CHAPTER 19

EXTERNALLY DESCRIBED PRINTER FILES

CHAPTER OBJECTIVES

Upon completion of this chapter, you should be able to

- Explain how data description specifications or DDS are used to describe printer files.
- Demonstrate how externally described printer files are used in application programs.

EXTERNALLY DESCRIBED PRINTER FILES

A **printer file** defined within an application program is called a **program-described printer file**. This means that the file, record, and field descriptions are defined internally within the data division. With this method, report specifications are *hard-coded* into the program and become part of the program's compiled object.

Printer files that contain report specifications can be defined externally to any program that uses them. This means that the report specifications for a printer file are described separately from any programs and compiled into a printer file object (Type=PRTF). A file described in this manner is called an **externally described printer file**.

One important advantage of using externally described printer files is that software developers can change the report format of the source DDS for the printer file, recompile the source printer file into a new printer file object, and not have to modify the programs that use the printer file object. This saves time and produces fewer errors when a report must be modified.

Externally described printer files are created in the same manner in which physical files are created. Once the report specifications have been determined, **data description specifications,** or **DDS**, are used to define printer files at the source level. The software developer uses CODE/400 or another editor to enter the DDS into a source physical file. During the initial creation of the source DDS for the printer file, the type of object that will result from the compilation of the source member is specified. When defining the printer file to the system, the Type field must be specified as **PRTF** for printer file. In this way, the system will know what type of object to create when compiling the source DDS.

Creating the source DDS for a printer file does not create the actual printer file object. A printer file or PRTF object is created by compiling the source DDS. While compiling the source DDS, the compiler inspects the DDS source statements for correct syntax. If there are no errors in the DDS, a listing of the printer file will be printed and the printer file object will be created. If errors are found, the compiler will produce a listing of the DDS source along with a description of the errors. The errors must be corrected in the source DDS and then recompiled. This cycle must be repeated until all errors have been corrected and the compiler has created a printer file object from the source DDS. Once compiled, a program that uses an externally described printer file can send output to the printer file for printing.

Printer files are similar to physical files, except that the focus is on record formats consisting of lines of information that will be printed. Printer files are similar to logical files in that they can have multiple record formats. Each record format defines one or more output lines of the report.

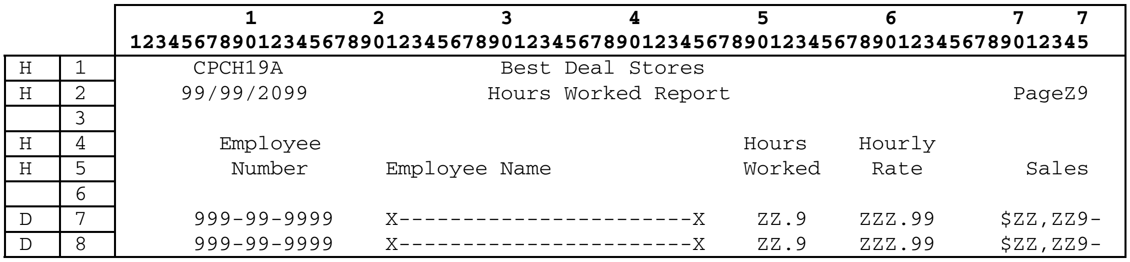
When a program is compiled that references an externally described printer file, the external description of the printer file and its attributes are retrieved from the actual file object and included in the compiled program object. Because the record formats are defined in the file object with DDS, they need not be defined in the `FILE SECTION` of the `DATA DIVISION`. Processing within the `PROCEDURE DIVISION` for an externally described printer file is the same as for a program-described printer file that has been described within the program.

These are the steps for creating an externally described printer file.

1. Determine the record formats for the printer file.
2. Write the DDS for the printer file.
3. Enter the DDS source member using CODE/400 or another editor.
4. Compile the DDS source member into a printer file object (Type=`PRTF`), correcting syntax errors when required.

CREATING A PRINTER FILE

To illustrate how to create a printer file, consider the printer spacing chart in Figure 19.1. This report is similar to many of the program-described reports produced in earlier chapters. It contains two rows of main title headings that describe the report, two rows of column headings that describe the detail information that will be printed, and detail lines that are single spaced. Thus, there are five different output lines printed on this report.



```
               1         2         3         4         5         6         7    7
      123456789012345678901234567890123456789012345678901234567890123456789012345
H  1       CPCH19A                 Best Deal Stores
H  2      99/99/2099              Hours Worked Report                      PageZ9
   3
H  4         Employee                                 Hours    Hourly
H  5          Number      Employee Name               Worked    Rate        Sales
   6
D  7       999-99-9999    X-----------------------X    ZZ.9    ZZZ.99     $ZZ,ZZ9-
D  8       999-99-9999    X-----------------------X    ZZ.9    ZZZ.99     $ZZ,ZZ9-
```

Figure 19.1 Printer spacing chart for employee hours worked report.

In a program-described program, there would be five different record formats defined in the `WORKING-STORAGE SECTION`. In a printer file, however, output lines that are printed at the same time can be defined within one record format. Let us examine the DDS for the printer file in Figure 19.2 that produces the report in Figure 19.1.

Notice that only two record formats are defined in this printer file. They are identified as `HEADING` and `DETAIL`. A record format in a printer file can define several output lines using the same record format name. These lines can then be printed as a group in one single output operation. In this example, the record format, `HEADING`, defines the four heading lines. The record format, `DETAIL`, defines the output for detail printing. Each record format defines

1. Which fields and literals are to print as part of the record format.

2. The length and data type of each field. The number of decimal positions for numeric fields.
3. The location on the output line where the fields and literals are to begin printing.
4. What line spacing is to be used when the output line is printed.
5. What editing, if any, is to be performed on numeric fields.

SPACING AND SKIPPING

Advancing the paper a fixed number of lines is called **spacing**. In an externally described printer file, the paper can be spaced before or after a line is printed. The **SPACEB** and **SPACEA** keywords are specified with a number within parentheses that denotes the number of lines to space before or after printing. The SPACEB keyword tells the printer to space before printing a line. Spacing after a line is performed with the SPACEA keyword. To advance the paper both before and after printing, use SPACEB and SPACEA.

In addition, the paper can be advanced to a specific line by using **skipping**. The **SKIPB** and **SKIPA** keywords enable the output to print on a specific line. For example, you usually skip to the beginning of a page to print the first heading. Another example of the SKIPB and SKIPA keywords is the printing of employee paychecks. Most of the data printed on employee paychecks is printed on the same line for every check. Since the line location where the data is to print is predetermined, the SKIPB keyword is used to jump to that specific line before printing. When specific data is printed on the same line of every page, using the SKIPB keyword instead of the SPACEB keyword helps identify more clearly where the data is being printed.

These keywords can be specified at the file level or the record level. If specified at the file level, the appropriate skipping or spacing will be performed when the record format is written from the program. If you wish to change the line position of the printer within a record format, the keyword is specified at a field level.

DATE KEYWORD

DATE is a variable keyword that internally stores the system date. When specified, the system date will be retrieved and printed in the location specified in the Positions column (positions 42–44). The format for the DATE keyword is

```
DATE(*SYS *YY)
```

The *SYS parameter specifies that the current system date is printed. The *YY specifies that four digits are used to represent the year.

PAGE KEYWORD

The PAGNBR (page number) keyword is used in printer files to print page numbers at the top of reports. **PAGNBR** is a variable keyword that internally stores the page number. When specified, the page number is printed in the location specified in the Positions column (positions 42–44). The PAGNBR keyword will print 1 on the first page, 2 on the second page, and so on. To print the literal 'PAGE' along with the number 'PAGE' is defined separately as a literal.

