Migrating to IBM COBOL for AIX

Migrating to IBM COBOL for AIX

Note
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 27.

Second Edition (June 2015)

This edition applies to IBM COBOL for AIX, V3.1 or later compilers until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

You can view or download softcopy publications free of charge at www.ibm.com/shop/publications/order/.

Contents

Chapter 1. Introduction

When you plan to migrate COBOL source programs from non-IBM® compilers to IBM compilers, you can read the IBM extensions for language assistance. In addition, you can use the source conversion utility (`scu`) to help with the source program conversion.

This document presents topics to help you migrate COBOL source programs to IBM COBOL for AIX®, V3.1 or later.

COBOL programs created using a COBOL compiler other than the IBM COBOL for AIX compiler will normally have some source code differences that will need to be modified to become compatible with IBM COBOL for AIX. These differences are often the result of implementer extensions to Standard COBOL, but might also be due to differences in the operating system, or maybe for items in the COBOL standard that are specified as "implementor defined", or some combination of these. The number of differences will depend largely on how many extensions to Standard COBOL were coded in the COBOL source members that you are migrating.

In this document, the term Standard COBOL 2002 refers to ISO/IEC 1989:2002(E) Information technology — Programming languages — COBOL. This does not imply that IBM COBOL for AIX has implemented the new features found in Standard COBOL 2002 (although some have been implemented), we are simply using Standard COBOL 2002 to help you to identify extensions to Standard COBOL that you might encounter as you do your migration, differences between IBM COBOL for AIX and Standard COBOL, and to help you to determine the changes you might need to make to resolve these. It will be useful for you to refer to the IBM COBOL for AIX Language Reference and Programming Guide when modifying COBOL code. You can find these books at the IBM COBOL for AIX Library website: www.ibm.com/software/awdtools/cobol/aix/library/.

Chapter 2. Format changes

To run your source programs on IBM COBOL for AIX, make the required format changes to your source programs.

File suffix .cob for COBOL source

You can now compile a COBOL program by using a `.cob` suffix in addition to the `.cbl` suffix.

IBM COBOL for AIX supports the `.cob` suffix for COBOL source files.

See the following example:

```
$ cob2 -o tcCobol tcCobol.cob
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1, program TEST-CASE, no statements flagged.
```

Free-format COBOL source files

IBM COBOL for AIX suppresses some messages that are related to enforcing fixed-format source code.

When you use IBM COBOL for AIX to compile source code that contains free-format COBOL source, you will receive the error messages in the following example:

```
$ cat tcFreeFormat.cbl
     000100 IDENTIFICATION DIVISION.
     000200 PROGRAM-ID. FreeFormat.
     000300 ENVIRONMENT DIVISION.
     000400 DATA DIVISION.
     000500
     000600 PROCEDURE DIVISION.
     000700 BEGIN-MY-PROGRAM.
     000800 DISPLAY "COBOL ON AIX 5.1.0...".
     000900 STOP RUN.
$ cob2 tcFreeFormat.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID Message code Message text
    6  IGYPS0017-E  "PROCEDURE" should begin in area "A". It was
processed as if found in area "A".
    8  IGYPS0009-E  "DISPLAY" should not begin in area "A". It was
processed as if found in area "B".
Messages Total Informational Warning Error Severe Terminating
Printed:  2                            2
End of compilation 1, program FREEFORMAT, highest severity: Error.
Return code 8
```

You can rewrite your source in fixed format, or use `scu` to convert your source from free-format source to fixed-format source:

```
$ cat sol-tcFreeFormat.cbl
000001 IDENTIFICATION DIVISION.
000002 PROGRAM-ID. FreeFormat.
000003 ENVIRONMENT DIVISION.
000004 DATA DIVISION.
000005 PROCEDURE DIVISION.
000006
000007 BEGIN-MY-PROGRAM.
```

```
000008
000009      DISPLAY "COBOL ON AIX 5.1.0...".
000010 STOP-MY-PROGRAM.
000011      STOP RUN.
```

Alphanumeric literals longer than 160 characters in COBOL source files

The length of an alphanumeric literal, excluding the separators that delimit the literal, must be greater than zero and less than or equal to 160 alphanumeric character positions.

When you use IBM COBOL for AIX to compile source code that contains one or more alphanumeric literals longer than 160 characters, you will receive the error message in the following example:

```
$ cat tcAlphanumeric160.cbl
   000100 IDENTIFICATION DIVISION.
   000200 PROGRAM-ID. TEST-CASE.
   000300 ENVIRONMENT DIVISION.
   000400
   000500 DATA DIVISION.
   000600 WORKING-STORAGE SECTION.
   000700
   000800 01 VARIABLE PIC N(300) VALUE "BEGIN = = = = = = =
   000900-       " = = = = = = = = = = = = = = = = = = = = =
   001000-       " = = = = = = = = = = = = = = = = = = = = =
   001100-       " = = = = = = = = = = = = = = = = = = = = =
   001200-       "= = = = = = = = = = = = = = END".
   001300
   001400 PROCEDURE DIVISION.
   001500     DISPLAY VARIABLE.
   001600     STOP RUN.

$ cob2 -o tcAlphanumeric160 tcAlphanumeric160.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
  8     IGYDS0026-E   An alphanumeric literal that was longer than 160
                     characters was specified.  The literal was
                      truncated to 160 characters.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      1                                   1
End of compilation 1,  program TEST-CASE,  highest severity: Error.
Return code 8
```

According to Standard COBOL 2002, section 8.3.1.2.1.2, the length of an alphanumeric literal, excluding the separators that delimit the literal, must be greater than zero and less than or equal to 160 alphanumeric character positions.

Change your COBOL source to follow Standard COBOL 2002. You can change your code to split long literals. As shown in the following example, `MOVE "longliteral" to ITEM-1` can be split:

```
MOVE "long" to ITEM-1(1:4).
MOVE "literal" to ITEM-1(5:).
```

Extra and misplaced periods in COBOL source

You must remove extra or misplaced periods in the fixed-format source.

