SC33-0035-2
File No. S360/S370-29

DOS
PL/I Transient Library: Messages

Program Product

Program Number 5736-LM5
(This program product is available as
part of composite package 5736-PL3)

IBM

Third Edition (August, 1977)

This edition is a reprint of SC33-0035-1 incorporating changes released in Technical Newsletters SN33-6120 (dated June, 1974), SN33-6126 (dated December, 1974), and SN33-6171 (dated October, 1976).

This edition applies to Version 1, Release 5, Modification 0 of the DOS PL/I Transient Library, Program Product 5736-LM5, and to any subsequent version, release, and modification.

Information in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters. Before using the publication, consult the latest IBM System/370 Bibliography, GC20-0001, and the technical newsletters that amend the bibliography, to learn which edition and technical newsletters are applicable and current.

Requests for copies of IBM publications should be made to the IBM branch office that serves you.

Comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California 95150. All comments and suggestions become the property of IBM.

Preface

This publication lists the execution-time messages that can be produced by the DOS PL/I Transient Library when this library is used during the execution of a PL/I program compiled by the DOS PL/I Optimizing Compiler. Most of these messages are accompanied by additional information intended to illustrate the detected condition and to point to the appropriate corrective action.

The compile-time(IEL) messages produced by the DOS PL/I Optimizing Compiler are listed in a separate publication: DOS PL/I Optimizing Compiler: Messages, Order No. SC33-0021.

Associated Publications

DOS PL/I Optimizing Compiler:

Specifications, Order No. GC33-0016

General Information, Order No. GC33-0004

Programmer's Guide, Order No. SC33-0008

Execution Logic, Order No. SC33-0019

Installation, Order No. SC33-0020

Reference Data, Order No. SX33-6001

Language Reference Manual, Order No. GC33-0005

Availability of Publications

The availability of a publication is indicated by its use key, the first letter in the order number. The use keys are:

G - General: available to users of IBM systems, products, and services without charge, in quantities to meet their normal requirements; can also be purchased by anyone through IBM branch offices.

S - Sell: can be purchased by anyone through IBM branch offices.

Contents

Introduction

Execution-time messages are printed on the device associated with the file SYSPRINT. If SYSPRINT is unusable the message will be printed on the operator's console. However, any message associated with the system action for the CHECK condition or the COPY or SNAP options will not be transmitted to the operator's console.

Messages are printed at execution time when:

1. An error occurs for which there is no specific on-condition in PL/I. (A message is printed and the ERROR condition is raised).
2. An on-condition is raised for which there is no on-unit in the program and for which the standard system action includes printing an execution-time message.

Format of Execution-time Messages

Execution-time messages have the following format:

message-number - ONCODE=nnnn 'condition-name' CONDITION RAISED - text - location message

Each message number is of the form IBMnnn, where "IBM" indicates that the message is a PL/I execution-time message, and "nnn" is the number of the message. The final character "I" or "A" indicates to the operator whether the message is informatory or whether he should take some action.

The on-code is that for the condition name that follows. The text "ONCODE=" and the actual on-code generated by the system only appear when the message is printed; they are not listed in the message texts in this publication.

The condition name is the PL/I on-condition raised in association with the exceptional or error condition that caused the message to be printed. The condition name and the text "CONDITION RAISED" are not normally listed in the message texts in this publication, although they are always included when the message is printed. The text of the message may consist of one or two parts, according to the requirements of the individual message. The parts are a fixed part and a variable part. The fixed part only of each message is given in this publication. The variable part of a message can be one of the following:

- a condition name
- an entry name
- a file name

according to the requirements of the message.

The nature of the location message will depend on whether the PL/I program was compiled with the GOSTMT option or the NOGOSTMT option. If GOSTMT was specified, the location message will be as follows:

```
  IN STATEMENT nnn AT OFFSET nnnnnn IN PROCEDURE WITH ENTRY entry-name
```

If NOGOSTMT was specified, the location message will be as follows:

```
  AT OFFSET nnnnnn IN PROCEDURE WITH ENTRY entry-name
```

The location messages generated by the system only appear when the messages are printed; they are not listed in the message texts in this publication.

An example of an execution-time message in the form in which it will be printed is given below:

```
IBM037I  ONCODE=0612 'CONVERSION' CONDITION RAISED CONVERSION FROM
         CHARACTER TO BIT ON INPUT AFTER 'TRANSMIT' DETECTED IN
         STATEMENT 207 AT OFFSET 004A8C IN PROCEDURE WITH ENTRY PROG1
```

Before Calling IBM...

Unless the programmer response for the message specifies otherwise, before calling IBM for programming support with regard to an execution-time error:

1. Ensure that the program includes an active ERROR on-unit that includes the statement:

   ```
   CALL PLIDUMP('HB');
   ```

2. Run the program again.

If the problem recurs, ensure that the following are available:

1. Listings of the source program, the object program, and the job control statements for the execution of the program.

2. Relevant data sets.

3. Job stream (job control statements and data) in machine readable form.

The requirements for problem determination and APAR submission are given in the programmer's guide for this compiler.

Execution-time (IBM) Messages

IBM002I INTERRUPT DURING PL/I PROGRAM-MANAGEMENT LIBRARY ROUTINE. PROGRAM TERMINATED

Explanation: An interrupt has occurred during the handling of a PL/I on-condition or error condition in the program management routine or a routine invoked by it. It indicates that a disastrous error has occurred during execution of the program, such as the overwriting of control blocks or sections of code. The program is terminated, and if the DUMP option of the OPTION job control statement is requested for the job, a dump of the partition is produced by the DOS dump program. The most common cause of this type of error is the overwriting of control information by the PL/I program. The program may contain arrays which have exceeded their bounds.

Programmer Response: Refer to chapter on "Program Checkout" in the programmer's guide for this compiler for advice on how to tackle this type of error. The program should be recompiled with STRINGRANGE and SUBSCRIPTRANGE enabled, and reexecuted before calling IBM.

IBM006I NO MAIN PROCEDURE, PROGRAM NOT EXECUTED

Explanation: An attempt has been made to execute a program containing one or more external PL/I procedures, none of which has the MAIN option in its PROCEDURE statement.

Programmer Response: Ensure that the first external PL/I procedure to be invoked has the MAIN option in its PROCEDURE statement.

IBM007A WAITING FOR REPLY TO 'DISPLAY' MESSAGE

Explanation: The purpose of this message is to prompt the operator to make an appropriate reply. This message is always issued when a DISPLAY statement with the REPLY option is executed.

IBM008I NO MAIN STORAGE AVAILABLE, NEEDS xxxxxxxx MORE BYTES

Explanation:

1. There is insufficient main storage to begin program execution. The one line message which appears means that xxxxxxxx more bytes of storage must be made available before execution of the main procedure can begin. This does not include the storage required once execution of the main procedure begins.

2. The three line message which appears during program

execution means that a request for xxxxxxxx bytes of storage has been made either by a PL/I statement or by a library module called by a PL/I statement. To satisfy the request, xxxxxxxx more bytes of contiguous free storage must be made available.

Programmer Response:

1. If the message appears during program initialization, the program is not executed.

2. If the message appears during program execution, the program is terminated.

In both cases, control is returned to the caller with a return code of 4012 in Register 15, or a CANCEL macro instruction is issued. The programmer should check the space requirements of the job and either reduce those requirements by amending the program or increase the size of the partition in which the job is to run.

IBM020I 'CONVERSION' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the 'CONVERSION' condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the CONVERSION condition in the program.

IBM021I UNKNOWN SOURCE ATTRIBUTES ON INPUT

Explanation: The CONVERSION condition has been raised within a GET LIST or GET DATA statement with the FILE option. The attributes of the source data cannot be determined. For example:

```
DCL (A,B) CHAR(14);
GET LIST (A,B)
```

Where the input stream contains 'PIG'C,'DOG',--- the condition will be raised when the first item is encountered. The value for ONSOURCE will be: "'PIG'C", and for ONCHAR will be: "C".

IBM022I UNKNOWN SOURCE ATTRIBUTES ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: The CONVERSION condition has been raised after an error has caused the TRANSMIT condition to be raised. For an explanation of the conversion error, see the explanation given for message IBM021I.

IBM023I UNKNOWN SOURCE ATTRIBUTES

Explanation: The CONVERSION condition has been raised within a GET LIST STRING or GET DATA STRING statement. For an explanation of the conversion error, see the explanation given for message IBM021I.

IBM024I CONVERSION FROM F-FORMAT ON INPUT

Explanation: An invalid character has been detected in an F-format input field.

Programmer Response: Include a suitable on-unit in the program to monitor errors in the input data that are revealed by the CONVERSION condition. Use the ONSOURCE and ONCHAR built-in functions to identify the error and the ONSOURCE and ONCHAR pseudovariables to assign a valid numeric value so that the program can continue to be executed normally. Otherwise check that all input is in the correct format before executing the program.

IBM025I CONVERSION FROM F-FORMAT ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: An invalid character has been detected in an F-format input field. A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps given for the preceding message.

