

CGTM #92
M.A. Fisherkeller
June 1970

**MASTER COPY
DO NOT REMOVE**

An Interactive Debugging Program

This memo will describe an interactive debugging routine for use with IBM/360 Assembler language programs. The user console may be either an IBM 2250 or the Graphic Interpretation Facility. Communication with the debugging program is via a DEBUG MACRO. The MACRO is inserted in the user program at points where interaction is desired. For the ability to refer to symbols in the user program, the TEST option is selected at assembly time and the symbol table produced by TESTSTRAN is then used by the debug program. The TEST option must also be specified for the LINKEDIT processor for the symbol table to be included in the load module. If the TEST option is omitted, all references must be made to registers and hex addresses. At least one DEBUG MACRO must be included in the user program to establish communication with the debugging routine.

When the first DEBUG MACRO is encountered, the symbol table is read into core. If the table overflows the area reserved for it, a message appears on the screen and the user may cancel his run or continue with the partial table. If he continues or if there is no table overflow, a list of user options is displayed. The options are:

- DISPLAY CELL
- WRITE TO CELL
- DISPLAY GP REGS.
- DISPLAY FP REGS.
- TRANSFER CONTROL
- CONTINUE
- CANCEL
- INSERT BREAKPOINT
- REMOVE BREAKPOINT

The lightpen is used to select an option. The teletype or the 2250 keyboard is used for input. On input, parameters are separated by commas and the final delimiter is the semi-colon. The delimiter only can be used to cancel an input request. The options will now be described.

DISPLAY CELL

A message appears indicating that a symbol or a 6 character hex address (followed by a slash) may be typed in. If a symbol is typed that can be located in the symbol table, the name of the associated CSECT (if any), the hex location and the contents are displayed. For instructions, the mnemonic is also shown. For non-instructions, eight bytes are displayed.

WRITE TO CELL

This option is used to store into core, but may only be used to change the contents of a symbolic location displayed in a previous DISPLAY CELL request. The value entered must be either a hex value or a character string.

DISPLAY GP REGS.

The contents of the general purpose registers are displayed in hex. The lightpen may be used to select a register in order to change its contents. The input must be eight hex digits. To return to the option list display, the last line of the display is lightpenned.

DISPLAY FP REGS.

The contents of the floating point registers are displayed in hex and in E24.14 format. The lightpen may be used to select a register in order to change its contents. The input must be in E24.14 format. To return to the option list display, the last line of the display is lightpenned.

TRANSFER CONTROL

This option may be selected to return to the user program at a point other than the current return point. The current return point is either where a DEBUG MACRO was encountered or at a program interruption point. A base register must be designated for the transfer of control. If a GP REG. contains the base address or has been loaded via the DEBUG option, the register number is typed in. Alternatively, a register number and a symbol which contains the base address may be entered. From a program interruption point, the register is not actually used for the return, but the PSW in the PIE is changed to the value requested. As an example, if your program begins at symbolic location START, it may be rerun by typing 15,START;.

CONTINUE

This option causes transfer of control to the user's program at the current return point.

CANCEL

This option terminates the user program with an ABEND 555 and dump.

INSERT BREAKPOINT

The DISPLAY CELL message appears and a symbol or a 6 character hex address may be entered. Control will be trapped at this location and the options of the DEBUG program may be exercised. Only one breakpoint may be active at a time, so if one is already active at the time of this request, the first one is removed before the new breakpoint is made active. The breakpoint feature is implemented by replacing the instruction at the breakpoint location with a zero operation code. This code is then trapped by operation exception.

When the CONTINUE option is selected after a breakpoint, the user's instruction is executed before returning to the user program. Due to this implementation, the following restrictions on breakpoint locations hold:

Because the instruction at the breakpoint location is executed with an Execute instruction -- IT CANNOT BE AN EXECUTE INSTRUCTION.

Because control must be returned to the system from the interrupt routine after the breakpoint instruction has been executed-- IT CANNOT BE A BRANCH.

Because the system will restore Registers 14-2 from the Program Interruption Element after the instruction has been executed -- ANY OPERATION ON REGISTERS 14-2 WILL BE NULLIFIED.

Because the breakpoint instruction is executed in an interrupt routine--ANY PROGRAM CHECK IN THE INSTRUCTION ITSELF WILL CAUSE TASK TERMINATION.

REMOVE BREAKPOINT

The instruction at the breakpoint location will be restored to the user program.

PROGRAM INTERRUPTION

All program interruptions are trapped and a message displayed giving the type of interruption, the location in hex, the associated CSECT name (if any) and the nearest symbol (if any). The options described above may be exercised but in the case of an imprecise interrupt, the CONTINUE option will produce unpredictable results. Also in the case of an imprecise interrupt, if there are multiple exceptions, only one type will be indicated in the interrupt message.

ABNORMAL TERMINATIONS

User Code

555	CANCEL option exercised
444	User's load module cannot be found
333	Program interruption occurred outside the load module.