When you use IBM COBOL for AIX to compile source code that contains extra and misplaced periods, you will receive the error messages in the following example:

```
$ cat tcPeriods.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 PROCEDURE DIVISION.
     000070     PERFORM GREETING.
     000080    .
     000090
     000100     PERFORM END-PROGRAM.
     000110
     000120 GREETING
     000130 .
     000140
     000150     DISPLAY "HELLO FROM IBM.".
     000160 END-PROGRAM.
     000170     STOP RUN.

$ cob2 -o tcPeriods tcPeriods.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
    8   IGYPS0009-E   ". " should not begin in area "A".  It was processed
                      as if found in area "B". Same message on line:     13
    8   IGYPS0019-W   No COBOL statement was found between periods.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:        3                         1         2
End of compilation 1,  program TEST-CASE,  highest severity: Error.
Return code 8
```

According to Standard COBOL 2002, section 6.2, Fixed-form reference format, fixed-format source cannot include extra or missing periods in the source.

Change your COBOL source to follow Standard COBOL 2002, or use `scu` to remove extra and misplaced periods:

```
$ cat sol-tcPeriods.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 PROCEDURE DIVISION.
     000070     PERFORM GREETING.
     000080
     000090
     000100     PERFORM END-PROGRAM.
     000110
     000120 GREETING.
     000130
     000140
     000150     DISPLAY "HELLO FROM IBM.".
     000160 END-PROGRAM.
     000170     STOP RUN.
```

User-defined words

Within a source element, a user-defined word can be used as only one type of source element.

Example 1

In the following example, the procedure name is the same as the program identifier:

```
$ cat tcSameName.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. GREETING.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 PROCEDURE DIVISION.
     000070
     000080 MY-PARAGRAPH.
     000090     PERFORM GREETING.
     000100     PERFORM END-PARAGRAPH.
     000110
     000120 GREETING.
     000130     DISPLAY "WELCOME TO IBM COBOL FOR AIX 5.1.0...".
     000140
     000150 END-PARAGRAPH.
     000160     DISPLAY "HAVE A NICE DAY!".
     000170     STOP RUN.
$ cob2 -o tcSameName tcSameName.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     9  IGYPA3007-S   "GREETING" was not defined as a procedure-name.  The
                      statement was discarded.
    12  IGYPS2007-E   "GREETING" was defined as both a procedure-name and
                      another data type.  The procedure-name definition was
                      discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      2                                    1         1
End of compilation 1, program GREETING,  highest severity: Severe.
Return code 12
```

Example 2

In the following example, GREETING is used as both a data name and a procedure name:

```
$ cat tcSameName2.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070
     000080 01 GREETING PIC X(20) VALUE "FROM IBM.".
     000090 PROCEDURE DIVISION.
     000100 MY-PROGRAM.
     000110     PERFORM GREETING.
     000120     PERFORM END-PROGRAM.
     000130
     000140 GREETING.
     000150     DISPLAY "HELLO " GREETING.
     000160 END-PROGRAM.
     000170     STOP RUN.

$ cob2 -o tcSameName tcSameName.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     9  IGYPA3007-S   "GREETING" was not defined as a procedure-name.  The
                      statement was discarded.
    12  IGYPS2007-E   "GREETING" was defined as both a procedure-name and
                      another data type.  The procedure-name definition was
                      discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      2                                    1         1
End of compilation 1,  program GREETING,  highest severity: Severe.
Return code 12
```

According to Standard COBOL 2002, section 8.3.1.1, User-defined words, within a source element, a user-defined word might be used as only one type of user-defined word. Further rules for uniqueness are specified in section 8.4.1, Uniqueness of reference.

Change your COBOL source to follow Standard COBOL 2002.

For Example 1:

```
$ cat sol-tcSameName.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. GREETING.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 PROCEDURE DIVISION.
     000070
     000060 MY-PARAGRAPH.
     000080     PERFORM HELLO.
     000090     PERFORM END-PARAGRAPH.
     000100
     000110 HELLO.
     000120     DISPLAY "WELCOME TO IBM COBOL FOR AIX 5.1.0...".
     000130
     000140 END-PARAGRAPH.
     000150     DISPLAY "HAVE A NICE DAY!".
     000160     STOP RUN.
```

For Example 2:

```
$ cat sol-tcSameName2.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070
     000080 01 GREETING PIC X(20) VALUE "FROM IBM.".
     000090 PROCEDURE DIVISION.
     000100 MY-PROGRAM.
     000110     PERFORM MY-GREETING.
     000120     PERFORM END-PROGRAM.
     000130
     000140 MY-GREETING.
     000150     DISPLAY "HELLO " GREETING.
     000160 END-PROGRAM.
     000170     STOP RUN.
```

REPORT SECTION and SCREEN SECTION

IBM COBOL for AIX does not support `REPORT SECTION` or `SCREEN SECTION`.

When you use IBM COBOL for AIX to compile source code that contains `REPORT SECTION` or `SCREEN SECTION`, you will receive the error messages in the following example:

```
$ cat tcReportScreenSection.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 REPORT SECTION.
     000070 01 TYPE IS PAGE HEADING.
```

```
     000080 01 TYPE IS PAGE FOOTING.
     000090
     000100 SCREEN SECTION.

$ cob2 -o tcReportScreenSection tcReportScreenSection.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     6  IGYDS0148-S   "REPORT" is a reserved word related to language not
                      supported by this compiler.  The statement was discarded.
     6  IGYDS1089-S   "REPORT" was invalid.  Scanning was resumed at the
                      next area "A" item, level-number, or the start of the next
                      clause.
    10  IGYDS0009-E   "SCREEN" should not begin in area "A".  It was
                      processed as if found in area "B".
    10  IGYDS1089-S   "SCREEN" was invalid.  Scanning was resumed at the next
                      area "A" item, level-number, or the start of the next clause.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:     10                                    4         6
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

For details about REPORT SECTION and SCREEN SECTION, see Standard COBOL 2002, section 13.7, Report section, and section 13.8, Screen section.

Embedded null characters in variables

When you use IBM COBOL for AIX to compile source code that contains a variable with embedded null characters, null characters will be removed.

See the output of MY-DATA in the following example. The null characters are removed.

