off-line can regain access to the files at a later time by reestablishing communications with INTERCOM; the terminal-id is still associated with the files. The commands described in the following paragraphs can be entered from a remote batch terminal to control the disposition of jobs and files associated with the terminal.

SITUATE

Before a job or file can be diverted to another terminal or user, you must know the id of that terminal or user.

When you are logged in, the SITUATE command can be used to obtain a list of currently logged-in users. Dedicated-line remote batch terminals, whether logged-in or not, are included in the list. SITUATE is an interactive command and, therefore, cannot be used unless the terminal is logged in.

To display a list of terminal and user ids similar to that shown in figure 4-1, enter:

```
SITUATE
```

```
USERS WITH SAME PASSWORD

    BA-USERA

OTHERS

    BC-OPS          CX-USERB        *BZ-USERC

    HB-PSDRPE       BD-HSBT

BATCH TERMINALS

    AS-200 UT       HB-71X          AF-XSBT
```

Figure 4-1. SITUATE Display

If your password is restricted, users sharing your password are listed first when you enter the SITUATE command, and other user names follow. If your password is unrestricted, you receive only the names of users who share that password. An asterisk preceding the user-id indicates that the terminal has been locked (by the LOCK command) to prevent the displaying of messages from other terminals.

Figure 4-1 shows that BA is the user-id of the requestor. BC, HB, CX, BD, and BZ are other logged-in users. USERC (user-id BZ) has locked the terminal against SEND utility routine messages being sent from another terminal. The terminal-ids (AS, HB, and AF) listed under BATCH TERMINALS identify either hard-wired batch terminals or dial-up terminals that are operating in batch mode. Terminal types represented by XSBT include the high-, medium-, and low-speed batch terminals (HSBT, MSBT, and LSBT); 71X represents a 711 or 714 terminal.

When a user at a dial-up terminal enters batch mode, the user-id is displayed with both the password and the batch terminal type. For example, the user shown in figure 4-1 that is logged-in with password PSDRPE is assigned user-id HB. HB is displayed with the password (PSDRPE) under OTHER because the password is not the same as the requestor. When user HB enters batch mode, HB is also displayed with the terminal type (71X) under BATCH TERMINALS. HB-71X indicates that the user with the user-id HB is using either a 711 or 714 terminal for batch processing.

DIVERT

Files and jobs associated with your terminal can be assigned to another user or location by using the DIVERT command. When a job or file is diverted to another location, its files are no longer accessible to the terminal issuing the command. The format of the DIVERT command is:

```
DIVERT,[jobname],[id],[q],[DEF]
```

The jobname identifies the job. The terminal-id or user-id to be assigned to the job output is indicated by the id parameter. The output type (Print or Punch) is specified by the q parameter. DEF causes all subsequent print or punch files to be diverted as soon as they become available.

To divert the output for job EXAMPAB to the central site, enter:

```
DIVERT,EXAMPAB
```

The job name can be abbreviated to the last two characters; therefore, you could enter:

```
DIVERT,AB
```

When no id and q parameters are specified, all input, output, and punch files associated with job EXAMPAB are diverted to the central site. After the DIVERT,AB command has been entered, the input and output files for the job are associated with the central site. All output is processed at the central site.

To transfer (to the central site) all input (I), print (O), and punch (P) files currently associated with your terminal, enter:

```
DIVERT
```

When jobname is not specified, all jobs at your terminal are diverted. The central site is the default for the id parameter. All queues are diverted when no q is specified.

To divert the output for job EXAMPAA to terminal AF, enter:

```
DIVERT,AA,AF
```

The output from job EXAMPAA is diverted to the terminal identified in the SITUATE display as AF. Files associated with job EXAMPAA are no longer accessible to you after the DIVERT command is entered.

To divert the output for job TESTD01 to user BD, enter:

```
DIVERT,01,BD
```

All output from job TESTD01 is diverted to the user identified in the SITUATE display as BD.

To divert only the punch output for job SAMPLA6 to the central site, enter:

```
DIVERT,A6,,P
```

Because the id parameter is not specified, the punch file is diverted to the central site by default. To divert the punch output for job SAMPLA6 to terminal AF, enter:

```
DIVERT,A6,AF,P
```

To transfer all output files from your terminal to terminal AC, enter:

```
DIVERT,,AC
```

Because no job name is specified, all job files from your terminal are diverted to terminal AC.

To divert (to the central site) all current punch files generated at your terminal, enter:

```
DIVERT,,,P
```

To divert (to the central site) all current and future punch files generated at your terminal, enter:

```
DIVERT,,,P,DEF
```

The DEF option causes all subsequent punch files to be diverted as soon as they become available.

All current and future print files could be diverted to terminal AS by entering:

```
DIVERT,,AS,O,DEF
```

The DEF option cannot be selected with a q entry of I, or in a DIVERT command containing a job name.

To transfer (to the central site) all current and future output associated with your terminal, enter:

```
DIVERT,,,,DEF
```

No job name or q is specified; therefore, all files associated with the terminal are diverted. Because the central site is selected when no id is specified, all output for the terminal is diverted to the central site. The DEF option causes all files, present and future, to be diverted until a REVERT command cancels the effect of the DEF option.

REVERT

To cancel the effect of a DIVERT command entered with the DEF option selected, enter:

```
REVERT
```

Subsequent output files generated by jobs or files associated with your terminal are returned to your terminal for output. Files already diverted are not returned.

To cancel the effect of the DEF option on only the print or punch queue, enter:

```
REVERT,O        (to cancel the divert on the print
                queue)
```

or

```
REVERT,P        (to cancel the divert on the punch
                queue)
```

The effect of the DIVERT command is negated, and subsequent punch or print output is no longer diverted.

PRIOR

The PRIOR command is used to change the priority of an output file that is not currently active. The format of the PRIOR command is:

```
PRIOR,jobname,priority,(q)
```

To change the priority of a print file named JOBTEX7 to 100_8, enter:

```
PRIOR,JOBTEX7,100,O
```

To change the priority of a punch file named FILE207 to 745_8, enter:

```
PRIOR,07,745,P
```

To change the priority of all print and punch files for JOBAA22 to 172_8, enter:

```
PRIOR,JOBAA22,172
```

When the queue is omitted, all output files associated with the job change priority. Jobs with priority 0000 remain in the queue; they are not punched or printed.

EVICT

Files associated with a remote batch job can be eliminated from the system input (I), print (O), and punch (P) queues by using the EVICT command. To evict all files associated with job TESTX2T from all three queues, enter either of the following:

```
EVICT,TESTX2T
EVICT,2T
```

Note that the job name can be abbreviated to the last two characters.

To eliminate all punch files for job TESTX2F, enter:

```
EVICT,2F,P
```

To eliminate all print files for job TESTX2F, enter:

```
EVICT,2F,O
```

To eliminate all input files for job TESTX2F, enter:

```
EVICT,TESTX2F,I
```

DROP

A job in execution is terminated, and the output files already generated by the job are placed in the OUTPUT queue, by entering the DROP command. To drop job ANYJO7X from execution, enter either of the following:

```
DROP,ANYJO7X

DROP,7X
```

The remaining output file could be eliminated by using the EVICT command.

KILL

The effects of the DROP and EVICT commands are combined in the KILL command. To terminate execution of job ABCDEXY and eliminate all output from the system, enter either of the following:

```
KILL,ABCDEXY
KILL,XY
```

No output or dayfile is produced for job ABCDEXY.

FILE CONTROL STATEMENTS

Files associated with a remote batch job can be manipulated during job execution by placing file control statements in the control statement section of the job deck. The order in which the file control commands described in the following paragraphs are presented illustrates a sequence that could be used for job processing. It is unlikely that all of these statements would normally be used in one job. The order in which some of the commands are used will vary with different file manipulation operations. No attempt is made here to describe all of the control statement options available; refer to the NOS/BE 1 Reference Manual or NOS/BE 1 User's Guide for more detailed information.

A FORTRAN program, called SAMPLE, which computes the sum and average of input data values, is used as the sample job for the following control statement descriptions. Only the program card is shown to represent the SAMPLE program. The name of the job punched on the job card is ANY JOB NAME. The job produces output, which is to be both printed and punched.

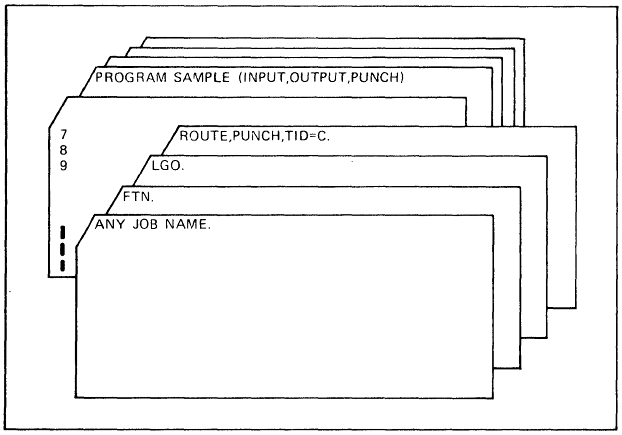
ROUTE

Output files generated by a remote batch job can be directed to another terminal, or to the central site, by using the ROUTE control statement. Because many remote batch terminals do not include a card punch, it is frequently necessary to route card punch output to a site equipped with a card punch.

To have card punch output (contained in the punch file) punched at the central site, place a ROUTE statement in the control statement section of your job deck, as shown in figure 4-2.

