

ADL Guide 2020

Ross, Tom
Captain COBOL

Advanced usage of compiler options and how to
use the compiler message exit!



COBOL compiler options defaults

IGYCDOPT

- This is the program for setting installation default compiler options
 - Syntax is different from CBL/PROCESS and PARM= methods for setting options
 - Lookup sample job IGYWDOPT in Customization Guide for instructions on how to install defaults
- Some interesting options that you might not know about:
 - ALLOWCBL=YES|NO
 - You can disallow individuals from setting compiler options within programs using CBL/PROCESS statements
 - ALLOWCOPYLOC=YES|NO
 - Allow or disallow COPYLOC compiler option, which tells the compiler to look in more places for COPY members, such as z/OS Unix file system (zFS)
 - ALLOWDEFINE=YES|NO
 - Allow or disallow the DEFINE compiler option, which is used to define compilation variables for conditional compilation

COBOL compiler user exits

EXIT compiler option

- Used to allow customization of compiler behavior
 - Compiler invocation example: `EXIT(MSGEXIT(mod5))`
 - For defaults(in IGYCDOPT), `MSGEXIT=mod5`
- Use ADEXIT to post process the ADATA if you want to
- Use INEXIT to set up your own ‘macro phase’ to preprocess source
- Use LIBEXIT to process the COPY members as they are read in
- Use PRTEXTIT to modify the listing as it is written
- Use MSGEXIT to change severity of compiler diagnostic and other messages

IGYMSGXT sample MSGEXIT

MSGEXIT compiler option

- In compiler dataset SIGYSAMP there is a sample MSGEXIT program, IGYMSGXT
 - IGYMSGXT is explained in the Programming Guide
 - You can reduce severity of W level messages
 - You can suppress W or I level messages
 - You can raise E level messages to S

IGYMSGXT sample MSGEXIT

To use this user exit program, make the link-edited load module/program object available to your compiles that will use the MSGEXIT suboption of the EXIT compiler option. Also, the name should be changed, since IBM recommends that you avoid having programs with names that start with IGY.

Sample steps to take:

- 1) Make your customizations
- 2) Change program name (E.G. MYEXIT)
- 3) Compile and link into a dataset
- 4) Include that dataset in your compile JCL concatenation for the compile step.

If you link into USER.COBOLLIB:

```
//COBOL.STEPLIB DD DSN=SYS1.SIGYCOMP,DISP=SHR  
//              DD DSN=USER.COBOLLIB,DISP=SHR
```

- 5) Finally, compile your programs with the EXIT compiler option, EG:

```
EXIT (MSGEXIT (MYEXIT) )
```

IGYMSGXT sample MSGEXIT

```
Procedure Division Using  EXIT-TYPE          EXIT-OPERATION
                          EXIT-RETURNCODE  EXIT-WORK-AREA
                          EXIT-DUMMY       EXIT-MESSAGE-PARMS
                          EXIT-STRING      EXIT-DUMMY
                          EXIT-DUMMY       EXIT-DUMMY.
```

```
    Compute EXIT-RETURNCODE = 0
    Evaluate TRUE
```

```
*****
* Handle a bad invocation of this exit by the compiler.          *
* This could happen if this routine was used for one of the     *
* other EXITs, such as INEXIT, PRTEXTIT or LIBEXIT.             *
*****
```

```
    When EXIT-TYPE Not = 6
        Move EXIT-TYPE to EXIT-TYPEN
        Display '**** Invalid exit routine identifier'
        Display '**** EXIT TYPE = ' EXIT-TYPE
        Compute EXIT-RETURNCODE = 16
```

IGYMSGXT sample MSGEXIT

```
*****
* Handle the customize message severity call to this exit      *
*       Display information about every customized severity.    *
*****

When EXIT-OPERATION = 5
*       Display 'MSGEXIT called with MSGSEV'
       If EXIT-MESSAGE-NUM < 8000 Then
           Perform Error-Messages-Severity
       Else
           Perform FIPS-Messages-Severity
       End-If

*       If EXIT-RETURNCODE = 4 Then
*           Display '>>>> Customizing message ' EXIT=MESSAGE-NUM
*               ' with new severity ' EXIT=USER-SEV ' <<<<'
*           If EXIT-MESSAGE-NUM > 8000 Then
*               Display 'FIPS sev =' EXIT-DEFAULT-SEV-FIPS '<<<<'
*           End-If
*       End-If
```

