INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Scientific Programming Department
I.C.T. 1900 Series

FORTRAN NOTE 6
3.5.65.

Trace System for FORTRAN 4 and Current 1900 FORTRAN

The first part of this note describes, from the user's point of view, the trace system to be implemented for FORTRAN 4. The second part of the note indicates how the current 1900 FORTRAN system differs.

This series of notes is intended for reference by the writers of the compilers and associated software. This particular note is being given a wider circulation than usual because it gives details of interest to programmers using the current 1900 FORTRAN compiler.

Trace System for FORTRAN 4

1. Outline

The FORTRAN Trace System provides assistance in diagnosing faults in a FORTRAN program by providing additional output from the program while it is running.

To use the trace system, the program, or alternatively only selected segments of the program, must be compiled in trace mode. When this is done, a special library segment is included in the object program and is called once during the execution of every statement.

If trace is to be used the source program must define one output and one input device to be 'monitor' peripherals (the method for FORTRAN 4 is described on page 4 of Note 2). The same devices may also be used for normal input or output.

2. Steering List

To operate the trace system a steering list must be prepared defining certain labelled statements as "triggering statements" and specifying the action required at these triggering statements.

The list is prepared as follows:-

n	NAME OF SEGMENT	Heading line	
l_1	d_1 c_1	Detailed line	
l_2	d_2 c_2	" "	List for one segment
.....		" "	
l_n	d_n c_n	" "	
n	NAME OF SEGMENT		
l_1	d_1 c_1		
.....			Lists for other segments
.....			
0			

The heading line for each segment contains an integer n in columns 1 to 6 which defines the number of detaillines to follow. The name of the segment must be given starting in column 7 and must not contain any spaces.

The detail line(s) consist of three integers separated by one or more spaces.

The first integer (l) is the label of a statement within the segment. This label must not appear in more than one detail line.

The second integer (d) specifies a "delay" and may be in the range -99 to 2047.

The third integer (e) specifies a "count" and may be in the range 0 to 4095.

After the lists for each segment to be traced, there must be a terminating line containing a zero in column 1.

If the steering list is prepared on paper tape, "tab" must not be used.

3. Output

When the object program executes a triggering statement:-

1. Any trace output still in progress from a previous triggering statement is terminated.

2. A line is printed giving the label number and segment name of the triggering statement. This is preceded by a blank line.

3. If the delay (d) is negative and the count (c) is non-zero, then statement lines are printed starting d statements in the past up to the present statement (except that at most c lines are printed).

4. Arrangements are then made to print statement lines as they are executed commencing with the current statement if $d \leq 0$ or after a delay d such that the total number of lines (both past and future) = c.

Each statement line consists of:-

1. The line number. The triggering statement itself ≙ line number 0. Other lines are numbered positively or negatively from this.

2. The label if any.

3. An abbreviation of the type of statement.

4. V if the overflow indicator is set.

5. The result of the statement, if applicable.

If a change of segment occurs between statement lines, the name of the new segment is given on a line by itself.

The end of a DO loop, although not written as a statement in the source program, is treated as a statement (type LOOP) by the trace system and is "executed" each time around the loop.

The statements, their abbreviations and results are as follows:-

STATEMENT	ABBREVIATION	RESULT
Arithmetic assignment	ARTH	Value of right hand side
Logical assignment	LOGC	Value of r.h. side: TRUE or FALS.
Arithmetic IF	IF	Value of expression in parenthesis.
Logical IF	LIF	Value of logical expression: TRUE or FALS.
GO TO	GO	
DO (	DO	m_1
(	LOOP	m_2-i
Computed GO TO	CGO	i
Assigned GO TO	AGO	
READ	READ	
WRITE	WRTE	
PAUSE	PAUS	
STOP	STOP	
CALL	CALL	
RETURN	RETN	
ENDFILE	ENDF	
CONTINUE	CONT	
BACKSPACE	BACK	
REWIND	REW	
ASSIGN	ASGN	

4. Notes on Operation of Trace System

a) The steering list will normally be read immediately after the object program has been loaded, but a new steering list may be read at any time and will replace any previous list, after which the program may be restarted.

b) Trace output may be switched on or off at any time during the execution of the object program.

c) When the object program is run in trace mode all array subscripting is checked and:-

ARRAY SUBSCRIPT ERROR

will be output if the selected element does not lie within the confines of the array. The program cannot be continued beyond this point.

d) Any time after the program has run it is possible to output the last 100 statements executed. This is useful if the program should reach an "illegal" operation or otherwise comes to an undesired halt.

Trace System for current 1900 FORTRAN

This is similar to the system described above. The method of assigning monitor peripherals is to call them MONIN and MONOUT for input and output respectively. The names are assigned in PERIPHERAL Statements in the normal way, e.g.

PERIPHERAL (MONIN, TRO), (MONOUT, LPO)

The list of possible statements given in 3 above is slightly shorter: Logical assignment, Logical IF, Assigned GO TO, BACKSPACE, REWIND and ASSIGN are not available.

K.F. James.