EDITING NUMERIC FIELDS

Edit codes and edit words may be used to make numeric fields easier to read. If editing is performed on output fields, the length of the field is adjusted accordingly to allow for the additional edit characters. The same edit codes and edit words explained in Chapter 14, Interactive Processing, are used with numeric fields defined in a printer file. When edit codes and edit words are used in a printer file, the keywords EDTCDE (edit code) and EDTWRD (edit word) are used, followed by the appropriate edit code or edit word enclosed within parentheses. For example, using the keyword EDTCDE(Y) with the DATE keyword will produce the print field 99/99/2099.

```
     A*  1    1    2    2    3    3    4    4    5    5    6    6
     6789012345678901234567890123456789012345678901234567890123456789
     AAN01N02N03T.Name++++++Rlen++TDpBLinPosFunctions++++++++++++++++
 100  * Printer File for CPCH19APRT
 200 A                                      REF(EMPPAYPF)
 300 A          R HEADING                   SKIPB(1)
 400  * Report title 1 prints on line 1
 500                                       6'CPCH19A'
 600 A                                    30'Best Deal Stores'
 700  * Report title 2 prints on line 2
 800 A                                     5DATE(*SYS *YY)
 900 A                                      EDTCDE(Y)
1000 A                                      SKIPB(2)
1100 A                                    29'Hours Worked Report'
1200 A                                    72 PAGNBR
1300 A                                       EDTCDE(3)
1400 A                                    70'Page'
1500  * Column heading 1 prints on line 4
1600 A                                     8'Employee'
1700 A                                      SKIPB(4)
1800 A                                    49'Hours'
1900 A                                    58'Hourly'
2000  * Column heading 2 prints on line 5
2100 A                                     9'Number'
2200 A                                      SKIPB(5)
2300 A                                    21'Employee Name'
2400 A                                    49'Worked'
2500 A                                    59'Rate'
2600 A                                    71'Sales'
2700 A                                      SKIPA(6)
2800  * Detail Line with single spacing
2900 A          R DETAIL                    SPACEB(1)
3000 A            EMPLOYEENOR              6ALIAS(DL_EMPLOYEE_NUMBER)
3100 A                                      EDTWRD('0   -  -    ')
3200 A            ENAME         25A       21ALIAS(DL_EMPLOYEE_NAME)
3300 A            HRSWORKED R             50ALIAS(DL_HOURS_WORKED)
3400 A                                      EDTCDE(3)
3500 A            HOURLYRATER             58ALIAS(DL_HOURLY_RATE)
3600 A                                      EDTCDE(3)
3700 A            SALES     R             69ALIAS(DL_SALES)
3800 A                                      EDTCDE(J $)
```

Figure 19.2 DDS for printer file.

Comment lines are indicated by an * in the Comment field (position seven). Comments help identify the printer file. Blank lines are used to separate different sections to make the printer file easier to read.

Line 2.00: The file-level keyword REF indicates that there are fields in this display file that are defined in another file but referenced in this display file. The REF keyword is used to identify the file that will be referenced to obtain the field attributes. In this example, the display file references the physical file EMPPAYPF.

Line 3.00: The letter R in the Name Type field (position 17) identifies this line as a record definition. This record-format line assigns the unique name HEADING (positions 19–28) to the record format. Note that a printer file can have more than one record format.

Since the headings are to print beginning on the first line of each page, the keyword `SKIPB(1)` is specified at the record level. The keyword `SKIPB(1)` tells the system to skip to line one before printing.

Line 4.00: A comment line indicating the beginning of the output specifications for the first heading line that will always print on line one.

Line 5.00–6.00: These lines define two constants that appear on the first heading line of the report. The literal `CPCH19A` is enclosed in apostrophes and is *right-justified* in the Functions field (positions 45–80). The Positions field (positions 42–44) is used to specify the beginning location of fields and literals in the output record. The literal `CPCH19A` will begin printing in position six. The second literal, `Best Deal Stores`, will print beginning in position 30.

Line 7.00: A comment line indicating the beginning of the output specifications for the second heading line that will always print on line two.

Lines 8.00–10.00: The keyword `DATE`, *left-justified* in the Functions field, prints the date in the heading of the report. The date will be located in the output record beginning in position five.

`DATE` causes a date consisting of month, day, and year to print. The `*SYS` parameter tells the system to use the current system date. The parameter `*YY` is specified so that the year is printed as a four-digit year. To edit the date so that it prints with slashes as month/day/year, the keyword `EDTCDE(Y)` is specified.

The report produced from this printer file contains four heading lines. A `SKIP` or `SPACE` keyword controls each heading. Since the second heading line is to print on line two of the report, the `SKIPB(2)` keyword is used. When a keyword follows a field or literal, the keyword applies to that field or literal. Thus, the `SKIPB(2)` keyword on line 10 is associated with the `DATE` field specified on line 8.00. This forces the printer to skip to line two before printing the date.

Line 11.00: The literal `Hours Worked Report` enclosed within apostrophes is defined in the Functions area. The 29 *right-justified* in the Positions field (positions 42–44) indicates that the literal will print beginning in position 29.

Lines 12.00–14.00: The literal **`Page`** enclosed within apostrophes will print in the output record beginning in position 70. Following the literal `Page` is the keyword `PAGNBR, which` prints the page number. The page number is printed beginning in position 72. The edit code, `EDTCDE(3)`, specifies that the `PAGNBR` keyword, a four-digit numeric value, is zero-suppressed and printed without commas.

Line 15.00: A comment line indicating the beginning of the output specifications for the first column heading line that will always print on line four.

Lines 16.00–19.00: These lines define the literals for the column heading line that will print on line four of the report. The `SKIPB(4)` keyword on line 17 is specified after the literal `Employee` on line 16 to force the printer to jump to line four before printing the literal `Employee`.

Line 20.00: A comment line indicating the beginning of the output specifications for the second column heading line.

Lines 21.00–27.00: These lines define the literals for the column heading line that will print on line five of the report. The SKIPB(5) keyword on line 22 is specified after the literal Number on line 21 to force the printer to jump to line five before printing the literal Number. The SKIPA(1) keyword on line 27 is specified after the literal Sales to force the printer to jump to line six after printing the column heading line. This allows a blank line between the column headings and the first detail line.

Line 28.00: A comment line indicating the beginning of the output specifications for the detail line.

Line 29.00: The letter R in the Name Type field (position 17) identifies this line as a record-format line with the unique name DETAIL. The keyword SPACEB(1) tells the system to space one line (single spacing) before printing. When a print operation is executed in the program for the record format DETAIL, the printer will space down the page one line and print a detail line.

ALIAS Keyword: Printer files allow a data field to have an alternative (ALIAS) name of up to 30 characters. When the ALIAS keyword is used, the ALIAS name may be used in the program instead of the field name in the Name field (positions 19–28).

Lines 30.00-31.00: The Employee Number field, EMPLOYEENO, is *left-justified* in the Name field (positions 19-28). The R in the Reference field is associated with the REF keyword on line 2.00 and tells the system to retrieve the attributes for this field in the EMPPAYPF file. Thus, the field is 9 bytes long with 0 decimal positions.

Printer files allow a data field to have an alternative (ALIAS) name of up to 30 characters. The alternative name is assigned to this field with the ALIAS keyword in position 45. When the ALIAS keyword is used, the ALIAS name DL_EMPLOYEE_NUMBER is copied into the program instead of the field name EMPLOYEENO in Name field (positions 19-28).

The edit word, EDTWRD(‘0   -   -   ‘), is enclosed in apostrophes or single quotes and is used to print dashes or hyphens in a Social Security number (Employee Number). Note that the sending field consists of nine digits, while the edit word occupies 11 positions in the EDTWRD keyword.

Line 32.00: The Employee Name, ENAME, field is described as a 25-byte alphanumeric (character) field. An alias name of DL_EMPLOYEE_NAME is assigned using the ALIAS keyword. The field will print beginning in position 21 of the output detail line.

