

System/36

COBOL Messages



When You Are:

Planning to
Install Your
Computer

Getting Your
Computer
Ready to Use

Operating
Your
Computer

Operating and
Using the
Utilities

**Programming
Your
Computer**

Communicating
with Another
Computer or
Remote Device

**Determining
the Cause
of a Problem**

You Can Find Information In:

What to Do Before Your Computer Arrives
or
Converting from System/34 to System/36

Setting Up Your Computer
Performing the First System Configuration For Your System
System Security Guide

Learning About Your Computer
Operating Your Computer

Source Entry Utility Guide
Data File Utility Guide
Creating Displays
Work Station Utility Guide
Utilities Messages

Concepts and Programmer's Guide
System Reference
Sort Guide
Work Station Utility Guide
Programming with COBOL
COBOL Messages

(communication manuals)
(communication message manuals)

System Messages
COBOL Messages
System Problem Determination

IBM System/36

COBOL Messages

Program Number 5727-CB1
5727-CB6

File Number
S36-24

Order Number
SC21-7941-3

Fourth Edition (January 1986)

This major revision obsoletes SC21-7941-2. All messages have been updated. Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions.

This edition applies to Release 4, Modification Level 0, of IBM System/36 COBOL Program Product (Program 5727-CB1 and 5727-CB6) and to all subsequent releases and modifications until otherwise indicated in new editions.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Canada Ltd., Information Development, Department 849, 895 Don Mills Road, Don Mills, Ontario, Canada, M3C 1W3. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Contents

About This Manual	v
Who should use this manual . . .	v
How this manual is arranged . . .	v
What you should know . . .	v
If you need more information . . .	v
Chapter 1. Introduction	1-1
Format of Compiler Diagnostic Messages	1-1
Diagnostic Levels	1-3
Format of Displayed Messages	1-3
Severity Levels	1-5
Automatic Response Levels	1-5
Before Calling for Service	1-6
Chapter 2. Compiler Diagnostic Messages	2-1
Chapter 3. Displayed Messages	3-1



About This Manual

Who should use this manual . . .

This manual lists and describes all COBOL messages that are printed or displayed. This manual is intended to help the display station operator, the system operator, or the programmer respond to these messages or decide whom to contact for further help.

How this manual is arranged . . .

This manual has three parts:

- Introduction, which describes the format of the compiler diagnostic messages and the displayed messages
- Compiler Diagnostic Messages, which describes all COBOL compiler messages
- Displayed Messages, which describes all displayed COBOL messages

What you should know . . .

Before you use this manual, you should be familiar with COBOL programming. That information is contained in the IBM System/36 manual, *Programming with COBOL*, SC21-9007.

If you need more information . . .

System Messages, SC21-7938, describes the System/36 printed and displayed system messages.

Utilities Messages, SC21-7939, describes the messages for the System/36 utilities.

System Problem Determination, SC21-7919, explains how to determine whether a problem is in an IBM product or in your program. This manual also contains the *Problem Summary Form*.



Chapter 1. Introduction

The COBOL messages are listed by the message identification code in ascending order. The message identification code consists of the characters CBL followed by a hyphen and a four-digit number (for example, CBL-4020).

Format of Compiler Diagnostic Messages

Compiler-generated messages indicate conditions encountered during program compilation. This compiler listing of messages describes the invalid use of COBOL syntax or the violation of system requirements. Object-time messages and displayed compiler messages appear in the *Displayed Messages* section of this manual.

Error messages are described using the following format:

Message Identification Code	Diagnostic Level	Message
------------------------------------	-------------------------	----------------

This is what the message actually looks like.

Explanation

What the error is and what causes it.

Additional Information

What the compiler did as a result.

User Response

What the programmer can do to correct it.

In the following example, an error message is documented using this format:

Message Identification Code	Diagnostic Level	Message
CBL-0017	C	NAME[name]BEGINS OR ENDS WITH HYPHEN--ACCEPTED

Explanation

A COBOL name can have embedded hyphens, but must not begin or end with a hyphen.

Additional Information

The name and references to it are accepted.

User Response

Replace the name and all references to the name with a name that does not begin or end with a hyphen.

Two general types of messages are supplied with source diagnostics produced at compile time.

1. In the first type, an assumption is made or a default option is taken. By referencing the statement number supplied with the message, the programmer can read the explanation, and the meaning of the message as it applies to the condition should be clear.
2. In the second type, no assumptions are made and no default options are taken. Additional text describing the condition is given at the time the message is produced on the printer.

Variable content of diagnostic messages is always included within a pair of square brackets ([. . .]).

The compiler always attempts to provide full diagnostics of all source text in the program, even when errors are discovered. If the compiler cannot continue on a given statement because no logic exists to place the remainder of the statement in proper context, the message states that the compiler cannot continue and that it will ignore the rest of the statement. If this occurs, you should examine the entire statement.

Diagnostic Levels

System/36 COBOL provides the following three levels of diagnostics:

Code Meaning

W Warning. A possible error was detected in the source program. Examine the statement to ensure that no coding mistake was made.

C Conditional. An error was detected, and the compiler is taking a recovery that may (conditionally) yield the correct code desired by the programmer.

E Error. A serious error was made. The program should not be run.

Note: W and C messages are suppressed when the PROCESS statement FLAG option is used.

Format of Displayed Messages

Error messages displayed at a display station are described in this section using the following format:

Message Identification Code Message

This is what the message actually looks like.

Severity:

The severity level.

Auto Response:

The recommended response for the message.

Explanation

What the error is and what causes it.

Additional Information

What the compiler did as a result.

User Response

What the programmer can do to correct it.

In the following example is a documented message using this format:

Message Identification Code	Message
CBL-1001	No DATA DIVISION OR PROCEDURE DIVISION found

Severity: 3 **Auto Response:** 2

Explanation

The Data Division or the Procedure Division portion of the COBOL source program could not be found while compiling the program.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1001).

Additional Information

While processing the source program, the compiler could not locate a Data Division or a Procedure Division header statement. If after properly inserting these required statements, the problem continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

The following information applies to displayed messages:

- Messages CBL-1001 through CBL-1099 are printed on the compiler listing and also displayed at the display station.
- Displayed messages CBL-4001 through CBL-4022 can be issued by the COBOL procedures, COBOLONL, COBOLC, COBSDA, and COBSEU. Auto response and severity only apply as indicated for each message.
- Displayed message CBL-4021 can be issued by the COBOL procedure COBSEU. All other messages will be issued from the source entry utility (SEU). For a complete explanation of SEU, see the *Source Entry Utility Guide*, SC21-7901.
- When you are using COBSDA, all messages will be issued from the screen design aid (SDA) utility. For a complete explanation of SDA, see the manual *Creating Displays: Screen Design Aid and Support Program*, SC21-7902.

Severity Levels

The severity level is used by the system to determine whether a message should be automatically responded to. For more information, see the *System Reference* manual, SC21-9020.

System/36 COBOL provides the following five levels of severity:

Code	Meaning
1	Information messages. Option 0 is the only available response.
2	This refers to messages with one response option, or messages with two response options where one option is the retry option.
3	This refers to program error messages. These messages usually have more than one response option.
4	This refers to messages for severe errors, such as permanent input/output errors or hardware errors.
5	No automatic response value can be defined for these messages.

Automatic Response Levels

The automatic response is the recommended response for that particular message, if you should choose to use automatic response for the messages. For more information, see the *System Reference* manual, SC21-9020.

System/36 COBOL provides the following four levels of automatic responses:

Code	Meaning
0	Ignore the situation and continue processing the job step.
1	Retry the situation and continue processing the job step.
2	The job step has ended, continue with the next step.
3	The job is cancelled.

Before Calling for Service

There may be instances when the error is not corrected or keeps occurring for no apparent reason, even though you follow the instructions in the Operator Information and in the Additional Information. Sometimes the message tells you to notify your programming service representative.

You can help the service representative determine what the problem is and correct it as quickly as possible if you will do the following before you call:

- Fill out a *Problem Summary Form* from the *System Problem Determination* manual, SC21-7919. This is a form on which you can record information about the system conditions at the time the error occurred.
- If you are able to use the system keyboard, do the following:
 - Press the Print key to print the information that was on the display when the error occurred.
 - If you initiated a storage dump, run the APAR or DUMP procedure to save a copy of the dump area for the service representative.
 - Run the HISTORY procedure to save a copy of the history file for the service representative.

For information about running the APAR, DUMP, and HISTORY procedures, see the *System Reference* manual.

Chapter 2. Compiler Diagnostic Messages

CBL —0002 C INVALID CHARACTER IN COLUMN 7 -- ASSUMED BLANK

Explanation:

A line may contain only a D, an asterisk, a slash, a hyphen, or a blank in column 7. One of the lines in this statement contains an invalid character in column 7.

Additional Information:

The line is processed as if column 7 contained a blank.

User Response:

Place a valid character in column 7 before recompiling.

CBL —0003 C CONTINUATION LINE INVALID AT THIS POINT -- COLUMN 7 ASSUMED BLANK

Explanation:

A continuation line (hyphen in column 7) must be immediately preceded by a normal line (blank in column 7) or by another continuation line.

Additional Information:

The line is processed as if column 7 contained a blank.

User Response:

Place statements in proper sequence, include any omitted statements, or remove any interspersed comment lines before recompiling.

CBL —0004 W CONTINUATION LINE BLANK -- LINE IGNORED

Explanation:

A continuation line (hyphen in column 7) contains no source code in columns 8 through 72.

Additional Information:

The continuation line is ignored.

User Response:

Continue the preceding line on the continuation line, remove the continuation line, or replace the hyphen in column 7 with a blank.

CBL —0005 C FIRST CHARACTER ON CONTINUATION LINE IN AREA A -- LINE ACCEPTED

Explanation:

Any sentence or entry that requires more than one line must be continued by starting the subsequent line(s) in area B.

Additional Information:

The continuation line is processed as if it started in area B.

User Response:

Begin the continuation line(s) in area B.

CBL —0006 C FIRST CHARACTER ON CONTINUATION LINE NOT QUOTE(S) -- LITERAL TERMINATED

Explanation:

An apostrophe placed anywhere in area B must precede the continuation of a literal. An apostrophe must be represented by the character (') if the QUOTE option was specified, or by the character (') if the APOST option was specified. If no option was specified, the character (') is required.

Additional Information:

The literal is ended at the end of the line in which it appears. The continuation line is processed as if it contained a blank in column 7.

User Response:

Ensure that an apostrophe precedes continuation of the literal in area B before recompiling.

CBL —0007 C NO CONTINUATION OF NONNUMERIC LITERAL -- LITERAL TERMINATED

Explanation:

When a nonnumeric is continued from one line to another, a hyphen must be placed in column 7 of the continuation line.

Additional Information:

The literal is ended at the end of the line in which it appears.

User Response:

End the literal on the previous line with an apostrophe, or continue the literal by placing a hyphen in column 7 of the continuation line (and an apostrophe before the continuation of the literal in area B.)

CBL —0008 E SIZE OF NONNUMERIC LITERAL EXCEEDS 120 -- LITERAL TRUNCATED

Explanation:

A nonnumeric literal must be composed of from 1 through 120 EBCDIC characters (excluding the quotation mark) enclosed in apostrophes.

Additional Information:

The nonnumeric literal is truncated to 120 characters.

User Response:

Ensure that the nonnumeric literal contains no more than 120 characters before recompiling.

CBL —0009 C EMPTY NONNUMERIC LITERAL — ONE BLANK ASSUMED

Explanation:

At least 1 character, blank or nonblank, must appear between the opening apostrophe and the closing apostrophe of a literal. A literal cannot be specified by apostrophes in consecutive columns.

Additional Information:

The literal is assumed to contain one blank.

User Response:

Specify contents of nonnumeric literal before recompiling.

CBL —0010 C INVALID ITEM IN AREA A -- ACCEPTED AS IF IN AREA B

Explanation:

Area A is reserved for the beginning of division headers, section names, paragraph names, the FD, the SD, and level numbers.

Additional Information:

The item is assumed to begin in area B.

User Response:

Begin the indicated item in area B, correct the item if it was intended to be a valid area B item, or place the item in the correct area.

CBL —0011 E INVALID NUMERIC LITERAL [literal] -- IGNORED

Explanation:

The given item is not a valid numeric literal although it contains numeric characters.

Additional Information:

The indicated item is discarded.

User Response:

Ensure that the indicated item is a valid word, literal, or other language element. If it was intended to be a literal, check for a misplaced or invalid sign, a misplaced decimal point, or an invalid decimal point character. (The period is valid unless DECIMAL POINT IS COMMA was specified, in which case the comma is valid.)

CBL —0012 E COPY INVALID WITHIN A COPY -- IGNORED

Explanation:

A COPY statement cannot appear within source text that is itself being included in a COPY statement.

Additional Information:

The nested COPY clause is ignored.

User Response:

Correct the library member before recompiling.

CBL —0013 W [nnnnn] SEQUENCE ERROR(S) IN PROGRAM

Explanation:

The numbers in the source statement sequence fields (columns 1 through 6) must be in ascending sequence (except for a blank field and for statements included via the COPY statement). The number of occurrences of nonascending sequence is included in the diagnostic.

Additional Information:

Each statement out of sequence is indicated by an S on the source listing. Otherwise, all compiler processing is the same.

User Response:

Rearrange the source program so that the statements are in sequence. If they are in proper sequence, renumber the source statements in the sequence number field (columns 1 through 6).

CBL —0014 E INVALID CURRENCY SIGN LITERAL -- IGNORED

Explanation:

The currency-sign literal may be any EBCDIC character except the following:

*	Multiplication sign
+	Plus sign
-	Minus sign or hyphen
=	Equal sign
/	Slash or division sign
.	Period
;	Semicolon
(Left parenthesis
)	Right parenthesis
,	Comma
' or "	Apostrophe or quotation mark
Space	Space or blank
Numeric	0-9
Alphabetic	A, B, C, D, L, P, R, S, V, X, Z

Additional Information:

The character is not used by the compiler in scanning PICTURE clauses. The default character \$ is used instead.

User Response:

Substitute a valid character in the CURRENCY clause before recompiling.

CBL —0015 E INVALID PROGRAM-NAME IN CALL STATEMENT -- IGNORED

Explanation:

The program name in a CALL statement must conform to the rules for forming a procedure name. If it does not conform, the name is identified by the compiler as being a nonnumeric literal.

Additional Information:

The literal is discarded.

User Response:

Correct the nonnumeric literal before recompiling.

CBL —0016 W INVALID IDEOGRAPHIC LITERAL DETECTED, NONNUMERIC LITERAL ASSUMED

Explanation:

A literal was found that began with an apostrophe followed immediately by the S/O control character, but one of the following occurred:

- An odd number of one-byte characters was found between the S/O and the terminating S/I control characters.
- The terminating S/I control character was not immediately followed by an apostrophe.
- The ideographic literal spans multiple source lines and does not follow the rules for continuation of ideographic literals.

Additional Information:

The literal is processed as if it was a nonnumeric literal.

User Response:

Correct the literal so that it conforms to the rules for an ideographic literal.

CBL —0017 C NAME [name] BEGINS OR ENDS WITH HYPHEN -- ACCEPTED

Explanation:

A COBOL name can have embedded hyphens, but it must not begin or end with a hyphen.

Additional Information:

The name and all references to it are accepted.

User Response:

Replace the name and all references to the name with a name that does not begin or end with a hyphen.

CBL —0018 C [literal] CHARACTER MAY NOT PRECEDE [literal] CHARACTER -- ACCEPTED

Explanation:

In COBOL, certain characters may not precede other characters; for example, a plus sign may not precede an alphabetic character.

Additional Information:

If either of the characters is a space, the compiler assumes the space is to be omitted. If neither is a space, the compiler attempts to construct a valid item or items. If no other diagnostics are listed for the statement, the recovery was probably successful.

Note: The following are the possible characters:

.	Period
(Left parenthesis
)	Right parenthesis
,	Comma or semicolon

*	*Multiplication (*) or exponentiation (**) characters
/	Slash (division sign)
+	Plus sign
-	Minus sign or hyphen
=/GREATER/	Equal sign, greater than, or
LESS THAN	less than
Space(s)	Spaces or blanks
' or "	Apostrophe or quotation mark
Numeric	0, 1, 2, . . . , 99
Alphabetic	A, B, C, . . . , Z, or a character not in the COBOL character set

User Response:

Correct the order of the indicated characters before recompiling.

CBL —0019 C [name or identifier] IS INVALID GROUP OF CHARACTERS — IGNORED

Explanation:

The indicated continuous string of characters does not conform to the rules for forming a COBOL language element.

Additional Information:

The indicated character string is not processed by the compiler.

User Response:

Correct the character string so it is a valid COBOL item before recompiling.

CBL —0020 E [name or item] EXCEEDS 30 CHARACTERS — TRUNCATED

Explanation:

A character string that represents a PICTURE or a name must not be longer than 30 characters.

Additional Information:

The character string is processed as if it contained only its first 30 characters.

User Response:

If the item is a PICTURE string, use the repetition factor (nnn) to reduce the length of the character string. If the item is a name, shorten it and all references to it to a unique name of 30 or fewer characters.

CBL —0021 E COPY STATEMENT NOT TERMINATED WITH PERIOD AND SPACE -- COPY IGNORED

Explanation:

A COPY statement must be ended by a period and a space.

Additional Information:

The COPY statement is ignored.

User Response:

Enter a period and space(s) at the end of the COPY statement before recompiling.

CBL —0023 E LIBRARY MEMBER NOT FOUND -- COPY IGNORED

Explanation:

The first 8 characters of the library name in a COPY statement, after the compiler may have converted any special characters, must correspond to an entry in the user's source library.

Additional Information:

The COPY statement is ignored.

User Response:

Either correct the misspelled library name, ensure that the member is in the source library, insert a LIBRARY option if the source library is not in the same library as the COBOL compiler, or use the IN/OF library name option of the COPY statement.

CBL —0024 E INVALID LIBRARY-NAME FORMAT -- COPY IGNORED

Explanation:

The library-name in a COPY statement must conform to the rules for forming a COBOL procedure name.

Additional Information:

The COPY statement is ignored.

User Response:

Correct the spelling of the library-name before recompiling.

CBL —0025 W INVALID ELEMENT [element]

Explanation:

In the PROCESS statement, a character or set of characters is unexpected in relation to its preceding items.

Additional Information:

If element is the first or only word of another option, the previous option is discontinued, and processing begins with

element. If element does not start an option, element is discarded, and processing for the previous option continues.

User Response:

Correct the syntax or spelling of the option before recompiling.

**CBL —0027 C IDENTIFICATION DIVISION
MISSING**

Explanation:

The first COBOL source statement must be Identification Division.

Additional Information:

None.

User Response:

Insert the Identification Division header before the COBOL source statements (and after the PROCESS statement, if any) or correct the spelling or location of the word, IDENTIFICATION.

**CBL —0028 C PARAGRAPH OUT OF
SEQUENCE -- ACCEPTED**

Explanation:

Paragraphs in the Identification Division must be in this order: PROGRAM ID, AUTHOR, INSTALLATION, DATE-WRITTEN, SECURITY.

Additional Information:

None.

User Response:

Place the paragraphs in the proper sequence.

**CBL —0029 C DUPLICATE ELEMENT --
ACCEPTED**

Explanation:

A paragraph name for the Identification Division and the Identification Division header can appear only once.

Additional Information:

The current item overrides the previous one.

User Response:

Delete the duplicate element.

**CBL —0030 C INVALID ITEM [item] --
RESTARTING AT NEXT
KEYWORD IN AREA A**

Explanation:

The listed item is unexpected in the Identification Division.

Additional Information:

If the item is a key word in area A, the scan resumes with the item; if not, the scan resumes at the next key word in area A.

User Response:

Correct the syntax of the paragraph or division header in error or move the indicated item to its proper location in the source program.

CBL —0031 C PROGRAM-ID MISSING

Explanation:

The PROGRAM-ID paragraph is required within the Identification Division.

Additional Information:

The program is processed as if the program name CBLOBJ were specified.

User Response:

Place a PROGRAM-ID paragraph after the Identification Division header.

**CBL —0032 C PERIOD AND SPACE NOT
FOUND -- ACCEPTED**

Explanation:

COBOL paragraph and division headers of the Identification Division must end with a period followed by at least one space.

Additional Information:

Processing continues as if a period and space had been specified.

User Response:

Insert a period and space(s) after the header.

**CBL —0033 C ELEMENT NOT IN AREA A --
ACCEPTED**

Explanation:

Division and paragraph headers of the Identification Division must begin in area A (columns 1 through 7).

Additional Information:

The header is assumed to begin in area A.

User Response:

Ensure that the header begins in area A before recompiling.

**CBL —0034 C INVALID PROGRAM-NAME --
IGNORED**

Explanation:

The program name must conform to the rules for forming a program name.

Additional Information:

The name is ignored, and processing continues as though the program name CBLOBJ had been specified.