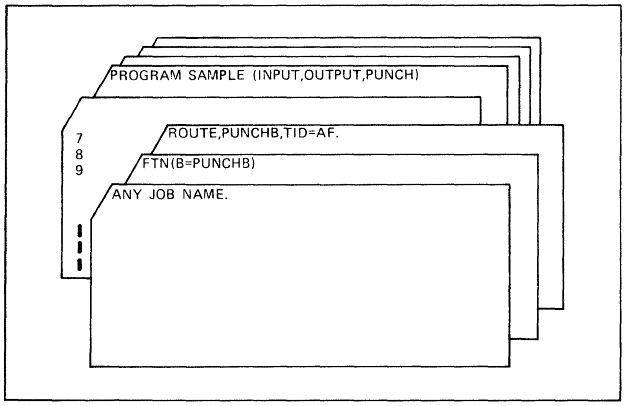
To punch a binary object program deck on the card punch at remote batch terminal AF, place the statements shown in figure 4-3 in the control statement section of your job deck.

The sample program will be compiled but not executed because no load and execute (PUNCHB) statement is included in the control statement section.



```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
ROUTE,PUNCH,TID=C.
LGO.
FTN.
ANY JOB NAME.
```

Figure 4-2. ROUTE to Central Site



```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
ROUTE,PUNCHB,TID=AF.
FTN(B=PUNCHB)
ANY JOB NAME.
```

Figure 4-3. ROUTE to Terminal AF

REQUEST

Files created by a job during execution normally disappear when, or a short time after, the job that created them completes execution. It is sometimes desirable to retain a file in the system, on a mass storage device, for future use. A file that exists longer than the life of a job is called a permanent file.

Suppose you want to run job ANY JOB NAME daily. Each day you will change only the data used. Rather than reading the program deck every day, you could store the compiled binary object program on a permanent file. Then, each time the job is to be run, it is necessary only to access the permanent job file and submit a new data deck.

Before a file can be made permanent, it must be written on a mass storage unit that has been designated as a permanent file device. One way to make sure that the object program is written on a permanent file device is to place a REQUEST statement before the compiler call statement in the job deck, as shown in figure 4-4. The *PF parameter included in the REQUEST statement shown in figure 4-4 specifies that the file LGO is to be written on a permanent file device. When a file name other than the default file name (LGO) is specified in the compiler call statement, that file name must appear on the REQUEST statement, as shown in figure 4-5. The program shown in figures 4-4 and 4-5 will not be executed because no load and execute statement is included in the control statement section.

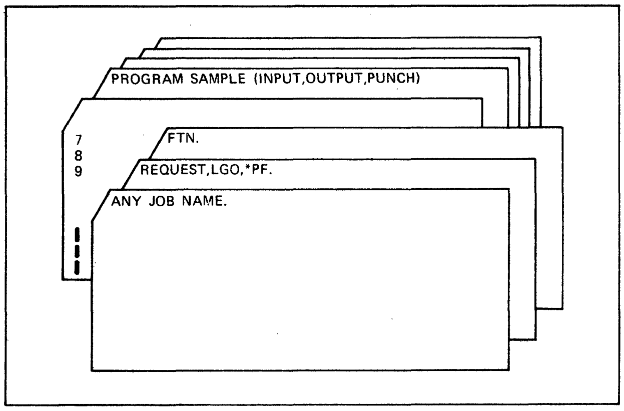

```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
FTN.
REQUEST,LGO,*PF.
ANY JOB NAME.
```

Figure 4-4. REQUEST, LGO Statement Example

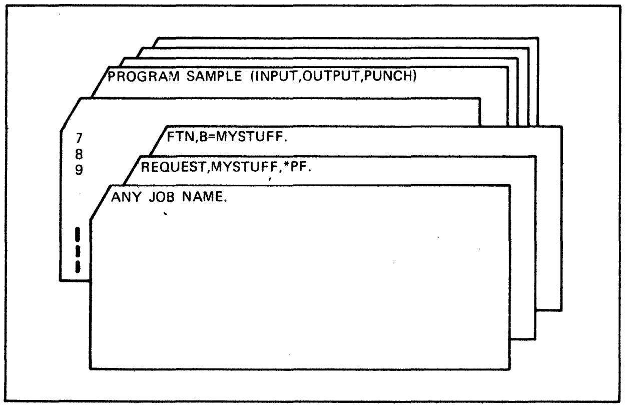

```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
FTN,B=MYSTUFF.
REQUEST,MYSTUFF,*PF.
ANY JOB NAME.
```

Figure 4-5. REQUEST Statement with User-Assigned File Name

A file does not automatically become permanent when it is written on a permanent file device; writing it on a permanent file device only assures that the file can be made permanent. Another control statement (CATALOG) must be exeuted to create a permanent file. The CATALOG control statement is explained later in this section.

COPY

Another way to make sure that a file is written on a permanent file device is to copy the local file to a permanent file device. An example of a control statement section that creates a file named NEWFILE on a permanent file device, copies the default load file (LGO) to the permanent file device and then executes the program, is shown in figure 4-6.

REWIND

REWIND positions a file at the beginning-of-information. In many cases, when a file is requested for a job, the file is automatically positioned at the beginning-of-information. However, because of variations in installation parameters and procedures, automatic positioning does not always occur; therefore, as a matter of standard practice, it is best to rewind files before use.

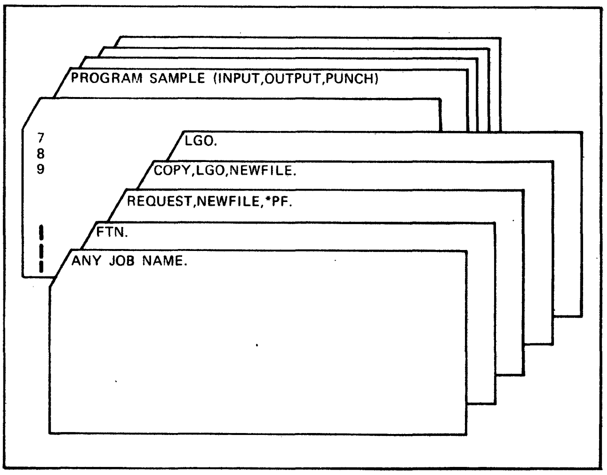

```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
LGO.
COPY,LGO,NEWFILE.
REQUEST,NEWFILE,*PF.
FTN.
ANY JOB NAME.
```

Figure 4-6. Example of REQUEST and COPY Statements

When the COPY statement is placed after the load and execute statement in the control statement section, the load file must be rewound before it is copied. A control statement section that includes a REWIND statement is shown in figure 4-7.

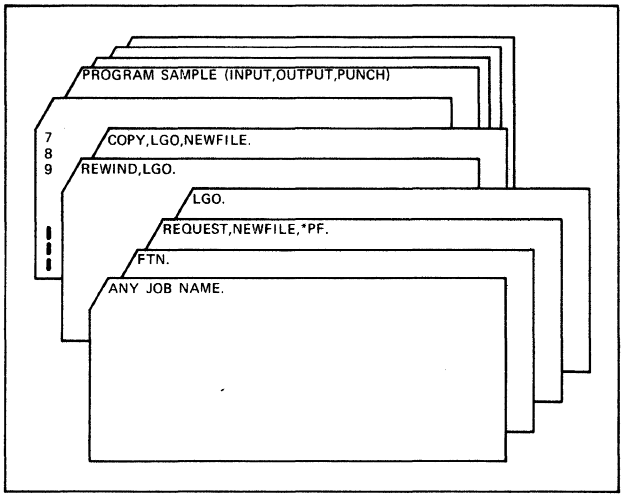

```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
COPY,LGO,NEWFILE.
REWIND,LGO.
LGO.
REQUEST,NEWFILE,*PF.
FTN.
ANY JOB NAME.
```

Figure 4-7. REWIND Statement Example

CATALOG

CATALOG makes an existing local file permanent, provided the file resides on a permanent file device. This process, called cataloging, results in the system recording information about the file in a system table known as the permanent file directory. Figures 4-4 through 4-7 show ways in which a file could be written to a permanent file device. The CATALOG control statement is used to make those files permanent.

A suggested format of the CATALOG control statement is:

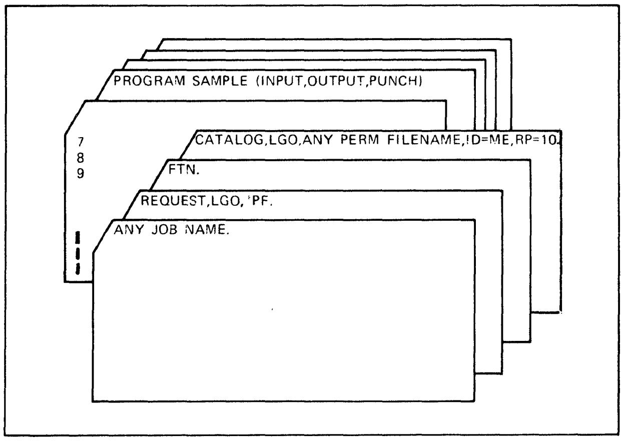
```
CATALOG,lfn,pfn,ID=id,RP=n.
```

The lfn parameter is the logical file name by which the file is known to the job; it must be the same as the name used on the REQUEST control statement to request a permanent file device. The lfn parameter is the first parameter on the CATALOG control statement.

The pfn parameter is the permanent file name, which is the name the file is to have in the system permanent file tables. A pfn parameter can specify the same name as the lfn parameter. Permanent file names can contain up to 40 letters or digits and need not begin with a letter. When a pfn parameter does not follow the lfn parameter, the system uses the logical file name as the permanent file name.

Parameters after pfn can appear in any order. The ID parameter is required at most installations, but the system might supply it. The RP parameter with a value other than zero specifies the number of days the file is to remain in the system. Some computer centers might purge your file if an RP parameter is not used.

To make the LGO file created in figure 4-4 permanent, place the CATALOG statement after the FTN statement, as shown in figure 4-8.



```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
7
8
9
CATALOG,LGO,ANY PERM FILENAME,ID=ME,RP=10
FTN.
REQUEST,LGO,*PF.
ANY JOB NAME.
```

Figure 4-8. CATALOG, LGO Statement

Although the permanent file name (ANY PERM FILENAME) is not required, it is recommended that you assign a permanent file name when cataloging a file named LGO; otherwise, the permanent file will be named LGO. If another job, while using the default load and execute parameter (LGO), tries to attach the permanent file named LGO without supplying a pfn, an error condition will exist and the file will not be attached. A job cannot have two files with the same name attached at one time.