Lines 33.00–34.00: The Hours Worked field is assigned the alias (ALIAS) name of DL_HOURS_WORKED. This field references the HRSWORKED field in the EMPPAYPF file that is defined as a three-digit numeric field, including one decimal. The field will print beginning in position 50 of the output detail line. An edit code, EDTCDE(3), is specified to print zero balances without commas or a sign.

Lines 35.00–36.00: The Hourly Rate field is assigned the alias (ALIAS) name of DL_HOURLY_RATE. This field references the HOURLYRATE field in the EMPPAYPF file and is defined as a five-digit numeric field, including two decimals. The field will print beginning in position 58 of the output detail line. An edit code, EDTCDE(3), is specified to print zero balances without commas or a sign.

Lines 37.00–38.00: The Sales field is assigned the alias (ALIAS) name of DL_SALES. This field references the SALES field in the EMPPAYPF data file and is defined as a five-digit numeric field with zero decimals. The field will print beginning in position 69 of the output detail line. An edit code, EDTCDE(J 3), is specified to get (1) commas to print where appropriate, (2) zeros, not blanks, to print when sales field is equal to zero, and (3) a negative sign (–) to print after the value if sales field contains a negative value.

USING AN EXTERNALLY DESCRIBED PRINTER FILE

SELECTING A PRINTER FILE

When a software developer wishes to use an externally described printer file in a program, the printer file must be defined in the FILE-CONTROL paragraph in the ENVIRONMENT DIVISION. An externally described printer file is defined this way:

```
*A   B   1   2   2   2   3   3   4   4   4   5   5   6   6   6   7
789012345678901234567890123456789012345678901234567890123456789012
 FILE-CONTROL.

     SELECT HOURS-WORKED-REPORT-FILE
         ASSIGN TO FORMATFILE-CPCH19APRT.
```

The SELECT clause is used to select or choose the file for processing. All files that are defined in the FILE-CONTROL paragraph, including externally described printer files, must be identified by an FD entry in the DATA DIVISION.

The ASSIGN clause associates the externally described printer file with a printer device. We have seen that a device type of PRINTER is used in the ASSIGN clause for program-described printer files. For externally described printer files, a device type of **FORMATFILE** is used in the ASSIGN clause.

DESCRIBING AN EXTERNALLY DESCRIBED PRINTER FILE

Once a printer file has been named or selected in the ENVIRONMENT DIVISION, it must be described with an FD entry in the DATA DIVISION. Here is an example of an FD entry in the DATA DIVISION that describes the externally described printer file:

```
*A   B   1   2   2   2   3   3   4   4   4   5   5   6   6   6   7
789012345678901234567890123456789012345678901234567890123456789012
 FD  HOURS-WORKED-REPORT-FILE.

 01  PRINT-RECORD-OUT.
 COPY DD-ALL-FORMATS-O OF CPCH19APRT.
```

The Format 2 COPY statement is used in the FD for the printer file. When the program is compiled, the Format 2 COPY statement creates the DATA DIVISION statements that describe the printer file. The compiler obtains the printer file description from the compiled printer file object.

There are various options that can be used with the Format 2 COPY statement. We discuss only one here. When the statement, COPY DD-ALL-FORMATS-O OF CPCH19APRT, is used, a storage area is established in memory and shared by all record formats for the printer file. Any data placed into an output record or format overlaps the data from the previous output record or format.

The DD option tells the system that the program may use the alias field names identified with the ALIAS keyword in the externally described printer file. Since ALL is used in the COPY statement, all formats from the printer file object are copied into the program. CPCH19APRT identifies which externally described printer file the COPY statement is using to obtain the output formats.

USING A PRINTER FILE

Before a program can write to a printer file, it must be opened. A printer file, which must be opened as output, is opened using a normal OPEN statement:

```
OPEN OUTPUT HOURS-WORKED-REPORT-FILE.
```

When the program is finished using a printer file, it must be closed. The CLOSE statement is used to close an externally described printer file in the same way other files are closed:

```
CLOSE HOURS-WORKED-REPORT-FILE.
```

The WRITE statement is used to send output to an externally described printer file. The Format 3 WRITE statement is used to write to an externally described printer file. A Format 3 WRITE statement has this format:

```
WRITE PRINT-RECORD-OUT
     FORMAT IS 'HEADING'.
```

The value specified in the FORMAT phrase is the name of the record format that is used in this output operation. Thus, the record format, HEADING, is written to the report file. HEADING is identified in the DDS of the printer file with the letter R in position 17 and HEADING in positions 19–28.

The ADVANCING phrase cannot be specified in WRITE statements for externally described printer files with an ASSIGN to device type FORMATFILE. Advancing of lines is controlled in a FORMATFILE file through DDS keywords, such as SKIPB, SKIPA, SPACEB, and SPACEA.

Figure 19.3 illustrates the complete program that uses the externally described printer file to produce the Hours Worked Report.

```
      *A   B   1   2   2   2   3   3   4   4   4   5   5   6   6   6 7
      78901234567890123456789012345678901234567890123456789012345678890
000100 PROCESS APOST.
000200
000300 IDENTIFICATION DIVISION.
000400
000500 PROGRAM-ID. CPCH19A.
000600
000700*****************************************************************
000800*                                                               *
000900*  This program reads the externally described Employee Pay    *
001000*  File (Keyed Sequence) and prints the Employee Hours Worked  *
001100*  Report.                                                      *
001200*                                                               *
001300*****************************************************************
001400
001500 ENVIRONMENT DIVISION.
001600
001700 INPUT-OUTPUT SECTION.
001800
001900 FILE-CONTROL.
002000     SELECT EMPLOYEE-PAY-FILE
002100         ASSIGN TO DATABASE-EMPPAYPF
002200         ORGANIZATION IS INDEXED
002300         ACCESS IS SEQUENTIAL
002400         RECORD KEY IS EXTERNALLY-DESCRIBED-KEY.
002500
```