IGYMSGXT sample MSGEXIT

This code is for your optional use to debug this module, you can display what is going on as the exit is run! Just remove the asterisks from column 7 and you will get display messages in your compile output about customizations!

```
*      If EXIT-RETURNCODE = 4 Then
*          Display '>>>> Customizing message ' EXIT=MESSAGE-NUM
*              ' with new severity ' EXIT-USER-SEV ' <<<<'
*          If EXIT-MESSAGE-NUM > 8000 Then
*              Display 'FIPS sev =' EXIT-DEFAULT-SEV-FIPS '<<<<'
*          End-If
*      End-If
```


IGYMSGXT sample MSGEXIT

```
*****  
*      ERROR MESSAGE      PROCESSOR                               *  
*****
```

Error-Messages-Severity.

```
*      Assume message severity will be customized...  
      Compute EXIT-RETURNCODE = 4
```

```
      Evaluate EXIT-MESSAGE-NUM
```

```
*****  
*      Change severity of message 1154(W) to 12 ('S')           *  
*      This is the case of redefining a large item             *  
*      with a smaller item, IBM Req # MR0904063236             *  
*****
```

```
      When(1154)
```

```
          Compute EXIT-USER-SEV = 12
```

IGYMSGXT sample MSGEXIT

```
*****  
*   Example for message 1154   *  
*****
```

```
000016          1 G1.  
000017          2 A pic X.  
000018          2 Ar redefines A pic X(95).
```

```
==000018==> IGYDS1154-S "AR" redefined a smaller item. The program  
was accepted as written.
```

IGYMSGXT sample MSGEXIT

```
*****
*   Modify the severity of RULES messages to enforce coding      *
*   standards or highlight coding that you want to avoid.      *
*   Here are the message numbers and what they flag:          *
*   1158  RULES(NOOMITODOMIN)   Missing min idx in ODO table def*
*   1348  RULES(NOEVENTPACK)    Even digit packed-decimal items *
*   1353  RULES(NOSLACKBYTES)   Slack bytes within records      *
*   1379  RULES(NOSLACKBYTES)   Slack bytes between records      *
*****
      When(1158)                *> Disallow omitting ODO table min
          Compute EXIT-USER-SEV = 12
      When(1348)                *> Disallow even-digit Comp-3
          Compute EXIT-USER-SEV = 12
      When(1353) When(1379)     *> Disallow slack bytes
          Compute EXIT-USER-SEV = 12
```

IGYMSGXT sample MSGEXIT

000016 1 G1.

000020 2 occurs 10 times depending on Obj.

==000020==> IGYDS1158-S **RULES(NOOMITODOMIN)** The first integer in the "OCCURS" clause was omitted. Please note: The integer that was specified was interpreted as the maximum number of elements in the table.

==000020==> IGYDS1379-S **RULES(NOSLACKBYTES)** Table item "FILLER" caused the compiler to add 3 slack bytes between table occurrences.

000021 3 CCCC POINTER SYNC.

000022 3 C pic 9(4) PACKED-DECIMAL.

==000022==> IGYDS1348-S **RULES(NOEVENPACK)** Data item "C" was defined with an even number of digits, but "PACKED-DECIMAL" data items should always be defined with an odd number of digit positions.

000023 3 CC POINTER SYNC.

==000023==> IGYDS1353-S **RULES(NOSLACKBYTES)** Data item "CC" caused the compiler to add 3 bytes into the containing group item.

IGYMSGXT sample MSGEXIT

```
*****
*   Modify the severity of RULES messages to enforce coding      *
*   standards or highlight coding that you want to avoid.      *
*   Here are the message numbers and what they flag:           *
*   2159  RULES(NOENDPERIOD)      Cond stmt terminated by period *
*   2262  RULES(NOUNREFALL)       Unref'd items (source/copybook) *
*   2262  RULES(NOUNREFSOURCE)    Unref'd items (source only)    *
*****
      When(2159)                *> Disallow period-termination
          Compute EXIT-USER-SEV = 12    *> of conditional stmts
      When(2262)                *> Disallow unref'd data items
          Compute EXIT-USER-SEV = 12
```

IGYMSGXT sample MSGEXIT

000010 1 P pointer.

IGYSC2262-S **RULES(NOUNREFSOURCE)** No references to "P", nor any of its subordinates if "P" is a group item, were found in the program.