To catalog the file created in figure 4-5, it is acceptable to omit the pfn parameter. A CATALOG statement, in which the logical file name is used for the permanent file name, is shown in figure 4-9. This statement should be placed after the FTN statement to make file MYSTUFF permanent.

To make the file NEWFILE created in figure 4-7 permanent, a CATALOG control statement is added to the control statement section, as shown in figure 4-10. The local file NEWFILE, which was written on a permanent file device, is given the permanent file name YOUR FTN JOB. Any reference to the permanent file just cataloged must be by the name YOUR FTN JOB.

ATTACH

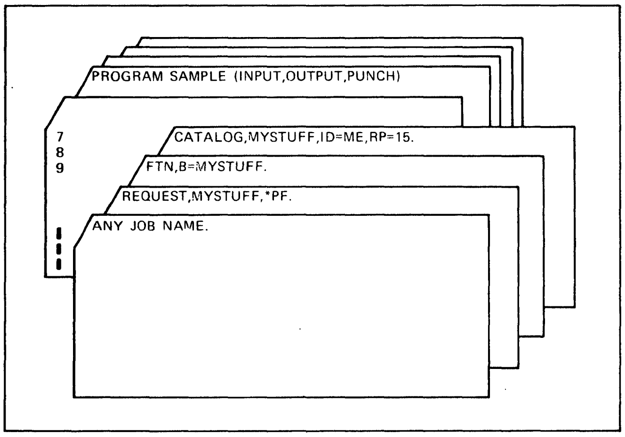
Now that you have created a permanent file, you will probably want to use it. The ATTACH control statement is used to access a previously created permanent file. Attaching a permanent file results in a local file with the status of an attached permanent file. A permanent file must be attached before the file is referenced by another control statement in the job. A suggested ATTACH statement format is:

ATTACH,lfn,pfn,ID=id.

The lfn parameter can be any name that begins with a letter and is made up of one to seven letters or digits. The lfn is the name that identifies the file to the job; it has nothing to do with the lfn used when the file was created or cataloged.

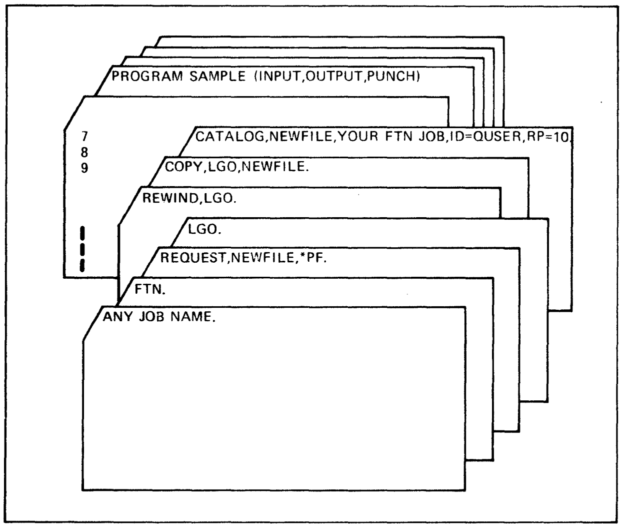
When a file is being attached, the pfn parameter must duplicate the permanent file name assigned when the file was cataloged. The ID parameter used in the ATTACH statement must be the same ID used when the file was cataloged.

If the permanent file name is a legal logical file name (begins with a letter and consists of one to seven letters or digits), you can use the permanent file name as the lfn parameter and omit the pfn parameter. When the pfn parameter is omitted, the permanent file name is assumed to be the same as the logical file name.



```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
7
8
9
CATALOG,MYSTUFF,ID=ME,RP=15.
FTN,B=MYSTUFF.
REQUEST,MYSTUFF,*PF.
ANY JOB NAME.
```

Figure 4-9. CATALOG Statement Using Default pfn



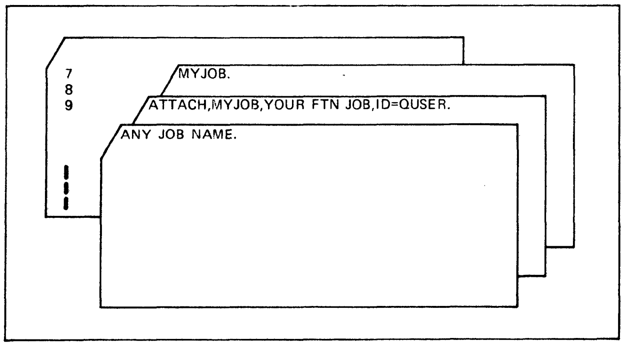
```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
7
8
9
CATALOG,NEWFILE,YOUR FTN JOB,ID=QUSER,RP=10
COPY,LGO,NEWFILE.
REWIND,LGO.
LGO.
REQUEST,NEWFILE,*PF.
FTN.
ANY JOB NAME.
```

Figure 4-10. Control Statement Section After Adding the CATALOG Statement

The following example shows a statement that attaches the permanent file named AJOB5 with the logical file name AJOB5; QUSER is the ID that was used when the file was cataloged:

```
ATTACH, AJOB5, ID=QUSER.
```

To use the FORTRAN program previously written on a permanent file and cataloged as YOUR FTN JOB, the control statement section shown in figure 4-11 could be used. MYJOB is the logical file name of the object program file. It is loaded and executed by the statement MYJOB.



```
MYJOB.
ATTACH,MYJOB,YOUR FTN JOB,ID=QUSER.
ANY JOB NAME.
```

Figure 4-11. Control Statement Section to Attach and Execute a Permanent File

AUDIT

Information about permanent files, such as the expiration date of the retention period, can be obtained by using the AUDIT utility.

To audit all permanent files cataloged with your ID parameter, enter:

```
AUDIT,AI=P,ID=id.
```

where AI is a mnemonic for audit information, and P implies you only want a partial list.

To audit the permanent file created in figure 4-10, the job shown in figure 4-12 could be submitted. This job produces, on the file OUTPUT, the output shown in figure 4-13.

The information produced by AUDIT gives a history of file creation and use. The header of the output listing shows the date and time the audit was made. Other columns of the listing provide the following information:

① Owner is the value from the ID parameter of CATALOG.

② Permanent File Name is the value from the pfn parameter of CATALOG.

③ Cycle is the value from the CY parameter of CATALOG; or, if it was omitted, the value assigned by the system.

④ Account is the value from the AC parameter of CATALOG; it might be assigned automatically from other accounting information within your job.

⑤ VSN is the volume serial number of the disk pack on which the file resides. The pack is a member of the device set identified by the SETNAME heading. If you are using a private device set, VSN is the value from a VSN parameter and the set name is the value from an SN parameter.

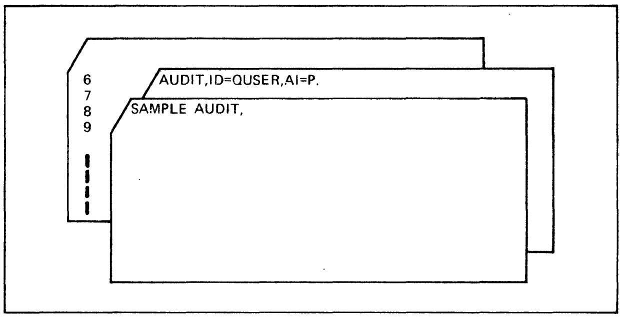
⑥ PRUs is the size of the file measured in the system unit known as a PRU (physical record unit). A PRU is 64 central memory words.

⑦ Creation is the date the file was cataloged.

⑧ Expiration is the date the retention period expires. The system calculates this date from the creation date and the value of the RP parameter.

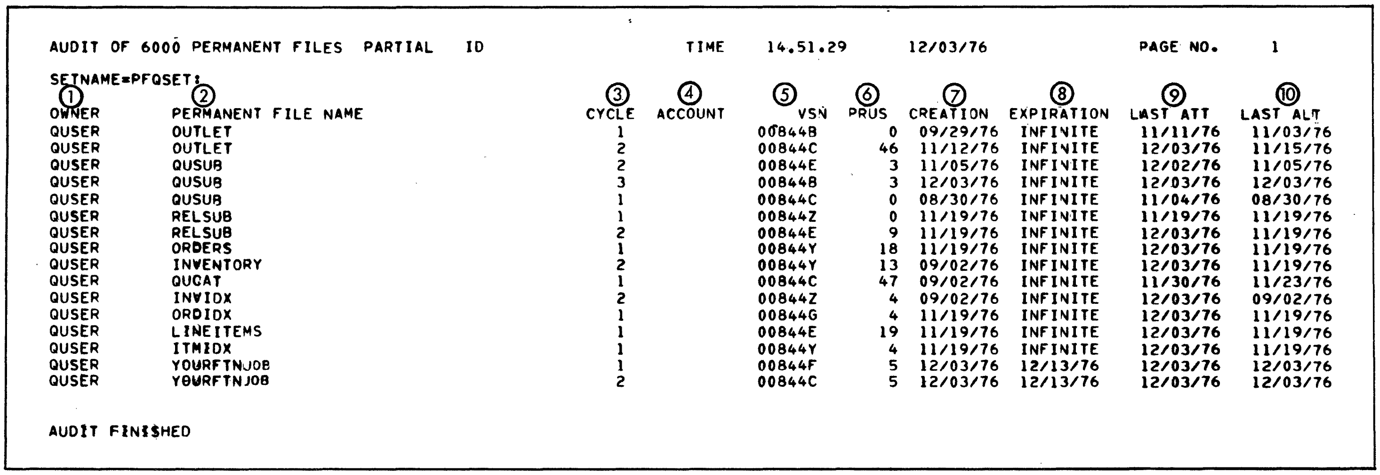
⑨ Last Attach is the date the file was last used.

