

RTS: 161 Illegal Intermediate Code - What does this mean and how can you find the cause?

The on-line help for this error states:

"The intermediate code which is currently being processed is not valid code. You are probably trying to execute a corrupted file or one which has not been submitted to your COBOL system successfully."

A common cause of this error is when the loaded copy of the file, not the file on disk, gets corrupted as a result of a programming logic error

There are a number of scenarios that can cause this corruption and a number of ways of finding them. Mostly, it is a coding error that causes this error - try compiling your INT code to GNT code - if this works it means the INT code produced by the compiler is valid and therefore it is likely that the RTS: 161 is being generated because of a end-user program bug.

Potential causes:

1. The corruption occurs in a CALLED program - the main program calls the subprogram with parameters which don't match correctly causing Data Division and / or Procedure Division code to be corrupted.
2. The corruption occurs because a subscripted table is being written to beyond its boundaries.
3. The program is making use of Reference Modification and is erroneously overwriting memory.

Suggested remedies:

1. Check the length of the parameters in the "CALL 'progrname' USING ..." statement in the CALLING program with the length of the parameters specified on the "PROCEDURE DIVISION USING..." statement in the CALLED program - they must match.
2. Try compiling your program with additional directive "BOUND". This directive specifies that the subscript or index value is to be checked to ensure it is within the limits defined by the OCCURS clause. If it is the case that you are exceeding your subscript limit, you will get an RTS: 153 Subscript out of range error at the point that this occurs while running or debugging your program.
3. Try compiling your program with the additional directive "SSRANGE". This directive specifies that, at run-time, reference modified items are checked for negative or zero value. If this doesn't help, add a large dummy data item at the very end of Working-Storage and monitor this. If this changes value, then something bad is happening! You could set a breakpoint on data change on this variable and the debugger should then stop at the line causing the offending behavior.

If none of the above suggestions help, then it is possible that there is an issue with the Micro Focus compiler and/or debugger. You should report the issue to your local SupportLine group for further analysis.