User Response:

Correct the program name before recompiling.

CBL —0035 C PARAGRAPH NOT IN IDENTIFICATION DIVISION -- ACCEPTED

Explanation:

The Identification Division header must precede all paragraphs for the Identification Division.

Additional Information:

The paragraph is accepted as if it were in the Identification Division.

User Response:

Place the Identification Division header before all source statements (and after the PROCESS statement, if used).

CBL —0036 C INVALID IDENTIFICATION DIVISION HEADER -- ACCEPTED

Explanation:

The word DIVISION is required after the word IDENTIFICATION in area A.

Additional Information:

The statement is accepted as if it had been coded correctly.

User Response:

Correct the Identification Division header.

CBL —0105 C MISSING PERIOD -- ACCEPTED

Explanation:

All sentences in a COBOL program must end with a period.

Additional Information:

The statement is accepted as if it were ended by a period.

User Response:

Supply missing period before recompiling.

CBL —0107 E SELECT CLAUSE FOR FILE INVALID OR MISSING -- FILE IGNORED

Explanation:

Either the SELECT clause had an invalid system name or no SELECT clause was specified for the file.

Additional Information:

The file is ignored.

User Response:

Correct invalid SELECT clause before recompiling.

CBL —0108 C BLOCK CONTAINS CLAUSE IS ZERO -- UNBLOCKED RECORDS ASSUMED

Explanation:

The computed size of the BLOCK CONTAINS clause is zero.

Additional Information:

The records are assumed to be unblocked.

User Response:

Correct the invalid BLOCK clause specification before recompiling.

CBL —0110 C MAXIMUM BLOCKSIZE EXCEEDED -- UNBLOCKED ASSUMED

Explanation:

A BLOCK CONTAINS clause has a block size greater than 9999.

Additional Information:

The compiler assumes that records are not blocked.

User Response:

Ensure that the block size is 9999 or less before recompiling.

CBL —0111 E MAXIMUM RECORD SIZE EXCEEDED -- CLAUSE IGNORED

Explanation:

The RECORD CONTAINS clause has a record size greater than 4096 characters.

Additional Information:

The RECORD CONTAINS clause is ignored.

User Response:

Ensure that the record size is 4096 or less before recompiling.

**CBL —0112 C INVALID ENVIRONMENT
DIVISION HEADER --
ACCEPTED**

Explanation:

The word DIVISION or a period and space is missing from the Environment Division header.

Additional Information:

The Environment Division header is assumed to be correct.

User Response:

Correct syntax of indicated statement before recompiling.

**CBL —0113 E ENVIRONMENT DIVISION IN
INVALID LOCATION**

Explanation:

The Environment Division must follow the Identification Division.

Additional Information:

Processing continues at the next section or division. All statements up to the next section or division are ignored; at that point, processing continues.

User Response:

Supply valid Environment Division header before recompiling.

**CBL —0114 E MORE THAN 25 FILES
SPECIFIED -- IGNORED**

Explanation:

The number of files in a program must not exceed 25.

Additional Information:

The file is ignored.

User Response:

Reduce the number of files in the program to 25 before recompiling.

**CBL —0115 C ENVIRONMENT DIVISION
MISSING -- ACCEPTED**

Explanation:

No Environment Division header was found.

Additional Information:

The compiler assumes that there is no Environment Division.

User Response:

Include Environment Division header before recompiling.

**CBL —0117 E [key word] NOT IN
ENVIRONMENT DIVISION**

Explanation:

The indicated word does not appear in the Environment Division.

Additional Information:

The statement is ignored, and the compiler skips to the next section, paragraph, or division.

User Response:

Ensure that the indicated word is in the Environment Division before recompiling.

**CBL —0118 C [key word] PARAGRAPH,
SECTION, OR KEYWORD IN
INVALID LOCATION --
ACCEPTED**

Explanation:

The paragraph name, the section header, or a key word is not in proper sequence or is duplicated in the Environment Division.

Additional Information:

The paragraph name or section header is accepted.

User Response:

Ensure that all paragraph names and section headers are unique and in proper sequence before recompiling.

**CBL —0119 C INVALID CONFIGURATION
SECTION HEADER --
ACCEPTED**

Explanation:

The word SECTION or a period and space is missing from the Configuration Section.

Additional Information:

The Configuration Section header is accepted as if it were correct.

User Response:

Correct the section header before recompiling.

**CBL —0120 E [name or key word] NOT IN
CONFIGURATION SECTION**

Explanation:

The indicated name or key word is not within the Configuration Section.

Additional Information:

The statement is ignored, and the compiler skips to the next paragraph, section, or division.

User Response:

Ensure that the indicated item is in the Configuration Section before recompiling.

**CBL —0121 W INVALID OBJECT-COMPUTER
OR SOURCE-COMPUTER NAME
-- IGNORED**

Explanation:

The name following SOURCE-COMPUTER or OBJECT-COMPUTER is not IBM-S36.

Additional Information:

The paragraph is treated as documentation.

User Response:

Supply the valid name before recompiling.

**CBL —0122 C INVALID OBJECT-COMPUTER
MEMORY SIZE CLAUSE --
IGNORED**

Explanation:

The OBJECT-COMPUTER MEMORY SIZE clause is not an integer from 1 through 65,536.

Additional Information:

The MEMORY SIZE clause is ignored (the size of the current compiler region is assumed).

User Response:

Correct the invalid statement before recompiling.

**CBL —0123 E INVALID FUNCTION-NAME IN
SPECIAL-NAMES PARAGRAPH
-- IGNORED**

Explanation:

The function name is not one of the following: SYSTEM-CONSOLE, REQUESTOR, CSP, C01, LOCAL-DATA, ATTRIBUTE-DATA, UPSI-0, UPSI-1, UPSI-2, UPSI-3, UPSI-4, UPSI-5, UPSI-6, UPSI-7, SYSTEM-SHUTDOWN; or the specified clause was not a CURRENCY SIGN, DECIMAL-POINT IS COMMA, or an alphabet name clause.

Additional Information:

The name is ignored, and the compiler skips to the next function name or clause, if any.

User Response:

Correct the function name or clause before recompiling.

**CBL —0124 W INVALID SYNTAX IN
SPECIAL-NAMES**

Explanation:

The word, IS, is missing from the ON or OFF STATUS clause in a SPECIAL-NAMES statement for a switch.

Additional Information:

The statement is ignored, and the compiler scans for the next function name in the SPECIAL-NAMES paragraph.

User Response:

Correct the invalid clause before recompiling.

**CBL —0125 E INVALID MNEMONIC-NAME OR
CONDITION-NAME IN
SPECIAL-NAMES**

Explanation:

The mnemonic name or condition name in a SPECIAL-NAMES paragraph is not a valid data name.

Additional Information:

The statement is ignored, and the compiler scans for the next function name in the SPECIAL-NAMES paragraph.

User Response:

Correct the name before recompiling.

**CBL —0126 C INVALID SYNTAX IN
CURRENCY SIGN CLAUSE --
ACCEPTED**

Explanation:

The word, IS, was omitted from or is misplaced in the CURRENCY-SIGN clause.

Additional Information:

The clause is assumed to be correct.

User Response:

Correct the syntax of the CURRENCY-SIGN clause before recompiling.

**CBL —0127 C DUPLICATE CLAUSE FOUND IN
SPECIAL-NAMES -- FIRST ONE
ACCEPTED**

Explanation:

A duplicate clause is detected in the SPECIAL-NAMES statement. The first clause is used.

Additional Information:

The duplicate clause is ignored.

User Response:

Delete the duplicate clause before recompiling.

**CBL —0128 C INVALID DECIMAL-POINT IS
COMMA CLAUSE -- ACCEPTED**

Explanation:

Either the word, IS, or the word, COMMA, is missing from or is misplaced in the DECIMAL-POINT IS COMMA clause.

Additional Information:

DECIMAL-POINT IS COMMA is assumed.

User Response:

Correct the syntax of the DECIMAL-POINT IS COMMA clause before recompiling.

**CBL —0129 C INVALID INPUT-OUTPUT
SECTION HEADER --
ACCEPTED**

Explanation:

Either SECTION or a period and space is missing from the Input-Output Section header.

Additional Information:

The header is assumed to be correct.

User Response:

Correct the invalid statement before recompiling.

**CBL —0130 E [name or key word] NOT IN
INPUT-OUTPUT SECTION --
IGNORED**

Explanation:

Either the I-O CONTROL paragraph or the FILE-CONTROL paragraph is not in the Input-Output Section.

Additional Information:

The paragraph is ignored, and the compiler skips to the next section, paragraph, or division.

User Response:

Ensure that the paragraph is in the Input-Output Section before recompiling.

**CBL —0131 C INVALID ITEM [name or key word]
IN FILE-CONTROL PARAGRAPH
-- ITEM IGNORED**

Explanation:

The indicated name or key word is invalid as used in the FILE-CONTROL paragraph.

Additional Information:

The indicated item is ignored.

User Response:

Correct the indicated item before recompiling.

**CBL —0132 C INVALID ITEM [name or key word]
IN SELECT CLAUSE -- ITEM
IGNORED**

Explanation:

The indicated name or key word is invalid as used in the SELECT clause.

Additional Information:

The indicated item is ignored.

User Response:

Correct the invalid name or key word before recompiling.

**CBL —0133 E INVALID FILE-NAME IN
SELECT CLAUSE -- CLAUSE
IGNORED**

Explanation:

The word SELECT is not followed by a valid file name. (A name is not a valid file name if it is a reserved word, a duplicate of a data name, or is not uniquely defined in a valid FD or SD.)

Additional Information:

The SELECT clause is ignored.

User Response:

Correct the invalid file name before recompiling.

**CBL —0134 E INVALID SYSTEM-NAME IN
ASSIGN CLAUSE -- FILE
IGNORED**

Explanation:

The format or combinations for the system name are incorrect.

Additional Information:

The file is ignored.

User Response:

Correct the system name specification before recompiling.

**CBL —0136 E INVALID DATA ITEM IN
CONTROL-AREA — CLAUSE
IGNORED**

Explanation:

The data name in the CONTROL-AREA clause was omitted or was invalid for one of the following reasons:

- The data name was not unique.
- The data name was less than 12 characters long.
- The data name was not an alphanumeric item.
- The data name was not in the Working-Storage or Linkage Section.
- The data name was not a group item.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid data name before recompiling.

**CBL —0137 C INVALID ACCESS CLAUSE —
SEQUENTIAL ASSUMED**

Explanation:

The word, DYNAMIC, RANDOM, or SEQUENTIAL is missing or misplaced from the ACCESS MODE clause.

Additional Information:

ACCESS IS SEQUENTIAL is assumed.

User Response:

Specify the access mode before recompiling.

**CBL —0138 C INVALID ORGANIZATION
CLAUSE — SEQUENTIAL
ASSUMED**

Explanation:

The word RELATIVE, SEQUENTIAL, INDEXED, or TRANSACTION is missing from or is misplaced in the ORGANIZATION clause.

Additional Information:

The compiler assumes ORGANIZATION IS SEQUENTIAL for any further processing of the affected file.

User Response:

Correct the syntax of the ORGANIZATION clause before recompiling.

**CBL —0139 E INVALID DATA-NAME IN KEY
CLAUSE -- CLAUSE IGNORED**

Explanation:

The error was caused by one of the following:

- The data name of the key is missing.
- The RELATIVE KEY data name was not in the Working-Storage Section or is not a numeric DISPLAY or COMPUTATIONAL(COMP) data item.
- The RECORD KEY is not between 1 and 120 bytes long (inclusive), or is an edited field.
- The RECORD KEY is not within the record area of the file to which it belongs, or is not defined in the program.
- The key is unqualified and the data name is not unique.
- The noncontiguous key fields overlap, are not unique, exceed three in number, exceed a total size of 120 characters, or are not all found in the same logical record.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid statement before recompiling.

**CBL —0140 W INVALID RESERVE CLAUSE --
ONE AREA ASSUMED**

Explanation:

An invalid value of zero or a noninteger was specified in the RESERVE clause.

Additional Information:

RESERVE 1 AREA is assumed.

User Response:

Correct the invalid specification before recompiling.

**CBL —0141 E [name or key word] CLAUSE
DUPLICATED IN SELECT
CLAUSE — FIRST ONE
ACCEPTED**

Explanation:

The indicated name or key word was duplicated in another SELECT clause.

Additional Information:

The first occurrence is accepted, and duplicate entries are ignored.

User Response:

Delete the duplicate name or key word before recompiling.

**CBL —0142 W FORMAT FILE TYPE INVALID
-- SFGR SCREEN FORMAT FILE
ASSUMED**

Explanation:

The format type must be either a 'S' for an SFGR load member or a 'C' for an IDDU format file. SFGR load member is assumed.

Additional Information:

The compiler assumes the format is an SFGR load member when the format type for the TRANSACTION file is not clear or is incorrectly specified.

User Response:

Choose either an SFGR load member ('S') or an IDDU format file ('C') before recompiling.

**CBL —0143 E NO RELATIVE KEY FOR
RANDOM/DYNAMIC FILE--FILE
PROCESSED WITHOUT KEY**

Explanation:

The RELATIVE KEY clause must be specified for random access or dynamic access direct disk files.

Additional Information:

Results at run time are unpredictable for random access operations.

User Response:

Specify the RELATIVE KEY before recompiling.

**CBL —0144 E INVALID KEY TYPE FOR
ACCESS METHOD OR
ORGANIZATION -- KEY
IGNORED**

Explanation:

A RELATIVE KEY or RECORD KEY was specified for the wrong disk file organization or for a non-disk file.

Additional Information:

The KEY clause is ignored.

User Response:

Correct the invalid specification before recompiling.

**CBL —0145 E NO ASSIGN CLAUSE IN
FILE-CONTROL ENTRY -- FILE
IGNORED**

Explanation:

All files used in a program (named in a SELECT clause) must be assigned to an external medium by using the ASSIGN clause.

Additional Information:

The file is ignored.

User Response:

Supply the ASSIGN clause before recompiling.

**CBL —0146 E INVALID ORGANIZATION FOR
TRANSACTION PROCESSING --
TRANSACTION ASSUMED**

Explanation:

The organization clause is missing or specified RELATIVE, SEQUENTIAL, OR INDEXED, but the implementer name is that of a TRANSACTION file.

Additional Information:

The compiler assumes ORGANIZATION IS TRANSACTION for any further processing of the affected file.

User Response:

Correct the invalid ORGANIZATION clause before recompiling.

**CBL —0147 C [element] INVALID IN
I-O-CONTROL PARAGRAPH --
IGNORED**

Explanation:

The indicated element cannot appear in the I-O-CONTROL paragraph.

Additional Information:

The invalid item is ignored.

User Response:

Correct invalid item before recompiling.

**CBL —0148 C NUMBER OF SAME AREA
CLAUSES EXCEEDS 15 --
CLAUSE IGNORED**

Explanation:

No more than 15 SAME AREA clauses or 15 SAME SORT/SORT-MERGE AREA clauses can appear in a program.

Additional Information:

The first 15 SAME AREA clauses or 15 SAME SORT/SORT-MERGE AREA clauses are recognized; any more than 15 are ignored.

User Response:

Ensure that no more than 15 SAME AREA clauses appear before recompiling.

CBL —0149 C INVALID FILE-NAME IN SAME AREA CLAUSE -- FILE NAME IGNORED

Explanation:

An invalid file name is in the SAME AREA clause.

Additional Information:

The file name is ignored.

User Response:

Ensure that file name is valid in the SAME AREA clause.

CBL —0151 E FILE-NAME IN MULTIPLE RERUN CLAUSES -- IGNORED

Explanation:

A given file name cannot be specified in more than one RERUN clause.

Additional Information:

All RERUN entries are treated as comments.

User Response:

Ensure that a given file name appears in only one RERUN clause before recompiling.

CBL —0152 E INVALID FILE-NAME IN RERUN CLAUSE -- CLAUSE IGNORED

Explanation:

The file name was omitted or was invalid in a RERUN clause.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid statement before recompiling.

CBL —0153 E INVALID NUMBER OF RECORDS IN RERUN CLAUSE -- CLAUSE IGNORED

Explanation:

The number of records in the RERUN clause must not exceed 32,767.

Additional Information:

The clause is ignored.

User Response:

Ensure that the number of records does not exceed 32,767 before recompiling.

CBL —0154 E INVALID SYSTEM-NAME IN RERUN CLAUSE -- CLAUSE IGNORED

Explanation:

The system name is not DISK-nnnnnnnn, where nnnnnnnn is an external name composed of from 1 through 8 characters.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid specification before recompiling.

CBL —0155 C SYNTAX ERROR IN RERUN CLAUSE -- ACCEPTED

Explanation:

One or more of the required key words is missing or misplaced.

Additional Information:

The clause is processed as if the required key words were present.

User Response:

Include all required words in the RERUN clause before recompiling.

CBL —0156 E INVALID RESERVE CLAUSE -- CLAUSE IGNORED

Explanation:

The RESERVE clause is not valid in a SELECT statement for a TRANSACTION file.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid specification before recompiling.

CBL —0157 E INVALID CONTROL-AREA CLAUSE -- CLAUSE IGNORED

Explanation:

The CONTROL-AREA clause is not valid in a SELECT statement for a disk or printer file. (The CONTROL-AREA clause is valid only for a TRANSACTION file.)

Additional Information:

The clause is ignored.

User Response:

Correct the invalid specification before recompiling.

**CBL —0158 W MORE THAN TWO RESERVE
AREAS SPECIFIED -- TWO
ASSUMED**

Explanation:

More than two reserve areas were specified.

Additional Information:

Two reserve areas are assumed.

User Response:

Correct the syntax of the RESERVE clause before recompiling.

**CBL —0159 E INVALID OPTIONAL CLAUSE --
CLAUSE IGNORED**

Explanation:

The OPTIONAL clause is only valid in a SELECT statement for a SEQUENTIAL file.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid specification before recompiling.

**CBL —0160 E NO RECORD KEY SPECIFIED
FOR INDEXED FILE -- FILE
PROCESSED WITHOUT KEY**

Explanation:

An indexed file must have a RECORD KEY specification.

Additional Information:

The results at run time are unpredictable.

User Response:

Specify the RECORD KEY clause before recompiling.

**CBL —0161 W EMPTY PARAGRAPH IN
ENVIRONMENT DIVISION**

Explanation:

No statements were found in a paragraph of the Environment Division.

Additional Information:

The empty paragraph is ignored.

User Response:

Delete the paragraph header before recompiling.

**CBL —0162 E INVALID ACCESS MODE
CLAUSE -- SEQUENTIAL
ASSUMED**

Explanation:

The ACCESS MODE clause specifies DYNAMIC or RANDOM processing; both of these modes are incompatible with TRANSACTION file processing.

Additional Information:

The compiler assumes sequential.

User Response:

Correct the invalid specification before recompiling.

**CBL —0163 C INVALID LOCATION OF
RELATIVE KEY PHRASE --
ACCEPTED**

Explanation:

The RELATIVE KEY IS phrase, when used, should immediately follow the ACCESS MODE IS clause.

Additional Information:

The RELATIVE KEY IS phrase is processed as if it were part of the ACCESS MODE IS clause.

User Response:

Ensure that the RELATIVE KEY IS phrase is placed within the ACCESS MODE IS clause before recompiling.

**CBL —0164 E INVALID FILE STATUS CLAUSE
-- CLAUSE IGNORED**

Explanation:

The key word STATUS is either missing or misplaced. The key word FILE was found, but the key word STATUS does not follow it.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid clause before recompiling.

**CBL —0165 E INVALID DATA ITEM IN FILE
STATUS -- CLAUSE IGNORED**

Explanation:

The data name in the FILE STATUS clause was invalid, either because it was omitted or because of one of the following:

- The data name was not unique.
- The data name was not 2 characters long.
- The data name was not an alphanumeric item.
- The data name was not in the Working-Storage Section or Linkage Section.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid data name before recompiling.

CBL —0166 C INVALID ELEMENT [name or key word] IN STATEMENT -- IGNORED

Explanation:

An invalid element was found in a section header or paragraph in the Environment Division.

Additional Information:

The remainder of the header is ignored.

User Response:

Correct the indicated element before recompiling.

CBL —0167 E INVALID ALPHABET NAME IN PROGRAM COLLATING SEQUENCE CLAUSE -- IGNORED

Explanation:

The ALPHABET NAME in the PROGRAM COLLATING SEQUENCE clause was invalid because it was omitted or because it was not unique.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid ALPHABET NAME before recompiling.

CBL —0168 C DUPLICATE CLAUSE IN OBJECT-COMPUTER PARAGRAPH

Explanation:

A duplicate MEMORY SIZE, PROGRAM COLLATING SEQUENCE or SEGMENT LIMIT clause was found in the indicated paragraph.

Additional Information:

The first occurrence is accepted, and duplicate entries are ignored.