IBM027I CONVERSION FROM E-FORMAT ON INPUT

Explanation: An invalid character has been detected in an E-format input field.

Programmer Response: Take the steps advised for conversion errors in message IBM024I.

IBM028I CONVERSION FROM E-FORMAT ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: An invalid character has been detected in an E-format input field. A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps advised for message IBM024I.

IBM029I CONVERSION FROM B-FORMAT ON INPUT

Explanation: An invalid character has been detected in a B-format input field.

Programmer Response: Include a suitable on-unit in the program to monitor errors in the input data that are revealed by the

CONVERSION condition. Use the ONSOURCE and ONCHAR built-in functions to identify the error and the ONSOURCE and ONCHAR pseudovariables to assign a valid bit character so that the program can continue to be executed normally. Otherwise check that all input is in the correct format before executing the program.

IBM031I CONVERSION FROM B-FORMAT ITEM ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: An invalid character has been detected in a B-format input field. A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps given for message IBM029I.

IBM032I CONVERSION FROM 'CHARACTER' TO ARITHMETIC

Explanation: An invalid character has been detected in a character string that is being converted to an arithmetic data type.

Programmer Response: If the error is in the conversion of a PL/I source program constant or in the conversion of a character string created during the execution of the program, correct the source program, recompile it and rerun it.

IBM033I CONVERSION FROM CHARACTER TO ARITHMETIC ON INPUT OR OUTPUT

Explanation: A character which is invalid for conversion to an arithmetic form has been detected in one of the following:

1. An arithmetic constant in a list-directed or data-directed item.
2. A character constant being converted to an arithmetic form in a list-directed or data-directed item.
3. An A-format input field being converted to an arithmetic form.

Programmer Response: Take the steps advised for message IBM024I.

IBM034I CONVERSION FROM CHARACTER ON INPUT AFTER TRANSMIT DETECTED

Explanation: A character which is invalid for conversion to an arithmetic form has been detected in one of the following:

1. An arithmetic constant in a list-directed or data-directed input item.
2. A character constant being converted to an arithmetic form

in a list-directed or data-directed input item.

3. An A-format input field being converted to an arithmetic form.

A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps advised for message IBM024I.

IBM035I CONVERSION FROM 'CHARACTER' TO 'BIT'

Explanation: An invalid character has occurred in a character string that is being converted to a bit string.

Programmer Response: If the error is in the conversion of a PL/I source program constant or in the conversion of a character string created during the execution of the program, correct the source program, recompile it, and rerun it.

IBM036I CONVERSION FROM CHARACTER TO BIT ON INPUT OR OUTPUT

Explanation: A character other than 0 or 1 appears in one of the following:

1. A bit constant in a list-directed or data-directed item.

2. A character constant being converted to bit form in a list-directed or data-directed item.

3. An A format input field being converted to bit form.

4. A B-format input field (excluding any leading or trailing blanks).

IBM037I CONVERSION FROM CHARACTER TO BIT ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: A character other than 1 or 0 appears in one of the following:

1. A bit constant in a list-directed or data-directed input item.

2. A character constant being converted to bit form in a list-directed or data-directed input item.

3. An A-format input field being converted to bit form.

4. A B-format input field (excluding any leading or trailing blanks).

A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps advised for message IBM024I.

IBM038I CONVERSION TO PICTURE CHARACTER STRING

Explanation: A character that does not match the picture specification has occurred in a conversion to a PICTURE character string.

Programmer Response: Ensure that the character string that is to be converted to a PICTURE character string is suitable for the conversion. If necessary, use the ONSOURCE and ONCHAR built-in functions to identify the error and the ONSOURCE and ONCHAR pseudovariables to replace an erroneous character with a character that is valid for conversion.

IBM039I CONVERSION TO PICTURE CHARACTER STRING ON INPUT OR OUTPUT

Explanation: A character that does not match the picture specification has occurred in a stream-oriented item that requires conversion to a PICTURE character string.

Programmer Response: Either ensure that all input data to the program is in the correct format or take the steps given for the preceding message to ensure that the program has adequate error recovery facilities so that it can comment on any invalid data found in its input, and continue processing.

IBM040I CONVERSION TO PICTURE CHARACTER STRING ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: A character that does not match the picture specification has occurred in a stream-oriented input item that requires conversion to a PICTURE character string. A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps advised for message IBM039I.

IBM042I CO VERSION FROM P-FORMAT (ARITH) ON INPUT

Explanation: An edit-directed P-format input item contains a character that does not match the picture specification.

Programmer Response: Either ensure that all input data to the program is in the correct format before executing the program or use the program to check the data. If necessary, use the ONSOURCE and ONCHAR built-in functions to identify the error and the ONSOURCE and ONCHAR pseudovariables to replace an erroneous character with a character that is valid for conversion.

IBM043I CONVERSION FROM P-FORMAT (ARITH) ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: An invalid character has been detected in a P-format (arithmetic) input field. A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps advised for message IBM042I.

IBM045I CONVERSION FROM P-FORMAT (CHAR) ON INPUT

Explanation: An invalid character has been detected in a P-format input item.

Programmer Response: Either ensure that all input data to the program is in the correct format before executing the program or use the program to check the data. If necessary, use the ONSOURCE and ONCHAR built-in functions to identify the error and the ONSOURCE and ONCHAR pseudovariables to replace an erroneous character with a character that is valid for conversion.

IBM046I CONVERSION FROM P-FORMAT (CHAR) ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: An invalid character has been detected in a P-format (character) input item. A transmission error has also occurred; the conversion error may be directly attributable to the transmission error.

Programmer Response: If the conversion error recurs after the transmission error has been eliminated, take the steps advised for message IBM045I.

IBM050I RESTRICTED STREAM CONVERSION

Explanation: The combination of attributes of the character string source and the target variable require a conversion that is not supported in a program compiled with the LIMSCONV option.

IBM051I RESTRICTED STREAM CONVERSION ON INPUT

Explanation: The combination of attributes of the stream source and the target variable require a conversion that is not supported in a program compiled with the LIMSCONV option.

IBM052I RESTRICTED STREAM CONVERSION ON INPUT AFTER 'TRANSMIT' DETECTED

Explanation: The combination of attributes of the stream source and the target variable require a conversion that is not supported in a program compiled with the LIMSCONV option. A transmission error has also occurred and the attributes of the stream source may have been determined erroneously as a result.

IBM100I 'NAME' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a signal statement to raise the NAME condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the NAME condition in the program.

IBM101I INVALID ELEMENT-VARIABLE IN STREAM FOR 'GET FILE DATA'

Explanation: The NAME condition is raised immediately any of the following errors is detected:

1. When an identifier in the input stream has no counterpart in the data list of the GET statement, or when the GET statement has no data list and an identifier that is not known in the block is encountered in the stream.
2. When invalid blank characters are found within an identifier in the input stream.
3. When the name field, or part of a qualified name, is omitted.
4. When there are more than 256 characters in a fully-qualified name.
5. When blanks are found within an array subscript other than between the optional sign and the decimal digits.
6. When an array subscript is missing or indicates too many dimensions.
7. When a value in a subscript is not a decimal digit.
8. When the subscript is beyond the declared range of subscripts for a particular array.
9. If the left-parenthesis is missing after the name of an array.
10. If a character other than "=" or a blank is found after a right-parenthesis that delimits an array subscript in the input stream.
11. If the end-of-file or a non-blank delimiter is found before "=" in an item in the input stream.

Programmer Response: Use the DATAFIELD condition built-in function in a NAME on-unit to obtain the invalid data item.

IBM120I 'RECORD' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: A SIGNAL statement to raise the RECORD condition has been executed. There was no on-unit for this condition.

Programmer Response: Supply an on-unit for the RECORD condition or remove the SIGNAL statement.

IBM121I LENGTH OF RECORD VARIABLE LESS THAN RECORD LENGTH

Explanation: A record in a record-oriented input/output statement is longer than the record variable.

This message is produced for records that are longer than the associated PL/I variable. For a READ statement, the record is truncated to the length of the variable in the INTO option. For a LOCATE statement (F-format records only), a buffer is not allocated. For a WRITE statement (F-format records only), the record is transmitted with additional padding bytes to make up the length. The contents of the padding bytes are undefined. For a REWRITE statement, the record is replaced by the shorter record made up to the correct length with the appropriate number of padding bytes, the contents of which are undefined.

Programmer Response: Either supply an on-unit for the RECORD condition so that the program can continue to be executed, or modify the program to make the length of the record variable the same as the length of the records on the data set. The language reference manual for this compiler gives details of how such records are handled when the RECORD condition is raised.

IBM122I LENGTH OF RECORD VARIABLE GREATER THAN RECORD LENGTH

Explanation: A record in a record-oriented input/output statement is shorter than the record variable.

This message is produced for records that are shorter than the associated PL/I variable. For a READ statement using F-format records and a fixed-length variable in the INTO option, the excess bytes in the variable are undefined. For a LOCATE statement, where the maximum length of the records is less than the length of the PL/I variable, the buffer is not allocated. For a WRITE statement, the variable in the FROM option is longer than the maximum length of the records and is truncated to the maximum record length. For a REWRITE statement, the variable in the FROM option is longer than the record it is to replace, and is truncated to the length of this record.

