CHAPTER 6
BASIC

6.1 INTRODUCTION

OS/78 BASIC* is an interactive programming language used in scientific and business environments to solve mathematical problems with a minimum of programming effort. It also is used by educators and students as a problem-solving tool and as an aid to learning through programmed instruction and simulation.

In many respects the BASIC language is similar to other programming languages (such as FORTRAN IV), but BASIC is aimed at facilitating communication between the user and the computer. BASIC simply requires that you type in the computational procedure as a series of numbered statements, making use of common English words and familiar mathematical notations. Because of the small number of commands necessary and its easy application in solving problems, BASIC is an easy computer language to learn. With experience, the advanced techniques available can be added in the language to perform more intricate manipulations or to express a problem more efficiently and concisely.

6.2 MAJOR COMPONENTS OF OS/78 BASIC

The BASIC subsystem has four major components.

1. BASIC Editor (BASIC.SV)
2. Compiler (BCOMP.SV)
3. Loader (BLOAD.SV)
4. BASIC Run Time System (BRTS.SV, BASIC.AF, BASIC.SF, BASIC.FF)

The BASIC editor is used to create and edit program source files. During this process, the editor creates a file called BASIC.WS containing the current program.

Once the program has been prepared for execution, entering a RUN command causes the editor to chain to the BASIC compiler. The compiler converts the statements in BASIC.WS into relocatable binary instructions.

Following compilation, the BASIC loader is automatically requested. The loader converts the relocatable binary data output by the compiler into executable form and loads the result into memory.

The BASIC loader then chains to the BASIC Run Time System (BRTS) which executes the program. The modules BASIC.AF, BASIC.FF, and BASIC.SF are overlays to BRTS.SV.

6.3 BASIC INSTRUCTION REPERTOIRE

BASIC instructions and commands can be grouped in three categories as follows:

1. BASIC Editor commands that allow you to

    a. Create or modify a program,
    b. Execute a program,
    c. Retrieve a program from diskette, and
    d. Save a program.

*BASIC is a registered trademark of the trustees of Dartmouth College.

2. BASIC statements, comprising the BASIC language, that are the building blocks used to create and structure BASIC programs.
3. BASIC Functions, represented by subroutines, that are built into BASIC primarily to facilitate problem solving activities.

6.4 CALLING BASIC

To enter the BASIC subsystem, type

```
.BASIC
```

in response to the Monitor dot. This command invokes the BASIC editor.

6.5 BASIC EDITOR COMMANDS

6.5.1 Using the BASIC Editor

The BASIC editor incorporates all the tools and capabilities necessary to create, correct, modify, execute, save and retrieve BASIC programs. After calling BASIC, the BASIC editor responds with the displayed query

```
NEW OR OLD---
```

Editing is now continued in one of two ways:

1. Typing NEW with a file name instructs the system to initiate the creation of a new file.
2. Typing OLD with a file name instructs the system to retrieve an old file containing a previously-generated program.

Figure 6-1 summarizes user actions implemented by the BASIC editor. This figure shows two arbitrarily selected file names (MAKER/ALTER). File names may contain no more than six alphanumeric symbols.

The left side of Figure 6-1 shows the types and patterns of activities that you ordinarily pursue after typing the NEW command. The right side of the illustration shows activities frequently carried out after typing the OLD command. Note that most of the BASIC editor commands appear on both sides of the figure. Only the sequence in which they are used differs.