```
$ cat tcBlankReplaceNull.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. "TEST-CASE".
     000030 ENVIRONMENT DIVISION.
     000040 CONFIGURATION SECTION.
     000050 DATA DIVISION.
     000060 WORKING-STORAGE SECTION.
     000080   77 MY-DATA PIC X(5) VALUE X"4F4E00004E".
     000120 PROCEDURE DIVISION.
     000125     DISPLAY MY-DATA "<-".
     000130     STOP RUN.
 $ cob2 tcBlankReplaceNull.cbl
  PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1,  program TEST-CASE,  no statements flagged.
       $ a.out
ONN<-
```

Format of floating point literals

IBM COBOL for AIX uses this format for floating point literals: [+|-] <mantissa>E [+|-] <exponent>.

When you use IBM COBOL for AIX to compile source programs containing floating point literals that do not use an E-notation; for example, if 1500.0 is used instead of 1.5E3, you will receive the error messages in the following example:

```
$ cat tcFloatingFormat.cbl
     000100 IDENTIFICATION DIVISION.
     000200 PROGRAM-ID. TEST-CASE.
     000300 ENVIRONMENT DIVISION.
     000400 DATA DIVISION.
     000500
```

```
     000600 WORKING-STORAGE SECTION.
     000700 01 FLOAT1 COMP-1 VALUE 1357.9.
     000800 01 FLOAT2 COMP-1 VALUE 2468.0.
     000900
     001000 PROCEDURE DIVISION.
     001100     DISPLAY "FLOAT 1 IS " FLOAT1.
     001200     DISPLAY "FLOAT 2 IS " FLOAT2.
     001300     STOP RUN.

$ cob2 -o tcFloatingFormat tcFloatingFormat.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     7  IGYGR1080-S   A "VALUE" clause literal was not compatible with the
                      data category of the subject data item.  The "VALUE"
                      clause was discarded.
Same message on line:      8
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      2                                              2
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

You can also use scu to convert such format of floating point literals.

RECORD SEQUENTIAL file organization

IBM COBOL for AIX does not support RECORD SEQUENTIAL file organization.

If your source contains the RECORD SEQUENTIAL file organization, you will receive the error messages in the following example.

Change RECORD SEQUENTIAL to LINE SEQUENTIAL, and then compile again. You can also use scu to replace RECORD SEQUENTIAL with LINE SEQUENTIAL.

```
$cat tcRecordSequential.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000031
     000032 INPUT-OUTPUT SECTION.
     000033 FILE-CONTROL.
     000034     SELECT CONTACTS
     000035        ASSIGN TO "MYFILE.DAT"
     000036        ORGANIZATION IS RECORD SEQUENTIAL.
     000037
     000040 DATA DIVISION.
     000050 FILE SECTION.
     000051 FD CONTACTS
     000052     RECORD CONTAINS 20 CHARACTERS.
     000053 01 CONTACT-RECORD.
     000054    05 FNAME  PIC X(10).
     000055    05 LNAME  PIC X(10).
     000073
     000090 PROCEDURE DIVISION.
     000100     PERFORM OPEN-FILE.
     000110     PERFORM INPUT-DATA.
     000120     PERFORM SAVE-DATA.
     000130     PERFORM CLOSE-FILE.
     000140     PERFORM END-PROGRAM.
     000150
     000160 INPUT-DATA.
     000170     DISPLAY "ENTER FIRST NAME.".
     000180     ACCEPT FNAME.
     000190     DISPLAY "ENTER LAST NAME.".
     000200     ACCEPT LNAME.
     000210
```

```
     000220 OPEN-FILE.
     000230     OPEN OUTPUT CONTACTS.
     000240
     000250 SAVE-DATA.
     000260     WRITE CONTACT-RECORD.
     000270
     000280 CLOSE-FILE.
     000290     CLOSE CONTACTS.
     000300
     000310 END-PROGRAM.
     000320     STOP RUN.
cob2 -o tcRecordSequential tcRecordSequential.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     9  IGYDS0093-S   "RECORD" was found in the "ORGANIZATION" clause.
                      The clause was discarded.
     9  IGYDS1335-S   Expected a data-name in a "RECORD KEY" clause, but
                      found "SEQUENTIAL".  The clause was discarded.
    13  IGYGR1216-I   A "RECORDING MODE" of "F" was assumed for file
                      "CONTACTS".
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:        3           1                                2
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

Chapter 3. IDENTIFICATION DIVISION changes

IDENTIFICATION DIVISION must be the first division in each COBOL source program, factory definition, object definition, and method definition.

When you use IBM COBOL for AIX to compile source code that contains an IDENTIFICATION DIVISION not as the first division in a COBOL source program, you will receive the error messages in the following example:

```
$ cat tcIdentificationDivision.cbl
     000010 ENVIRONMENT DIVISION.
     000020 DATA DIVISION.
     000030
     000040 WORKING-STORAGE SECTION.
     000050 01 GREETING GLOBAL PIC X(50) VALUE "HELLO FROM IBM...".
     000060 01 GOODBYE GLOBAL PIC X(50) VALUE "HAVE A NICE DAY...".
     000070
     000080 IDENTIFICATION DIVISION.
     000090 PROGRAM-ID. TEST-CASE.
     000100
     000110 PROCEDURE DIVISION.
     000120 DISPLAY GREETING.
     000130 DISPLAY GOODBYE.
     000140 STOP RUN.$

cob2 -o tcIdentificationDivision tcIdentificationDivision.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID Message code Message text
     1 IGYDS1101-S  Expected "IDENTIFICATION DIVISION" header, but found
       "ENVIRONMENT". "IDENTIFICATION DIVISION" header was assumed
       before "ENVIRONMENT".
     8 IGYDS1003-E  A "PROGRAM-ID" paragraph was not found. Program-name
                    "CBLNAM01" was assumed.
    39 IGYSC0136-E  End of source file was encountered or an "END
                    PROGRAM" marker for a containing program was found,
                    but an "END PROGRAM" marker for program "TEST-CASE"
                    was not found. "END PROGRAM TEST-CASE" was assumed.
    39 IGYSC0136-E  End of source file was encountered or an "END
                    PROGRAM" marker for a containing program was found,
                    but an "END PROGRAM" marker for program "CBLNAM01"
                    was not found. "END PROGRAM CBLNAM01" was assumed.
Messages Total Informational Warning Error Severe
Terminating
Printed:  4                             3      1
End of compilation 1, program CBLNAM01, highest severity: Severe.
Return code 12
```

According to Standard COBOL 2002, section E4.1.1, Source level organization, a source unit is a set of COBOL statements as specified in this document and consists of an IDENTIFICATION DIVISION followed optionally by any of the ENVIRONMENT DIVISION, DATA DIVISION, or PROCEDURE DIVISION, or a combination of those divisions.

Change your COBOL source to follow Standard COBOL 2002:

```
$ cat sol-tcIdentificationDivision.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
```

```
000060 WORKING-STORAGE SECTION.
000070 01 GREETING GLOBAL PIC X(50) VALUE "HELLO FROM IBM...".
000080 01 GOODBYE GLOBAL PIC X(50) VALUE "HAVE A NICE DAY...".
000090
000100 PROCEDURE DIVISION.
000110     DISPLAY GREETING.
000120     DISPLAY GOODBYE.
000130     STOP RUN.
```

Chapter 4. Hyphens in an ASSIGN statement

If a hyphen exists in the environment variable or data name specified in an ASSIGN statement, the value of the environment variable or the data name is processed as described in this topic.

When you use IBM COBOL for AIX to compile source code that contains a hyphen in the ASSIGN statement, you will receive the error message in the following example:

```
$ cat tcHyphenAssign.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. "TEST-CASE".
     000030 ENVIRONMENT DIVISION.
     000040 CONFIGURATION SECTION.
     000050 INPUT-OUTPUT SECTION.
     000060 FILE-CONTROL.
     000070     SELECT DATA-INFO
              ASSIGN TO "SYS-FILENAME"
              ORGANIZATION IS LINE SEQUENTIAL.
     000080 DATA DIVISION.
     000090 FILE SECTION.
     000100 FD DATA-INFO.
     000110 01 INFO-1     PIC N(132).
     000120 PROCEDURE DIVISION.
     000130     STOP RUN.
$ cob2 -o tcHyphenAssign tcHyphenAssign.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
   12  IGYGR1387-S   File system id SYS was specified for a file with
                     "ORGANIZATION LINE SEQUENTIAL".  The file system id was changed
                     to NAT.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      1                                             1
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12$
```

As described in the *COBOL for AIX Language Reference*, the format of the ASSIGN clause is '*<file-system-ID>-<system-file name>*'. If a hyphen exists in the environment variable or the data name value, the first 3 characters to the left of the leftmost hyphen are treated as the file system identifier. The character string to the right of the leftmost hyphen is then used as the system file name, possibly including drive and path names.

In the previous example, the compiler takes the first 3 characters to the left of the leftmost hyphen; in this case, SYS is treated as the file system identifier. The character string to the right of the leftmost hyphen is then used as the system file name, in this case, FILENAME, is used as the system file name.

If no hyphen exists, or the character string to the left of the leftmost hyphen has fewer than 3 characters, the entire character string is used as the system file name, possibly including drive and path names.

Specifying S-FILENAME in the ASSIGN clause works instead because the number of characters before the hyphen is fewer than 3 characters. Alternatively, you can specify SYSFILENAME in the ASSIGN clause.

```
$ cat sol-tcHyphenAssign.cbl
       000010 IDENTIFICATION DIVISION.
       000020 PROGRAM-ID. "TEST-CASE".
       000030 ENVIRONMENT DIVISION.
       000040 CONFIGURATION SECTION.
       000050 INPUT-OUTPUT SECTION.
       000060 FILE-CONTROL.
       000070     SELECT DATA-INFO
              ASSIGN TO "SYSFILENAME"
              ORGANIZATION IS LINE SEQUENTIAL.
       000080 DATA DIVISION.
       000090 FILE SECTION.
       000100 FD DATA-INFO.
       000110 01 INFO-1     PIC N(132).
       000120 PROCEDURE DIVISION.
       000130     STOP RUN.
```

IBM COBOL for AIX also handles `LINE SEQUENTIAL` by using the native (NAT) file system identifier. The same behavior results if you specified `STL-FILENAME`.

Chapter 5. DATA DIVISION changes

To run your source programs on IBM COBOL for AIX, make the required DATA DIVISION changes to your source programs.

OCCURS clauses in level 01

According to Standard COBOL 2002, you must follow the syntax rules for the OCCURS clause.

When you use IBM COBOL for AIX to compile source code that contains an OCCURS clause in level 01, you will receive the error messages in the following example:

```
$ cat tcOccursClause.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000061
     000062
     000063 01 EMPLOYEES OCCURS 100 TIMES.
     000064    05 FNAME PIC X(10).
     000065    05 LNAME PIC X(10).
     000066    05 SALARY PIC 9(5).
     000067000068 77 COUNTER PIC 9.
     000070
     000071 PROCEDURE DIVISION.
     000072     MOVE "DAVID" TO FNAME(1).
     000073     MOVE "SMITH" TO LNAME(1).
     000074     MOVE "60000" TO SALARY(1).
     000075     MOVE "JENNY" TO FNAME(2).
     000076     MOVE "HU"    TO LNAME(2).
     000077     MOVE "55000" TO SALARY(2).
     000078
     000080 DISPLAY-RESULT.
     000081      PERFORM VARYING COUNTER FROM 1 BY 1 UNTIL COUNTER > 2
     000082       DISPLAY "EMPLOYEE: " FNAME IN EMPLOYEES(COUNTER)" "
     000083               LNAME IN EMPLOYEES(COUNTER)" "
     000084              "Salary: " SALARY IN EMPLOYEES(COUNTER)
     000085      END-PERFORM.
     000086 STOP-PROGRAM.
     000090      STOP RUN.