User Response:

Delete the duplicated clause(s) before recompiling.

CBL —0169 C SYNTAX ERROR IN SEGMENT-LIMIT CLAUSE -- ACCEPTED

Explanation:

The required key word, IS, is either missing or misplaced.

Additional Information:

The clause is processed as if the required word were present.

User Response:

Include the word IS in the SEGMENT-LIMIT clause before recompiling.

CBL —0170 E INVALID SEGMENT-LIMIT CLAUSE -- NONE ASSUMED

Explanation:

The segment number specified on the SEGMENT-LIMIT clause is either missing or invalid. It is invalid if it is nonnumeric or not between 01 and 49, inclusive.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid segment number before recompiling.

CBL —0171 E INVALID CLAUSE IN SD FILE-CONTROL ENTRY - CLAUSE IGNORED

Explanation:

The only clauses allowed in an SD entry are the SELECT and ASSIGN clauses; any other clauses are invalid.

Additional Information:

The clause(s) is ignored.

User Response:

Remove the invalid clause(s) before recompiling.

CBL —0172 C INVALID CLAUSE IN SD FILE DESCRIPTION - CLAUSE IGNORED

Explanation:

Only a RECORD CONTAINS or DATA RECORDS clause is allowed to appear on an SD file description.

Additional Information:

The clause(s) is ignored.

User Response:

Remove the invalid clause(s) before recompiling.

**CBL —0173 E ALPHABET-NAME PREVIOUSLY
DEFINED -- CLAUSE IGNORED**

Explanation:

The ALPHABET NAME was used in a previous ALPHABET NAME clause.

Additional Information:

The clause is ignored, and the compiler skips to the next function name or clause, if any.

User Response:

Correct the ALPHABET NAME before recompiling.

**CBL —0174 E PROGRAM COLLATING
SEQUENCE NOT DEFINED**

Explanation:

The ALPHABET NAME specified in the PROGRAM COLLATING SEQUENCE clause was not defined with an ALPHABET NAME clause.

Additional Information:

The SPECIAL-NAMES paragraph is accepted as is.

User Response:

Add the required ALPHABET NAME clause before recompiling.

**CBL —0175 E CHARACTER PREVIOUSLY
USED IN THIS COLLATING
SEQUENCE -- CHARACTER
IGNORED**

Explanation:

The character value specified in the literal was used previously in the same collating sequence.

Additional Information:

The value is ignored, and the compiler continues with the next value or field.

User Response:

Correct the literal before recompiling.

**CBL —0176 E INVALID PHRASE IN
ALPHABET-NAME CLAUSE --
PHRASE IGNORED**

Explanation:

The ALPHABET NAME clause contains a phrase other than: STANDARD-1, NATIVE, or a literal phrase.

Additional Information:

The phrase is ignored, and the compiler skips to the next function name or clause, if any.

User Response:

Correct the ALPHABET NAME clause before recompiling.

**CBL —0177 E INVALID LITERAL IN
ALPHABET-NAME CLAUSE --
LITERAL IGNORED**

Explanation:

The field is one of the following:

- A numeric literal not in the range of 1 through 256.
- A nonnumeric literal of more than 1 character after the THROUGH or ALSO phrase.

Additional Information:

The field is ignored, and the compiler skips to the next field.

User Response:

Correct the literal phrase before recompiling.

**CBL —0178 C REQUIRED
SOURCE-COMPUTER
KEYWORD NOT FOUND**

Explanation:

The compiler expects to find the required SOURCE-COMPUTER paragraph in the Environment Division. This paragraph either was not specified correctly or was misplaced.

Additional Information:

The compiler assumes that SOURCE-COMPUTER. IBM-S36. was specified.

User Response:

Specify the SOURCE-COMPUTER paragraph correctly before recompiling.

**CBL —0179 C REQUIRED
OBJECT-COMPUTER KEYWORD
NOT FOUND**

Explanation:

The compiler expects to find the required OBJECT-COMPUTER paragraph in the Environment Division. This paragraph was either not specified correctly or was misplaced.

Additional Information:

The compiler assumes that OBJECT-COMPUTER. IBM-S36. was specified.

User Response:

Specify the OBJECT-COMPUTER paragraph correctly before recompiling.

**CBL —0180 W RELATIVE KEY DATA ITEM
LENGTH GREATER THAN 7 -- 7
ASSUMED**

Explanation:

A direct file must not have a key length greater than seven (7) bytes.

Additional Information:

The compiler assumes that the length of the RELATIVE KEY data item is 7 bytes.

User Response:

Ensure that the length of the RELATIVE KEY data item is 7 bytes or less.

**CBL —0181 C ONLY ONE FILE FOUND IN
SAME AREA CLAUSE --
ACCEPTED**

Explanation:

This clause normally specifies two or more files.

Additional Information:

The SAME AREA clause is syntax-checked and treated as comments.

**CBL —0186 E INVALID ACCESS MODE FOR
PRINTER OR SEQUENTIAL
DISK FILE -- SEQUENTIAL
ASSUMED**

Explanation:

The ACCESS clause for a printer or a sequential disk file was other than SEQUENTIAL.

Additional Information:

The ACCESS clause is ignored, and SEQUENTIAL is assumed.

User Response:

Correct the ACCESS clause before recompiling.

**CBL —0189 W APPLY CORE-INDEX
SENTENCE IS NOT NEEDED --
IGNORED**

Explanation:

The APPLY clause is treated as comments.

Additional Information:

The sentence is ignored.

**CBL —0200 E INVALID LOCATION OF FILE
SECTION**

Explanation:

The File Section must begin on the line following the Data Division header.

Additional Information:

The File Section header is ignored, and the compiler skips to the next section header, FD, SD, or data item.

User Response:

Ensure that the File Section begins on the line following the Data Division header before recompiling.

**CBL —0201 E FD OR SD NOT IN FILE
SECTION**

Explanation:

The File Section header must precede all file control entries (FD, SD).

Additional Information:

The FD/SD is ignored, and the compiler skips to the next section header, FD, SD, or data item.

User Response:

Ensure that FD/SD is located in the File Section before recompiling.

**CBL —0202 E INVALID FILE-NAME
FOLLOWING FD OR SD -- FILE
IGNORED**

Explanation:

Each file description entry must consist of a level indicator. FD or SD, followed by a unique file name and a series of independent clauses.

Additional Information:

The file description is ignored.

User Response:

Ensure that the level indicators are followed by a valid file name before recompiling.

CBL —0203 E INVALID OR MISSING NAME AFTER LEVEL NUMBER

Explanation:

A data name or the word FILLER must be the first word following the level number in each data description entry.

Additional Information:

The level item is ignored.

User Response:

Ensure that a data name or the word FILLER is the first word following the level number before recompiling.

CBL —0204 E INVALID OCCURS CLAUSE -- CLAUSE IGNORED

Explanation:

The error was caused by one of the following:

- The literals in the OCCURS clause are missing or do not have a value between 1 and 32,768.
- The second literal is not greater than the first.
- The DEPENDING ON clause is not followed by a data name.
- One of the required key words is missing.

Additional Information:

The clause is ignored.

User Response:

Correct the literal(s) before recompiling.

CBL —0205 E INVALID REDEFINES CLAUSE -- CLAUSE IGNORED

Explanation:

No data name was found, the clause was preceded by another clause, or the redefining item describes a variable-length table.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid clause before recompiling.

CBL —0206 E/W INVALID PICTURE CLAUSE -- CLAUSE IGNORED

Explanation:

The PICTURE literal is invalid.

Additional Information:

The data item is ignored.

User Response:

Correct the invalid clause before recompiling.

CBL —0207 E INVALID LITERAL IN VALUE CLAUSE -- CLAUSE IGNORED

Explanation:

The error was caused by one of the following:

- A numeric literal was specified for a nonnumeric data item.
- The number of significant digits in the numeric literal exceeds the number of significant digits in the data item.
- The value is not within an allowable range for this data item.
- An invalid literal or no literal followed the ALL literal phrase.
- An alphanumeric or character literal was specified for a numeric data item.

Additional Information:

The VALUE clause is ignored.

User Response:

Correct the invalid literal before recompiling.

CBL —0208 E INVALID LEVEL NUMBER -- ITEM IGNORED

Explanation:

Either the level number is not 01 through 49, 66, 77, or 88 or the level number of a group item is invalid.

Additional Information:

The data item is ignored.

User Response:

Supply correct level number before recompiling.

CBL —0209 E INVALID OBJECT OF REDEFINES -- CLAUSE IGNORED

Explanation:

The object of the REDEFINES clause is an undefined data name, contains a REDEFINES clause, is not at the same level, or is separated from the statement containing the REDEFINES clause by a data item that is not part of a REDEFINES clause.

Additional Information:

The REDEFINES clause is ignored.

User Response:

Correct invalid specification before recompiling.

**CBL —0210 E NUMERIC ITEM EXCEEDS 18 --
ITEM IGNORED**

Explanation:

The numeric item has more than 18 digits.

Additional Information:

The numeric item is ignored.

User Response:

Ensure that the numeric item does not exceed 18 digits before recompiling.

**CBL —0211 E VALUE EXCEEDS SIZE OF ITEM
-- CLAUSE IGNORED**

Explanation:

The number of digits to the left of the decimal point in the numeric literal is greater than the number of digits to the left of the decimal point in the data item.

Additional Information:

The VALUE clause is ignored, and the data item is not initialized.

User Response:

Correct size of data item before recompiling.

**CBL —0212 E SIZE OF EDITED ITEM
EXCEEDS 127 -- ITEM IGNORED**

Explanation:

The size of a numeric edited data item may not exceed 127 characters (bytes) of storage.

Additional Information:

The item is ignored.

User Response:

Correct size of item before recompiling.

**CBL —0213 E VALUE IN OCCURS ITEM --
VALUE CLAUSE IGNORED**

Explanation:

OCCURS and VALUE clauses cannot both be specified for a data item.

Additional Information:

The VALUE clause is ignored.

User Response:

Correct invalid specification before recompiling.

**CBL —0214 E INVALID PARENT OF
VARIABLE LENGTH TABLE --
TABLE IGNORED**

Explanation:

An item containing an OCCURS . . . DEPENDING ON clause is subordinate to an item containing either another OCCURS clause or a REDEFINES clause.

Additional Information:

The subordinate OCCURS clause is ignored.

User Response:

Ensure that no parent of the variable-length table contains an OCCURS clause or REDEFINES clause or that the subordinate table is of fixed length.

**CBL —0215 E NUMBER OF NESTED OCCURS
EXCEEDS LIMIT -- FIRST
THREE ACCEPTED**

Explanation:

Only three nested levels of the OCCURS clause are allowed.

Additional Information:

All levels of the OCCURS clause beyond the third are ignored. Run-time results are unpredictable.

User Response:

Ensure that only three nested levels of the OCCURS clause are present before recompiling.

**CBL —0216 E INVALID INDEX-NAME IN
OCCURS CLAUSE -- OPTION
IGNORED**

Explanation:

INDEXED BY is not followed by a data name.

Additional Information:

The INDEXED BY phrase is ignored.

User Response:

Ensure that the INDEXED BY phrase is followed by a data name before recompiling.

**CBL —0217 E 77 LEVEL IN FILE SECTION --
ITEM IGNORED**

Explanation:

Level-77 items can appear only in the Working-Storage Section or the Linkage Section.

Additional Information:

The item is ignored.

User Response:

Change the level number from 77 to 01 before recompiling.

**CBL —0218 E CONDITIONAL VARIABLE
MISSING -- CONDITION-NAME
IGNORED**

Explanation:

A condition name (88) had no immediate qualifier.

Additional Information:

The condition name (88) is ignored.

User Response:

Insert an immediate qualifier for the condition name before recompiling.

**CBL —0219 E NO VALUE IN
CONDITION-NAME -- ITEM
IGNORED**

Explanation:

The VALUE clause was omitted from the condition name (88).

Additional Information:

The condition name is ignored.

User Response:

Supply the VALUE clause before recompiling.

**CBL —0220 E INVALID LOCATION OF
WORKING-STORAGE SECTION**

Explanation:

The Working-Storage Section header was either duplicated or preceded by the Linkage Section header.

Additional Information:

The Working-Storage Section header is ignored.

User Response:

Correct invalid statement before recompiling.

**CBL —0221 E INVALID LOCATION OF
LINKAGE SECTION**

Explanation:

The Linkage Section header was duplicated.

Additional Information:

The second Linkage Section header is ignored.

User Response:

Delete second Linkage Section header before recompiling.

**CBL —0222 E REDEFINES OF ITEM WITH
OCCURS CLAUSE -- REDEFINES
IGNORED**

Explanation:

The item being redefined cannot contain an OCCURS clause or be subordinate to an item that contains an OCCURS clause.

Additional Information:

The REDEFINES clause is ignored.

User Response:

Remove the OCCURS clause from the description of the object of the REDEFINES clause, or eliminate the use of the REDEFINES clause before recompiling.

**CBL —0223 E VALUE IN REDEFINES ITEM --
VALUE CLAUSE IGNORED**

Explanation:

The VALUE clause cannot be specified in a data description entry that contains a REDEFINES clause or in an entry that is subordinate to an entry containing a REDEFINES clause.

Additional Information:

The VALUE clause is ignored.

User Response:

Delete the invalid VALUE clause before recompiling.

**CBL —0224 E NON-UNIQUE INDEX-NAME --
IGNORED**

Explanation:

A data name in an INDEXED BY phrase cannot be defined more than once.

Additional Information:

The duplicate index name in the INDEXED BY phrase is ignored.

User Response:

Make the index name unique before recompiling.

**CBL —0225 E/W PRECEDING ITEM [name]
REDEFINES A SMALLER ITEM**

Explanation:

The indicated data item (other than a 01 item) redefines a smaller item.

Additional Information:

Sufficient space is allocated for the larger item. (Both the redefined and redefining fields will have the same left address.)

User Response:

Correct invalid specification before recompiling.

**CBL —0226 E ITEM SIZE EXCEEDS 32,768 --
IGNORED**

Explanation:

An item of over 32,768 bytes was found.

Additional Information:

The item is ignored.

User Response:

Correct item size before recompiling.

**CBL —0227 E INVALID INDICATOR CLAUSE --
IGNORED**

Explanation:

The INDICATOR clause was specified for a data item that is not Boolean.

Additional Information:

The INDICATOR clause is ignored.

User Response:

Correct the invalid specification before recompiling.

**CBL —0228 E NON-UNIQUE FILE-NAME --
FILE IGNORED**

Explanation:

A file description entry contains a file name that was previously defined.

Additional Information:

The file is ignored.

User Response:

Ensure that each file has a unique name before recompiling.

**CBL —0229 C INVALID HIERARCHY --
ACCEPTED**

Explanation:

A level item does not have a preceding level-01 item; a variable-length table is not the last structure within a record description; or a level-66 item is not the last item in a record description.

Additional Information:

The items are accepted.

User Response:

Correct the invalid hierarchy before recompiling.

**CBL —0230 E INVALID OBJECT OF
OCCURS...DEPENDING ON
CLAUSE**

Explanation:

The data description of the object of an OCCURS . . . DEPENDING ON clause does not define an elementary unedited numeric item, or it is defined within the structure of the variable-length table.

Additional Information:

The maximum length of the variable-length table is assumed.

User Response:

Ensure that the description of the variable-length table contains a valid OCCURS...DEPENDING ON clause.

**CBL —0231 E INVALID CLAUSE IN DATA
DIVISION -- FOUND [element]**

Explanation:

The indicated element was found in the Data Division. where its use is invalid.

Additional Information:

The element and all elements up to the next section header. level indicator, or level number are ignored.

User Response:

Move the invalid item to the proper location or delete before recompiling.

**CBL —0232 E INVALID CLAUSE IN FD ENTRY
-- FOUND [element]**

Explanation:

The element found was not one of the following: BLOCK CONTAINS, RECORD CONTAINS, LABEL RECORDS.

LINAGE, CODE-SET, DATA RECORDS, or VALUE OF clause; a period, or a space.

Additional Information:

The element is ignored, and the remainder of the FD is processed.

User Response:

Delete invalid element before recompiling.

**CBL —0233 E VALUE DEFINED IN ITEM WITH
GROUP VALUE -- GROUP
VALUE ASSUMED**

Explanation:

If the VALUE clause is specified in an entry at the group level, it cannot be specified at subordinate levels within this group.

Additional Information:

The group value is processed, and the lower level is ignored.

User Response:

Before recompiling, ensure that a VALUE clause does not appear both on the group level and on a level subordinate to the group level.

**CBL —0234 E INVALID INDICATOR VALUE --
CLAUSE IGNORED**

Explanation:

The value for an INDICATOR clause was not an integer between 1 and 99, inclusive.

Additional Information:

The INDICATOR clause is ignored.

User Response:

Correct the incorrect value before recompiling.

**CBL —0235 E VALUE CLAUSE INVALID IN
SECTION -- IGNORED**

Explanation:

A VALUE clause was specified for a nonlevel-88 item in the File Section or Linkage Section.

Additional Information:

The VALUE clause is ignored.

User Response:

Correct invalid specification before recompiling.

**CBL —0236 E NO PICTURE IN PRECEDING
ELEMENTARY ITEM**

Explanation:

A higher-level item was found following a lower-level item with no PICTURE clause.

Additional Information:

The compiler substitutes a dummy item for the invalid item. The results at run time are unpredictable.

User Response:

Provide the PICTURE clause before recompiling.

**CBL —0237 E INVALID CHARACTER IN
PICTURE -- ITEM IGNORED**

Explanation:

The categories of data that can be described with a PICTURE clause are made up of certain combinations of characters in the COBOL character set. A character that is not allowed was found in the PICTURE clause.

Additional Information:

The data item is ignored.

User Response:

Correct the invalid PICTURE clause before recompiling.

**CBL —0238 E USAGE AND PICTURE NOT
COMPATIBLE -- USAGE
IGNORED**

Explanation:

An alphabetic, alphanumeric, or numeric edited PICTURE clause has a usage other than DISPLAY.

Additional Information:

The USAGE clause is ignored.

User Response:

Specify USAGE IS DISPLAY before recompiling.

**CBL —0239 E GROUP CONTAINS PICTURE
CLAUSE -- GROUP PICTURE
IGNORED**

Explanation:

A group data item cannot have a PICTURE clause.

Additional Information:

The PICTURE clause is ignored in the group data item.

User Response:

Correct invalid specification before recompiling.

**CBL —0240 E INVALID VALUE IN GROUP
ITEM**

Explanation:

A numeric literal was found as the value for a group item.

Additional Information:

The VALUE clause is dropped, and the group item is not initialized.

User Response:

Ensure that the value specified is a figurative constant or a nonnumeric literal before recompiling.

**CBL —0241 E ALPHANUMERIC VALUE
EXCEEDS ITEM SIZE -- VALUE
CLAUSE IGNORED**

Explanation:

The nonnumeric literal in the VALUE clause must not exceed the size of the alphanumeric item.

Additional Information:

The VALUE clause is ignored, and the data item is not initialized.

User Response:

Ensure that the literal in the VALUE clause is less than or equal to the item size before recompiling.

**CBL —0242 E FILE NAME APPEARS ON
MORE THAN ONE SAME AREA
CLAUSE**

Explanation:

One or more of the file names appearing on this SAME AREA clause appeared in at least one previous SAME AREA clause.

Additional Information:

This clause is ignored.

User Response:

Correct the invalid file name(s) before recompiling.

**CBL —0243 E FILE NAME APPEARS ON
MORE THAN ONE SAME
RECORD AREA CLAUSE**

Explanation:

One or more of the file names appearing in this SAME RECORD AREA clause appeared on at least one previous SAME RECORD AREA clause.

Additional Information:

This clause is ignored.

User Response:

Correct the invalid file name(s) before recompiling.

**CBL —0244 E FILE NAME(S) APPEARING ON
SAME AREA CLAUSES DO NOT
ALL APPEAR ON SAME
RECORD AREA CLAUSE**

Explanation:

If one file from a SAME AREA clause appears on a SAME RECORD AREA clause, all files named on the SAME AREA clause must appear on the SAME RECORD AREA clause.

Additional Information:

This clause is ignored.

User Response:

Correct the invalid file name(s) before recompiling.

**CBL —0245 E DUPLICATED CLAUSE IN DATA
DESCRIPTION -- FIRST ONE
ACCEPTED**

Explanation:

The same clause occurred more than once in the same data description.

Additional Information:

The first clause is processed, and all duplicate entries are ignored.

User Response:

Delete duplications before recompiling.

**CBL —0246 E USAGE IN ITEM NOT SAME AS
GROUP USAGE -- GROUP
USAGE USED**

Explanation:

The usage of an elementary item cannot contradict the usage of a group to which the elementary item belongs.

Additional Information:

The group usage overrides the elementary usage.

User Response:

Correct the usage of the elementary item before recompiling.

