

CHAPTER 4

SYSTEM STATEMENTS

System statements are instructions to the ALGOL-20 translator which may be used to modify certain aspects of the translation process. That is, a system statement is executed by the translator at compile time rather than by the object program at execution time. System statements are executed as they are encountered by the translator as it scans once through the ALGOL source program and take effect immediately thereafter. All system statements except those marked with "+" may be used anywhere in the source program.

Each system statement is punched on a separate card which contains "SY" in columns 1 and 2. The system statement itself may be punched on the card anywhere between column 4 and the current right margin (see RIGHT MARGIN below).

A system statement generally has the form:

< statement name > < parameters >

Those system statements which have a fixed number of parameters are terminated by a blank following the last parameter. The rest of the card may be used for comments. The system statements which have a variable number of parameters are terminated by the end of the card, so comments cannot be included on such a card. System statements which may not contain a comment on the same card are marked with a dollar sign (\$).

Each system statement type will now be described and explained. In the following, "n" will always stand for an unsigned integer. The system statements marked with asterisks (*) control printing but will never themselves be printed. Printing of the other system statements may be suppressed by the system statement: "PRINT NO SYSTEM".

Blanks are not ignored when scanning system statements. There must be at least one blank between words and/or numbers and none in words or numbers.

PRINT CONTROL

(1) PAGE (No parameters)

The effect is to skip the compilation listing to the top of the next page. The PAGE statement itself will be printed on the first line of the new page.

*(2) LINE n

The effect is to upspace the printer by n lines. An attempt to upspace beyond the bottom of the current page will leave the paper at the top of the next page.

*(3) SINGLE (No parameters)

*(4) DOUBLE (No parameters)

These statements cause the compilation listing which follows to be printed with single or double line spacing, respectively. If neither statement is given, the translator assumes SINGLE.

*(5) INDENT	n	$K \leftarrow n$
INDENT	+n	$K \leftarrow K + n$
INDENT	-n	$K \leftarrow K - n$

The indentation constant, K, specifies the number of print positions to the right of the text left margin that the compilation listing will be printed. The translator normally assumes "IDENT 0". An IDENT card modifies K as given above. If this rule leaves K outside of the range $0 \leq K \leq 21$, then $K \leftarrow \max(0, \min(K, 21))$. Note the difference between "IDENT 2" and "IDENT +2": The former sets K to 2 and the latter increments K by 2. See Chapter 6c for a discussion of the format of the compilation listing.

\$(6) PRINT

The user has the ability to turn on or off the printing of various aspects of his source program. In general, if he does not specify otherwise, his source program along with octal addresses, notes from the translator, and system statements will be printed, while routines accessed from the symbolic library will not. The

* Not printed

\$ Comment not allowed

printing of each of ADDRESSES, NOTES, SYSTEM statements and LIBRARY routines may be controlled individually by the programmer by suitable PRINT statements. We have the following syntax:

```

<PRINT statement> ::= PRINT <parameter string>
<parameter string> ::= <parameter>, | <parameter string> <parameter>,
<parameter> ::= <control word> | NO <control word> | NO | EACH
<control word> ::= PROGRAM | ADDRESSES | NOTES | SYSTEM | LIBRARY

```

A PRINT statement is interpreted by treating each of the parameters in the parameter string in order from left to right across the card. The control word ADDRESSES refers to the octal addresses printed down the left side of the page. NOTES refers to possible error notes printed by the translator. (See Chapter 6b.) SYSTEM refers to system statements (except those which never print). LIBRARY refers to routines accessed from the symbolic library, as described below. PROGRAM refers to the listing of the source statements, along with notes, addresses and associated system statements.

A parameter consisting solely of a control word has the effect of turning on the printing of the corresponding part of the assembly listing as described above, while a parameter of NO followed by a control word turns off that part.

The parameter EACH is equivalent to the parameter string "PROGRAM, ADDRESSES, NOTES, SYSTEM, LIBRARY", and the parameter NO is equivalent to "NO PROGRAM, NO LIBRARY".