Programmer Response: Either supply an on-unit for the RECORD condition so that the program can continue to be executed, or modify the program to make the length of the record variable the same as the length of the records on the data set. The language reference manual for this compiler gives details of how such records are handled when the RECORD condition is raised.

IBM123I 'WRITE' OR 'LOCATE' VARIABLE HAS ZERO LENGTH

Explanation: A WRITE or REWRITE statement has attempted to transmit a record variable of zero length, or a LOCATE statement has attempted to obtain buffer space for a zero length record variable.

Programmer Response: Modify the program to ensure that the varying-length string used as a record variable is not a null string when the WRITE, REWRITE or LOCATE statement is executed.

IBM124I ZERO LENGTH RECORD READ FROM REGIONAL DATA SET

Explanation: A record of zero length has been read from a REGIONAL data set associated with a DIRECT file. A zero-length record on a direct-access device indicates the end of the data set. However, the message above will only be produced if the data set has been created incorrectly.

Programmer Response: Check that the data set was created correctly as a regional data set. Recreate the data set if necessary and possible. Check also that the record has been accessed with a key that is valid for the data set.

IBM125I 'WRITE' or 'LOCATE' VARIABLE TOO SHORT TO CONTAIN EMBEDDED KEY.

Explanation: A WRITE or REWRITE statement has attempted to transmit, or a LOCATE statement has attempted to allocate buffer space for, a record variable too short to contain the data set embedded key. For a WRITE or REWRITE statement, no transmission takes place; for a LOCATE statement, a buffer is not allocated.

IBM140I 'TRANSMIT' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the TRANSMIT condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the TRANSMIT condition in the program.

IBM141I UNCORRECTABLE ERROR IN OUTPUT

Explanation: Data management has detected an uncorrectable error while transmitting output data between main storage and an external storage device. The condition is raised on the completion of a WRITE, REWRITE, or LOCATE statement. For BUFFERED files, this condition may be raised only after the execution of several I/O statements.after the I/O statement which transmitted the record. No further processing of an OUTPUT file other than a file associated with a unit record device can occur. Processing of an UPDATE file may continue. For INDEXED data sets, the condition can also occur while searching through the indexes or tracing an overflow record.

Programmer Response: If the error recurs, obtain a dump of the input/output buffer areas by using PLIDUMP in a TRANSMIT on-unit. See the programmer's guide for details of PLIDUMP. The resultant output, together with all relevant listings and data sets should be preserved for examination by IBM.

IBM142I UNCORRECTABLE ERROR IN INPUT

Explanation: Data management has detected an uncorrectable error while transmitting input data between main storage and an external storage device. The condition is raised on the completion of a READ or REWRITE statement for each record in the block that contains the error and for every item transmitted by GET statements from a block that contains the error. The contents of the record or data item are undefined. However, processing of subsequent records in the input file can be continued. For INDEXED data sets, the condition can be raised while searching the indexes or tracing an overflow record.

Programmer Response: If the error recurs, obtain a dump of the input/output buffers by using PLIDUMP in a TRANSMIT on-unit. See the programmer's guide for details of PLIDUMP. The resultant output, together with all relevant listings and data sets should be preserved for examination by IBM.

IBM143I UNREADABLE OMR DATA

Explanation: One of more OMR columns contain a marginal mark, weak mark, or poor erasure that cannot be read. The condition is raised on completion of the READ operation for the card. A X'3F' character is substituted for unreadable characters, and also put in the last byte of the record. The card is stacker selected to the alternate stacker.

IBM144I WRITE ERROR IN INDEX SET.

Explanation: Data management has detected a physical error whilst attempting to write on the index set of a VSAM KSDS. The condition is raised on the completion of a WRITE, REWRITE, LOCATE, or DELETE statement. No further processing of an OUTPUT file can occur. Processing of an UPDATE file may continue.

IBM145I READ ERROR IN INDEX SET.

Explanation: Data management has detected a physical error whilst attempting to read from the index set of a VSAM KSDS. The condition is raised on the completion of a READ, WRITE, REWRITE, LOCATE, or DELETE statement. No further processing of an OUTPUT file can occur. Processing of an UPDATE file may continue. If the error occurs on a READ statement, no data is transferred to the record variable. For sequential access, data set positioning may be lost, causing a subsequent READ without KEY to raise ERROR (see message IBM831I)

IBM146I WRITE ERROR IN SEQUENCE SET.

Explanation: Data management has detected a physical error whilst attempting to write on the sequence set of a VSAM KSDS. The condition is raised on the completion of a WRITE, REWRITE, LOCATE, or DELETE statement. No further processing of an OUTPUT file can occur. Processing of an UPDATE file may continue.

IBM147I READ ERROR IN SEQUENCE SET.

Explanation: Data management has detected a physical error whilst attempting to read from the sequence set of a VSAM KSDS. The condition is raised on the completion of a READ, WRITE, REWRITE, LOCATE, or DELETE statement. No further processing of an OUTPUT file can occur. Processing of an UPDATE file may continue. If the error occurs on a READ statement, no data is transferred to the record variable. For sequential access, data set positioning may be lost, causing a subsequent READ without KEY to raise ERROR (see message IBM831I)

IBM160I 'KEY' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the KEY condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the KEY condition in the program.

IBM161I KEY SPECIFIED CANNOT BE FOUND

Explanation: A READ or REWRITE statement specified a recorded key which could not be found on the data set. In the case of an INDEXED data set, the key in error is either higher than the highest level index or the record is not in the prime area or the overflow areas of the data set. In the case of a DIRECT file associated with a data set with REGIONAL organization, the key in error is not in the specified region.

Programmer Response: Determine why the key was incorrect and modify the program or the data set as necessary. Use of the ONKEY built-in function in a KEY on-unit will aid in determining the value of the erroneous key.

IBM162I KEY SPECIFIED ALREADY IN USE ON DATA SET

Explanation: In the case of data set with INDEXED organization, an attempt has been made to transmit a keyed record to a data set which already holds a record with the same key. In the case of a data set with REGIONAL(1) organization that is being created sequentially, an attempt has been made to transmit a record to a region that already contains a record.

Programmer Response: Either check the validity of the data that is being processed before executing the program or use the program to check the data. Use of the ONKEY built-in function in a KEY on-unit will aid in identifying an erroneous key, in correcting it, and in permitting processing to continue normally.

IBM163I KEY SPECIFIED IS LESS THAN VALUE OF PREVIOUS KEY

Explanation: A key with a value that is less than the value of the preceding key has been detected during the creation or extension of an INDEXED or REGIONAL SEQUENTIAL data set.

Programmer Response: Ensure that the records that are to be written onto an INDEXED or REGIONAL data set that is being created or extended are in the correct ascending key sequence order. Otherwise use a KEY on-unit to comment on the error and, where possible, to permit processing to continue normally.

IBM164I KEY SPECIFIED CANNOT BE CONVERTED TO VALID DATA

Explanation: A WRITE, READ, REWRITE, LOCATE statement for a REGIONAL data set specified a key with a character-string value consisting entirely of blanks or containing characters other than 0-9 or blank as part of the region number.

Programmer Response: Ensure that the key is in the correct format. If necessary, use the ONKEY built-in function in a KEY on-unit to identify the erroneous key. The on-unit can be used to report any such errors and allow processing to continue. Records associated with the erroneous keys can be transmitted in a subsequent run for which the keys have been corrected.

IBM165I KEY SPECIFIED IS INVALID

Explanation: For an INDEXED data set, either the KEY or KEYFROM expression is a null string or an attempt has been made to rewrite a record where the embedded key of the replacement record is not equal to that of the record that is to be overwritten. For a REGIONAL data set, the key specified is a null string.

Programmer Response: As for the previous message.

IBM166I KEY SPECIFIES POSITION OUTSIDE REGIONAL DATA SET

Explanation: A WRITE, READ, or REWRITE statement specifies a key whose relative record or track value exceeds the number of records or tracks respectively for the REGIONAL data set.

Programmer Response: As for message IBM164I.

IBM167I NO SPACE AVAILABLE TO ADD KEYED RECORD

Explanation: For a SEQUENTIAL file associated with an INDEXED data set, an attempt has been made to write or locate a record during the creation or extension of such a data set when the space allocated to the data set is full. For a DIRECT file associated with an INDEXED data set, there is no space in the available overflow areas to accept the overflow record caused by the insertion of a new record by a WRITE statement.

For a DIRECT file associated with a REGIONAL data set, there is no space to add the record in the specified region. Note that the data set is not necessarily full.

Programmer Response: Use the ONKEY built-in function to identify the key value that caused the error. If the key is in error, correct it and recommence the job from the point reached when the error occurred. If the key is correct, organize the data set so that the rejected record can be accommodated.

IBM168I 'KEYFROM' VALUE LIES OUTSIDE KEY RANGE(S) DEFINED FOR DATA SET.