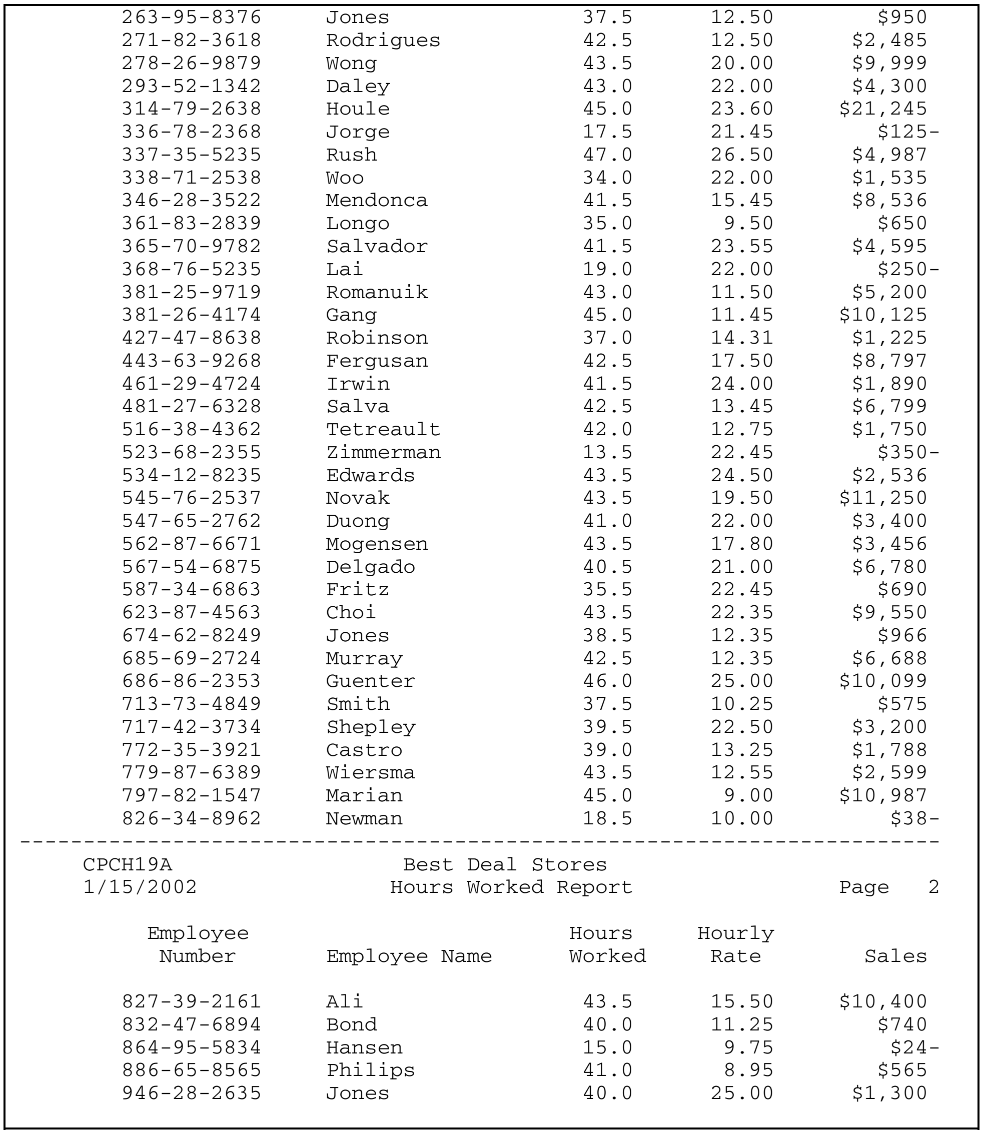
```
 002600     SELECT HOURS-WORKED-REPORT-FILE
 002700         ASSIGN TO FORMATFILE-CPCH19APRT.
 002800
 002900 DATA DIVISION.
 003000
 003100 FILE SECTION.
 003200
 003300 FD  EMPLOYEE-PAY-FILE.
 003400
 003500 01  EMPLOYEE-PAY-RECORD.
 003600 COPY DD-EMPPAYR OF COBOL2DLIB-EMPPAYPF.
+000001*  I-O FORMAT:EMPPAYR  FROM FILE EMPPAYPF  OF LIBRARY COBOL2DLIB
+000002*                          Employee Pay Record
+000003*THE KEY DEFINITIONS FOR RECORD FORMAT  EMPPAYR
+000004*  NUMBER               NAME                RETRIEVAL     ALTSEQ
+000005*   0001   EP-EMPLOYEE-NUMBER               ASCENDING       NO
+000006       05  EMPPAYR.
+000007           06 EP-EMPLOYEE-NUMBER    PIC S9(9).
+000008           06 EP-STORE-NUMBER       PIC S9(4).
+000009           06 EP-FIRST-NAME         PIC X(15).
+000010           06 EP-MIDDLE-INITIAL     PIC X(1).
+000011           06 EP-LAST-NAME          PIC X(15).
+000012           06 EP-DEPARTMENT         PIC S9(3).
+000013           06 EP-HOURLY-RATE        PIC S9(3)V9(2)    COMP-3.
+000014           06 EP-HOURS-WORKED       PIC S9(2)V9(1)    COMP-3.
+000015           06 EP-SALES              PIC S9(5)         COMP-3.
 003700
 003800 FD  HOURS-WORKED-REPORT-FILE.
 003900
 004000 01  PRINT-RECORD-OUT.
 004100 COPY DD-ALL-FORMATS-O OF CPCH19APRT.
+000001       05  CPCH19APRT-RECORD PIC X(47).
+000002* I-O FORMAT:HEADING  FROM FILE CPCH19APRT OF LIBRARY COBOL2ELIB
+000003*
+000004*      05  HEADING       REDEFINES CPCH19APRT-RECORD.
+000005* OUTPUT FORMAT:DETAIL FROM FILE CPCH19APRT OF LIBRARY COBOL2ELIB
+000006*
+000007       05  DETAIL-O      REDEFINES CPCH19APRT-RECORD.
+000008           06 DL-EMPLOYEE-NUMBER    PIC S9(9).
+000009           06 DL-EMPLOYEE-NAME      PIC X(25).
+000010           06 DL-HOURS-WORKED       PIC S9(2)V9(1).
+000011           06 DL-HOURLY-RATE        PIC S9(3)V9(2).
+000012           06 DL-SALES              PIC S9(5).
 004200
 004300 WORKING-STORAGE SECTION.
 004400
 004500 01  WS-CONTROL-FIELDS.
 004600     05  ARE-THERE-MORE-RECORDS      PIC X(3)    VALUE 'YES'.
 004700         88  NO-MORE-RECORDS                     VALUE 'NO '.
 004800     05  WS-LINE-COUNTER             PIC 9(3)    VALUE 61.
 004900     05  WS-LINE-LIMIT               PIC 9(3)    PACKED-DECIMAL
 005000                                                 VALUE 60.
 005100
 005200 PROCEDURE DIVISION.
 005300
 005400 000-MAIN-MODULE.
 005500
 005600     PERFORM 100-INITIALIZATION-RTN.
 005700     PERFORM 200-PROCESS-RECORD-RTN
 005800         UNTIL NO-MORE-RECORDS.
 005900     PERFORM 300-TERMINATION-RTN.
 006000     STOP RUN.
 006100
 006200 100-INITIALIZATION-RTN.
 006300
 006400     OPEN INPUT  EMPLOYEE-PAY-FILE
 006500          OUTPUT HOURS-WORKED-REPORT-FILE.
 006600     READ EMPLOYEE-PAY-FILE
 006700         AT END
 006800             SET NO-MORE-RECORDS TO TRUE
 006900     END-READ.
 007000
 007100 200-PROCESS-RECORD-RTN.
 007200     MOVE EP-EMPLOYEE-NUMBER TO DL-EMPLOYEE-NUMBER.
 007300     MOVE EP-HOURS-WORKED    TO DL-HOURS-WORKED.
 007400     MOVE EP-HOURLY-RATE     TO DL-HOURLY-RATE.
 007500     MOVE EP-SALES           TO DL-SALES.
 007600
```

```
007700     EVALUATE EP-MIDDLE-INITIAL
007800         WHEN SPACES        PERFORM 230-CONCATENATE-NAME-1-RTN
007900         WHEN NOT SPACES    PERFORM 240-CONCATENATE-NAME-2-RTN
008000     END-EVALUATE.
008100
008200     IF WS-LINE-COUNTER IS >= WS-LINE-LIMIT
008300         PERFORM 220-HEADING-RTN
008400     END-IF
008500     WRITE PRINT-RECORD-OUT
008600         FORMAT IS 'DETAIL'.
008700     ADD 2 TO WS-LINE-COUNTER.
008800     READ EMPLOYEE-PAY-FILE
008900         AT END
009000             SET NO-MORE-RECORDS TO TRUE
009100     END-READ.
009200
009300 220-HEADING-RTN.
009400     WRITE PRINT-RECORD-OUT
009500         FORMAT IS 'HEADING'.
009600     MOVE 5 TO WS-LINE-COUNTER.
009700
009800 230-CONCATENATE-NAME-1-RTN.
009900     INITIALIZE DL-EMPLOYEE-NAME.
010000     STRING
010100         EP-FIRST-NAME DELIMITED BY '  '
010200         ' ' DELIMITED BY SIZE
010300         EP-LAST-NAME DELIMITED BY ' '
010400         INTO DL-EMPLOYEE-NAME
010500     END-STRING.
010600
010700 240-CONCATENATE-NAME-2-RTN.
010800     INITIALIZE DL-EMPLOYEE-NAME.
010900     STRING
011000         EP-FIRST-NAME DELIMITED BY '  '
011100         ' ' DELIMITED BY SIZE
011200         EP-MIDDLE-INITIAL DELIMITED BY SIZE
011300         '. ' DELIMITED BY SIZE
011400         EP-LAST-NAME DELIMITED BY ' '
011500         INTO DL-EMPLOYEE-NAME
011600     END-STRING.
011700
011800 300-TERMINATION-RTN.
011900     CLOSE EMPLOYEE-PAY-FILE
012000           HOURS-WORKED-REPORT-FILE.
```

Figure 19.3 ***Program CPCH19A that uses externally described printer file.***

Figure 19.4 illustrates the Hours Worked Report produced from program CPCH19A.

