

See Chapter 16 for specific data. Special assembly language mnemonics may be used. Note that all machine language code must be preceded by the command MACHINE followed by a linenumber. See MACHINE and CODEL.

Examples:

```
1000 MACHINE 1100
1010 CODE"LDA,F2,2,STA,DF,4,RTS"
1100 END
```

The number in memory location 2F2 is loaded into the accumulator and then stored in memory location 4DF. Note that the memory locations are in hex, and that all machine language code must end in RTS. Also the X register must not be altered by the machine language code. If you use the X register, you must first save it and then restore it before the RTS. If you want to use a decimal number in a CODE statement, it must not exceed 255 and it must be preceded by !.

```
100 MACHINE 200
110 CODE"LDAIM,!15,STA,DF,4,RTS"
```

Here, the decimal number 15 is loaded into the accumulator (LDAIM stands for LOAD immediate--see Appendix D for a list of mnemonics). The 15 is then stored in 4DF.

Linenumbers can be used in the CODE command. They must be preceded by @ if used with branch commands, or by # if used with JMP or JSR. In the following example, if the number in 2F2 is non-zero, the program goes to 120 and returns; otherwise 1 is loaded to the accumulator and stored in 2F2 before the program returns. The last example illustrates the use of the # symbol.

```
100 MACHINE 200
110 CODE"LDA,2F,2,BNE,@120,LDAIM,1,STA,2F,2"
120 CODE"RTS"
200 END
```

```
100 MACHINE 200
110 CODE"LDA,2F,2,BNE,@120,LDAIM,1,STA,DF,4,JMP,#130
120 CODE"LDAIM,0,STA,DF,4"
130 CODE"INC,2F,2,RTS"
200 END
```

CODEL

Type: BASIC command

Formats:	CODEL(variablename)	CODEL(variablename+"L")
	CODEL(linenumber)	CODEL(variablename+"H")
	CODEL(variablename+integer)	CODEL(linenumber+"L")
	CODEL(linenumber+integer)	CODEL(linenumber+"H")

Description: Generates code for the address of a linenumber or a variable. If you use the +integer option (e.g., CODEL(T%+2)), the compiler adds the integer to the address of the linenumber or variable. If you use the +"L" or +"H" options, the compiler generates the low order or high order byte of