Explanation: A WRITE or LOCATE statement specified a key with a value outside the key ranges specified for the data set when it was defined. (VSAM KSDS).

IBM180I 'ENDFILE' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the ENDFILE condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the ENDFILE condition in the program.

IBM181I 'ENDFILE' CONDITION RAISED

Explanation: The end of an input file has been detected.

Programmer Response: Include an on-unit for the ENDFILE condition for each input file in the program to handle the end-of-file processing.

| IBM182I END OF FILE PREVIOUSLY ENCOUNTERED ON STREAM INPUT

Explanation: The ENDFILE condition was raised when the file mark was encountered but an attempt is being made to read beyond the end of the file. Either an ENDFILE on-unit has been executed and a further attempt to read the file is being made or the end-of-file mark was encountered between items in the data list of the current GET statement.

Programmer Response: If the program contains an ENDFILE on-unit, make sure that it does not attempt to read the file after the ENDFILE condition has been raised for it. If the error occurred during execution of a GET statement with two or more items in the data list, make sure that the GET statement can be completed by providing sufficient data items before the end-of-file mark is encountered.

IBM200I 'UNDEFINEDFILE' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the UNDEFINEDFILE condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the UNDEFINEDFILE condition in the program.

IBM205I I/O ERROR - 'REGIONAL' DATA SET CANNOT BE FORMATTED

Explanation: When a REGIONAL data set was being opened for output the open process could not format it with dummy records for a REGIONAL(1) SEQUENTIAL OUTPUT file or with capacity records for a REGIONAL(3) SEQUENTIAL OUTPUT file or a REGIONAL(1) or REGIONAL(3) DIRECT OUTPUT file. This is due to an incorrectable input/output error that is similar to the type of input/output error that raises the TRANSMIT condition.

Programmer Response: If the error recurs after resubmitting the job, use PLIDUMP to obtain a storage dump and retain all the relevant documentation for study by IBM.

IBM206I 'LINESIZE' OR 'PAGESIZE' OUTSIDE IMPLEMENTATION-DEFINED LIMITS

Explanation: The implementation-defined maximum or minimum for the LINESIZE option of the ENVIRONMENT attribute has been exceeded. For F and U-format records, the maximum is 32,759; for V-format records, the maximum is 32,751. The minimum in all cases is 1.

Programmer Response: Check that the argument to the LINESIZE option is within the prescribed limits. If the argument is a variable, check that it is a FIXED BINARY (31,0) STATIC variable that was correctly initialized before the file was opened.

IBM208I WRONG BLOCKSIZE OR RECORD LENGTH SPECIFIED

Explanation: One of the following errors may have occurred:

1. The block size is less than the record length.

2. For FB-format records, the blocksize is not a multiple of the record length.

3. For an indexed sequential INPUT or UPDATE file, the

blocksize specified on the file declaration is less than the actual blocksize of the data set.

4. Record length is greater than 80 bytes for SYSIPT or SYSPCH.

5. For a diskette file the record length is greater than 128.

Programmer Response: Check the blocksize and record length specified in the BLKSIZE and RECSIZE option of the ENVIRONMENT attribute for the file. If the argument of either option is a variable, check that it is FIXED BINARY(31,0) STATIC and that it has been initialized.

IBM212I KEYLENGTH NEGATIVE OR GREATER THAN 255

Explanation: The KEYLENGTH option of the ENVIRONMENT attribute for this file has an invalid keylength that is greater than 255 or is negative.

Programmer Response: The argument of the KEYLENGTH option should be checked to ensure that it is either a constant or a variable with the attributes FIXED BINARY (31,0) STATIC whose value neither exceeds 255 nor is negative when the file is opened. If the argument is a variable, check that it has been correctly initialized.

IBM213I INVALID KEYLOC VALUE

Explanation:

1. The offset of the key within a record is invalid. The sum of the KEYLOC value and the key length is greater than the record length.

2. For blocked ISAM files, either KEYLOC has not been specified or KEYLOC(0) was specified. Both are invalid.

Programmer Response:

1. Check the value of the argument to the KEYLOC option. If the argument is a variable, check that it is FIXED BINARY (31,0) STATIC and that it has been correctly initialized.

2. Specify a KEYLOC value that is greater than zero.

IBM214I CONFLICTING OR INVALID ENVIRONMENT OPTIONS

Explanation: This message will be produced when a program is executed for which the compilation produced a message indicating precisely the conflicting or invalid environment options.

Programmer Response: Correct the source program.

IBM215I INVALID BUFOFF VALUE

Explanation: The values that can be specified in the BUFOFF option for an ASCII input data set are in the range 0 thru 99.

Programmer Response: Ensure that the value specified in the BUFOFF option is within the range of values given above. If the argument is a variable, ensure that it is correctly initialized.

IBM216I INVALID OFLTRACKS VALUE

Explanation: The maximum number of overflow tracks per cylinder is 9 for a model 2311, 19 for models 2314 and 2321, 18 for model 3330, and 11 for model 3340.

Programmer Response: Ensure that the number specified for the OFLTRACKS option is neither negative nor too large.

IBM217I SYMBOLIC DEVICE NAME FOR THIS FILE ASSIGNED WITH IGN OPTION

Explanation: The DOS LIOCS open routines have ignored a request by the PL/I program to open a file associated with a symbolic device name that was assigned with the IGN option.

IBM218I INVALID STACKER OPTION

Explanation: The value of the stacker option specified was not 1 or 2.

IBM225I VALUE OF ENVIRONMENT OPTION DOES NOT MATCH ACTUAL DATA SET VALUE.

Explanation: For VSAM data sets the values of KEYLOC, KEYLENGTH, and RECSIZE are specified when the data set is defined. If values are specified on any file declarations they must match the defined values.

IBM228I PASSWORD INVALID OR NOT SPECIFIED.

Explanation: For VSAM data sets defined with a password, ENV (PASSWORD) must be specified. If this password is invalid or is not specified the system operator is allowed a number of attempts to specify the correct password (the number of retries is specified when the data set is defined). If these attempts fail UNDEFINEDFILE is raised.

IBM229I NO ENTRY IN VSAM CATALOG.

Explanation: Before using a VSAM data set a catalog entry must be made and space allocated for the data set, using the Access Method Services DEFINE utility. This message may mean that ENV(VSAM) has been specified for a file, but the data set has not been converted from ISAM to VSAM.

IBM230I I/O ERROR READING CATALOG OR VOLUME LABEL.

Explanation: An I/O error prevented the reading of a VSAM catalog or a volume label.

IBM231I TIMESTAMP MISMATCH.

Explanation: For VSAM data sets the index and data can be updated separately and the time of the latest update of each is recorded. If these times do not match, the integrity of the data is uncertain. Similarly the timestamp in the data set catalog record may not match the timestamp on the volume containing the data set; this indicates the extent information in the catalog record may not agree with the extents indicated in the VTOC for the volume.

IBM232I DATA SET NOT AVAILABLE.

Explanation: The data set to be accessed is already being used by another program and is not shareable. Refer to the Programmer's Guide for further information.

IBM233I DATA SET NOT PROPERLY CLOSED.

Explanation: The last time the data set was opened the close operation failed, leaving the data set in an unusable state. Use of the access method services VERIFY utility program may restore the data set to a usable state.

IBM234I DATA SET NEVER LOADED.

Explanation: A file cannot be opened for INPUT or UPDATE to access a VSAM data set until one or more records have been loaded into the data set using a SEQUENTIAL OUTPUT file. Having once loaded records into the data set records can be added using a DIRECT UPDATE file even after all records have been deleted from the data set.

IBM235I UNIDENTIFIED ERROR DURING VSAM OPEN.

Explanation: The VSAM routines have detected an error during the open process, the cause of which cannot be determined explicitly.

Programmer Response: If the error recurs after resubmitting the job, use PLIDUMP to obtain a storage dump and retain all the relevant documentation for study by IBM.

IBM236I OPERATING SYSTEM UNABLE TO OPEN FILE ccc.

Explanation: When a VSAM data set is opened through the ISAM compatibility interface, this condition will occur if VSAM detects errors during the open process.

IBM237I SYMBOLIC UNIT NOT ASSIGNED OR INVALID.

Explanation: The address in the ASSGN statement for the logical unit was UA or IGN or the logical unit was invalid.

IBM238I ERROR IN EXTENT STATEMENT.

<u>Explanation</u>:

1. The symbolic unit specified in the EXTENT statement was invalid.
2. The volume serial numbers in the EXTENT statement do not match those in the catalog entry.
3. More than 16 extents were specified.

IBM239I INSUFFICIENT SPACE FOR MASTER INDEX OR CYLINDER INDEX.

<u>Explanation</u>:

1. The master index is not large enough to reference the prime data area.
2. The cylinder index area is not large enough to reference the prime data area.

IBM240I UNABLE TO MOUNT REQUIRED VOLUME.

<u>Explanation</u>: Either an attempt was made to mount two volumes on the same unit or the operator was unable to mount the required volume.