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
     CPCH19A                   Best Deal Stores
     1/15/2002                Hours Worked Report               Page   1

          Employee                        Hours     Hourly
           Number       Employee Name     Worked     Rate         Sales

        103-42-9376     Lee                35.0       9.95         $563
        123-72-8964     Murphy             45.0      23.21      $10,125
        129-47-2987     Chang              44.0      16.65       $6,786
        132-13-5478     Porter             10.0       8.55           $0
        137-29-8794     Fernandes          43.5      24.50         $899
        147-59-2716     Collins            41.5      24.50      $14,767
        162-45-2825     Nabbouh            44.0      17.45       $5,325
        167-53-4862     Alexander          42.0      15.50       $2,400
        178-64-8964     Najjar             13.5      26.00          $85-
        218-37-6482     Drabicki           38.0      15.75       $1,400
        219-26-2389     Moscone            43.5      12.00       $2,599
        223-64-9622     Simpson            20.0      17.90          $99-
        228-72-5876     Fields             40.0       9.85       $4,555
        235-23-5658     Ryckman            42.5      14.45       $7,866
        237-28-9568     Xavier             35.0      25.00       $1,999
        246-85-3789     Doyle              13.5      12.45         $235-
        248-26-4826     Mohammed           32.0      22.35         $167
        263-57-2354     Mariconda          42.5      24.50       $6,700
```



```
        263-95-8376     Jones               37.5      12.50        $950
        271-82-3618     Rodrigues           42.5      12.50      $2,485
        278-26-9879     Wong                43.5      20.00      $9,999
        293-52-1342     Daley               43.0      22.00      $4,300
        314-79-2638     Houle               45.0      23.60     $21,245
        336-78-2368     Jorge               17.5      21.45        $125-
        337-35-5235     Rush                47.0      26.50      $4,987
        338-71-2538     Woo                 34.0      22.00      $1,535
        346-28-3522     Mendonca            41.5      15.45      $8,536
        361-83-2839     Longo               35.0       9.50        $650
        365-70-9782     Salvador            41.5      23.55      $4,595
        368-76-5235     Lai                 19.0      22.00        $250-
        381-25-9719     Romanuik            43.0      11.50      $5,200
        381-26-4174     Gang                45.0      11.45     $10,125
        427-47-8638     Robinson            37.0      14.31      $1,225
        443-63-9268     Fergusan            42.5      17.50      $8,797
        461-29-4724     Irwin               41.5      24.00      $1,890
        481-27-6328     Salva               42.5      13.45      $6,799
        516-38-4362     Tetreault           42.0      12.75      $1,750
        523-68-2355     Zimmerman           13.5      22.45        $350-
        534-12-8235     Edwards             43.5      24.50      $2,536
        545-76-2537     Novak               43.5      19.50     $11,250
        547-65-2762     Duong               41.0      22.00      $3,400
        562-87-6671     Mogensen            43.5      17.80      $3,456
        567-54-6875     Delgado             40.5      21.00      $6,780
        587-34-6863     Fritz               35.5      22.45        $690
        623-87-4563     Choi                43.5      22.35      $9,550
        674-62-8249     Jones               38.5      12.35        $966
        685-69-2724     Murray              42.5      12.35      $6,688
        686-86-2353     Guenter             46.0      25.00     $10,099
        713-73-4849     Smith               37.5      10.25        $575
        717-42-3734     Shepley             39.5      22.50      $3,200
        772-35-3921     Castro              39.0      13.25      $1,788
        779-87-6389     Wiersma             43.5      12.55      $2,599
        797-82-1547     Marian              45.0       9.00     $10,987
        826-34-8962     Newman              18.5      10.00         $38-
------------------------------------------------------------------------
     CPCH19A                  Best Deal Stores
     1/15/2002               Hours Worked Report                Page   2

          Employee                         Hours     Hourly
           Number     Employee Name       Worked     Rate        Sales

        827-39-2161     Ali                 43.5      15.50     $10,400
        832-47-6894     Bond                40.0      11.25        $740
        864-95-5834     Hansen              15.0       9.75         $24-
        886-65-8565     Philips             41.0       8.95        $565
        946-28-2635     Jones               40.0      25.00      $1,300
```

Figure 19.4 Report produced by program CPCH19A.

CASE PROBLEM: USING PRINTER FILES WITH CONTROL-BREAK PROGRAMS

In this case problem, the payroll department manager of the Best Deal Stores Company wants to print a Sales Report, as shown in Figure 19.5. This report summarizes the total sales by department within store and also prints a final company total for all stores.

```
      |         1         2         3         4         5         6         7
      |1234567890123456789012345678901234567890123456789012345678901234567890
H | 1 |       CPCH19B         Best Deal Stores
H | 2 |       Z9/99/2099        Sales Report          PageZZZ9
H | 3 |                          Store 9999
  | 4 |
H | 5 |                  Dept
H | 6 |                 Number            Sales
T | 7 |                   ZZ9       $$,$$$,$$$.00-
T | 8 |                   ZZ9       $$,$$$,$$$.00-
  |   |
T |   |  Total for store 9999 is $,$$$,$$$,$$9.99 *
  |   |
T |   |         Company total is $,$$$,$$$,$$9.99 **
```

Figure 19.5 ***Printer spacing chart for control-break report.***

The data necessary for this problem is stored in the Employee Pay file (EMPPAYPF), shown in Figure 19.6.

```
6...1....+....2....+....3....+....4....+....5....+....6....+....7..
A*n01n02n03R.Name++++++.Len++TDPULinPosKeywords++++++++++++++++++++
 ****************************************************************
 ***             Employee Pay File                            ***
 ***          Physical File ** EMPPAYPF                       ***
 ***             Key:  Employee Number                        ***
 ****************************************************************
A                                      UNIQUE
A          R EMPPAYR                   TEXT('Employee Pay Record')
A            EMPLOYEENO     9S 0       ALIAS(EP_EMPLOYEE_NUMBER)
A            STORENO        4S 0       ALIAS(EP_STORE_NUMBER)
A            FIRSTNAME     15A         ALIAS(EP_FIRST_NAME)
A            MIDDLEINIT     1A         ALIAS(EP_MIDDLE_INITIAL)
A            LASTNAME      15A         ALIAS(EP_LAST_NAME)
A            DEPARTMENT     3S 0       ALIAS(EP_DEPARTMENT)
A            HOURLYRATE     5P 2       ALIAS(EP_HOURLY_RATE)
A            HRSWORKED      3P 1       ALIAS(EP_HOURS_WORKED)
A            SALES          5P 0       ALIAS(EP_SALES)
A          K EMPLOYEENO
```

Figure 19.6 Data description specifications for employee pay file EMPPAYPF.

To obtain the records in the correct sequence, a logical file must be built over the physical file. The DDS for the logical file CPCH19BL are shown in Figure 19.7.

```
6...1....+....2....+....3....+....4....+....5....+....6....+....7..
A*n01n02n03R.Name++++++.Len++TDPULinPosKeywords++++++++++++++++++++
A****************************************************************
A***             Employee Pay File                            ***
A***          Logical File ** CPCH19BL                        ***
A***             Key:  Store Number                           ***
A***                   Department Number                      ***
A****************************************************************
A          R EMPPAYR                   PFILE(EMPPAYPF)
A          K STORENO
A          K DEPARTMENT
```

Figure 19.7 ***Logical file CPCH19BL.***

The printer file, shown in Figure 19.8, produces the Sales Report shown in Figure 19.10. The program that reads the logical file and uses the externally described printer file to produce the Sales Report is shown in Figure 19.9.