000092 If ec = 0 then

==000092==> IGYPS2159-S **RULES(NOENDPERIOD)** The scope of conditional statement "IF" was terminated by a period on line 99 instead of by an explicit scope terminator.

```
000093           1           Display "MSGXIT successful, " ok " tests"
000094           1           Move 2 to return-code
000095                   Else
000096           1           Display "MSGXIT unsuccessful, "
000097           1                   ec " failures, " ok " passed "
000098           1           Move 16 to return-code
000099                   .
```

IGYMSGXT sample MSGEXIT

```
*****
*   Change severity of 'optimization' messages to suppress them
*   so that compilation Return Code can be zero (RC=0)
* 7300: The code from lines &2 in program '&1' can never
*       be executed and was therefore discarded.
* 7301: A zero base was raised to a zero power in a numeric
*       literal exponentiation. The result was set to 1.
* 7302: A zero base was raised to a negative power in a numeric
*       literal exponentiation. The result was set to 0.
* 7304: An exception "&1" occurred while processing numeric
*       literals. The result of the operation was set to zero.
* 7307: This statement may cause a program exception at execution time.
* 7309: There may be a loop from the "PERFORM" statement at "
*       "PERFORM (line &1)" to itself.
* 7312: Procedure starting at 'proc-name' (line 'num') was copied one
*       or more times to be inlined for PERFORM statements.  xxx total
*       were generated from copying that procedure.
*****
      When(7300) When(7301) When(7302) When(7304)
      When(7307) When(7309) When(7312)
      Compute EXIT-USER-SEV = -1      *> Suppress the messages
```

IGYMSGXT sample MSGEXIT

```
*****  
*   Change severity of message 7311(W) to 12 ('S')           *  
*   This is the case of INITCHECK messages about           *  
*   uninitialized data items                               *  
*****
```

When(7311)

Compute EXIT-USER-SEV = 12

IGYMSGXT sample MSGEXIT

Unitialized data items is causing problems with z14 Vector Packed Decimal Facility instructions!

Example:

```
MOVE  X'F040F1' TO WS-PURCHASE-X
```

*> Example of random data

```
MOVE WS-PURCHASE-X TO PACK-NUM
```

*> Alpha is treated as numeric

*> display when receiver is numeric

*> The MOVE will ABEND 0C7 with ARCH(12), but no ABEND with ARCH(11)

*> or lower!

IGYMSGXT sample MSGEXIT

```
000014          1 ok pic 999.
```

```
000088          Procedure division.
```

```
000089          Add 1 to OK
```

```
89  IGYCB7311-S   The data item 'OK' may be used at this statement  
                   before it is set.
```

*> Note well! INITCHECK will not work for programs with RC=12, so if
*> we change messages to 'S' level and we get some instances of the RULES
*> suboptions messages, then INITCHECK will not be performed since the
*> code generator is no called for programs with RC=12.

*> I tried using COMPILE to force the backend to run, but INITCHECK
*> did not work! It only works for 'clean' programs, so another option
*> is to customize the example messages to '8' instead of '12'
*> You can still customize 7311 to 'S' since it is a backend message.

IGYMSGXT sample MSGEXIT

Ever heard of FIPS messages? FIPS = Federal Information Processing Standard

FLAGSTD is the compiler option that tells the compiler to do COBOL standards-based flagging example:

```
// PARM.COBOL= 'FLAGSTD (M) '
```

To see all messages that the compiler can put out, compile a program with PROGRAM-ID. ERRMSG. and look at the resulting listing!

There is a separate section in the IGYMSGXT sample to take advantage of FIPS messages:

IGYMSGXT sample MSGEXIT

```
*****
*   FIPS MESSAGE    PROCESSOR                               *
*****
Fips-Messages-Severity.
*   Assume message severity will be customized...
    Compute EXIT-RETURNCODE = 4

*   Convert numeric FIPS(FLAGSTD) 'category' to character
*   See the Programming Guide for description of FIPS category
    EVALUATE EXIT-DEFAULT-SEV

    . . .
```

IGYMSGXT sample MSGEXIT

EVALUATE EXIT-DEFAULT-SEV

When 81

MOVE 'D' TO EXIT-DEFAULT-SEV-FIPS

When 82

MOVE 'E' TO EXIT-DEFAULT-SEV-FIPS

When 83

MOVE 'H' TO EXIT-DEFAULT-SEV-FIPS

When 84

MOVE 'I' TO EXIT-DEFAULT-SEV-FIPS

When 85

MOVE 'N' TO EXIT-DEFAULT-SEV-FIPS

When 86

MOVE 'O' TO EXIT-DEFAULT-SEV-FIPS

When 87

MOVE 'Q' TO EXIT-DEFAULT-SEV-FIPS

When 88

MOVE 'S' TO EXIT-DEFAULT-SEV-FIPS

When Other

Continue

End-Evaluate

IGYMSGXT sample MSGEXIT

```
*****
*   Example of using FIPS category to force coding
*   restrictions.  This is not a recommendation!
*
*   Ex:  Change severity of all OBSOLETE item FIPS
*         messages to 'S'
*****

*       If EXIT-DEFAULT-SEV-FIPS = 'O' Then
*         Display '>>>> Default customizing FIPS category '
*           EXIT-DEFAULT-SEV-FIPS ' msg ' EXIT-MESSAGE-NUM '<<<<'
*         Compute EXIT-USER-SEV = 12
*       End-If
```