IBM241I 'REUSE' OPTION SPECIFIED FOR A NON-REUSEABLE DATA SET.

<u>Explanation</u>: The ENVIRONMENT option REUSE can only be specified with VSAM data sets which have been defined, during their creation by Access Method Services, as reuseable.

IBM242I ALTERNATE INDEX PATH IS EMPTY.

<u>Explanation</u>: A path can become empty by having all of its pointers deleted. Such a path cannot be opened.

IBM243I ATTEMPT TO POSITION AT LAST RECORD FAILED.

<u>Explanation</u>: When the ENVIRONMENT option BKWD is used, on opening, the file must be positioned at the last record. If the attempt to position at the last record fails, the file is closed and the UNDEFINEDFILE condition is raised with this message. This message may also be given when BKWD is specified for an empty data set.

IBM300I 'ZERODIVIDE' CONDITION RAISED BY 'SIGNAL' STATEMENT

<u>Explanation</u>: The program contains a SIGNAL statement to raise the ZERODIVIDE condition for which there is no on-unit.

<u>Programmer Response</u>: Either remove the SIGNAL statement or

include an on-unit for the ZERODIVIDE condition in the program.

IBM301I 'ZERODIVIDE' CONDITION RAISED

Explanation: The program has attempted to execute a statement in which a value of zero has been used as the divisor in a division operation.

Programmer Response: Either check the data that could produce a zero divisor before running the program or insert a ZERODIVIDE on-unit to handle the condition whenever it arises.

IBM320I 'UNDERFLOW' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the UNDERFLOW condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the UNDERFLOW condition in the program.

IBM321I 'UNDERFLOW' CONDITION RAISED

Explanation: The magnitude of a floating-point number is smaller than the permitted minimum.

IBM340I 'SIZE' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the SIZE condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the SIZE condition in the program.

IBM341I 'SIZE' CONDITION RAISED IN I/O STATEMENT

Explanation: The high-order (i.e. leftmost) significant binary or decimal digits are lost in an input/output operation where the size of the value being transmitted exceeds the declared (or default) size of the data item.

IBM342I 'SIZE' CONDITION RAISED

Explanation: The high-order (i.e., leftmost) significant binary or decimal digits are lost in an assignment to a variable or temporary variable where the size of the value being assigned exceeds the declared (or default) size of the data item.

Programmer Response: Either modify the program so that the data item is large enough for the value being assigned to it or use a SIZE on-unit to permit processing to continue when the SIZE condition is raised.

360I 'STRINGRANGE' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the STRINGRANGE condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include on on-unit for the STRINGRANGE condition in the program.

361I 'STRINGRANGE' CONDITION RAISED

Explanation: In the expression SUBSTR(S,I,J), I and J are such that the substring does not lie wholly within the string S.

Programmer Response: It should be possible to modify the source program so that this condition cannot occur.

IBM380I 'AREA' CONDITION RAISED BY THE 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the AREA condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the AREA condition in the program.

IBM381I 'AREA' ASSIGNMENT NOT EXECUTED, TARGET AREA TOO SMALL

Explanation: In an assignment of an area variable, the current extent of the area on the right-hand side of the assignment statement is greater than the size of the area to which it is to be assigned.

Programmer Response: Correct the program or insert an area on-unit to permit the target area to be allocated with fresh extents.

IBM382I NOT ENOUGH CONTIGUOUS SPACE IN THE AREA FOR ALLOCATION

Explanation: In the execution of an ALLOCATE statement, insufficient space is available in the specified area for the allocation.

Programmer Response: Provide an on-unit to permit the allocation to be reattempted. If necessary, change the value of the pointer qualifying the reference to the inadequate area so that it points to another area in which the allocation can be reattempted.

IBM400I 'CONDITION' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the CONDITION condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the CONDITION condition in the program.

IBM420I 'SUBSCRIPTRANGE' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the SUBSCRIPTRANGE condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the SUBSCRIPTRANGE condition in the program.

IBM421I 'SUBSCRIPTRANGE' CONDITION RAISED

Explanation: An array subscript has been found to have a value exceeding the declared bound for the array.

Programmer Response: In order to ensure that the program can continue to execute after encountering a subscript range error, include an on-unit for this condition with the source program, and recompile it. Note that array handling operations are slower when SUBSCRIPTRANGE is enabled than when the condition is disabled.

IBM440I 'STRINGSIZE' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the STRINGSIZE condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the STRINGSIZE condition in the program.

IBM441I 'STRINGSIZE' CONDITION RAISED

Explanation: The 'STRINGSIZE' condition is raised when a string is assigned to a shorter string, causing right-hand characters or bits in the source string to be truncated.

Programmer Response: Determine whether or not truncation of the right-hand characters or bits in the source string is correct. Use an on-unit to record the relevant data or modify the program as is necessary. Note that when STRINGSIZE is enabled, string-handling operations are slower than when the condition is disabled.

IBM460I 'OVERFLOW' CONDITION RAISED BY 'SIGNAL' STATEMENT

Explanation: The program contains a SIGNAL statement to raise the OVERFLOW condition for which there is no on-unit.

Programmer Response: Either remove the SIGNAL statement or include an on-unit for the OVERFLOW condition in the program.

IBM461I 'OVERFLOW' CONDITION RAISED

Explanation: The OVERFLOW condition occurs when the magnitude of a floating-point number exceeds the permitted maximum.

Programmer Response: Modify the program to ensure that the condition does not recur, or provide an on-unit to handle the condition if it is liable to recur.

IBM480I 'FIXEDOVERFLOW' CONDITION RAISED BY SIGNAL STATEMENT

Explanation: The FIXEDOVERFLOW condition, for which no on-unit has been provided, has been raised by the execution of a SIGNAL statement.

Programmer Response: Modify the program so that it can continue to execute either by providing an on-unit for the FIXEDOVERFLOW condition, or by removing the SIGNAL statement that raises this condition.

IBM482I 'FIXEDOVERFLOW' CONDITION RAISED

Explanation: The FIXEDOVERLOW condition occurs when the length of the result of a fixed-point arithmetic operation exceeds the permitted maximum (15 for decimal values, and 31 for binary values).

Programmer Response: Modify the program to ensure that the condition does not recur, or provide an on-unit to handle the condition if it is liable to recur.

IBM531I OPERATION EXCEPTION

Explanation: An attempt has been made to execute an instruction with an invalid System/360 or 370 operation code.

Programmer Response: It is possible that an error in the program has caused part of the executable instructions to be overwritten by data. Refer to the section on program checkout in the programmer's guide for suggestions for deleting and correcting such errors. Other possible causes of an operation exception might be an attempt to invoke an external procedure or other routine that was not incorporated into the executable program by the linkage editor or the execution of a branch instruction that has been made incorrect because a control block had previously been overwritten. Consequently, it is advisable to check the linkage editor diagnostics to ensure that all requested external procedures and subroutines have in fact been incorporated into the executable program, and any overlay phases are loaded into the correct storage locations and do not overwrite any phases that are still active.

IBM532I PRIVILEGED OPERATION EXCEPTION

Explanation: An attempt has been made to execute certain System/360 or 370 instructions which can only be executed by the supervisor program. This condition can only be raised for a PL/I program which includes a non-PL/I routine that contains such an instruction or in which an error has occurred causing an executable instruction in the program to be overwritten with data that is identical to one of the privileged instructions.

Programmer Response: If the error is not in a non-PL/I routine included in the executable program, the PL/I program should be checked for an error that could cause the executable instructions to be overwritten by data that matches a privileged operation. The section on program checkout in the programmer's guide contains suggestions for detecting and correcting such errors.

IBM533I EXECUTE EXCEPTION

Explanation: An attempt has been made to use an IBM System/360 or 370 EXECUTE instruction to execute another EXECUTE instruction. This can occur if a routine that contains this error has been included in the PL/I program, or if an executable instruction that is the subject of a compiler-generated EXECUTE instruction has been overwritten by data that matches the operation code for the EXECUTE instruction.

Programmer Response: If the error is not in a non-PL/I routine included in the executable program, the PL/I program should be checked for an error that could cause the executable instruction to be overwritten by data that matches the operation code for the EXECUTE instruction on. The section on program checkout in the programmer's guide contains suggestions for detecting and correcting such errors.

IBM534I PROTECTION EXCEPTION

Explanation: An attempt has been made to store data in main storage that is outside the partition allocated to the program.

Programmer Response: If the error is not in a non-PL/I routine included in the executable program, the PL/I program should be checked for an error that could cause the address used by the store instruction to be corrupted. The section on program checkout in the programmer's guide contains suggestions for detecting and correcting such errors.

IBM535I ADDRESSING EXCEPTION

Explanation: An invalid address has been supplied as an operand to an IBM System/360 or 370 instruction.

Programmer Response: If the error is not in a non-PL/I routine included in the executable program, the PL/I program should be checked for an error that could cause the address to be corrupted. The section on program checkout in the programmer's guide contains suggestions for detecting and correcting such errors.

IBM536I SPECIFICATION EXCEPTION

Explanation: An alignment error in the operands of an IBM System/360 or 370 instruction, or an error in the specification of the operands, has occurred.