```
     A*  1    1    2    2    3    3    4    4    5    5    6    6    7
     67890123456789012345678901234567890123456789012345678901234567890123456789
     AAN01N02N03T.Name++++++Rlen++TDpBLinPosFunctions+++++++++++++++++++++
 100  * Printer File for CPCH19BPRT
 200
 300 A          R HEADING                   SKIPB(1)
 400  * Report heading prints on line 1
 500 A                                     8'CPCH19B'
 600 A                                    24'Best Deal Stores'
 700  * Report headings prints on line 2
 800 A                                     8DATE(*SYS *YY)
 900 A                                      EDTCDE(Y)
1000 A                                      SKIPB(2)
1100 A                                    26'Sales Report'
1200 A                                    48'Page'
1300 A                                    52 PAGNBR
1400 A                                       EDTCDE(3)
1500  * Report heading prints on line 3
1600 A                                    27'Store'
1700 A                                      SKIPB(3)
1800 A            HLSTORE        4S 0     +1ALIAS(HL_STORE_NUMBER)
1900  * Column heading prints on line 5
2000 A                                    19'Dept'
2100 A                                      SKIPB(5)
2200  * Column heading prints on line 6
2300 A                                    18'Number'
2400 A                                      SKIPB(6)
2500 A                                    36'Sales'
2600
2700  * Department Total Line with double spacing
2800
2900 A          R DEPTTOTAL                 SPACEB(2)
3000 A            TLDEPT         3S 0     20ALIAS(DTL_DEPARTMENT_NUMBER)
3100 A            DEPTSALES      9S 2     30ALIAS(DTL_DEPARTMENT_TOTAL)
3200 A                                      EDTCDE(J $)
3300
3400  * Store Total Line with double spacing
3500
3600 A          R STORETOTAL                SPACEB(2)
3700 A                                     3'Total for store'
3800 A            TLSTORE        4S 0     +1ALIAS(STL_STORE_NUMBER)
3900 A                                    +1'is'
4000 A            TLSALES       11S 2     28ALIAS(STL_STORE_TOTAL)
4100 A                                      EDTCDE(J $)
4200 A                                    44'*'
```

Figure 19.8 DDS for printer file CPCH19BPRT.

```
      *A   B   1   2   2   2   3   3   4   4   4   5   5   6   6   6   7
      789012345678901234567890123456789012345678901234567890123456789012
00100 PROCESS APOST.
00200
00300 IDENTIFICATION DIVISION.
00400
00500 PROGRAM-ID. CPCH19B.
00600
00700 AUTHOR.     JILL PROGRAMMER.
00800
00900*****************************************************************
01000*                                                               *
01100*  This program uses an externally described printer file to    *
01200*  produce a Sales Report.                                      *
01300*  Control break procedures are used to print department        *
01400*  and store totals.                                            *
01500*                                                               *
01600*****************************************************************
01700
01800 ENVIRONMENT DIVISION.
01900
02000 INPUT-OUTPUT SECTION.
02100
02200 FILE-CONTROL.
02300     SELECT EMPLOYEE-PAY-FILE
02400         ASSIGN TO DATABASE-CPCH19BL
02500         ORGANIZATION IS INDEXED
02600         ACCESS IS SEQUENTIAL
02700         RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
```

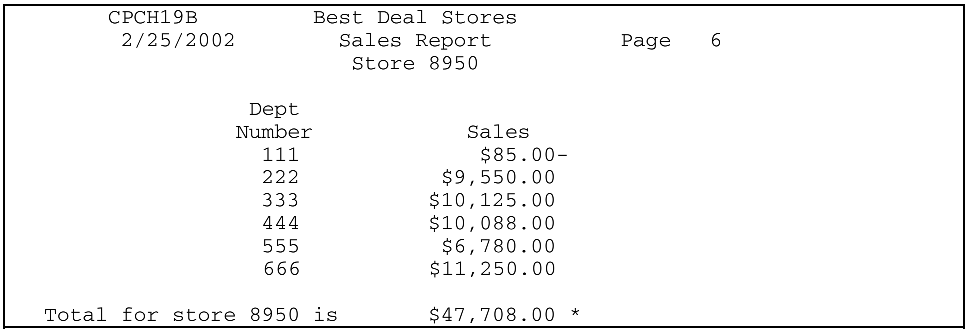
```
02800             WITH DUPLICATES.
02900
03000     SELECT SALES-REPORT-FILE
03100         ASSIGN TO FORMATFILE-CPCH19BPRT.
03200
03300 DATA DIVISION.
03400
03500 FILE SECTION.
03600
03700 FD  EMPLOYEE-PAY-FILE.
03800
03900 01  EMPLOYEE-PAY-RECORD.
04000 COPY DD-EMPPAYR OF CPCH19BL.
00001*    I-O FORMAT:EMPPAYR    FROM FILE CPCH19BL   OF LIBRARY COBOL2DLIB
00002*                          Employee Pay Record
00003*THE KEY DEFINITIONS FOR RECORD FORMAT  EMPPAYR
00004*  NUMBER               NAME                RETRIEVAL       ALTSEQ
00005*   0001   EP-STORE-NUMBER                    ASCENDING         NO
00006*   0002   EP-DEPARTMENT                      ASCENDING         NO
00007       05  EMPPAYR.
00008           06 EP-EMPLOYEE-NUMBER    PIC S9(9).
00009           06 EP-STORE-NUMBER       PIC S9(4).
00010           06 EP-FIRST-NAME         PIC X(15).
00011           06 EP-MIDDLE-INITIAL     PIC X(1).
00012           06 EP-LAST-NAME          PIC X(15).
00013           06 EP-DEPARTMENT         PIC S9(3).
00014           06 EP-HOURLY-RATE        PIC S9(3)V9(2)    COMP-3.
00015           06 EP-HOURS-WORKED       PIC S9(2)V9(1)    COMP-3.
00016           06 EP-SALES              PIC S9(5)         COMP-3.
04100
04200 FD  SALES-REPORT-FILE.
04300
04400 01  PRINT-RECORD-OUT.
04500 COPY DD-ALL-FORMATS-O OF CPCH19BPRT.
00001       05  CPCH19BPRT-RECORD PIC X(15).
00002* OUTPUT FORMAT:HEADING    FROM FILE CPCH19BPRT OF LIBRARY COBOL2ELIB
00003*
00004       05  HEADING-O     REDEFINES CPCH19BPRT-RECORD.
00005           06 HL-STORE-NUMBER       PIC S9(4).
00006* OUTPUT FORMAT:DEPTTOTAL  FROM FILE CPCH19BPRT OF LIBRARY COBOL2ELIB
00007*
00008       05  DEPTTOTAL-O   REDEFINES CPCH19BPRT-RECORD.
00009           06 DTL-DEPARTMENT-NUMBER PIC S9(3).
00010           06 DTL-DEPARTMENT-TOTAL  PIC S9(7)V9(2).
00011* OUTPUT FORMAT:STORETOTAL FROM FILE CPCH19BPRT OF LIBRARY COBOL2ELIB
00012*
00013       05  STORETOTAL-O  REDEFINES CPCH19BPRT-RECORD.
00014           06 STL-STORE-NUMBER      PIC S9(4).
00015           06 STL-STORE-TOTAL       PIC S9(9)V9(2).
04600
04700 WORKING-STORAGE SECTION.
04800
04900 01  WS-CONTROL-FIELDS.
05000     05  ARE-THERE-MORE-RECORDS  PIC X(3)    VALUE 'YES'.
05100         88  MORE-RECORDS                    VALUE 'YES'.
05200         88  NO-MORE-RECORDS                 VALUE 'NO '.
05300     05  WS-PREVIOUS-DEPARTMENT      LIKE EP-DEPARTMENT
     * LIKE: WS-PREVIOUS-DEPARTMENT has inherited PICTURE S9(3)
05400                                             VALUE ZEROS.
05500     05  WS-PREVIOUS-STORE-NUMBER    LIKE EP-STORE-NUMBER
     * LIKE: WS-PREVIOUS-STORE-NUMBER has inherited PICTURE S9(4)
05600                                             VALUE ZEROS.
05700 01  WS-TEMPORARY-WORK-FIELDS.
05800     05  WS-DEPARTMENT-TOTAL         LIKE EP-SALES (+2)
     * LIKE: WS-DEPARTMENT-TOTAL has inherited PICTURE S9(7) USAGE
     *          PACKED-DECIMAL
05900                                             VALUE ZEROS.
06000     05  WS-STORE-TOTAL              LIKE EP-SALES (+2)
     * LIKE: WS-STORE-TOTAL has inherited PICTURE S9(7) USAGE
     *          PACKED-DECIMAL
06100                                             VALUE ZEROS.
06200
06300 PROCEDURE DIVISION.
06400
06500 000-MAIN-MODULE.
06600     PERFORM 100-INITIALIZATION-RTN.
06700     PERFORM 200-PROCESS-RECORD-RTN
06800         UNTIL NO-MORE-RECORDS.
```