6.5.2 BASIC Editor Commands

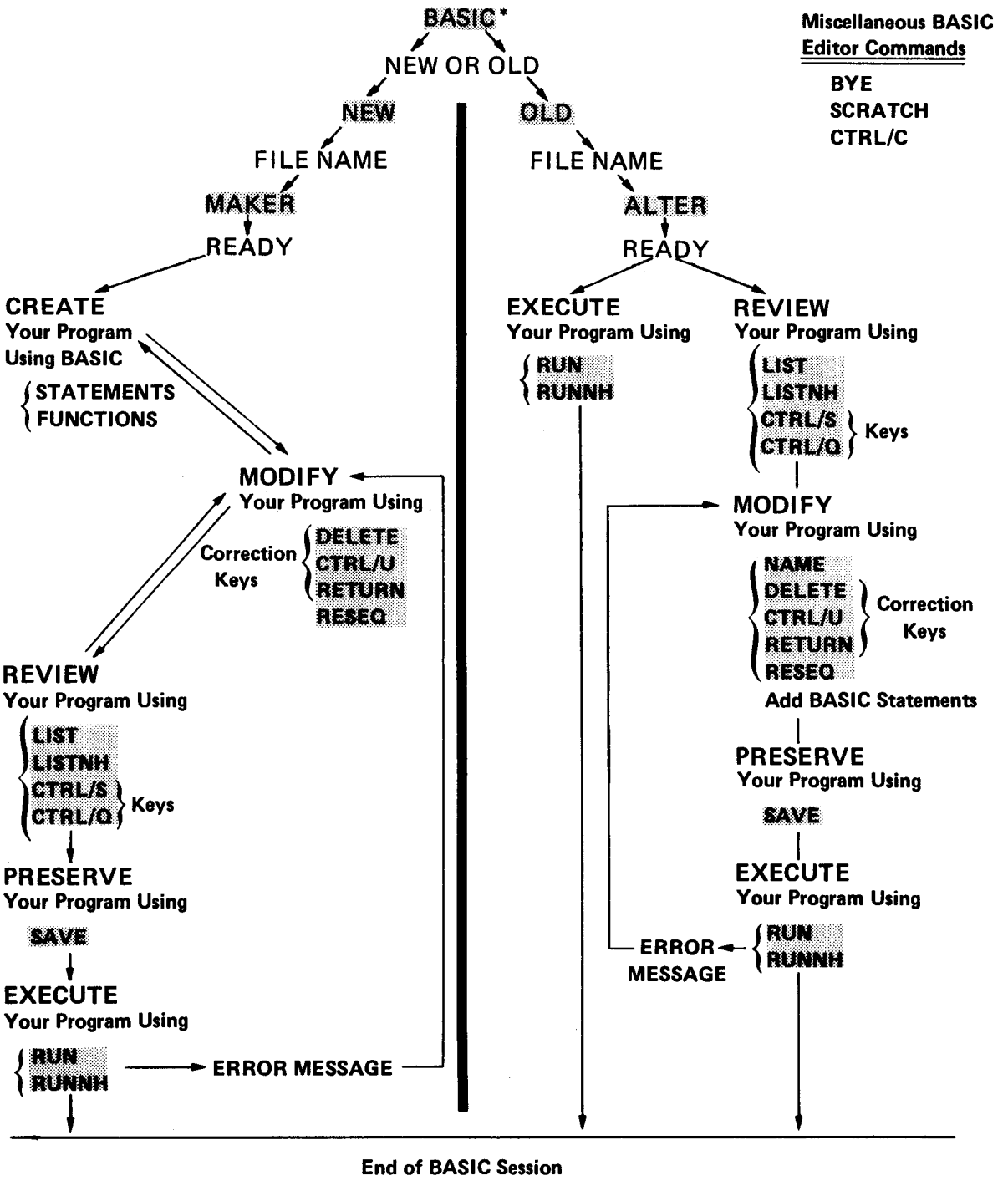
This section summarizes the BASIC editor commands. Letters that are not required in naming the command are shown as lower case in the Command/Parameter description. For example, only NE is required to be recognized by the BASIC editor as the NEW command.

NOTE

The RETURN key must be pressed following each BASIC editor command.

6.5.2.1 NEW Command – The NEW command clears the memory workspace and specifies the name of the program that is to be input.

Command	Parameters
NEw	file[.ex]
file.ex	is the new file name and extension of the program about to be typed in. If the extension is omitted, the editor assigns .BA.



1. *Keyboard Monitor Command
2. Shaded areas indicate user action at the console

Figure 6-1 BASIC Editor Commands and Related Uses

An alternate method is to type NEW without a file name, followed by the RETURN key. BASIC displays

```
FILE NAME --
```

in response to which the file name and extension is typed.

For example, to clear the workspace and name the new program "TEST.BA", type

```
NEW TEST
```

or

```
NE TEST.BA
```

6.5.2.2 OLD Command – The OLD command clears the memory workspace, and causes the editor to find a program on a diskette and place it into the workspace.

Command	Parameters
OLd	dev:file[.ex]
dev:file.ex	is the device, file name, and extension of the program on the disk. If the extension is omitted, BASIC assumes ".BA".

Another method is to type OLD without a file name. BASIC displays

```
FILE NAME --
```

in response to which the device, file name, and extension is typed. When no device is specified, the BASIC editor defaults to DSK (usually = SYS).

For example, to bring TEST.BA into the workspace from RXA1, type

```
OLD RXA1:TEST.BA
```

or

```
OL RXA1:TEST
```

6.5.2.3 LIST/LISTNH Commands – The LIST command displays the current program along with a header line, containing the program name, date, and the revision number of BASIC. The date is displayed only if the current date has been entered into the system.

Command	Parameters
LIst	[n]

If n is omitted all program statements in the workspace are displayed. When n is specified, line n and all subsequent lines are displayed. Type CTRL/O to terminate a listing.

For example, typing LIST (or LI) displays the program PROG.BA:

```
LIST

PROG BA       5A       26-JUL-77

10 FOR A=1 TO 5
20 PRINT A
30  NEXT A
40  END

READY
```

Typing LI 30 displays line 30 and all subsequent lines:

```
LI 30

PROG BA       5A       26-JUL-77

30 NEXT A
40 END

READY
```

The LISTNH command also displays the program statements in the workspace but without the header.

Command	**Parameters**
LISTNH	[n]
n	has the same effects as specified for the LIST command, but does not display the header.

6.5.2.4 SAVE Command — The SAVE command writes the program in the workspace onto the diskette as a permanently saved file. Do not confuse this command with the Monitor SAVE command.

Command	**Parameters**
SAve	[dev:file.ex]
dev	is the device on which you want to store your program.
file.ex	is the file name and extension that the program will have on the diskette. If both are left out, BASIC will use the current file name and extension of the program in the workspace. If only the extension is omitted BASIC assigns .BA. If DEV: is omitted, BASIC assumes the device is DSK.

In the following example, the program "TEST.BA" is in the workspace. To store it on RXA1 under the same file name and extension, type

```
SAVE RXA1:TEST.BA
```

or

```
SA RXA1:TEST
```