⑩ Last Alter is the date the file contents were last changed, or the date the file was created.



```
AUDIT,ID=QUSER,AI=P.
SAMPLE AUDIT,
```

Figure 4-12. Audit Program



```
AUDIT OF 6000 PERMANENT FILES  PARTIAL   ID                     TIME    14.51.29      12/03/76             PAGE NO.     1

SETNAME=PFQSET:
 ①               ②                                                ③      ④          ⑤      ⑥       ⑦          ⑧          ⑨          ⑩
OWNER           PERMANENT FILE NAME                             CYCLE  ACCOUNT       VSN  PRUS  CREATION  EXPIRATION  LAST ATT   LAST ALT
QUSER           OUTLET                                            1               00844B     0  09/29/76  INFINITE    11/11/76   11/03/76
QUSER           OUTLET                                            2               00844C    46  11/12/76  INFINITE    12/03/76   11/15/76
QUSER           QUSUB                                             2               00844E     3  11/05/76  INFINITE    12/02/76   11/05/76
QUSER           QUSUB                                             3               00844B     3  12/03/76  INFINITE    12/03/76   12/03/76
QUSER           QUSUB                                             1               00844C     0  08/30/76  INFINITE    11/04/76   08/30/76
QUSER           RELSUB                                            1               00844Z     0  11/19/76  INFINITE    11/19/76   11/19/76
QUSER           RELSUB                                            2               00844E     9  11/19/76  INFINITE    12/03/76   11/19/76
QUSER           ORDERS                                            1               00844Y    18  11/19/76  INFINITE    12/03/76   11/19/76
QUSER           INVENTORY                                         2               00844Y    13  09/02/76  INFINITE    12/03/76   11/19/76
QUSER           QUCAT                                             1               00844C    47  09/02/76  INFINITE    11/30/76   11/23/76
QUSER           INVIDX                                            2               00844Z     4  09/02/76  INFINITE    12/03/76   09/02/76
QUSER           ORDIDX                                            1               00844G     4  11/19/76  INFINITE    12/03/76   11/19/76
QUSER           LINEITEMS                                         1               00844E    19  11/19/76  INFINITE    12/03/76   11/19/76
QUSER           ITMIDX                                            1               00844Y     4  11/19/76  INFINITE    12/03/76   11/19/76
QUSER           YOURFTNJOB                                        1               00844F     5  12/03/76  12/13/76    12/03/76   12/03/76
QUSER           YOURFTNJOB                                        2               00844C     5  12/03/76  12/13/76    12/03/76   12/03/76


AUDIT FINISHED
```

Figure 4-13. Permanent File Audit Output

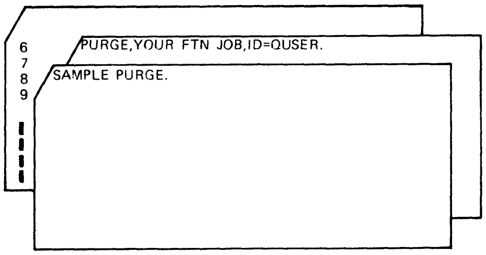
PURGE

To remove a file from permanent file status, place a PURGE control statement in the control statement section of your job deck. Once the PURGE statement is executed, the system tables associated with the file are deleted. The file itself remains attached to the job; however, once the job terminates, the file disappears.

A permanent file can be purged after it has been attached, but a permanent file does not have to be attached to be purged. An attached permanent file, with the logical file name MINE, could be purged by using the control statement shown in the following example:

```
PURGE, MINE.
```

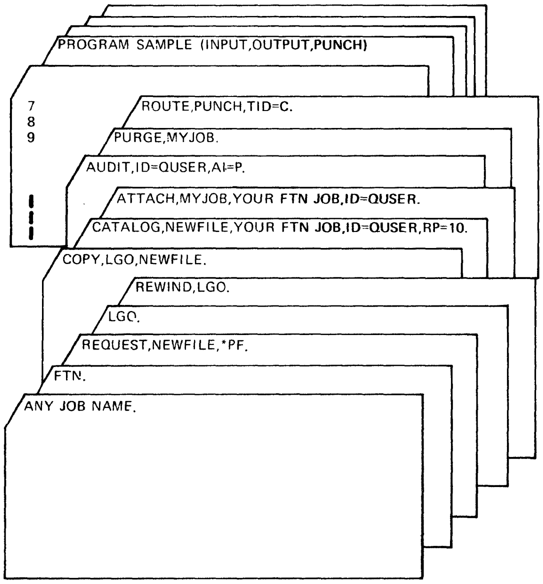
To eliminate file YOUR FTN JOB from the catalog of permanent files shown in figure 4-13, the job shown in figure 4-14 could be used. The ID=QUSER parameter is required on the purge statement shown in this example because QUSER is the ID that was used when the file was cataloged in figure 4-10. Note that the file is not attached before it is purged.



```
PURGE,YOUR FTN JOB,ID=QUSER.
SAMPLE PURGE.
```

Figure 4-14. Job to Purge Unattached Permanent File

All of the file control statements described in this section are shown in the control statement section pictured in figure 4-15. Each control statement is executed as it is encountered. No permanent file is retained after the job completes execution because the purge statement eliminates the file from the catalog of permanent files.



```
PROGRAM SAMPLE (INPUT,OUTPUT,PUNCH)
ROUTE,PUNCH,TID=C.
PURGE,MYJOB.
AUDIT,ID=QUSER,AI=P.
ATTACH,MYJOB,YOUR FTN JOB,ID=QUSER.
CATALOG,NEWFILE,YOUR FTN JOB,ID=QUSER,RP=10.
COPY,LGO,NEWFILE.
REWIND,LGO.
LGO.
REQUEST,NEWFILE,*PF.
FTN.
ANY JOB NAME.
```

Figure 4-15. Control Statement Section

STANDARD CHARACTER SETS A

CONTROL DATA operating systems offer the following variations of a basic character set:

- CDC 64-character set
- CDC 63-character set
- ASCII 64-character set
- ASCII 63-character set

The set in use at a particular installation was specified when the operating system was installed.

Depending on another installation option, the system assumes an input deck has been punched either in O26 or in O29 mode (regardless of the character set in use). Under NOS/BE the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of the job statement or any 7/8/9 card. The specified mode remains in effect through the end of the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

Graphic character representation appearing at a terminal or printer depends on the installation character set and the terminal type. Characters shown in the CDC Graphic column of the standard character set (table A-1) are applicable to BCD terminals; ASCII graphic characters are applicable to ASCII-CRT terminals.

Table A-1. STANDARD CHARACTER SET

CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code	CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code
:†	:	00††	8-2	00	8-2	072	6	6	41	6	06	6	066
A	A	01	12-1	61	12-1	101	7	7	42	7	07	7	067
B	B	02	12-2	62	12-2	102	8	8	43	8	10	8	070
C	C	03	12-3	63	12-3	103	9	9	44	9	11	9	071
D	D	04	12-4	64	12-4	104	+	+	45	12	60	12-8-6	053
E	E	05	12-5	65	12-5	105	–	–	46	11	40	11	055
F	F	06	12-6	66	12-6	106	*	*	47	11-8-4	54	11-8-4	052
G	G	07	12-7	67	12-7	107	/	/	50	0-1	21	0-1	057
H	H	10	12-8	70	12-8	110	(	(	51	0-8-4	34	12-8-5	050
I	I	11	12-9	71	12-9	111	)	)	52	12-8-4	74	11-8-5	051
J	J	12	11-1	41	11-1	112	$	$	53	11-8-3	53	11-8-3	044
K	K	13	11-2	42	11-2	113	=	=	54	8-3	13	8-6	075
L	L	14	11-3	43	11-3	114	blank	blank	55	no punch	20	no punch	040
M	M	15	11-4	44	11-4	115	, (comma)	, (comma)	56	0-8-3	33	0-8-3	054
N	N	16	11-5	45	11-5	116	. (period)	. (period)	57	12-8-3	73	12-8-3	056
O	O	17	11-6	46	11-6	117	≡	#	60	0-8-6	36	8-3	043
P	P	20	11-7	47	11-7	120	[	[	61	8-7	17	12-8-2	133
Q	Q	21	11-8	50	11-8	121	]	]	62	0-8-2	32	11-8-2	135
R	R	22	11-9	51	11-9	122	%	%	63††	8-6	16	0-8-4	045
S	S	23	0-2	22	0-2	123	≠	" (quote)	64	8-4	14	8-7	042
T	T	24	0-3	23	0-3	124	→	_ (underline)	65	0-8-5	35	0-8-5	137
U	U	25	0-4	24	0-4	125	∨	!	66	11-0 or 11-8-2†††	52	12-8-7 or 11-0†††	041
V	V	26	0-5	25	0-5	126							
W	W	27	0-6	26	0-6	127	∧	&	67	0-8-7	37	12	046
X	X	30	0-7	27	0-7	130	↑	' (apostrophe)	70	11-8-5	55	8-5	047
Y	Y	31	0-8	30	0-8	131	↓	?	71	11-8-6	56	0-8-7	077
Z	Z	32	0-9	31	0-9	132	<	<	72	12-0 or 12-8-2†††	72	12-8-4 or 12-0†††	074
0	0	33	0	12	0	060							
1	1	34	1	01	1	061	>	>	73	11-8-7	57	0-8-6	076
2	2	35	2	02	2	062	≤	@	74	8-5	15	8-4	100
3	3	36	3	03	3	063	≥	\	75	12-8-5	75	0-8-2	134
4	4	37	4	04	4	064	¬	⌢(circumflex)	76	12-8-6	76	11-8-7	136
5	5	40	5	05	5	065	; (semicolon)	; (semicolon)	77	12-8-7	77	11-8-6	073

†Twelve or more zero bits at the end of a 60-bit word are interpreted as end-of-line mark rather than two colons. End-of-line mark is converted to external BCD 1632.