**CBL —0247 E LEVEL ITEMS IN FILE SECTION
-- NO PRECEDING FD OR SD**

Explanation:

A file description entry must precede the first level-01 data item in the File Section.

Additional Information:

Space is allocated for the level-01 item, but there is no corresponding file for I/O verbs.

User Response:

Insert a valid file description (FD or SD) before the level-01 data item, or remove the level-01 item before recompiling.

**CBL —0248 E JUSTIFIED OR USAGE CLAUSE
INVALID IN GROUP ITEM WITH
VALUE -- CLAUSE IGNORED**

Explanation:

A VALUE clause cannot be specified for a group containing items with descriptions including a JUSTIFIED or USAGE clause.

Additional Information:

The JUSTIFIED or USAGE clause is ignored.

User Response:

Delete the JUSTIFIED, USAGE, or VALUE clause before recompiling.

**CBL —0249 E VALUE IN PRECEDING GROUP
ITEM EXCEEDS GROUP ITEM
SIZE**

Explanation:

The literal specified in the VALUE clause in a group item is too large.

Additional Information:

The VALUE clause is dropped, and the group item is not initialized.

User Response:

Ensure that the length of the literal is less than or equal to the size of the group item before recompiling.

**CBL —0250 E KEY CLAUSE ON BOOLEAN
ITEM -- CLAUSE IGNORED**

Explanation:

ASCENDING/DESCENDING KEY clause was specified for a Boolean data item.

Additional Information:

The ASCENDING/DESCENDING KEY clause is ignored.

User Response:

Remove the invalid clause before recompiling.

**CBL —0252 C INVALID DATA DIVISION
HEADER -- ACCEPTED**

Explanation:

The word DIVISION or a period and a space were omitted or misplaced in the Data Division header.

Additional Information:

The Data Division header is assumed to be correct.

User Response:

Supply the proper Data Division specification before compiling.

**CBL —0253 C COMPUTED RECORD SIZE NOT
EQUAL TO DEFINED RECORD
SIZE**

Explanation:

The RECORD CONTAINS clause in the FD or SD entry contains an integer larger than the largest level-01 record definition or smaller than the smallest level-01 record definition for the file.

Additional Information:

The clause is treated as comments.

User Response:

Correct the clause before recompiling.

**CBL —0254 C INVALID WORD/CHARACTER(S)
[element] IN CLAUSE -- IGNORED**

Explanation:

An invalid clause was found in a level item.

Additional Information:

The compiler skips to the next key word or level item.

User Response:

Delete the invalid element before recompiling.

**CBL —0255 C INVALID USAGE CLAUSE --
DISPLAY ASSUMED**

Explanation:

The USAGE clause must contain one of the following key words: DISPLAY, COMPUTATIONAL(COMP), COMPUTATIONAL-3(COMP-3), COMPUTATIONAL-4(COMP-4), or INDEX.

Additional Information:

The USAGE is assumed to be DISPLAY.

CBL —0256 C INVALID USE OF BLANK WHEN ZERO CLAUSE -- IGNORED

Explanation:

The BLANK WHEN ZERO clause can be specified only at the elementary level for numeric edited or numeric items; it cannot be specified for level-88 data items and cannot be used with * in the PICTURE clause.

Additional Information:

The BLANK WHEN ZERO clause is ignored.

User Response:

Delete the invalid BLANK WHEN ZERO clause before recompiling.

CBL —0257 C INVALID USE OF JUSTIFIED CLAUSE -- IGNORED

Explanation:

The JUSTIFIED clause can only be specified at the elementary level for nonnumeric unedited items.

Additional Information:

The JUSTIFIED clause is ignored.

User Response:

Remove the invalid JUSTIFIED clause before recompiling.

CBL —0258 C INVALID CLAUSE IN 88 LEVEL ITEM -- IGNORED

Explanation:

A clause other than VALUE was specified for a level-88 data item.

Additional Information:

The invalid clause is ignored.

User Response:

Delete the invalid clause from level-88 item before recompiling.

CBL —0259 C INVALID CLAUSE IN INDEX DATA ITEM -- IGNORED

Explanation:

JUSTIFIED, BLANK, PICTURE, and VALUE clauses cannot be specified for a data item with USAGE INDEX.

Additional Information:

The invalid clause is ignored.

User Response:

Delete the invalid clause before recompiling.

CBL —0260 C REDEFINES OF ITEM WITH REDEFINES CLAUSE -- REDEFINES ACCEPTED

Explanation:

The object of the REDEFINES clause contains a REDEFINES clause.

Additional Information:

Both REDEFINES clauses are accepted.

User Response:

Ensure that the object of both REDEFINES clause is the same data name before recompiling.

CBL —0261 C OCCURS IN LEVEL 01 OR 77 ITEM -- ACCEPTED

Explanation:

The OCCURS clause cannot be specified in a data description that is a level-01 or level-77 number.

Additional Information:

The OCCURS clause is accepted.

User Response:

Correct the level number before recompiling.

CBL —0262 C REDEFINES IN 01 LEVEL IN FILE SECTION -- IGNORED

Explanation:

The REDEFINES clause should not be used in level-01 entries in the File Section, because multiple level-01's are an implicit redefinition.

Additional Information:

The REDEFINES clause is ignored.

User Response:

Delete the REDEFINES clause before recompiling.

CBL —0263 C INVALID CLAUSE IN LEVEL-66 ITEM -- CLAUSE IGNORED

Explanation:

A clause other than RENAMES was specified for a level-66 data item.

Additional Information:

The invalid clause is ignored.

User Response:

Delete invalid clause from level-66 item before recompiling.

**CBL —0264 E INVALID RENAMES CLAUSE --
CLAUSE IGNORED**

Explanation:

The key word RENAMES or THROUGH was not followed by a data name, the second data name (if present) was the same as the first, or the RENAMES clause was specified for an item whose level is other than 66.

Additional Information:

The clause is ignored.

User Response:

Correct invalid clause before recompiling.

**CBL —0265 E NO RENAMES IN LEVEL-66
ITEM -- ITEM IGNORED**

Explanation:

The RENAMES clause was omitted from the level-66 item.

Additional Information:

The item is ignored.

User Response:

Supply the RENAMES clause before recompiling.

**CBL —0266 E INVALID OBJECT OF RENAMES
CLAUSE -- ITEM IGNORED**

Explanation:

The object of the RENAMES clause is not an item or valid range of items within the immediately preceding record description, it describes an item whose length is variable, or the renamed items contain or are subordinate to an occurs clause or are levels-01, -66, -77, or -88.

Additional Information:

The entire level-66 item is ignored.

User Response:

Correct the invalid specification before recompiling.

**CBL —0267 C SECTION IS MISSING IN
SECTION HEADER --
ACCEPTED**

Explanation:

The word SECTION is missing from the File Section, Working-Storage Section, or Linkage Section header.

Additional Information:

The word SECTION is assumed to be present.

**CBL —0268 C INVALID BLOCK CLAUSE --
IGNORED**

Explanation:

A syntax error was found in the BLOCK CONTAINS clause. The integer is missing, misplaced, too large, or specified for a TRANSACTION file.

Additional Information:

The clause is ignored.

User Response:

Correct the syntax of the clause before recompiling.

**CBL —0269 C INVALID RECORD CLAUSE --
IGNORED**

Explanation:

A syntax error was found in the RECORD CONTAINS clause. The integer is either missing or misplaced.

Additional Information:

The clause is ignored.

User Response:

Correct the syntax of the clause before recompiling.

**CBL —0270 C LABEL CLAUSE FOR FILE
INVALID OR MISSING**

Explanation:

OMITTED labels were specified for a disk file, or the LABEL RECORDS clause was omitted for the FD entry.

Additional Information:

The file name is processed as if the clause were specified correctly.

User Response:

Correct the clause before recompiling.

**CBL —0271 C INVALID
ASCENDING/DESCENDING KEY
PHRASE IN OCCURS CLAUSE --
PHRASE IGNORED**

Explanation:

No data name was found, a qualifier connective (IN or OF) was not followed by a data name, or an invalid key was specified.

The first key must be either the subject of the OCCURS clause or subordinate to the subject; all following keys must be subordinate to the subject. Any key that is not the subject of the OCCURS

clause must not contain or be subordinate to another OCCURS clause.

Additional Information:

The ASCENDING/DESCENDING phrase is ignored.

User Response:

Correct the invalid phrase before recompiling.

CBL —0273 C DUPLICATE CLAUSE IN FILE DESCRIPTION - FIRST ONE USED

Explanation:

More than one specification for a clause was found in the file description entry.

Additional Information:

The first clause is used, and duplicate entries are ignored.

User Response:

Delete duplications before recompiling.

CBL —0275 C INVALID NUMERIC LITERAL IN VALUE CLAUSE FOR A SCALED ITEM -- ZEROS WERE ASSUMED IN THE SCALED POSITIONS.

Explanation:

The numeric literal in the VALUE clause is not within the range indicated by the character string of the PICTURE clause.

Additional Information:

Zeros are assumed in the scaled positions in the character string of the PICTURE clause.

User Response:

The PICTURE clause character string or the numeric literal must be corrected before recompiling.

CBL —0280 E/W PRECEDING ITEM [name] REDEFINES A LARGER ITEM

Explanation:

The indicated data item (other than a level-01 item) redefines a larger item.

A W level message occurs when the redefined item is a level-01; otherwise, an E level message occurs.

Additional Information:

The difference in the sizes of the two items is allocated following the smaller item. (Both the redefined and redefining fields will have the same left address.)

User Response:

Increase the size of the smaller item before recompiling.

CBL —0285 E NO VALID SECTION PRECEDING DATA ITEM -- WORKING-STORAGE ASSUMED

Explanation:

A level item was encountered immediately following the Data Division header.

Additional Information:

The item is assumed to be in the Working-Storage Section.

User Response:

Provide a valid section header before recompiling.

CBL —0286 E DATA ITEM AFTER FD OR SD NOT LEVEL 01 -- FILE IGNORED

Explanation:

The first data item following an FD or SD entry must have a level-01 entry.

Additional Information:

The file is ignored.

User Response:

Ensure that the FD or SD entry is followed by a level-01 record description before recompiling.

CBL —0287 E REQUIRED KEYWORDS(S) MISSING FROM SIGN CLAUSE -- CLAUSE IGNORED

Explanation:

The SIGN clause did not contain the words LEADING or TRAILING, or SEPARATE was not found with the key word CHARACTER.

Additional Information:

The SIGN clause is ignored.

User Response:

Correct the SIGN clause before recompiling.

CBL —0288 E SIGN CLAUSE FOR NON-NUMERIC DISPLAY ELEMENTARY ITEM -- CLAUSE IGNORED

Explanation:

A SIGN clause is specified for an elementary item that is not a numeric display item.

Additional Information:

The SIGN clause is ignored.

User Response:

Specify USAGE IS DISPLAY before recompiling.

**CBL —0289 E SIGN CLAUSE INVALID FOR
ITEM WITH UNSIGNED
PICTURE -- SIGN CLAUSE
IGNORED**

Explanation:

The SIGN clause can be specified only for a numeric data description entry whose PICTURE clause contains the character S or for a group item containing at least one numeric data description entry.

Additional Information:

The SIGN clause is ignored.

User Response:

Correct the PICTURE or SIGN clause before recompiling.

**CBL —0290 E SIGN CLAUSE IN ELEMENTARY
ITEM DIFFERS FROM SIGN OF
GROUP -- GROUP SIGN USED**

Explanation:

If a group item contains a SIGN clause, any SIGN clauses within the group must be the same as the group SIGN clause. The level items in the group do not have to have a SIGN clause.

Additional Information:

The group sign is used.

User Response:

Correct invalid specification before recompiling.

**CBL —0291 W TWO SFGR LOAD MEMBERS
OR IDDU FORMAT FILES
SPECIFIED -- SECOND
IGNORED**

Explanation:

Only one IDDU format file and one SFGR load member can be specified in an ASSIGN clause.

Additional Information:

All additional SFGR load members containing screen formats after the first load member specified are ignored; and all

additional IDDU format files containing communication formats after the first IDDU format file specified are ignored.

User Response:

Remove all additional SFGR load members or IDDU format files so only one of each type is specified before recompiling.

**CBL —0292 C INVALID INTEGER OR
DATA-NAME IN LINAGE
CLAUSE**

Explanation:

The LINAGE value integer was equal to 0. If the name of a variable was specified instead of a numeric literal, the variable either is not in the Data Division or is invalid.

Additional Information:

The compiler allows this value to default to the system value for the number of lines per page at execution time.

User Response:

Correct the invalid specification before recompiling.

**CBL —0293 C LINAGE CLAUSE SPECIFIED
FOR INVALID DEVICE --
IGNORED**

Explanation:

The LINAGE clause can only be specified for a printer.

Additional Information:

The clause is ignored.

User Response:

Correct the invalid specification before recompiling.

**CBL —0294 W IDDU FILE USED BUT NO
DATA DICTIONARY SPECIFIED
-- IDDU FILE IGNORED**

Explanation:

An IDDU format file containing communication formats for use by a TRANSACTION file has been specified without specifying the DATA DICTIONARY name where the IDDU format file is located.

Additional Information:

When using IDDU format files, both the IDDU format file name and the DATA DICTIONARY name must be specified. If the DATA DICTIONARY name is not specified in the COBOLC or COBOLONL procedures, the IDDU format file name is ignored.

User Response:

A DATA DICTIONARY name must be specified, or the IDDU format file name corrected, before recompiling.

**CBL —0295 W PICTURE UNSIGNED AND
VALUE SIGNED -- SIGN
IGNORED**

Explanation:

The literal following the VALUE clause should not be signed if the PICTURE clause is unsigned.

Additional Information:

The sign is ignored.

User Response:

Correct the invalid sign before recompiling.

**CBL —0296 C INVALID INTEGER OR
DATA-NAME IN FOOTING
CLAUSE**

Explanation:

The FOOTING clause has a syntax error, the integer value specified is greater than the LINAGE clause value, or the value is equal to 0. If the name of a variable was specified instead of a numeric literal, the variable either is not in the Data Division or is of an invalid type.

Additional Information:

The FOOTING value is assumed to be equal to the LINAGE value.

User Response:

Correct the invalid specification before recompiling.

**CBL —0297 C INVALID INTEGER OR
DATA-NAME IN TOP CLAUSE**

Explanation:

The TOP clause has a syntax error. If the name of a variable was specified instead of a numeric literal, the variable either is not in the Data Division or is invalid.

Additional Information:

The TOP value is assumed to be 0.

User Response:

Correct the invalid specification before recompiling.

**CBL —0298 C INVALID INTEGER OR
DATA-NAME IN BOTTOM
CLAUSE**

Explanation:

The BOTTOM clause has a syntax error. If the name of a variable was specified instead of a numeric literal, the variable either is not in the Data Division or is invalid.

Additional Information:

The BOTTOM clause value is assumed to be 0.

User Response:

Correct the invalid specification before recompiling.

**CBL —0300 E [name] IS DEFINED AS BOTH
PARAGRAPH AND SECTION**

Explanation:

The indicated name appears as both a paragraph and a section header.

Additional Information:

The duplicate header is ignored. All procedure references to the name are made to the first procedure header.

User Response:

Correct or delete the duplicated name before recompiling.

CBL —0301 E DUPLICATE SECTION [name]

Explanation:

The indicated section name element is duplicated in the program.

Additional Information:

The duplicate section header is ignored. All references to the name will be resolved to the first appearance of the section header.

User Response:

Correct or delete the duplicated name before recompiling.

**CBL —0302 E DUPLICATE PARAGRAPH [name]
IN SECTION**

Explanation:

Two or more paragraphs within the same Procedure Division section have the same name.

Additional Information:

The duplicate paragraph name is ignored. All references to the name will be resolved to the first appearance of the paragraph.

User Response:

Correct or delete the duplicated name before recompiling.

**CBL —0303 E INVALID QUALIFICATION OF
[name or key word]**

Explanation:

The indicated name is improperly qualified.

Additional Information:

The compiler substitutes a dummy item for the invalid term, and processing continues. Incorrect code, however, will be generated for the statement.

User Response:

Correct the indicated element before recompiling.

CBL —0304 E INSUFFICIENT OR MISSING QUALIFICATION OF [name or key word]

Explanation:

The indicated name is not unique.

Additional Information:

The compiler substitutes a dummy item for the invalid item, and processing continues. Incorrect code, however, will be generated for the statement.

User Response:

Supply the missing qualification for the indicated element before recompiling.

CBL —0305 E LEVEL OF QUALIFICATION EXCEEDS LIMIT

Explanation:

Only 51 levels of qualification are allowed for a data name, and only 2 levels are allowed for a procedure name.

Additional Information:

The compiler substitutes a dummy item for the invalid item, and processing continues. Invalid code, however, will be generated for the statement.

User Response:

Ensure that the level of qualifications does not exceed the limit before recompiling.

CBL —0306 E [name] IS INVALID PARAGRAPH TO ALTER

Explanation:

A paragraph referenced by an ALTER statement must contain only a single GO TO statement.

Additional Information:

The ALTER statement is not generated.

User Response:

Correct the invalid specification before recompiling.

CBL —0307 C PERIOD REQUIRED -- PERIOD ASSUMED

Explanation:

The last statement preceding a paragraph does not end with a period.

Additional Information:

The period is assumed to be present.

User Response:

Supply the required period before recompiling.

CBL —0308 W PROCEDURE DIVISION HEADER ERROR

Explanation:

The Procedure Division must begin with the header PROCEDURE DIVISION followed by a period or a USING phrase. This message also appears if the statement following the Procedure Division header is invalid.

Additional Information:

The key words PROCEDURE and DIVISION are assumed to be correct. The USING phrase, if present, is processed as if the two key words were correct.

User Response:

Correct the error in the header before recompiling.

CBL —0309 W EXIT MUST BE ONLY STATEMENT IN PARAGRAPH

Explanation:

A paragraph contains one or more statements in addition to the EXIT statement.

Additional Information:

Code is generated as if the EXIT statement were correct.

User Response:

Correct error in paragraph before recompiling.

CBL —0310 E TOO MANY SUBSCRIPTS OR INDEXES SPECIFIED FOR [name]

Explanation:

The number of subscript or index specifications exceeds the level of occurrences of the name.

Additional Information:

The subscript or index values are assumed to be 1.

User Response:

Correct the subscript or index specification before recompiling.

CBL —0311 E TOO FEW SUBSCRIPTS OR INDEXES SPECIFIED FOR [name or key word]

Explanation:

The number of occurrences for the name exceeds the number of subscripts or indexes specified.

Additional Information:

The subscript or index values are assumed to be 1.

User Response:

Correct the subscript or index specification before recompiling.

CBL —0313 E MISSING RIGHT PARENTHESIS -- STATEMENT IGNORED

Explanation:

Each left parenthesis must be paired with a right parenthesis.

Additional Information:

The statement is dropped with no further syntax checking.

User Response:

Supply the missing parenthesis before recompiling.

CBL —0314 E LINKAGE SECTION NAME [name] NOT IN USING STATEMENT -- IGNORED

Explanation:

The indicated item is level-01 (or level-77) or is subordinate to a level-01 data description in the Linkage Section whose name does not appear in the USING phrase of the Procedure Division header.

Additional Information:

The compiler substitutes a dummy item for the invalid item, and processing continues. Incorrect code, however, will be generated for the statement.

User Response:

Correct the USING phrase before recompiling.

CBL —0315 E [name or key word] INVALID INDEXED OR SUBSCRIPTED ITEM

Explanation:

Either the indicated element is not a data name or its data description does not have an OCCURS clause.

Additional Information:

The expression within the parentheses following the element is discarded.

User Response:

Correct the invalid specification before recompiling.

CBL —0316 E [name or literal] NOT NUMERIC -- FOUND IN ARITHMETIC EXPRESSION

Explanation:

The indicated element is neither a numeric elementary item nor a numeric literal.

Additional Information:

The expression is ignored.

User Response:

Replace the indicated element with a numeric item before recompiling.

CBL —0320 E MISSING LEFT PARENTHESIS -- STATEMENT IGNORED

Explanation:

A right parenthesis was found without a corresponding left parenthesis.

Additional Information:

The statement is dropped with no further syntax checking.

User Response:

Supply the missing parenthesis before recompiling.

CBL —0322 E [name] NOT A GROUP ITEM -- STATEMENT IGNORED

Explanation:

Each identifier in an ADD/SUBTRACT/MOVE statement with the CORRESPONDING phrase must identify a group item.

Additional Information:

The statement is dropped with no further processing.

User Response:

Ensure that both items are group items.

CBL —0323 W NO CORRESPONDING ITEMS FOUND - STATEMENT IGNORED

Explanation:

No corresponding items were found for the group items specified in an ADD/SUBTRACT/MOVE statement with the CORRESPONDING phrase.

Additional Information:

The statement is dropped.

User Response:

Probable user error. Ensure that there is at least one corresponding item pair.