Programmer Response: If the error is not in a non-PL/I routine included in the executable program, the PL/I program should be checked for an error that could cause the operand to be corrupted by overwriting control blocks or sections of executable code. The section on program checkout in the programmer's guide contains suggestions for detecting and correcting such errors.

IBM537I DATA EXCEPTION

Explanation: An attempt has been made to process FIXED DECIMAL data that is not in the correct format.

Programmer Response: The PL/I program should be checked for an error such as an operation on a FIXED DECIMAL data item before it has been initialized, or an error which could cause the data item to be overwritten. Refer to the chapter on program checkout in the programmer's guide for hints on how to trace such errors.

IBM538I OPERATION EXCEPTION. FLOATING POINT INSTRUCTIONS NOT SUPPORTED

Explantion: An attempt has been made to execute a floating-point instruction on a machine that does not have hardware facilities for floating-point arithmetic. The floating-point instruction is contained either in instructions generated by the compiler, or in a non-PL/I routine in this program.

IBM560I EVENT VARIABLE AS ARG TO CPLN P-V ALREADY IN USE WITH FILE XXX

Explanation: The event variable used in this statement is already active and is associated with an input/output operation on the named file.

Programmer Response: Modify the program so that the COMPLETION pseudovariable refers to the event variable when it is inactive when the statement is executed.

IBM562I EVENT VARIABLE AS ARG TO CPLN P-V ALREADY IN USE WITH 'DISPLAY' STMT

Explanation: The event variable used in this statement is already active and is associated with a DISPLAY statement.

Programmer Response: Modify the program so that the COMPLETION pseudovariable refers to the event variable when it is inactive when the statement is executed.

IBM563I EVENT VARIABLE ALREADY IN USE WITH 'FILE' XXX

Explanation: The event variable used in this statement is already active and is associated with another input/output operation on the named file.

Programmer Response: Modify the program so that the input/output operation refers to another event variable, or include a WAIT statement to prevent execution of the statement until the active event is complete.

IBM564I EVENT VARIABLE ASSIGNED TO, ALREADY IN USE WITH 'FILE' XXX

Explanation: An attempt has been made to assign a value to an event variable while it is still associated with an input/output operation.

Programmer Response: Modify the program so that the event variable used as the target in the assignment, or as the argument of the COMPLETION pseudovariable is not the same event variable associated with an input/output operation. Alternatively, include a WAIT statement to prevent execution of this statement until the active event is complete.

IBM567I WAIT IN ON-UNIT FOR I/O EVENT BEING WAITED FOR

Explanation: This error is caused when a WAIT statement specifies an event variable and the completion of the event causes entry to an on-unit for an I/O condition which contains another WAIT statement for the same event variable.

For example:

```
ON RECORD(F) BEGIN;
                    .
                    .
                    .
                    WAIT(E);
                    .
                    .
                    .
                    END;

WRITE FILE(F)...EVENT(E);
WAIT(E);  /* THIS STATEMENT RAISES
                        THE RECORD CONDITION */
```

Programmer Response: Remove the WAIT statement from the on-unit for the input/output condition.

IBM568I EVENT VARIABLE ASSIGNED TO, ALREADY IN USE WITH 'DISPLAY' STATEMENT

Explanation: The event variable specified as the argument of the COMPLETION built-in function or used as the target in an assignment is still associated with a DISPLAY statement.

Programmer Response: Modify the program so that the event variable used as the target in the assignment or as the argument of the COMPLETION pseudovariable is not the same event variable associated with the DISPLAY statement. Alternatively, include a WAIT statement to prevent execution of this statement until the active event is complete.

IBM571I EVENT VARIABLE ALREADY IN USE WITH 'DISPLAY' STATEMENT

Explanation: The event variable specified in the statement is already associated with a DISPLAY statement.

Programmer Response: Either use a different event variable or insert a WAIT statement so that the DISPLAY statement is complete before this statement is executed.

IBM600I INCORRECT VALUES FOR W,P,S FIELDS IN E-FORMAT SPECIFICATION

Explanation: An edit-directed input/output operation for an E-format item has been specified incorrectly.

Programmer Response: Correct the E-format item according to the language rules.

IBM601I VALUE OF W FIELD TOO SMALL IN F-FORMAT SPECIFICATION

Explanation: An edit-directed input/output operation for an F-format item has been specified incorrectly with a W-specification that is too small to allow room for the decimal-point when the number of fractional digits is specified as zero.

Programmer Response: Correct the F-format item according to the language rules.

IBM604I INVALID ASSIGNMENT TO PICTURED CHARACTER STRING

Explanation: A data item which is not a character string cannot be assigned to a pictured character string because it does not match the declared characteristics of the pictured target variable.

Programmer Response: Modify the program so that the assignment can be possible by altering the characteristics either of the source variable or of the target variable.

IBM605I ITERATION FACTOR IN FORMAT LIST OR DEPTH OF R-FORMAT NESTING TOO LARGE

IBM606I INVALID REMOTE FORMAT ITEM IN FORMAT LIST

IBM607I REMOTE FORMAT STATEMENT IS OUTSIDE THE CURRENT BLOCK

IBM608I LABEL VARIABLE IN R-FORMAT ITEM DOES NOT REFER TO R-FORMAT STATEMENT LABEL

IBM650I SOURCE NOT MODIFIED IN CONVERSION ON-UNIT, RETRY NOT ATTEMPTED

Explanation: The CONVERSION condition has been raised by the presence of an invalid character in the string to be converted. The character has not been corrected in an on-unit, either by the ONCHAR or by the ONSOURCE pseudovariable.

Programmer Response: Modify the CONVERSION on-unit to use either the ONCHAR or ONSOURCE pseudovariable to assign a valid character to replace the invalid character in the source string.

IBM651I 'ONSOURCE' OR 'ONCHAR' PSEUDOVARIABLE USED OUT OF CONTEXT

Explanation: Either ONSOURCE or ONCHAR has been used as a pseudovariable in a statement not contained in a CONVERSION on-unit or in a block that is dynamically-descendent from such an on-unit.

Programmer Response: Modify the source program so that ONCHAR and ONSOURCE are used as pseudovariables in the correct context.

IBM670I X LT 0 IN SQRT(X)

Explanation: The built-in function SQRT has been invoked with an argument that is less than zero.

On-codes associated with this message are:

1500 Short floating-point SQRT error
1501 Long floating-point SQRT error

Programmer Response: Modify the program so that the argument of the SQRT built-in function can never be less than zero.

IBM671I X LE 0 IN LOG(X), LOG2(X), OR LOG10(X)

Explanation: One of the built-in functions, LOG, LOG2, or LOG10, has been invoked with an argument that is less than or equal to zero. The invocation may have been direct or as part of the evaluation of an exponentiation calculation.

On-codes associated with this message are:

1504 Short floating-point Log, Log 2 or Log 10 error

1505 Long floating-point Log, Log 2 or Log 10 error.

Programmer Response: If a direct invocation, then modify the program so that the argument of the LOG, LOG2, or LOG10 built-in function is greater than zero. If part of an exponentiation, then ensure that the argument is greater than zero.

IBM672I ABS(X) TOO LARGE IN SIN(X), COS(X), SIND(X), COSD(X), TAN(X), OR TAND(X)

Explanation: The argument passed to TAN, TAND, SIN, SIND, COS, or COSD exceeds the limit specified below:

Floating-Point Precision	Limit		
Binary p ≤ 21 Decimal p ≤ 6	X < (2**18) * K	where	K = pi for X in radians (SIN, COS, or TAN)
Binary 21 < p ≤ 53 Decimal 6 < p ≤ 16	X < (2**50) * K	or	K = 180 for X in degrees (SIND, COSD, TAND)

The error has arisen during one of the following:

1. The evaluation of SIN, SIND, COS, COSD, TAN, or TAND when invoked implicitly.

2. The evaluation of TAN, when invoked during the evaluation of TAN or TANH with a complex argument.

3. The evaluation of SIN or COS, when invoked during the evaluation of EXP, SIN, SINH, COS, COSH, TAN or TANH with a complex argument.

4. The evaluation of a general exponentiation function with complex arguments.

On-codes associated with this message are:

1506 Short floating-point SIN, SIND, COS or COSD error
1507 Long floating-point SIN, SIND, COS or COSD error
1508 Short floating-point TAN or TAND error
1509 Long floating-point TAN or TAND error

IBM674I X=Y=O IN ATAN(Y,X) OR ATAND(Y,X)

Explanation: Two arguments, both zero, have been given for the ATAN or ATAND built-in function. ATAN or ATAND has been invoked either directly with a real argument or indirectly in the evaluation of the LOG built-in function with a complex argument.

On-codes associated with this message are:

1510 Short floating-point ATAN or ATAND error
1511 Long floating-point ATAN or ATAND error

Programmer Response: Modify the program so that the arguments of ATAN or ATAND are not both zero.

IBM675I ABS(X) GE 1 IN ATANH(X)

Explanation: The ATANH built-in function has been used with a floating-point argument with an absolute value that equals or exceeds 1.