††In installations using a 63-graphic set, display code 00 has no associated graphic or card code; display code 63 is the colon (8-2 punch). The % graphic and related card codes do not exist and translations from ASCII/EBCDIC % yield a blank (55$_8$).

†††The alternate Hollerith (026) and ASCII (029) punches are accepted for input only.

GLOSSARY

ASSEMBLE - To transform a COMPASS language program into a form that the computer can execute directly. During assembly, machine language operation codes are substituted for COMPASS codes and machine addresses for symbolic addresses. COMPASS performs the assembly.

ATTACH - The control statement that allows a job to gain access to a permanent file. The type of access can be controlled by passwords on the CATALOG control statement that makes the file permanent.

BANNER PAGE - A page printed at the beginning of a print file, showing the name of the file in large letters.

BATCH MODE - The state of the terminal during which batch data is transmitted from the terminal's card reader to central files and/or from central files to the terminal's line printer or card punch.

BEGINNING-OF-INFORMATION (BOI) - The start of the first user record in a file. System-supplied information, such as an index block or control word, does not affect beginning-of-information.

BINARY-CODED CARD - A card containing punches that correspond directly to bit representation within central memory.

BUFFER - An intermediate storage area used to compensate for a difference in rates of data flow, or times of event occurrence, when transmitting data between central memory and an external device during input/output operations.

BYTE - A group of bits. Unless prefixed (for example, a 6-bit byte), the term implies 8-bit groups. When used for encoding character data, a byte represents a single character.

CARRIAGE CONTROL - The control exercised over the format of printed output. The leftmost character of the data line to be printed is the carriage control character. Carriage control is also called format control.

CATALOG - The control statement that makes a file permanent by entering location, identification, and other pertinent information into a permanent file directory maintained by the system so that the file is always known and readily available for use.

CENTRAL FILE - A file whose name and attributes are maintained by the operating system at the host computer.

CENTRAL MEMORY - The area of the computer in which all programs reside when they are being executed. A program in central memory has access to the central processor.

CENTRAL PROCESSOR UNIT (CPU) - The high-speed arithmetic unit that performs the addition, subtraction, multiplication, division, incrementing, logical operations, and branching instructions needed to execute programs.

CHARACTER - Any alphabetic, numeric, or special symbol that can be represented by a 6-bit display code.

COBOL (Common Business Oriented Language) - The COBOL compiler or the COBOL programming language. Also, the control statement used to compile a program written in the COBOL programming language.

COMMENT - A remark inserted within a program or job. Comments can follow the terminator in control statements.

COMPASS - The standard assembly language used with the CDC CYBER 170, CDC CYBER 70, and CDC 6000 series computer systems. Also, the control statement request used to assemble a program written in the COMPASS assembly language.

COMPILE - To prepare a machine language program from a program written in another programming language.

CONTROL POINT - A number to which a job is assigned when it enters execution; system resources are allocated to the job and it becomes eligible to timeshare the central processor. Control points are the concept by which the multiprogramming capability of the computer is utilized.

CONTROL STATEMENTS - The cards comprising the first section within a job. These statements tell the operating system how the job is to be processed and which files are to be referenced.

CONTROL STATEMENT SECTION - The first part of a job deck that contains all control statements.

CONTROL WORD – A system-supplied word that precedes each W-type record in storage.

COPY – The control statement used to copy the entire contents of a file.

COPYBF – The control statement used for copying binary files.

COPYBR – The control statement used for copying binary records.

COPYCF – The control statement used for copying coded files.

COPYCR – The control statement used for copying coded records.

COPYSBF – The control statement used for copying a file that does not have carriage control characters. A blank is added as the first character of each line so that the file can be printed without losing the first character of each line.

CYBER RECORD MANAGER – A software package running under the NOS/BE operating system that allows a variety of record types, blocking types, and file organizations to be created and accessed. The execution time input/output of COBOL 4, COBOL 5, FORTRAN Extended 4, Sort/Merge 4, ALGOL 4, and the DMS-170 products is implemented through CYBER Record Manager. Neither the input/output of the NOS/BE operating system nor any of the system utilities, such as COPY or SKIPF, is implemented through CYBER Record Manager. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

CYCLE – One of five separate and distinct files under a permanent file name and owner identification. Each cycle is identified by the permanent file name, owner, and cycle number.

DAYFILE – A chronological file created during job execution, which forms a permanent accounting and job history record. Dayfile messages are generated by operator action or by the system when control statements are processed or other significant action occurs. A copy of the dayfile is printed with the output for each job. The dayfile is assumed to be the last logical record of an output file.

DEBUG – To detect, locate, and remove mistakes from a program.

DEFAULT VALUE – A value that the operating system automatically assigns for an optional parameter in a control statement if that parameter is not included in the statement. Most default parameters are selected by the systems analyst when the operating system is installed.

DIAGNOSTIC – Pertaining to the detection and isolation of a malfunction or mistake. Also, a message printed when an assembler or compiler detects a program error.

DIRECT ACCESS FILE – In the context of CYBER Record Manager, a direct access file is one of the five file organizations. It is characterized by the system hashing of the unique key within each file record to distribute records randomly in blocks called home blocks of the file.

DIRECTIVES – Instructions that supplement processing defined by a control statement. Directives do not appear in the control statement section. They are usually in a separate section of INPUT or a file referenced in a control statement call. A directive is required for the EDITLIB utility, among others.

DISPLAY CODE – The internal representation of character data, where each character occupies six bits. Representations of the various characters are shown in appendix A.

DISK – A unit composed of one or more flat, circular plates with magnetic material on both sides that is used to store large amounts of data or programs.

DISK PACK – A group of disks with magnetically encoded information. Disk packs can be removed from the system with the stored information intact.

END-OF-INFORMATION (EOI) – Physical end of data. In card decks submitted through INTERCOM, a card with a 6/7/8/9 multiple-punch in column 1. CYBER Record Manager defines end-of-information in terms of the file organization and file residence.

File Organization	File Residence	Physical Position
Sequential	Mass storage	After last user record.
	Labeled tape in SI, I, X, S, L, format	After last user record and before any file trailer labels.
	Unlabeled tape in SI, I, X format	After last user record and before any file trailer labels.
	Unlabeled tape in S or L format	Undefined.
Word addressable	Mass storage	After last word allocated to file, which might be beyond the last user record.
Indexed Sequential, Actual Key	Mass storage	After record with highest key value.
Direct Access	Mass storage	After last record in most recently created overflow block or home block with the highest relative address.

END-OF-RECORD (EOR) – End of a logical record of data. In card decks submitted through INTERCOM, a card with a 7/8/9 multiple-punch in column 1.

ENTRY POINT – A unique name of a location within a section of code that can be referenced from other modules.

EXECUTE – The control statement used to execute a program that has been loaded into central memory.

EXECUTION – An input job is in execution after it is selected by the operating system and assigned to a control point. A job remains in execution until terminated, but it can be temporarily swapped or rolled out by the operating system.

EXTERNAL BINARY-CODED DECIMAL (BCD) FORMAT – A code in which information can be stored on magnetic tape. Each character is represented by two octal digits (6 bits), but this representation differs from display code.

FIELD LENGTH – The number of central memory words assigned to a job.

FILE – A set of information that has a logical file name. A file has a beginning called the beginning of information before which no data exists, and an end-of-information after which no data exists.

FILE INFORMATION TABLE (FIT) – A table through which a user program communicates with CYBER Record Manager.

FILE NAME – The name by which a file can be referenced; equivalent to logical file name. Not the same as permanent file name.

FILE POSITIONING – Altering the physical sequence in which a file is printed by backspacing or skipping forward a specified number of file sectors. A file can also be positioned forward to the beginning of the dayfile.

FILE SECTOR – Also known as physical record unit (PRU); the smallest amount of information transmitted by a single physical operation of a specified equipment. File sector is measured in central memory words; it contains sixty-four 60-bit computer words.

FORMS CODE – An arbitrary two-character identifier used to associate a print file with a printer form.

FORMAT CONTROL – The spacing information before or after printing a line, indicated by the leftmost character of the data that makes up a print line. If format control is suppressed, the file is printed with uniform single spaces between lines. Format control is synonymous with carriage control.

FTN – The control statement used to compile a program written in the FORTRAN Extended programming language.

HOLLERITH-CODED CARD – A punched card on which each of 80 columns can contain a set of one, two, or three punches representing one character.

HOME BLOCK – Mass storage allocated for a file with direct access organization at the time the file is created.

INDEXED SEQUENTIAL – A file organization in which records are stored in ascending order by key.

INPUT – The logical file name by which a job deck is known to the system once it is selected for execution.

INPUT FILE – A file entered into the system at the host computer or at a terminal, in the form of a card deck fed into the card reader. A valid input file is an executable job beginning with a job card and containing the required control statements for the desired operation.

INPUT QUEUE – A set of input files waiting to be assigned to control points by the operating system.

INTERACTIVE MODE – The state of the terminal during which commands and control statements entered from the keyboard are used to control job execution. Commands and control statements are executed in the sequence that they are entered. If the system is not too heavily loaded, execution appears instantaneous to the user.

JANUS – A group of system peripheral processor routines that control the processing of up to four card readers, three card punches, and 12 line printers. It normally functions at control point 1, but can be assigned to another control point.

JOB – Information beginning with a job statement and ending with an EOI statement. See Job Deck.

JOB DAYFILE – A special log the system maintains during execution of a job. At job termination, it is appended to the file OUTPUT for the job. All control statements executed by the job, significant information such as file assignment or file disposition, all operator interactions with a job, and errors are recorded in this file.

JOB DECK – The physical representation of a job, before execution, as a deck of cards or a group of card images. The first section of the deck begins with a job statement and contains control statements, which are used to control the job. Following sections contain the programs and data that the job requires for execution of the control statements. The job deck is terminated by a 6/7/8/9 card. Cards with 7/8/9 multipunched in column 1 separate sections within the deck.