**CBL —0324 E INVALID OR MISSING
KEYWORDS IN ALTER -- TO
ASSUMED**

Explanation:

An ALTER statement was found with a missing or incorrect key word(s) following the first procedure name of a pair.

Additional Information:

The key word TO is assumed and replaces all key words present.

User Response:

Correct the invalid statement before recompiling.

**CBL —0325 E [name or key word] INVALID AT
THIS POINT**

Explanation:

An unexpected element was encountered.

Additional Information:

The compiler ignores the invalid element and any associated operands. Incorrect code will be generated.

User Response:

Correct the indicated element before recompiling.

CBL —0326 E [insert] IS NOT A DEFINED NAME

Explanation:

An undefined name was found.

Additional Information:

The compiler substitutes a dummy item for the undefined name, and processing continues. Incorrect code, however, will be generated for the statement.

User Response:

Supply a valid name before recompiling.

**CBL —0328 C MISSING OR INVALID LITERAL
IN INDEX SPECIFICATION --
FIRST OCCURRENCE USED**

Explanation:

An index specification contains a plus or minus sign that is not followed by a valid numeric literal.

Additional Information:

All index values are assumed to be 1.

User Response:

Correct the invalid literal before recompiling.

**CBL —0329 E SUBSCRIPT OR INDEX LITERAL
NOT INTEGER**

Explanation:

A literal was not a positive integer in one of the following cases:

- Within an index or subscript specification
- Within a SET statement
- Within a VARYING phrase of a PERFORM statement that references an index name
- Within a relational condition referencing an index name

Additional Information:

If the error occurred within an index or a subscript specification, all indexes or subscripts are assumed to have a value of 1. If it occurred elsewhere, a value of 1 is substituted for the invalid literal.

User Response:

Ensure that the literal is a positive integer before recompiling.

**CBL —0330 E INVALID LITERAL IN INDEX OR
SUBSCRIPT**

Explanation:

In one of the following cases, the subscript or index literal either exceeds 32,768, is negative, or is 0:

- Within an index or subscript specification
- Within a SET statement
- Within a VARYING phrase of a PERFORM statement that references an index name
- Within a relational condition referencing an index name

Additional Information:

If the error occurred within an index or subscript specification, all subscripts or indexes are assumed to have a value of 1. If it occurred elsewhere and if the literal was greater than 32,768, a value of 1 is substituted. If it occurred elsewhere and the literal was negative, its absolute value is substituted.

User Response:

Correct the invalid literal before recompiling.

CBL —0331 E [name] IS NOT DEFINED IN THE DATA DIVISION

Explanation:

The identifier was not defined.

Additional Information:

The compiler substitutes a dummy item for the invalid item, and processing continues. However, incorrect code will be generated for the statement.

User Response:

Supply the valid name before recompiling.

CBL —0332 E [name or key word] IS NOT A VALID SUBSCRIPT

Explanation:

The identifier used as a subscript either is a nonnumeric data name or it requires subscripting itself.

Additional Information:

All subscripts or indexes are assumed to have a value of 1.

User Response:

Supply the subscript item before recompiling.

CBL —0333 E NAME NOT DEFINED AS INDEXED BY [name]

Explanation:

The indicated name and index belong to different tables. A name can be indexed only by index names of the table to which it belongs. In addition, the index names must be in the proper order within the index specifications.

Additional Information:

The value of all subscripts and indexes is assumed to be 1.

User Response:

Correct invalid index specification before recompiling.

CBL —0334 E NO INDEX DEFINED FOR TABLE IN SEARCH - STATEMENT IGNORED

Explanation:

The Data Division description of the table to be searched must contain the INDEXED BY phrase.

Additional Information:

The statement is dropped.

User Response:

Ensure that the Data Division description of the table to be searched contains the INDEXED BY phrase.

CBL —0335 E KEY SPECIFIED ON START STATEMENT IS INVALID

Explanation:

An incorrect key is specified on the START statement for the indexed or direct file. The left position of the START KEY data-name may not coincide with the left position of the RECORD KEY data-name, or the length of the key is greater than the length of the RECORD KEY data-name and an invalid partial key has been defined.

Additional Information:

Remainder of the clause is ignored.

User Response:

Correct the invalid key before recompiling.

CBL —0337 C TABLE [name] IN SEARCH STATEMENT HAS SUBSCRIPT/INDEX -- SUBSCRIPT/INDEX IGNORED

Explanation:

The name of the table to be searched must not be subscripted or indexed.

Additional Information:

The subscript/index(s) is dropped.

User Response:

Ensure that the table specified in the SEARCH statement is not subscripted or indexed.

CBL —0338 E [name] IS BOTH DATA-NAME AND PROCEDURE-NAME

Explanation:

The indicated name is defined in both the Data Division and Procedure Division.

Additional Information:

The compiler substitutes a dummy item for the invalid item, and processing continues. Incorrect code, however, will be generated for the statement.

User Response:

Correct the indicated item before recompiling.

**CBL —0339 C [name] CONTAINS AMBIGUOUS
INDICATOR ASSOCIATION
VALUES**

Explanation:

The INDICATOR data item specified in the WRITE statement is a group item that contains an indicator definition subordinate to an item that contains an OCCURS clause.

Additional Information:

The INDICATOR data item is accepted. Indicators defined subordinate to an item with an OCCURS clause are assigned to their first occurrence in the table.

User Response:

Ensure that any table subordinate to the INDICATOR data item does not contain mixed indicator and nonindicator elementary items.

**CBL —0340 E ALL FOLLOWED BY INVALID
LITERAL**

Explanation:

The literal following ALL must be either a nonnumeric literal or a figurative constant.

Additional Information:

If the literal is numeric, the ALL is dropped.

User Response:

Correct the invalid literal before recompiling.

**CBL —0341 E INVALID 'ALL' LITERAL - 'ALL'
IGNORED**

Explanation:

The figurative constant ALL literal is invalid in a DISPLAY, STRING, or STOP statement.

Additional Information:

The key word ALL is dropped.

User Response:

Ensure that the figurative constant ALL literal is not specified in a DISPLAY, STRING, or STOP statement.

**CBL —0342 C ILLEGAL OPERATOR IN
SEARCH ALL..WHEN PHRASE --
ACCEPTED**

Explanation:

The conditional operators EQUAL and AND are the only conditional operators permitted in the WHEN phrase of a format 2 SEARCH statement.

Additional Information:

Code is generated as if the invalid operators were valid.

User Response:

Ensure that the WHEN phrase of the format 2 SEARCH statement contains no conditional operators except EQUAL or AND.

**CBL —0343 C MULTIPLE VALUE
CONDITION-NAME IN SEARCH
ALL..WHEN PHRASE --
ACCEPTED**

Explanation:

In format 2 of the SEARCH statement, condition names must be defined as having only a single value.

Additional Information:

The multiple value condition name is accepted. Code is generated to test the multiple values.

User Response:

Ensure that all referenced condition names are defined as having only a single value.

**CBL —0344 E INVALID NAME OR LITERAL
ENTRY IN USING CLAUSE**

Explanation:

A parameter following the USING clause was not a name, was a name that did not appear in the Linkage Section, or was not a level-01 or -77 item for a Procedure Division header.

Additional Information:

If the parameter was not a name, the remainder of the USING clause is ignored. If the name did not appear in the Linkage Section, or if the name was not the right level item, that name is ignored.

User Response:

Correct the invalid entry before recompiling.

**CBL —0345 C NO SECTION-NAME AFTER
PROCEDURE DIVISION
HEADER -- ACCEPTED**

Explanation:

If sections are used within the Procedure Division, a section header must immediately follow the Procedure Division header.

Additional Information:

None.

User Response:

Ensure that the section header precedes the first paragraph of the Procedure Division before recompiling.

CBL —0347 E NUMBER OF LINKAGE NAMES EXCEEDS 15

Explanation:

A maximum of 15 parameters may appear in the USING clause.

Additional Information:

All parameters after the first 15 are ignored.

User Response:

Ensure that not more than 15 parameters appear before recompiling.

CBL —0348 E INVALID ELEMENT AFTER PROCEDURE HEADER -- IGNORED

Explanation:

Extraneous elements were found following a paragraph or section header.

Additional Information:

The extraneous element or elements are ignored. Processing continues at the next statement.

User Response:

Correct the paragraph header, the section header, or the incorrect elements before recompiling.

CBL —0349 E DATA-NAME OR LITERAL EXPECTED -- FOUND [insert]

Explanation:

The indicated element was found where a data name or literal was required, either in a subscript or index specification or in a SET statement.

Additional Information:

If the error occurred inside the parentheses for a subscript or index item, the value of all subscript or index items is assumed to be 1. If the error occurred in a SET statement, the compiler substitutes a dummy element, and processing continues. Incorrect code, however, will be generated for the statement.

User Response:

Replace the indicated element with an appropriate data name or literal before recompiling.

CBL —0350 C INVALID OR MULTIPLE NOT(S) IN CONDITIONAL EXPRESSION -- IGNORED

Explanation:

The word NOT cannot precede a condition if the condition itself contains a NOT.

Additional Information:

Both NOTs are ignored.

User Response:

Ensure that multiple NOTs do not appear in the conditional expression before recompiling.

CBL —0351 E INVALID ABBREVIATED COMBINED RELATION CONDITION -- STATEMENT IGNORED

Explanation:

The compiler is unable to process the conditional expression because it is neither a complete conditional expression nor a valid abbreviated combined relation condition.

Additional Information:

The statement is dropped.

User Response:

Ensure that the conditional expression is a complete conditional expression or a valid abbreviated combined relation condition.

CBL —0352 E MIXED INDEXES AND SUBSCRIPTS IN [name] -- FIRST OCCURRENCE USED

Explanation:

The indicated item is invalid because subscripting and indexing must not be used together in a single reference.

Additional Information:

The value of all subscripts or indexes is assumed to be 1.

User Response:

Specify either all subscripts or all indexes for the indicated item before recompiling.

CBL —0353 C INVALID OBJECT OF OCCURS...DEPENDING ON ASSOCIATED WITH [name] -- MAXIMUM SIZE ASSUMED

Explanation:

The object of the OCCURS...DEPENDING ON clause is invalid. (Message CBL-0230 may precede this message. See message CBL-0230 for additional information.)

Additional Information:

The maximum length of the variable-length table is assumed.

User Response:

Ensure that the description of the variable-length table contains a valid OCCURS...DEPENDING ON clause.

CBL —0354 C NO PARAGRAPH NAME AFTER SECTION OR DIVISION HEADER -- ACCEPTED

Explanation:

A paragraph name should follow the section or the Procedure Division header.

Additional Information:

If no section or paragraph name follows the Procedure Division header the statements preceding the first section or paragraph name will not be run.

User Response:

Supply the paragraph name before recompiling.

CBL —0355 E TOO MANY ERRORS IN STATEMENT -- NOT ALL SYNTAX CHECKED

Explanation:

The statement contains so many errors that syntax checking is suspended.

Additional Information:

The entire statement is ignored.

User Response:

Correct the syntax of the indicated statement before recompiling.

CBL —0359 C [name] IS NOT A FILE-NAME -- NAME IGNORED

Explanation:

The name specified in the USE ON EXCEPTION/ERROR sentence is not a file name.

Additional Information:

The name is ignored. If more than one name was specified, the other names are processed in the normal manner.

User Response:

Ensure that all names specified in the USE ON EXCEPTION/ERROR sentence are file names.

CBL —0360 E [name] SPECIFIED MORE THAN ONCE -- IGNORED

Explanation:

For any given file, only one EXCEPTION/ERROR procedure can be specified.

Additional Information:

The file name is ignored.

User Response:

Ensure that for any given file, only one EXCEPTION/ERROR procedure is specified.

CBL —0361 E NO SECTION PRECEDING DECLARATIVE SENTENCE -- SENTENCE IGNORED

Explanation:

Each declarative sentence must directly follow a section header in the Declaratives Section.

Additional Information:

The declarative sentence is ignored.

User Response:

Ensure that each declarative sentence is directly preceded by a section header.

CBL —0362 C INVALID USE SENTENCE, NO CONTROLS SPECIFIED -- SENTENCE IGNORED

Explanation:

A required key word or file name was not specified in the USE ON EXCEPTION/ERROR sentence.

Additional Information:

The USE sentence is ignored.

User Response:

Ensure that either a file name or a key word is specified in the USE ON EXCEPTION/ERROR sentence.

CBL —0363 E INVALID OR MISSING SEGMENT NUMBER -- USING PREVIOUS VALID SEGMENT NUMBER

Explanation:

The segment number must be an integer ranging in value from 0 through 99.

Additional Information:

The segment number of the previous section is assumed.

User Response:

Supply a valid segment number before recompiling.

**CBL —0366 W SEGMENT NUMBER SIGNED --
SIGN IGNORED**

Explanation:

The segment number must be unsigned.

Additional Information:

The segment number is assumed to be unsigned.

User Response:

Provide an unsigned segment number before recompiling.

**CBL —0368 E INVALID SYNTAX --
STATEMENT DROPPED**

Explanation:

The compiler was attempting to process a conditional or arithmetic expression when an invalid sequence of operands or operators was encountered.

Additional Information:

The statement is ignored.

User Response:

Correct the invalid syntax before recompiling.

**CBL —0369 W USE SENTENCE NOT ENDED
WITH PERIOD SPACE --
ASSUMED**

Explanation:

The USE sentence must be ended with a period and space.

Additional Information:

The missing period and space are assumed.

User Response:

Ensure that the USE sentence is followed by a period and space.

**CBL —0370 E LOGICAL OR ARITHMETIC
EXPRESSION TOO COMPLEX**

Explanation:

Either a compound condition or an arithmetic expression exceeds the compiler's capacity for temporary storage.

Additional Information:

The entire statement is dropped.

User Response:

Simplify the statement either by separating it into two or more statements or by removing excessive parentheses.

**CBL —0371 C [name] IS INVALID AS AN
INDICATOR DATA-ITEM**

Explanation:

An INDICATOR data item must be an elementary Boolean data item specified without the OCCURS clause or a group item that has elementary Boolean data items subordinate to it.

Additional Information:

The specified INDICATOR data item is accepted, but the indicator associated values generated for this WRITE statement might be invalid.

User Response:

Ensure that the specified INDICATOR data item either is an elementary Boolean data item without the OCCURS clause or is a group item.

**CBL —0372 E REQUIRED KEYWORD MISSING
IN 'USE' SENTENCE --
SENTENCE IGNORED**

Explanation:

The USE sentence must have one of the following forms:

- USE AFTER EXCEPTION/ERROR
- USE FOR DEBUGGING

Additional Information:

The USE sentence is ignored.

User Response:

Ensure that one of the two forms of the USE sentence is specified.

**CBL —0373 C SECTION/PARAGRAPH
PRECEDES DECLARATIVES
HEADER**

Explanation:

The key word DECLARATIVES must be written on the next line after the Procedure Division header.

Additional Information:

The section or paragraph name is accepted, but any reference to this section or paragraph will produce unpredictable results.

User Response:

Ensure that the Declaratives header immediately follows the Procedure Division header.

**CBL —0374 C PROCEDURE NAME INVALID
AFTER 'ALL PROCEDURES' --
'USE' SENTENCE IGNORED**

Explanation:

When the ALL PROCEDURES clause is specified in a USE FOR DEBUGGING sentence, a procedure name must not be specified in any USE FOR DEBUGGING sentence.

Additional Information:

The USE FOR DEBUGGING sentence is ignored.

User Response:

Ensure that no procedure names are specified in any USE FOR DEBUGGING sentence if ALL PROCEDURES is specified on a USE FOR DEBUGGING sentence.

**CBL —0375 W INVALID SEGMENT PRIORITY
FOR DECLARATIVES SECTION
-- ZERO ASSUMED**

Explanation:

Sections in the DECLARATIVES must specify a segment priority less than 50.

Additional Information:

A segment priority of 0 is assumed.

User Response:

If a segment priority is specified, ensure that it is in a valid format and a value less than 50.

**CBL —0376 C 'USE' SENTENCE DOES NOT
FOLLOW SECTION HEADER**

Explanation:

In the Declaratives Section of the Procedure Division, a USE sentence must follow a section header.

Additional Information:

None.

User Response:

Ensure that the section header in the Declaratives Section is followed by a USE sentence.

**CBL —0377 E SECTION HEADER MUST
IMMEDIATELY FOLLOW 'END
DECLARATIVES'**

Explanation:

A section header does not immediately follow the END DECLARATIVES statement.

Additional Information:

The first section or paragraph name is used as the program entry point.

User Response:

Ensure that a section header follows the END DECLARATIVES statement.

**CBL —0378 C PROCEDURE [name] IS IN THE
DECLARATIVES SECTION**

Explanation:

Procedure names within debugging Declarative Sections must not appear in any USE FOR DEBUGGING sentence.

Additional Information:

The procedure name is ignored.

User Response:

Ensure that procedure names within debugging Declarative Sections do not appear in any USE FOR DEBUGGING sentence.

**CBL —0379 E 'ALL PROCEDURES' INVALID
AFTER PROCEDURE NAME --
'USE' SENTENCE IGNORED**

Explanation:

When the ALL PROCEDURES clause is specified in a USE FOR DEBUGGING sentence, procedure names must not be specified in any USE FOR DEBUGGING sentence.

Additional Information:

The USE FOR DEBUGGING sentence is ignored.

User Response:

Ensure that ALL PROCEDURES is not specified in a USE FOR DEBUGGING sentence after a procedure name has been specified in a USE FOR DEBUGGING sentence.

**CBL —0380 C PROCEDURE NAME [name]
ALREADY SPECIFIED --
IGNORED**

Explanation:

A procedure name may appear in only one USE FOR DEBUGGING sentence, and it may appear only once in that sentence.

Additional Information:

The procedure name is ignored.

User Response:

Ensure that a procedure name is specified only once in USE FOR DEBUGGING sentences.

CBL —0382 C ALL OTHER DECLARATIVE PROCEDURES MUST FOLLOW ALL 'USE FOR DEBUGGING' PROCEDURES -- ACCEPTED

Explanation:

When specified, all debugging sections must be written immediately after the Declaratives Section header.

Additional Information:

Normal processing.

User Response:

Ensure that all debugging sections immediately follow the Declaratives Section header.

CBL —0383 E PROCEDURE NAME [name] IS AN INVALID PROCEDURE REFERENCE FOR THIS SECTION

Explanation:

Procedure references must conform to the following rules:

- Within a declarative procedure, there must be no reference to any nondeclarative procedure.
- In the nondeclarative portion of the program, there must be no reference to procedure names that appear in an EXCEPTION/ERROR declarative procedure, except that PERFORM statements may refer to an EXCEPTION/ERROR procedure or to procedures associated with it.
- Statements appearing outside the debugging sections must not refer to procedure names defined within the debugging sections.
- Except for the USE FOR DEBUGGING sentence itself, statements within a debugging declarative section may refer to procedure names defined in a different USE procedure only through the PERFORM statement.

Additional Information:

A null procedure reference is generated.

User Response:

Ensure that all procedure references conform to the above rules.

CBL —0384 E PROCEDURE-1 AND PROCEDURE-2 NOT IN THE SAME DECLARATIVES SECTION

Explanation:

In a PERFORM statement, when 'procedure-1 THROUGH procedure-2' is specified and either is a procedure in the

declarative section of the program, then both procedures must be in the same declarative section.

Additional Information:

The procedure names are accepted. Incorrect code may be generated.

User Response:

Ensure that when PERFORM specifies procedure names contained in the Declarative Section of the program, both procedure names are in the same Declarative Section.

CBL —0385 C MULTIPLE 'USE' SENTENCES FOLLOW SECTION HEADER

Explanation:

Only one USE sentence should follow a section header in the Declaratives Section.

Additional Information:

If the section being processed is a debugging Declarative Section, all USE sentences except the first are ignored.

If the section being processed is an EXCEPTION/ERROR Declarative Section, all USE sentences are accepted and processed.

User Response:

Ensure that only one USE sentences follows a section header in the Declaratives Section.

CBL —0386 E NO 'END DECLARATIVES' FOUND

Explanation:

A DECLARATIVES statement was found in the source program, but no END DECLARATIVES statement was detected following the Declaratives Section.

Additional Information:

None. Incorrect code will be generated.

User Response:

Ensure an END DECLARATIVES statement exists after a DECLARATIVES statement.

CBL —0387 E [name] IS AN INVALID KEY IN A SORT/MERGE STATEMENT, KEY IGNORED

Explanation:

A key defined in an ASCENDING/DESCENDING KEY clause of a SORT statement violates one of the restrictions on how it can be defined.

Additional Information:

The key is ignored.

User Response:

Determine why the key is not valid and correct the error.

**CBL —0388 C TOTAL KEY LENGTH OF
SORT/MERGE KEYS EXCEED
256 BYTES, LAST KEY(S)
IGNORED**

Explanation:

The compiler has a maximum of 256 bytes of storage for all keys associated with a SORT/MERGE statement. This can include some bytes used internally by the SORT/MERGE utility. This SORT/MERGE statement exceeded that limit.