```
06900     PERFORM 300-TERMINATION-RTN.
07000     STOP RUN.
07100
07200 100-INITIALIZATION-RTN.
07300     OPEN INPUT  EMPLOYEE-PAY-FILE
07400          OUTPUT SALES-REPORT-FILE.
07500     READ EMPLOYEE-PAY-FILE
07600          AT END
07700              SET NO-MORE-RECORDS TO TRUE
07800     END-READ
07900     MOVE EP-DEPARTMENT TO WS-PREVIOUS-DEPARTMENT.
08000     MOVE EP-STORE-NUMBER TO WS-PREVIOUS-STORE-NUMBER.
08100     PERFORM 225-HEADING-RTN.
08200
08300 200-PROCESS-RECORD-RTN.
08400     IF  EP-STORE-NUMBER NOT = WS-PREVIOUS-STORE-NUMBER
08500         PERFORM 210-DEPT-CONTROL-BREAK-RTN
08600         PERFORM 220-STORE-CONTROL-BREAK-RTN
08700     ELSE
08800         IF EP-DEPARTMENT NOT = WS-PREVIOUS-DEPARTMENT
08900             PERFORM 210-DEPT-CONTROL-BREAK-RTN
09000         END-IF
09100     END-IF
09200     ADD EP-SALES TO WS-DEPARTMENT-TOTAL.
09300     READ EMPLOYEE-PAY-FILE
09400          AT END
09500              SET NO-MORE-RECORDS TO TRUE
09600     END-READ.
09700
09800 210-DEPT-CONTROL-BREAK-RTN.
09900     MOVE WS-PREVIOUS-DEPARTMENT TO DTL-DEPARTMENT-NUMBER.
10000     MOVE WS-DEPARTMENT-TOTAL TO DTL-DEPARTMENT-TOTAL.
10100     WRITE PRINT-RECORD-OUT
10200         FORMAT IS 'DEPTTOTAL'.
10300     ADD WS-DEPARTMENT-TOTAL TO WS-STORE-TOTAL.
10400     IF MORE-RECORDS
10500         INITIALIZE WS-DEPARTMENT-TOTAL
10600         MOVE EP-DEPARTMENT TO WS-PREVIOUS-DEPARTMENT
10700     END-IF.
10800
10900 220-STORE-CONTROL-BREAK-RTN.
11000     MOVE WS-PREVIOUS-STORE-NUMBER TO STL-STORE-NUMBER.
11100     MOVE WS-STORE-TOTAL TO STL-STORE-TOTAL.
11200     WRITE PRINT-RECORD-OUT
11300         FORMAT IS 'STORETOTAL'.
11400     IF MORE-RECORDS
11500         INITIALIZE WS-STORE-TOTAL
11600         MOVE EP-STORE-NUMBER TO WS-PREVIOUS-STORE-NUMBER
11700         PERFORM 225-HEADING-RTN
11800     END-IF.
11900
12000 225-HEADING-RTN.
12100     MOVE EP-STORE-NUMBER TO HL-STORE-NUMBER.
12200     WRITE PRINT-RECORD-OUT
12300        FORMAT IS 'HEADING'.
12400
12500 300-TERMINATION-RTN.
12600     PERFORM 210-DEPT-CONTROL-BREAK-RTN.
12700     PERFORM 220-STORE-CONTROL-BREAK-RTN.
12800     CLOSE EMPLOYEE-PAY-FILE
12900           SALES-REPORT-FILE.
```

Figure 19.9 Program that produces a control-break report using a printer file

Figure 19.10 illustrates the Sales Report produced from program CPCH19B.

```
       CPCH19B         Best Deal Stores
        2/25/2002        Sales Report          Page   1
                          Store 1133

                  Dept
                 Number            Sales
                   111          $10,775.00
                   222           $1,750.00
                   333           $1,890.00
                   444             $731.00
                   555          $11,000.00
                   666          $24,445.00

  Total for store 1133 is       $50,591.00 *
-------------------------------------------------------
       CPCH19B         Best Deal Stores
        2/25/2002        Sales Report          Page   2
                          Store 2257

                  Dept
                 Number            Sales
                   111          $21,553.00
                   222           $2,199.00
                   333           $2,275.00
                   444              $62.00-
                   555           $1,690.00
                   666           $2,625.00

  Total for store 2257 is       $30,280.00 *
-------------------------------------------------------
       CPCH19B         Best Deal Stores
        2/25/2002        Sales Report          Page   3
                          Store 4464

                  Dept
                 Number            Sales
                   111           $6,055.00
                   222             $183.00-
                   333           $8,429.00
                   444           $9,920.00
                   555          $16,187.00
                   666           $9,999.00

  Total for store 4464 is       $50,407.00 *
-------------------------------------------------------
       CPCH19B         Best Deal Stores
        2/25/2002        Sales Report          Page   4
                          Store 5003

                  Dept
                 Number            Sales
                   111          $15,086.00
                   222           $9,111.00
                   333              $99.00-
                   444           $2,599.00
                   555          $10,585.00
                   666          $11,354.00

  Total for store 5003 is       $48,636.00 *
-------------------------------------------------------
       CPCH19B         Best Deal Stores
        2/25/2002        Sales Report          Page   5
                          Store 7315

                  Dept
                 Number            Sales
                   111           $2,536.00
                   222           $1,535.00
                   333           $2,235.00
                   444             $690.00
                   555             $565.00
                   666          $12,399.00

  Total for store 7315 is       $19,960.00 *
-------------------------------------------------------
```



```
     CPCH19B        Best Deal Stores
      2/25/2002       Sales Report          Page   6
                       Store 8950

                Dept
               Number           Sales
                 111              $85.00-
                 222           $9,550.00
                 333          $10,125.00
                 444          $10,088.00
                 555           $6,780.00
                 666          $11,250.00