$ cob2 -o tcOccursClause tcOccursClause.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     9  IGYDS1063-E   An "OCCURS" clause was found in the definition of a
                      level-1 item.  The "OCCURS" clause was discarded.
    17  IGYPS2120-S   Expected a reference-modification specification but
                      found ")".  The "MOVE" statement was discarded. Same
                      message on line:     18     19     20     21     22
    26  IGYPS2120-S   Expected a reference-modification specification but
                      found ")".  The "DISPLAY" statement was discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      8                                    1         7
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

According to Standard COBOL 2002, section 13.16.36.2, Syntax rules for the OCCURS clause, the OCCURS clause cannot be specified in a data description entry that has any of the following items:

- A level-number of 01, 66, 77, or 88
- A variable-occurrence data item subordinate to it

Change your COBOL source to follow Standard COBOL 2002. For example, if you change the code in the previous example by using the definition of EMPLOYEES that is moved from level 01 to level 03, it compiles without messages.

```
$ cat sol-tcOccursClause.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000061
     000062 01.
     000063 03 EMPLOYEES OCCURS 100 TIMES.
     000065    05 FNAME PIC X(10).
     000066    05 LNAME PIC X(10).
     000067    05 SALARY PIC 9(5).
     000068
     000069 77 COUNTER PIC 9.000070
     000071 PROCEDURE DIVISION.
     000072     MOVE "DAVID" TO FNAME(1).
     000073     MOVE "SMITH" TO LNAME(1).
     000074     MOVE "60000" TO SALARY(1).
     000075     MOVE "JENNY" TO FNAME(2).
     000076     MOVE "HU"    TO LNAME(2).
     000077     MOVE "55000" TO SALARY(2).
     000078
     000080 DISPLAY-RESULT.
     000082      PERFORM VARYING COUNTER FROM 1 BY 1 UNTIL COUNTER > 2
     000084       DISPLAY "EMPLOYEE: " FNAME IN EMPLOYEE(COUNTER) " "
     000085               LNAME IN EMPLOYEE(COUNTER) " "
     000085              "Salary: " SALARY IN EMPLOYEE(COUNTER)
     000086      END-PERFORM.
     000087 STOP-PROGRAM.
     000120      STOP RUN.
```

COMP-X

IBM COBOL for AIX does not support COMP-X.

When you use IBM COBOL for AIX to compile source code that contains COMP-X, you will receive the error message in the following example:

```
$ cat tcCompX.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070 01 STU-ID PIC S9(4) COMP-X VALUE ZERO.

$ cob2 -o tcCompX tcCompX.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     7  IGYDS1089-S   "COMP-X" was invalid.  Scanning was resumed at the
                      next area "A" item, level-number, or the start of
                      the next clause.
```

```
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      1                                              1
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

Change COMP-X to COMP-5, and make corresponding changes to the PICTURE clause. You must keep the same allocated storage for the data item.

The following table shows the COMP-5 storage allocation:

Table 1. COMP-5 storage allocation

Picture	Storage representation
S9(1) through S9(4)	Binary halfword (2 bytes)
S9(5) through S9(9)	Binary fullword (4 bytes)
S9(10) through S9(18)	Binary doubleword (8 bytes)
9(1) through 9(4)	Binary halfword (2 bytes)
9(5) through 9(9)	Binary fullword (4 bytes)
9(10) through 9(18)	Binary doubleword (8 bytes)

Runtime errors because of uninitialized variables being converted to a display type

You will receive runtime error messages if you move variable from one area of storage to another area where the variable being moved has not been initialized or it does not contain valid data.

When you use IBM COBOL for AIX to compile source code that contains an uninitialized variable, you will receive the error messages in the following example:

```
$cat tcConvert2DisplayType.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. "TEST-CASE".
     000030 ENVIRONMENT DIVISION.
     000040 CONFIGURATION SECTION.
     000050 DATA DIVISION.
     000060 WORKING-STORAGE SECTION.
     000070 01 EMP-INFO.
     000080    05  EMP-ID PIC S9(5) sign is trailing separate.
     000090 01 EMP-ID2 PIC S9(4).
     000100 PROCEDURE DIVISION.
     000120     MOVE EMP-ID OF EMP-INFO TO EMP-ID2.
     000130     STOP RUN.
$ cob2 -o tcConvert2DisplayType tcConvert2DisplayType.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1,  program TEST-CASE,  no statements flagged.
$ tcConvert2DisplayType
IWZ040S  An invalid separate sign was detected.
         Message routine called from offset 0x48 of routine
         iwzWriteERRmsg. iwzWriteERRmsg called from offset
         0xb4 of routine _iwzcBCD_CONV_ZndTS_To_ZndTO.
         _iwzcBCD_CONV_ZndTS_To_ZndTO called from offset
         0x130 of routine TEST-CASE.
IWZ901S  Program exits due to severe or critical error.
IOT/Abort trap(coredump)
```

Initialize EMP-ID of EMP-INFO to 0:

```
$ cat sol-tcConvert2DisplayType.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. "TEST-CASE".
     000030 ENVIRONMENT DIVISION.
     000040 CONFIGURATION SECTION.
     000050 DATA DIVISION.
     000060 WORKING-STORAGE SECTION.
     000070 01 EMP-INFO.
     000080    05  EMP-ID PIC S9(5) SIGN IS TRAILING SEPARATE
               VALUE ZERO.
     000090 01 EMP-ID2 PIC S9(4).
     000100 PROCEDURE DIVISION.
     000120     MOVE EMP-ID OF EMP-INFO TO EMP-ID2.
     000130     STOP RUN.
```

In addition to initializing EMP-ID to 0, you can compile by using -qwsclear, which clears a program's WORKING-STORAGE to binary zeros when the program is initialized. The storage is cleared before any VALUE clauses are applied. However, this approach is not preferred because of performance considerations.

Redefined data items and OCCURS clauses

IBM COBOL for AIX does not support a redefined data item or an OCCURS clause for the same data item.

When you use IBM COBOL for AIX to compile source code that contains a redefined object and the OCCURS clause, you will receive the error messages in the following example:

```
$ cat tcRedefineOccurs.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040
     000050 DATA DIVISION.
     000060 WORKING-STORAGE SECTION.
     000070
     000080 01 EMPLOYEE-INFO.
     000090    03 EMPLOYEE OCCURS 50 TIMES.
     000100       05 NAME  PIC X(10).
     000110       05 AGE   PIC 9(3).
     000120    03 NEW-EMPLOYEE REDEFINES EMPLOYEE.
     000130       05 FNAME PIC X(10).
     000130       05 LNAME PIC X(20).
     000140       05 AGE   PIC 9(3).
     000150
     000160 PROCEDURE DIVISION.
     000170     STOP RUN.

$ cob2 -o tcRedefineOccurs tcRedefineOccurs.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
    12  IGYDS1152-S   A definition of an object of a "REDEFINES" clause
                      contained an "OCCURS" clause.  The "REDEFINES" clause
                      was discarded.
    13  IGYDS1073-I   "NEW-EMPLOYEE" redefined a larger item.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      2                                     2
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

According to Standard COBOL 2002, section 13.16.42.2, Syntax rules, the data item being redefined cannot contain an OCCURS clause.

Change your COBOL source to followStandard COBOL 2002. For example, if you change the code in the previous example by adding a group item and renumbering, it compiles without messages.

```
$ cat tcRedefineOccurs.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040
     000050 DATA DIVISION.
     000060 WORKING-STORAGE SECTION.
     000070
     000080 01 EMPLOYEE-INFO.
     000081 02 EMPLOYEE-GROUP-1.
     000090 03 EMPLOYEE OCCURS 50 TIMES.
     000100 05 NAME PIC X(30).
     000110 05 AGE PIC 9(3).
     000115 02 EMPLOYEE-GROUP-2 REDEFINES EMPLOYEE-GROUP-1.
     000116 03 EMPLOYEE OCCURS 50 TIMES.
     000117 05 FNAME PIC X(10).
     000118 05 LNAME PIC X(20).
     000119 05 AGE PIC 9(3).
     000150
     000160 PROCEDURE DIVISION.
     000170 STOP RUN.
```

Modifying boolean PICTURE character strings and boolean literals

When you use IBM COBOL for AIX to compile source code that contains a boolean `PICTURE` character string and a boolean literal, you will receive error messages.

See the following example:

```
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. TEST-CASE.
000030 ENVIRONMENT DIVISION.
000040 DATA DIVISION.
000050
000060 WORKING-STORAGE SECTION.
000061
000062 01 IND-AREA.
000063    05 MESSAGES OCCURS 99 TIMES PIC 1 INDICATOR1.
000064       88 MESSAGE-OFF            VALUE B'0'.
000065       88 MESSAGE-ON             VALUE B'1'.
000066 01 MESSAGE-SET.
000067    05 MESSAGE-1 PIC 9(2) VALUE 01.
000070
000071 PROCEDURE DIVISION.
000072      SET MESSAGE-OFF(MESSAGE-1) TO TRUE.
000074
000075      IF MESSAGE-ON(MESSAGE-1)
000076         DISPLAY "IT'S TRUE"
000077      ELSE
000078         DISPLAY "IT'S FALSE"
000079      END-IF.
000080 STOP-PROGRAM.
000081      STOP RUN.

> cob2 tcBooleanLiteral.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     9  IGYDS1149-S   "1" was not a valid combination of "PICTURE" string
                      characters.  A "PICTURE" string of "S9(1)" was assumed.
     9  IGYDS1089-S   "INDICATOR1" was invalid.  Scanning was resumed at
                      the next area "A" item, level-number, or the start of
                      the next clause.
```

```
    10  IGYDS0001-W   A blank was missing before character "'" in column 47.
                      A blank was assumed. Same message on line:     11
    10  IGYGR0145-S   "B" was not defined as a "SYMBOLIC CHARACTERS" figurative
                      constant.  "B" was discarded. Same message on line:     11
    10  IGYGR1239-S   Level-88 "VALUE" literal "B" was not compatible with
                      the data category of the conditional variable.  The
                      literal was discarded. Same message on line:     11
    10  IGYGR1239-S   Level-88 "VALUE" literal "'0'" was not compatible
                      with the data category of the conditional variable.
                      The literal was discarded.
    11  IGYGR1239-S   Level-88 "VALUE" literal "'1'" was not compatible with
                      the data category of the conditional variable.  The
                      literal was discarded.
    16  IGYPS2061-S   An error was found in the definition of condition-name
                      "MESSAGE-OFF".  The reference to this condition-name was
                      discarded.
    16  IGYPS2120-S   Expected a reference-modification specification but
                      found ")".  The "SET" statement was discarded.
    18  IGYPS2061-S   An error was found in the definition of condition-name
                      "MESSAGE-ON".  The reference to this condition-name was
                      discarded.
    18  IGYPS2120-S   Expected a reference-modification specification but
                      found ")".  The "IF" statement was discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:     14                          2                12
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

Change the PICTURE character-string symbol to X and use hexadecimal notation for alphanumeric literals for the values of X.