The parameter NO PROGRAM suppresses printing of the source program along with the associated addresses, notes and system statements, overriding any previous parameter of ADDRESSES, NOTES or SYSTEM. If PRINT PROGRAM is in effect, however, then NO NOTES, NO ADDRESSES or NO SYSTEM will suppress printing of these individual features. It is not possible to print notes, addresses or system statements without printing the corresponding source program images. If PRINT NO PROGRAM is in effect, PAGE and LINE have no meaning and thus are ignored by the translator.

The parameter LIBRARY has a function analogous to PROGRAM, except that LIBRARY takes effect only when a subsequent SY LIBRARY system

statement starts inserting library images; then, having PRINT LIBRARY (PRINT NO LIBRARY) in the "main" program text has the same effect as having PRINT PROGRAM (PRINT NO PROGRAM) as the first library image. These library images may themselves contain PRINT system statements; these will control printing only within the library segment, so that the PRINT status in effect when the SY LIBRARY statement was encountered will be restored at the end of the LIBRARY segment. If SY LIBRARY occurs within a set of library images, print control works, as described above, calling the outer set of library images the main program and the inner set the library images. The PRINT parameters are always "pushed down" when an SY LIBRARY system statement is encountered, and the "LIBRARY" switch on one level becomes "PROGRAM" switch on the next level.

In the absence of any PRINT system statements, the ALGOL-20 translator assumes PRINT EACH, NO LIBRARY.

MISCELLANEOUS

(7) RIGHT MARGIN n

Starting on the next card, the translator will scan column 4 through n for the text of ALGOL, WHAT, and system statements, where $40 \leq n \leq 80$. If n is not in the proper range, an error message is given and the right margin is not changed. The translator initially assumes RIGHT MARGIN 72.

(8) LIBRARY <identifier>

The translator inserts into the program at this point the segment of ALGOL source program text (generally a procedure) which is filed in the symbolic library under the name <identifier>.

†(9) n ABCONS

The translator will reserve n G-20 locations for storing "abcons" and n locations for storing "adcons" during both translation and execution. Abcons are numbers and alpha-numeric strings which do not appear in format primaries. Adcons are constants and temps

† Before the first begin only

for format replicators and small integer constants used as actual parameters to procedures.

If no ABCON statement is given, the translator assumes "200 ABCONS".

An ABCON statement may only occur at the very beginning of the program before the first begin.

(10) SEGMENT n1, n2

The integer n1 specifies the segment number, and must usually satisfy $1 \leq n1 \leq N$. Segments 1 through N are temporary segment and thus are not saved after the end of the user's run; permanent segments with numbers greater than N, are available upon request to the Computation Center. Since the number of temporary segments may change in the future, no value for N is given here. Its value can be found in the Users Manual, Section 1.5. The integer n2 specifies the number of files which will be required for the segment; each file contains 10240₁₀ words. If segment 1 requires 2 files, the next available segment is segment 3.

The number of "words" printed out at the end of an Algol program is the number that must be dumped out if the program is dumped as a segment.

See Chapter 6f for a complete discussion of segments.

(11) RELEASE WHAT

(12) RELEASE SYMBOLIC LIBRARY

The user who does not need WHAT or the symbolic library may reclaim the space used by these parts of the ALGOL compiler. This allows longer programs to be compiled. Since WHAT is below the library processor in memory, no space can be reclaimed until WHAT is released. The RELEASE's may be done, however, in either order and at any time during the compilation. Releasing WHAT reclaims /1400 (768₁₀) words; releasing the library will reclaim an additional /600 (384₁₀) words. Attempting to use WHAT or the symbolic library after it is released will cause a compile error.

(13) DEBUG n

This system statement is designed for the user with some knowledge of G-20 machine code and a general knowledge of the Algol-20 translator who wishes more specific information on the translation of a particular statement. This statement controls the printing of up to four columns of information after each ALGOL text card:

character scanned

postfix produced

code produced

internal variable equivalents

This information is printed one item per line as it is generated. An internal variable equivalent is printed out for each identifier declared. $n > 0$ turns on the printing, $n = 0$ turns it off.