Total for store 8950 is       $47,708.00 *
```

Figure 19.10 Report produced by program CPCH19B.

END-OF-CHAPTER AIDS

CHAPTER SUMMARY

1. There are two ways to define a printer file in a program.
    a. Program-described printer file.
    b. Externally described printer file.
2. Using an externally described printer file offers several advantages.
    a. Some changes can be made to a printer file without having to recompile the program using it.
    b. It eliminates the specifications necessary to define the report within the program.
    c. There are some software tools that will allow you to generate reports and produce the DDS.
    c. Several programs can share the same printer file.

KEY TERMS

Externally described printer file
`FORMATFILE`
`PAGNBR`
Printer file
Program-described printer file
`PRTF`
`SKIPA`
`SKIPB`
Skipping
`SPACEA`
`SPACEB`
Spacing

CHAPTER SELF-TEST

TRUE-FALSE QUESTIONS

___1. Externally described printer files are created in the same manner that physical files are created, except that they are defined as file type `PRTF`.
___2. Advancing the paper a fixed number of lines is called skipping.
___3. `SYSDATE` is a variable keyword that internally stores the system date.
___4. A printer file defined within a program is called a program-described printer file.
___5. The `PAGNBR` keyword is used to instruct the printer file to print page numbers.

FILL-IN-THE-BLANKS

1. Report specifications for a printer file that are described separately from any programs are called a(n) _______________ printer file.
2. _______________ are used to define printer files at the source level.
3. Advancing the paper to a specific line is called _______________.
4. The _______________ and _______________ keywords enable the output to print on a specific line.
5. Printer files must be defined with an entry in the _______________ paragraph in the _______________ `DIVISION`.

CHAPTER REVIEW QUESTIONS

GENERAL QUESTIONS

1. How is a printer file created?

2. What is the difference between program-described files and externally described printer files?
3. How are printer files similar to physical and logical files?

LAB ASSIGNMENT

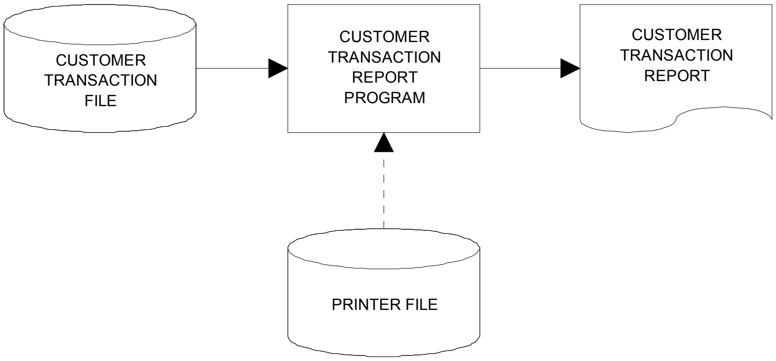
1. Create program `CPCH19A`:
   a. Use CODE/400 to enter the specifications for the externally described printer file CPCH19APRT that was described in this chapter.
   b. Compile the printer file into a printer file object (Type=PRTF).
   c. Modify a previous program or enter the specifications for program CPCH19A that was described in this chapter.
   d. Compile program CPCH19A into a program object.
   e. Test the program.

2. Create program CPCH19B:
   a. Use CODE/400 to enter the specifications for the externally described printer file CPCH19BPRT defined in this chapter.
   b. Compile the printer file into a printer file object (Type=PRTF).
   c. Modify a previous program or enter the specifications for program CPCH19B that was described in this chapter.
   d. Compile program CPCH19B into a program object.
   e. Test the program.

PROGRAMMING ASSIGNMENTS

1. Write a program that reads a Customer Transaction file and uses an externally described printer file to print a Customer Transaction Report, as shown in this problem definition:

   Systems flowchart


CUSTOMER TRANSACTION FILE
CUSTOMER TRANSACTION REPORT PROGRAM
CUSTOMER TRANSACTION REPORT
PRINTER FILE

Record description layout for customer transaction file

Field Description	**Type**	**Size**	**COBOL Field-name**
First Name	A	9	CT-FIRST-NAME
Middle Initial	A	1	CT-MIDDLE-INITIAL
Last Name	A	15	CT-LAST-NAME
Date of Transaction	L		CT-DATE-OF-TRANSACTION
Transaction Amount	P	7,2	CT-TRANSACTION-AMOUNT

Printer spacing chart for customer transaction report

```
                1          2          3          4           5          6          7           8
      123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
H   1  99/99/2099             Customer Transaction Report                  Page Z9
    2
H   3           Customer Name            Transaction Date    Amount of Transaction
    4
D   5           X. X. X-------------X         99/99/2099           $ZZ,ZZ$.99
D   6           X. X. X-------------X         99/99/2099           $ZZ,ZZ$.99
    7
```

2. Write a program that reads the records from the Payroll Master file and uses an externally described printer file to print a Payroll Report. The problem definition is shown below:

Record description layout for payroll master file

Field Description	Type	Size	COBOL Field-name
Employee Number	S	5,0	PM-EMPLOYEE-NUMBER
Employee Name	A	20	PM-EMPLOYEE-NAME
Territory Number	S	2,0	PM-TERRITORY-NUMBER
Office Number	S	2,0	PM-OFFICE-NUMBER
Annual Salary	P	7,0	PM-ANNUAL-SALARY
Social Security Number	S	9,0	PM-SOCIAL-SECURITY-NUMBER

Printer spacing chart for payroll report

```
                1          2          3          4           5          6          7           8
      123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
H   1  99/99/2099                  Payroll Report                          Page Z9
    2
H   3   Employee                          Terr     Office     Annual
H   4    Number    Employee Name           No.       No.      Salary      Soc Sec No.
    5
D   6    99999     X------------------X    99        99     $$,$$$,$$9    999-99-9999
D   7    99999     X------------------X    99        99     $$,$$$,$$9    999-99-9999
    8
```

3. The Light-Em-Up Utility Company has a Customer Usage file containing customer records for electricity and gas consumption. Write a program that reads the Customer Usage file and uses an externally described printer file to print a Customer Electricity and Gas Usage Report. The problem definition is below:

Record description layout for customer usage file

Field Description	Type	Size	COBOL Field-name
Account Number	S	5,0	CU-ACCOUNT-NUMBER
First Name	A	10	CU-FIRST-NAME
Last Name	A	15	CU-LAST-NAME
Street Address	A	20	CU-STREET-ADDRESS
Kilowatt Hours of Elect. Used	P	5,0	CU-HOURS-OF-ELECT-USED
Gas Used	P	5,0	CU-GAS-USED
Electricity Bill	P	5,2	CU-ELECTRICITY-BILL
Gas Bill	P	5,2	CU-GAS-BILL

Printer spacing chart for customer electricity and gas usage report

```
                1          2          3          4           5          6          7           8
      123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
H   1  99/99/2099        Customer Electticity and Gas Usage Report        Page Z9
    2
H   3      Account                              Elect    Elect     Gas      Gas
H   4      Number   Customer Name                Used     Bill     Used     Bill
    5
D   6      99999    X----X X----------------X   ZZ,ZZ9   $$$.99   ZZ,ZZ9   $$$.99
D   7      99999    X----X X----------------X   ZZ,ZZ9   $$$.99   ZZ,ZZ9   $$$.99
    8
```

4. Write a program that uses an externally described printer file to print a Sales Report. The data is contained in a physical Sales file that is not keyed and in no particular order. Thus, a logical file will have to be created over the physical file, so the program will receive records in Day sequence.

Notes:

a. There is a record for each sale made by an employee; thus, there are an undetermined number of input records.

b. Records must be in sequence by day number, which ranges from one to five (Monday-Friday).

Record description layout for sales file

Field Description	Type	Size	COBOL Field-name
Employee Number	S	3,0	SL-EMPLOYEE-NUMBER
Day Number	S	1,0	SL-DAY-NUMBER
Amount of Sale	P	7,2	SL-AMOUNT-OF-SALES

Printer spacing chart for weekly sales report

```
     |         1         2         3         4         5         6         7         8
     |12345678901234567890123456789012345678901234567890123456789012345678901234567890
H  1 | 99/99/2099   Weekly Sales Report      Page Z9
   2 |
H  3 |           Day             Total Sales
   4 |
T  5 |         Monday           $$,$$$.99
T  6 |         Tuesday          $$,$$$.99
T  7 |         Wednesday        $$,$$$.99
T  8 |         Thursday         $$,$$$.99
T    |         Friday           $$,$$$.99
     |
T    |     Total weekly sales is $$$,$$$.99 *
     |
```