On-codes associated with this message are:

1514 Short floating-point ATANH error
1515 Long floating-point ATANH error

Programmer Response: Modify the program so that the absolute value of a floating-point assignment to the ATANH built-in function does not equal or exceed 1.

IBM676I ABS(X) GT 1 IN ASIN(X) OR ACOS(X)

Explanation: The absolute value of the floating-point argument of the ASIN or ACOS built-in function exceeds 1.

On-codes associated with this message are:

1518 Short floating-point ASIN or ACOS error
1519 Long floating-point ASIN or ACOS error

Programmer Response: Modify the program so that the ASIN or ACOS built-in function is never invoked with a floating-point argument whose absolute value exceeds 1.

IBM700I ATTEMPT TO ASSIGN TO UNALLOCATED CONTROLLED VARIABLE IN GET DATA FOR FILE FILENAME

Example:

```
DCL X CONTROLLED FIXED BIN;
GET DATA(X);

Input stream contains: 'x=5;.....'
```

Explanation: A variable occurring in the stream for a GET FILE DATA is CONTROLLED, but has no current allocation.

IBM701I ATTEMPT TO ASSIGN TO UNALLOCATED CONTROLLED VARIABLE IN GET DATA

Example:

```
DCL STR CHAR(4)INIT('X=5;')
  X CONTROLLED FIXED BIN;
GET STRING(STR) DATA(X);
```

Explanation: A variable occurring in the string for a GET STRING DATA is CONTROLLED, but has no current allocation.

IBM722I X=O AND Y NOT REAL AND POSITIVE IN X ** Y

Explanation: In an exponentiation operation the floating-point base is zero and the exponent is not positive and real.

On-codes associated with this message are:

1550 Short floating-point real base with integer exponent.
1551 Long floating-point real base with integer exponent.
1552 Short floating-point real base with floating-point exponent.
1553 Long floating-point real base with floating-point exponent.
1554 Short floating-point complex base with integer exponent.
1555 Long floating-point complex base with integer exponent.
1556 Short floating-point complex base with complex exponent.
1557 Long floating-point complex base with complex exponent.

Programmer Response: Modify the program so that the exponentiation operation involves a non-zero floating-point base or a positive real exponent.

IBM724I Z = +1I OR -1I IN ATAN(Z) OR Z=+1 OR -1 IN ATANH(Z)

Explanation: The complex floating-point argument of the ATAN built-in function has the value of +1I or -1I. Alternatively, the complex floating-point argument of the ATANH built-in function has the value +1 or -1.

On-codes associated with this message are:

1558 Short floating-point complex ATAN or ATANH error 1559 Long floating-point complex ATAN or ATANH error

Programmer Response: Modify the program so that the complex floating-point argument of ATAN can never be +1I or -1I, or the complex floating-point argument of the ATANH built-in function never has the value +1 or -1.

IBM750I 'GOTO' TO AN INVALID BLOCK ATTEMPTED

Example:

```
DCL L LABEL;
    BEGIN;
 A: L = A;
    END;
    GOTO L;
```

Explanation: 1. A GOTO statement that transfers control to a label variable is invalid because the generation of the block that was active when the label variable was assigned was no longer active when the GOTO statement was executed.

2. A GOTO label variable statement was executed, but the label variable was uninitialised or contained an invalid address.

Programmer Response: 1. Modify the program so that the GOTO statement transfers control to a label variable that was assigned in a block that is still active.

2. Initialise the call variable correctly.

IBM772I 'WAIT' WITH MULTIPLE 'EVENTS' NOT IN THIS SYSTEM

Explanation: A WAIT statement with more than one event variable has been encountered. The DOS PL/I Transient Library for this system was generated to handle WAIT statements for single events only.

Programmer Response: Modify the program so that the WAIT statement specifies one event only.

IBM780I NO 'OTHERWISE' CLAUSE AND NO 'WHEN' CLAUSES SATISFIED.

Explanation: It is an error if no WHEN clauses of a SELECT statement are selected and no OTHERWISE clause exits.

IBM802I GET/PUT STRING EXCEEDS STRING SIZE

Explanation: For input, a GET statement has attempted to access data that exceeds the length of the source string. For output, a PUT statement has attempted to assign data that is longer than the target string.

IBM803I FURTHER OUTPUT PREVENTED BY PRIOR CONDITION FOR FILE FILENAME

Explanation: A PL/I WRITE, LOCATE, or PUT statement has been issued for either a buffered output file on magnetic tape or an indexed data set, when a previous attempt to transmit a record raised the TRANSMIT condition.

IBM804I 'PRINT' OPTION/FORMAT ITEM USED WITH NON-'PRINT' FILE FILENAME

Explanation: An attempt has been made to use one of the option PAGE or LINE.

IBM805I 'DISPLAY' WITH 'REPLY' OPTION HAS ZERO LENGTH STRING

Explanation: The current length of the character string to be displayed or the maximum length of the character string to which the reply is to be assigned is zero.

IBM806I LAST 'READ' BEFORE THIS 'REWRITE' IS INCOMPLETE FOR FILE FILENAME

Explanation: An attempt has been made to execute a REWRITE statement before a preceding READ statement with the EVENT option for the same file has been completed.

Programmer Response: Modify the program so that the REWRITE statement is executed after completion of the READ statement by inserting a WAIT statement for the given event variable into the flow of control between the two statements.

IBM807I NO PRECEDING 'READ SET' OR 'READ INTO' FOR 'REWRITE' ON FILE FILENAME

Explanation: A REWRITE statement without the KEY option has been executed when the last input/output operation on the file was not a READ statement with the INTO or SET option or was a READ statement with the IGNORE option.

IBM808I INVALID ELEMENT VARIABLE IN STRING FOR 'GET STRING DATA'

Explanation: The identifier in the string named in the STRING option of a GET STRING DATA statement does not match the identifier in the data specification. Note that the DATAFIELD built-in function will not return a value in this case.

Programmer Response: Modify the program so that the string contains the identifier in the data specification.

IBM809I INVALID FILE OPERATION FOR FILE FILENAME

Explanation: An attempt has been made to carry out an operation on a file that is impossible from the file declaration. For example, it is not possible to execute a REWRITE statement on a STREAM file, read an output file, or write an input file. A list of other possible conflicts follows:

Statement and Option	Conflicting File Attribute or Organization
Any record I/O statement	STREAM
Any stream I/O statement	RECORD
READ SET	UNBUFFERED
READ EVENT	BUFFERED
READ KEY	REGIONAL SEQUENTIAL or CONSECUTIVE
READ IGNORE	DIRECT
WRITE	INDEXED or VSAM SEQUENTIAL UPDATE
	INDEXED DIRECT NOWRITE
	REGIONAL (not KEYED)
WRITE EVENT	BUFFERED
REWRITE	INPUT or OUTPUT
REWRITE (without FROM)	UNBUFFERED or DIRECT
REWRITE KEY	SEQUENTIAL
REWRITE EVENT	BUFFERED
DELETE	INPUT or OUTPUT
	VSAM (ESDS)
LOCATE	INPUT or UPDATE
	UNBUFFERED
	DIRECT
LOCATE KEYFROM	INDEXED or REGIONAL (without KEYED) or VSAM (ESDS)
GET	OUTPUT
PUT	INPUT

Programmer Response: Ensure that the file declaration and the input/output statements for that file are compatible.

IBM811I I/O ERROR. CAUSE NOT KNOWN FOR FILE FILENAME

Explanation: The data management routines have detected an error during an input/output operation, the cause of which could not be determined explicitly.

Programmer Response: If the error recurs after resubmitting the job, use PLIDUMP to obtain a storage dump and retain all the relevant documentation for study by IBM.

IBM812I NO PRECEDING 'READ SET' OR 'READ INTO' FOR 'REWRITE' ON FILE FILENAME

Explanation: A REWRITE statement has been executed for which no preceding READ statement, either with the INTO option or with the SET option, has been executed.

Programmer Response: Modify the program so that either the REWRITE statement is preceded by a READ statement or the REWRITE statement is replaced by a WRITE statement, according to the requirements of the program.

IBM813I LAST 'READ' BEFORE THIS 'REWRITE' IS INCOMPLETE FOR FILE FILENAME

Explanation: A REWRITE statement has been attempted before the preceding READ statement with the EVENT option has been completed.

Programmer Response: Include a WAIT statement so that the READ statement is complete before the REWRITE statement is executed.

IBM814I TOO MANY INCOMPLETE I/O OPERATIONS

Explanation: An attempt has been made to initiate an input/output operation beyond the limit imposed by the operating system.

Programmer Response: Modify the program so that the input/output operation is not initiated until an incomplete input/output operation has been completed.

IBM816I IMPLICIT 'OPEN' UNSUCCESSFUL FOR FILE FILENAME

Explanation: An error has occurred during the implicit opening of a file. The UNDEFINEDFILE condition was raised and a normal return was made from the associated on-unit, but the file was still unopened.

