

**Chapter
2**

**IDENTIFICATION
AND ENVIRONMENT
DIVISIONS**

*Get on the
Fast Track!*



TM

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- Purpose and function of the Identification Division.
- Purpose and function of the Environment Division.
- How to use headers in the Identification Division.
- Role of the Configuration and Input-Output Sections.

1 Identification Division

The Identification Division specifies information which serves to identify and document the program. It must be the first division coded in a COBOL program.

The Identification Division names the program.

The Identification Division consists of a division header and paragraph headers.

1.1 Identification Division Header

The purpose of the Identification Division Header is to indicate the beginning of a COBOL program.

It always begins on a line by itself in Margin A, and is followed by a period.

The abbreviation for IDENTIFICATION is ID.

```
|1      |7|8      |12
```

```
=====
```

```
001010  IDENTIFICATION DIVISION.
```

1.2 Identification Division - Paragraph Headers

COBOL provides for seven paragraph headers in the Identification Division. One paragraph header is required and six are optional.

These paragraph headers must begin in Margin A and be followed by a period and a space.

The words or sentences following the paragraph headers must be contained within Margin B.

The seven headers are:

1. PROGRAM-ID
2. AUTHOR
3. INSTALLATION
4. DATE-WRITTEN
5. DATE-COMPILED
6. SECURITY
7. REMARKS

PROGRAM-ID

The PROGRAM-ID header is required.

The purpose of this header is to name the program. It must be the first paragraph header coded in the identification division and be followed by a program name and a period.

The program name following PROGRAM-ID may be up to 8 alphabetic or numeric characters in length, the first of which must be alphabetic. For some environments the program name can be a maximum of 30 characters.

It is required and must be the first paragraph in the Identification Division.

```
|1      |7|8      |12
=====
001010  IDENTIFICATION DIVISION.
001020  PROGRAM-ID. PRWK0001.
```

AUTHOR

The purpose of this optional header is to identify the author of the program.

```
|1      |7|8      |12
=====
001010  IDENTIFICATION DIVISION.
001020  PROGRAM-ID. PRWK0001.
001030  AUTHOR.      JOHN Q. PROGRAMMER.
```

INSTALLATION

The purpose of this optional header is to identify the originating installation of the program. Most shops do not use this option.

```
|1      |7|8      |12
=====
001010  IDENTIFICATION DIVISION.
001020  PROGRAM-ID.  PRWK0001.
001030  AUTHOR.      JOHN Q. PROGRAMMER.
001040  INSTALLATION. SYSED.
```

DATE-WRITTEN

The purpose of this optional header is to document the date the program was written.

```
|1      |7|8      |12
```

```
001010  IDENTIFICATION DIVISION.  
001020  PROGRAM-ID.      PRWK0001.  
001030  AUTHOR.         JOHN Q. PROGRAMMER.  
001040  INSTALLATION.   SYSED.  
001050  DATE-WRITTEN.   JUNE 29, 2009.
```

DATE-COMPILED

The purpose of this optional header is to document the compilation date of the program. The actual date is not code. The compiler will insert the date in the source listing.

```
|1      |7|8      |12
```

```
001010  IDENTIFICATION DIVISION.  
001020  PROGRAM-ID.      PRWK0001.  
001030  AUTHOR.         JOHN Q. PROGRAMMER.  
001040  INSTALLATION.   SYSED.  
001050  DATE-WRITTEN.   JUNE 29, 2009.  
001060  DATE-COMPILED.
```

SECURITY

The purpose of this optional header is to document the security requirements of the program.

```
|1      |7|8      |12
```

```
001010  IDENTIFICATION DIVISION.  
001020  PROGRAM-ID.      PRWK0001.  
001030  AUTHOR.         JOHN Q. PROGRAMMER.  
001040  INSTALLATION.   SYSED.  
001050  DATE-WRITTEN.   JUNE 29, 2009.  
001060  DATE-COMPILED.  
001070  SECURITY.       TOP SECRET.
```

REMARKS

The purpose of this optional header is to allow for the inclusion of general comments about the program.