```
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. TEST-CASE.
000030 ENVIRONMENT DIVISION.
000040 DATA DIVISION.
000050
000060 WORKING-STORAGE SECTION.
000061
000062 01 IND-AREA.
000063    05 MESSAGES OCCURS 99 TIMES PIC X.
000064       88 MESSAGE-OFF            VALUE X'00'.
000065       88 MESSAGE-ON             VALUE X'01' THROUGH X'FF'.
000066 01 MESSAGE-SET.
000067    05 MESSAGE-1 PIC 9(2) VALUE 01.
000070
000071 PROCEDURE DIVISION.
000072      SET MESSAGE-OFF(MESSAGE-1) TO TRUE.
000074
000075      IF MESSAGE-ON(MESSAGE-1)
000076         DISPLAY "IT'S TRUE"
000077      ELSE
000078         DISPLAY "IT'S FALSE"
000079      END-IF.
000080 STOP-PROGRAM.
000081      STOP RUN.

> cob2 sol-tcBooleanLiteral.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1,  program TEST-CASE,  no statements flagged.
> a.out
IT'S FALSE
```

Chapter 6. PROCEDURE DIVISION changes

To run your source programs on IBM COBOL for AIX, make the required PROCEDURE DIVISION changes to your source programs.

Moving national data items to alphanumeric data items

You cannot move a national data item to an alphanumeric data item.

When you use IBM COBOL for AIX to compile source code that contains a MOVE statement from a national data item to an alphanumeric data item, you will receive the error message in the following example:

```
$ cat tcNational2Alphanumeric.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. ND2AD.
     000030 ENVIRONMENT DIVISION.
     000040 CONFIGURATION SECTION.
     000050
     000060 DATA DIVISION.
     000070 WORKING-STORAGE SECTION.
     000080
     000090 01 DataNational      PIC N(100).
     000100
     000110 01 DataAlphanumeric    PIC X(100).
     000120
     000130 PROCEDURE DIVISION.
     000140       MOVE DataNational TO DataAlphanumeric.
     000150 STOP-MY-PROGRAM.
     000160       STOP RUN.


$ cob2 -o tcNational2Alphanumeric tcNational2Alphanumeric.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
    14  IGYPA3005-S   "DATANATIONAL (NATIONAL ITEM)" and
                      "DATAALPHANUMERIC (ALPHANUMERIC)" did
                      not follow the "MOVE" statement compatibility
                      rules.  The statement was discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      1                                              1
End of compilation 1,  program ND2AD,  highest severity: Severe.
Return code 12
```

According to Standard COBOL 2002, section 14.8.24.2, Syntax rules for the MOVE statement, you cannot move a national data item to an alphanumeric data item.

Change your COBOL source to follow Standard COBOL 2002.

Undefined symbol errors for C functions being called

Use the PGMNAME compiler option to control how the compiler handles program-names.

When you use IBM COBOL for AIX to compile source code that contains a call to a C function in mixed or lowercase characters, you will receive the error messages in the following example:

```
$ cat cTest.c
     void CFunction(){
     printf("HELLO FROM C...\n");
     }
$ cat tcCallCFunction.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 CONFIGURATION SECTION.
     000050 DATA DIVISION.
     000060 WORKING-STORAGE SECTION.
     000070
     000080 PROCEDURE DIVISION.
     000090     CALL "CFunction"
     000100     DISPLAY "HELLO FROM COBOL...".
     000110     STOP RUN.
$ xlc -c cTest.c
$ cob2 -o tcCallCFunction tcCallCFunction.cbl cTest.o
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1,  program TEST-CASE,  no statements flagged.
ld: 0711-317 ERROR: Undefined symbol: .CFUNCTION
ld: 0711-345 Use the -bloadmap or -bnoquiet option to obtain more
             information.
```

IBM COBOL for AIX folds program-names to uppercase. If you have COBOL source that contains a call to a C function in mixed or lowercase characters, this function is folded to uppercase characters. The linker will not find the program, and an error message is displayed to indicate an unresolved symbol.

You can use the PGMNAME compiler option to control how the compiler handles program-names. The default is PGMNAME(UPPER), but you can use PGMNAME(MIXED) to process the program-name as is, without truncation, translation, or folding to uppercase. When you specify PGMNAME(MIXED), use the literal format of the program-name; that is, make the program-name a literal string such as "programname", or you will see the following message:

```
IGYDS1046-E   A user-defined word was found as a "PROGRAM-ID" name under
the "PGMNAME(LONGMIXED)" compiler option.
```

For an example about a COBOL program that calls C functions, see *Example: COBOL program calling C functions* in the COBOL for AIX Programming Guide.