Additional Information:

The remaining keys from the point at which the key length exceeded 256 are ignored.

User Response:

Reduce the number and/or size of the keys specified for the SORT/MERGE statement.

**CBL —0389 E NO VALID
ASCENDING/DESCENDING
KEYS FOUND FOR A
SORT/MERGE STATEMENT**

Explanation:

The compiler could find no valid keys specified for a SORT/MERGE statement.

Additional Information:

The compiler will generate a call to SORT with no keys defined.

User Response:

Ensure that there is at least one valid key specified for a SORT/MERGE statement.

**CBL —0390 E SORT VERB ENCOUNTERED
WITH FILE NAMED "WORK"
DEFINED**

Explanation:

SORT uses a file named WORK to perform certain functions during execution. Because the COBOL programmer has defined a file named WORK, SORT will attempt to use the programmer's file during execution.

Additional Information:

The program will be generated. Problems will occur when SORT attempts to use the programmer's WORK file.

User Response:

Change the name of the file from WORK to another valid file name.

**CBL —0400 E OPEN OPTION INCOMPATIBLE
WITH FILE SPECIFICATION
FOR [name]**

Explanation:

The OPEN statement is invalid for the file name specified.

Additional Information:

Code is not generated for the OPEN of this file, but it is generated for all other valid file names in the statement.

User Response:

Correct invalid OPEN statement before recompiling.

**CBL —0401 E MISSING KEYWORD INPUT,
EXTEND, OUTPUT, OR I-O IN
OPEN STATEMENT**

Explanation:

One of the required key words in the OPEN statement was omitted.

Additional Information:

All input elements are discarded until one of the required key words is found. Normal processing continues after the key word is located.

User Response:

Supply valid key word before recompiling.

**CBL —0402 E DEVICE TYPE OF DATABASE
MUST BE ASSIGNED TO
[device-name]**

Explanation:

The device type was not DATABASE.

Additional Information:

The READ statement specified FIRST, LAST, or PRIOR but the device type was not DATABASE.

User Response:

Change the device type to DATABASE before recompiling.

**CBL —0403 E START/DELETE
INCOMPATIBLE WITH FILE
SPECIFICATION FOR [name]**

Explanation:

A START statement was issued for a sequentially organized file, or for a file of any organization that was being accessed

randomly; or a DELETE statement was issued for a sequential file.

Additional Information:

No code is generated for the START statement.

User Response:

Correct invalid statement before recompiling.

CBL —0404 E [name] NOT COMPATIBLE WITH GIVEN ACCESS MODE

Explanation:

The access mode specified was not DYNAMIC.

Additional Information:

The READ statement specified FIRST, LAST, or PRIOR but the access mode was not DYNAMIC.

User Response:

Change the access mode to DYNAMIC before recompiling.

CBL —0405 E READ INCOMPATIBLE WITH FILE SPECIFICATION FOR [name]

Explanation:

A READ statement was issued for a file that can only be opened for output.

Additional Information:

No code is generated for the READ statement.

User Response:

Correct invalid statement before recompiling.

CBL —0406 C INTO OPTION INVALID FOR MULTIPLE LOGICAL RECORDS OF [name] -- ACCEPTED

Explanation:

The INTO option is invalid for files containing multiple level-01 records of varying length in the File Section.

Additional Information:

The first record description found for the file is used.

User Response:

Correct invalid READ INTO statement before recompiling.

CBL —0407 E [name] NOT COMPATIBLE WITH GIVEN FILE ORGANIZATION

Explanation:

The file organization specified was not INDEXED.

Additional Information:

The READ statement specified FIRST, LAST, or PRIOR but the file organization was not INDEXED.

User Response:

Change the file organization to INDEXED before recompiling.

CBL —0408 E WRITE INCOMPATIBLE WITH FILE SPECIFICATION FOR [name]

Explanation:

A WRITE statement was issued for a file described as an input file.

Additional Information:

No code is generated for the WRITE statement.

User Response:

Correct invalid statement before recompiling.

CBL —0409 E [insert] INCOMPATIBLE WITH FILE SPECIFICATION FOR [name]

Explanation:

Printer options were specified for a file that is not a print file (EOP), an ADVANCING phrase was specified for a file that is not unit-record output, or the mnemonic name used in the ADVANCING phrase is unsuitable.

Additional Information:

No code is generated for the WRITE statement.

User Response:

Correct invalid specification before recompiling.

CBL —0410 C LINAGE CLAUSE NOT SPECIFIED FOR [name] -- END-OF-PAGE OPTION INVALID

Explanation:

An EOP option was specified for a file for which no LINAGE clause was specified.

Additional Information:

The EOP option is ignored. The code intended as an EOP routine will be executed after each WRITE.

User Response:

Supply required LINAGE clause before recompiling.

**CBL —0411 E REWRITE INCOMPATIBLE
WITH FILE SPECIFICATION
FOR [name]**

Explanation:

A REWRITE statement was issued for a file that cannot be opened in I-O mode.

Additional Information:

No code is generated for the REWRITE statement.

User Response:

Correct invalid statement before recompiling.

**CBL —0412 W MORE THAN ONE USE
PROCEDURE SPECIFIED FOR
FILE [name]**

Explanation:

The file being opened has an implicit USE PROCEDURE defined for INPUT, OUTPUT, I-O, or EXTEND and an explicit USE PROCEDURE for this file.

Additional Information:

The explicit USE PROCEDURE will be used for this file.

User Response:

If the implicit USE PROCEDURE was desired, remove the explicit USE PROCEDURE for this file before recompiling.

**CBL —0413 E INVALID SECTION-NAME
SPECIFIED FOR SORT/MERGE
INPUT OR OUTPUT
PROCEDURE**

Explanation:

The name specified in the INPUT or OUTPUT PROCEDURE is not a section name.

Additional Information:

Code generation and syntax checking are discontinued at the point where the message was issued and are resumed at the next statement. The completeness of generated code is unpredictable.

User Response:

Correct the invalid section name before recompiling.

**CBL —0414 E NUMBER OF KEYS IN A SORT
OR MERGE STATEMENT
EXCEEDS 12**

Explanation:

The number of ASCENDING/DESCENDING keys in a SORT or MERGE statement must not exceed 12.

Additional Information:

All keys beyond the twelfth are discarded without syntax checking. Code is generated for the SORT or MERGE statement. Only the first 12 keys are used.

User Response:

Ensure that no more than 12 keys are specified before recompiling.

**CBL —0415 E NUMBER OF FILES IN A SORT
OR MERGE USING CLAUSE
EXCEEDS 8**

Explanation:

The number of files must not exceed 8 in the USING clause.

Additional Information:

All files beyond the eighth are discarded without syntax checking. Code is generated for the USING clause and only the first eight files are used.

User Response:

Ensure that no more than eight files appear after USING before recompiling.

**CBL —0416 W [name] EXCEEDS 256
CHARACTERS -- TRUNCATED
FOR COMPARISON**

Explanation:

The length of the identifier exceeds the 256-character limit of comparison for EXHIBIT CHANGED.

Additional Information:

Only the leftmost 256 characters are analyzed to determine whether the value has changed.

User Response:

Check that useful information is not being discarded by the truncation.

**CBL —0418 E NUMBER OF
PROCEDURE-NAMES IN GO TO
DEPENDING ON STATEMENT
EXCEEDS 99**

Explanation:

The number of procedure names in the GO TO
DEPENDING ON statement must not exceed 99.

Additional Information:

The first 99 procedure names are used.

User Response:

Ensure that the number of procedure names in the GO TO . . . DEPENDING ON statement does not exceed 99 before recompiling.

CBL —0419 E [name] AND [name] ARE IN DIFFERENT SEGMENTS -- CANNOT ALTER

Explanation:

An ALTER statement was issued for a GO TO statement in an independent segment from a segment of different priority.

Additional Information:

No code is generated.

User Response:

Correct the ALTER statement, or revise priority numbers before recompiling.

CBL —0420 W NUMBER OF DIGITS IN [name] EXCEEDS 4 -- TRUNCATED

Explanation:

The PICTURE for the integer in the GO TO . . . DEPENDING ON statement has more than the allowable number of 9s.

Additional Information:

The 4 low-order digits are used.

User Response:

Ensure that the high-order positions contain no significant digits, or specify the PICTURE for the identifier again, before recompiling.

CBL —0421 E [name] AND [name] ARE IN DIFFERENT SEGMENTS -- CANNOT PERFORM

Explanation:

The PERFORM range includes segments of unequal priority, or a PERFORM statement in an independent segment refers to a paragraph in a different independent segment.

Additional Information:

No code is generated.

User Response:

Correct invalid statement, or revise priority numbers, before recompiling.

CBL —0422 E DEPTH OF VARYING EXCEEDS 3

Explanation:

The number of AFTER phrases in a PERFORM statement must not exceed 2.

Additional Information:

The third AFTER phrase and the remainder of the statement are ignored. Code is generated for the statement up to the third AFTER phrase.

User Response:

Delete third AFTER phrase before recompiling.

CBL —0423 E [name] IS INVALID CALL PARAMETER

Explanation:

A file name, a procedure name, a key word, a literal, or an index was found after the USING clause.

Additional Information:

The invalid parameter is counted in the check for maximum--15 CALL parameters, and a dummy address is entered for it in the object-time parameter list.

User Response:

Supply a valid CALL parameter before recompiling.

CBL —0424 E NUMBER OF PARAMETERS IN CALL STATEMENT EXCEEDS 15

Explanation:

The number of parameters following the USING phrase must not exceed 15.

Additional Information:

All parameters beyond the fifteenth are discarded without syntax checking. Code is generated for the CALL statement and for the first 15 parameters.

User Response:

Ensure that no more than 15 parameters appear with the USING phrase before recompiling.

CBL —0427 E INVALID MOVE STATEMENT -- [name or literal] AND [name or literal] INCOMPATIBLE

Explanation:

The sending and receiving fields in the MOVE statement are incompatible.

Additional Information:

The MOVE is not generated.

User Response:

Correct the invalid statement before recompiling.

**CBL —0428 E INVALID INDEX
MODIFICATION -- [name or literal]
AND [name or literal]
INCOMPATIBLE**

Explanation:

The two phrases of the SET statement do not form a valid pair (for example, SET index UP BY index).

Note: This message may also be issued for a PERFORM VARYING statement, because SET is implied when the variable is an index.

Additional Information:

The index modification in the SET or PERFORM statement is not generated.

User Response:

Correct the invalid statement before recompiling.

**CBL —0429 E INVALID RELATIONAL TEST --
[name or literal] AND [name or
literal] INCOMPATIBLE**

Explanation:

An invalid comparison was made (for example, an arithmetic expression with an alphanumeric data name, a literal with a literal, an index data item with a binary data name).

Additional Information:

No code is generated for the comparison; or within a compound condition, one of the fields may arbitrarily be compared with itself or to 0.

User Response:

Correct the invalid statement before recompiling.

**CBL —0430 W INVALID SIGN TEST OF [name]
-- ACCEPTED**

Explanation:

A NEGATIVE condition was specified for an unsigned numeric field.

Additional Information:

The test is generated.

User Response:

Correct the invalid specification before recompiling.

**CBL —0431 E INVALID SIGN TEST OF [name] --
IGNORED**

Explanation:

A sign test was applied to a nonnumeric field.

Additional Information:

No code is generated except within a compound condition.

User Response:

Ensure that the indicated element is numeric before recompiling.

CBL —0432 E INVALID CLASS TEST OF [name]

Explanation:

Either a numeric test was requested for an alphanumeric or computational field or an alphabetic test was requested for a numeric field.

Additional Information:

Code generation and syntax checking are discontinued at the point where the message was issued and are resumed at the next statement (the true-condition routine). The results are unpredictable.

User Response:

Correct the invalid specification before recompiling.

**CBL —0433 C CONDITIONAL STATEMENT
INVALID AT THIS POINT --
ACCEPTED**

Explanation:

A conditional statement was found within another conditional statement.

Additional Information:

The indicated conditional statement is accepted. However, the next statement executed after that conditional statement is unpredictable.

User Response:

Ensure that no more than one conditional statement is specified in a sentence before recompiling.

**CBL —0434 E LOGICAL EXPRESSION TOO
COMPLEX**

Explanation:

Too many parentheses were used. The level of parentheses in a compound condition exceeds the compiler's capacity for temporary storage.

Additional Information:

Code generation and syntax checking are discontinued at the point where the message was issued and are resumed at the next statement (the true-condition routine). The results are unpredictable.

User Response:

Make the compound condition a series of simple conditions before recompiling.

**CBL —0435 W DECIMAL ALIGNMENT
CAUSES [name or literal] TO BE
TREATED AS ZERO**

Explanation:

The difference in scaling between two numeric fields in a MOVE or arithmetic statement is such that all significant digits of one of the two fields are truncated.

Additional Information:

All fields are aligned on the decimal point.

User Response:

Check the length and the scaling of all elements in the statement before recompiling.

**CBL —0436 W HIGH-ORDER TRUNCATION
MAY OCCUR**

Explanation:

Overflow of high-order digits may cause significance to be lost. Overflow occurs when either an intermediate result field or the final result field is too small to hold possible computed values. For more information on intermediate result fields, see the manual, *Programming with COBOL*.

Additional Information:

None.

User Response:

Ensure that truncation will not produce incorrect results.

**CBL —0437 W RESULT DOES NOT HAVE
MORE DECIMALS THAN
RECEIVING FIELD —
ROUNDING IGNORED**

Explanation:

The size of a fractional result does not exceed the number of places provided for its storage. (For example, ADD PIC 99V9 to PIC 99V9 ROUNDED.)

Additional Information:

No rounding takes place.

User Response:

Correct the specification before recompiling.

CBL —0438 C [insert] INVALID -- IGNORED

Explanation:

An extraneous element was found in an otherwise valid statement.

Additional Information:

The element is dropped, and processing continues.

User Response:

Delete the extraneous element before recompiling.

CBL —0439 C [key word] MISSING -- ASSUMED

Explanation:

A required key word was omitted.

Additional Information:

The missing key word is assumed to be present.

User Response:

Supply the required key word before recompiling.

**CBL —0440 E OPTION MISSING —
STATEMENT IGNORED**

Explanation:

One of several key words is required to decide the meaning of some statements. If such a key word is missing, the compiler is unable to make any assumptions.

Additional Information:

No code is generated for the statement.

User Response:

Supply missing option before recompiling.

**CBL —0441 E [insert] EXPECTED -- [insert]
FOUND — REST OF
STATEMENT IGNORED**

Explanation:

A source element is not recognized in the context of its statement.

Additional Information:

Code generation and syntax checking are discontinued at the point where the message is issued and are resumed at the next statement.

User Response:

Correct the invalid statement before recompiling.

**CBL —0442 E [insert] INVALID RECEIVING
FIELD**

Explanation:

A literal, a procedure name, or a file name was specified as a receiving field.

Additional Information:

The data in the sending field of the MOVE statement or the result of an arithmetic operation will not be stored.

User Response:

Correct the invalid statement before recompiling.

CBL —0443 E [insert] INVALID HERE -- NOT NUMERIC

Explanation:

A nonnumeric field was specified as an operand for an arithmetic statement.

Additional Information:

Code is not generated or, in some cases, zero is substituted for the operand.

User Response:

Correct the invalid statement before recompiling.

CBL —0444 C [name or literal] INVALID HERE -- NOT AN INTEGER

Explanation:

An exponent used in an arithmetic expression must be an integer. An integer is required for the GO TO . . . DEPENDING ON and the PERFORM . . . TIMES statements. The element specified contains one or more positions to the right of the decimal point.

Additional Information:

The decimal positions are truncated from the exponent. In all other cases, the indicated element is assumed to have a value of 0.

User Response:

Correct the invalid statement before recompiling.

CBL —0445 C NAMED CHANGED SHOULD BE CHANGED NAMED -- ACCEPTED

Explanation:

The required order in an EXHIBIT statement is EXHIBIT CHANGED NAMED identifier.

Additional Information:

The statement is accepted.

User Response:

Correct the invalid statement before recompiling.

CBL —0446 E [name or literal] INVALID IN THIS CONTEXT

Explanation:

The operand does not fulfill a special requirement for the statement in question.

Additional Information:

No code is generated.

User Response:

Correct the invalid statement before recompiling.

CBL —0447 E STATEMENT INCOMPLETE

Explanation:

No statement appears between a test condition and the end of the statement (for example, IF A = B) or a statement ends before all required elements were found.

Additional Information:

For an incomplete condition, the next sequential statement is run whether the condition is satisfied or not. For other incomplete statements, results are unpredictable.

User Response:

Supply the missing statement before recompiling.

CBL —0448 E ELSE IS UNMATCHED BY IF -- IGNORED

Explanation:

Each ELSE must be matched by an IF statement.

Additional Information:

The unmatched ELSE is ignored, and the code is generated for the statement following the ELSE.

User Response:

Supply the missing IF statement, or delete the unmatched ELSE before recompiling.

CBL —0449 C NEXT SENTENCE CLAUSE INVALID HERE -- ACCEPTED

Explanation:

The NEXT SENTENCE clause must not follow the INVALID KEY phrase or AT END condition.

Additional Information:

An invalid key or end-of-file condition causes a branch to the NEXT SENTENCE clause.

User Response:

Correct the invalid statement before recompiling.

CBL —0450 W STATEMENT WILL NOT BE EXECUTED

Explanation:

A statement follows a GO TO, STOP RUN, or NEXT SENTENCE statement without an intervening period and a space or an ELSE statement.

Additional Information:

Correct code is generated.

User Response:

Insert a period or an ELSE statement where required by program logic before recompiling.

CBL —0452 E ASSUMING [insert] STANDS FOR [insert]

Explanation:

A source element is invalid where it appears; however, the following element is valid, indicating that perhaps the invalid element is the result of a keying error.

Additional Information:

The necessary key word is assumed, and the incorrect element is dropped.

User Response:

Correct the invalid statement before recompiling.

CBL —0453 C MAXIMUM NUMBER OF COMPARISON OPERANDS EXCEEDED -- EXCESS IGNORED

Explanation:

No more than 15 comparison operands can be specified for a REPLACING phrase or a TALLYING phrase.

Additional Information:

All comparison operands after the first 15 are ignored.

User Response:

Ensure that the REPLACING phrase or TALLYING phrase contains no more than 15 comparison operands before recompiling.

CBL —0456 E NAME OR LITERAL MISSING BEFORE [insert]

Explanation:

A key word was found where only an operand would be valid.

Additional Information:

Syntax checking continues as if a valid operand had been found, but either no code or arbitrary code is generated for the statement in error. Results are unpredictable.

User Response:

Correct invalid statement before recompiling.

CBL —0457 E INVALID TRANSACTION FILE TERMINAL ID LENGTH OR TYPE

Explanation:

The terminal ID must be 2 characters; a data name specified as a terminal ID must be alphanumeric; a literal specified as a terminal ID must be nonnumeric.

Additional Information:

The entire statement is ignored.

User Response:

Correct the item before recompiling.

CBL —0458 E [insert] IS NOT DEFINED AS A TRANSACTION FILE

Explanation:

The file name specified on the ACQUIRE or DROP statement is not defined as a TRANSACTION file.

Additional Information:

The entire statement is ignored.

User Response:

Correct the invalid file specification before recompiling.

CBL —0459 E PRECEDING ELSE PHRASE INCOMPLETE -- NEXT SENTENCE ASSUMED

Explanation:

ELSE phrase does not contain a valid verb.

Additional Information:

Code for NEXT SENTENCE clause is generated.

User Response:

Correct the invalid ELSE phrase before recompiling.

CBL —0460 E MORE THAN 15 DELIMITERS SPECIFIED -- [name or literal] IGNORED

Explanation:

The number of delimiters specified for an UNSTRING statement must not exceed 15.

Additional Information:

All delimiters beyond the fifteenth are discarded without syntax checking. Code is generated for the UNSTRING statement using only the first 15 delimiters.

User Response:

Ensure that no more than 15 delimiters are specified for the UNSTRING statement before recompiling.

**CBL —0461 E 'DELIMITER IN' IS NOT VALID
UNLESS 'DELIMITED BY' IS
SPECIFIED -- [name or literal]
IGNORED**

Explanation:

The DELIMITER IN phrase is specified without the DELIMITED BY phrase for an UNSTRING statement.

Additional Information:

The DELIMITER IN phrase is discarded without syntax checking. Code is generated for the UNSTRING statement without the DELIMITER IN phrase.

User Response:

Correct the invalid statement before recompiling.

**CBL —0462 E 'COUNT IN' IS NOT VALID
UNLESS 'DELIMITED BY' IS
SPECIFIED -- [name or literal]
IGNORED**

Explanation:

The COUNT IN phrase is specified without the DELIMITED BY phrase for an UNSTRING statement.