IGYMSGXT sample MSGEXIT

Evaluate EXIT-MESSAGE-NUM

```
*****  
*           Change severity of message 8062(O) to 8 ('E')  
*           8062 = GO TO without proc name (For ALTER)  
*****
```

When(8062)

Compute EXIT-USER-SEV = 8

```
*****  
*           Change severity of message 8235(E) to 8 (Error)  
*           to disallow Complex Occurs Depending On  
*           8235 = Complex Occurs Depending On  
*****
```

When(8235)

Compute EXIT-USER-SEV = 08

IGYMSGXT sample MSGEXIT

Changing the 'severity' of FIPS messages changes them into diagnostic messages! They will no longer appear with other FIPS messages and will instead appear with other diagnostic messages.

BEFORE CUSTOMIZATION:

Line.Col	Code	FIPS message text
	IGYSC8254	"NODYNAM" compiler option: nonconforming nonstandard, IBM extension to ANS/ISO 1985.
	IGYSC8259	"SEQUENCE" compiler option: nonconforming nonstandard, IBM extension to ANS/ISO 1985.
8.21	IGYDS8188	"USAGE IS COMPUTATIONAL-3": nonconforming nonstandard, IBM extension to ANS/ISO 1985.
96.12	IGYPS8062	"GO TO" statement without procedure-name: nonconforming standard, ANS/ISO 1985 high subset.

IGYMSGXT sample MSGEXIT

Changing the 'severity' of FIPS messages changes them into diagnostic messages! They will no longer appear with other FIPS messages and will instead appear with other diagnostic messages.

AFTER CUSTOMIZATION:

LineID	Message code	Message text
79	IGYDS1154-S	"FR" redefined a smaller item. The program was accepted as written.
96	IGYPS8062-E	"GO TO" statement without procedure-name: nonconforming standard, ANS/ISO 1985 high subset.

IGYMSGXT sample MSGEXIT

Sample program with GO TO. (NOT Customized)

```
000093          Alter Alteredgoto To Goback-para
000094          .
000095          Alteredgoto.
000096          Go To.
=>000096.12 IGYPS8062  "GO TO" statement without procedure-name:
                       nonconforming standard, ANS/ISO 1985 high subset.

000097          Goback-para.
```

IGYMSGXT sample MSGEXIT

Sample program with GO TO. (Customized)

```
000093          Alter Alteredgoto To Goback-para
000094          .
000095          Alteredgoto.
000096          Go To.

==000096==> IGYPS8062-E "GO TO" statement without procedure-name:
              nonconforming standard, ANS/ISO 1985 high subset.

000097          Goback-para.
```

IGYMSGXT sample MSGEXIT

Sample program with complex ODO. (NOT Customized)

```
000021          2 A pic X.  
000022          2 Ar redefines A pic X(95  
000023          2 B pic S9(8) binary sync.  
000024          2          occurs 10 times depending on Obj.  
000025          3 CCCC POINTER SYNC.  
000026          3          occurs 10 times depending on Obj.
```

=>000026.10 IGYDS8235 Complex "OCCURS DEPENDING ON": nonconforming
nonstandard, IBM extension to ANS/ISO 1985.

IGYMSGXT sample MSGEXIT

Sample program with complex ODO. (Customized)

```
000021          2 A pic X.  
000022          2 Ar redefines A pic X(95).  
000023          2 B pic S9(8) binary sync.  
000024          2          occurs 10 times depending on Obj.  
000025          3 CCCC POINTER SYNC.  
000026          3          occurs 10 times depending on Obj.
```

==000026==> IGYDS8235-E Complex "OCCURS DEPENDING ON": nonconforming
nonstandard, IBM extension to ANS/ISO 1985.

IGYMSGXT sample MSGEXIT

There are many many FIPS messages, so you might want to make the other FIPS messages 'go away'! Like this

```
*****  
*      Message severity Not customized  
*****
```

When Other
Compute EXIT-USER-SEV = -1

IGYMSGXT sample MSGEXIT

Finally, how do you know if messages got customized, or maybe more importantly, suppressed? At the bottom of the listing!

Messages	Total	Informational	Warning	Error	Severe
Terminating					
Printed:	50	7	3	13	27
Messages suppressed by MSGEXIT: 45					
Messages with severity modified by MSGEXIT: 40					