```
$ cat sol-tcCallCFunction.cbl
     000010 CBL PGMNAME(LONGMIXED)
     000020 IDENTIFICATION DIVISION.
     000030 PROGRAM-ID. "TEST-CASE".
     000040 ENVIRONMENT DIVISION.
     000050 CONFIGURATION SECTION.
     000060 DATA DIVISION.
     000070 WORKING-STORAGE SECTION.
     000080
     000090 PROCEDURE DIVISION.
     000100     CALL "CFunction"
     000110     DISPLAY "HELLO FROM COBOL...".
     000120     STOP RUN.
$ cob2 -o sol-tcCallCFunction sol-tcCallCFunction.cbl cTest.o
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1,  program TEST-CASE,  no statements flagged.
$ sol-tcCallCFunction
HELLO FROM C...
HELLO FROM COBOL...
```

COBOL programs containing an ACCEPT statement

In IBM COBOL for AIX, the ACCEPT statement assigns an input line to the data item.

- If the input line is shorter than the data item, the data item is padded with spaces of the appropriate representation.
- If the input line is longer than the data item:
  - When a program reads from a screen, the remaining characters are discarded.
  - When a program reads from a file, the remaining characters are retained as the next input line for the file.

If the input data is longer than the receiving area, then IBM COBOL for AIX pads the area with spaces of the appropriate representation for the receiving area.

According to Standard COBOL 2002, section 14.8.1.3, General rules for the ACCEPT statement, the implementer must define the size of a data transfer for each hardware device.

BINARY formatted data in an ACCEPT statement

According to Standard COBOL 2002, section 14.8.1.3, General, the ACCEPT statement causes the transfer of data from the hardware device. This data replaces the content of the data item referenced by *identifier-1*. Any conversion of data that is required between the hardware device and the data item referenced by *identifier-1* is defined by the implementer. The IBM COBOL for AIX implementation allows only displayable data (display, national, numeric, alphanumeric) which precludes binary formatted data.

When you use IBM COBOL for AIX to compile source code that contains BINARY formatted data in an ACCEPT statement, you will receive the error message in the following example:

```
$ cat tcAcceptBinary.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070 01 EMPLOYEE-ID PIC S9(4) COMP-5 VALUE ZERO.
     000080
     000090 PROCEDURE DIVISION.
     000100 MY-PROGRAM.
     000110     DISPLAY "PLEASE INSERT EMPLOYEE'S ID NUMBER..".
     000120     ACCEPT EMPLOYEE-ID.
     000130
     000140     STOP RUN.

$ cob2 tcAcceptBinary tcAcceptBinary.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
  12  IGYPA3018-S   Identifier "EMPLOYEE-ID (BINARY INTEGER)" was used
in an "ACCEPT" statement.  The statement was discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      1                                             1
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

Change the data type from COMP-5 to DISPLAY, and make further changes to the input source as well.

```
$ cat sol-tcAcceptBinary.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070 01 EMP-ID PIC 9(4) VALUE ZEROES.
     000080 01 EMPLOYEE-ID PIC S9(4) COMP-5 VALUE ZERO.
     000090
     000100 PROCEDURE DIVISION.
     000110 MY-PROGRAM.
     000120     DISPLAY "PLEASE INSERT EMPLOYEE'S ID NUMBER..".
     000130     ACCEPT EMP-ID.
     000140     MOVE EMP-ID TO EMPLOYEE-ID.
     000150     STOP RUN.
```

NULL parameters in CALL statements

IBM COBOL for AIX does not support NULL parameters in a CALL statement.

When you use IBM COBOL for AIX to compile source code that contains NULL parameters in a CALL statement, you will receive the error message in the following example:

```
$ cat tcCallNull.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070
     000080 PROCEDURE DIVISION.
     000090     CALL "TEST-CASE2" USING NULL.
     000100     STOP RUN.

$ cob2 -o tcCallNull tcCallNull.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
LineID  Message code  Message text
     9  IGYPS2106-S   "NULL" was found in the "CALL" statement.  It was not
                      allowed in this context.  The statement was discarded.
Messages    Total    Informational    Warning    Error    Severe    Terminating
Printed:      1                                             1
End of compilation 1,  program TEST-CASE,  highest severity: Severe.
Return code 12
```

According to Standard COBOL 2002, section 14.4.8.2, syntax rules do not identify NULL as a valid parameter for the CALL statement.

```
$ cat sol-tcCallNull.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040 DATA DIVISION.
     000050
     000060 WORKING-STORAGE SECTION.
     000070 01 NULLPTR USAGE POINTER VALUE NULL.
     000080
     000110 PROCEDURE DIVISION.
     000140     CALL "TEST-CASE2" USING VARIABLE.
     000150     STOP RUN.
```

Chapter 7. Output for positive numbers

IBM COBOL for AIX output does not contain a plus sign (+) for positive numbers.

For example, when you use IBM COBOL for AIX to compile source code that contains a positive number, the output does not display the plus sign (+):

```
$ cat tcPlusSign.cbl
     000010 IDENTIFICATION DIVISION.
     000020 PROGRAM-ID. TEST-CASE.
     000030 ENVIRONMENT DIVISION.
     000040
     000050 INPUT-OUTPUT SECTION.
     000060
     000070 DATA DIVISION.
     000080 WORKING-STORAGE SECTION.
     000090 01 POSITIVE-NUM PIC S9(4) COMP-5 VALUE ZERO.
     000100 01 NEGATIVE-NUM PIC S9(4) COMP-5 VALUE ZERO.
     000110
     000120 PROCEDURE DIVISION.
     000130     COMPUTE POSITIVE-NUM = 10 - 3.
     000140     COMPUTE NEGATIVE-NUM = 3 - 10.
     000150     DISPLAY "10 - 3 = " POSITIVE-NUM.
     000160     DISPLAY "3 - 10 = " NEGATIVE-NUM.
     000170     STOP RUN.

$ cob2 -o cat tcPlusSign tcPlusSign.cbl
PP 5724-Z87 IBM COBOL for AIX 5.1.0 in progress ...
End of compilation 1,  program TEST-CASE,  no statements flagged.

$ tcPlusSign
10 - 3 = 00007
3 - 10 = -00007
```

Note: This rule does not apply to numeric-edited items. If you use an editing sign control symbol in a variable declared with USAGE DISPLAY or NATIONAL, such as the plus sign (+) in the following example:

```
000090 01 POSITIVE-NUM PIC +9(4).
000100 01 NEGATIVE-NUM PIC +9(4).
```

The output is as follows:

```
10 - 3 = +0007
3 - 10 = -0007
```

Notices

Programming interfaces: Intended programming interfaces allow the customer to write programs to obtain the services of IBM COBOL for AIX.

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2010, 2015.

PRIVACY POLICY CONSIDERATIONS:

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, or to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Printed in USA