Additional Information:

The COUNT IN phrase is discarded without syntax checking. Code is generated for the UNSTRING statement without the COUNT IN phrase.

User Response:

Correct the invalid statement before recompiling.

**CBL —0463 C DELIMITER [name or literal]
EXCEEDS 1 CHARACTER --
TRUNCATED**

Explanation:

For INSPECT REPLACING CHARACTERS clause, the size of a delimiter is limited to 1 character.

Additional Information:

The leftmost character of the delimiter is used.

User Response:

Reduce the size of the delimiter before recompiling.

**CBL —0464 E REPLACING STRING [name or
literal] SMALLER THAN
REPLACEMENT STRING**

Explanation:

In an INSPECT statement, the length of a replacement string exceeds the length of the replacing string.

Additional Information:

Code generation is discontinued but syntax checking continues for this statement.

User Response:

Correct the size of the strings so that they match.

**CBL —0465 C REPLACING STRING [name or
literal] LARGER THAN
REPLACEMENT STRING --
TRUNCATED**

Explanation:

In an INSPECT statement, the length of a replacement string is exceeded by the length of the replacing string.

Additional Information:

The replacing string is truncated on the right to match the size of the replacement string.

User Response:

Correct the size of the strings so that they match.

**CBL —0466 C TERMINAL ID INVALID -- FOR
CLAUSE IGNORED**

Explanation:

The terminal ID specified on an ACCEPT or DISPLAY statement must be a 2-character alphanumeric field.

Additional Information:

The FOR clause is ignored.

User Response:

Correct the invalid terminal ID before recompiling.

**CBL —0467 C BOOLEAN COMPARISON MUST
BE EQUAL OR NOT EQUAL --
EQUAL ASSUMED**

Explanation:

The relational operator for a relational condition involving Boolean data must be equal or not equal.

Additional Information:

The test is generated with equal assumed.

User Response:

Correct the relational operator before recompiling.

**CBL —0468 E INVALID FORMAT NAME
SPECIFIED FOR A
TRANSACTION FILE WRITE**

Explanation:

The FORMAT NAME clause must not be greater than 8 characters; a data name specified for the FORMAT NAME clause must be alphanumeric; a literal specified for the FORMAT NAME clause must be nonnumeric.

Additional Information:

The entire statement is ignored.

User Response:

Correct the invalid item before recompiling.

**CBL —0469 E INCORRECT LINE/LINES
SPECIFIED FOR
STARTING/ROLLING/UP/DOWN
CLAUSE**

Explanation:

The data name specified for the LINES phrase must be defined as an elementary numeric item; the literal specified for the LINES phrase must be a numeric item.

Additional Information:

The entire statement is ignored.

User Response:

Correct the invalid item before recompiling.

**CBL —0470 E INVALID SYSTEM-NAME IN
ASSIGN CLAUSE FOR S/36
COBOL -- FILE IGNORED**

Explanation:

The system name specified in the SELECT clause is a device that is not supported by System/36 COBOL.

Additional Information:

The file is ignored.

User Response:

Remove all references to the invalid device before recompiling.

**CBL —0472 W [option] OPTION NOT
SUPPORTED FOR S/36 COBOL --
OPTION IGNORED**

Explanation:

The option specified on this COBOL statement is not supported by System/36 COBOL.

Additional Information:

The option is ignored.

User Response:

Remove the option before recompiling.

**CBL —0473 W LIBRARY NAME INVALID OR
LIBRARY NOT FOUND --
LIBRARY IGNORED**

Explanation:

The library name specified with the OBJECT, LINK, or SUBLIB parameter of the PROCESS statement is invalid. The library name specified with the LIBRARY parameter on the PROCESS statement is invalid or cannot be located.

Additional Information:

The option specified on the PROCESS statement is ignored.

User Response:

Correct the invalid name specified before recompiling the job.

**CBL —0474 C KEYWORD [key word] INVALID
IN THIS DIVISION -- IGNORED**

Explanation:

A key word was found in the wrong division.

Additional Information:

The key word is ignored.

User Response:

Remove the key word before recompiling.

**CBL —0475 C EXTRANEIOUS DATA ON THE
USE STATEMENT - REST OF
STATEMENT IGNORED**

Explanation:

Additional text was found on the USE statement.

Additional Information:

The additional text on the remainder of the USE statement is ignored.

User Response:

Remove the additional text before recompiling.

**CBL —0476 C DECLARATIVES MISSING --
STATEMENT IGNORED**

Explanation:

A declarative sentence (USE statement) was found without a preceding DECLARATIVE statement.

Additional Information:

The statement is ignored.

User Response:

Place the USE statement and procedure after a DECLARATIVES statement before recompiling.

**CBL —0477 C DECLARATIVES OR END
DECLARATIVES HAS ALREADY
BEEN PROCESSED --
STATEMENT IGNORED**

Explanation:

Another DECLARATIVES statement was detected.

Additional Information:

This DECLARATIVES statement is ignored.

User Response:

Remove one of the DECLARATIVES statements before recompiling.

**CBL —0478 C NO DECLARATIVES SECTION --
END DECLARATIVES IGNORED**

Explanation:

An END DECLARATIVES statement was found without a matching DECLARATIVES statement.

Additional Information:

The END DECLARATIVES statement is ignored.

User Response:

Remove the END DECLARATIVES statement or supply the missing DECLARATIVES statement before recompiling.

**CBL —0479 E COPY NOT ALLOWED SINCE
NOLIB INDICATED --
STATEMENT IGNORED**

Explanation:

The NOLIB option of the PROCESS statement was specified, which implies that COPY statements are not going to be used by this program.

Additional Information:

The COPY statement is ignored.

User Response:

Remove the COPY statement from the program or remove the NOLIB option from the PROCESS statement before recompiling

**CBL —0480 E INVALID IDENTIFIER IN COPY
STATEMENT -- COPY IGNORED**

Explanation:

An invalid identifier was found in the REPLACING clause of the COPY statement.

Additional Information:

The COPY statement is ignored.

User Response:

Correct the identifier used in the REPLACING clause of the COPY statement before recompiling.

**CBL —0481 E INVALID WORD IN COPY
STATEMENT -- COPY IGNORED**

Explanation:

An invalid word was found in the REPLACING clause of the COPY statement.

Additional Information:

The COPY statement is ignored.

User Response:

Correct the word used in the REPLACING clause of the COPY statement before recompiling.

**CBL —0482 E INCOMPLETE IDENTIFIER IN
COPY STATEMENT -- COPY
IGNORED**

Explanation:

The identifier was incomplete or the parenthesis was not properly closed in the REPLACING clause of the COPY statement.

Additional Information:

The COPY statement is ignored.

User Response:

Correct the identifier or the parenthesis in the REPLACING clause of the COPY statement before recompiling.

**CBL —0483 E INCOMPLETE COPY
STATEMENT -- COPY IGNORED**

Explanation:

The COPY statement was not complete.

Additional Information:

The COPY statement is ignored.

User Response:

Complete the COPY statement before recompiling.

CBL —0484 E OVERFLOW OF REPLACING TABLE -- COPY IGNORED

Explanation:

The REPLACING table was full; therefore the REPLACING clause was not processed.

Additional Information:

The COPY statement is ignored.

User Response:

Increase the region size for the compiler by using a REGION OCL statement before recompiling.

CBL —0485 E EXPECTING 'BY' IN THE COPY STATEMENT -- COPY IGNORED

Explanation:

The key word BY was not found in the REPLACING clause of the COPY statement.

Additional Information:

The COPY statement is ignored.

User Response:

Add the key word BY to the REPLACING clause before recompiling.

CBL —0486 E PSEUDO-TEXT-1 IS NULL IN THE COPY -- COPY IGNORED

Explanation:

Pseudo-text-1 in the REPLACING clause of the COPY statement was null. No replacement could be performed.

Additional Information:

The COPY statement is ignored.

User Response:

Correct pseudo-text-1 in the REPLACING clause of the COPY statement before recompiling.

CBL —0487 E AN ENDING DELIMITER IN THE COPY STATEMENT IS NOT

FOLLOWED BY A VALID SEPARATOR -- COPY IGNORED

Explanation:

A valid separator does not follow the ending delimiter of pseudo-text in the REPLACING clause of the COPY statement.

Additional Information:

The COPY statement is ignored.

User Response:

Correct the REPLACING clause of the COPY statement by adding a valid separator before recompiling.

CBL —0488 E INVALID ITEM IN AREA A OF THE COPY STATEMENT -- COPY IGNORED

Explanation:

Invalid clauses of the COPY statement were found starting in area A of the COBOL statement.

Additional Information:

The COPY statement is ignored.

User Response:

Correct the COPY statement by moving the clauses into area B before recompiling.

CBL —0489 E INVALID USE OF COMMA OR SEMICOLON IN COPY STATEMENT -- COPY IGNORED

Explanation:

A comma or a semicolon was found in an incorrect place in the REPLACING clause of a COPY statement.

Additional Information:

The COPY statement is ignored.

User Response:

Remove the invalid comma or semicolon in the COPY statement before recompiling.

CBL —0490 W DEBUGGING LINES INVALID WITHIN PSEUDO-TEXT-1 -- LINE IGNORED

Explanation:

A debugging line was found in pseudo-text-1 of the REPLACING clause of the COPY statement.

Additional Information:

The debugging line is ignored, and the COPY statement with the REPLACING clause is performed.

User Response:

Remove the debugging line from pseudo-text-1 in the REPLACING clause of the COPY statement before recompiling.

CBL —0491 W OVERFLOW IN COMMENT BUFFER

Explanation:

The maximum of eight comment statements was exceeded in a text name being copied into the COBOL program.

Additional Information:

Stops copying comment statements from the text name into the COBOL program.

User Response:

Remove some of the comments from the text name before recompiling.

CBL —0495 C A QUOTE/APOSTROPHE MUST BE THE FIRST ELEMENT OF A NON-NUMERIC LITERAL

Explanation:

A nonnumeric literal was preceded by a character other than a quote or an apostrophe.

Additional Information:

The literal is not processed by the compiler.

User Response:

Correct the character string so that it is a valid COBOL literal before recompiling.

CBL —0496 E INVALID BOOLEAN LITERAL -- LITERAL IGNORED

Explanation:

A Boolean literal contained a numeric other than 0 or 1.

Additional Information:

The Boolean literal is not processed by the compiler.

User Response:

Correct the character string so that it is a valid COBOL Boolean literal before recompiling.

CBL —0500 W WARNING -- [phrase] IS [level] WHICH EXCEEDS THE FIPS LEVEL SPECIFIED

Explanation:

The phrase is a higher level than specified on the LVL option of the PROCESS statement. The level of the indicated phrase is given by the level insert.

Additional Information:

Processing continues.

User Response:

If the program must be run with the level of support specified on the PROCESS statement, remove the indicated phrase from the program before recompiling.

CBL —0999 E FATAL COMPILER ERROR

Explanation:

A disruptive error has occurred in the COBOL compiler.

Additional Information:

Compilation is discontinued.

User Response:

If this error continues to occur, contact your programming service representative.



Chapter 3. Displayed Messages

CBL —1000 COBOL Compiler phase not found

Severity: 5 Auto Response:

Explanation:

The COBOL compiler was loaded incorrectly.

Additional Information:

Reload the COBOL Program Product into the system. If this problem continues to occur, contact your programming service representative.

CBL —1001 No DATA DIVISION or PROCEDURE DIVISION found

Severity: 3 Auto Response: 2

Explanation:

The Data Division or the Procedure Division portion of the COBOL source program could not be found while compiling the program.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1001)

Additional Information:

While processing the source program, the compiler could not locate a Data Division or a Procedure Division header statement. If after properly inserting these required statements, the problem continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1003 More than 65,535 statements in program

Severity: 3 Auto Response: 2

Explanation:

The COBOL compiler detected that the program contains more than the maximum 65,535 statements allowed in a COBOL program.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1003)

Additional Information:

The maximum number of statements allowed in a COBOL program is 65,535

If after reducing your program to within the maximum number of statements, the problem continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1005 More than 32,767 entries in name table

Severity: 3 Auto Response: 2

Explanation:

The COBOL compiler detected that the name table contains more than the maximum 32,767 entries allowed.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1005).

Additional Information:

The maximum number of statements allowed in a name table is 32,767.

If after reducing the name table to within the maximum allowable statements, the problem continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1006 Object program exceeds 65,535 bytes

Severity: 3 Auto Response: 2

Explanation:

The COBOL compiler detected that the object program exceeds the maximum 65,535 bytes allowed in a COBOL program.

This error may also be caused if the block size exceeds 64K.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1006).

Additional Information:

It may be possible to avoid this problem by segmenting the user program. For more information, see *Segmentation Feature* in the manual, *Programming with COBOL*.

If the problem continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1007 Permanent error reading program source

Severity: 3 Auto Response:

Explanation:

The COBOL compiler detected a permanent error while the input source member specified on a COPY statement was being read in.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1007).

Additional Information:

If this error continues to occur, fill out *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1008 Permanent I/O error on printer file

Severity: 3 Auto Response:

Explanation:

The COBOL compiler encountered a permanent I/O error while the compiler output listing was printing.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1008).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1009 Permanent disk I/O error

Severity: 3 Auto Response:

Explanation:

The program encountered a permanent disk I/O error on a COBOL compiler disk work file or the work file is assigned as a remote file.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1009).

Additional Information:

Check the NRD entry to make sure \$SOURCE, \$WORK, \$WORK2, or \$WORK3 is not registered.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1010 Insufficient storage for compilation

Severity: 3 Auto Response: 2

Explanation:

The COBOL program you are trying to compile does not have enough main storage available to complete the compilation.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1010).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1011 Insufficient space in \$WORK disk file

Severity: 3 Auto Response: 2

Explanation:

You tried to compile a COBOL program. However, not enough space is available in the \$WORK disk file. The compiler may have tried to extend the work file for you but there was not enough space on the system to complete the compilation at this time.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1011).

Additional Information:

This message may not be the result of an error. Try running the job again and request more space in the \$WORK file. For more information on increasing the \$WORK file, see the manual, *Programming with COBOL*.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1012 Insufficient space in \$SOURCE disk file

Severity: 3 Auto Response: 2

Explanation:

You tried to compile a COBOL program. However, not enough space is available in the \$SOURCE disk file. The compiler may have tried to extend the work file for you but there was not enough space on the system to complete the compilation.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1012).

Additional Information:

This message may not be the result of an error. Try running the job again and request more space in the \$SOURCE file. For more information on increasing the \$SOURCE file, see the manual, *Programming with COBOL*.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1013 Insufficient space in \$WORK2 disk file

Severity: 3 Auto Response: 2

Explanation:

You tried to compile a COBOL program. However, not enough space is available in the \$WORK2 disk file. The compiler may have tried to extend the work file for you but there was not enough space on the system to complete the compilation.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1013).

Additional Information:

This message may not be the result of an error. Try running the job again and request more space in the \$WORK2 file. For more information on increasing the \$WORK2 file, see the manual, *Programming with COBOL*.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1014 Insufficient space in \$WORK3 disk file

Severity: 3 Auto Response: 2

Explanation:

You tried to compile a COBOL program. However, not enough space is available in the \$WORK3 disk file. The compiler may have tried to extend the work file for you but there was not enough space on the system to complete the compilation.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1014).

Additional Information:

This message may not be the result of an error. Try running the job again and request more space in the \$WORK3 file. For more information on increasing the \$WORK3 file, see the manual, *Programming with COBOL*.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1015 Insufficient storage for XREF processing

Severity: 3 Auto Response: 2

Explanation:

A cross-reference listing was requested for this COBOL program compilation. However, there is insufficient storage available to complete cross-reference processing. This message is preceded by message CBL--1021 or CBL--1022.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1015).

Additional Information:

Increase the region size for the job, or remove the XREF option from the PROCESS statement.

If, after increasing the region size, this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1016 Subprogram name table is larger than 100 names

Severity: 3 Auto Response: 2

Explanation:

The program is processing a CALL statement to permit communications between a main COBOL program and one or more COBOL or other language subprograms; however, the name list that contains the subprogram names contains more than the maximum 100 names.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1016).

Additional Information:

The maximum number of names allowed in a name list is 100.

If, after reducing the name list to 100, this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1017 Too many nested IF statements

Severity: 3 Auto Response: 2

Explanation:

This condition may occur because of a long or complex IF statement. Simplifying the IF statement may eliminate this error.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1017).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1019 C or E level diagnostics detected

Severity: 3 Auto Response: 0

Explanation:

This error resulted because the LET option was not specified in the PROCESS statement. The System/36 COBOL compiler issued error messages that are either C-level (conditional) or E-level (error) messages.

To continue, enter option 0 or option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1019).

User Response:

Option 0: The job continues. The unlinked module is passed to the overlay linkage editor; however, the linkage editor may not be able to properly link the program.

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —1021 Insufficient storage to process all XREF names

Severity: 3 Auto Response: 0

Explanation:

A cross-reference listing was requested for this COBOL program compilation; however, there is not enough storage to process all data names or all procedure names.

To continue, enter option 0, 1, 2, or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1021).

Additional Information:

If, after increasing the region size, this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 0: Compilation and cross-reference processing continues, but all the names may not be included.

Option 1: Compilation continues, but cross-reference processing ends.

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —1022 Insufficient storage to process all XREF references

Severity: 3 Auto Response: 0

Explanation:

A cross-reference listing was requested for this COBOL compilation; however, there was not enough storage to process all references to either data names or procedure names.

To continue, enter option 0, 1, 2, or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1022).

Additional Information:

If, after increasing the region size, this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 0: Compilation and cross-reference processing continues, but all the references may not be included.

Option 1: Compilation continues, but cross-reference processing ends.

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —1023 Option to hold spool file is invalid

Severity: 3 Auto Response: 1

Explanation:

When the COBOLONL procedure is running or if the CRT parameter is specified for the COBOLC procedure, print spooling must be active.

To continue, enter option 1 or 3. If the problem continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-1023).

User Response:

Option 1: With the COBOLC procedure, this option changes the compiler listing option to PRINT, and the compilation continues.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —1098 Enter dump limits

Severity: 3 Auto Response:

Explanation:

This message does not indicate an error. The COBOL compiler CDEBUG facility with the STOP option specified is running.

Give the programmer the message identification code (CBL-1098).

Additional Information:

At this time, enter the dump limits in the form LLLLHHHH. LLLL is the lower-limit 4-digit address, and HHHH is the upper-limit 4-digit address. Entering FF ends the debug prompting. Entering FE ends the compilation.

User Response:

At this time, enter the dump limits in the form LLLLHHHH. LLLL is the lower-limit 4-digit address, and HHHH is the upper-limit 4-digit address. Entering FF ends the debug prompting. Entering FE ends the compilation.

CBL —1099 Unknown terminal compiler error

Severity: 3 Auto Response:

Explanation:

The program encountered an unknown terminal compiler error. The error code is identified in the message. Record this number for your programmer.

To continue, enter option 2. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-1099).

Additional Information:

This message is also printed on the compiler output listing.

If this error continues to occur, fill out *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

CBL —2000 COBOL procedure improperly evoked

Severity: 3 Auto Response:

Explanation:

A COBOL procedure was called by a JOBQ command or a //EVOKE statement; however, a source program name was not provided.

To continue, enter option 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-2000).

Additional Information:

If this message is sent to the system console, the first 2 characters of the job name, located to the right of the message, indicate the sending display station id.

If this error continues to occur, fill out a *Problem Summary Form* and notify your program support representative.

User Response:

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —2020 SEU must exist to run this procedure

Severity: N/A Auto Response: N/A

Explanation:

The source entry utility (SEU) was requested, but the SEU procedure is not currently installed on your system (in #LIBRARY). Give the person in your installation who is responsible for installing your program products the message identification code (CBL-2020).

User Response:

Option 3: The job is canceled.

CBL —2021 DSU or SEU must exist to run this procedure

Severity: N/A Auto Response: N/A

Explanation:

Either the source entry utility (SEU) or the development support utility (DSU) was requested, but the SEU or DSU procedure is not currently installed on your system (in #LIBRARY). Give the person in your installation who is responsible for installing your program products the message identification code (CBL-2021).

User Response:

Option 3: The job is canceled.

CBL —3001 Subscripted reference too large, negative, or zero

Severity: 5 Auto Response:

Explanation:

This is a run-time error. A subscripted reference is incorrect.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3001).

Additional Information:

The address calculated for a subscripted reference is incorrect for one of the following reasons:

- The subscripted reference is over 64K bytes (hex FFFF).
- The subscripted reference is offset more than 32K bytes (hex 7FFF) from the table.
- The subscripted reference was detected as 0 or negative.

To find out which source statement caused the error, compare the address given in message CBL-3018 with the address in the Procedure Division map given by the OFFSET option of the PROCESS statement.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3014 Invalid number of parameters in a CALL statement

Severity: 3 Auto Response:

Explanation:

The call statement contains an error. The number of parameters specified in the CALL statement does not equal the number of parameters given in the Procedure Division of the called program. The maximum number of parameters allowed is 15, but the number of parameters specified in the CALL statement must be the same as that specified in the Procedure Division.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the programmer the message identification code (CBL-3014).