JOB NAME – The name that the user must assign to the job on the job statement. It must begin with a letter and contain letters and digits; spaces are ignored. The system changes the name to the user's first five characters plus two system-supplied characters.

JOB STATEMENT – The first statement in a job that names the job and indicates system resources needed.

LABEL – The control statement giving the parameters for associating a tape file with a job and creating or checking a tape file label. Also, a block of information at the beginning or end of a volume or file, which serves to identify and delimit that volume or file.

LABELED TAPES – Tapes containing volume and file header and trailer labels in ANSI format.

LEVEL – For logical records, an octal number 0 through 17 in the system-supplied 48-bit marker that terminates a short or zero-length PRU.

LGO – The default name of the file to which language processors write executable code during program assembly or compilation.

LOAD - The control statement used to load executable code into central memory prior to execution.

LOADING – The placement of instructions and data into memory so that it is ready for execution. Loader input is obtained from local files and libraries. Upon completion of loading, execution of the program in the job's field length is optionally initiated.

Loading also involves performance of load-related services such as generation of a load map, presetting of unused core to a user-specified value, and generation of overlays or segments. A load that does not generate overlays or segments is referred to as a basic load.

LOAD POINT – A reflective marker on a magnetic tape after which data is recorded. A companion marker signifies the physical end-of-tape. Tape before the load point and after the end-of-tape mark is called the header and trailer, respectively.

LOCAL FILE – A file associated with a particular job on a temporary basis.

LOGICAL FILE NAME – The one to seven coded letters or digits by which the user's job recognizes a file. Every logical file name in a job must be unique and begin with a letter.

LOGICAL RECORD – An arbitrary logical grouping of data within a file, such as one or more cards followed by a 7/8/9 card in a card deck. See System-Logical-Record.

MAP – A printed listing of the names and locations of program entry points within a job; the names of all programs referencing those points; names and locations of common blocks; total length of all programs and common blocks loaded; length of the loader routine and tables used to load the job; and a list of any unsatisfied references. Also called a load map or core map.

NOISE RECORD – Number of characters the tape drivers discard as being extraneous noise rather than a valid record. Used to physically block out unerasable damaged areas on magnetic tape. Value depends on installation setting.

OBJECT DECK – A fully compiled or assembled program that is punched on cards and ready to be read into the computer.

OCTAL NUMBER SYSTEM – A number system organized on a base of 8.

OUTPUT – A mass storage file associated with each job. When the job is completed, the file OUTPUT is transferred to the output queue pending assignment to a printer.

OUTPUT QUEUE – The lineup of job output on mass storage that is awaiting access to a line printer.

OVERFLOW BLOCK – Mass storage the system adds to a file with direct access organization when records cannot be accommodated in the home block.

PARTIAL MAP – A listing identical to a full map, with one exception: entry point addresses for listed programs and subroutines are omitted.

PARTITION – CYBER Record Manager defines a partition as a division within a file with sequential organization. Generally, a partition contains several records or sections. Implementation of a partition boundary is affected by file structure and residence. Notice that in a file with W type records, a short PRU of level 0 terminates both a section and a partition.

Device	Record Type	Block Type	Physical Boundary
PRU device	W	I	A short PRU of level 0 containing a one-word deleted record pointing back to the last I block boundary, followed by a control word with a flag indicating a partition boundary.
	W	C	A short PRU of level 0 containing a control word with a flag indicating a partition boundary.
	D,F,R, T,U,Z	C	A short PRU of level 0 followed by a zero-length PRU of level 17.
S or L format tape	W	I	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a deleted one-word record pointing back to the last I block boundary, followed by a control word with a flag indicating a partition boundary.
	W	C	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a control word with a flag indicating a partition boundary.
	D,F,T, R,U,Z	C,K,E	Tapemark.
	S	–	Zero-length PRU of level number 0.
Any other tape format	–	–	Undefined.

PASSWORD – A parameter of one to ten characters in the log-in procedure that is assigned to the terminal user for access security purposes.

PERIPHERAL PROCESSOR UNIT (PPU) – An individual computer with its own memory, used for high-speed transfer of information (input and output) between peripheral devices, mass storage, and central memory.

PERMANENT FILE – A file known to the operating system as being permanent; that is, the file will survive between jobs. Permanent files are created by the CATALOG control statement. Permanent files can have certain restrictions for their access, such as access only with a special keyword identifier, or read-only access.

PERMANENT FILE NAME – The name, consisting of 1 to 20 letters or digits, that identifies a permanent file.

PHYSICAL RECORD UNIT (PRU) – Same as File Sector.

PRINT FILE – A central file that is in correct format to be printed at a terminal or host line printer.

PRINT QUEUE – A set of print files waiting to be printed.

PRIORITY – A value assigned to central files by the system or by a user, determining their precedence. When two or more print files are waiting to be transmitted to the same terminal, the one with the highest priority (largest priority number) is transmitted first.

PROGRAM – A sequence of coded instructions that solves a program.

PROGRAM LISTING – A printed copy of a source language program.

PROGRAM NAME – Also referred to as ident name; it is the name at the beginning of each program or routine module. A program name is one to seven characters; colons are illegal.

PRU – Under NOS/BE, the amount of information transmitted by a single physical operation of a specified device. The size of a PRU depends on the device: a PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

Device	Size in Number of 60-Bit Words
Mass storage	64
Tape in SI format with coded data	128
Tape in SI, X, or I format with binary data	512
Tape in other format	undefined

PRU DEVICE – Under NOS/BE, a mass storage device or a tape in SI, I, or X format, so called because records on these devices are written in PRUs.

PUNCH – The mass storage file that is automatically punched on Hollerith cards. When the job is completed, PUNCH is transferred to the punch queue to await assignment to a card punch.

PUNCHB – The mass storage file that is automatically punched on binary cards. When the job is completed, PUNCHB is transferred to the punch queue to await assignment to a card punch.

PUNCH FILE – A central file that is in correct format to be punched at a host computer card punch. 200 User Terminals are not equipped with card punches.

PUNCH QUEUE – A set of files waiting to be punched.

PURGE – To delete a permanent file from the system. This enables releasing its mass storage space, erasing its catalog entries, and so forth.

RANDOM FILE – In the context of CYBER Record Manager, a file with word addressable, indexed sequential, direct access, or actual key organization in which individual records can be accessed by the values of their keys.

In the context of the NOS/BE operating system, a file with the random bit set in the file environment table in which individual records are accessed by their relative PRU numbers.

RECORD – CYBER Record Manager defines a record as a group of related characters. A record or a portion thereof is the smallest collection of information passed between CYBER Record Manager and a user program. Eight different record types exist, as defined by the RT field of the file information table.

Other parts of the operating system and its products might have additional or different definition of records.

RECORD TYPE – The term record type can have one of several meanings, depending on the context. CYBER Record Manager defines eight record types established by the RT field in the file information table. Tables output by the loader are classified as record types such as text, relocatable, or absolute, depending on the first few words of the tables.

REFERENCE ADDRESS (RA) – The first word in the field length of a job. Because of dynamic relocation, the RA frequently changes with respect to the first word in central memory; but it always remains the same with respect to other addresses within the job's field length.

REGISTERS – Units within the central processor used to hold operands. The A registers contain the addresses of words within central memory; the X registers contain operands used in calculations and the results of calculations; the B registers are used for incrementing and indexing.

RELATIVE ADDRESS – The address of any location with respect to the job's reference address.

RELOCATE – In programming, to move a routine from one portion of internal storage to another, and to adjust the necessary address references so that the routine, in its new location, can be executed.

REPEAT COUNT – A decimal value specified by the user that indicates the number of additional times the file is to be output. Initially, the repeat count is zero.

REQUEST – A control statement specifying the need for a system resource, such as a magnetic tape unit; a named file is assigned to that device.

RETENTION PERIOD – The number of days after creation that a permanent file or labeled magnetic tape is to be retained. The retention period is determined when the file is created. The user can specify the retention period on CATALOG or LABEL control statements, or allow the system to assign the default retention period.

ROLL-OUT – The act of removing a job from central memory to mass storage before execution is complete so memory can be assigned to a higher priority job.

SECTION – CYBER Record Manager defines a section as a division within a file with sequential organization. Generally, a section contains more than one record and is a division within a partition of a file. A section terminates with a physical representation of a section boundary.

The NOS/BE operating system equates a section with a logical record of level 0 through 16_8.

Device	Record Type	Block Type	Physical Representation
PRU	W	I	Deleted one-word record pointing back to the last I block boundary followed by a control word with flags indicating a section boundary. At least the control word is in a short PRU of level 0.
	W	C	A control word with flags indicating a section boundary. The control word is in a short PRU of level 0.
	D,F,R, T,U,Z	C	A short PRU with a level less than 17_8.
	S	Any	Undefined.
S or L format tape	W	I	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a deleted one-word record pointing back to the last I block boundary, followed by a control word with flags indicating a section boundary.
	W	C	A separate tape block containing as many deleted records of record length 0 as required to exceed noise record size, followed by a control word with flags indicating a section boundary.
	D,F,R, T,U,Z	C,K,E	Undefined.
	S	Any	Undefined.
Any other tape format	–	–	Undefined.

SECTOR – One mass storage physical record unit of sixty-four 60-bit words (640 6-bit characters).

SEPARATOR – A delimiter used to separate parameters from each other on a control statement; it can be a comma or left parenthesis.

SEQUENTIAL FILE – A collection of records that are placed in physical rather than logical order. Given the location of one record, the location of the next can be determined by the physical position of the previous record. Tape files, punch card files, printer files, and so forth, are classified as sequential.

SHORT PRU – A PRU that does not contain as much user data as the PRU can hold and that is terminated by a system terminator with a level number.