REMARKS are useful for describing the basic function, input and output files, and modifications made to the program. Instead of a REMARKS paragraph, many programmers use an asterisk in column 7 to represent a comment.

|1 |7|8 |12

```
001010        IDENTIFICATION DIVISION.  
001020        PROGRAM-ID.            PRWK0001.  
001030        AUTHOR.                JOHN Q. PROGRAMMER.  
001040        INSTALLATION.         SYSED.  
001050        DATE-WRITTEN.         JUNE 29, 2009.  
001060        DATE-COMPILED.  
001070        SECURITY.               TOP SECRET.  
001080        REMARKS.               THIS IS A SAMPLE COBOL PROGRAM  
001090                                TO DEMONSTRATE THE FOUR  
001100                                DIVISIONS OF COBOL.
```

2 Environment Division

The Environment Division describes the physical characteristics of the input-output equipment used by the program, and the processing which is dependent upon and specific the equipment.

The Environment Division consists of two sections, both are optional.

Configuration Section	Input-Output Section
-----------------------	----------------------

2.1 Environment Division Header

The purpose of this header is to identify the Environment Division of the COBOL program.

It must be the second division coded in a COBOL program.

It always begins on a line by itself in Margin A, and must be followed by a period.

2.2 Configuration Section

The Configuration Section is used to identify the:

Source Computer	The computer on which the program is compiled.
Object Computer	The computer on which the compiled program will execute.

At most installations, the source and object computers are the same.

The SOURCE-COMPUTER has a useful option. WITH DEBUGGING MODE activates a compile-time switch for debugging lines written in the source text. A debugging line is a statement that is compiled only when the compile-time switch is activated. In order to specify a debugging line in a program, code a D in column 7; this is the indicator area. Successive debugging lines can be included on the same line; however, each must have a D in column 7 and character strings cannot be split among across lines.

The Special Names paragraph designates a mnemonic name to a predefined IBM function name. It can be used in the WRITE statement for skipping to a new page.

```

|1      |7|8      |12
=====
001010  IDENTIFICATION DIVISION.
001020  PROGRAM-ID.          PRWK0001.
001030  AUTHOR.             JOHN Q. PROGRAMMER.
001040  INSTALLATION.       SYSED.
001050  DATE-WRITTEN.       JUNE 29, 2009.
001060  DATE-COMPILED.     JULY 1, 2009.
001070  SECURITY.          TOP SECRET.
001080  REMARKS.           THIS IS A SAMPLE COBOL PROGRAM
001090                      USED TO DEMONSTRATE THE FOUR
001100                      DIVISIONS OF COBOL.
001200  ENVIRONMENT DIVISION.
001300  CONFIGURATION SECTION.
001400  SOURCE-COMPUTER.    IBM Z9.
001500  OBJECT-COMPUTER.    IBM Z9.
001600  SPECIAL-NAMES.     C01 IS TOP-OF-PAGE.
    
```

2.3 Input-Output Section

The Input-Output Section is used to specify information necessary for controlling the transfer of data between the object program and an I-O device.

In addition to the required paragraph and section headers, every data file used by the program must be identified by a SELECT statement.

The keyword FILE-CONTROL and the period can be omitted if no file-control-paragraph is specified and there are no files defined in the program.

The FILE-CONTROL paragraph associates each file in the COBOL program with an external data set, and specifies file organization, access mode, and other information.

The ASSIGN clause associates the name of a file in a program with the actual external name of the data file.

The format of a SELECT statement for a sequential file is:

```
SELECT file-name ASSIGN TO assignment-name.
```

```
|1      |7|8      |12
=====
001010  IDENTIFICATION DIVISION.
001020  PROGRAM-ID.          PRWK0001.
001030  AUTHOR.             JOHN Q. PROGRAMMER.
001040  INSTALLATION.       SYSED.
001050  DATE-WRITTEN.       JUNE 29, 2009.
001060  DATE-COMPILED.
001070  SECURITY.           TOP SECRET.
001080  REMARKS.            THIS IS A SAMPLE COBOL PROGRAM
001090                          USED TO DEMONSTRATE THE FOUR
001100                          DIVISIONS OF COBOL.
001200  ENVIRONMENT DIVISION.
001300  CONFIGURATION SECTION.
001120  SOURCE-COMPUTER.    IBM Z9.
001130  OBJECT-COMPUTER.    IBM Z9.
001140  INPUT-OUTPUT SECTION.
001150  FILE-CONTROL.
001160          SELECT TRANS-FILE ASSIGN TO INTRANS.
```