Programmer Response: Ensure that the file has been completely and correctly declared, and that the input/output statement that implcitly opens the file is not in conflict with the file declaration.

IBM818I UNEXPECTED END OF FILE STRING DETECTED IN STREAM INPUT

Explanation: The end of the file has been detected before the completion of a GET FILE statement.

Programmer Response: For edit-directed input, ensure that the last item of data in the stream has the same number of characters as specified in the associated format item. If the error occurs during execution of an X-format item, ensure that the same number of characters to be skipped are present before the last data item in the stream.

For list-directed and data-directed input, ensure that the last item of data in the data set, if a string, is terminated by a quote character and a 'B', if a bit string, that precedes the end-of-file marker.

IBM821I I/O STATEMENT OCCURRED BEFORE PREVIOUS 'READ' COMPLETED BY 'WAIT' FOR FILE FILENAME

Explanation: While an indexed sequential file was open for direct updating, an input/output statement was attempted before the completion of a previous READ statement with the EVENT option.

Programmer Response: Include a WAIT statement so that the erroneous input/output statement cannot be executed until the completion of the previous READ statement with the EVENT option.

IBM822I NO SPACE FOR RECORD IN SEQUENTIAL OUTPUT DATA SET. ASSOCIATED WITH THE FILENAME

Explanation: There is insufficient space on the data set in which to add the record transmitted by this statement. For regional data sets, this can be caused by a WRITE or LOCATE statement that specifies a region that is beyond the limit established for the data set.

Programmer Response: Enlarge the extent made available to the data set. For regional data sets, check that the region used in a WRITE or LOCATE statement can be contained in the enlarged data set.

IBM823I INVALID CONTROL FORMAT ITEM FOR GET/PUT STRING.

Explanation: An invalid control format item (PAGE, LINE, SKIP, or COL) has been detected in a remote format list for a GET or PUT STRING statement. For example:

```
          DCL(A,B) CHAR(10),
                C CHAR(80);
          F:  FORMAT(A(10), SKIP,A(10));
              A='FRED'; B = 'HARRY';
              PUT STRING(C) EDIT(A,B) (R(F));
```

Programmer Response: Modify the source program so that GET or PUT STRING statements do not attempt to use invalid control format items in remote format lists.

IBM825I EVENT VARIABLE ALREADY IN USE WITH FILE

Explanation: An input/output statement with an EVENT option has been attempted while a previous input/output statement with an EVENT option that uses the same event variable is still incomplete.

Programmer Response: Either change the event variable used in the second EVENT option or insert a WAIT statement for the event variable between the two input/output statements.

IBM826I EVENT VARIABLE ALREADY IN USE WITH DISPLAY STATEMENT

Explanation: An input/output statement with an EVENT option has been attempted while a previous DISPLAY statement with an EVENT option that uses the same event variable is still incomplete.

Programmer Response: Either change the event variable used in the second EVENT option or insert a WAIT statement for the event variable between the DISPLAY statement and the input/output statement.

IBM828I INCORRECT SEQUENCE OF I/O OPERATIONS ON AN ASSOCIATED FILE

Explanation: Operations on a set of associated files were not carried out in the correct sequence, as follows:

1. Appropriate I/O operations were not carried out in the sequence Read-Punch-Print. Only the Print operation may be omitted.

2. An attempt was made to print more than the maximum number of lines on a card, using a print file that was associated with a read or a punch file.

IBM829I INSUFFICIENT VIRTUAL STORAGE AVAILABLE TO VSAM.

Explanation: During an OPEN/CLOSE or any other operation on a VSAM data set insufficient virtual storage was available for workspace and control blocks or for VSAM routines. Insufficient virtual storage may occur in the partition, or in the shared virtual area (SVA).

IBM830I I/O ERROR DURING 'CLOSE'.

Explanation: An I/O error occurred while a VSAM close routine was either reading or writing a catalog record, or completing an outstanding I/O request.

IBM831I NO POSITIONING ESTABLISHED FOR SEQUENTIAL READ.

Explanation: A READ statement without the KEY option has been attempted on a VSAM data set, after sequential positioning has been lost as the result of a previous error during sequential processing (for example, read error on index set).

IBM832I INSUFFICIENT SPACE FOR VSAM DATA SET.

Explanation: VSAM has been unable to allocate additional DASD space for the data set (ESDS or KSDS). The condition is raised on attempting to write or locate a record during the sequential creation or extension of such a data set when the space allocated to the data set is full. For a KSDS, the condition may also occur when the associated PL/I file is opened for update and attempts are made to write new records or to increase the size of existing records by the WRITE and REWRITE statements respectively.

IBM834I REQUESTED RECORD LIES ON NON-MOUNTED VOLUME.

Explanation: The requested record lies on a non-mounted volume of a VSAM data set spanning several volumes.

IBM835I ATTEMPT TO REPOSITION FOR SEQUENTIAL READ FAILED.

Explanation: The attempt to reposition to the next highest key for subsequent sequential retrieval on a VSAM KSDS, after the 'key not found' condition, has failed. If processing of the file is continued, the next I/O statement should have a positioning KEY option.

IBM836I TOO MANY CONCURRENT OPERATIONS ON DATA SET.

Explanation: Several files may be accessing a VSAM data set by means of the same DLBL statement (that is, using the same title), but only one I/O operation may be active on the data set at any given time. In addition, a READ-REWRITE pair on a SEQUENTIAL UPDATE file counts as one operation, and also leaves a position established.

IBM837I ERROR IN INDEX UPGRADE.

Explanation: A change to a base cluster cannot be reflected in one of the indexes of the cluster's upgrade set.

IBM838I MAXIMUM NUMBER OF ALTERNATE INDEX POINTERS EXCEEDED.

Explanation: The maximum number of pointers allowed in an alternate index depends on the record length specified for the alternate index, but cannot exceed an overall maximum of 32767.

IBM839I INVALID ALTERNATE INDEX POINTER.

Explanation: A pointer in the alternate index is invalid. This can be caused by incorrect use of the alternate index as a Key Sequenced Data Set (KSDS).

IBM840I INVALID SEQUENTIAL WRITE.

Explanation: A WRITE statement on a file associated with a Relative Record Data Set (RRDS) did not specify a relative record number. This resulted in an attempt to write in a slot already containing a record.

IBM850I AGGREGATE LENGTH EXCEEDS 2**24 BYTES

Explanation: The length of the structure or array to be mapped is greater than 2^{24} thereby exceeding the limits of addressability.

Programmer Response: Reduce the size of the array or structure to a size that can be accommodated within the main storage available. If a variable is used to specify the dimension or length, check that it has been correctly initialized before the storage is allocated to the aggregate.

IBM851I UNABLE TO MAP ARRAY STRUCTURE ELEMENT

Explanation: The program contains a structure with an adjustable element and an array element with extents that cause the relative virtual origin to exceed $(2^{32})-1$. For example:

```
DCL 1 A,
      2 B CHAR(N),
      2 C (32766:32767,32766:32767,32766:32767)
         CHAR(32767);
```

Programmer Response: Ensure that aggregates with array elements remain within the limit of addressability (2^{32}).

IBM852I AGGREGATE CANNOT BE MAPPED IN COBOL OR FORTRAN

Explanation: An attempt has been made to either pass to or obtain from a FORTRAN routine an array of more than 7 dimensions, or to pass to or obtain from a COBOL routine a structure with more than three levels.

Programmer Response: Ensure that PL/I aggregates, passed to or from COBOL or FORTRAN routines are within the limits described above.

IBM880I A PROGRAM CHECK HAS OCCURRED IN THE SORT/MERGE PROGRAM

Explanation: An error has occurred during execution of the sort/merge program when invoked from a PL/I program by means of the PL/I sort interface facilities. The sort program was unable to continue and control has been passed to the PL/I error-handler.

Programmer Response: Since the problem has occurred during execution of the sort/merge program, refer to the appropriate sort/merge program manual for an explanation of any diagnostic messages produced by the sort program and for any other information that may be necessary to correct the fault.

IBM900I 'WAIT' STATEMENT WOULD CAUSE PERMANENT WAIT. PROGRAM TERMINATED

Example:

```
        COMPLETION (E1) = '0'B;
              WAIT(E1);
```

Explanation: A WAIT statement that can never be completed has been encountered. In the example, the event E1 is inactive and incomplete.

Programmer Response: Modify the program so that the WAIT statement can never wait for an event that is inactive and incomplete.

IBM921I GOTO OUT OF ON-UNIT MAY CAUSE FURTHER USE OF THIS COBOL SUBROUTINE TO FAIL

Explanation: If a COBOL subroutine is reinvoked after an interrupt in the previous invocation was handled by a PL/I on-unit that was terminated by a GOTO statement, the COBOL subroutine will fail.

IBM9251 PLIRETC VALUE REDUCED TO 999.

Explanation: The value passed to the PLIRETC built-in procedure is greater than 999. The value has been reduced to 999 which is the maximum permitted user value.

SC33-0035-2

DOS PL/I Transient Library: Messages File No. S360/S370-29 Printed in U.S.A. SC33-0035-2

IBM®