Under NOS/BE, a short PRU defines the end of a system-logical-record. In the CYBER Record Manager context, a short PRU can have several interpretations, depending on the record and blocking types.

SI TAPE – A system internal tape. These are the standard tapes produced by the NOS/BE operating system. These tapes contain fixed-length sections and special system marks peculiar to CDC operating systems.

SKIPB – The control statement used to skip backwards on a file.

SOFTWARE – The collection of programs and routines associated with a computer system; for example, compilers, library routines.

STANDARD DUMP – A printed listing that contains the contents of the operating system exchange package (including the current contents of all central processor registers); the contents of the first 100_8 words in the job's field length; the contents of the last 100_8 locations immediately preceding the location where the job terminated; and the first 100_8 locations immediately following this point.

SWAP-OUT – The process of removing jobs from central memory to mass storage before execution is complete, so control point and memory can be assigned to another job. A job is swapped out when it is waiting for an external event, or when its control point and/or central memory is needed by a higher priority job.

SYMBOLIC REFERENCE TABLE – A printed listing showing the names and addresses of all variables used in or referenced by a user program, and the names and addresses of entry points within system routines referenced by the program.

SYSTEM LIBRARY – That file or group of files containing the core image system overlays and the system relocatable code available to all users on a read-only basis.

SYSTEM TABLES – Tables used by the operating system, and which lie outside of the user's field length.

TABLE – A collection of data, each item being uniquely identified either by some label or by its relative position.

TERMINATOR – On a control statement, a period or right parenthesis used as the delimiter following the request or final parameter; only informative messages or comments can follow.

UNIT RECORD DEVICE – A device such as a card reader, line printer, or card punch.

UNLABELED TAPE – An S or L tape that contains no label, or an SI tape that only contains the file trailer label automatically written by the operating system at the end of the last data on the tape.

UNLOAD – To remove a tape from ready status by rewinding beyond the load point; the tape is then no longer under control of the computer. Also, the control statement that requests this action.

UNSATISFIED EXTERNAL - An external reference for which the loader has not yet loaded a module containing the matching entry point.

W-TYPE RECORD - One of the eight record types supported by CYBER Record Manager. Such records appear in storage preceded by a system-supplied control word. The existence of the control word allows files with sequential organization to have both partition and section boundaries.

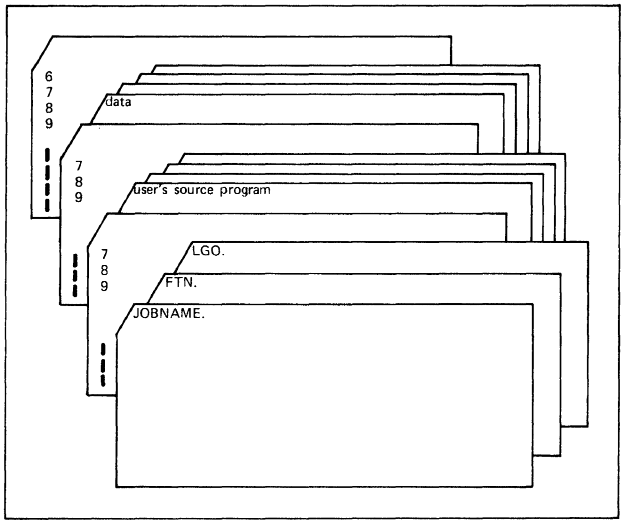
WORD - The unit of memory or storage containing 60 bits or ten 6-bit characters.

ZERO-BYTE TERMINATOR - Twelve bits of zero in the low-order position of a word that marks the end of line to be displayed at a terminal or printed on a line printer. The image of cards input through the card reader or terminal also has such a terminator.

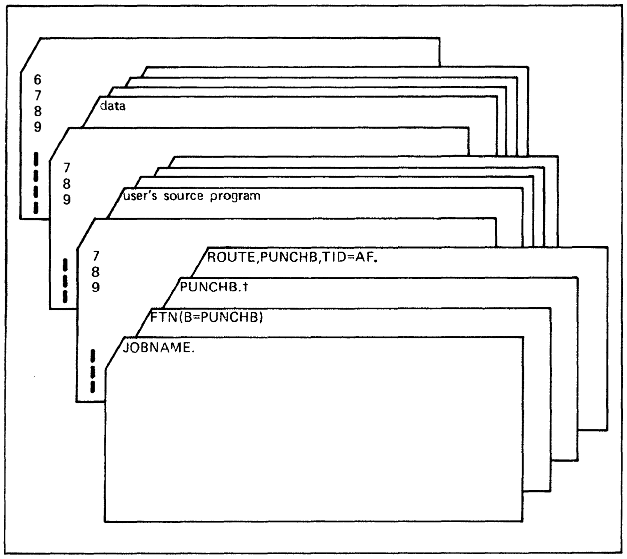
ZERO-LENGTH PRU - A PRU that contains system information, but no user data. Under CYBER Record Manager, a zero-length PRU of level 17 is a partition boundary.

SAMPLE JOB DECKS C

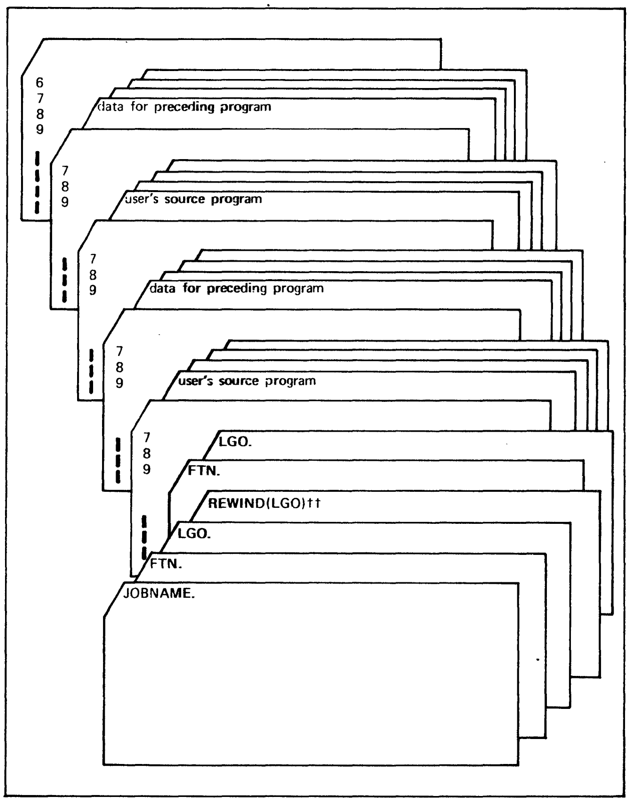
Example 1. Compiling and executing a FORTRAN Extended source program.



Example 2. Compiling, punching an object deck, and executing a FORTRAN Extended source program. The object deck is routed to terminal AF for punching.



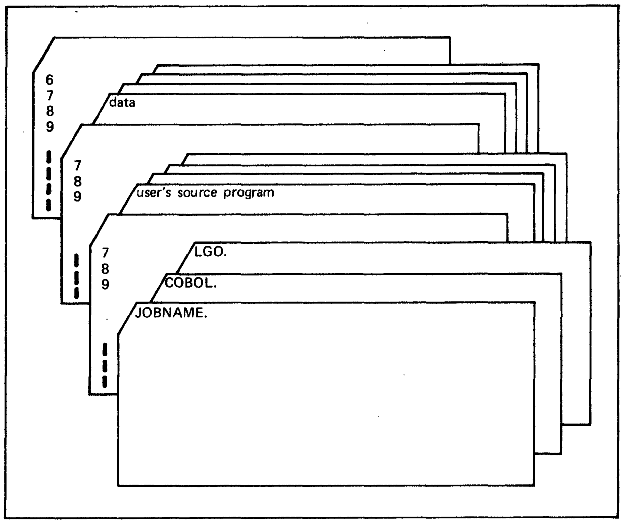
Example 3. Compiling and executing two FORTRAN Extended programs, each with its own set of data.



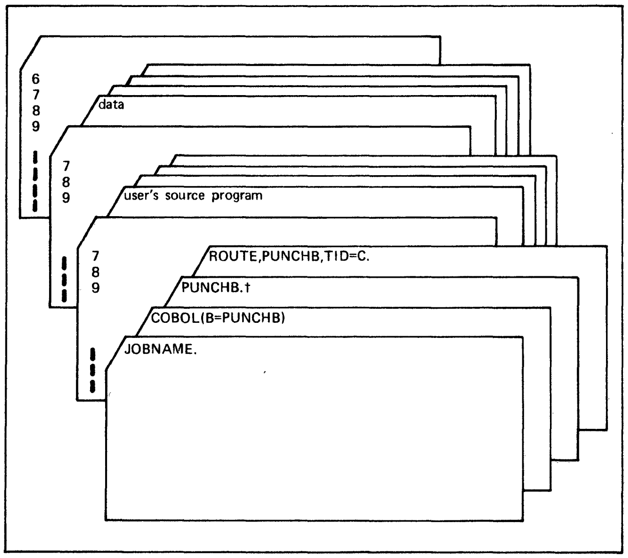
†PUNCHB. must be used because, in this case, FORTRAN Extended puts the object code only on PUNCHB.

††The REWIND(LGO) statement is needed for the following reason. The executable code from the first program is put on LGO. When LGO is loaded, the loader rewinds LGO, loads LGO, and leaves it positioned at the end of code for the first program. If the REWIND were not there, the second compilation would cause the code of the second program to be placed after the code for the first program. Then, when the second load operation occurred, the loader would rewind LGO, causing the first program instead of the second program to be loaded.