Additional Information:

If this error continues to occur, fill out *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3015 Cannot execute an ACCEPT stmt after /* terminator

Severity: 3 Auto Response:

Explanation:

The program encountered an end-of-data character (/*) while the input data for the ACCEPT statement was processing. Following the end-of-data characters was another ACCEPT statement from the same input source. The second ACCEPT statement is incorrect because of the end-of-data condition.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3015).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3017 COBOL STOP literal displayed

Severity: 3 Auto Response:

Explanation:

A COBOL STOP LITERAL statement was encountered. The first 50 characters of the literal on the COBOL STOP LITERAL statement are displayed as the message.

To continue, enter option 0, 2, or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3017).

Additional Information:

This literal is displayed at the user program display station if the program has an attached display station, or at the system console if there is not an attached display station.

User Response:

Option 0: The job continues with the next statement in the sequence.

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3018 Error at [main storage address]

Severity: N/A Auto Response: N/A

Explanation:

An error occurred in the COBOL program. The main storage address where the error occurred is given in the message text. Another message will be issued following this message. Note this address and refer to it for the recovery procedure.

Additional Information:

The main storage address identified in the message may be compared with the address on a Procedure Division map of the program to determine which source statement encountered the error. To obtain a Procedure Division map, recompile using the PROCESS statement OFFSET option.

CBL —3023 Enter required ACCEPT data

Severity: N/A Auto Response: N/A

Explanation:

This is a prompting message. The COBOL program is running an ACCEPT statement.

Enter the required ACCEPT data. If the ACCEPT statement is issued from the system console, the data must be preceded by the reply ID number associated with the prompt.

For more information about the COBOL ACCEPT statement, refer to the manual, *Programming with COBOL*.

CBL —3024 **stno = #####, preceding were #####

Severity: N/A Auto Response: N/A

Explanation:

A COBOL program is running with TRACE statements. The current and preceding statement numbers are given in the message.

The program continues to run, displaying statement numbers for statements that ran while TRACE was active.

For more information about the COBOL TRACE statement, refer to the manual, *Programming with COBOL*.

CBL —3025 Enter additional ACCEPT data

Severity: 3 Auto Response:

Explanation:

This is the second prompting message of an ACCEPT statement. The length of the ACCEPT identifier exceeds the input record size, so another record is required to fill the identifier.

Enter the additional ACCEPT data. If the ACCEPT statement is issued from the system console, the data must be preceded by the reply ID number associated with the prompt.

For more information about the COBOL ACCEPT statement, refer to the manual, *Programming with COBOL*.

**CBL —3026 Enter object-time debug switch, 0-on
1-off**

Severity: 3 Auto Response:

Explanation:

This is a prompting message. A COBOL program with debug is running. Enter the debug switch value (0 = on, 1 = off).

User Response:

Option 0: Object time debugging is activated. The job step continues running.

Option 1: Object time debugging is not activated, but the job step continues running.

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —3027 'OCCURS DEPENDING ON' variable is 0, > max, or neg

Severity: 3 Auto Response:

Explanation:

An error occurred when the program was run while using the OCCURS clause. An index or subscript referenced a variable-length table. The variable that determines the length of this table is 0, negative, or greater than the maximum size of the table.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3027).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3030 SORT/MERGE operation unsuccessful

Severity: 3 Auto Response:

Explanation:

This is a programming error.

A COBOL program tried a SORT/MERGE operation and encountered an error. A previous message indicated the nature of the error that was encountered during the operation.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3030).

Additional Information:

If this error continues to occur, fill out *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3031 SORT/MERGE control field length exceeds maximum

Severity: 3 Auto Response:

Explanation:

An error occurred while the program was using SORT/MERGE statements. The sum of the lengths of the control fields specified in the requested SORT/MERGE statement exceeds 256 bytes.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3031).

Additional Information:

For more information about SORT/MERGE statements, see the manual, *Programming with COBOL*.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —3032 Read to an unopened or incorrectly opened MCR file

Severity: 3 Auto Response: 0

Explanation:

Either COBOL subroutine CBMICR or CBEMCR attempted to satisfy a read request for a MCR file, but the file was not open. This was caused by either the read request not being preceded by an open request, or by doing a read request after an end of file condition occurred.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3032).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3033 Invalid array given for MCR file open

Severity: 3 Auto Response: 0

Explanation:

Either COBOL subroutine CBMICO or CBEMCO was unable to satisfy an open request for a MCR file. The array named in the COBOL CALL statement may not be formatted correctly, the length given for the stacker array may be incorrect, or the length given for the I/O buffer may not be correct.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3033).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is saved, and the job can continue with the next step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3034 Insufficient space for SORT/MERGE sequence specs

Severity: 3 Auto Response:

Explanation:

The sequence specifications required for the requested sort/merge operation exceed the maximum allowed when the loadable sort interface is used.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3034).

Additional Information:

Reduce the number of sort keys, or use the sort facility independent of COBOL.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3035 I/O Op failed, required error handler is missing

Severity: 3 Auto Response: 1

Explanation:

An error occurred during an I/O operation on a PRINTER or DISK file. According to the return code for that operation, the error handler cannot properly identify the situation. For example, an end-of-file condition occurred and no AT END clause or USE procedure was coded.

To continue, enter option 0, 2, or 3.

Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3035).

Additional Information:

Specify a USE procedure for each file so that corrective action can be taken in the case of an unsuccessful file operation.

Even when you specify a USE procedure, this error can occur if you try to access a disk file or printer file that has *not* been opened in the program. Make sure all your files are open before your program uses them.

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 0: The job continues, but the requested file operation will not be successful. It is the responsibility of the user to detect the situation.

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3036 Non-requester used to ACCEPT/DISPLAY Local Data

Severity: 5 Auto Response:

Explanation:

A data terminal ID was specified for accepting or displaying the local data area. This function is allowed only for requesting terminals or sessions.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3036).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3037 Improper BOOLEAN value in INDICATOR item

Severity: 5 Auto Response:

Explanation:

A WRITE request to a file with transaction organization included an INDICATOR phrase referencing a data item that contains something other than a correct Boolean value.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3037).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3039 Terminal associated with UPSI switch not found

Severity: 5 Auto Response:

Explanation:

You tried to use the SET source statement in a COBOL program to set a user program switch (UPSI) for a display station, but the requested terminal ID was not a valid display station for the program.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3039).

Additional Information:

The SET UPSI-switch TO ON or OFF source statement modifies the external switches associated with a display station. The terminal ID is obtained from the last read operation from a requestor. When the SET statement was run, one of the following situations occurred:

- The requested display station was not attached to the program.
- The requested display station was a data terminal.
- The requested display station could not be found in the system configuration.
- The ID given was for a printer device.

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3042 Invalid parameters given for call to CBREMV or CBINST

Severity: 3 Auto Response: 0

Explanation:

The number of parameters used in the CALL to subroutine CBREMV or CBINST is not the number the subroutine is expecting.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3042).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is saved, and the job can continue with the next step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3200 Accessing a TRANSACTION file that is not opened

Severity: 3 Auto Response:

Explanation:

An error occurred when you tried to access a TRANSACTION file that was closed or was never successfully opened.

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3200).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —3201 Multiple TRANSACTION files error

Severity: 3 Auto Response:

Explanation:

You tried to do one of the following:

- Access a TRANSACTION file that was not currently opened
- Open another TRANSACTION file in a program where one was currently opened

To continue, enter option 2 or 3. Notify the programmer responsible for maintaining this job and give the message identification code (CBL-3201).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 2: The job step is ended. Any data created up to this point is preserved, and the job can continue with the next job step.

Option 3: The job is canceled. Data created by previous steps in this job is saved. Record changes, additions, and deletions made by this step are also saved, but any files created by this step are lost.

CBL —4001 Source member not found in specified library

Severity: N/A Auto Response: N/A

Explanation:

See *Options*.

If you do not know the name of a correct source member, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4001).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter the name of a correct source member in a correct library on the display.

CBL —4002 Input library not found

Severity: N/A Auto Response: N/A

Explanation:

You entered an incorrect input library name.

See *Options*.

If you do not know the name of the input library, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4002).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter the correct name of input library on the display.

Option 0: Selecting this option causes the default of the current library to be used.

CBL —4003 Output library not found

Severity: N/A Auto Response: N/A

Explanation:

You entered an incorrect output library name.

See *Options*.

If you do not know the name of the correct output library name to use notify the programmer responsible for maintaining this job and give the message identification code (CBL-4003).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter the correct output library name on the display.

Option 0: Selecting this option causes a default to the input library.

CBL —4004 View source option is invalid

Severity: N/A Auto Response: N/A

Explanation:

The character you entered for the create/change source option was not Y (yes), N (no), or blank. Reenter proper value.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4004).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

CBL —4005 Compiler listing option is invalid

Severity: N/A Auto Response: N/A

Explanation:

The value you entered for the compiler listing option was not CRT, PRINT, NOPRINT, or a blank.

See Options.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4005).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter CRT, PRINT, PRINT, or NOPRINT for the compiler listing option, or a blank on the display.

Option 0: Selecting this option causes the default value, PRINT, to be used.

CBL —4006 Size specified for compiler workfiles is too small

Severity: N/A Auto Response: N/A

Explanation:

You specified a 0 (zero) or a negative number as the size for the compiler work files.

See Options.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4006).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter a value greater than 0 (zero) for the size of compiler work files on the display.

Option 0: Selecting this option causes the default value of 40 blocks to be used.

CBL —4007 Source print override is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than SOURCE, NOSOURCE, or a blank for the source print override.

See Options.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4007).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter SOURCE, NOSOURCE, or a blank for the source print override option on the display

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4008 Source debug override is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than DEBUG, NODEBUG, or a blank for the source debug override option.

See Options.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4008).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter DEBUG, NODEBUG, or a blank for the source debug override option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4009 Data division map override is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than MAP, NOMAP, or a blank for the Data Division map override option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4009).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter MAP, NOMAP, or a blank for the Data Division map override option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4010 Cross-reference listing option is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than XREF, NOXREF, or a blank for the cross-reference listing option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4010).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter XREF, NOXREF, or a blank for the cross-reference listing option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4011 Procedure division map override is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than OFFSET, NOOFFSET, or a blank for the Procedure Division map override option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4011).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter OFFSET, NOOFFSET, or a blank for the Procedure Division map override option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4012 Copy library not found

Severity: N/A Auto Response: N/A

Explanation:

The library name you entered for the copy library name was not found.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4012).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter a correct library name for the copy library name option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4013 Subroutine library not found

Severity: N/A Auto Response: N/A

Explanation:

The library name you entered for the subroutine library name override option was not found.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4013).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue, and then enter a correct library name for the subroutine library name override option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4014 Never-ending program option is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than NEP, NONEP, or a blank for the never-ending program option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4014).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter NEP, NONEP, or a blank for the never-ending program option on the display.

Option 0: Selecting this option causes the default value of NONEP to be used.

CBL —4015 Size entered for MRT is a negative number

Severity: N/A Auto Response: N/A

Explanation:

You specified a negative number for the maximum number of requesting display stations option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4015).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue, and then enter a positive number for the maximum number of display stations on the display.

Option 0: Selecting this option causes the default value of 0 to be used.

CBL —4016 View compiler listing option is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than Y or N for the view compiler listing option. Reenter proper value.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4016).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

CBL —4017 Continue option is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than Y or N for the continue procedure option. Reenter proper value.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4017).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

CBL —4018 Diagnosed source member option is invalid.

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than DSM, NODSM, or a blank for the diagnosed source member option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4018).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter DSM, NODSM, or a blank as the value for the diagnosed source member option on the display.

Option 0: Selecting this option causes the default value of NODSM to be used.

CBL —4019 Object module override is invalid.

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than OBJECT, NOOBJECT, or a blank for the module override option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code(CBL-4019).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter OBJECT, NOOBJECT, or a blank for the object module override option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4020 COBOL spool file is in error or cannot be found.

Severity: 5 Auto Response:

Explanation:

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4020).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option 1: Select this option to attempt to retrieve the spool file again.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —4021 Parameter not recognized - must be P or S

Severity: N/A Auto Response: N/A

Explanation:

You entered an incorrect SEU member type. The SEU member type must be a P (procedure) or S (source).

Additional Information:

If this error continues to occur after corrections are made, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue, and then enter P for procedure or S for source on the display.

Option 0: Selecting this option causes the default of S (source) to be used.

Option 3: The job is canceled. Data created by previous steps in this job is saved, but data created by this step is lost.

CBL —4022 Memory resident overlay override is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than MRO, NOMRO, or a blank for the memory resident overlay option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4022).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter MRO, NOMRO, or a blank for the memory resident overlay option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4023 Linkedit override is invalid

Severity: N/A Auto Response: N/A

Explanation:

You specified a value other than LINK, NOLINK, or a blank for the link-edit option.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4023).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter LINK, NOLINK, or a blank for the link-edit option on the display.

Option 0: Selecting this option causes the default value of blank to be used.

CBL —4024 MRO can not be specified with NOLINK

Severity: N/A Auto Response: N/A

Explanation:

You specified a value MRO for memory resident overlay option while NOLINK is specified for link-edit option. This combination is not valid.

See *Options*.

If this error continues to occur, notify the programmer responsible for maintaining this job and give the message identification code (CBL-4024).

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form* and notify your programming service representative.

User Response:

Option H: Select this option to continue and then enter NOMRO or a blank for memory resident overlay option, or enter LINK for link-edit option on the display.

Option 0: Selecting this option causes the default value of blank for memory resident overlay option to be used.

CBL —4026 Data dictionary not found

Severity: N/A Auto Response: N/A

Explanation:

The data dictionary name you entered for the data dictionary option was not found.

Additional Information:

If this error continues to occur, fill out a *Problem Summary Form*, and notify your programming service representative.

User Response:

Option H: Select this option to continue and enter the correct data dictionary name on the display.

Option 0: Select this option to continue and use the data dictionary name specified. This data dictionary must be created before the program is executed.

Option 3: The job is canceled. Data created up to this step is saved.

Your comments can help us produce better manuals. Please take a few minutes to evaluate this manual as soon as you become familiar with it. Circle Y (Yes) or N (No) for each question that applies. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

FINDING INFORMATION

- Y N Is the table of contents helpful?
What would make it more helpful?

- Y N Is the index complete?
List specific terms that are missing.

- Y N Are the chapter titles and other headings meaningful?
What would make them more meaningful?

- Y N Is information organized appropriately?
What would improve the organization?

- Y N Does the manual refer you to the appropriate places for more information?
List specific references that are wrong or missing.

UNDERSTANDING INFORMATION

- Y N Is the purpose of this manual clear?
What would make it clearer?

- Y N Is the information explained clearly?
Which topics are unclear?

- Y N Are the examples clear?
Which examples are unclear?

- Y N Are examples provided where they are needed?
Where should examples be added or deleted?

- Y N Are terms defined clearly?
Which terms are unclear?

- Y N Are terms used consistently?
Which terms are inconsistent?

- Y N Are too many abbreviations and acronyms used?
Which ones are not understandable?

- Y N Are the illustrations clear?
Which ones are unclear?

USING INFORMATION

- Y N Does the information apply to your situation?
Which topics do not apply?

- Y N Is the information accurate?
What information is inaccurate?

- Y N Is the information complete?
What information is missing?

- Y N Is only necessary information included?
What information is unnecessary?

- Y N Are the examples useful models?
What would make them more useful?

- Y N Is the format of the manual (shape, size, color) effective?
What would make the format more effective?

OTHER COMMENTS

Use the space below for any other opinions about this manual or about the entire set of manuals for this system.

YOUR BACKGROUND

- What is your job title?

- What is your primary job responsibility?

- How many years have you used computers?

- Which programming languages do you use?

- How many times per month do you use this manual?

- Your name _____
- Company name _____
- Street address _____
- City, State, ZIP _____

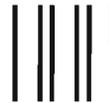
No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape

Please do not staple

Fold and tape



NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Information Development
Department 532
Rochester, Minnesota, U.S.A. 55901



Fold and tape

Please do not staple

Fold and tape



International Business Machines Corporation

IBM System/360 COBOL Messages File No. S36-24 Printed in U.S.A. SC21-7941-3

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s):

Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name _____

Company or
Organization _____

Address _____

No postage necessary if mailed in the U.S.A.

City _____ State _____ Zip Code _____

Cut Along Line

Fold and tape

Please do not staple

Fold and tape



NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL		
FIRST CLASS	PERMIT NO. 40	ARMONK, N. Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Information Development
Department 245
Rochester, Minnesota, U.S.A. 55901

Fold and tape

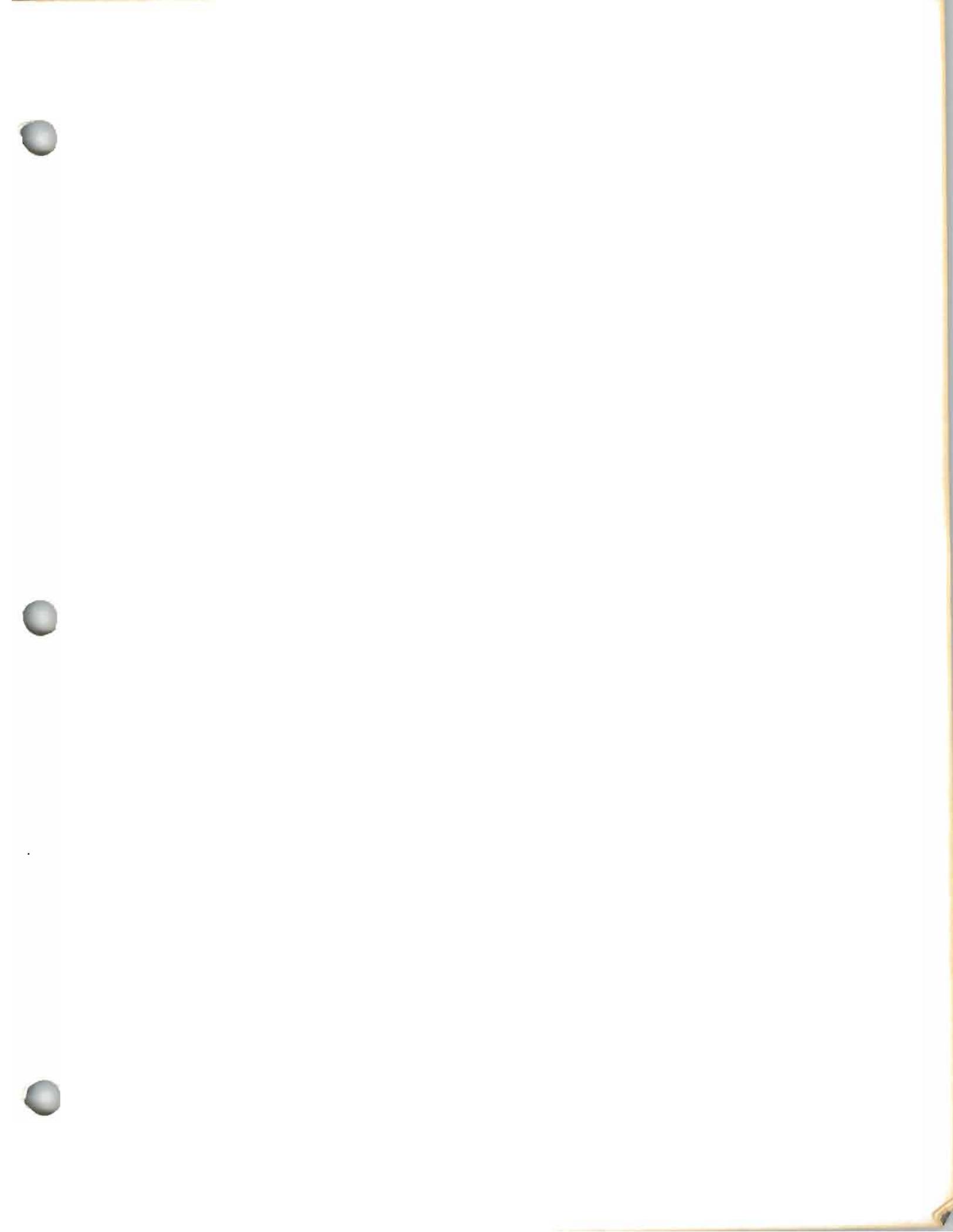
Please do not staple

Fold and tape



International Business Machines Corporation

IBM System/36: J60L Messages File No. S36-24 Printed in U.S.A. SC21-7941-3





COBOL Messages

International Business Machines Corporation

File Number
S36-24

Order Number
SC21-7941-3

Part Number
59X3993

Printed in U.S.A.

SC21-7941-03