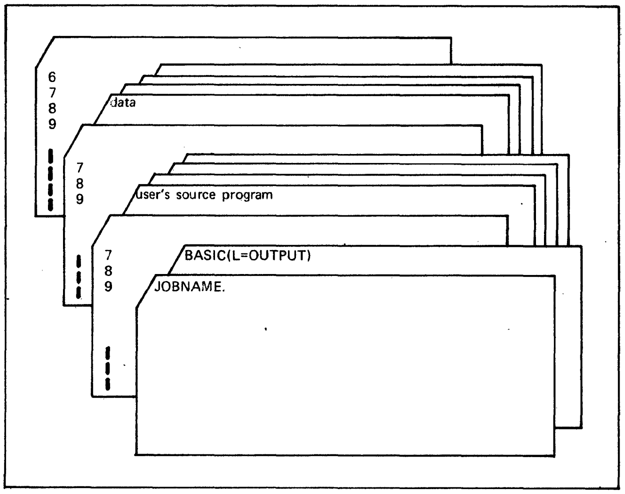
Example 4. Compiling and executing a COBOL source program.



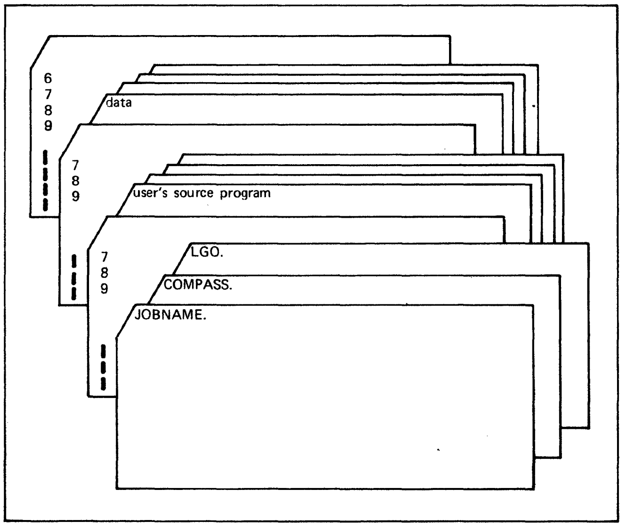
Example 5. Compiling, punching binary cards, and executing a COBOL source program. The binary deck is punched at the central site.



Example 6. Compiling and executing a BASIC source program.

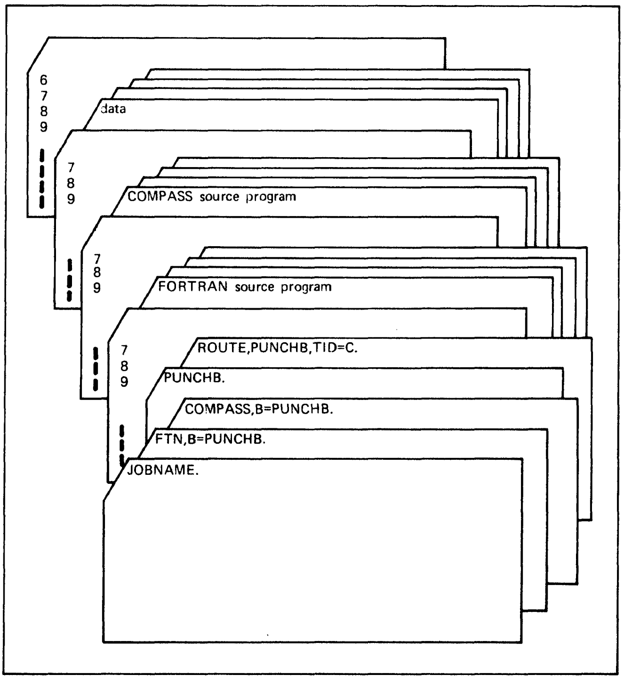


Example 7. Assembling and executing a COMPASS source program.



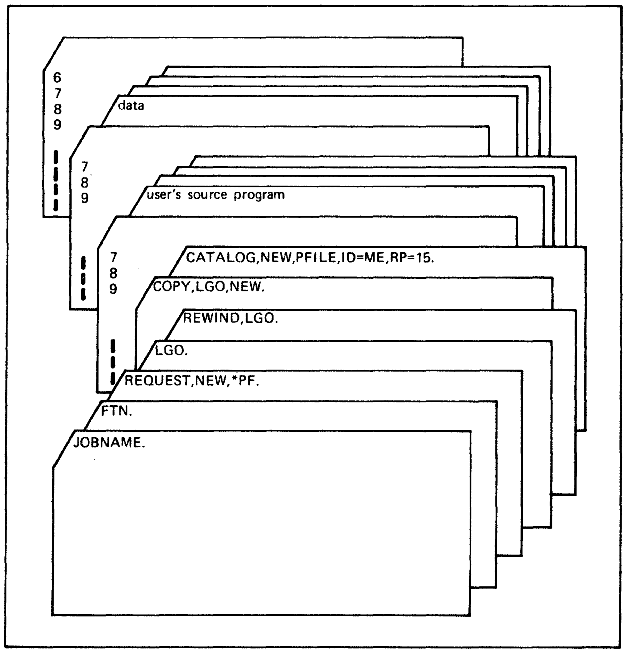
†PUNCHB. must be used because, in this case, COBOL puts the executable code only on PUNCHB.

Example 8. Compiling/assembling, punching binary cards, and executing a program containing both FORTRAN Extended and COMPASS source code. The binary deck is punched at the central site.



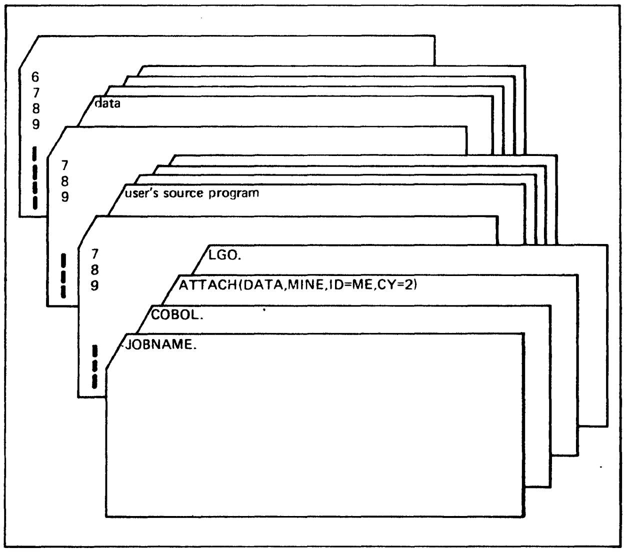
```
JOBNAME.
FTN,B=PUNCHB.
COMPASS,B=PUNCHB.
PUNCHB.
ROUTE,PUNCHB,TID=C.
```

Example 9. Compiling and executing a FORTRAN program and creating a permanent file containing the binary of the program.



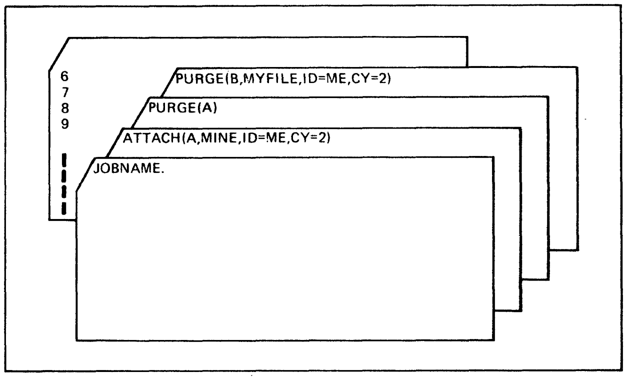
```
JOBNAME.
FTN.
REQUEST,NEW,*PF.
LGO.
REWIND,LGO.
COPY,LGO,NEW.
CATALOG,NEW,PFILE,ID=ME,RP=15.
```

Example 10. Compiling a COBOL program, attaching a permanent file, and executing the COBOL program. The program uses data from the data deck as well as data from the attached file.



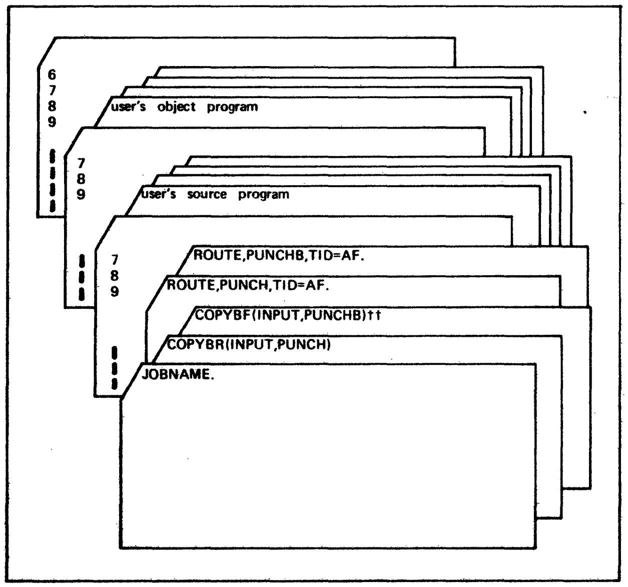
```
JOBNAME.
COBOL.
ATTACH(DATA,MINE,ID=ME,CY=2)
LGO.
```

Example 11. Purging two permanent files; file A is attached and then purged and file B is purged without being attached.



```
JOBNAME.
ATTACH(A,MINE,ID=ME,CY=2)
PURGE(A)
PURGE(B,MYFILE,ID=ME,CY=2)
```

Example 12. Making a copy of a source program and a binary program. Punch output is routed to the card punch at terminal AF.



[†] COPYBR copies the source program to the file PUNCH to have it punched in Hollerith format.

[††] COPYBF copies the binary program to the file PUNCHB to have it punched in binary format.

INDEX

COMMENT SHEET

MANUAL TITLE CDC INTERCOM Version 4 Remote Batch User's Guide

PUBLICATION NO. 60496100 REVISION B

FROM: NAME:

BUSINESS
ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Revision Record page at the front of the manual. Customer engineers are urged to use the TAR.

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

CUT ALONG LINE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

CUT ALONG LINE

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

FOLD

FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINNESOTA 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION